

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Tím č.5

## **Interaktivita mobilného zariadenia a televízie**

Tímový projekt

Študijný program: Počítačové a komunikačné systémy a siete

Študijný odbor: 9.2.4 Počítačové inžinierstvo

Miesto vypracovania: Ústav počítačových systémov a sietí, FIIT STU Bratislava

Členovia tímu:

Bc. Filip Filip

Bc. Koren Jan

Bc. Meliš Peter

Bc. Ondruš Vladimír

LS 2013/2014

## Obsah

Úvod.....	3
1. Analýza.....	4
1.1 Analýza dostupného technického a programového vybavenia.....	4
1.1.1. AzBox Hd Premium Plus HDTV satelitný prijímač .....	4
1.1.2. A-Link DTU .....	6
1.1.3. OpenRsi .....	6
1.1.4. Možnosti OpenRsi.....	7
1.2 Analýza technológií, protokolov a aplikácií prepojenia TV a mobilného zariadenia .	8
1.2.1. Chromecast.....	8
1.2.2. Google TV pairing protocol .....	9
1.2.3. AirPlay .....	10
1.2.4. DIAL API.....	11
1.2.5. UPnP protokol .....	12
1.2.6. Dostupné aplikácie .....	13
1.2.7. Streaming audiovizuálneho obsahu.....	14
1.2.7.1. Prenos audiovizuálneho obsahu po sieti.....	14
1.2.7.2. Audiovizuálne súbory a formáty .....	16
1.2.7.3. Protokoly .....	17
1.2.7.4. Programy pre streaming audiovizuálneho obsahu.....	18
2. Prototyp.....	20
2.1. Mobilná aplikácia (platforma Android).....	20
2.1.1. Smart Connection .....	20
2.1.2. Ping.....	22
2.1.3. Stream.....	23
2.1.4. Send http request .....	24
2.1.5. Playlist.....	26
2.1.6. Diagram tried.....	27
2.1.7. Zhodnotenie mobilnej aplikácie .....	27
2.2. Streamovací server .....	28
2.2.1. Server .....	28
2.2.2. Prijem digitálneho signálu.....	28
2.2.3. Streamovanie .....	29
2.2.4. Ďalší vývoj streamovacieho servera.....	30
2.2.5. Zhodnotenie streamovacieho servera.....	31

3.	Špecifikácia požiadaviek.....	33
4.	Návrh riešenia .....	35
4.1.	Serverová aplikácia.....	36
4.2.	Aplikácia pre platformu Android .....	37
4.3.	Web aplikácia .....	37
4.4.	Databáza .....	37
4.5.	Prípady použitia.....	38
4.6.	Opis komunikácie .....	39
4.6.1.	Autorizácia do systému .....	39
4.6.2.	Spustenie televízneho vysielania.....	40
4.6.3.	Prenesenie televízneho vysielania na iné zariadenie.....	41
4.6.4.	Zobrazenie elektronického programového sprievodcu .....	41
4.6.5.	Zobrazenie doplnujúcich informácií o programe .....	42
5.	Implementácia riešenia.....	43
5.1.	Serverová aplikácia.....	43
5.1.1.	Knižnica Vlcj.....	45
5.1.2.	Knižnica Gson .....	45
5.1.3.	Log4j .....	46
5.1.4.	OMDbAPI.....	46
5.2.	Aplikácia pre platformu Android .....	46
5.2.1.	Spustenie prehrávania živého televízneho obsahu .....	47
5.2.2.	Spustenie streamu na inom zariadení .....	47
5.2.3.	Zobrazenie elektronického programového sprievodcu .....	48
5.3.	Aplikácia pre webové rozhranie .....	48
5.4.	Databáza .....	50
6.	Overenie riešenia.....	52
7.	Zhodnotenie.....	53
7.1.	Zhodnotenie požiadaviek.....	53
7.1.1.	Funkcionálne požiadavky.....	53
7.1.2.	Požiadavky pre server .....	53
7.1.3.	Požiadavky pre mobilnú aplikáciu .....	54
7.1.4.	Požiadavky pre webovú aplikáciu.....	54
7.1.5.	Nefunkcionálne požiadavky .....	54
7.2.	Čo sme nestihli .....	55
7.3.	Čo sme sa naučili .....	56
7.4.	Celkové zhodnotenie .....	56
8.	Zdroje .....	58



# Úvod

---

Tento dokument vznikol v rámci predmetu Tímový projekt na STU FIIT. V nasledujúcom dokumente sa nachádza analýza a implementáciu prototypu tímového projektu.

## **Zadanie Projektu**

Mobilné zariadenia máme stále so sebou, pomáhajú nám v rôznych situáciách. Čo tak ich využiť aj pri sledovaní filmu v obývačke? Navrhnite a zrealizujte aplikáciu/prostredie, ktoré umožní používateľovi pri odchode z miestnosti odniesť si zo sebou aj sledovaný TV program a sledovať ho ďalej na mobilnom zariadení. Opačne, pri sledovaní TV umožní používateľovi pozrieť si dodatkové informácie k sledovanému programu, prípadne ich gestom preniesť z mobilu/tabletu na TV. Ovládanie TV pomocou mobilného zariadenia je vítanou súčasťou riešenia.

## **Motivácia**

Cieľom nášho tímového projektu je navrhnúť riešenie pre vzájomnú interakciu medzi TV a mobilným zariadením (tablet, smartfón). Výsledný produkt by nám mal poskytnúť možnosť vziať si so sebou sledovaný televízny program kdekoľvek so sebou, mal by nám poskytnúť určitú pridanú hodnotu pri sledovaní TV programu doma v obývačke, ako napríklad dodatočné informácie o programe, alebo ich gestom preniesť na obrazovku nášho televízora, prípadne ovládanie funkcií televízora priamo z mobilného zariadenia.

# 1. Analýza

---

V tejto kapitole sú opísané hlavné oblasti pre vypracovanie projektu.

## 1.1 Analýza dostupného technického a programového vybavenia

V tejto podkapitole analyzujeme dostupné technické produkty a protokoly na prenos signálu.

### 1.1.1. AzBox Hd Premium Plus HDTV satelitný prijímač

AzBox je set-top-box umožňujúci prijímať televízny signál prostredníctvom zabudovaného DVB-T tunera. Tento box má v sebe zabudovanú podporu pre full HD formát, poskytuje viaceré rozhrania a taktiež prístup do siete, či už cez WiFi alebo Ethernet. Medzi integrovanými rozhraniami sa nachádzajú dva sloty pre tunery, pri ktorých je možné použiť rôznu kombináciu typov tunerov. Predinštalovaný prehrávač médií je kompatibilný s viacerými formátmi súborov, napríklad MP3 alebo MKV.

Model Premium má v sebe zabudovanú širokú škálu konektorov poskytujúcich viacero možností konektivity. Oplatí sa spomenúť 2 USB konektory, jeden v prednej časti boxu a druhý v zadnej časti. Konektor v zadnej časti slúži aj pre pripojenie externého HDD, ktorý môže byť použitý na nahrávanie programov alebo ako interný disk pre TimeShifting funkciu.

TimeShifting funkcia slúži na zastavenie a prípadné posunutie práve prehrávaného programu. Okrem pripojenia externého HDD je možné do boxu zapojiť aj interný disk cez rozhranie SATA do veľkosti 1TB.

Prístupné sú tiež HDMI slot pre prehrávanie vo formáte full HD, digitálny optický výstup, video a stereo RCA konektor a v neposlednom rade aj scart konektor. Poskytovaním všetkých týchto rozhraní je AzBox vhodný pre akýkoľvek prenos digitálneho televízneho signálu.

Zabudovaný WiFi modul s podporou 802.11 b/g umožňuje prístup do siete a konektivitu mobilných zariadení s vhodnými aplikáciami. Niektoré z týchto aplikácií, spomenuté v práci, boli úspešne otestované. Tento modul podporuje všetky typy bezdrôtového zakódovania signálu od WEP po WPA2.

### **Technické parametre zariadenia:**

- Operačný systém Linux
- Procesor Sigma Designs SMP8634LF MIPS-CPU 300MHz
- Systémová pamäť DDR 128MB
- Video pamäť DDR 128MB
- Tuner DVB-S2-Tuner/optional 2. Tuner DVBS2, DVB-T, DVB-C
- DiSEqC 1.0, 1.2, 1.3 USALS
- Flash 8MB, DOM 256MB
- Video PAL/ NTSC
- Video Decoder MPEG-1, MPEG-2, MPEG-4.2, MPEG4.10(H.264), WMV9, VC-1,
- Divx 3/4/5, Xvid
- Audio Decoder Dolby AC3, DTS, MPEG 1 Layer 1/2/3 (MP3), AAC, WMA9, OGG, FLAC
- Rozlíšenie 576i (480i), 576p (480p), 720p, 1080i, 1080p
- WiFi 802.11b/g miniPCI
- Sieť10/100 Base-T Ethernet
- Predný panel 1x čítačka kariet
- 2x CI-Slot pre dekódovacie moduly
- 1 x USB 2.0
- VFD Displej
- IR Modul
- Zadný panel 1 x HDMI
- 1 x Component (YCbCr)
- 1 x RCA Video
- 1 x RCA Audio (L/R)
- 1 x RJ45
- 1 x USB 2.0
- 1 x Optický audio výstup S/PDIF
- 1 x SCART
- Sieťový vypínač

- Rozmery 340 x 243 x 66 (mm)
- Napájanie AC 12V 3,4A / 24V 0.8A
- Spotreba max. 60W
- Standby 5W

### 1.1.2. A-Link DTU

Okrem AzBoxu máme k dispozícii USB TV prijímač značky A-Link.

#### Technické parametre:

- HDTV príjem
- DVB titulky
- nahrávanie TV signálu na PC
- Time-Shifting nahrávanie
- Schedule recording
- EPG
- teletext
- príjem FM digitálneho rádia (analog nie je podporovaný)
- USB 2.0 compatible

Je k dispozícii klasický koaxiálny konektor, možnosť pripojenia externej antény pre lepší signál.

Momentálne pracujeme s oboma zariadeniami. Prioritnejšie sa snažíme spojzdníť AzBox, vzhľadom na jeho rozšírené možnosti pripojenia a minimalizovanie potrebného hardvéru preferujeme túto cestu. Úspešne sa nám podarilo flashnúť pôvodný systém, ktorý sme následne pomocou vlastného kábla nahrali verziu (5.69). Táto verzia má voľne dostupný firmware na úprave vysielaného signálu. V prípade druhej možnosti, kedy by sme používali TV reciever A-Link by sme na spracovanie signálu použili „open source“ knižnice DVB-IS.

### 1.1.3. OpenRsi

Je to operačný softvér (programové vybavenie elektronického zariadenia riadiace toto zariadenie) vyvíjaný používateľmi, ktorí chcú zlepšiť funkcionality svojich set-top-boxov.

Podporované set-top-boxy AZBox HD, AZBox Me, GigaBlue HD QUAD a GigaBlue 800SE/UE. Jedná sa o operačný systém, ktorý je založený na OpenPLi a TV-Linux, ktoré



poskytujú jadro pre ďalší vývoj modifikovaného firmware OpenRsi 3.0. Firmware umožňuje vývoj ľubovoľných rozšírení softvéru v programovacom jazyku python. Inštalácia rozšírení je možná cez internet alebo cez USB pamäť.

Procedúra na prechod z oficiálneho firmware na modifikovaný je trochu zložitá a rozlišuje sa podľa set-top-boxu. Pri nahrávaní modifikovaného softvéru sme sa stretli s problémom, že sa AZBox zablokoval v stave „booting“ a bolo potrebné pomocou sériového pripojenia urobiť „oživenie“ set-top-boxu.

#### **1.1.4. Možnosti OpenRsi**

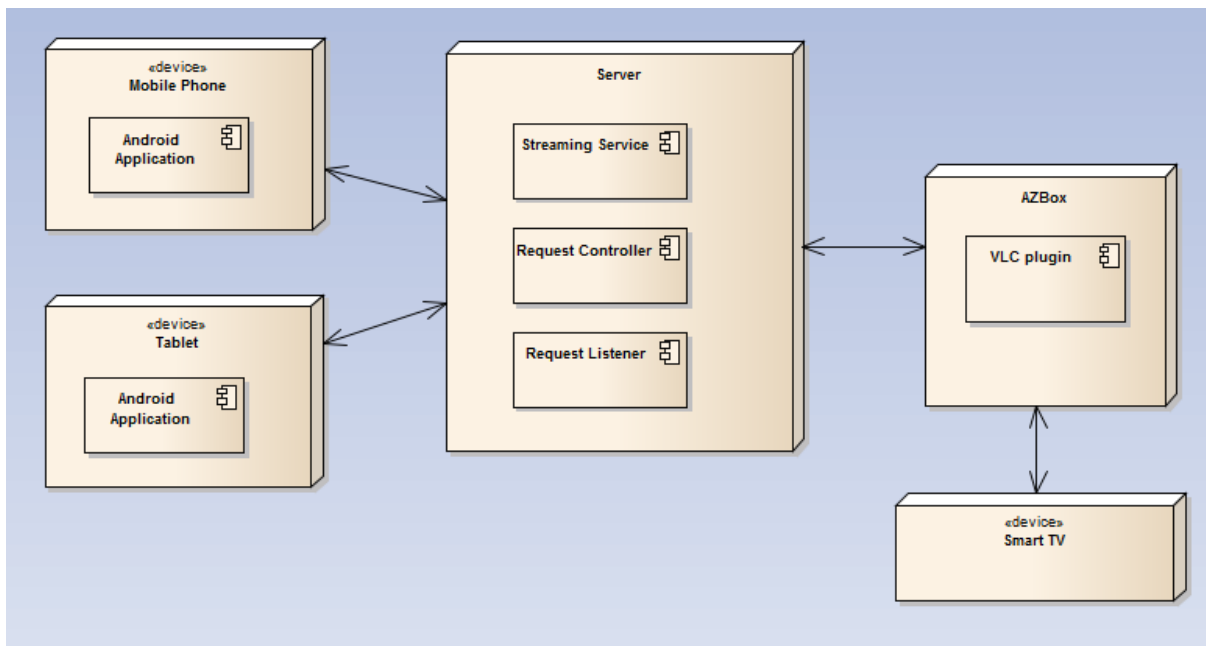
Jednou z najväčších výhod je možnosť programovať ľubovoľné rozšírenia. Softvér má možnosť inštalovať rozšírenia z internetu pomocou rss zdrojov. Pomocou rss zdrojov sme našli niekoľko dobrých rozšírení.

Existujúce rozšírenia:

- VLC player - umožňuje pripojenie na vysielací server, dokáže stiahnuť prehrávaciu listu a súborovú listu pomocou protokolu http
- VPS - Virtual private server, možnosť pripojenia na virtuálny server
- FTP client - pripojenie na ftp server
- Webový prehliadač
- IMBd - integrácia na medzinárodnú filmovú databázu, ktorá poskytuje dodatočné informácie o filme

Ďalšou výhodou „Media Center“, ktoré umožňuje kategorizované zobrazenie súborov na dátovom úložisku a tiež z využitím predinštalovaného prehrávača, umožňuje prehrávanie súborov.

Aktualizácia softvéru je jednoduchšia keď už je nainštalovaný OpenRsi. Stačí využiť automatickú aktualizáciu.



Obr.1.1: Diagram architektúry s využitím AZboxu

## 1.2 Analýza technológií, protokolov a aplikácií prepojenia TV a mobilného zariadenia

V nasledujúcej časti popíšeme technológie, protokoly a možnosti, ktoré sú použiteľné pre realizáciu nášho projektu.

### 1.2.1. Chromecast

Google Chromecast je malé zariadenie, ktoré vyzerá ako flash pamäť a je možné ho pripojiť na domáci televízor cez HDMI port. Spája sa na počítač alebo smartfón bezdrôtovo, čo umožňuje pozerať internetového obsahu na domacom televízore, alebo použitie počítača alebo mobilu ako diaľkového ovládača. Zariadenie je veľmi jednoduché, stačí ho fyzicky pripojiť, spojiť na bezdrôtovú sieť, a už funguje.

Chromecast je kompatibilný s mnohými bezdrôtovými zariadeniami, vrátane operačných systémov Android a Apple iOS. Tiež umožňuje veci ako viac obrazoviek a synchronizované prehrávanie na viacerých zariadeniach. [1]

Výhoda Chromecastu je to, že neobsahuje žiadne nové menu, používateľské rozhrania, nič, spravuje ho softvér, ktorý všetci už používame denne. Funguje na veľmi jednoduchom princípe: klikom na jedno tlačidlo na mobile alebo počítači obraz „kopírujeme“ na TV, čo sa

vlastne spraví tak, že Chromecast začne tento obsah streamovať priamo Internetu. V tomto okamihu sa aj funkcionality mobilného zariadenia mení, z neho sa stáva diaľkový ovládač, ale si zachováva plnú funkcionality, takže si môžeme Facebook, Twitter, maily alebo čo už potrebujeme pozrieť na mobile. Ako sme spomenuli, toto zariadenie podporuje aj synchronizované prehrávanie, čo znamená, že si to, čo pozeráme môžeme kedykoľvek zastaviť a spustiť ďalej na nejakom inom zariadení pripojenom na Chromecast.

Chromecast podporu je možné pridať do existujúcich aplikácií využitím špeciálneho SDK (Google Cast API) a bude podporovať aj automatický update, čo znamená, že aplikácie s pridanou podporou budú fungovať takmer ihneď potom, ako budú dostupné. [2]

Špecifikácie:

- Marvell 88DE3005 procesor
- 512MB RAM
- 2GB flash pamäte
- Video výstup – HDMI, maximálne rozlíšenie 1080p
- Konektivita: 2.4 GHz WiFi 802.11 b/g/n
- Napájanie: cez USB
- Podporované operačné systémy: Android 2.3+, iOS 6+, Windows 7+, Mac OS X 10.7+, Chrome OS [3]

Chromecast v tomto projekte nebudeme používať, keďže má problémy so smerovačmi viacerých výrobcov a zle doriešenú podporu multicastu, ktorý táto technológia používa. Kvôli tomuto nemôžeme zabezpečiť funkčnosť na všetkých sieťach a to nás môže obmedziť. Medzi nepodporovanými zariadeniami sú napríklad aj niektoré rozšírené smerovače spoločností Cisco, Netgear, D-Link a iných.

### **1.2.2. Google TV pairing protocol**

Tento protokol slúži na komunikáciu medzi klientmi a servermi na lokálnej sieti, napríklad mobilného zariadenia a Google TV. Tento protokol používa Chromecast na komunikáciu s inými zariadeniami. Funguje to tak, že klient kontaktuje server a tento potom odpovedá na túto požiadavku klientu. Server väčšinou ukáže klientu nejaký kód, klient tento kód prepošle späť serveru ako odpoveď. Napríklad, môže to byť také niečo, že server umožní klientu aby zadal alfanumerický kód na klávesnici klienta.

Aby sa nadviazalo spojenie, klient musí poslať serveru taký typ požiadavky, ktoré tento môže prijať. Napríklad, ak klient obsahuje klávesnicu, ale nie kameru, môže odpovedať iba na alfanumerické znaky ale nie na čiarové kódy.

Párovanie prebieha ako krátkotrvajúce SSL spojenie. Klient pošle niekoľko správ serveru, každá správa čaká na špeciálne potvrdenie zo servera. [7]

Funguje to takto:

- Klient pošle PairingRequest serveru
- Server odpovedá s PairingRequestAck
- Klient posielajú možnosti pre správu požiadaviek (aké môže prijať)
- Server posielajú možnosti pre správu požiadaviek (aké môže posielajú a aké odpovede môže prijať)
- Klient posielajú konfiguračné detaily v Config správe
- Server odpovedá s ConfigAck
- Server a klient vymieňajú heslo – najprv server pošle požiadavku, potom používateľ odpovedá (v praxi to funguje tak že server zobrazí nejaký kód a používateľ ho opíše)
- Keď používateľ reagoval na požiadavku, klient skontroluje odpoveď a ak je správna, pošle Secret správu
- Server skontroluje požiadavku a ak je správna pošle SecretAck

Každá správa pozostáva z dvoch častí: vonkajšej (outer), ktorá zapuzdruje všetky správy a obsahuje spoločné hlavičky a vnútornej správy (inner), ktorá obsahuje jednu z 8 preddefinovaných protokolových správ. Vonkajšia správa obsahuje dve záväzné polia (protocol\_version a status) a dve polu-záväzné polia (type, payload). Vnútorne správy môžu byť napríklad PairingRequest, Options, Configuration a iné.

Správy medzi klientom a serverom sú posielané na TCP kanáli vytvorenom z klienta na server. Keďže je dĺžka správy variabilná, správam je pridaná jednoduchá hlavička. Pre jednu správu, dáta vyslané cez sieť sú 4-bajtový unsigned integer označujúci dĺžku payload-u a serializovaná správa (typu OuterMessage). [7]

### **1.2.3. AirPlay**

AirPlay je niečo podobné ako Chromecast, iba vytvorené spoločnosťou Apple. Tiež umožňuje bezdrôtové prenášanie hudobných a videozáznamov zo zariadenia na zariadenie.

Funguje tak, že ak chceme niečo z Internetu pozerat' na našom televízore, môžeme jednoducho štartovat' film na Apple zariadení, z možností vybrať Apple TV a po niekoľkých sekundách sa film objaví na TV. Tento potom môžeme spravovat' z mobilu alebo tabletu. Najväčšia nevýhoda tohto je že je pre funkčnosť nevyhnutné mať set top box AppleTV.

Princíp funkčnosti AirPlay je podobný ako pri Chromecast, ak zanedbáme nevyhnutnosť vlastniť AppleTV. Aj táto technológia môže bežať iba ako proces v úzadí, čím umožní používateľom robiť niečo iné so svojím mobilom a tabletom počas prehrávania na televízore. Zaujímavé je, že AirPlay podporuje aj prácu s obrázkami a hudbou, tak môžeme napríklad hudbu z tabletu prehrávať na reproduktorech pripojených na wifi sieť. [4]

#### **1.2.4. DIAL API**

DIAL API slúži na spojenie webového prehliadača Google Chrome so zariadeniami na sieti. DIAL je protokol (Discovery and Launch) je protokol, ktorý má značne zjednodušiť prehrávanie audio a video obsahu medzi rôznymi zariadeniami. Štandardne, poznáme „prvé“ a „druhé“ obrazovky, kde do prvých patrí TV a do druhých mobil, počítač alebo tablet. DIAL slúži práve na prepojenie týchto medzi sebou, takže umožňuje obsah z mobilu alebo počítača prehrávať na TV. Spojenie medzi týmito zariadeniami zvyčajne funguje takto:

1. Aplikácia na mobile detektuje službu DIAL na televízore na sieti
2. Aplikácia na mobile použije DIAL aby oznámila televízoru, aby zapol svoju aplikáciu
3. Aplikácia mobile deketuje aplikáciu na TV
4. Aplikácia na mobile a aplikácia na TV nadviažu spojenie a ukážu film

DIAL protokol obsahuje dve súčasti, DIAL Service Discovery a DIAL REST Service. Service Discovery umožňuje zariadeniu klientu aby našlo DIAL servery na lokálnej sieti a získalo prístup DIAL REST Service službám na týchto zariadeniach. DIAL REST Service slúži na to, aby klient mohol štartovať, zastavovať a spravovať aplikácie na DIAL Server zariadení. DIAL Service Discovery na vyhľadávanie používa SSDP protokol zadanovaný v rámci technológie uPnP a dodatočnú hlavičku ako odpoveď na http request pre opis uPnP zariadenia. Prístup na DIAL REST Service je umožnený cez http protokol, kde http 1.0 podpora je záväzná a http 1.1 odporúčaná. [5]

## **1.2.5. UPnP protokol**

### **Technológia**

UPnP technológia definuje architektúru pre peer-to-peer sieťovú komunikáciu medzi periférnymi zariadeniami v domácnosti. Jedná sa o distribuovaný otvorený protokol založený na štandardoch ako sú TCP/IP, HTTP, XML a SOAP. Architektúra UPnP podporuje metódu zero configuration networking, ktorá automaticky vytvorí použiteľnú počítačovú sieť pre prepojené zariadenia. DHCP a DNS servery sú použiteľné len ak sú prístupné na sieti. Zariadenia sa môžu odpojiť zo siete automaticky bez zanechania informácie o ich stave.

### **Protokol**

UPnP používa HTTP pod UDP protokolom

### **Adresovanie:**

Každé nové zariadenie musí implementovať DHCP klienta a vyhľadať DHCP server. V prípade ak na sieti nie je DHCP server, zariadenie si musí priradiť IP adresu.

### **Vyhľadanie zariadenia**

UPnP vyhľadávací protokol sa nazýva SSDP. Keď je zariadenie pridané do siete, SSDP umožňuje tomuto zariadeniu inzerovať jeho služby na kontrolu bodov v sieti. V prípade ak je kontrolný bod pridaný do siete, SSDP umožní vyhľadávanie ostatných zariadení.

### **UPnP AV komponenty:**

#### **Media server**

Počítačový systém umožňujúci ukladať dáta a zdieľať medzi zariadeniami v sieti.

#### **Media server control point**

UPnP klient, podriadené zariadenie, ktoré autodetekuje UPnP servery na sieti na prehľadávanie a streamovanie dát.

#### **UPnP Media renderer DCP**

#### **UPnP Rendering control**

Nastavovanie jasů, hlasitosti, RGB a pod.

#### **UPnP remote user interface**

Posielanie a prijímanie príkazov medzi UPnP-client a UPnP-server v sieti

### **Problémy s UPnP**

Nepodporuje autentifikáciu, zariadenia musia mať implementovaný systém autentifikácie Požiadavky sú kódované, existujú 4 druhy kódovania v závislosti od typu požiadavky:

1. QRCODE – čiarový kód
2. ALPHANUMERIC – textový reťazec, obsahujúci písmená a číslice (0-9, A-Z)
3. HEXADECIMAL – textový reťazec, obsahujúci písmená a číslice (0-9, A-F)
4. NUMERICKÉ – reťazec obsahujúci číslice (0-9)

Počas inicializácie obe strany informujú aké úlohy môžu vykonávať a aké typy požiadaviek existujú. [7]

### **1.2.6. Dostupné aplikácie**

#### **Samsung SmartView**

SmartView je aplikácia pre Samsung Smart TV, ktorá umožňuje používateľom, aby svoj multimedialny obsah prehrávali na kompatibilných mobilných zariadeniach. Stačí spojiť televízor a mobilné zariadenie na rovnakú sieť, inštalovať aplikáciu na zariadenie a spustiť ju a môžeme pozerať obsah z TV na mobile a tablete. Hlavná nevýhoda tejto aplikácie je to, že je kompatibilná iba so Samsung Smart TV televízormi a Samsung mobilnými zariadeniami novšieho dátumu výroby. [6]

#### **Možnosti aplikácie**

- Diaľkové ovládanie
- Ovládanie TV a prihlásených zariadení (napríklad STB)
- Pozeranie TV
- Možnosť pozerať rovnaký obsah na TV alebo odlišný obsah z iného zdroja

#### **SmartTV Remote**

SmartTV Remote slúži na ovládanie TV pomocou mobilného zariadenia, je to jedna z najpopulárnejších neoficiálnych aplikácií pre ovládanie Samsung TV. Funguje na jednoduchom princípe a to tak, že sa TV a mobilné zariadenie pripoja na rovnakú sieť a potom ich už možno ovládať. Jedna z možností je zmena programu nielen pomocou čísla, ale aj pomocou ikony kanálu, takže tieto už nemusia byť sortované na TV, ale môžu byť aj priamo na diaľkovom ovládači. Aplikácia bola otestovaná na viacerých Samsung televíznych zariadeniach, ale je možné, že niektoré ešte nie sú kompatibilné. [8]

### **1.2.7. Streaming audiovizuálneho obsahu**

Streaming audiovizuálneho obsahu znamená priebežný prenos bez nutnosti ukladania celého prenášaného súboru na strane príjemcu. Umožňuje prehrávanie audiovizuálneho obsahu krátko po nadviazaní spojenia so serverom.

#### **1.2.7.1. Prenos audiovizuálneho obsahu po sieti**

S rozvojom vysokorýchlostných dátových sietí sa stále viac dostáva do popredia popri klasických VoIP (Voice over IP) služieb aj prenos videa cez IP siete. Rovnako, ako pri hlasových službách, je aj pri videu potreba stabilnej linky s garantovanou prenosovou rýchlosťou a odozvou. Väčšina VoIP a video prenosov je v súčasnosti realizovaná na základe Ethernetu.

Pre prenos videa medzi rôznymi systémami je potrebné prispôbiť jeho kvalitu pre dané médiá. Väčšina video formátov v súčasnosti na prenos po sieti využíva protokoly postavené na TCP/IP, najmä na UDP paketoch, ktorých prijatie nie je potrebné potvrdzovať, čo nám dovoľuje prehrávanie videa v reálnom čase. Na prenos videa cez sieť je možné používať aj TCP pakety, ktorých prijatie sa potvrdzuje, čo je ale pre potreby prenosu videa nepotrebné, dokonca v niektorých prípadoch aj nežiaduce.

V prípade, že je použitá kvalitná linka, nedochádza pri prenose k výpadkom paketov a obraz u príjemcu je teda kvalitný. So zhoršujúcou sa kvalitou linky, zníženou rýchlosťou a odozvou sa stáva obraz čoraz menej kvalitný. V prípade použitia TCP paketov sa síce chybné prijaté pakety prepošlú znovu, no zväčšuje to odozvu, ktorá nám znemožňuje prehrávanie v reálnom čase. V prípade použitia UDP paketov nie je síce zaručený bezchybný prenos, no je zaručená rýchla odozva, čo je pri online komunikácii kľúčové.

Súčasný Ethernetové siete majú implementované mechanizmy (QoS), ktoré pri sieťovej prevádzke uprednostňujú video a audio prenos pred bežnou prevádzkou. V Ethernete je najpoužívanejší štandard IEEE 802.1p, v ktorom je definovaných 8 stupňov priority paketov. Najdôležitejšia trieda má najvyššiu prioritu, teda hodnotu 7 – slúži na riadenie systému a siete. V bezdrôtových wifi sieťach je ekvivalentom štandard 802.11e.

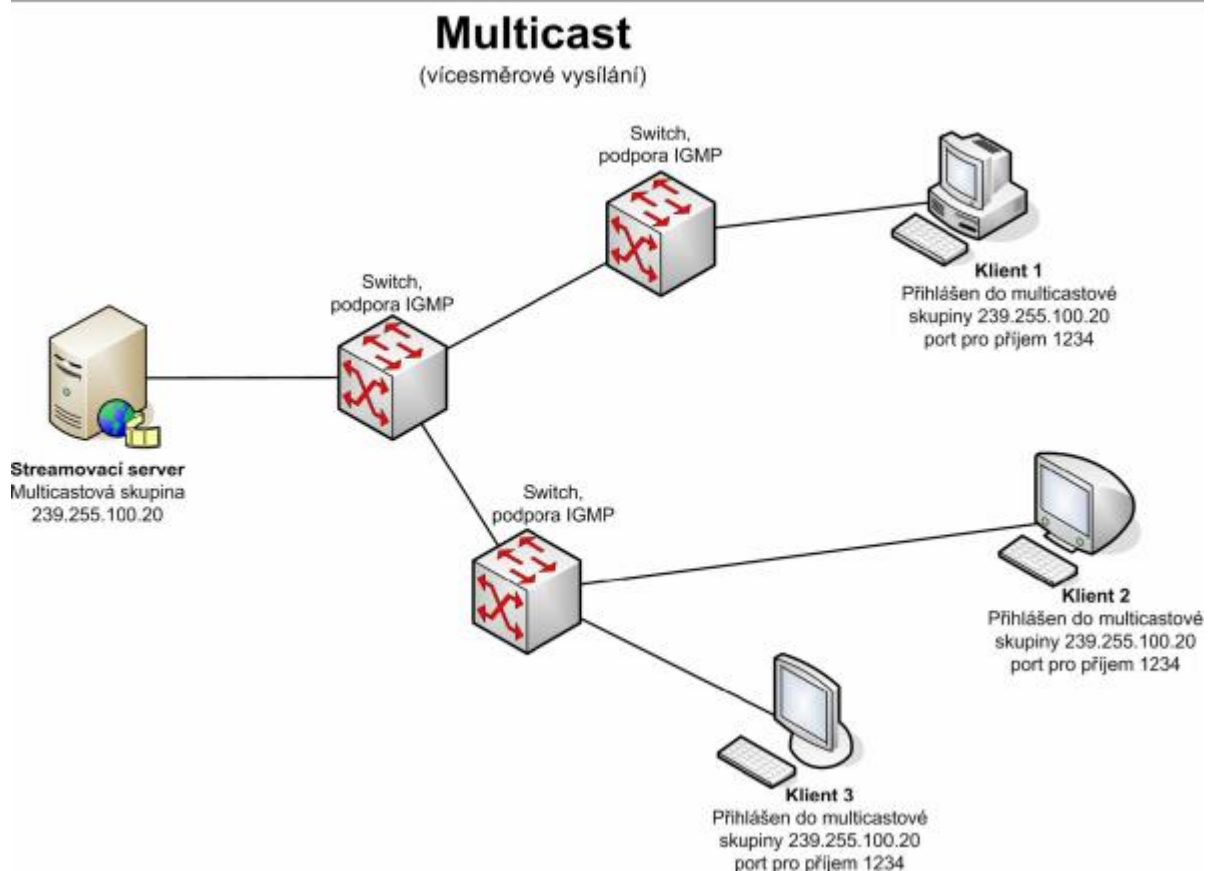
Na prenos videa a hlasových služieb na rôznorodých systémoch je potrebné prenášané dáta optimalizovať. Na prenos streamu, ktorý potrebuje napríklad prenosovú rýchlosť 5Mbps nie je postačujúca linka o maximálnej prenosovej rýchlosti 4 Mbps. Aby bolo možné využiť túto linku na prenos streamu, bude potrebná kompresia, ktorá môže byť stratová, alebo bezstratová. V prípade bezstratovej kompresie sa zmenší prenosová informácia bez straty dát.



V prípade stratovej kompresie sa z prenášaných dát úmyselne vypúšťajú fragmenty, ktoré nie sú pri rekonštrukcii dôležité a na zobrazenie pôvodných dát nemajú takmer žiaden vplyv (nerátajúc kvalitu zobrazovaného obsahu). Práve tento typ kompresie je najčastejšie používaný pri prenosoch audiovizuálneho obsahu.

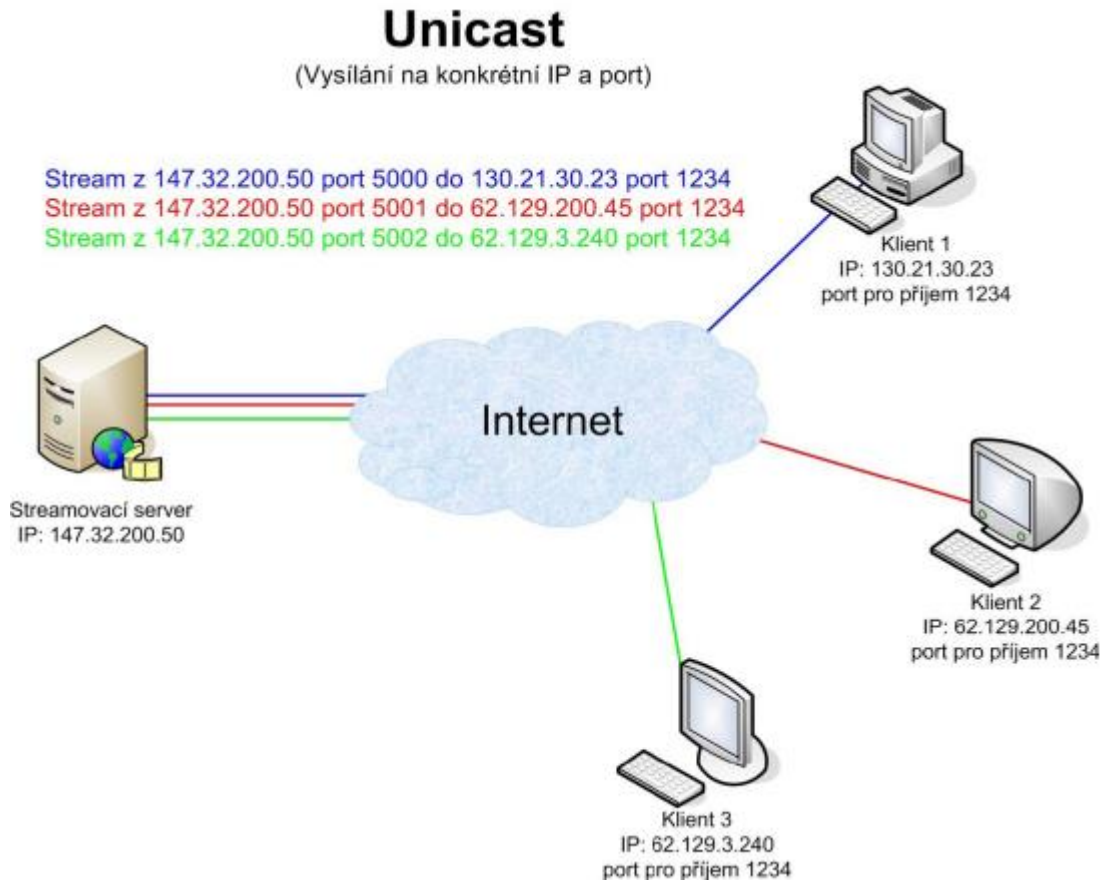
Audiovizuálny obsah je možné prenášať po sieti viacerými spôsobmi. Popísané spôsoby sú založené na protokole UDP. Je potrebné ale rozlišovať vysielanie na lokálnej sieti a vysielanie v internete. Na lokálnej sieti je samozrejme možné použiť vyššiu kvalitu, keďže je k dispozícii väčšia prenosová kapacita.

V prípade lokálnej siete je možné využiť multicast (viac príjemcov) vysielanie. V tomto prípade server vysiela streamovaný obsah na jednu IP adresu z rozsahu 224.0.0.0 – 239.255.255.255. Jednotliví príjemcovia si potom pomocou protokolu IGMP (Internet Group Management Protocol) zarezervujú prijímanie streamovaného obsahu. Výhodou tohto spôsobu je fakt, že server vysiela iba jeden stream, ktorý sa na multicastovom smerovači duplikuje pre všetkých členov multicastovej skupiny. Naopak nevýhodou je, že tento spôsob nie je možný použiť v internete. Princíp multicast vysielania je znázornený na obrázku obr.1.2.



Obr.1.2: Princíp multicast vysielania (zdroj: <http://www.comtel.cz/files/download.php?id=3281>)

Riešením pre vysielanie v internete je unicast vysielanie, teda vysielanie jednému uzlu (príjemcovi). Problémom tu je fakt, že je potrebné vyslať toľko streamov, koľko príjemcov existuje, a teda v súčte je potrebné mať k dispozícii linku s pomerne veľkou odosielacou kapacitou. Princíp unicast vysielania je znázornený na obrázku obr.1.3.



Obr.1.3: Princíp unicast vysielania. (zdroj: <http://www.comtel.cz/files/download.php?id=3281>)

### 1.2.7.2. Audiovizuálne súbory a formáty

Audiovizuálne formáty bývajú obvykle vhodné pre ukladanie na rôznych médiách, ale sú taktiež vhodné aj na streamovanie audiovizuálneho obsahu, pre ktoré sa často využívajú tzv. kontajnerové formáty.

Kontajnerové formáty určujú rôzne typy multimedialného obsahu. Môžu obsahovať rôzne kombinácie obsahu, napríklad iba video, alebo audio s videom a iným obsahom (napríklad titulky, EPG, a pod.).

Pri streamovaní sa ale často nepoužívajú klasické kontajnerové formáty. Pred streamovaním sa teda obsah kontajnerov rozbalí na samotné audio a video, ktoré je potom pomocou streamovacieho protokolu (v prípade RTP sa napríklad používa RTP payload format

– obsahuje napríklad G.721 audio, H.263 video, alebo MPEG-TS audio + video) streamované po sieti.

### 1.2.7.3. Protokoly

Audiovizuálny obsah býva často pri streamovaní po sieti prenášaný za pomoci protokolu RTP (Real-time Transport Protocol – definovaný v RFC 3550) s použitím protokolov RTSP (Real Time Streaming Protocol – definovaný v RFC 2326), RTCP (Real Time Control Protocol – definovaný v RFC 1889), H.323 (napríklad H.225 a H.245). Na streamovanie audiovizuálneho obsahu býva použitý aj HTTP protokol (definovaný v RFC 2616).

RTP je protokol zaisťujúci multimediálne prenosy v reálnom čase. Tento protokol nezaručuje doručenie datagramov, ale dáva im časové pečiatky, pomocou ktorých sa môžu na strane príjemcu správne usporiadať prijaté pakety. RTP protokol najčastejšie používa UDP (protokolová architektúra je znázornená v tabuľke 1.) protokol a typicky využíva čísla portov 5004, 5005 a 6970, no môže využiť aj iné. RTP používajú protokoly SIP a H.323.

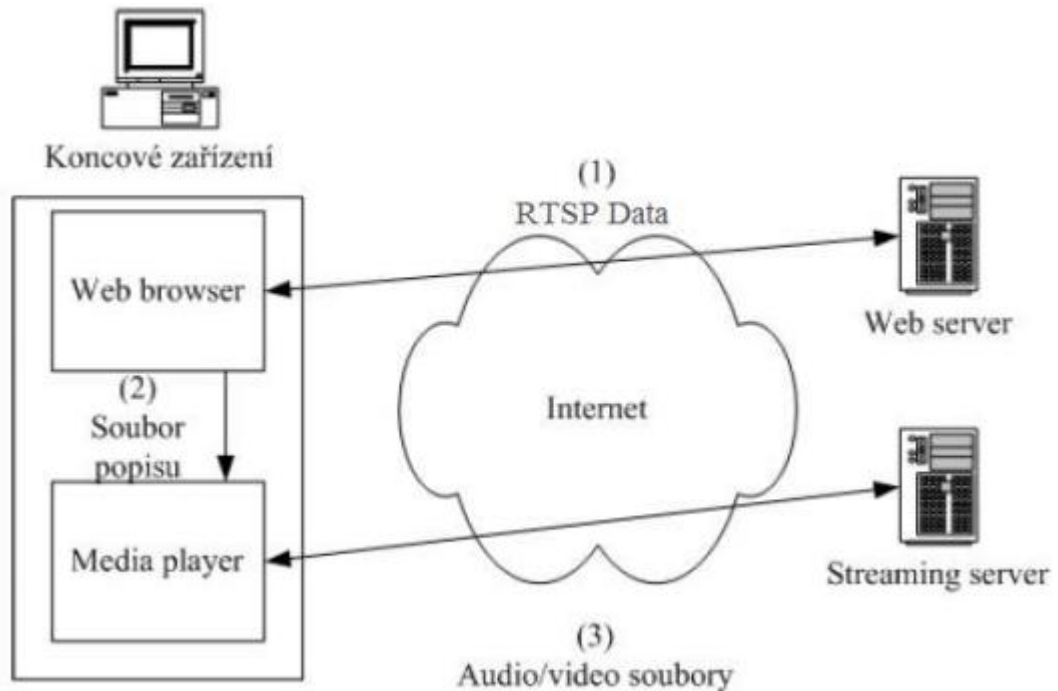
Tabuľka 1.1: Protokolová architektúra

PCM, H.261, MPEG, MPEG-1, MPEG-2...
RTP
UDP
IP
Sieťová infraštruktúra

RTP k multimediálnemu obsahu pripojí hlavičku, v ktorej definuje číslo paketu a označenie typu obsahu – formát multimediálneho súboru, ktorý prenáša. K typom dát RTP správ patria G.721 audio, GSM audio, G.722 audio, MPEG audio, G.728 audio, H.261, MPEG-1 a MPEG-2 video.

Protokol RTCP rieši monitorovanie doručovania paketov, keďže samotný protokol RTP túto možnosť nepodporuje. RTCP teda spolupracuje s protokolom RTP pri prenose obsahu v reálnom čase. RTCP pomáha monitorovať doručovanie dát v rozsiahlych sieťach so skupinovými vysielaniami, čo pomáha príjemcovi detegovať stratu paketov a kompenzovať ju oneskorením v sieti. RTCP využíva port o 1 vyšší, než RTP.

RTSP je signalizačný protokol pre riadenie prenosu dát (napríklad protokolom RTP), využíva port 554, funguje na báze klient – server a podporuje multicast vysielanie. Popis komunikácie s využitím RTSP je znázornený na obrázku obr.1.4.



Obr.1.4: Popis komunikácie s využitím RTSP (zdroj: <http://manik.korh.cz/files/škola/bmds/otázky%20bmds%202010-11.pdf>)

MPEG Transport stream (MPEG-TS) sa používa na prenos a uchovávanie audiovizuálneho obsahu. Typicky sa používa v broadcast systémoch ako DVB (Digital Video Broadcasting), alebo IPTV (Internet Protocol Television). Kontajner MPEG-TS môže súčasne prenášať aj viacero na sebe nezávislých videí. Je prispôsobený na streamovanie po sieťach, na ktorých sú časté poruchy, alebo rušenie signálu.

#### 1.2.7.4. Programy pre streaming audiovizuálneho obsahu

Na prijímanie formátov pre audio a video streamovanie a vytváranie streamov na viacerých platformách sa používa program VLC media player od videolan-u.

Program VLC je voľne šíriteľný, multiplatformový multimedialny prehrávač s otvoreným zdrojovým kódom. Umožňuje prácu s rôznymi protokolmi používanými na

streamovanie. Poskytuje možnosti pre RTP/UDP streamovanie s využitím RTSP, čo je vhodné pre realizáciu nášho zadania.

Je dostupný na internetovej stránke <http://www.videolan.org/vlc/>.

## 2.Prototyp

---

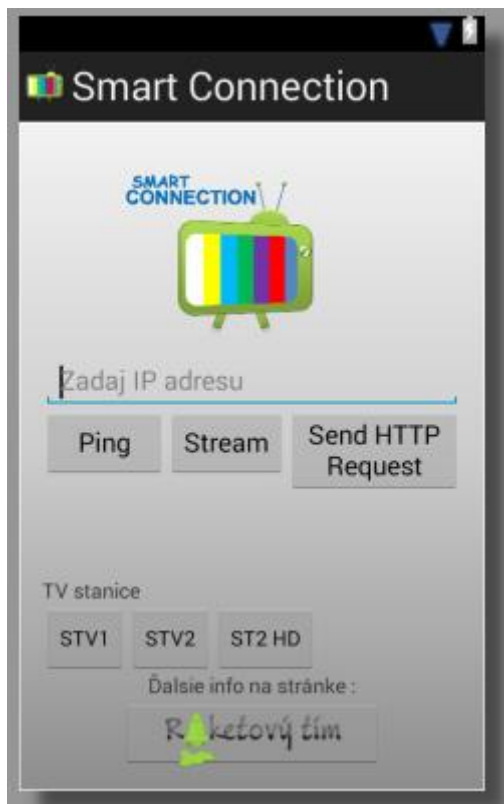
V nasledujúcej časti popíšeme výsledok našej práce na prototypu a zhodnotíme jednotlivé časti.

### 2.1. Mobilná aplikácia (platforma Android)

Táto časť dokumentu obsahuje popis vývoja mobilnej aplikácie a ďalej tu predstavíme aktuálny stav mobilnej aplikácie.

#### 2.1.1. Smart Connection

Cieľom prototypu mobilnej aplikácie Smart Connection bolo vytvorenie funkčných častí aplikácie, ktoré budú tvoriť základ nášho výsledného produktu. Úvodná obrazovka aplikácie je zobrazená na obrázku obr.2.1.

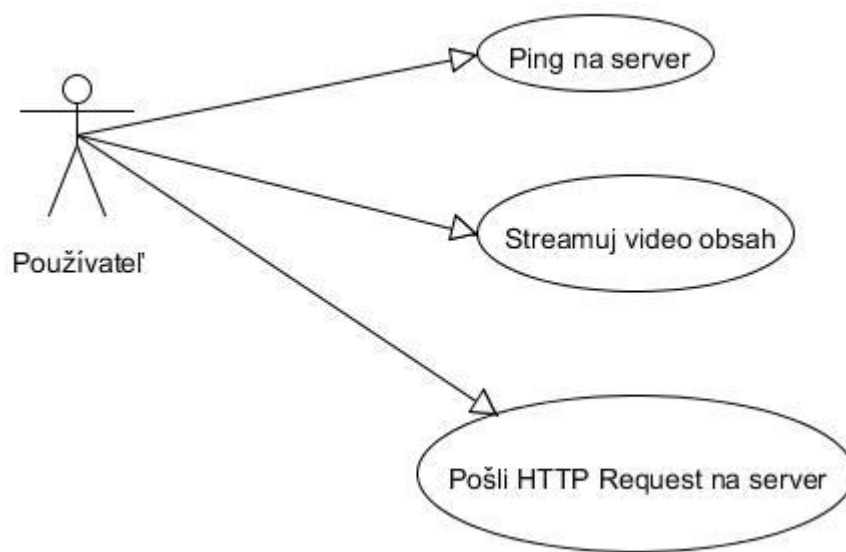


Obr 2.1: Prototyp aplikácie Smart Connection.

Základné funkcionality prototypu aplikácie Smart Connection :

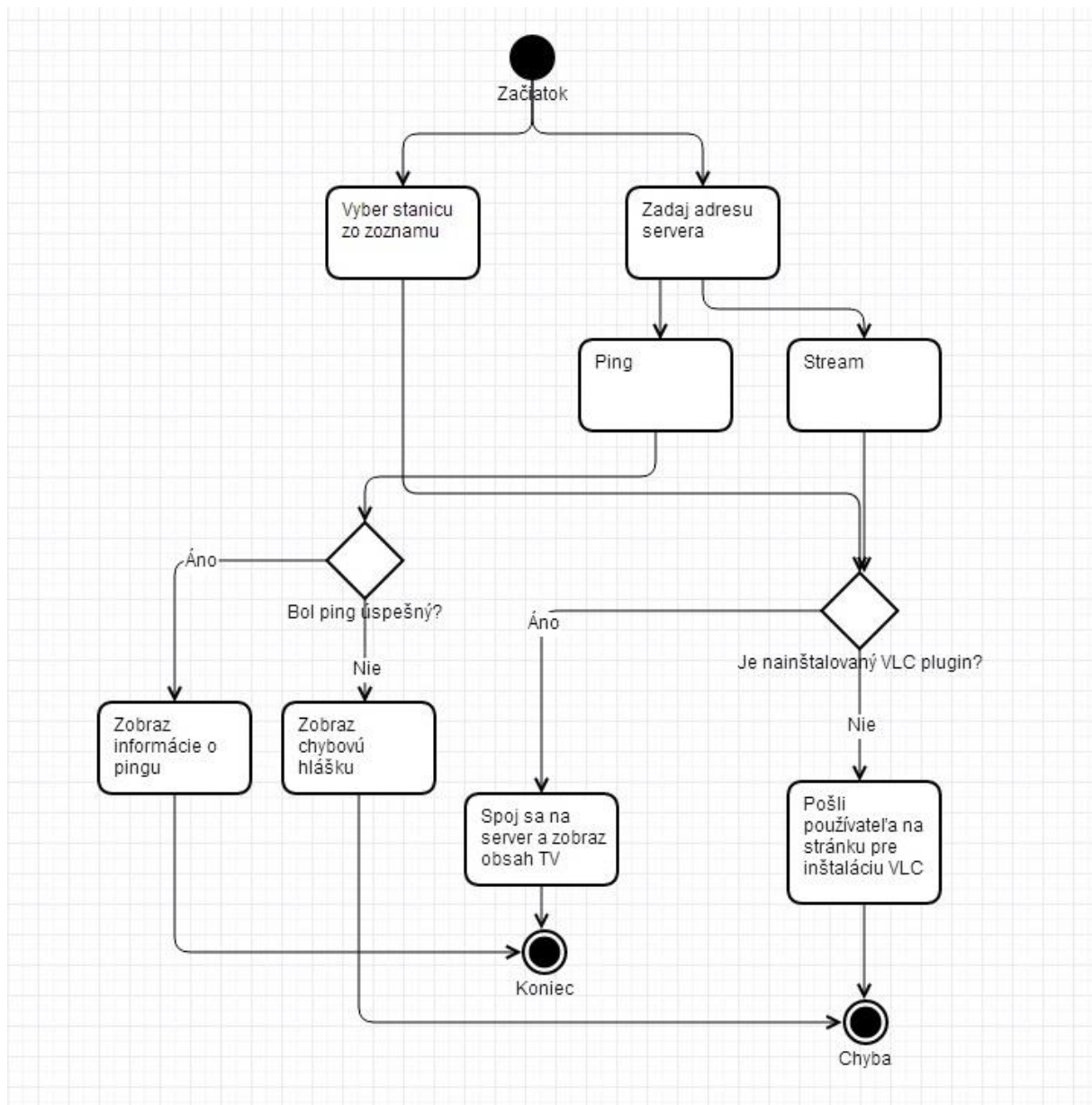
- Ping
- Stream
- Send http request
- Playlist
- Odkaz na webovú stránku tímového projektu

Na obrázku obr.2.2 sa nachádza diagram prípadov použitia aplikácie.



Obr. 2.2: Diagram prípadov použitia.

Na obrázku obr.2.3 je znázornený diagram aktivít pre navrhnutú mobilnú aplikáciu.

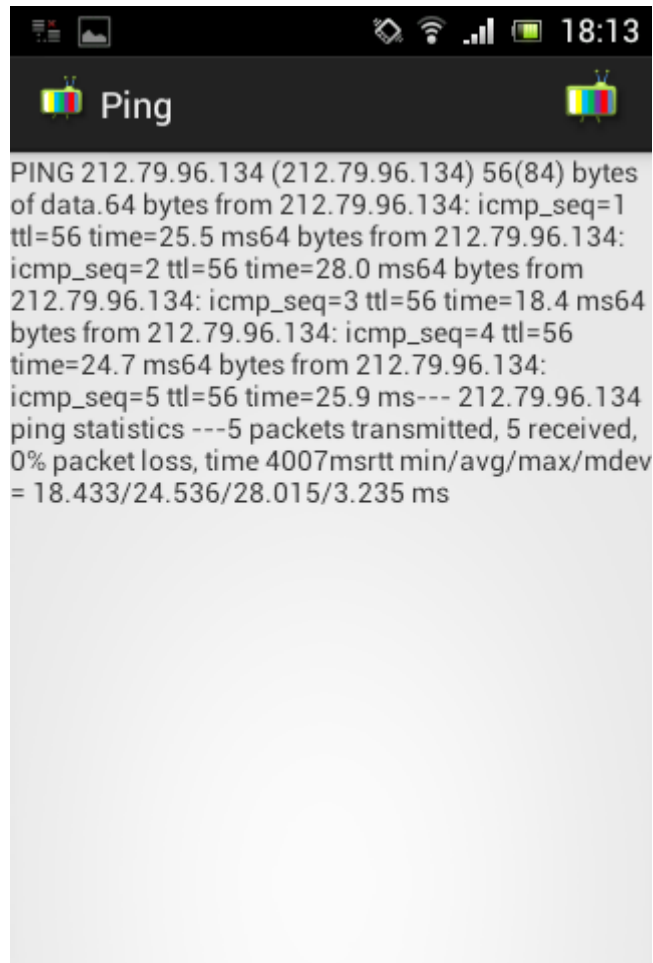


Obr. 2.3: Diagram aktivít pre mobilnú aplikáciu.

### 2.1.2. Ping

Táto funkcionality vykoná ping na zvolenú IP adresu alebo webovú stránku. Funguje tak, že používateľ do textového poľa zadá adresu alebo stránku a potom sa vykoná ping. Výsledky sa vypíšu v novom okne aplikácie. Výpis pingu je rovnaký ako keď spúšťame ping na OS Linux alebo Windows, a ak je neúspešný (zadáme nesprávnu alebo neexistujúcu IP adresu), tak sa vypíše iba „ping started“. Na obrázku obr.2.4 je zobrazená obrazovka aplikácie po spustení funkcie Ping na IP adresu 212.79.96.134.





Obr. 2.4: Výpis Ping.

### 2.1.3. Stream

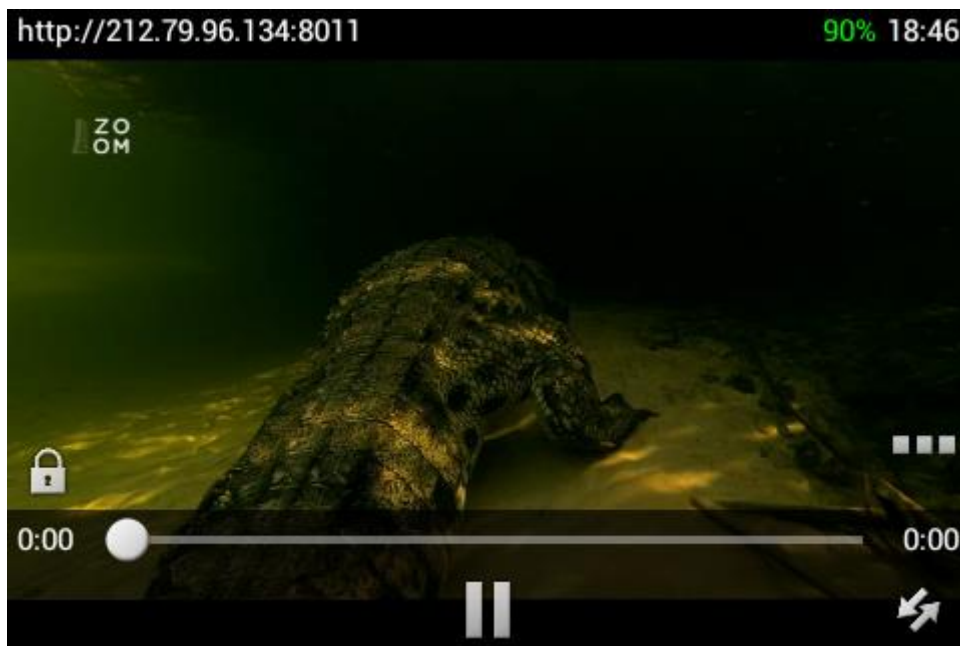
Funkcionalita Stream vykonáva streamovanie video obsahu zo servera na mobilnom zariadení. Popis fungovania funkcionality Stream:

1. Používateľ zadá adresu streamovacieho servera (IP adresa + port).
2. Používateľ klikne na tlačidlo Stream.
3. Stream sa začne prehrávať zo zadaného servera, na toto sa používa VLC plugin.
4. Žiadaný obsah sa zobrazí na mobilnom zariadení.

V prípade, že nie je na zariadení nainštalovaný VLC prehrávač, aplikácia presmeruje používateľa na Play Store, odkiaľ je možné inštalovať VLC.

Na začiatku vývoja aplikácie sme skúšali namiesto VLC plugin použiť štandardný Android plugin na prehrávanie video obsahu. Toto fungovalo správne na školskom tablete, na ktorom sme pôvodne testovali, problém nastal však pri testovaní aplikácie na inom mobilnom zariadení, kde identický obsah nebolo možné prehrávať. Zistili sme, že dôvod spôsobujúci

tento problém je veľmi slabá podpora kodekov v prípade niektorých Android zariadení. Preto sme sa rozhodli pre VLC plugin, ktorý má takmer univerzálnu podporu kodekov, čo zaručí správnu funkčnosť prehrávania streamovaného obsahu na väčšine novších Android zariadení. Funkčnosť streamovania bola otestovaná na viacerých dostupných streamoch, používajúcich viaceré protokoly (http, rtsp, rtp). Na obrázku obr.2.5 je zachytené prehrávanie http streamu.



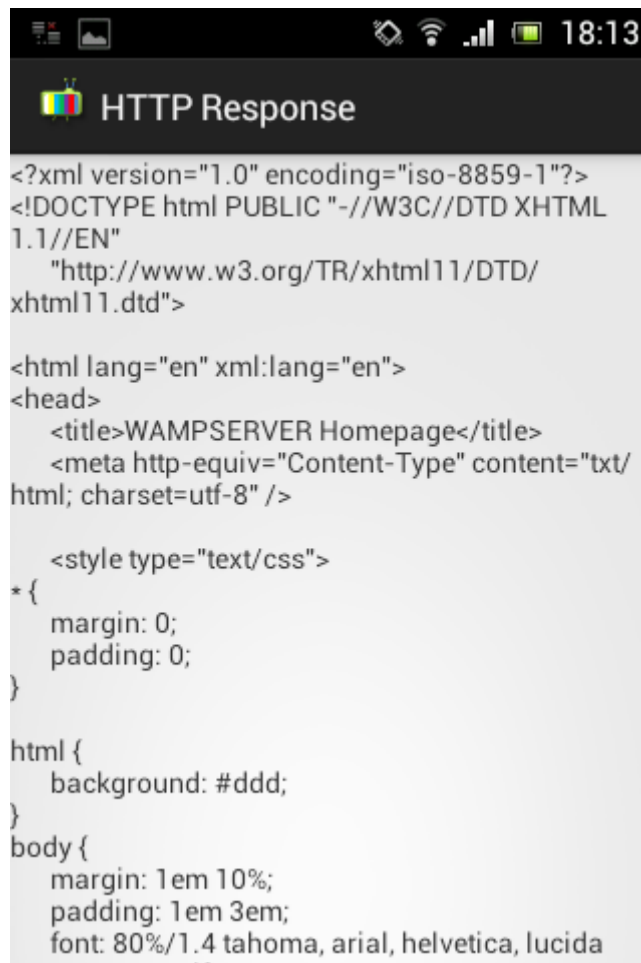
Obr. 2.5: Prehrávanie http streamu.

#### 2.1.4. Send http request

Táto funkcionálna je experimentálna a bude rozšírená neskôr. Funguje tak, že na zadanú adresu pošle http požiadavku (request) a vypíše obsah XML súboru, ktorý server vráti ako odpoveď (response). V prípade, že server nie je dostupný, alebo má zatvorené porty, vypíše sa chybové hlásenie. Na obrázku obr.2.6 je zobrazená obrazovka aplikácie po zobrazení odpovede zo servera na http požiadavku.

Čo sa týka zadávania adries, všetky adresy sa zadávajú do rovnakého textového poľa s tým rozdielom, že v prípade streamovania treba zadať celú adresu (aj s protokolom) a vo zvyšných prípadoch netreba zadávať žiadne protokoly.

Táto funkcionálna bude v priebehu ďalšej práce rozšírená spôsobom, ktorý umožní zariadeniu pripojenému na streamovací server komunikáciu so serverom. Toto bude využité napríklad pre diaľkové ovládanie streamovacieho servera priamo z nášho zariadenia.



```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/
xhtml11.dtd">

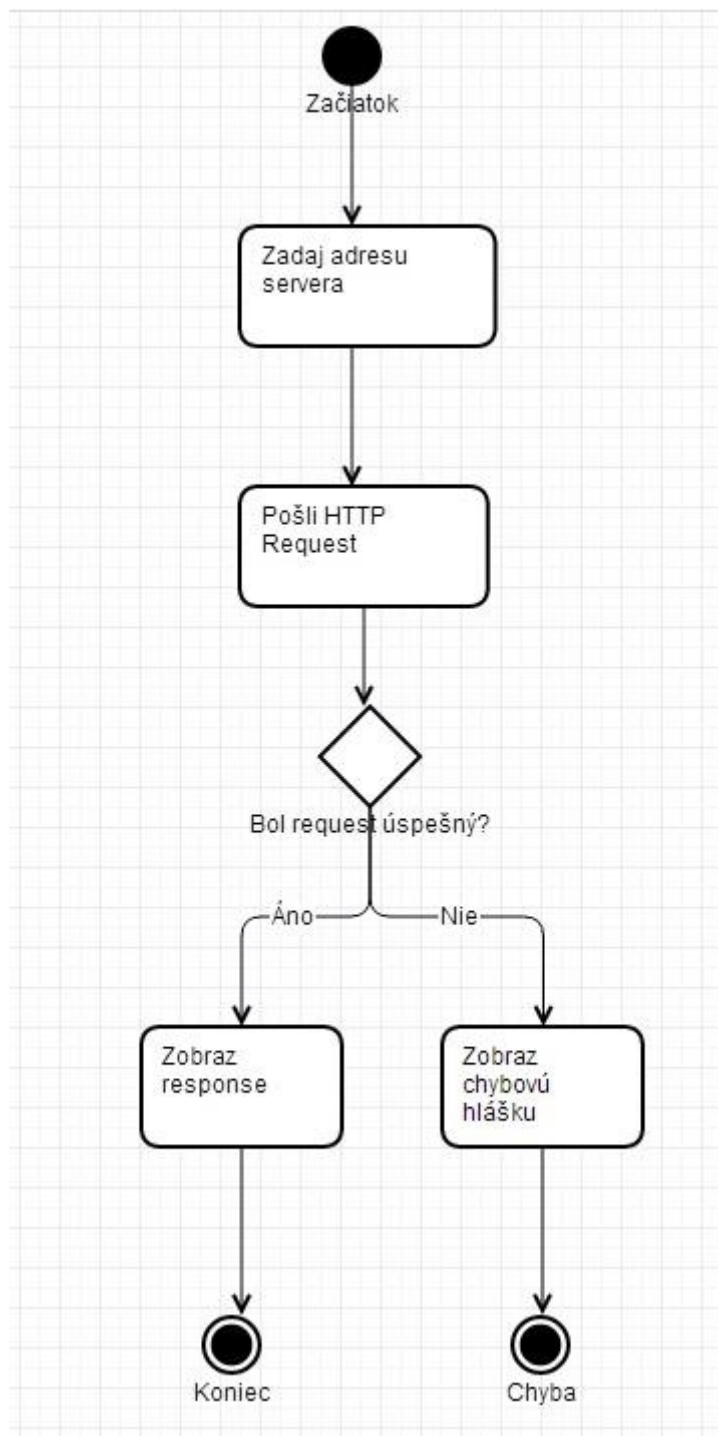
<html lang="en" xml:lang="en">
<head>
  <title>WAMPSEVER Homepage</title>
  <meta http-equiv="Content-Type" content="txt/
html; charset=utf-8" />

  <style type="text/css">
*{
  margin: 0;
padding: 0;
}

html {
  background: #ddd;
}
body {
  margin: 1em 10%;
padding: 1em 3em;
font: 80%/1.4 tahoma, arial, helvetica, lucida
```

Obr. 2.6: Odpoveď zo servera na http požiadavku.

Na obrázku obr.2.7 je znázornený diagram aktivít pre funkcionality Send http request.



Obr. 2.7: Diagram aktivít pre funkcionlitu Send http request.

### 2.1.5. Playlist

Funkcionalita Playlist (TV stanice) funguje na podobnom princípe ako funkcia Stream a umožňuje používateľovi rýchly prístup k vybraným televíznym staniciam(STV1, STV2,

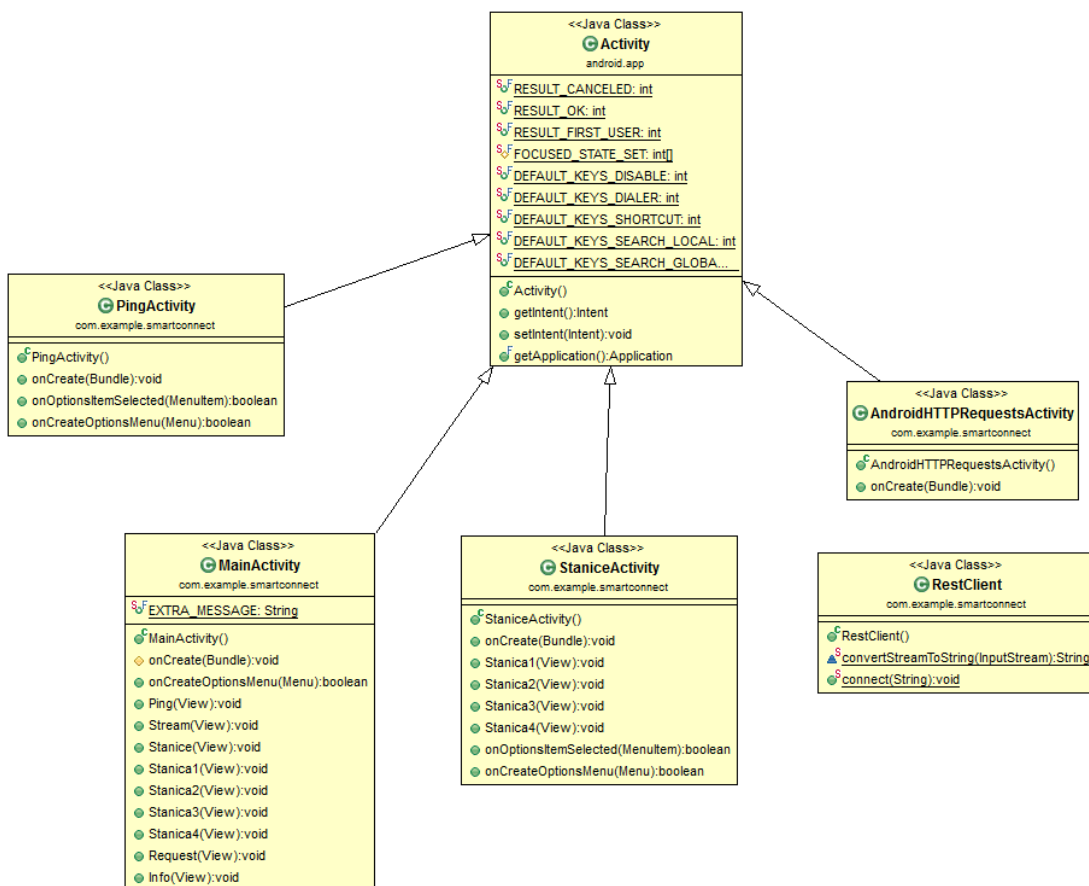
STV2 HD), bez potreby aby zadával adresu streamovacieho servera. Na obrázku obr.2.8 je zobrazená časť aplikácie, ktorá predstavuje funkcionality Playlist.



Obr. 2.8. Playlist.

### 2.1.6. Diagram tried

Na obrázku obr. 2.9 je znázornený diagram tried vytvorenej mobilnej aplikácie.



Obr. 2.9. Diagram tried vytvorenej mobilnej aplikácie.

### 2.1.7. Zhodnotenie mobilnej aplikácie

Podarila sa nám vytvoriť aplikáciu, ktorá bude tvoriť základ pre výslednú aplikáciu tímového projektu. Okrem spomenutých funkcionalít sme vytvorili aj formulár, ktorý slúži na

registráciu a prihlásenie používateľov (s využitím PHP a MySQL), ktorý neskôr pridáme k našej aplikácii. Aplikácia bola testovaná a optimalizovaná (vzhľadom na rozloženie prvkov) na zariadeniach Sony Xperia Mini(480 x 320) a ZTE Blade G (854 x 480).

## **2.2. Streamovací server**

Táto časť dokumentu obsahuje popis spustenia a konfigurácie streamovacieho servera a ďalej tu predstavíme aktuálny stav konfigurácie servera s jeho možnosťami.

Ako server mal slúžiť set-top-box AZbox HD Premium, no po problémoch s firmvérom sme prešli na variantu serveru spusteného na počítači pod operačným systémom Linux s príjmom signálu cez USB TV kartu.

### **2.2.1. Server**

Ako server pre naše účely mal fungovať virtuálny stroj na počítači dostupnom v školskom laboratóriu. Problémy ale nastali po pripojení USB TV prijímača. Zariadenie síce vyzeralo ako zdetegované a funkčné, no po spustení VLC prehrávača a pokusoch spustiť digitálne vysielanie sme narážali na chybu – žiadne zariadenie. Po vyskúšaní iného USB TV prijímača a následnom zapájaní USB flash diskov sme zistili, že dostupné USB rozhrania na tomto počítači nie sú funkčné.

Po týchto problémoch a konzultácií s vedúcim tímového projektu sme pristúpili k riešeniu, keď serverová časť nášho prototypu je spustená na osobnom počítači člena tímu, na ktorom je nainštalovaný operačný systém Linux Ubuntu Server 12.04.3.

### **2.2.2. Príjem digitálneho signálu**

Digitálny signál prijíma server z terestriálneho vysielania DVB-T na USB TV prijímač.

K dispozícií sme mali A-Link DTU, no po problémoch so zariadením (prvé zariadenie nefunkčné) sme sa rozhodli využívať zariadenie zakúpené tímom Pinnacle PCTV 73e. Toto zariadenie je schopné prijímať signál z jedného streamu (multiplexu) DVB-T v rovnakom čase. Ak by sme teda chceli prijímať signál z dvoch streamov súčasne, potrebovali by sme dve zariadenia.

Na zistenie aktuálne dostupných televíznych kanálov v prijímanom streame slúži príkaz:

```
scan /usr/share/dvb/dvb-t/"oblast"
```

ktorý nám poskytne informácie o názve kanála, frekvencií, šírke pásma, modulácií a identifikátore služby (SID) [17].

### 2.2.3. Streamovanie

Na streamovanie prijímaného obsahu do lokálnej siete sme sa rozhodli využívať program Dvblast od VideoLan-u. Pomocou tohto programu je možné streamovať na konkrétnu IP adresu a port konkrétny televízny kanál.

Pred spustením programu Dvblast je potrebné urobiť konfiguračný súbor, ktorého obsah je nasledovný:

„IPadresa:port“      „always-on flag“      „SID“

Konfiguračný súbor na našom serveri je uvedený nižšie:

```
;Channel stv 1
239.255.1.1:50021      3001
;Channel stv 2 hd
239.255.1.1:50042      3003
;Channel stv 2
239.255.1.1:50033      3002
;NOE tv
;239.255.1.1:5000      0      2
;Slagr TV
;239.255.1.1:5001      0      3
;TV 8
;239.255.1.1:5005      0      4
;ZSTV
;239.255.1.1:5006      0      5
;TV JOJ
;239.255.1.1:5007      0      2001
;TV JOJ PLUS
;239.255.1.1:5008      0      2002
;TV MARKIZA
```

```
;239.255.1.1:5009      0      2003
;DOMA
;239.255.1.1:5010      0      2004
;TA3
;239.255.1.1:5011      0      2005
```

Po napísaní konfiguračného súboru je možné spustiť program Dvblast nasledujúcim príkazom (my spúšťame s frekvenciou 522166 kHz pretože má najlepší príjem signálu):

```
dvblast -a „číslo tuneru“ -c „cesta ku konfiguračnému súboru“ -f „frekvencia streamu“  
-m „modulácia“ -b „šírka pásma“.
```

Momentálne sme schopní streamovať obsah prijímaný z DVB-T do lokálnej siete s využitím multicastu. Je potrebné teda tento obsah streamovať aj unicast do internetu, aby bolo možné sledovať streamované kanály aj mimo domu na mobilných zariadeniach.

Unicast streamovanie zabezpečujeme s využitím programu Udpxy [18], ktorý re-streamuje rtp stream vysielaný z dvblast ako http stream na zadaný port. Program Udpxy použijeme nasledovným príkazom:

```
udpxy -p „port“.
```

Po spustení programu udpxy je možné príkazom

```
http://“IP adresa servera“:“port“/rtp/“IP adresa multicast“:“port“
```

spustiť prehrávanie stream-u na konkrétnej klientskej stanici.

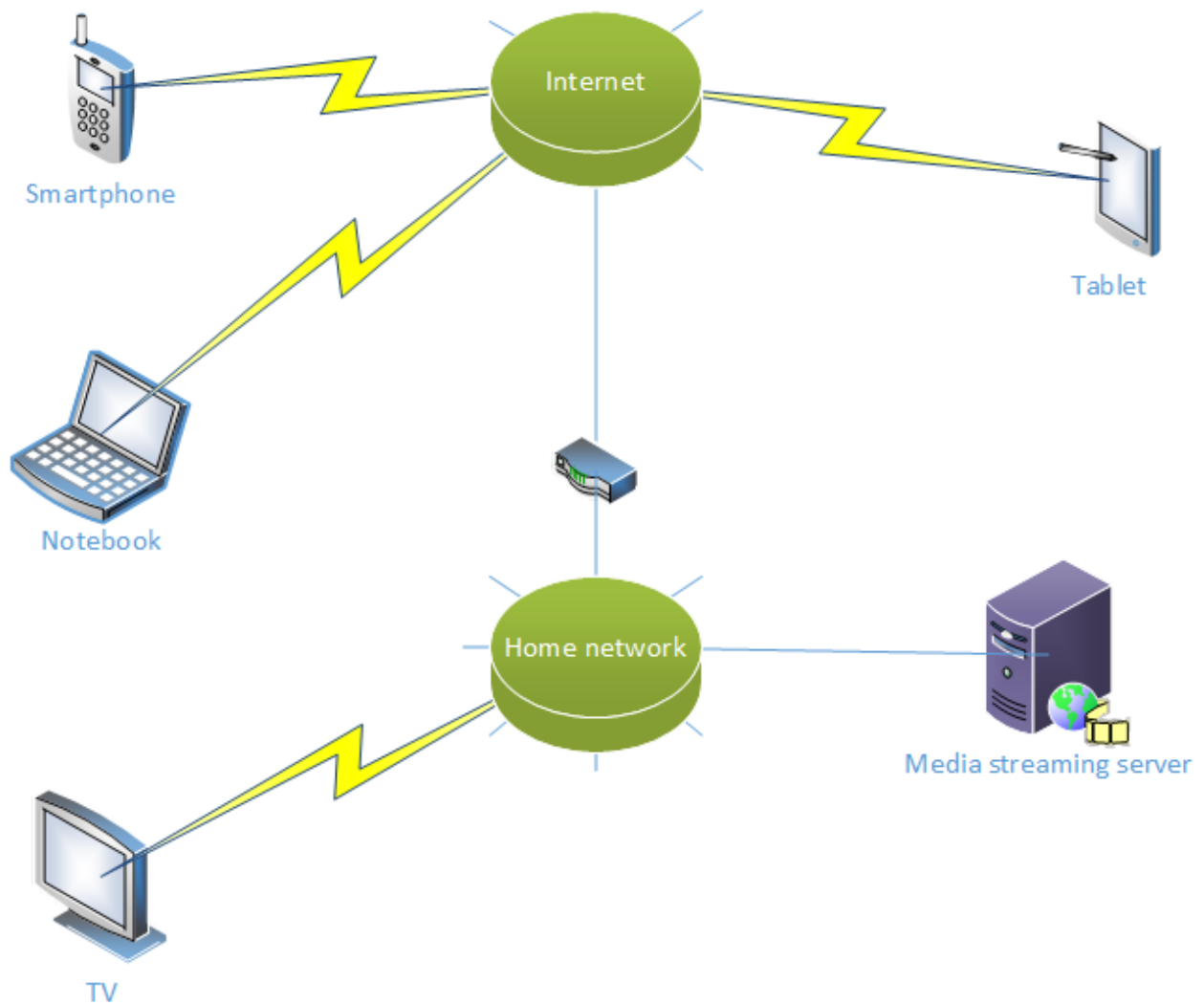
#### **2.2.4. Ďalší vývoj streamovacieho servera**

V ďalšom semestri chceme implementovať nové možnosti na serveri a rozšíriť server o aplikáciu, ktorá bude riadiť stream s možnosťou zmeny prehrávanej frekvencii. V krátkosti povedané riadenie programu dvblast cez program, ktorý budeme riadiť pomocou prijatých požiadaviek (npr.: zmena frekvencie, zobrazenie dodatočnej informácie o sledovanom programe..). Server bude poskytovať prehrávaciu listu. Vytvorenie prehrávacej liste sa skladá z troch krokov:

1. Program „scan“, získanie aktuálne dostupných frekvencií
2. Aplikácia spracuje výstup programu „scan“

Aplikácia vytvorí prehrávaciu listu





Obr. 2.10 Plánovaná architektúra riešenia

Na obrázku 2.10 je znázornená plánovaná architektúra celého riešenia, kde streamovací server je pripojený do domácej siete ku ktorej je pripojený i televízor. Domácu sieť riadi smerovač, ktorý je pripojený na internet a má zapnutý NAT na potrebných portoch. Server poskytuje rtp a http stream. Rtp stream poskytuje pre domácu sieť a http stream pre pripojenie cez internet.

### 2.2.5. Zhodnotenie streamovacieho servera

Streamovací server je v súčasnosti schopný re-streamovania obsahu prijatého z DVB-T na lokálnu sieť pomocou multicast vysielania, čo dokáže ušetriť kapacitu sieťovej linky od servera. Ďalej je server schopný streamovať obsah aj do internetu využitím unicast, nakoľko multicast do internetu nie je možný. Na strane servera je možné obmedziť maximálny počet

zariadení prijímajúcich streamovaný obsah a taktiež je možné sledovať aktuálne pripojené zariadenia.

# 3.Špecifikácia požiadaviek

---

Výstupom projektu bude fungujúci systém umožňujúci sledovanie živého televízneho vysielania na mobilných zariadeniach s aktívnym pripojením k internetu s doplnkovými funkciami.

Systém sa bude skladať z troch základných funkčných blokov:

- Serverová aplikácia,
- Aplikácia pre mobilné zariadenia,
- Aplikácia pre webové rozhranie.

Serverová aplikácia bude realizovaná ako Java server aplikácia spúšťaná pod operačným systémom Linux Ubuntu 12.04server. Úlohou serverovej aplikácie bude prijímať živé televízne vysielanie, spracovávať ho a ďalej poskytovať podľa požiadaviek. Požiadavky bude prijímať od aplikácie pre mobilné zariadenia a aplikácie pre webové rozhranie.

Serverová aplikácia bude zvládať funkcionality ako:

- poskytnutie dostupných televíznych staníc,
- transkódovanie prijímaného televízneho vysielania do http streamov,
- poskytnutie živého vysielanie požadovanej televíznej stanice,
- sprostredkovať spustenie televízneho vysielania z jedného zariadenia na druhom zariadení,
- sprostredkovať poskytnutie doplňujúcich informácií o programoch.

Aplikácia pre mobilné zariadenia bude realizovaná ako Android aplikácia. Úlohou aplikácie pre mobilné zariadenia bude primárne sprostredkovať požiadavky od používateľa k serveru. Táto aplikácia bude prehrávať poskytnuté televízne vysielanie a sprostredkovať požiadavky používateľa ako sú:

- získanie dostupných televíznych staníc,
- spustenie vysielania požadovanej stanice na serveri,
- spustenie živého televízneho vysielania na inom zariadení,
- zobrazenie elektronického programového sprievodcu k programom.
- zobrazenie doplňujúcich informácií o programe,
- spustiť televízne vysielanie z iného zariadenia na základe jeho požiadavky.

Aplikácia pre webové rozhranie bude realizovaná ako php aplikácia. Úlohou aplikácie bude primárne spravovanie používateľov, zariadení, streamov, ale bude možné pomocou nej

využiť aj funkcie poskytované aplikáciou pre mobilné zariadenia. Aplikácia pre webové rozhranie bude schopná zvládať funkcionality ako:

- pridanie nového používateľa,
- spárovanie zariadení,
- získanie dostupných televíznych staníc,
- spustenie vysielania požadovanej stanice na serveri,
- zobrazenie elektronického programového sprievodcu k programom,
- zobrazenie doplňujúcich informácií o programe,
- zobrazenie štatistík o sledovaní jednotlivých staníc.

Požiadavky na vytváraný systém môžeme rozdeliť na funkcionálne a nefunkcionálne.

Funkcionálne požiadavky:

- Vytvorený systém musí zvládnuť prehrať živé televízne vysielanie na mobilnom zariadení.
- Jednotlivé časti systému musia zvládať funkcionality opísané vyššie v tejto kapitole.
- Server musí zvládnuť transkódovať vysielanie troch televíznych staníc súčasne.
- Poskytované streamy televíznych staníc musia byť spustiteľné na štandardných mobilných zariadeniach s operačným systémom Android a prehrávačom médií VLC media player.

Nefunkcionálne požiadavky:

- Vytvorený systém musí mať jednoduché používateľské rozhranie vo všetkých troch častiach systému a používateľské rozhranie jednotlivých častí systému sa musí držať zaužívaných konvencií.
- Vytvorený systém musí poskytovať základnú úroveň bezpečnosti – musí byť odolný voči neautorizovanému prístupu.
- V prípade vyskytnutia chyby ju musí vytvorený systém rozpoznať, vyriešiť a zrozumiteľne informovať používateľa.

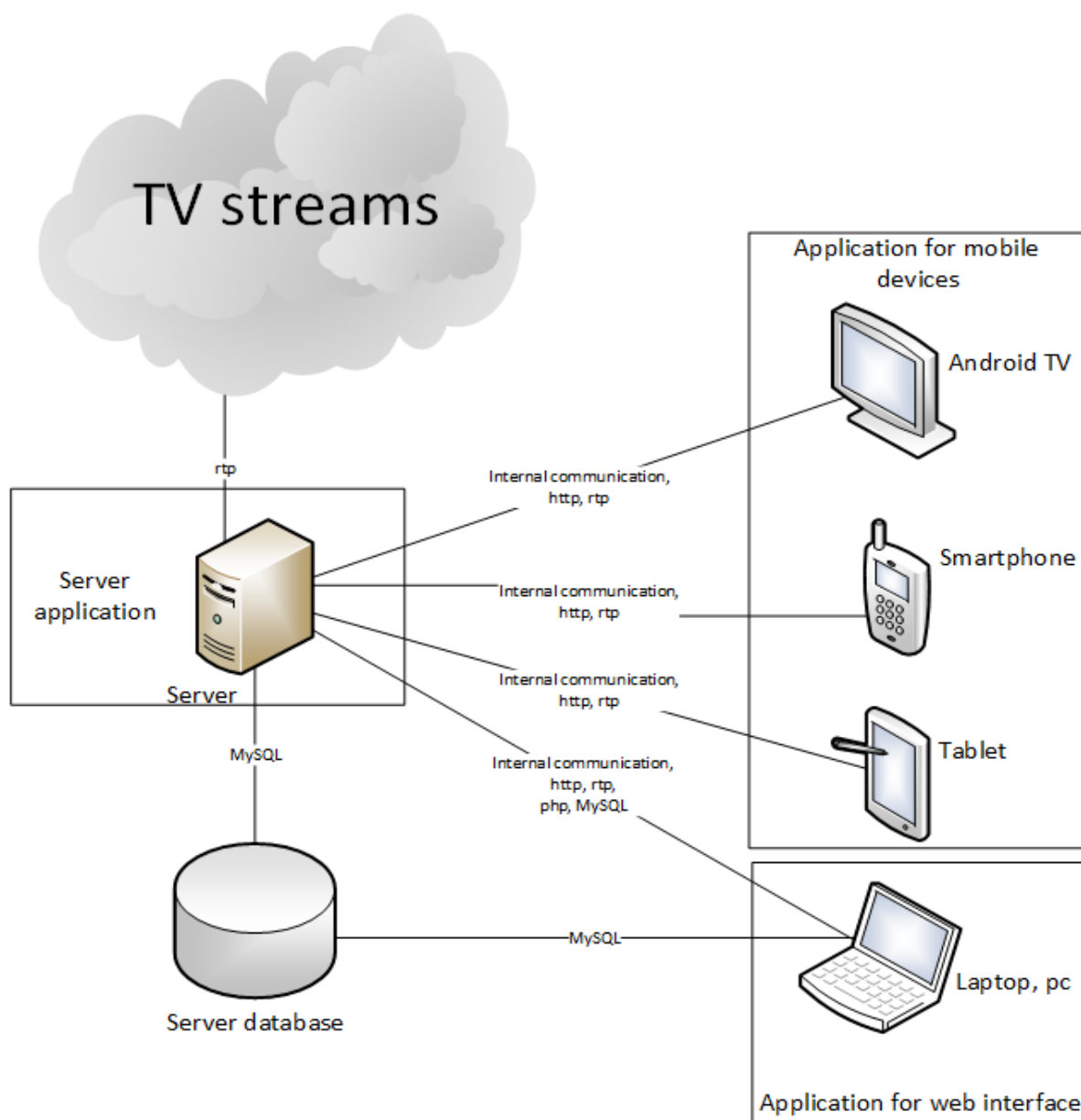
Vytváraný systém budeme navrhovať s ohľadom na špecifikáciu požiadaviek a v závere zhodnotíme splnenie jednotlivých požiadaviek.

## 4.Návrh riešenia

---

V nasledujúcej časti priblížime návrh vytváraného systému, aby bolo zrejmé, z akých častí sa bude skladať.

Na obrázku obr.4.1 sa nachádza model nami navrhovaného systému. Zdroj televízneho obsahu sú dostupné internetové streamy, ktoré následne bude spracovávať server (restreamovať, transkódovať). Na serveri sa o toto bude starať aplikácia, ktorá bude taktiež obsluhovať požiadavky od pripojených mobilných zariadení. Server si bude uchovávať svoje údaje (používatelia, dostupné stanice, pripojené zariadenia,...) v databáze. Na mobilných zariadeniach a Android TV sa bude o komunikáciu so serverom starať aplikácia pre platformu Android, prípadne web aplikácia v bežnom internetovom prehliadači.



Obr.4.1: Model navrhnutého systému

## 4.1. Serverová aplikácia

Aplikácia zabezpečujúca správnu funkčnosť servera musí zvládnuť spracovať TV signál prijatý v našom prípade z internetového zdroja a tento signál ďalej restreamovať a transkódovať podľa požadovaného rozlíšenia. Obsah, ktorý bude streamovať pre klientské zariadenia, bude aj manažovať, aby nenastala situácia, keď by z dôvodu príliš veľkého množstva požiadaviek na sledovanie rôznych kanálov nastalo zlyhanie systému. Ak by aj nastalo zlyhanie systému, tento musí byť schopný zlyhanie vyriešiť a znovu sa spustiť.

Server bude ďalej schopný komunikovať s databázou, ktorej logický model sa nachádza na obr.4.2.

Server bude schopný komunikovať aj s aplikáciou pre platformu Android, ktorá bude používaná na mobilných zariadeniach a na Android TV, prípadne s web aplikáciou, ktorá bude používaná v bežnom internetovom prehliadači. Komunikovať bude len so zariadeniami, ktoré budú autorizované, respektíve ktoré budú využívať autorizovaní používatelia.

## **4.2. Aplikácia pre platformu Android**

V aplikácií pre platformu Android bude mať používateľ možnosť prehrať si niektorý z dostupných streamov, či už na svojom zariadení, alebo na zariadení, ktoré je s tým jeho spárované. V prípade, že používateľ má záujem sledovať niektorú zo staníc, ktoré sa aktuálne nenachádzajú v dostupných streamoch, bude môcť vyslať požiadavku na server na spustenie streamovania požadovanej televíznej stanice. Ďalej si používateľ bude mať možnosť pozrieť elektronického programového sprievodcu pre dostupné stanice, prípadne bude mať možnosť získať doplňujúce informácie o filmoch z filmovej databázy IMDb.

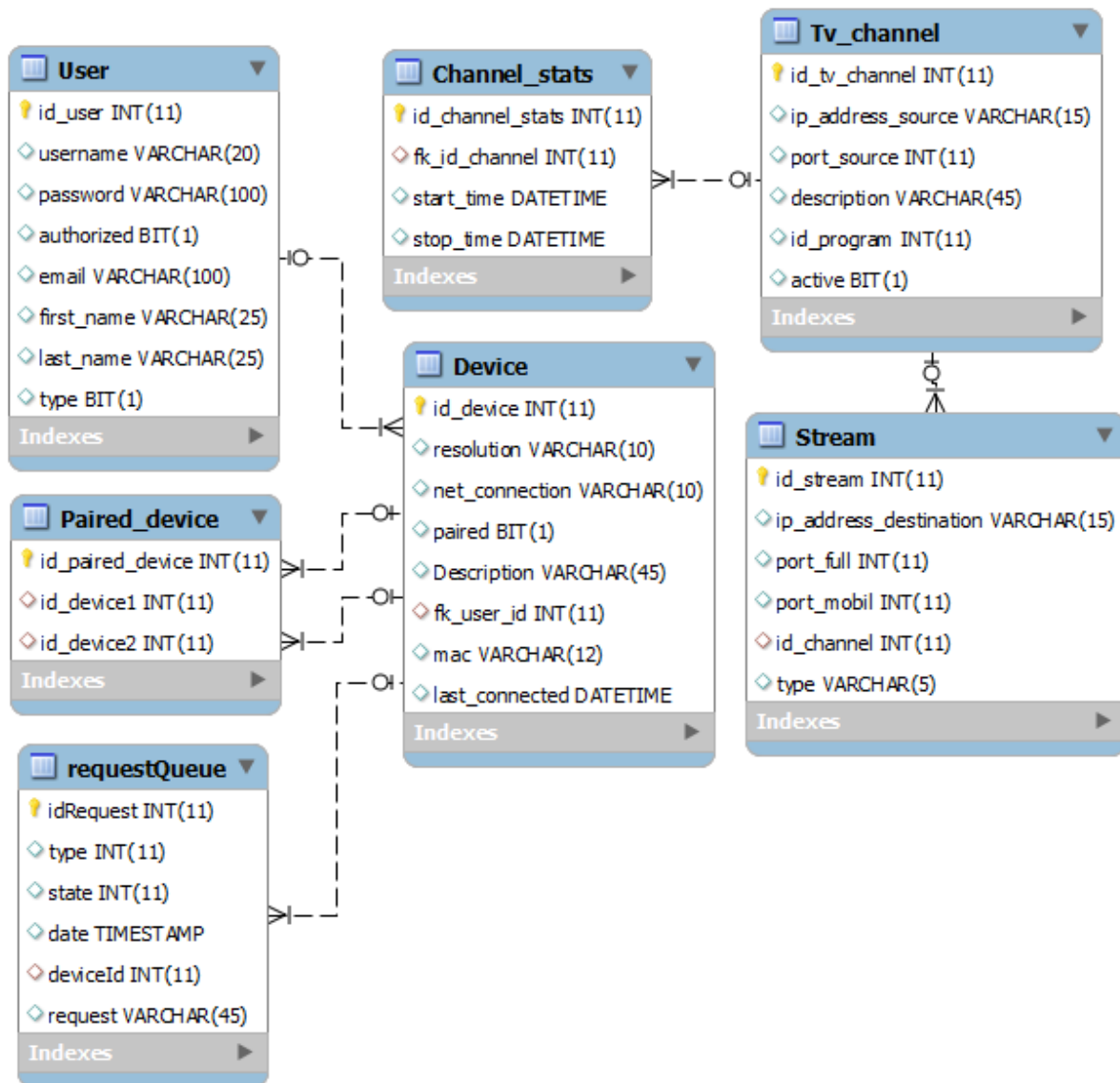
## **4.3. Web aplikácia**

Úlohou web aplikácie bude čiastočne nahrádzať klientsku úlohu mobilnej aplikácie a hlavne bude slúžiť na spravovanie systému. Pomocou aplikácie pre webové rozhranie bude možné spravovať používateľov, zariadenia a streamované televízne stanice. Vo webovej aplikácii bude môcť používateľ taktiež sledovať živé televízne vysielanie z dostupných staníc. Web aplikácia bude taktiež ponúkať funkcie ako elektronický programový sprievodca, doplňujúce informácie o filmoch z filmovej databázy, ale budú tu aj štatistiky o sledovaní televíznych staníc.

## **4.4. Databáza**

V databáze sa budú uchovávať informácie, s ktorými bude pracovať serverová aplikácia, či už priamo, alebo len ako sprostredkovateľ. Mobilná aplikácia nebude mať priamy

prístup do databázy. Logický model databázy sa nachádza na obrázku obr.2. Aplikácia pre webové rozhranie bude do databázy pristupovať, aby boli zabezpečené správčovské funkcie.



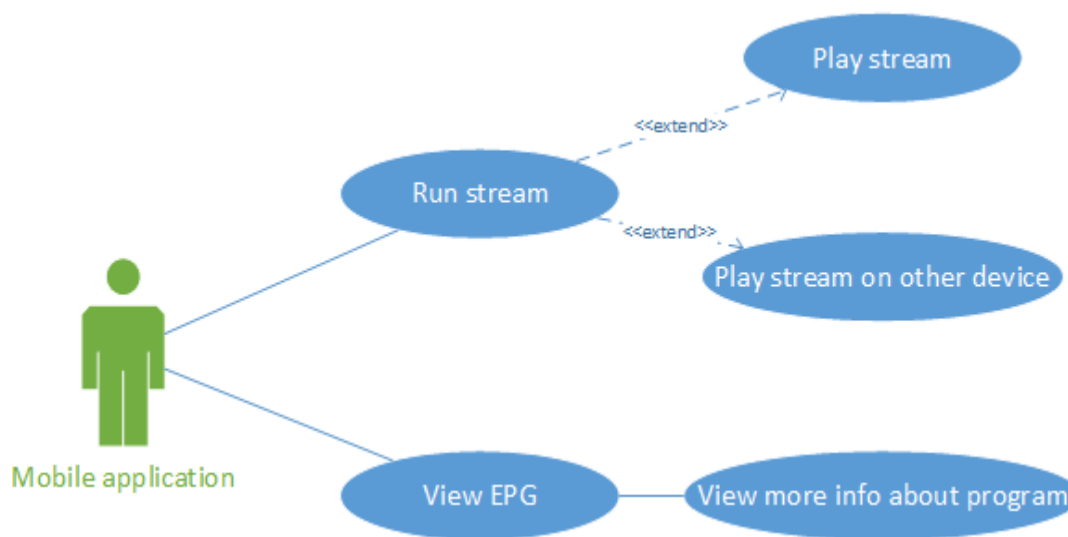
Obr.4.2: Logický dátový model

## 4.5. Prípady použitia

Diagram prípadov použitia mobilnej aplikácie je znázornený na obr.4.3. Používateľ má prostredníctvom mobilnej aplikácie možnosť spustiť stream a následne ho spustiť buď na svojom zariadení, alebo na inom zariadení, ktoré je prihlásené do systému a spárované s jeho zariadením. Ďalej má používateľ možnosť zobrazit' si elektronického programového



sprievodcu, z ktorého si môže následne zobrazit’ doplnkové informácie o jednotlivých programoch.



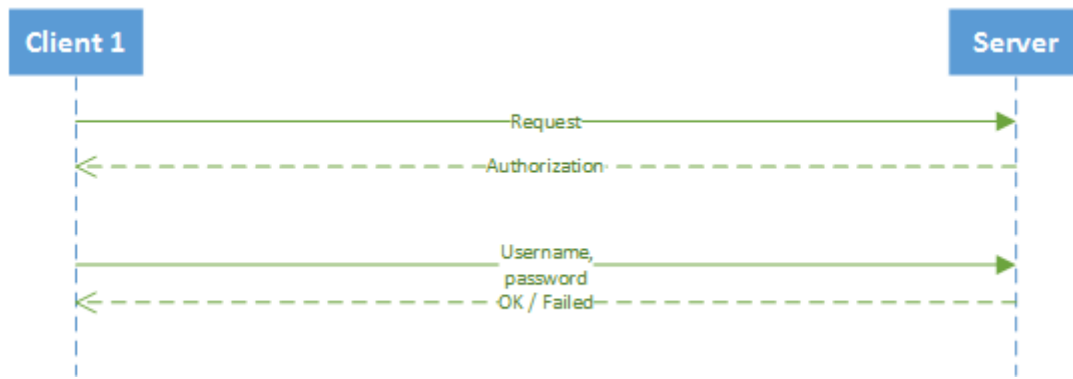
Obr.4.3: Diagram prípadov použitia mobilnej aplikácie

## 4.6. Opis komunikácie

V nasledujúcej časti dokumentu opíšeme jednotlivé prípady komunikácie medzi jednotlivými časťami vytváraného systému. Opis komunikácie bude slovný, ale aj formou diagramov.

### 4.6.1. Autorizácia do systému

Pre zachovanie bezpečnosti systému sa bude potrebné pred použitím autorizovať. Autorizácia bude prebiehať formou zadania prihlasovacích údajov, ako sú používateľské meno a heslo. Proces autorizácie bude iniciovať vždy klient požiadavkou smerom k serveru. Na obr.4.4 sa nachádza diagram toku správ prebiehajúcich medzi serverom a klientskym zariadením počas autorizácie.



Obr.4.4: Diagram toku správ počas autorizácie.

#### 4.6.2. Spustenie televízneho vysielania

Ak bude klient autorizovaný na serveri, bude môcť serveru zadávať požiadavky. Jedna z možných požiadaviek bude aj spustenie televízneho vysielania na klientskom zariadení.

Na úvod musí klient zažiadať o zoznam aktuálne dostupných staníc na serveri (playlist). Server následne odošle klientovi playlist v JSON objekte. Po vybratí kanálu na spustenie klient odošle serveru požiadavku s id požadovaného kanálu. Po prijatí požiadavky server vyhodnotí požiadavku a v prípade dostatku prostriedkov na spustenie streamovania odošle klientovi JSON objekt s informáciami potrebnými k spusteniu požadovaného kanála. Na obr.4.5 sa nachádza diagram toku správ prebiehajúcich medzi serverom a klientským zariadením počas požiadavky na spustenie televízneho vysielania.



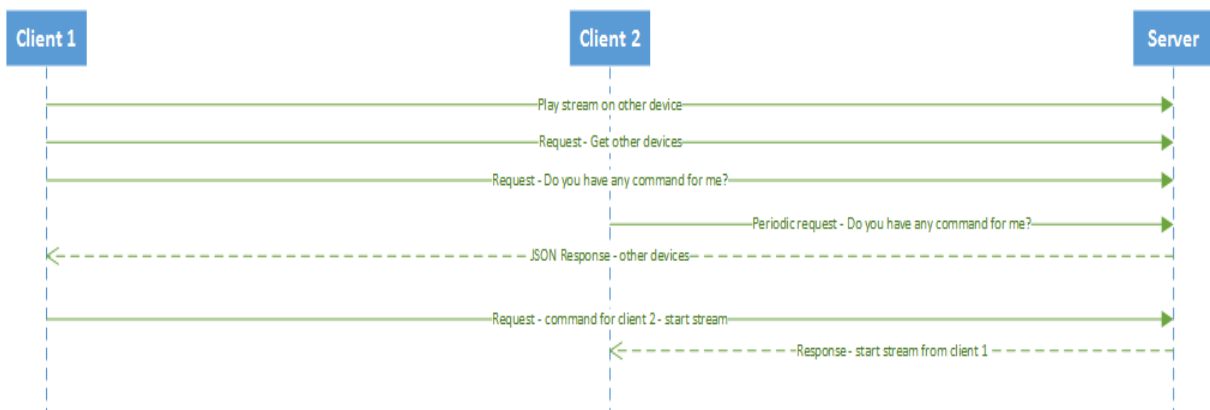
Obr.4.5: Diagram toku správ počas spustenia streamu

### 4.6.3. Prenesenie televízneho vysielania na iné zariadenie

V prípade, že používateľ bude chcieť spustiť televízne vysielanie na inom zariadení, než momentálne obsluhuje, musia byť obe zariadenia spárované.

Prenesenie televízneho vysielania na iné zariadenie bude iniciovať klientske zariadenie 1. To bude musieť najskôr vyslať požiadavku k serveru na získanie zoznamu spárovaným zariadení. Server mu odpovie formou JSON objektu, v ktorom sa bude nachádzať zoznam spárovaných zariadení. Po výbere klient zašle znovu požiadavku k serveru na spustenie požadovaného kanálu na klientskom zariadení 2. Server následne zadá príkaz klientskemu zariadeniu 2 na spustenie požadovaného televízneho vysielania.

Aby bola zabezpečená komunikácia zariadení so serverom pre spustenie televízneho vysielania z iného zariadenia, bude potrebné zasielať od každého zariadenia k serveru v pravidelných časových intervaloch overovacie požiadavky. Diagram toku správ prebiehajúcich medzi serverom a dvomi klientskymi zariadeniami počas procesu spustenia televízneho vysielania na inom zariadení je znázornený na obr.4.6.

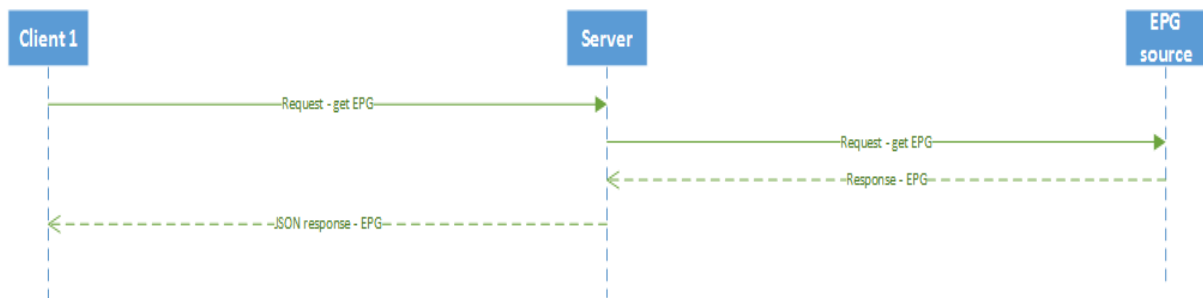


Obr.4.6: Diagram toku správ počas procesu spustenia streamu na inom zariadení

### 4.6.4. Zobrazenie elektronického programového sprievodcu

V prípade, že bude chcieť klient zobraziť elektronického programového sprievodcu k jednotlivým kanálom, bude musieť najskôr získať tieto informácie zo servera, ktorý ich musí taktiež získať z externého zdroja. V prípade, že informácie pre elektronického programového sprievodcu k požadovanému kanálu sú dostupné, server ich zašle v JSON objekte klientskemu zariadeniu, ktoré ich bude môcť zobraziť. Na obr.4.7 sa nachádza

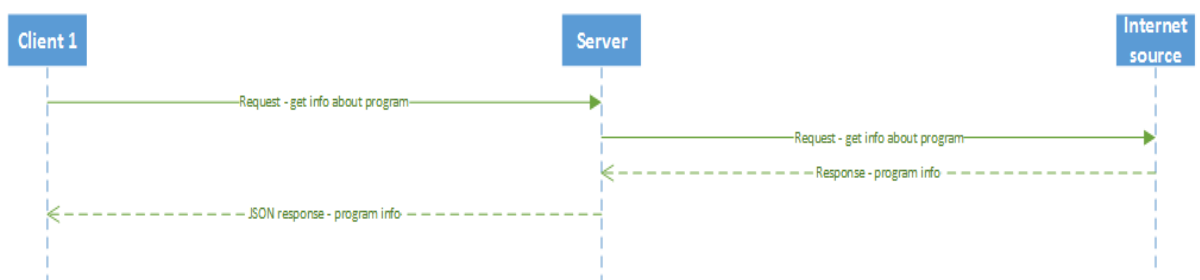
diagram toku správ prebiehajúcich medzi klientským zariadením, serverom a zdroj údajov pre elektronického programového sprievodcu.



Obr.4.7: Diagram toku správ pre získanie EPG

#### 4.6.5. Zobrazenie doplňujúcich informácií o programe

V prípade, že používateľovi nebudú stačiť informácie o programe dostupné v elektronickom programovom sprievodcovi, bude si môcť zažiadať aj o doplňujúce informácie o zvolenom programe z filmovej databázy IMDb. V žiadosti smerovanej serveru bude musieť uviesť program, o ktorom žiada doplňujúce informácie, server tieto získa z internetu a následne ich prepošle v JSON objekte ako odpoveď klientskemu zariadeniu. Na obr.4.8 sa nachádza diagram toku správ prebiehajúcich medzi klientským zariadením, serverom a internetovým zdrojom informácií počas získavania doplňujúcich informácií o programe.



Obr.4.8: Diagram toku správ pri získavaní doplňujúcich informácií o programe.

# 5. Implementácia riešenia

---

V časti implementácia podrobne opíšeme náš postup pri implementácii navrhnutého systému.

Z hľadiska implementácie je systém tvorený zo štyroch hlavných komponentov:

- Serverová aplikácia,
- Mobilná aplikácia,
- Web aplikácia,
- Databáza.

Jednotlivé komponenty sú samostatne funkčné, no sú závislé od komunikácie s ostatnými časťami systému.

## 5.1. Serverová aplikácia

Úloha serverovej aplikácie je primárne transkódovať a restreamovať prijatý televízny signál a obsluhovať požiadavky pripojených klientov, teda zvládnuť aj prístup do databázy s uloženými údajmi.

Na transkódovanie a restreamovanie prijatého televízneho signálu je možné použiť viacero nástrojov:

- Vlc – externý program vytvorený spoločnosťou VideoLAN, ktorý zvláda prehrávanie veľkého množstva formátov audiovizuálnych súborov, streamovanie audiovizuálneho obsahu s využitím viacerých protokolov (napr. http, rtp) a transkódovanie audiovizuálneho obsahu. V prípade jeho použitia by bolo nutné vo vytvorenej aplikácii spúšťať tento externý program.
- Ffmpeg/ffserver – externý program slúžiaci na streamovanie audiovizuálneho obsahu s využitím viacerých protokolov (napr. http, rtp) a transkódovanie tohto obsahu. V prípade jeho využitia by bolo nutné mať na serveri nainštalovaný ffserver a vo vytvorenej aplikácii spúšťať tento externý program.
- Java media framework – api od spoločnosti Oracle pomocou ktorého je možné používať v java aplikácii zabudované audio, video, alebo iné multimedialne applety.

- Vlcj knižnica pre programovací jazyk Java – knižnica obsahujúca funkcie na streamovanie a transkódovanie audiovizuálneho obsahu využívajúca vlc. Pre programátora teda rieši problém so spúšťaním externého programu vo vytvorenej aplikácii.

Pre implementáciu transkódovania a streamovania televízneho obsahu sme sa rozhodli využiť knižnicu vlcj pre programovací jazyk Java. Pre túto možnosť sme sa rozhodli z dôvodu, že ide o funkčné riešenie, ktoré je stále vo vývoji a poskytuje nám možnosť využívať knižničné funkcie. Nemusíme teda riešiť komunikáciu s operačným systémom za účelom spustenia externého programu.

Pre prístup k databáze z programu bude potrebné použiť konektor s databázou. Vzhľadom k rozhodnutiu používať MySQL databázu, rozhodli sme sa používať oficiálny konektor Connector/J ako ovládač pre MySQL.

Už pri vytváraní aplikácie je vhodné myslieť na dostatočné monitorovanie aplikácie, aby sa v prípade chyby výrazne znížil čas na jej identifikovanie. Monitorovanie aplikácie sa dá viacerými spôsobmi (napríklad výpisy na štandardný výstup), prípadne vytvorením vlastného API vytvárajúceho záznamy o činnosti aplikácie. V prípade našej serverovej aplikácie sme sa rozhodli využiť už vytvorené API, aby sme minimalizovali časové náklady. Z dostupných riešení sme sa rozhodli pre nástroj Log4j, čo je open-source projekt určený pre vytváranie záznamov v Java aplikáciach, ale je možné ho použiť aj v iných jazykoch.

Komunikácia s ostatnými časťami vytváraného systému je z veľkej časti zabezpečovaná pomocou JSON objektov. Vzhľadom na ich veľké rozšírenie je vhodné na ich vytváranie a spracovávanie použiť knižničné funkcie. Takéto funkcie poskytuje knižnica Gson.

Na zabezpečenie doplnkových informácií o filmoch z internetovej databázy filmov IMDb je potrebné pristupovať k tejto databáze vždy pri požiadavke na tieto informácie. Tieto informácie je možné využiť viacerými spôsobmi:

- priamo z internetovej stránky [www.imdb.com](http://www.imdb.com) – v tomto prípade nastáva problém s licenčnými podmienkami prevádzkovateľa serveru, v ktorých je zakázané takéto získavanie informácií priamo zo stránky.
- z textových súborov vytvorených zo zdrojovej databázy – prevádzkovateľ IMDb poskytuje na stiahnutie prepis databázy do textových súborov dostupný na

stiahnutie. Tieto súbory sú aktualizované raz za týždeň. V prípade použitia tejto možnosti je potrebné vytvoriť parser na roztriedenie a získanie požadovaných informácií.

- využitím dostupných api zameraných na IMDb – už implementované funkcie na roztriedenie veľkého množstva informácií dostupných v databáze IMDb a získanie požadovaných informácií. Existuje viacero takýchto api, väčšina z nich má ale obmedzený počet bezplatných dopytov. OMDbAPI ale poskytuje tieto informácie bezplatne a výsledok dopytu vráti formou JSON objektu.

### **5.1.1. Knižnica Vlcj**

Vlcj je open-source projekt vytvorený spoločnosťou Caprica Software poskytujúci funkcie VLC prehrávača od VideoLAN v prostredí Java. Pomocou Vlcj sa dá vytvoriť fungujúci server alebo klient pracujúci ako prehrávač multimediálnych súborov. Jednotlivé funkcie knižnice Vlcj sú namapované na príkazy programu VLC media player.

Pre spustenie streamovania sa používa funkcia playMedia(), ktorej argumentami sú parametre streamu (adresa na ktorú sa má streamovať, adresa zdroja, port).

Pre ukončenie streamovania sa používa funkcia stop().

### **5.1.2. Knižnica Gson**

Knižnica Gson je Java knižnica, ktorá umožňuje konvertovanie Java objektov do ich JSON reprezentácie a naopak. Gson je knižnica vyvinutá spoločnosťou google a je voľne dostupná.

JSON objekty sú otvoreným štandardným formátom na výmenu správ (ľahko čitateľných pre človeka) so zachovaním formátu jednotlivých informácií. Využívame ich na komunikáciu s klientskou Android aplikáciou, ale aj na získanie informácií z IMDb prostredníctvom OMDbAPI.

Pre získanie objektu z JSON sa používa fromJson(), ktorej argumentmi je samotný JSON a typ objektu, do ktorého chceme dáta prekonvertovať. Pre vytvorenie JSON objektu následne používame funkciu toJson(), ktorej argumentom je len objekt, z ktorého chceme vytvoriť JSON objekt.

### 5.1.3. Log4j

Log4j je nástroj pre monitorovanie stavu Java aplikácií. Má za úlohu uchovávať históriu jednotlivých činností počas vykonávania aplikácie. Jednotlivé záznamy sú chronologicky usporiadané a jednotlivé hlásky o stave činnosti aplikácie si programátor volí sám, podobne, ako štandardné testovacie výpisy. Monitorovacie výpisy sú roztriedené do kategórií ako:

- trace,
- debug,
- info,
- warn,
- error,
- fatal.

Podľa typu hlásky následne programátor zvolí funkciu na pridanie monitorovacích hlások do programu.

### 5.1.4. OMDbAPI

Keďže je zakázané využívať informácie získané priamo z IMDb stránky, ale môžu sa využívať iba informácie získané z dostupných súborov v čistom texte, bolo by potrebné implementovať funkcie na pravidelné aktualizovanie týchto údajov z IMDb serverov a ich triedenie. Tieto funkcionality už ale obsahuje OMDbAPI (<http://www.omdbapi.com>), ktoré poskytne dostupné informácie z IMDb formou JSON objektu využiteľného v našej serverovej aplikácii.

## 5.2. Aplikácia pre platformu Android

Úlohou aplikácie pre platformu Android je zabezpečiť rozhranie pre bežného používateľa systému, ktorý má záujem sledovať televízne vysielanie na mobilnom zariadení s pripojením do internetu. Aplikácia v rámci systému komunikuje iba so serverom, a preto nebolo potrebné riešiť prístup do databázy, keďže údaje z databázy získa na základe komunikácie so serverovou aplikáciou.



Program je implementovaný v programovacom jazyku Java, ktorý sa zvyčajne používa na vývoj Android aplikácií. Na samotný vývoj sme použili nástroj Android Developer Tools, čo je vlastne Eclipse rozšírený o funkcionality nevyhnutné na vývoj pre Android platformu. Minimálna podporovaná verzia operačného systému Android pre správnu funkčnosť našej aplikácie je 4.0.

Pre spojenie siete na socket bola použitá knižnica `java.net.socket`, ktorá poskytuje funkcie pre spravovanie sieťovej komunikácie.

Keďže na komunikáciu medzi serverom a aplikáciou sme využívali JSON objekty, bolo potrebné ich aj na strane Android aplikácie vytvárať a čítať. Na spravovanie JSON objektov sme použili knižnicu `org.json`, ktorá nám poskytla potrebné funkcie.

So serverovou aplikáciou má mobilná aplikácia spoločné triedy pre požiadavky, odpovede a sessions.

Aby bolo možné v aplikácií používať na prehrávanie streamov VLC prehrávač, bolo potrebné do nej doplniť spúšťač VLC, ktorého parametrom je zadaná adresa požadovaného streamu.

Aby sme zabezpečili funkcionality prenosu streamu na iné zariadenie, bol použitý podsystém, ktorý každých 10 sekúnd počúva na zvolenom porte a čaká na odpoveď, či nie je potrebné spustiť stream na základe požiadavky z iného zariadenia.

### **5.2.1. Spustenie prehrávania živého televízneho obsahu**

Prehrávanie streamu je vyriešené pomocou programu VLC media player beta pre operačný systém Android. Po kliknutí na tlačidlo spustenia streamu sa najskôr zašle požiadavka o štarte streamu na server a následne sa spustí VLC prehrávač. Ten sa pripojí na adresu streamu, ktorá príde v odpovedi od servera na požiadavku.

V prípade, že si používateľ zvolí možnosť prehratia streamu zo zadanej adresy, proces zaslania požiadavky na server je preskočený a spustí sa priamo prehrávač VLC so zadanou adresou od používateľa.

### **5.2.2. Spustenie streamu na inom zariadení**

Po kliknutí na spustenie streamu na inom zariadení vyšle aplikácia na server požiadavku pre získanie dostupných spárovaných zariadení. V odpovedi zaslanej formou JSON objektu získa tieto zariadenia, ktorých názvy následne vypíše pre používateľa na

obrazovku. Po výbere zariadenia aplikácie odošle požiadavku na server, v ktorej sa nachádza označenie zariadenia, na ktorom chce používateľ spustiť stream a id kanálu, ktorý tam má byť spustený.

Druhé zariadenie po prijatí požiadavky na spustenie streamu zo servera spustí prehrávač VLC, ktorý sa pripojí na adresu streamu, ktorú mu poskytne server v príkaze.

### **5.2.3. Zobrazenie elektronického programového sprievodcu**

Po zvolení možnosti zobrazenia elektronického programového sprievodcu aplikácia spustí internetový prehliadač, v ktorom následne otvorí adresu servera, na ktorej sa nachádza webová aplikácia s dostupným elektronickým programovým sprievodcom. O ďalšiu funkcionálnosť počas prezerania televízneho programu sa teda stará prehliadač v mobilnom zariadení a webová aplikácia na serveri.

## **5.3. Aplikácia pre webové rozhranie**

Cieľom webovej aplikácie je ponúknuť registrovanému používateľovi funkcie TV stream a TV program a administrátorovi možnosť správy používateľov a zobrazovanie štatistík o programe. Aplikácia je prepojená s databázou tímového projektu a s databázou `epg_guide`, kde sú uložené informácie o TV programe.

Program je implementovaný v programovacom jazyku PHP, ktorý sa používa na vývoj webových aplikácií.

### **5.3.1. Admin users**

Po kliknutí na ponuku Admin user sa používateľovi zobrazí stránka pre správu používateľov. Nachádza sa tu formulár pre pridávanie nových používateľov. Hlavným rozdielom oproti funkcii registrácia používateľov je možnosť pridať používateľa typu admin. Na stránke Admin user sa nachádza aj zoznam všetkých registrovaných používateľov v systéme s možnosťou ich odstránenia zo systému.

### **5.3.2. TV stream**

Po kliknutí na ponuku TV stream sa používateľovi zobrazí stránka zoznam staníc. Funkcia TV stream nie je momentálne ešte vo finálnej verzii. Momentálne ponúka len možnosť prehrať video zobrazené na stránke. Vo finálnej verzii bude funkcia TV stream umožňovať používateľom spustiť TV stream vybraného programu vo webovej aplikácii.

### **5.3.3. Statistics**

Po kliknutí na ponuku Statistics sa používateľovi zobrazí stránka o štatistikách systému. Táto funkcia bude slúžiť hlavne na zobrazovanie štatistík o používaní streamov televíznych staníc. Vďaka týmto štatistikám sa bude môcť aplikácia priebežne optimalizovať. Zdrojom dát pre graf sú informácie uložené v tabuľke Channel\_stats., kde sa priebežne ukladajú spustené streamy (ak sa spustí stream, do tabuľky Channel\_stats sa uloží id spusteného programu streamu spolu s informáciou o začatí streamu a koniec streamu, po ukončení streamu). Generovanie grafu nám umožňuje webová stránka [www.amcharts.com](http://www.amcharts.com), ktorá sa zaoberá vytváraním grafu. Jej hlavnou výhodou je možnosť generovanie grafov na základe dát z databázy MySQL a optimalizácia na viaceré webové prehliadače.

### **5.3.4. Programme**

Po kliknutí na ponuku Programme sa používateľovi zobrazí stránka TV program vybraných televíznych staníc na aktuálny deň. Dáta o programe sú uložené v databáze epg\_guide, ktorá vznikla v rámci diplomového projektu spolužiaka Jána Maťa. Ten tieto informácie vyberá z viacerých webových stránok a po parsovaní ich priebežne ukladá do databázy, čo zabezpečuje aktuálne informácie v databáze. Hlavným kritériom pre výber programu je atribút ID\_channel, ktorý nám určuje televíznu stanicu daného programu a atribút starttime, teda kedy sa program začína. Nevýhodou databázy je čiastočný problém s kódovaním, čo má za následok zlé zobrazenie niektorých znakov. Ďalšou nevýhodou databázy sú použité dátové typy pri atribútoch v tabuľkách, napríklad starttime nie je typu datetime, ale varchar. Okrem základného TV programu aplikácia umožňuje zobrazenie aj informácie o vybranom programe po kliknutí na daný program.

ID_programme	ID_artist	ID_channel	starttime	stoptime	title	title_origin	subtitle
1	0	1	20130328160000	20130328161500	Správy RTVS o 16.00	Správy STV	
2	0	1	20130328161500	20130328162000	Po?asie		
3	0	1	20130328162000	20130328162500	Góly, body, sekundy	Góly body sekundy	
4	2	1	20130328162500	20130328171500	Sila lásky	Stum der Liebe	
5	0	1	20130328171500	20130328175000	Galileo - špeciál		
6	0	1	20130328175000	20130328181500	Duel		Vedomosti
7	0	1	20130328181500	20130328190000	5 proti 5		
8	0	1	20130328190000	20130328194500	Správy RTVS	Správy STV	
9	0	1	20130328194500	20130328195000	Po?asie		
10	0	1	20130328195000	20130328201000	Góly, body, sekundy	Góly body sekundy	

Obr.5.1: Výpis z databázy epg\_guide. Na výpise je vidieť čiastočný problém s kódovaním, ktoré má databáza.

## 5.4.Databáza

Pre implementovanie databázovej časti sme si zvolili MySQL databázu. Rozhodli sme sa tak preto, pretože je overená jej spolupráca s ďalšími komponentmi, ako je Java aplikácia na strane servera.

Štruktúra databázy vychádza z navrhnutého modelu znázorneného na obr.2.

V tabuľke User sa uchováajú záznamy o používateľoch systému. Každý používateľ je reprezentovaný svojim unikátnym identifikačným číslom, používateľským menom (najviac 20 znakov), heslom, svojou emailovou adresou, krstným menom a priezviskom. Poslednými identifikátormi používateľa sú značka, či je autorizovaný administrátorom systému a značka o type používateľa, pomocou ktorej vieme používateľov rozdeliť na administrátorov a bežných používateľov.

Každý používateľ môže vlastniť jedno, viacero, alebo žiadne registrované zariadenie. Každé zariadenie je reprezentované svojim unikátnym identifikačným číslom, rozlíšením obrazovky, pripojením do internetu (rozlišujeme mobilné pripojenie a wifi pripojenie), fyzickou adresou sieťovej karty, cudzím kľúčom na svojho vlastníka do tabuľky používateľov, popisom a značkou, či je spárované s akýmkoľvek iným zariadením. V prípade jednotlivých zariadení v databáze uchováваме aj čas posledného pripojenia.

Tabuľka Paired\_device uchováva dvojice spárovaných zariadení. Každé zariadenie môže byť spárované s jedným, s viacerými, alebo žiadnym zariadením. Každý záznam v tabuľke spárovaných zariadení obsahuje svoje unikátne identifikačné číslo a dve cudzie kľúče na zariadenia do tabuľky zariadení.

V tabuľke Tv\_channel sa uchovávajú všetky dostupné zdroje televízneho obsahu na serveri. Tieto údaje sú určené výhradne pre serverovú aplikáciu. Každý dostupný kanál je reprezentovaný svojím unikátnym identifikačným číslom, zdrojovou IP adresou a zdrojovým portom, na ktorých je dostupný, svojím popisom a identifikačným číslom programu do tabuľky programov. Posledným parametrom je značka, ktorá hovorí o tom, či je televízny kanál aktuálne využívaný, alebo nie.

Každý televízny kanál môže byť streamovaný v jednom, viacero, alebo v žiadnom steame. V tabuľke Stream sú uchovávané údaje o aktuálne vysielaných kanáloch na serveri. Po vytvorení jednotlivého streamu sa údaje o ňom uchovávajú do tabuľky Stream a sú reprezentované identifikačným číslom aktuálne vysielaného kanálu, cieľovou IP adresou, na ktorú je tento kanál vysielaný, dvoma portami pre vysielanie bez transkódingu a s transkódingom, cudzím kľúčom do tabuľky Tv\_channel na identifikačné číslo kanálu dostupného na serveri a typom vysielaného streamu (http, rtp).

O každom dostupnom televíznom kanáli na serveri môže byť uchovávaný jeden, viacero, alebo žiaden záznam v tabuľke Channel\_stats. Každý záznam v tejto tabuľke je tvorený z unikátneho identifikačného čísla záznamu, cudzím kľúčom na číslo televízneho kanálu do tabuľky Tv\_channel, časom aktivovania (spustenie streamovania) tohto kanálu a časom ukončenia streamovania kanálu.

Poslednou tabuľkou v databáze je tabuľka requestQueue. Táto tabuľka uchováva záznamy o všetkých požiadavkách od zariadení smerom k serveru. Každý záznam v tabuľke je tvorený z unikátneho identifikačného čísla záznamu, typom požiadavky podľa triedy MessageTypes, stavom požiadavky, časovou pečiatkou, ktorá hovorí o jeho vzniku a cudzím kľúčom na identifikačné číslo zariadenia do tabuľky zariadení, ktorý hovorí o zariadení, na ktoré je požiadavka smerovaná.

## 6. Overenie riešenia

---

Funkčnosť vytvoreného systému sme testovali priebežne počas jeho vytvárania, keď sme testovali jednotlivé časti individuálne a po vytvorení komunikácie medzi komponentmi bol systém vždy otestovaný kompletne.

Pomocou záznamov z nástroja Log4j sme vedeli identifikovať jednotlivé aktivity, ktoré vykonával server počas testovania. Príklad záznamov z nástroja Log4j je na obr.6.1.

```
2014-05-20 21:49:39 INFO ClientMonitoring:24 - Connected clients:
2014-05-20 21:49:39 INFO ClientMonitoring:25 - -----
2014-05-20 21:49:39 INFO ClientMonitoring:31 - -----
2014-05-20 21:49:39 INFO SessionMonitoring:25 - Sessions:
2014-05-20 21:49:39 INFO SessionMonitoring:26 - -----
2014-05-20 21:49:39 INFO SessionMonitoring:31 - -----
2014-05-20 21:49:39 INFO StreamMonitoring:26 - Active Streams:
2014-05-20 21:49:39 INFO StreamMonitoring:27 - -----
2014-05-20 21:49:39 INFO StreamMonitoring:32 - -----
2014-05-20 21:49:39 DEBUG CleanerMonitor:33 - Cleaner worker
2014-05-20 21:49:39 DEBUG AppContext:25 - Resetting context
2014-05-20 21:49:39 INFO StreamServer:120 - ServerSocket.run() <127.0.1.1>:<1234>
2014-05-20 21:49:44 INFO StreamMonitoring:26 - Active Streams:
2014-05-20 21:49:44 INFO StreamMonitoring:27 - -----
2014-05-20 21:49:44 INFO StreamMonitoring:32 - -----
2014-05-20 21:49:44 INFO ClientMonitoring:24 - Connected clients:
2014-05-20 21:49:44 INFO ClientMonitoring:25 - -----
2014-05-20 21:49:44 INFO ClientMonitoring:31 - -----
2014-05-20 21:49:44 INFO SessionMonitoring:25 - Sessions:
2014-05-20 21:49:44 INFO SessionMonitoring:26 - -----
2014-05-20 21:49:44 INFO SessionMonitoring:31 - -----
2014-05-20 21:49:49 INFO SessionMonitoring:25 - Sessions:
2014-05-20 21:49:49 INFO SessionMonitoring:26 - -----
2014-05-20 21:49:49 INFO SessionMonitoring:31 - -----
2014-05-20 21:49:49 INFO StreamMonitoring:26 - Active Streams:
2014-05-20 21:49:49 INFO StreamMonitoring:27 - -----
2014-05-20 21:49:49 INFO StreamMonitoring:32 - -----
2014-05-20 21:49:49 INFO ClientMonitoring:24 - Connected clients:
2014-05-20 21:49:49 INFO ClientMonitoring:25 - -----
2014-05-20 21:49:49 INFO ClientMonitoring:31 - -----
2014-05-20 21:49:50 INFO ServerThread:61 - ServerThread() [Socket=/158.195.196.165:58002
2014-05-20 21:49:50 INFO ApplicationConfiguration:162 - Configuration properties: ConfigurationParameters [serverPort=1234, port={config.Port@5ffe49b5, config.Port@2f9
9304f, config.Port@95047f2, config.Port@10fd3ce3, config.Port@762c93cb}, ipAddress=147.175.204.27, description=Server skuska konfiguracie, pathToPlaylist={resources/pl
yList.txt, maxStreams=2}
2014-05-20 21:49:50 DEBUG ServerThread:101 - Authorization Object: Authorization [username=, password=, deviceMac=08:08:C2:F3:24:59]
2014-05-20 21:49:50 INFO ServerThread:432 - validateLogin() login: [password=
2014-05-20 21:49:50 INFO Session:58 - Created new session id: 214950661T006
2014-05-20 21:49:50 INFO ServerThread:125 - ServerThread().Client Authorized: /158.195.196.165:58002 Session:[Session [sessionId=214950661T006, deviceName=Galaxy Tab2,
deviceId=6]]
2014-05-20 21:49:50 INFO ServerThread:192 - ServerThread.handleRequest(): ComRequest [message=Play list request, type=5, channelId=0, deviceId=0, sessionId=214950661T0
06, deviceMAC=null]
2014-05-20 21:49:50 DEBUG ServerThread:158 - Request response: {"playlist":[{"id":1,"name":"Retro Music TV","address":"rtsp://8233.11.36.134:1234","port":0,"live":false},
{"id":2,"name":"Fohoda","address":"rtsp://8233.11.36.124:1234","port":0,"live":false}, {"id":3,"name":"TV Noe","address":"rtsp://8233.10.47.85:1234","port":0,"live":false}
```

Obr.6.1: Príklad výpisu záznamov z Log4j.

Počas testovania sme narážali na problém so stabilitou servera, no tento bol odstránený navýšením výpočtových prostriedkov.

Na základe vykonaného testovania vytvoreného systému, ktoré ale prebiehalo len v laboratórnych podmienkach, môžeme prehlásiť, že systém spĺňa špecifikáciu a je funkčný v deklarovanom rozsahu.

# 7. Zhodnotenie

---

V nasledujúcej časti sa nachádza zhodnotenie požiadaviek zadaných pred vytvorením systému a celkové zhodnotenie tímového projektu.

## 7.1. Zhodnotenie požiadaviek

V nasledujúcej časti zhodnotíme splnenie požiadaviek na systém vytvorených v časti špecifikácia požiadaviek.

### 7.1.1. Funkcionálne požiadavky

Po implementovaní a otestovaní môžeme konštatovať, že funkcionálne požiadavky kladené na systém sú splnené.

- Vytvorený systém je schopný prehrať živé televízne vysielanie na mobilnom zariadení s operačným systémom Android. Televízne vysielanie je prijímané na serveri a ďalej streamované pre mobilné zariadenia, alebo pre Android TV. Televízne vysielanie je navyše možné prehrať aj na ľubovoľnom osobnom počítači s nainštalovaným prehrávačom VLC media player.

### 7.1.2. Požiadavky pre server

- Server je schopný pomocou serverovej aplikácie poskytnúť pripojeným zariadeniam zoznam dostupných televíznych staníc.
- Živé televízne vysielanie je prijímané vo forme rtp streamov. Na serveri sú tieto streamy transkódované do http formátu, a tak je splnená aj požiadavka transkódovania.
- Server je schopný zariadiť spustenie požadovaného televízneho vysielania na inom zariadení, pokiaľ je požadované zariadenie zapnuté a pripojené k serveru.
- Server dokáže poskytnúť doplňujúce informácie o programoch, ktoré sa nachádzajú vo filmovej databáze IMDb. Údaje čerpá priamo z filmovej databázy IMDb pomocou dostupného API.
- Server zvládne transkódovať a streamovať tri televízne stanice súčasne, čo sa podarilo vďaka navýšeniu výpočtových prostriedkov na serveri.

### **7.1.3. Požiadavky pre mobilnú aplikáciu**

- Aplikácia pre mobilné zariadenie je schopná získať zoznam dostupných televíznych staníc zo servera a zobrazit' ich v zrozumiteľnom formáte používateľovi.
- Aplikácia pre mobilné zariadenie dokáže spustiť streamovanie požadovanej televíznej stanice na serveri, pokiaľ má server na toto streamovanie dostupné prostriedky. Po spustení streamovania je aplikácia schopná spustiť aj prehrávanie tejto televíznej stanice.
- V prípade, že sú súčasne pripojené viaceré zariadenia, ktoré sú navzájom spárované, je možné z jedného zariadenia spustiť živé televízne vysielanie na inom zariadení. V takomto prípade ide požiadavka na spustenie vysielania cez server.
- V prípade, že chce používateľ zobrazit' elektronického programového sprievodcu, aplikácia mu tieto informácie zobrazí vo webovom prehliadači. Po výbere konkrétneho programu si používateľ môže zvolit' možnosť zobrazenia doplnkových informácií o programe.
- V prípade, že niektoré iné pripojené spárované zariadenie zadá požiadavku na spustenie živého televízneho vysielania, aplikácia je schopná toto vysielanie spustiť. Predtým ale musí používateľ potvrdit', že vysielanie spustiť naozaj chce, aby sa zamedzilo neželanému spusteniu televízneho vysielania.

### **7.1.4. Požiadavky pre webovú aplikáciu**

- Pomocou webovej aplikácie je možné spravovať používateľov a zariadenia v systéme.
- Webová aplikácia ponúka možnosť spustenia požadovaného televízneho vysielania a prehrať ho priamo v prehliadači pomocou VLC plug-in.
- Vo výslednej aplikácii nie sú zobrazované štatistiky, nakoľko sme ich nestihli doimplementovať.

### **7.1.5. Nefunkcionálne požiadavky**

Všetky zadané nefunkcionálne požiadavky sú z nášho pohľadu splnené. Problematická zostala len požiadavka kladená na bezpečnosť systému, keď je



implementovaná len základná úroveň bezpečnosti, ktorá nedokáže plne zaručiť zamedzenie neautorizovanému prístupu.

## 7.2. Čo sme nestihli

Vzhľadom na slabšiu analýzu z prvej časti projektu sme museli obetovať dôkladnejšej analýze viac času aj v druhej časti projektu. Navyše počet ľudí v našom tíme sa po zimnom semestri znížil zo 6 na 4. Tieto faktory výrazne ovplyvnili prácu na projekte, a preto sme niektoré veci nedokončili do takej podoby, ako by sme si sami želali.

Najzávažnejším problémom vytvoreného systému môže byť jeho bezpečnosť. Počas dokončovania projektu sme sa sústredili na základnú funkcionality systému, a preto na kvalitné riešenie bezpečnosti nezostal čas.

Nestihli sme taktiež dokončiť implementáciu dynamického zoznamu dostupných kanálov pre webovú aplikáciu, nakoľko sme sa pre riešenie takejto funkcionality webovej aplikácie rozhodli až počas jej vývoja. Vo webovej aplikácii sa ale nachádzajú statické streamy, ktoré sú spustiteľné, takže je možné overiť túto funkcionality aspoň takýmto spôsobom.

Žiaľ, náš produkt sme nestihli dostatočne otestovať, a tak sa náhodne môže vyskytnúť nepredvídateľná chyba, ktorá môže spôsobiť zlyhanie serverovej aplikácie, no tieto prípady sme vyriešili aspoň automatickým reštartom aplikácie na serveri.

Okrem týchto spomenutých nedostatkov by bolo možné v budúcnosti rozšíriť náš projekt o množstvo doplnkových funkcií, ktoré by zvýšili jeho atraktivitu, a preto si myslíme, že tento projekt má stále potenciál na ďalší vývoj.

## 7.3. Čo sme sa naučili

Vzhľadom k faktu, že projekt sme vytvárali od začiatku, získali sme veľké množstvo poznatkov od samotného návrhu produktu, cez jeho realizáciu, až po jeho nasadenie a používanie.

Pre väčšinu z nás bola prínosom aj práca v tíme, nakoľko len jeden náš člen mal doposiaľ skúsenosti s tímovou prácou v praxi. Aj vďaka nemu sme sa naučili lepšej komunikácii v tíme, spoločnému riešeniu vzniknutých problémov a postupnosti aktivít pri vytváraní podobných produktov.

Vďaka nášmu pedagogickému vedúcemu sme získali cenné skúsenosti o vedení tímových stretnutí, keď práve stretnutia s ním nám v problematických chvíľach pomohli pohnúť sa s projektom ďalej.

Napokon sme sa naučili, že podobný projekt je potrebné realizovať vždy vopred dokonale teoreticky pripravení a veľmi dôležitá je kvalitná analýza problematiky.

Dovolíme si tvrdiť, že vďaka tomuto predmetu by sme v budúcnosti zvládli prácu na podobnom tímovom projekte s lepším využitím času a kvalitnejším celkovým výsledkom, no v rámci našich možností a vedomostí sme pre úspech projektu urobili maximum, čo sme mohli.

## 7.4. Celkové zhodnotenie

Primárnym cieľom nášho tímového projektu bolo navrhnuť a implementovať funkčný systém umožňujúci interakciu televízie s mobilnými zariadeniami. Najdôležitejšou funkcionalitou malo byť sprístupnenie živého televízneho vysielania na mobilnom zariadení v prípade, že používateľ nemá možnosť sledovať požadovaný televízny program na obrazovke svojho televízora. Táto funkcionalita je vo vytvorenom systéme implementovaná a funkčná. Rovnako implementované a funkčné sú aj ďalšie funkcionálne požiadavky, ktoré boli zadané.

Prostredie vytvoreného systému je z pohľadu bežného používateľa prehľadné a používanie používateľských častí (aplikácia pre mobilné zariadenia, webová aplikácia) je

intuitívne a jednoduché, na vykonanie základných aktivít nie je nutné vykonať prehnajú námahu.

Pozitívnu vlastnosťou vytvoreného systému je schopnosť použitia na viacerých platformách, keďže webová aplikácia je univerzálna a multiplatformová.

## 8.Zdroje

---

- [1]: Introducing Google Chromecast: <http://investvine.com/introducing-google-chromecast/>
- [2]: Google's Chromecast A Brilliant Play For The Living Room -- Especially With \$35 Price Tag: <http://www.forbes.com/sites/jasonevangelho/2013/07/24/googles-chromecast-a-brilliant-play-for-the-living-room-especially-with-35-price-tag/>
- [3]: Google Unveils \$35 Chromecast HDMI TV Stick: <http://www.cnx-software.com/2013/07/25/google-unveils-35-chromecast-hdmi-tv-stick/>
- [4]: Understanding AirPlay in Apple's iOS 4.2: [http://gadgetwise.blogs.nytimes.com/2010/11/22/understanding-airplay-in-apples-ios-4-2/?\\_r=0](http://gadgetwise.blogs.nytimes.com/2010/11/22/understanding-airplay-in-apples-ios-4-2/?_r=0)
- [5] DIAL - DIsccovery and LAunch protocol specification, Version 1.6.4, Netflix, 2012
- [6] Google Play Store - Samsung SmartView: <https://play.google.com/store/apps/details?id=com.sec.smartview.tz>
- [7] Google TV Pairing Protocol: <https://developers.google.com/tv/remote/docs/pairing?hl=sk&csw=1>
- [8] Google Play Store - Samsung SmartTV Remote: <https://play.google.com/store/apps/details?id=com.adi.remote.phone&hl=sk>
- [9] Slovenská informatická spoločnosť: Audio a video streaming - základné informácie (14.11.2013), <http://standardy.informatika.sk/node/28>
- [10] ČVÚT Praha, Fakulta elektrotechnická, Katedra telekomunikační techniky: Streamování v IP sítích (14.11.2013), <http://www.comtel.cz/files/download.php?id=3281>
- [11] Otázky 20bmds 202010-11 (14.11.2013), <http://manik.korh.cz/files/škola/bmds/otázky%20bmds%202010-11.pdf>
- [12] Rita Pužmanová: Streaming media (4): transportní protokoly RTP/RTCP (14.11.2013), <http://www.dsl.cz/clanek/60-streaming-media-4-transportni-protokoly-rtp-rtcp>
- [13] Slovenská informatická spoločnosť: Programy pre formáty MPEG-2 (14.11.2013), <http://standardy.informatika.sk/node/36>
- [14] VideoLAN organization: VLC media player (14.11.2013), <http://www.videolan.org/vlc/>
- [15] OpenRSI (14.11.2013), <http://www.openrsi.org/>
- [16] OpenPLi (14.11.2013), <http://openpli.org/>

[17]: Angry technician: How to stream every channel from freeview onto your network. (16.12.2013), <http://angrytechnician.wordpress.com/2010/07/23/how-to-stream-every-channel-from-freeview-onto-your-network/>

[18]: Pavel V. Cherenkov: Udpxy. (16.12.2013) <http://www.udpxy.com/index-en.html>