



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

НОВИ САД

Департман за рачунарство и аутоматику

Одсек за рачунарску технику и рачунарске комуникације

ЗАВРШНИ (BACHELOR) РАД

Кандидат: Павловић Бранимир

Број индекса: е12671

Тема рада: Реализација апликације за репродукцију мултимедијалних садржаја на Андроид базираном дигиталном ТВ пријемнику

Ментор рада: Проф. Др Теслић Никола

Нови Сад, месец, година



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Павловић Бранимир		
Ментор, МН:	Проф. др Теслић Никола		
Наслов рада, НР:	Реализација апликације за репродукцију мултимедијалног садржаја на Андроид базираном дигиталном ТВ пријемнику		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2012		
Издавач, ИЗ:	Ауторски репринг		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)	7/32/0/0/10/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	Андроид апликација, ДТВ пријемник		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	Опис реализације апликације за репродукцију мултимедијалног садржаја на Андроид базираном digitalnom TV пријемнику која подржава репродуковање локалног садржаја као и садржаја са ДЛНА послужиоца. Апликација такође подржава употребу превода током репродукције видео садржаја, као и листу за репродукцију мултимедијалног садржаја и могућност сортирања аудио и видео садржаја.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:		Потпис ментора
	Члан:		
	Члан, ментор:		



KEY WORDS DOCUMENTATION

Accession number, ANO:		
Identification number, INO:		
Document type, DT:	Monographic publication	
Type of record, TR:	Textual printed material	
Contents code, CC:	Bachelor Thesis	
Author, AU:	Pavlović Branimir	
Mentor, MN:	Teslić PhD Nikola	
Title, TI:	Realization of Media player application for Android based DTV	
Language of text, LT:	Serbian	
Language of abstract, LA:	Serbian	
Country of publication, CP:	Republic of Serbia	
Locality of publication, LP:	Vojvodina	
Publication year, PY:	2012	
Publisher, PB:	Author's reprint	
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	7/32/0/0/10/0/0	
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW:	Android application, STB	
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N:		
Abstract, AB:	Realization of Media player application for Android based DTV which supports playback of local media files and files from DLNA server. Application also supports using of titles during video playback, play list for audio and video files and possibility of sorting audio and video files.	
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:	
	Member:	Menthor's sign
	Member, Mentor:	



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



САДРЖАЈ

1.	Uvod	1
2.	Teorijske osnove.....	3
2.1	Android operativni sistem.....	3
2.2	DLNA poslužioci	7
3.	Koncept rešenja	8
3.1	Kreiranje korisničke sprege	8
3.2	Reprodukacija multimedijalnog sadržaja	10
3.3	Opis ciljne platforme.....	13
4.	Programsko rešenje.....	15
4.1	Pravljenje glavnog menija.....	15
4.2	Realizacija prevoda.....	19
4.3	Realizacija liste za reprodukciju	20
4.4	Realizacija kontroli za reprodukciju	21
5.	Ispitivanje i verifikacija	22
5.1	Ispitivanje i rezultati pri ispitivanju glavnog menija	22
5.2	Ispitivanje liste za reprodukciju i reprodukcije različitih tipova datoteka	23
5.2.1	Ispitivanje liste za reprodukciju	23
5.2.2	Ispitivanje reprodukcije različitih tipova datoteka	23
5.3	Ispitivanje usklađenosti prevoda sa reprodukovanim video datotekom	24
5.4	Ispitivanje pravilnog ponašanja kontrola reprodukcije.....	24
6.	Zaključak	25
7.	Literatura	26

SPISAK SLIKA

Slika 2.1 Arhitektura Android operativnog sistema	4
Slika 2.2 Životni ciklus aktivnosti	5
Slika 2.3 Primena DLNA protokola u kućnoj mreži	7
Slika 3.1 Hijerarhija objekata klase View.java I ViewGroup.java pri pravljenju izgleda aktivnosti	9
Slika 3.2 Dijagram stanja objekta klase MediaPlayer.java.....	12
Slika 3.3 Marvell BG2 SOC platforma	14
Slika 4.1 Prikaz glavnog menija sa izabranom medijatekom	18
Slika 4.2 Prikaz glavnog menija sa izabranim sortiranjem.....	19
Slika 4.3 Prikaz korišćenja prevoda.....	20
Slika 4.4 Prikaz kontrola za reprodukciju.....	21

SKRAĆENICE

- | | |
|-------------|---|
| DLNA | - <i>Digital Living Network Alliance</i> , Protokol za deljenje digitalnog sadržaja između multimedijalnih uređaja |
| HD | - <i>High Definition</i> , Visoka definicija |
| USB | - <i>Universal Serial Bus</i> , Univerzalna serijska magistrala |
| STB | - <i>Set Top Box</i> , Digitalni TV prijemnik |
| URL | - <i>Uniform resource locator</i> , Jednoobrazni identifikator resursa |
| DTV | - <i>Digital Television</i> , Digitalna televizija |
| UPnP | - <i>Universal plug and Play</i> , Skup mrežnih protokola koji dozvoljavaju multimedijalnim uređajima da ostvare međusobne veze i uspostave razne mrežne usluge |

1. Uvod

U ovom radu je realizovana aplikacija za reprodukciju multimedijalnih sadržaja na Android baziranom DTV prijemniku (u daljem tekstu STB prijemniku). Rešenje je realizovano na Marvell BG2 SOC platformi.

U okviru rada je dat pregled kreiranja korisničke sprege i podrške za reprodukciju audio i video sadržaja (u daljem tekstu multimedijalnih sadržaja).

U okviru rada su analizirane:

- realizacija prevoda koji se prikazuje prilikom reprodukcije video sadržaja,
- lista za reprodukciju multimedijalnog sadržaja,
- sortiranje multimedijalnog sadržaja.

Za ispitivanje i verifikaciju je napravljena prilagođena Java aplikacija.

Ovaj rad je sačinjen od sedam poglavlja.

U prvom poglavlju je dat kratak opis rada.

Druge poglavlje daje teorijske osnove Android operativnog sistema kao i njegove osnovne komponente, kao i kratko objašnjenje DLNA poslužioca.

U trećem poglavlju su opisane ključne komponente ovog rada, kao i osnovne klase za kreiranje korisničke sprege i reprodukciju multimedijalnih sadržaja.

Četvrto poglavlje sadrži detaljan opis realizacije korisničke sprege i podrške za reprodukciju multimedijalnog sadržaja.

U petom poglavlju su opisani načini ispitivanja i verifikovanja svih modula programske podrške.

Šesto poglavlje sadrži kratak pregled onoga što je urađeno u ovom radu i kakvi su dalji pravci razvoja.

U sedmom poglavlju dat je spisak korišćene literature za izradu rada.

2. Teorijske osnove

U ovom poglavlju je dat kratak opis Android operativnog sistema i njegovih elementarnih komponenti koje su korišćene u ovom radu, kao i kratak opis DLNA poslužioca.

2.1 Android operativni sistem

Android operativni sistem je prvobitno namenjen za mobilne uređaje, i zasnovan je na linuks (eng. Linux) operativnom sistemu. Optimizovan je za okruženje sa ograničenim resursima. Iako je Android namenjen za pametne telefone i tablet računare, njegova otvorena i prilagodljiva priroda mu dozvoljava da bude iskorišćen i u drugim elektronskim uređajima kao što su laptop računari, čitači elektronskih knjiga i što je najvažnije STB uređajima[4].

Arhitektura Android operativnog sistema (slika 2.1) je zasnovana na [3]:

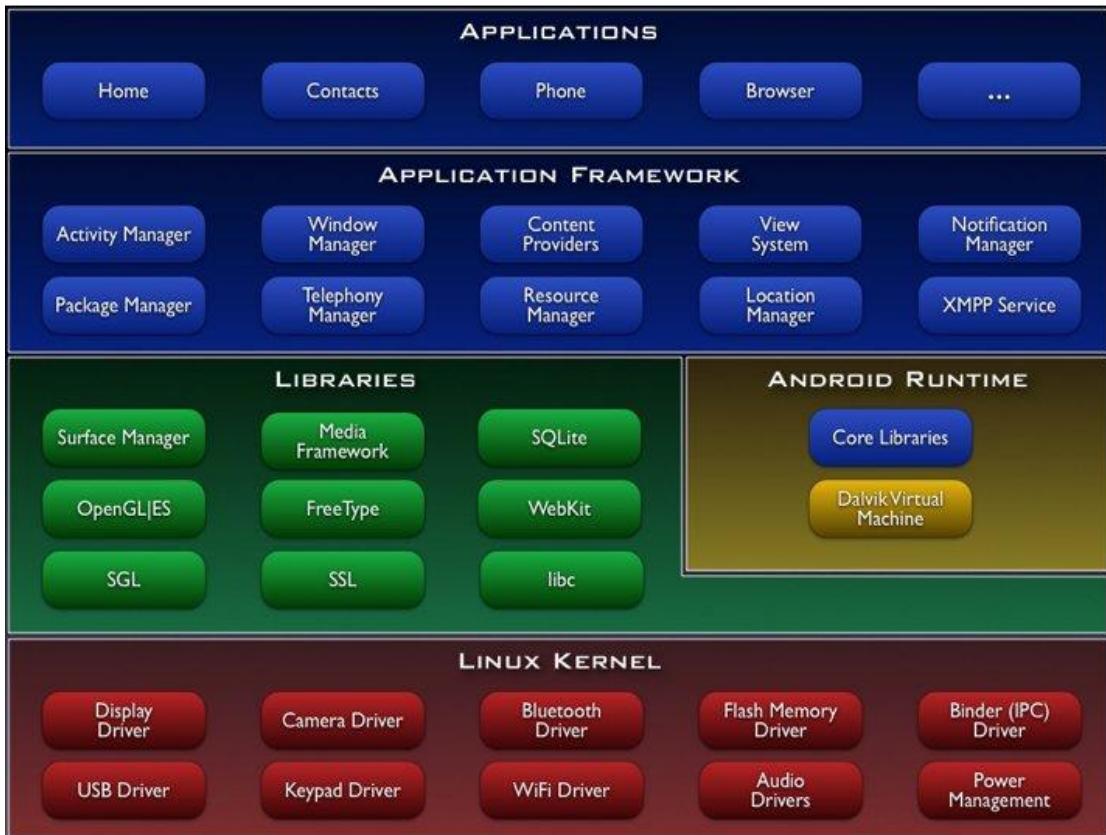
- jezgru linuksa,
- C/C++ bibliotekama u kojim se nalaze Androidova Dalvik virtuelna mašina i glavne biblioteke,
- rukovaocu aplikacijama i
- samim aplikacijama.

Elementi od kojih je moguće kreirati aplikacije [1] su:

- aktivnosti,
- primaoci emitovanih signala,
- dobavljači sadržaja i
- servisi.

Aktivnosti predstavljaju prezentacioni nivo Android aplikacije (ono što korisnik vidi). Android aplikacija može imati nekoliko aktivnosti i može da se prebacuje sa između njih u toku aplikacije. Aktivnosti imaju svoj životni tok (Slika 2.2). Aktivnostima se u Androidu upravlja kao magazinskom memorijom. Kada se nova aktivnost pokreće, postavlja se na vrh

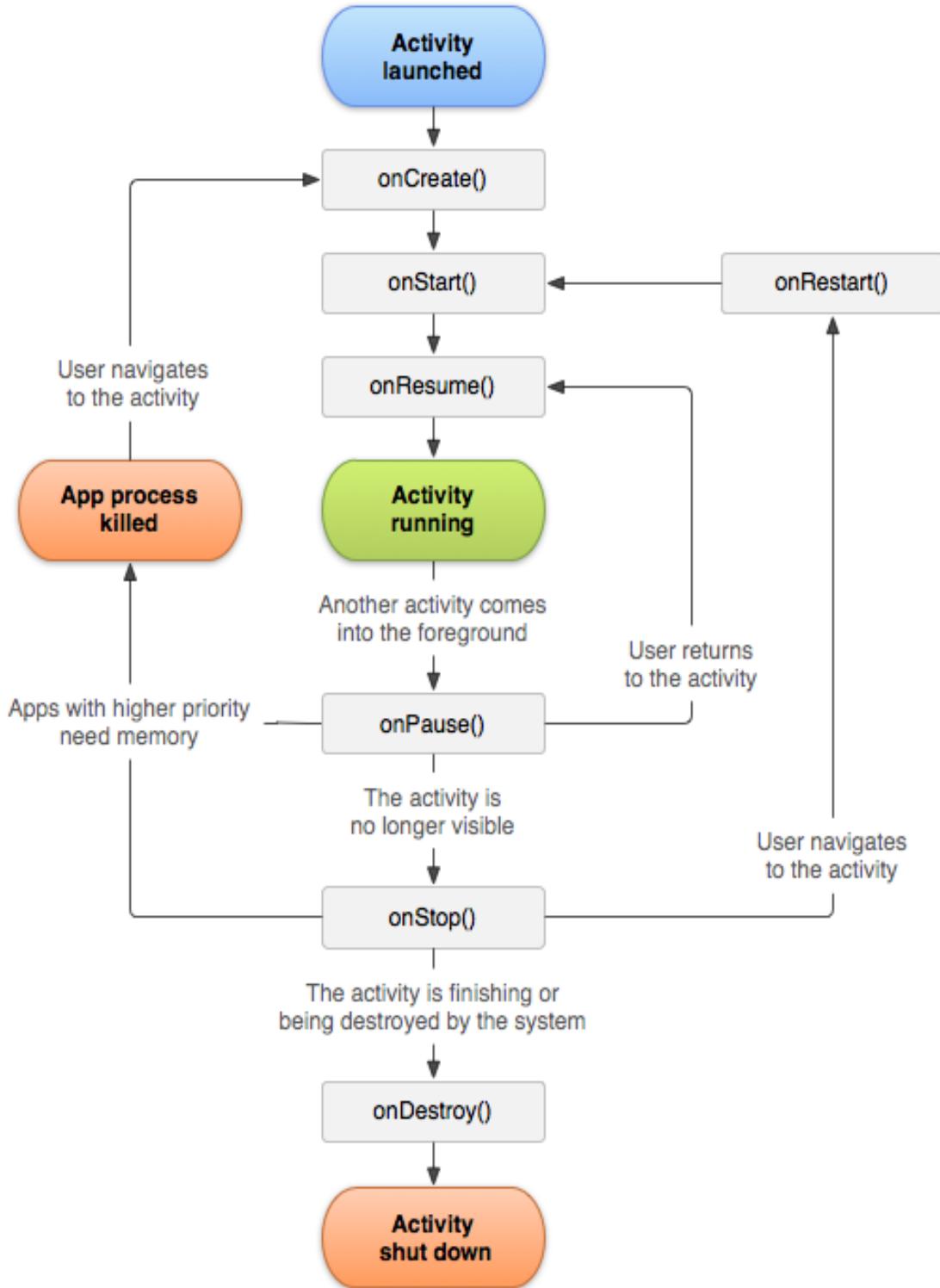
magazinske memorije aktivnosti i postaje trenutno aktivna aktivnost, dok prethodna aktivnost uvek ostaje ispod nje u magazinskoj memoriji aktivnosti i neće biti aktivna sve dok se trenutno aktivna aktivnost ne završi.



Slika 2.1 Arhitektura Android operativnog sistema

Aktivnost u osnovi ima četiri stanja [1]:

- Ako je aktivnost trenutno vidljiva na ekranu tj. na vrhu magazinske memorije aktivnosti je, ta aktivnost se naziva trenutno aktivna aktivnost.
- Ako je aktivnost izgubila fokus ali je i dalje vidljiva, onda je ona pauzirana. Pauzirana aktivnost je živa, ali operativni sistem je može nasilno završiti kada je u situaciji da ima veoma malo slobodne memorije.
- Ako je aktivnost potpuno prekinuta od strane druge aktivnosti onda je ona stopirana. Ona više nije vidljiva korisniku i često je operativni sistem može nasilno završiti ako je memorija potrebnija na drugom mestu.
- Ako je aktivnost pauzirana ili stopirana, operativni sistem može oslobođiti memoriju tako što će je „pitati“ da se završi ili je nasilno završiti. Kada se takva aktivnost prikazuje korisniku mora biti potpuno restartovana i vraćena u prethodno stanje.



Slika 2.2 Životni ciklus aktivnosti

Primaoc emitovanih signala [1] (eng. Broadcast Receiver) je glavna klasa koja prima poruke poslate preko metode `sendBroadcast()`. Aplikacija može dinamički da registruje instancu ove klase sa `Context.registerReceiver()` ili da statički prikaže implementaciju kroz `<receiver>` oznaku u `AndroidManifest.xml` datoteci. Postoje dva glavna tipa emitovanih signala koja mogu biti primljena:

- Normalni signali (signali poslati sa `Context.sendBroadcast()`) su potpuno asinhroni. Svi primaoci emitovanih signala se pokreću u nedefinisanom rasporedu, često u isto vreme.
- Naručeni signali (signali poslati sa `Context.sendOrderedBroadcast()`) se isporučuju jednom primaocu u jednom trenutku. Kako svaki primaoc izvršava svoj deo, on može da prosledi svoj rezultat sledećem primaocu, ili može da potpuno zaustavi naručeni signal tako da se ne prosleđuje drugim primaocima.

Mnogi signali potiču od operativnog sistema, npr. signal koji najavljuje da će se ekran uređaja ugasiti, da je baterija je prazna (kod mobilnih uređaja). Iako primaoc emitovanih signala ne prikazuje nista na ekranu, mogu kreirati obaveštenje na statusnoj liniji Android uređaja.

Dobavljači sadržaja [1] (eng. Content Providers) upravljaju pristupom struktuiranom skupu podataka. Oni obuhvataju podatke i pružaju mehanizam za definisanje sigurnosti podataka. Dobavljači sadržaja su standardna sprega koja povezuje podatke iz jednog procesa sa drugim procesima. Kada se želi pristupiti podacima pomoću dobavljača sadržaja, koristi se objekat klase `ContentResolver.java` da komunicira sa dobavljačem kao klijent. Dobavljač sadržaja prima zahteve za podacima od klijenata izvršava željenu akciju i vraća rezultat. Na primer svaka aplikacija sa određenom dozvolom može da kreira upit dobavljaču sadržaja da piše i čita informacije o određenom kontaktu.

Servis [1] je deo aplikacije koji predstavlja ili želju aplikacije da odradi dužu operaciju koja ne podrazumeva interakciju sa korisnikom ili da obezbedi neku funkcionalnost koju će druge aplikacije da koriste. Svaka klasa koja je i servis mora imati odgovarajuću `<service>` oznaku u `AndroidManifest.xml` datoteci. Servis se može pokrenuti sa komandom `Context.startService()` i `Context.bindService()`. Servisi kao i svaki drugi objekti, se izvršavaju u glavnoj niti procesa u kojem se nalaze. Ovo znači da, ako servis mora da izvršava neku procesorski intezivnu ili blokirajuću operaciju, ona treba da kreira svoju nit u kojoj će da radi taj posao. Za startovane servise postoje dva glavna načina u kome mogu da se izvršavaju:

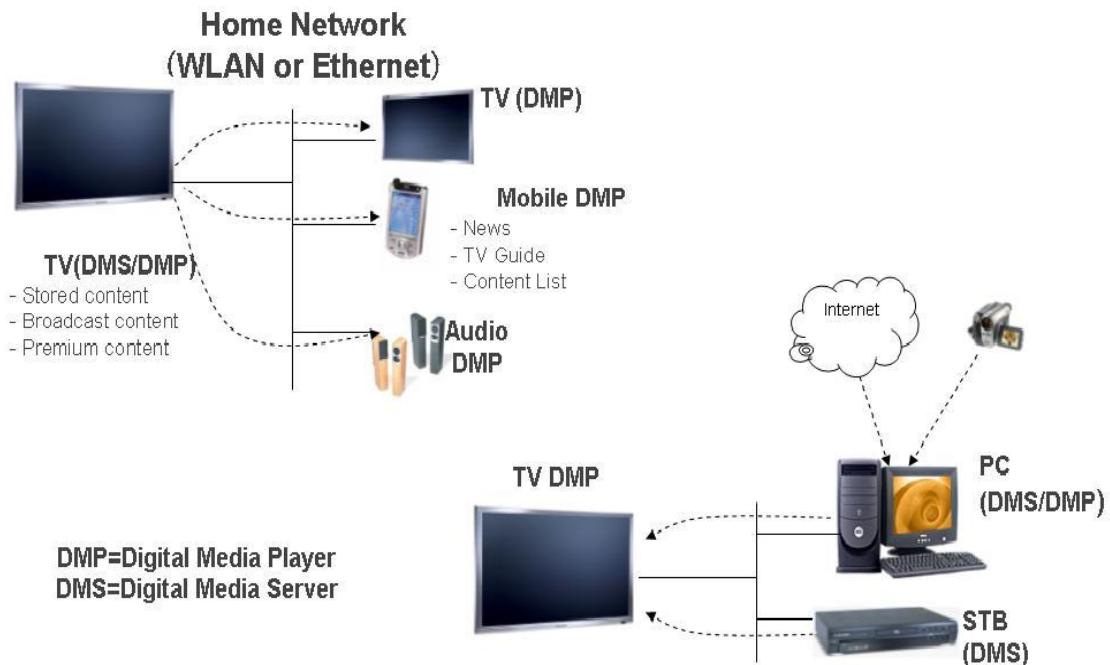
- `START_STICKY` način se koristi za servise koji se eksplicitno startuju i zaustavljaju kada je to potrebno,
- `START_NOT_STICKY` ili `START_REDELIVER_INTENT` se koriste kod servisa koji treba da se izvršava samo onoliko koliko mu treba da obradi podatke koji su mu poslati.

Servis je generalno od manjeg značaja operativnom sistemu nego trenutno aktivna aktivnost, ali od većeg značaja od aktivnosti koje su u pozadini.

2.2 DLNA poslužioci

DLNA protokol koristi UpnP protokol za upravljanje multimedijalnim sadržajem, njegovo pronalaženje i kontrolu. UPnP protokol definiše tipove uređaja koji podržavaju DLNA protokol: DLNA poslužioci, DLNA kontroleri, DLNA prikazivači (eng. renderer). UPnP protokol takođe definiše i mehanizam za pristup multimedijalnom sadržaju preko mreže.

DLNA poslužioci su uređaji koji čuvaju multimedijalni sadržaj i čine ga dostupnim drugim mrežnim uređajima. Najčešće se koristi u kućnoj mreži.



Slika 2.3 Primena DLNA protokola u kućnoj mreži

3. Koncept rešenja

U ovom poglavlju je opisan postupak pravljenja korisničke sprege i osnovne klase koje se koriste pri njegovom pravljanju, kao i ubacivanje mogućnosti reprodukcije multimedijalnog sadržaja u aplikaciju. Takođe u ovom poglavlju je dat kratak opis ciljne platforme.

3.1 Kreiranje korisničke sprege

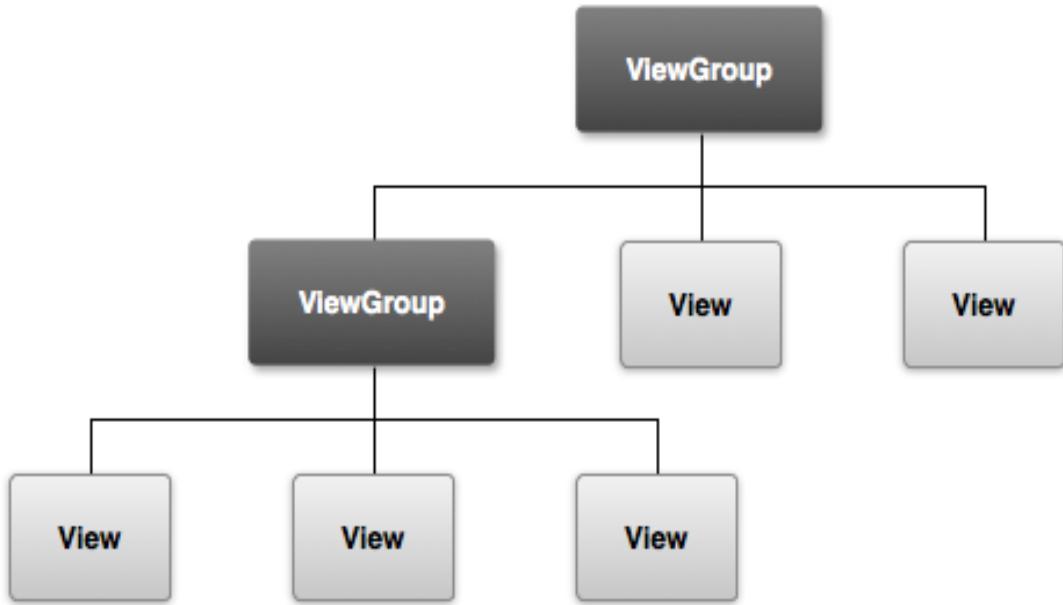
U Android operativnom sistemu korisnička sprega se kreira koristeći dve glavne klase: View.java i ViewGroup.java. Postoji mnogo potomaka klase View.java i ViewGroup.java koje se koriste i svi su naslednici klase View.java [3].

Objekti klase View.java su osnovne jedinice za kreiranje korisničke sprege u Android operativnom sistemu [1]. Klasa View.java služi kao osnova za njene podklase koje se nazivaju „vidžeti“ (eng. widgets), koje nude već implementirane objekte za kreiranje korisničke sprege kao što su dugmadi i polja za unos teksta. Klasa ViewGroup.java klasa služi kao osnova za njene podklase koje su klase za prostorni razmeštaj komponenti na ekranu (eng. layouts), koje nude različite opcije za prostorni razmeštaj komponenti kao što su: linearni razmeštaj, relativni razmeštaj, tabelarni.

Objekat klase View.java je struktura podataka koja sadrži parametre za prostorni razmeštaj i sadržaj specifične pravougaone oblasti ekrana. Objekat klase View.java upravlja svojim merenjima, iscrtavanjem, promenom fokusa, skrolovanjem i interakcijom sa korisnikom za pravougaonu oblast u kojoj se nalazi na ekranu. Kao objekat korisničke sprege, on je takođe i tačka interakcije sa korisnikom i primalac događaja koji potiču od korisnika.

U Android operativnom sistemu, izgled aktivnosti se definiše koristeći hijerarhiju objekata klase View.java i ViewGroup.java. [1] kao što je pokazano na slici 3.1. Ova

hijerarhija može biti prosta ili složena onoliko koliko je to potrebno da bude, i može se kreirati koristeći Androidov skup postojećih vidžeta i klase za prostorni razmeštaj komponenti ili se može koristiti potrebi prilagođen objekat klase View.java.



Slika 3.1 Hijerarhija objekata klase View.java I ViewGroup.java pri pravljenju izgleda aktivnosti

Da bi prikazali objekat klase View.java na ekranu uređaja, aktivnost mora pozvati metodu setContentView() sa referencom izgleda koji se prikazuje [3]. Android operativni sistem prima tu referencu i koristi je da izmeri i iscrta tu hijerarhiju na ekranu uređaja. Glavni objekat klase ViewGroup.java u ovoj hijerarhiji zahteva od njegovih potomaka da sami sebe ictaju. Potomci mogu da zahtevaju njihovu veličinu i poziciju u pretku u kome se nalaze, tj. njihov predak određuje njihovu poziciju i veličinu. Android operativni sistem obrađuje elemente klase za prostorni razmeštaj komponenti na ekranu sa vrha hijerarhije objekata ka dnu, instancira objekte i dodaje ih njihovim roditeljima (objektima klase ViewGroup.java). Posledica ovoga je da ako postoje elementi na istoj poziciji poslednji koji je iscrtan će se nalaziti preko ostalih elemenata.

Najčešći način da se definiše korisnička sprega u Android operativnom sistemu je preko xml datoteka [1]. Ove datoteke nude strukturu za prostorni razmeštaj elemenata na ekranu koju čovek može da pročita, i koja veoma liči na html datoteku. Svaki element xml datoteke je ili objekat klase View.java ili ViewGroup.java gde su objekti klase View.java kao lišće na drvetu, a objekti klase ViewGroup.java kao grane na drvetu (vidi sliku 3.1). Ime elementa u xml datoteci je isto kao i njegov odgovarači naziv u Javi. Tako da element <TextView> kreira TextView u korisničkoj sprezi, a <LinearLayout> kreira LinearLayout prostorni razmeštaj. Na

primer prosti linearni prostorni razmeštaj sa jednim prikaznim tekstom i dugmetom izgleda ovako:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Glavni meni aplikacije se mora prikazivati na ranije određeni taster na daljinskom uređaju. Isti mora biti vidljiv sa nekoliko metara od ekrana [5]. U glavnom meniju se biraju datoteke sa DLNA poslužioca ili lokala i u njemu je integriran pretraživač datoteka (eng. File explorer). U procesu pretraživanja datoteka je moguće dodati multimedijalnu datoteku u listu za reprodukciju kao i sama reprodukcija izabrane datoteke. Pri reprodukciji video datoteka potrebno je proveriti da li se na istoj putanji nalazi i bar jedna .srt datoteka, ako se nalazi potrebno je učitati tu datoteku u aplikaciju i prikazati je kao prevod tokom reprodukcije video datoteke. U slučaju da je pronađeno više .srt datoteka potrebno je prikazati korisniku jedan modalni dijalog u kome će da izabere odgovarajuću datoteku.

Prikazivanje liste za reprodukciju se vrši pritiskom na ranije određen taster na daljinskom uređaju kao i glavni meni. U listi se mogu naći video i muzičke datoteke sa DLNA poslužioca kao i sa lokala. Prilikom reprodukcije multimedije sa DLNA poslužioca iz liste za reprodukciju potrebno je proveriti trenutnu dostupnost željenog DLNA poslužioca i ako je isti nedostupan, selektovanu multimedijalni datoteku je potrebno izbaciti iz liste za reprodukciju.

3.2 Reprodukcija multimedijalnog sadržaja

Klasa MediaPlayer.java se u Android operativnom sistemu koristi za kontrolu reprodukcije multimedijalnog sadržaja [1]. Objekat ove klase može da pronađe, dekoduje i reprodukuje audio i video sadržaj uz minimalnu pripremu za reprodukciju i podržava nekoliko različitih izvora multimedijalnog sadržaja kao što su:

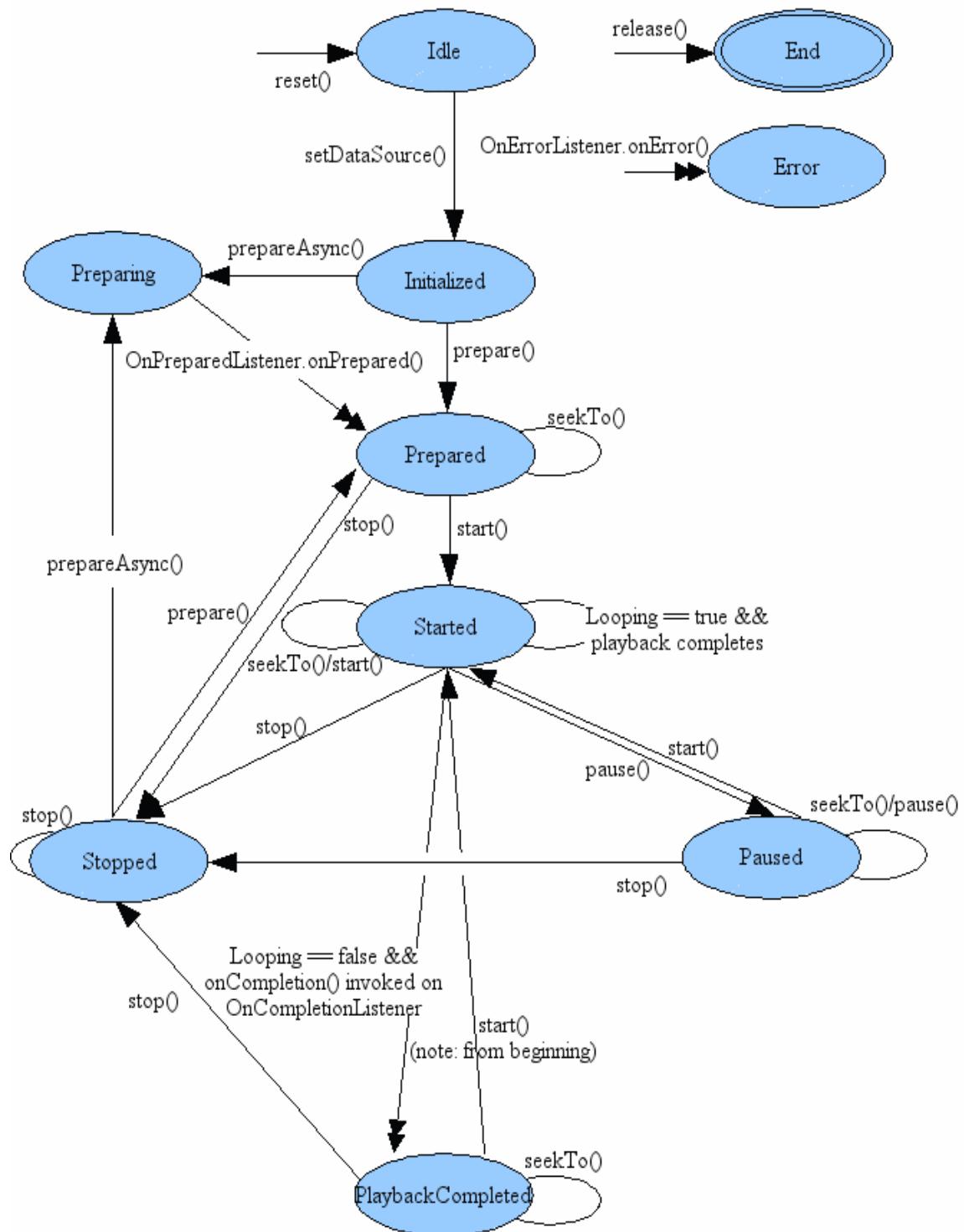
- Lokalni resursi (resursi unutar aplikacije),
- Unutrašnji resursi (oni resursi koje dobavljamo pomoću dobavljača sadržaja),
- Spoljašnji URL.

Kontrola reprodukcije multimedijalnog sadržaja je napravljena kao automat. Dijagram koji sledi prikazuje životni ciklus i stanja objekta klase MediaPlayer.java (slika 3.2), gde plave elipse predstavljaju stanja u kojima on može biti, a linije između njih predstavljaju kontrole pri reprodukciji. Postoje dva tipa linija:

- Sa jednom strelicom, koje predstavljaju pozive sinhronih metoda,
- Sa dve strelice, koje predstavljaju pozive asinhronih metoda.

Sa dijagramom stanja (slika 3.2) se vidi da objekat klase MediaPlayer.java može da se nađe u sledećim stanjima:

- Neaktivan (eng. Idle),
- Greška (eng. Error),
- Inicijalizovan (eng. Initialized),
- Spreman (eng. Prepared),
- Pokrenut (eng. Started),
- Pauziran (eng. Paused),
- Stopiran (eng. Stopped),
- Reprodukcija završena (eng. Playback Completed).



Slika 3.2 Dijagram stanja objekta klase MediaPlayer.java

Za reprodukciju video sadržaja pored klase MediaPlayer.java potrebna je i klasa SurfaceView.java koja služi za prikaz video sadržaja [3]. SurfaceView.java kreira rupu u svom okviru tako da omogućava da se njegova površina prikaže. Pristup površini za prikazivanje je obezbeđen kroz SurfaceHolder.java spregu, koja se može preuzeti od objekta klase SurfaceView.java pozivom metode `getHolder()`.

Primer povezivanja ove dve klase:

```

SurfaceHolder mPreview = new SurfaceHolder();
MediaPlayer mMediaPlayer = new MediaPlayer();
try {
    mMediaPlayer.setDataSource( currentPath );
    mMediaPlayer.setDisplay(mPreview.getHolder());
    mMediaPlayer.prepare();
} catch (Exception e) {}

```

U ovom radu je neophodno koristiti prothodno spomenute klase Android operativnog sistema za kreiranje korisničkog interfejsa i reprodukciju multimedijalnih datoteka [2].

Za pauziranje, prebacivanje na sledećeg ili prethodnog u listi za reprodukciju, pretraživanje po već reprodukovanoj multimedijalnoj datoteci, potrebno je napraviti i kontrole za MediaPlayer klasu koje će se pojavljivati na neki ranije određen taster, ali će imati i prečice za neke elementarne komande kao što su: pauziraj, nastavi reprodukciju, pusti sledećeg, pusti prethodnog, povećaj jačinu zvuka, smanji jačinu zvuka.

Za pristup i pretraživanje datoteka sa DLNA poslužioca kao i za neke osnovne komande kao što su: vraćanje male slike koja predstavlja željenu datoteku, putanja do željene datoteke (u obliku http veze), dužina trajanja multimedijalne datoteke, veličinu koja zauzima datoteka, se koristi gotova biblioteka.

3.3 Opis ciljne platforme

Platforma raspolaže sa satelitskim biračem kanala tipa Serit SP2230 [7] koji se konfiguriše putem I2C (eng. *Inter Integrated Circuit*) magistrale i koji u slučaju zaključavanja na frekvenciju propušta prenosni tok podataka do CIMax čipa proizvođača SmarDTV.

CIMax čip se koristi za usmeravanje prenosnog toka od birača kanala do dekodera. Ukoliko se servisi slobodno emituju, prenosni tok se od CIMax čipa usmerava direktno do dekodera.

Ako su servisi zaštićeni (eng. *Scrambled*), prenosni tok se usmerava preko modula uslovnog pristupa (eng. *Conditional access module*) do dekodera.

Modul uslovnog pristupa dešifruje (eng. *Descrable*) zaštićeni prenosni tok podataka i korisniku omogućava praćenje sadržaja koji se prikazuje. Da bi dešifrovanje bilo moguće neophodno je da korisnik ima korisničku karticu koja se stavlja u modul uslovnog pristupa. Kartica se dobija plaćanjem usluge operateru za praćenje sadržaja na zaštićenom programu.

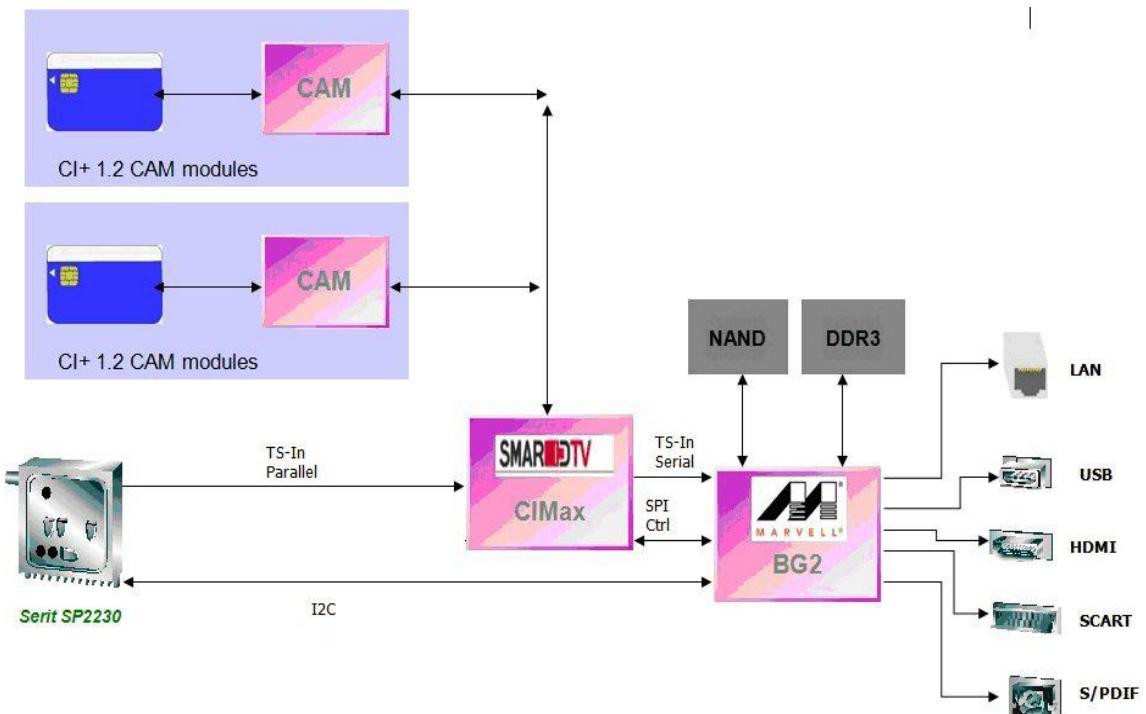
Svako od ovih usmeravanja, izvršava se na nivou fizičke arhitekture.

Na slici 3.3 prikazani su osnovni delovi ciljne platforme.

Ključne komponente:

- Satelitski birač kanala tipa Serit SP2230 – koristi se za preuzimanje signala sa satelita,

- CIMax čip proizvođača SmarDTV - koristi se za usmeravanje prenosnog toka
- Dva modula uslovnog pristupa – koriste se za dešifrovanje zaštićenog prenosnog toka
- Procesor ARM familije, Marvell BG2 88DE3100 [8]
- DDR3 operativna memorija Micron MT41J2G4THA – memorija koja se stavlja na raspolaganje sistemu
- NAND memorija Micron MT29F16G08 – memorija koja se koristi za smeštanje aplikacije na ciljnoj platformi



Slika 3.3 Marvell BG2 SOC platforma

4. Programsко rešenje

Programsko rešenje se svodi na pravljenje korisničke sprege i mehanizma za reprodukciju multimedijalnog sadržaja, kao i njihova kombinacija pri pravljenju kontrole za objekat klase MediaPlayer.java [2].

4.1 Pravljenje glavnog menija

Glavni meni (slika 4.1) se kreira preko video sloja i prikazuje se na predefinisan taster na daljinskom uređaju. Glavni meni se nalazi u dijalogu (dijalog je sastavni deo Androida i kreira se pomoću klase Dialog.java) koji je horizontalno i vertikalno centriran u odnosu na ecran [3]. Dijalog zauzima celu širinu i 5/6 visine ekrana radi veće preglednosti glavnog menija, i zbog skaliranja na različitim veličinama ekrana. On je takođe i potpuno transparentan [5] da bi se video sadržaj video kroz njega, dok je njegov sadržaj tj. objekti klase View.java imaju podešenu transparenciju u zavisnosti od njihovog položaja i to u vrednostima od 40%, 20% i 0% (potpuno vidljivi) [2].

Glavni meni je u zavisnosti od izbora opcija podeljen po vertikalnim nivoima. U prvom nivou se nalaze:

- Mediateka,
- Sorting.

Prilikom izbora prvog taba (Mediateke) prikazuje se bela prozirna površina koja zauzima ostatak dijaloga glavnog menija i u kojoj se prikazuju ostali nivoi mediateke. U drugom nivou mediateke se nalaze horizontalno poređani DLNA poslužioci od kojih prvi predstavlja lokalni sistem datoteka. Za slučaj da DLNA poslužioca ime previše da bi svi stali na ekran svi dugmići koji predstavljaju DLNA poslužioce se dodaju u objekat tipa ScrollView.java u kojem je omogućeno horizontalno pomeranje tako da se svakom dugmetu (DLNA poslužiocu) može pristupiti. Detalji realizacije se vide u kodu koji sledi, gde je

„dlna_computers_first_level_main“ xml datoteka u kojoj se nalazi ScrollView,a „dlna_computers_first_level_item“ xml datoteka koja sadrži objekat dugmeta (Button.java) sa već definisanim izgledom, a „dlnaServers“ je lista svih nađenih DLNA poslužioca:

```
public void fillDialogFirstDLNALevel(int numberOfFindDLNAComputers,
                                     LinearLayout layoutForInflate) {

    layoutFirstLevel = (LinearLayout) inflater.inflate(
        R.layout.dlna_computers_first_level_main, null);

    LinearLayout dlnaCompLayout = (LinearLayout) layoutFirstLevel
        .findViewById(R.id.linearlayoutDLNAComputersMain);

    for (int i = 0; i < numberOfFindDLNAComputers; i++) {
        LinearLayout layToInflate = (LinearLayout) inflater.inflate(
            R.layout.dlna_computers_first_level_item, null);
        Button button = (Button) layToInflate
            .findViewById(R.id.buttonDLNAFirstLevel);
        button.setText(dlnaServers.get(i));

        button.setEllipsize(TruncateAt.MARQUEE);
        button.setSingleLine(true);

        button.setBackgroundResource(com.duo_tv.R.drawable.dlna_comp_def_back);

        button.setOnFocusChangeListener(new DuoTVOnFocusChangeListener(
            activity, this, 1, 0));

        button.setOnKeyListener(new DuoTVKeyListener(activity, this, 1,
            i,
            dlnaServers.get(i)));

        buttonsFirstLevel.add(button);

        dlnaCompLayout.addView(layToInflate);
    }

    layoutForInflate.addView(layoutFirstLevel,
    DLNA_COMPUTERS_FIRST_LEVEL);
    layoutForInflate.refreshDrawableState();
}
```

Po izboru DLNA poslužioca u glavni meni se prikazuje sledeći nivo u kome se nalaze tri taba, i to:

- Muzika (eng. Music),

- Video (eng. Videos),
- Sve datoteke (eng. All files).

Trenutni nivo se koristi samo radi filtriranja prikaza u pretraživaču podataka koji se nalazi u sledećem nivou. Pri izboru jednog od ova tri taba otvara se sledeći nivo.

U četvrtom nivou se u sredini nalazi pretraživač podataka koji je organizovan kao rešetka visine maksimalno dva elementa a širine maksimalno šest elemenata, koji se popunjavaju sa leve strane ka desnoj i od gore ka dole. U slučaju da postoje više od dvanaest datoteka i fascikli pretraživač podataka se realizuje u nekoliko strana koje se menjaju pomoću strelica koje se nalaze sa leve i desne strane od pretraživača. Različiti tipovi datoteka ili fascikli se predstavljaju na različite načine u pretraživaču da bi se napravila razlika između video i muzičkih datoteka ili fascikli.

U zavisnosti od tipa datoteke u pretraživaču različito se reaguje na klik. Kada se izabere fascikla, u pretraživaču se prikazuje sadržaj izabrane fascikle u abecednom redu, Dok se trenutna putanja prikazuje iznad pretraživača.

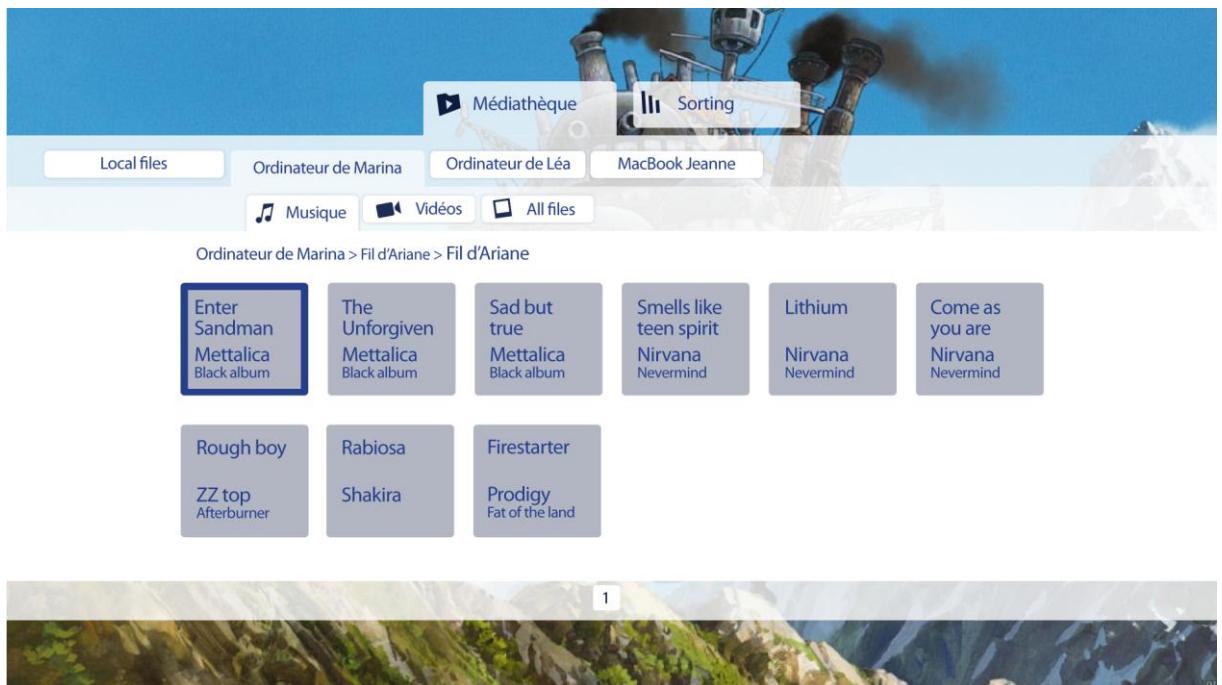
Pri izboru multimedijalnih datoteka razlikujemo kratak i dugi klik. Kada korisnik zadrži taster na daljinskom uređaju (dugi klik), a u fokusu je multimedijalna datoteka pojavljuje se dijalog sa opcijama koje korisnik može da izabere i koje su vezane za izabranu datoteku kao što su:

- Reprodukuj (eng Play),
- Dodaj u listu za reprodukciju (eng. Add to playlist),
- Izbriši izabranu datoteku (eng. Delete).

Kada korisnik izvrši kratki klik na daljinskom uređaju, a u fokusu je audio datoteka, označena datoteka se reproducuje tako što se kreira novi objekat klase MediaPlayer.java i u njegovu putanju se postavi putanja do željene datoteke. U slučaju da audio datoteka sadrži i malu sliku ona se postavlja u pozadinu ekrana na nivou ispod glavnog menija.

Pri izboru video datoteke, ona se takođe reproducuje kreiranjem novog objekta klase MediaPlayer.java sa razlikom da se taj objekat povezuje sa objektom klase SurfaceView.java koji se nalazi u pozadini aplikacije [3], i video reprodukcija se izvršava na površini na kojoj se nalazi SurfaceView. Ako se reprodukcija izvršava sa lokala automatski se proverava da li u istoj fascikli u kojoj se nalazi reproducovana datoteka postoji .srt datoteka radi prikaza prevoda. U slučaju da se pronađe više od jedne datoteke prikazuje se dijalog u kome su predstavljeni nazivi dataoteka i korisnik izabere odgovarajuću datoteku koja se parsira.

U petom nivou se nalazi prikazana trenutno aktivna strana pretraživača kao i broj ukupnih strana.



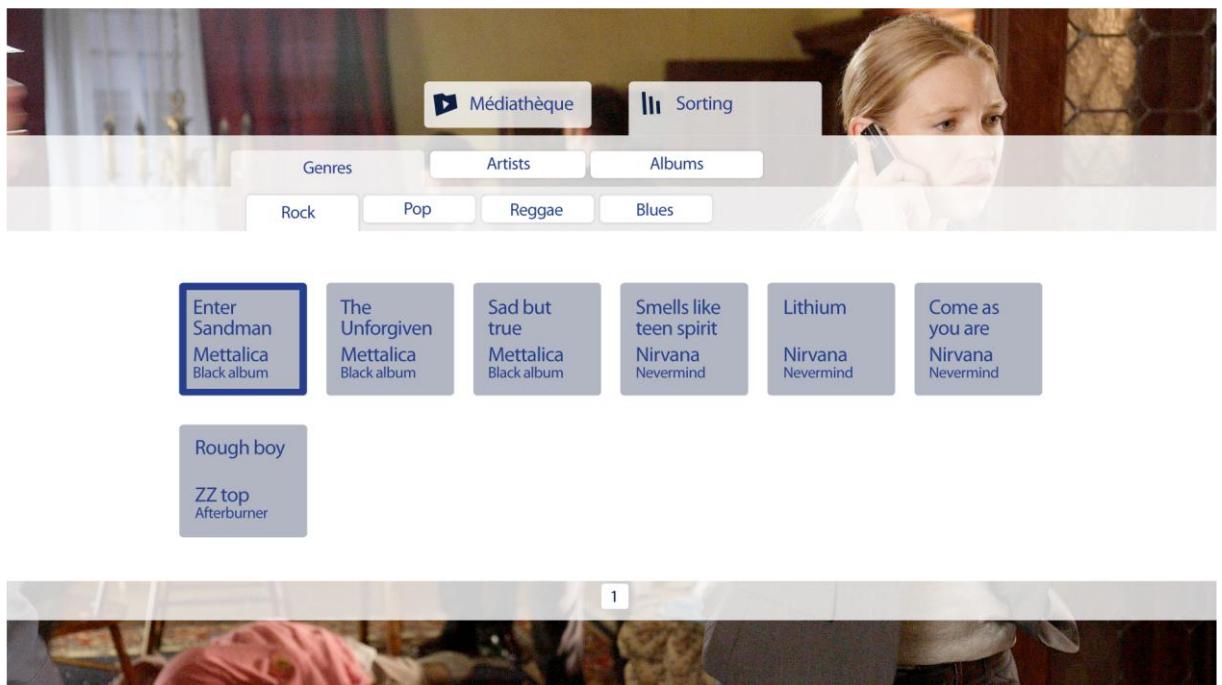
Slika 4.1 Prikaz glavnog menija sa izabranom medijatekom

Prilikom izbora drugog taba (Sortiranja) prikazuje se bela prozirna površina koja zauzima ostatak dijaloga glavnog menija i u kojoj se prikazuju ostali nivoi taba sortiranje (slika). U drugom nivou taba sortiranje se nalaze tabovi koji predstavljaju izbor vrste sortiranja. Lokalne datoteke se mogu sortirati po sledećim kriterijumima:

- Žanrovima,
- Umetnicima,
- Albumima.

U trećem nivou se nalaze imena razvrstanih lista kao na primer: rok, pop, bluz itd.

Sortiranje audio datoteka se svodi na čitanje ID3 taga koji se nalazi u .mp3 datoteci pomoću gotove biblioteke. Pretraga se realizuje za sve audio datoteke, čita se njihov ID3 tag i dodaju se u odgovarajuće liste. Datoteke koje ne sadrže ID3 tag ili se njegovim čitanjem nije ništa saznalo se ubacuju u listu „Ostali“ (eng. Other) za sve kriterijume.



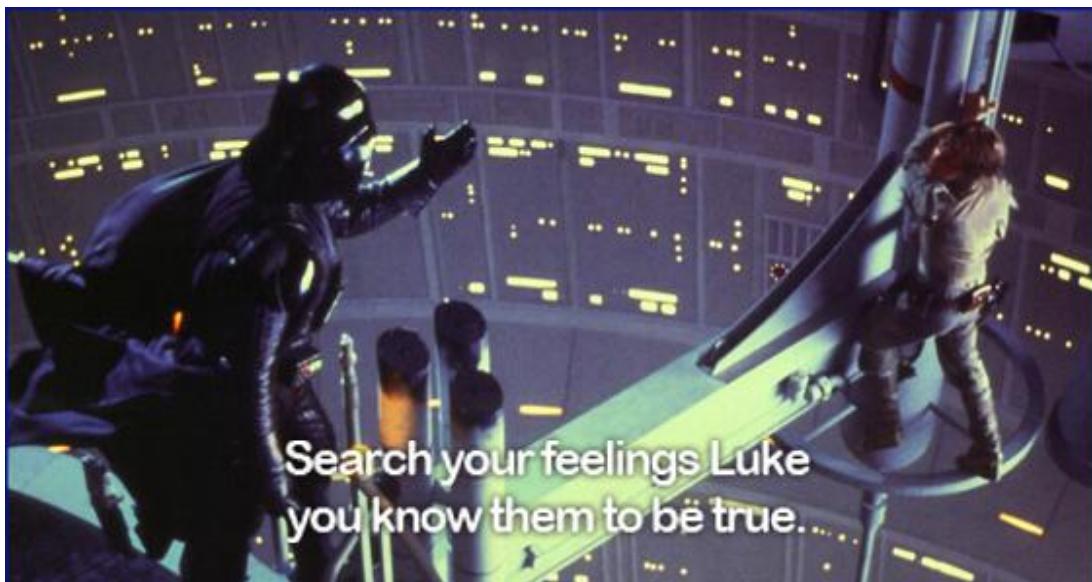
Slika 4.2 Prikaz glavnog menija sa izabranim sortiranjem

4.2 Realizacija prevoda

Parsiranje .srt datoteke se postiže njenim učitavanjem u objekat klase BufferedReader.java i njegovim čitanjem jedne linije datoteke po redu. Po čitanju linije, njen sadržaj od važnosti (tekst koji se prikazuje na ekranu, redni broj linije, početno vreme prikaza teksta i završno vreme prikaza teksta) se čuva u listu objekata ArrayList<LyricsSubtitle>. Klasa LyricsSubtitle sadrži u sebi nekoliko polja od važnosti koja su potrebna za pravilan prikaz na ekranu:

- Id ili redni broj linije prevoda,
- Početno vreme (eng. start time),
- Završno vreme (eng. end time),
- Prva linija teksta,
- Druga linija teksta.

Prikaz je realizovan upotrebom dva objekta klase TextView.java koji se nalaze na nivou iznad nivoa za reprodukciju multimedijalnih datoteka. Raspoređeni su jedan ispod drugog zbog mogućnosti prevoda da bude u jednom i u dva reda.



Slika 4.3 Prikaz korišćenja prevoda

4.3 Realizacija liste za reprodukciju

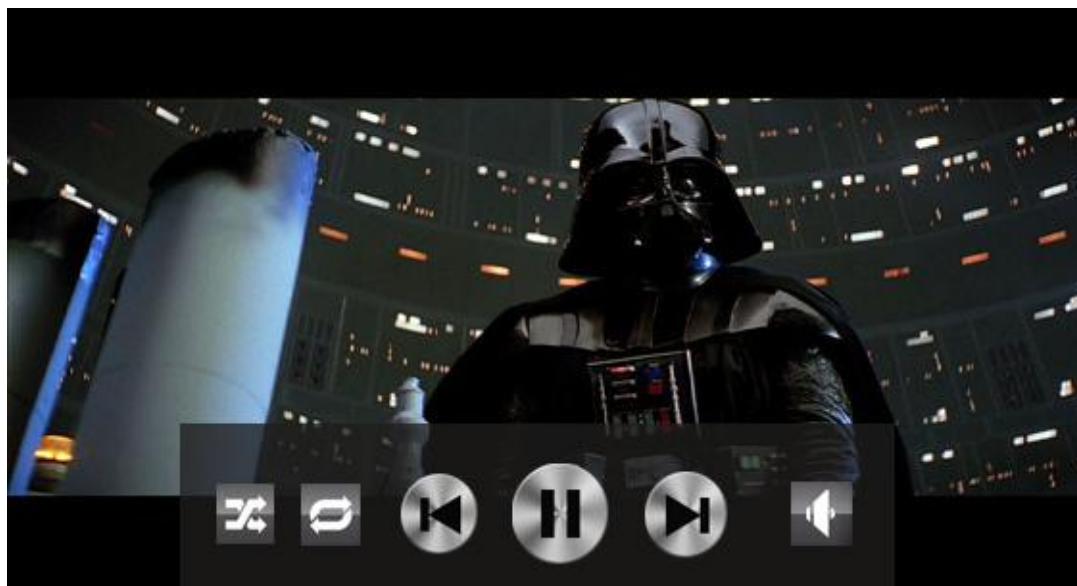
Lista za reprodukciju se prikazuje na predefinisan taster na daljinskom uređaju. Implementirana je koristeći Androidovu podrazumevanu klase ListView.java. Da bi ListView prikazala na ekranu željenu listu nasleđena je Androidova klasa BaseAdapter.java I prerađene su njene metode tako da prikazuje objekte tipa Media.java koji u sebi sadrže putanju do multimedijalne datoteke, njeni ime, tip i ako sadrži malu sliku putanju do slike ili u slučaju muzike upotrebljava se slika iz ID3 taga. Prilikom dugog klika na elemenat liste za reprodukciju otvara se dijalog u kome su opcije koje korisnik može da izabere kao što su:

- Reprodukuj,
- Izbriši iz liste.

U dijaluču liste za reprodukciju se nalazi i dugme za brisanje svih elemenata iz liste. Lista za reprodukciju je povezana sa dugmadima kontrole za reprodukciju „Ponavljam trenutno reprodukovani datoteku“ i „Reprodukuj nasumičnu datoteku iz liste“ MediaPlayer klase. Po završetku reprodukcije trenutne datoteke u zavisnosti od prethodno navedenih dugmadi bira se sledeća datoteka (objekat u listi za reprodukciju) koja će da se reprodukuje. Ako je „Ponavljanje“ uključeno ponavljaće se trenutno reprodukovana datoteka, ako nije proverava se stanje dugmeta „Reprodukuj nasumičnog“. U slučaju kada je ono u aktivnom stanju, na slučajan način se bira sledeća datoteka iz liste za reprodukciju, a ako je ono u neaktivnom stanju reprodukuje se sledeći iz liste ili se reprodukuje prvi iz liste ako nije reprodukovana datoteka iz liste.

4.4 Realizacija kontroli za reprodukciju

Kontrole za MediaPlayer se nalaze u dnu ekrana i prikazuju se na pritisak bilo kog tastera koji već nije rezervisan za prikazivanje menija i liste za reprodukciju i nalazi se iznad sloja za reprodukciju multimedijalnog sadržaja (slika 4.4). Sadrži dugmiće za pauziranje, nastavak reprodukcije (fizički isto dugme), za reprodukciju sledećeg u listi, prethodnog u listi, za ponavljanje trenutno reprodukovane datoteke, za reprodukovanje nasumičnog iz liste za reprodukciju itd., takođe sadrži i traku koja predstavlja napredovanje reprodukcije (eng. Progress Bar). Dijalog za kontrole se automatski sakriva posle pet sekundi.



Slika 4.4 Prikaz kontrola za reprodukciju

5. Ispitivanje i verifikacija

Ispitivanje i pregled rezultata ispitivanja se može podeliti na nekoliko celina. Posebno su napisani ispitni slučajevi za svaku od ovih celina zbog različitostima u ponašanju i drugačijim ispitnim slučajevima koji se koriste za različite celine u aplikaciji. Te celine možemo podeliti na:

- Ispitivanje glavnog menija,
- Ispitivanje liste za reprodukciju i reprodukciju različitih tipova datoteka,
- Ispitivanje usklađenosti prevoda,
- Ispitivanje pravilnog ponašanja kontrola za objekat klase MediaPlayer.java.

5.1 Ispitivanje i rezultati pri ispitivanju glavnog menija

Ispitivanje glavnog menija se svodi na pravilno ponašanje aplikacije pri prelaženju kroz razne nivoje glavnog menija i razne tabove. Ispitivanje se svodi na praćenje ponašanja objekata tipa Layout i View pri veštačkom pravljenju komandi. Rezultati se čuvaju u prostu tekstualnu datoteku gde se čuva svaka promena posmatranih objekata.

Provera se započinje tako što se prikaže dijalog glavnog menija pozivanjem funkcije showHideMainMenu(). Podrazumeva se da su tada vidljiva samo dva taba Medijateka i Sortiranje. Posle toga se šalju četiri komande DPAD_DOWN da bi se otvorio pretraživač podataka. Tada se podrazumeva da su svi objekti u stanju VISIBLE. Posle toga se šalje serija komandi koje se slučajno generišu: DPAD_DOWN, DPAD_UP, DPAD_RIGHT, DPAD_LEFT. Tada se takođe podrazumeva da su svi objekti u stanju VISIBLE. Zatim se šalje komanda BACK da bi se prešlo na treći nivo i sakrio pretraživač podataka, podrazumeva se da je stanje pretraživača u ovom trenutku INVISIBLE. Zatim se šalju dve komande DPAD_UP, proverava se vidljivost svih elemenata glavnog menija, samo dva glavna taba bi trebalo da su u stanju VISIBLE. Na sličan način se proverava i ponašanje taba „Sorting“.

Ispitivanje se ponavlja deset puta i njegovi rezultati se upisuju u tekstualnu datoteku. Za svako posebno ispitivanje se upoređuju rezultati i daje konačna ocena, desila se greška (ERROR) ili se nije desila greška (NO_ERROR).

5.2 Ispitivanje liste za reprodukciju i reprodukcije različitih tipova datoteka

5.2.1 Ispitivanje liste za reprodukciju

Ispitivanje liste za reprodukciju se svodi na ispitivanje komandi prečica jer su one analogne pritisku na odgovarajuće dugmiće na kontrolama za reprodukciju:

- reprodukuj prethodnog (pritisak na taster DPAD_LEFT pri svim zatvorenim dijalozima),
- reprodukuj sledećeg (pritisak na taster DPAD_RIGHT pri svim zatvorenim dijalozima).

Razlikujemo nekoliko kritičnih slučajeva: pritisak na bilo koji od ova dva tastera kada je lista prazna. U tom slučaju ne treba ništa da se desi i u datoteku se upisuje NO_ERROR, u suprotnom ERROR.

U slučaju kada lista nije prazna i reprodukuje se prvi u listi, na pritisak na taster DPAD_LEFT aplikacija ne reaguje. Rezultat se upisuje u datoteku.

Za slučaj kada se reprodukuje poslednji u listi, na pritisak na taster DPAD_RIGHT se ne reaguje. Rezultat se takođe upisuje u datoteku.

Za slučajeve između ova dva granična slučaja aplikacija pokušava da reprodukuje prethodnog ili sledećeg u zavisnosti od komande.

Takođe se ispituje i dugme „Obriši sve iz liste za reprodukciju“ tako što se pozove metoda `clearAll()` iz modula Playlist.java, dok je lista prazna i jednom kada lista nije prazna.

Ispitivanje se ponavlja za svaki slučaj po pet puta i rezultati se upisuju u datoteku.

Lista za reprodukciju se mora ručno puniti sa određenim brojem multimedijalnih datoteka.

5.2.2 Ispitivanje reprodukcije različitih tipova datoteka

Ovo ispitivanje se sadrži u ispitivanju liste za reprodukciju jer metoda za reprodukciju `setPathAndPlay()` za povratnu vrednost ima objekat tipa boolean koji označava da li je objekat klase MediaPlayer.java uspeo da reprodukuje datoteku.

Rezultat ovog ispitanja se upisuje u datoteku u sledećem formatu: putanja (ili URI) datoteke koja se reproducuje kao i uspešnost reprodukcije date datoteke.

5.3 Ispitivanje usklađenosti prevoda sa reprodukovanim video datotekom

Da bi ispitanje ovog segmenta aplikacije bio uspešan potrebno je pronaći reprezentativne .srt datoteke koje će se koristiti prilikom ispitanja. Odredite vremena u kojima će se upoređivati tekst prevoda na ekranu sa onim što bi tu trebalo da se nađe. Uzimaju se sledeća vremena za upoređivanje za svaku datoteku kojoj se pronađe datoteka za prevod: 30 sekundi, 60 sekundi, 120 sekundi, 240 sekundi. Ručno se u ispitanje unesu željeni tekstovi za izabrana vremena.

Proces upoređivanja je elementaran, u datim vremenima se upoređuje sadržaj prevoda komandom `textViewSubtitle.getText()` sa odgovarajućim tekstrom iz ispitanja. Ako se tekstovi poklapaju u datoteku se upisuje vreme provere i rezultat ispitanja (ERROR ili NO_ERROR).

5.4 Ispitivanje pravilnog ponašanja kontrola reprodukcije

Ispitanje komandi za reprodukciju prethodnog i sledećeg je već izvršeno u ispitanju liste za reprodukciju.

Ispitanje komandi pauziraj reprodukciju i nastavi reprodukciju se obavlja jednostavnim pozivom funkcije `isPlaying()` nad objektom klase `MediaPlayer.java`. Ispitanja se ponavljaju po deset puta i njihovi rezultati se upisuju u datoteku.

Komande „ponavljam trenutno reprodukovani“ i „reprodukuju slučajnog iz liste“ se ispituju proverom indeksa trenutno reprodukovane datoteke u listi za reprodukciju. Ovo ispitivanje proverava indeks trenutno reprodukovanih pri promeni datoteke. Ako je ponavljanje uključeno index ne treba da se menja i rezultat ispitanja je u tom slučaju NO_ERROR. Ako ponavljanje nije uključeno, tada se prati stanje reprodukuju slučajnog, ako je ono isključeno sledeći indeks mora biti uvek za jedan veći od prethodnog osim kada je poslednji u listi. Ako je sve u redu u datoteku se upisuje NO_ERROR u suprotnom ERROR.

6. Zaključak

U ovom radu je realizovana aplikacija za reprodukciju multimedijalnih sadržaja na Android baziranom DTV prijemniku.

Realizovano rešenje je univerzalno i može se koristiti za sve verzije Android operativnog sistema. Pretraga DLNA poslužioca je namenski realizovana za Marvell BG2 STB uređaj.

Rešenje je ispitivano na ciljnoj fizičkoj arhitekturi gde je ispitano ponašanje glavnog menija, liste za reprodukciju, kontrola za reprodukciju i usklađenost prevoda. Korisnička spregu je se pokazala kao intuitivna jer se sva navigacija realizuje pomoću četiri smerna tastera (eng. D-pads), tastera za ulaz i tastera za izlaz.

U postojeću aplikaciju je moguće dodati opciju za pregled slika sa lokala i sa DLNA poslužioca. Takođe je moguće dodati opciju da se reprodukcija ne završi po izlasku iz aplikacije uz minimalne izmene u kodu. Takođe, moguće je dodati učitavanje sadržaja sa DLNA poslužioca koje ne blokira aplikaciju nego učitava sadržaj i ažurira korisničku spregu asinhrono, radi prirodnijeg ponašanja aplikacije.

Moguće je poboljšati trenutnu korisničku spregu sa trodimenzionim efektima i dugmićima, kao i redosledom tabova.

7. Literatura

- [1] Sajt Android podrške za razvoj, <http://www.developer.android.com>, učitano 18.06.2012
- [2] John Hurley, *Android GUI Project*, CS 454
- [3] Predavanja iz predmeta Projektovanje namenskih računarskih struktura 1, <http://www.rt-rk.uns.ac.rs/studijski-program-2009/vi-2009/pnrs1/737-pnrs1-predavanja>
- [4] Wikipedia, the free encyclopedia, www.wikipedia.com, učitano 18.06.2012
- [5] M.Vidakovic, N.Teslic, T.Maruna, and V.Mihic: *Android4TV: a proposition for integration of DTV in Android devices*, IEEE 30th International Conference on Consumer Electronics (ICCE), Las Vegas, January 2012, pp. 441-442
- [6] Vladimir Kovačević, Miroslav Popović: *Sistemska programska podrška u realnom vremenu*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2002
- [7] Serit, *Digital Satelite Tuner SP2230MVb Specification*
- [8] Marvell, *88DE3010 Software API User Manual*