



**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: Младен Илић

Број индекса: 12360

Тема рада: Развој ДЛНА клијената за таблет/телефоне на бази Андроид оперативног система: развој ДЛНА ДМП клијената

Ментор рада: Проф. др Јелена Ковачевић

Нови Сад, јул, 2012



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОМЕНТАЦИЈСКА 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:	Младен Илић
Ментор, МН:	Проф.др Јелена Ковачевић
Наслов рада, НР:	Развој ДЛНА клијената за таблет/телефоне на бази Андроид ОС: развој ДЛНА ДМП клијената
Језик публикације, ЈП:	Српски / латиница
Језик извода, ЈИ:	Српски
Земља публикавања, ЗП:	Република Србија
Уже географско подручје, УГП:	Војводина
Година, ГО:	2012
Издавач, ИЗ:	Ауторски репринт
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, ФО: <small>(поглавља/страна/ цитата/табела/слика/графика/прилога)</small>	
Научна област, НО:	Електротехника и рачунарство
Научна дисциплина, НД:	Рачунарска техника
Предметна одредница/Кључне речи, ПО:	
УДК	
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, ВН:	
Извод, ИЗ:	Опис реализације апликације за репродукцију мултимедијалног садржаја на мобилним уређајима, телефонима и таблетима, заснованим на Андроид оперативном систему, која подржава репродуковање локалног садржаја као и садржаја са ДЛНА послужиоца.
Датум прихватања теме, ДП:	
Датум одбране, ДО:	
Чланови комисије, КО:	Председник: Др Мирослав Поповић
	Члан: Др Илија Башичевић
	Члан, ментор: Др Јелена Ковачевић
	Потпис ментора



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

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 :	Mladen Ilic
Mentor, MN :	PhD Jelena Kovacevic
Title, TI :	Development DLNA clients for tablet/phones based on Android OS: development of DLNA DMP clients
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 :	
Publisher, PB :	Author's reprint
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	
Scientific field, SF :	Electrical Engineering
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, S/KW :	
UC	
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :	
Abstract, AB :	Realization of digital media player application for mobile devices, phones and tablets, based on Android operating system, which supports playback of local media files and files from DLNA server.
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	
Defended Board, DB :	President: PhD Miroslav Popovic
	Member: PhD Ilija Basicovic
	Member, Mentor: PhD Jelena Kovacevic
	Mentor's sign

Zahvalnost

Posebno se zahvaljujem stručnim saradnicima Tomislavu Maruni i Nikoli Crvenkoviću na stručnoj pomoći tokom izrade završnog (*bachelor*) rada.

Zahvaljujem se svim kolegama koji su mi na bilo koji način pomogli u izradi ovog rada.

Na kraju se zahvaljujem porodici i prijateljima koji su mi bili velika podrška tokom studiranja.

SADRŽAJ

1. Uvod	1
2. Teorijske osnove	2
2.1 Android operativni sistem	2
2.2 Bionic libc	5
2.3 UPnP i DLNA.....	5
3. Koncept rešenja	8
4. Programsko rešenje	9
4.1 Građenje izvornog koda	9
4.2 Aplikativni sloj	11
4.2.1 Klasa FsLibActivity	11
4.2.2 Klasa BmpActivity	12
4.2.3 Klasa VideoActivity	13
5. Rezultati	15
6. Zaključak	19
7. Literatura.....	20

SPISAK SLIKA

Slika 1 – Arhitektura Android operativnog sistema	3
Slika 2 – Hronologija Android operativnog sistema	4
Slika 3 – Udeo različitih verzija Android operativnog sistema, Jun 2012	4
Slika 4 – Umreženi UPnP uređaji	6
Slika 5 – Kako DLNA funkcioniše	6
Slika 6 – Slojevi programske podrške	8
Slika 7 – Stablo izvornog C koda.....	10
Slika 8 – Prikaz klase FsLibActivity	11
Slika 9 – Prikaz klase BmpActivity	13
Slika 10 – Prikaz klase VideoActivity.....	14
Slika 11 – Izgled aplikacije pri pokretanju programa	15
Slika 12 – Izgled aplikacije pri reprodukciji slike	16
Slika 13 - Izgled aplikacije pri reprodukciji audio sadržaja	17
Slika 14 - Izgled aplikacije pri reprodukciji video sadržaja	17

SPISAK TABELA

Tabela 1 – Prikaz klasa projektnog zadatka.....	11
---	----

SKRAĆENICE

DLNA - *Digital Living Network Alliance*, Protokol za deljenje digitalnog sadržaja između multimedijalnih uređaja

DMP - *Digital Media Player*, Modul za reprodukciju u *DLNA* mreži

DMS - *Digital Media Server*, Poslužilac u *DLNA* mreži

DMC - *Digital Media Controller*, Modul za kontrolu u *DLNA* mreži

UPnP - *Universal Plug and Play*, Skup mrežnih protokola koji dozvoljavaju multimedijalnim uređajima da ostvare međusobne veze i uspostave razne mrežne usluge

JNI - *Java Native Interface*, Programska sprega u okviru *Android* operativnog sistema

URL - *Uniform Resource Locator*, Jedinstvena adresa resursa

IP - *Internet Protocol*, Internet protokol

TCP - *Transmission Control Protocol*, Protokol kontrole toka

HTTP - *Hypertext Transfer Protocol*, Aplikativni protokol za komunikaciju

XML - *Extensive Markup Language*, proširivi metajezik za označavanje tekstualnih dokumenata

1. Uvod

U ovom radu je predstavljen sistem za reprodukciju multimedijalnog (slika, audio, video) sadržaja, prilagođenog mobilnim uređajima, tabletima i mobilnim telefonima, zasnovanim na Android operativnom sistemu. Multimedijalni sadržaj se nalazi u lokalnoj mreži na DLNA poslužiocima kojima korisnik pristupa.

Prvi deo rada se odnosi na građenje izvornog (native) koda biblioteka libupnp, libdlna i libmedia za Android operativni sistem za mobilne telefone počevši od verzije 2.3 (Gingerbread) i novije. Za testiranje i verifikaciju potrebno je napraviti jednostavnu aplikaciju u Java programskom jeziku koja reprodukuje multimedijalni sadržaj sa DLNA poslužioca.

Ovaj rad je sačinjen od sedam poglavlja.

U drugom poglavlju date su teorijske osnove Android operativnog sistema, Android systemske biblioteke Bionic libc kao i kratak opis DLNA i UPnP protokola.

U trećem poglavlju se nalazi koncept rešenja sistema za reprodukciju multimedijalnog sadržaja.

Četvrto poglavlje daje opis programskog rešenja koje se sastoji od građenja dobijenog izvornog C koda za Android operativni sistem i realizaciju aplikacije u Java programskom jeziku.

U petom poglavlju dat je opis rezultata programskog rešenja.

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

U sedmom poglavlju dat je spisak literature koja je korišćena za izradu ovog rada.

2. Teorijske osnove

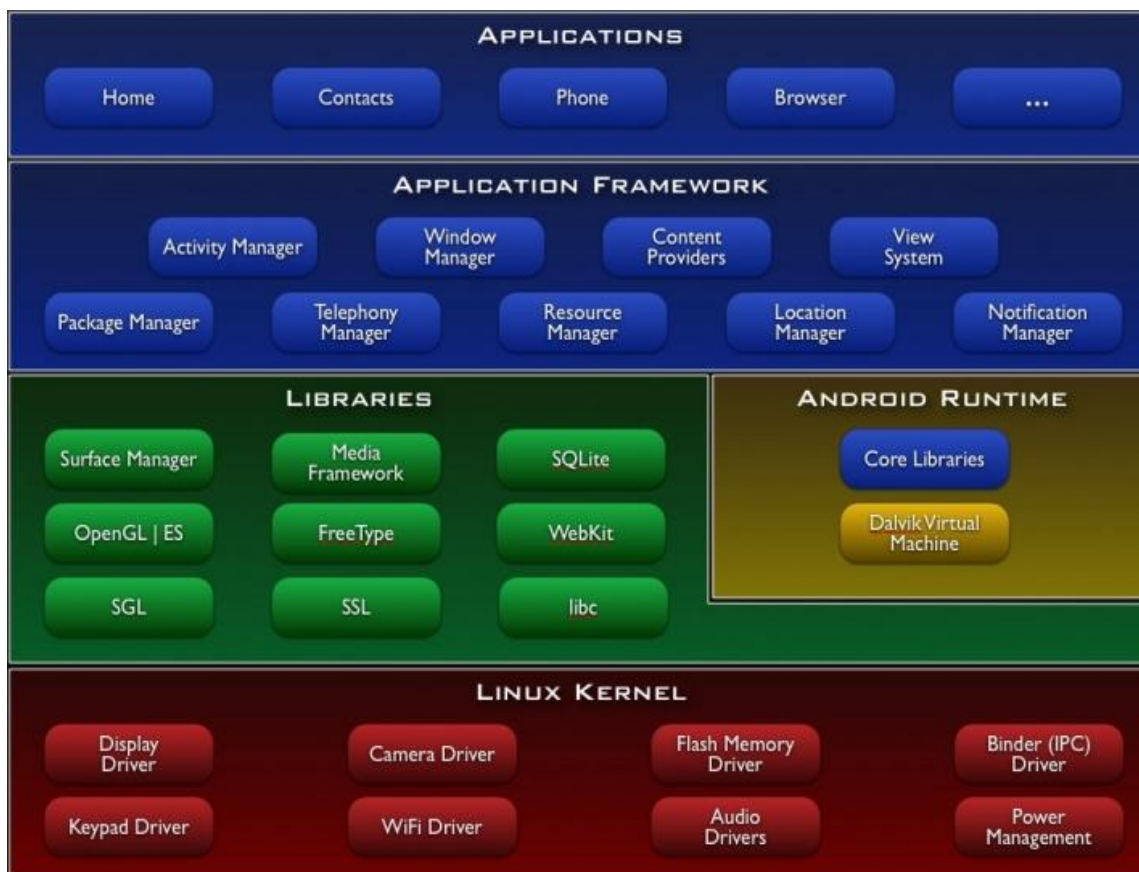
U ovom poglavlju dat je opis Android operativnog sistema za mobilne telefone i tablete na kojem aplikacija radi, Android sistemske biblioteke Bionic libc kao i kratak opis UPnP i DLNA protokola.

2.1 Android operativni sistem

Android operativni sistem je razvila grupacija firmi okupljena oko *Open Handset Alliance*, kao operativni sistem namenjen prevashodno mobilnim telefonima. Baziran je na Linuks operativnom sistemu pa se uspešno koristi i u drugim uređajima kao što su tablet, laptop i netbook računari, čitači elektronskih knjiga, digitalni televizijski prijemnici...

Na slici 1. prikazana je arhitektura Android operativnog sistema:

- Linuks kernel nalazi se na dnu Android arhitekture i deluje kao sloj apstrakcije izmedju hardvera i ostatka steka. Kernel obezbeđuje programsku podršku za različite delove mobilnih uređaja kao i kontrolu paljenja. Upravljanje memorijom i procesima kao i mrežnim stekom je takođe u nadležnosti linuks kernela
- Biblioteke obezbeđuju skup C i C++ biblioteka koje se koristi od strane različitih komponenti Android operativnog sistema. MediaFramework biblioteka se koristi za skladištenje i reprodukciju multimedijalnih sadržaja. Sqlite je baza podataka dostupna svim aplikacijama. SGL biblioteka se u osnovi koristi za prikazivanje dvodimenzionalne grafike, dok se OpenGL ES koristi za trodimenzionalnu. Biblioteka Libc biće detaljno opisana u sledećem poglavlju.

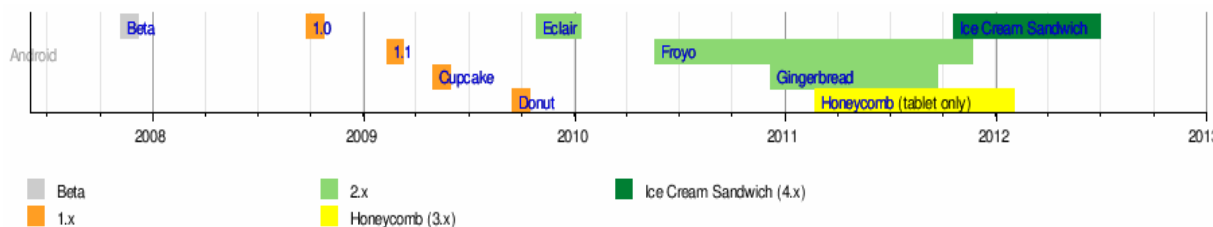


Slika 1 – Arhitektura Android operativnog sistema

- Android Runtime sadrži core biblioteke, koje obezbeđuju osnovnu funkcionalnost dostupnu u osnovnim bibliotekama Java programskog jezika, i Dalvik Virtual Machine čija je uloga da smanji trag memorije i da efikasno omogući rad više virtualnih mašina.
- Application Framework omogućava ponovnu upotrebu komponenti. Svaka aplikacija može da koristi mogućnosti drugih komponenti kao i da omogući korišćenje svojih. Svaka aplikacija u osnovi ima sledeće komponente: View System (sadrži dugmiće, liste, polja za unos teksta), Activity Manager (kontrolira kretanje i upravlja životnim ciklusom aplikacije), Resource Manager (omogućava pristup sadržajima kao što su grafike, izgledi, lokalni stringovi...), Content Provider (dozvoljava aplikaciji da podeli svoje podatke i da pristupa podacima drugih aplikacija).
- Application programi koji se dobijaju uz svaki Android uređaj. Kalendar, pretraživač, program za slanje tekstualnih poruka, proveru internet pošte...

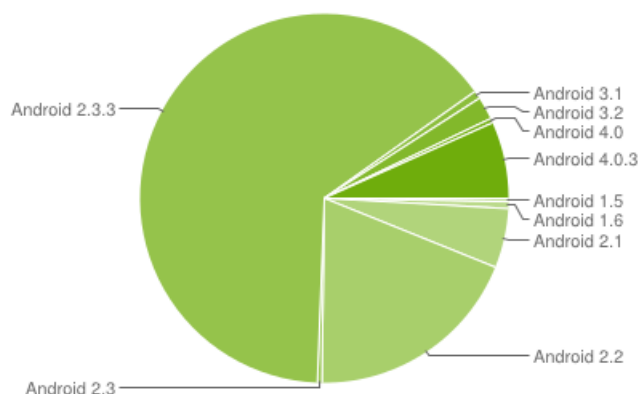
Prva komercijalna verzija Android operativnog sistema, pod radnim nazivom Apple pie, je predstavljena 28. Septembra 2008. godine u mobilnom telefonu HTC Desire.

Bazirana je na Linuks kernelu 2.6.25 i omogućava skidanje aplikacija sa Android prodavnice, poseduje internet pretraživač koji može da prikaže i uveća sadržaj HTML i XHTML internet stranica, media plejer, podršku za kameru i još dosta drugih mogućnosti. Sledeća verzija Banana bread donela je malu nadogradnju na dosadašnji sistem pre svega u otklanjanju



Slika 2 – Hronologija Android operativnog sistema

programskih grešaka koje su se uočile u prvoj verziji. Sa pojavom verzije pod nazivom Eclair ispravljene su mnoge postojeće greške u samom sistemu i dodate dodatne podrške za rad sa kamerom kao i podrška za Bluetooth 2.1 standard. Verzija Froyo predstavljena je maja 2010. godine ubrzala je rad sa memorijom i poboljšala performanse samog sistema. Sledeća verzija Android operativnog sistema Gingerbread prelazi na novi kernel 2.6.35 i dodatno poboljšava korisnički interfejs, donosi podršku za veće displeje kao i za neke dodatne senzore, ova verzija je trenutno na tržištu najrasprostranjenija sa oko 65% udela (1.06.2012.) u mobilnim uređajima koji koriste Android operativni sistem.



Slika 3 – Udeo različitih verzija Android operativnog sistema, Jun 2012

Verzija Honeycomb 3.0 je prilagođena tablet računarima, dodati su joj interfejs elementi kao što su system bar i action bar na dnu odnosno na vrhu displeja, pojednostavljena je upotreba multitaskinga, a od verzije 3.1 uvedena je podrška za spoljašnju tastaturu i džojstike za igranje. Poslednja verzija Android operativnog sistema koja se nalazi na tržištu je Ice

Cream Sandwich zasnovana je na Linuks kernelu 3.0.1, donosi dodatna poboljšanja kao što je snimanje u FullHD rezoluciji, prepoznavanje lica pri otključavanju displeja...

2.2 Bionic libc

Bionic libc, koji je nastao iz standardne libc biblioteke, razvila je *Google* kompanija u svrhu korišćenja u ugrađenim (embedded) sistemima kao i u mobilnim telefonima i tabletima zasnovanim na Android operativnom sistemu. Glavni razlozi korišćenja ove biblioteke su sledeći :

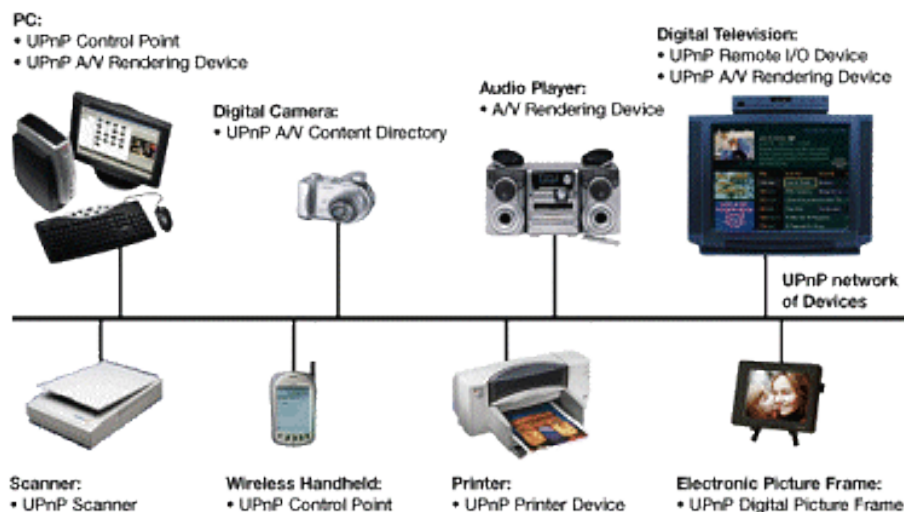
- BSD License: familija dopusnih licenci za besplatne programe.
- Veličina: bionic libc je male veličine jer se mora učitavati u svakom procesu
- Brzina: usled korišćenja u procesorima sa niskim frekvencijama biblioteka mora biti brza. To se postiže sa brzim kodovima putanja, kao i malom i brzom prilagođenom implementacijom POSIX niti (u daljem tekstu pthread).

Bionic poseduje podršku za važne tipične Android servise kao što su svojstva sistema i prijavljivanje.

Ova biblioteka se razlikuje od standardne glibc zato što ne podržava izuzetke u C++ programskom jeziku i široke karaktere (veće od 8 bita), koji nisu potrebni u Androidu. Podrška za pthread je potpuno nova i pravljen je samo za Android operativni sistem. Većinu pthread funkcija iz osnovne implementacije nije moguće koristiti kao što je *pthread_cancel()* koja ne postoji pa je nemoguće zatvoriti nit iz druge niti. Zbog svega ovoga navedenog ceo izvorni kod koji će se koristiti u aplikativnom delu mora se sastaviti u odnosu na bionic libc.

2.3 UPnP i DLNA

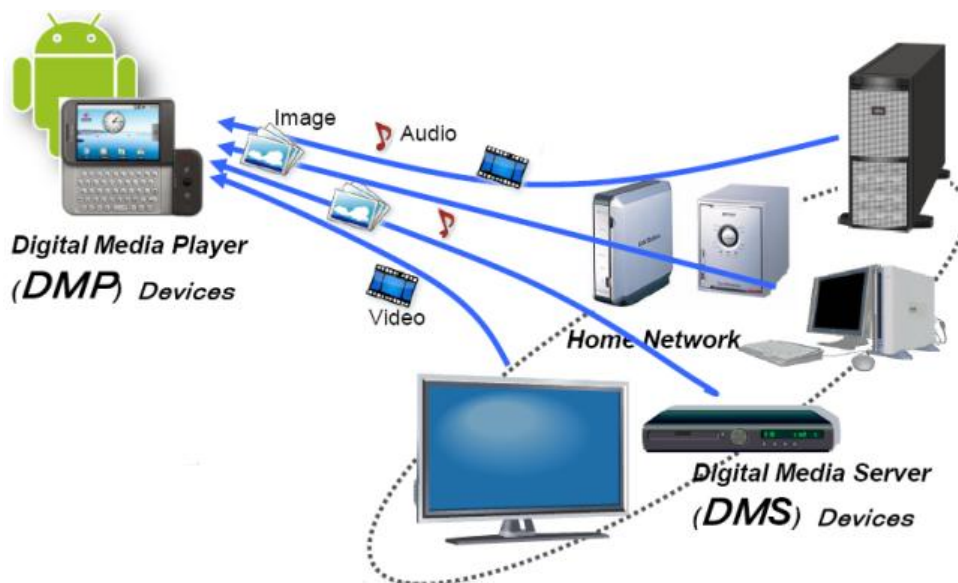
UPnP (Universal Plug and Play) je skup mrežnih protokola koji omogućavaju umreženim uređajima kao što su računari, štampači, bežične pristupne tačke, mobilni uređaji da bez problema otkriju prisustvo drugih uređaja u mreži i da uspostave mrežne usluge za deljenje podataka, komunikaciju i zabavu. Ovaj protokol je definisan od strane UPnP forum grupacije koju sačinjava preko 800 članova među kojima su i najveće svetske korporacije Microsoft, Nokia, Philips, Samsung, Intel, LG, iWedia. UPnP podržava nula konfiguraciju što znači da korisnik ne mora ništa ručno da konfigurise. Ovim se uređajima omogućava dinamički pristup mreži, dobijanje IP adrese i stavljanje na raspolaganje svoje sposobnosti drugima u mreži. Uređaj može napustiti mrežu u svakom trenutku ili automatski ako se više ne koristi. UPnP tehnologija je nezavisna od operativnog sistema i programskog jezika i izgrađena je na internet baziranim tehnologijama kao što su IP, TCP, UDP, HTTP i XML.



Slika 4 – Umreženi UPnP uređaji

DLNA (Digital Living Network Alliance) je neprofitna organizacija koja je definisala smernice za razmenu digitalnih multimedijalnih sadržaja između korisničkih uređaja kao što su računari, mobilni telefoni i kamere i ostalih multimedijalnih uređaja. DLNA koristi UPnP za traženje, kontrolu i upravljanje multimedijom. Uređaje koje DLNA podržava i mehanizam pristupa multimediji preko mreže definiše UPnP. DLNA uređaji su podeljeni u tri klase kućne mrežne uređaje, ručne mobilne uređaje i kućne infrastrukturne uređaje. Kućni mrežni uređaji se dele na :

- Digital Media Server (DMS) čuva sadržaj i čini ga dostupnim u mreži digital media player-u i digital media renderer-u
- Digital Media Pleyer (DMP) nalazi sadržaj na DMS-u i omogućuje reprodukciju i prikazivanje tog sadržaja



Slika 5 – Kako DLNA funkcioniše

- Digital Media Renderer (DMR) reprodukuje sadržaj primljen od strane digital media controller-a
- Digital Media Controller (DMC) pronalazi sadržaj na DMS-u i reprodukuje ga na DMR-u
- Digital Media Printer (DMPr) omogućuje štampanje sadržaja koga mogu koristiti DMP i DMC

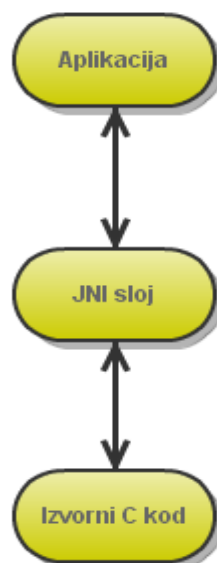
3. Koncept rešenja

Integracija DMP-a u Android operativni sistem zahteva postojanje više slojeva programske podrške. Svaki od slojeva predstavlja prilagođen servis za druge slojeve.

Na najvišem aplikativnom nivou se nalazi Android aplikacija koja omogućuje reprodukciju slika, audio i video sadržaja koji se nalaze na mrežnom poslužiocu.

Sledeći srednji sloj je Java Native Interface (JNI) sloj koji omogućuje komunikaciju između aplikativnog dela pisanog u Java programskom jeziku i izvornog koda pisanog u programskom jeziku C.

Poslednji nivo je izvorni C kod u kome su implementirane biblioteke libdlna, libupnp i libmedia koje omogućavaju puno korišćenje DLNA i UPnP resursa.



Slika 6 – Slojevi programske podrške

4. Programsko rešenje

U ovom poglavlju je opisano programsko rešenje koje se sastoji od građenja dobijenog izvornog C koda za Android operativni sistem i realizacija aplikacije u Java programskom jeziku.

4.1 Građenje izvornog koda

Za građenje izvornog C koda za rad na Android operativnom sistemu potrebno je po konvenciji napraviti datoteku sa ekstenzijom “.mk”. Glavna datoteka se uvek naziva Android.mk, i ona se nalazi na vrhu izvornog stabla. Sledi primer Android.mk, na vrhu izvornog stabla, sa kojim gradimo deljenu biblioteku u kojoj se nalaze funkcije koje preko JNI sloja koristimo u aplikaciji:

```
LOCAL_PATH:= $(call my-dir)
include $(CLEAR_VARS)
LOCAL_SRC_FILES:= \
    android_dtv_media_explorer_MediaExplorerControlNative.cpp \
    filelib.c \
    file-local.c
LOCAL_C_INCLUDES:= \
    $(LOCAL_PATH) \
LOCAL_MODULE := lib_media_explorer
LOCAL_WHOLE_STATIC_LIBRARIES:=libupnp libdlna
LOCAL_PRELINK_MODULE := false
include $(BUILD_SHARED_LIBRARY)
```

LOCAL_PATH - promenljivom počinje svaki Android.mk i ona pronalazi izvorne kodove u razvojnom stablu. Funkcija “my-dir” vraća putanju do direktorijuma gde se Android.mk nalazi.

CLEAR_VARS - promenjiva briše sve LOCAL_XXX osim LOCAL_PATH zato što se sve kontrolne datoteke parsiraju zajedno usled čega su sve promenjive globalnog karaktera.

LOCAL_SRC_FILES - navode se sve C ili C++ datoteke koje se grade.

LOCAL_C_INCLUDES - navode se putanje do direktorijuma u kojima se nalaze “.h” datoteke koje pozivaju C ili C++ datoteke.

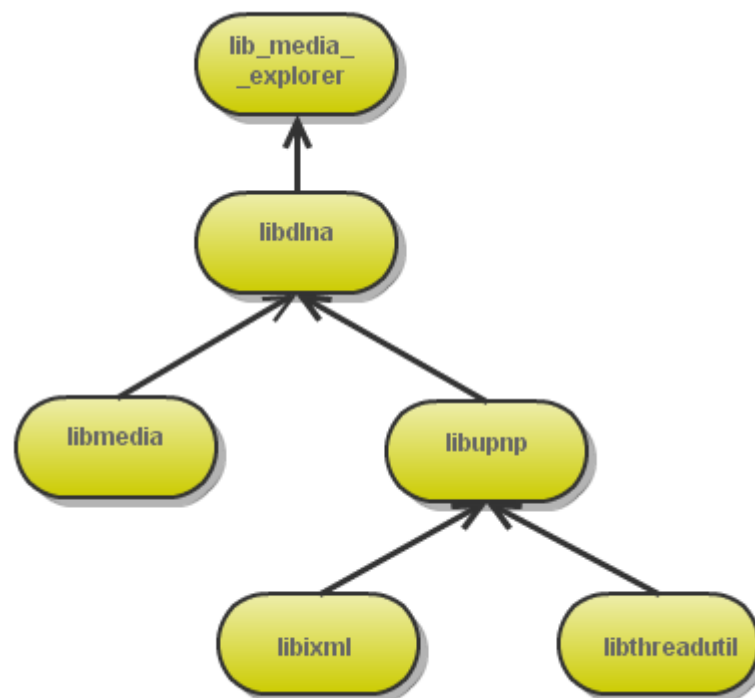
LOCAL_MODULE – naziv deljene biblioteke koju pravimo.

LOCAL_WHOLE_STATIC_LIBRARIES – navode se nazivi statičkih biblioteka koje su nam potrebne za građenje za Android operativni sistem.

LOCAL_PRELINK_MODULE – prevencija greške koje se mogu javiti pri povezivanju.

BUILD_SHARED_LIBRARY – građenje deljene biblioteke.

Na sličan način se grade i ostale biblioteke libxml, libthreadutil, libupnp, libmedia, libltna. Na slici 7. je prikazano stablo izvornog C koda:



Slika 7 – Stablo izvornog C koda

4.2 Aplikativni sloj

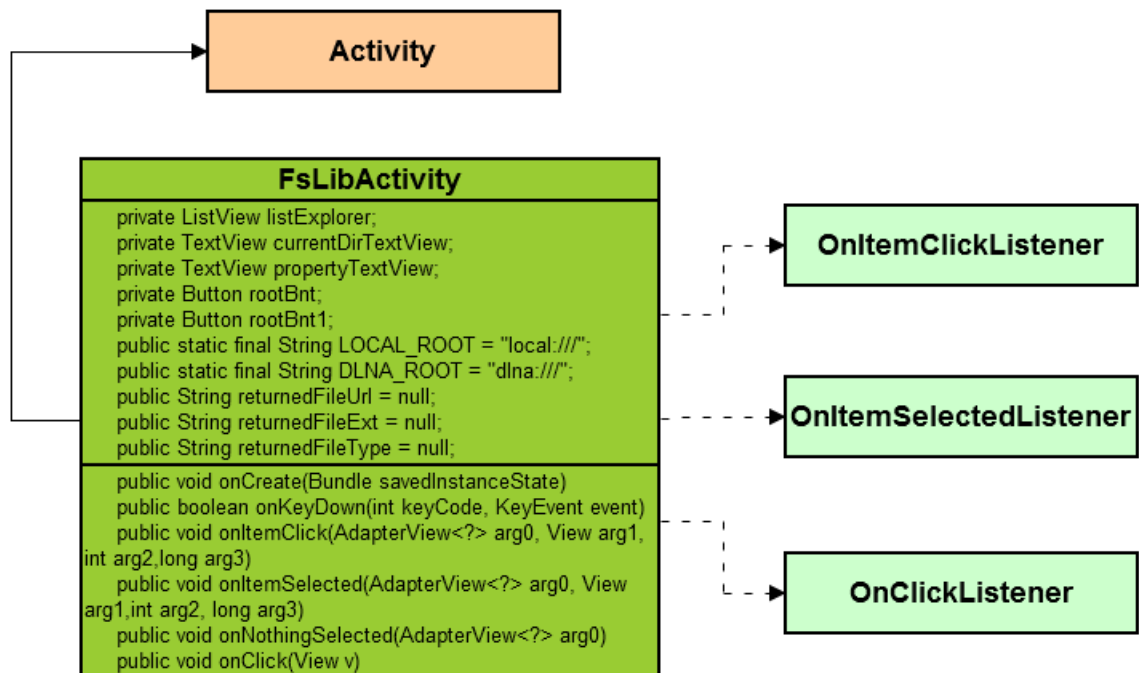
Kao razvojno okruženje za izradu ovog zadatka korišćeno je Eclipse Indigo sa ADT plugin-om i Android SDK Tools-om. Ovaj deo projektnog zadatka je pisan u Java programskom jeziku. Prikaz klasa koje su korišćene u izradi zadatka i njihov kratak opis dat je u tabeli 1.

<u>Klase sistema</u>	<u>Opis klasa sistema</u>
FsLibActivity.java	Klasa koja pokreće početnu aktivnost – korisnik odabira odakle želi sadržaj sa DLNA poslužioca ili sa lokala
BmpActivity.java	Klasa koja prikazuje sliku na displeju
VideoActivity.java	Klasa koja reprodukuje audio i video sadržaj

Tabela 1 – Prikaz klasa projektnog zadatka

4.2.1 Klasa FsLibActivity

Na slici 8. prikazana je FsLibActivity klasa sa svim svojim atributima i metodama, kao i međusobna zavisnost klasa i interfejsa.



Slika 8 – Prikaz klase FsLibActivity

Klasa *FsLibActivity* se prva instancira i nasleđuje Androidovu sistemsku klasu *Activity*. Ona pokreće program i prikazuje početni izgled, dva dugmeta koji nude mogućnost korisniku da izabere odakle će da reprodukuje muzički sadržaj sa svog uređaja ili sa DLNA poslužioca. Ova klasa implementira *OnClickListener* Androidov sistemski interfejs, na osnovu čega je moguće kliknuti na dugme. Pritiskom na prvo dugme korisnik dobija listu sa direktorijumima, koji se nalaze na uređaju, kao i mogućnost pretraživanja istih. Pritiskom na drugo dugme korisnik dobija listu svih DLNA poslužioca koji se trenutno nalaze u njegovoj mreži. Takođe klasa implementira i dva listenera *OnItemClickListener* i *OnItemSelectedListener* koji se koriste pri radu sa listama. Kada korisnik pronađe sliku koju želi da pogleda i klikne na nju, aktivira se *OnItemClickListener*, poziva se klasa *BmpActivity* koja obrađuje sliku, a ako želi da pogleda video ili da reprodukuje audio sadržaj poziva se klasa *VideoActivity*.

4.2.2 Klasa BmpActivity

Klasa *BmpActivity* je zadužena za prikaz slika na ekranu. Nasleđuje Androidovu sistemsku klasu *Activity*.

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.bmp);
    //Get URL from FsLibActivity
    Bundle bundle = getIntent().getExtras();
    valueUrl = bundle.getString("android.dtv.1");
    myImageView = (ImageView) findViewById(R.id.imageViewBmp);
    try {
        //Set picture as a screen content
        myImageView.setImageBitmap(getBitmapFromUrl(valueUrl));
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

Funkcija *public void onCreate(Bundle savedInstanceState)* iz klase *FsLibActivity* dobija adresu, u formi URL-a, na kojoj se nalazi slika. Poziva funkciju *public static Bitmap getBitmapFromUrl(final String url)* koja dekodira sliku i priprema je za prikaz.

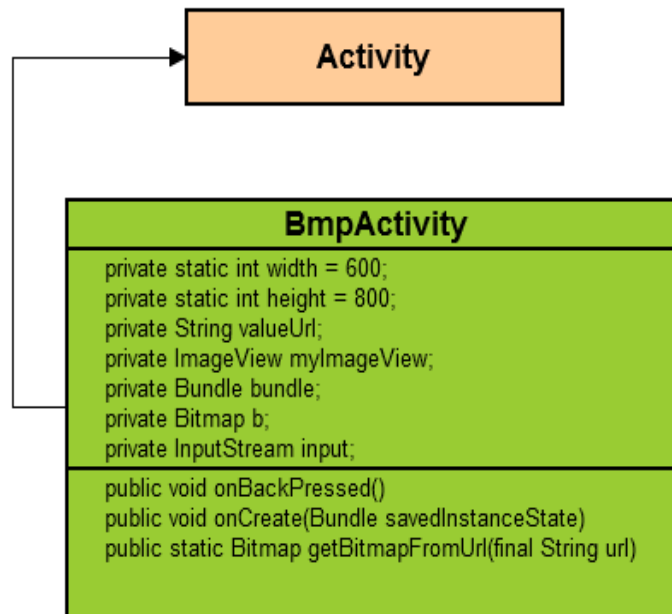
```
BitmapFactory.Options options=new BitmapFactory.Options();
options.inSampleSize = 2;
Bitmap b=BitmapFactory.decodeStream(input,null,options);
```

```

if ( height > options.outHeight || width > options.outWidth) {
    b = Bitmap.createScaledBitmap(b,options.outWidth,options.outHeight,false );
}

```

Kada se ovaj proces završi slika se iscrtava na ekran. Na slici 9. prikazana je BmpActivity klasa sa svim svojim atributima i metodama.



Slika 9 – Prikaz klase BmpActivity

4.2.3 Klasa VideoActivity

Klasa *VideoActivity* je zadužena za reprodukciju video i audio sadržaja. Nasleđuje Androidovu sistemsku klasu *Activity*. Funkcija `public void onCreate(Bundle savedInstanceState)` iz klase *FsLibActivity* dobija adresu, u formi URL-a, na kojoj se nalazi video ili audio sadržaj.

```

// Set video as a screen content
setContentView(R.layout.video);
//Create object of myVideoView
VideoView myVideoView = (VideoView)findViewById(R.id.myvideoview);
// Set video/audio path
myVideoView.setVideoPath(valueURL);

```

Nakon instanciranja modula koji će biti zadužen za reprodukciju multimedijalnog sadržaja, neophodno je omogućiti osnovne kontrole za reprodukciju. Funkcionalnost *MediaController* objekta se povezuje sa *VideoView* modulom, nakog čega oni čine jednu celinu, koja obezbeđuje reprodukciju audio i video sadržaja i kontrolu nad njim.

```

// Create object of media controller

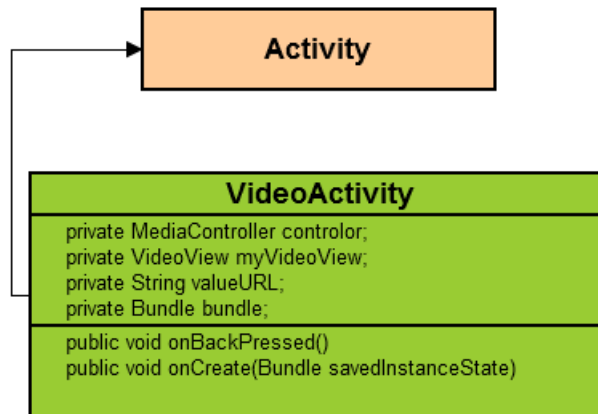
```

```
controlor = new MediaController(this);  
// Attach controller to view view object  
myVideoView.setMediaController(controlor);
```

Posle svega ovoga pokreće se reprodukcija multimedijalnog sadržaja.

```
// To force focus to a specific view  
myVideoView.requestFocus();  
// Start video/audio  
myVideoView.start();
```

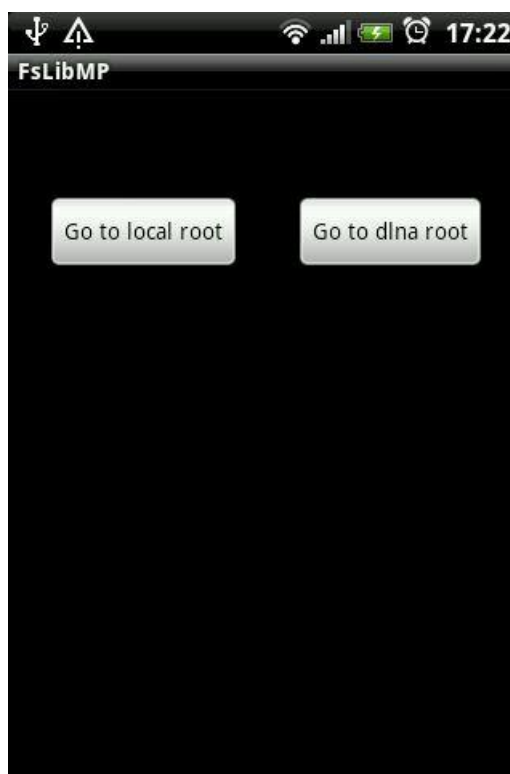
Na slici 10. prikazana je VideoActivity klasa sa svim svojim atributima i metodama.



Slika 10 – Prikaz klase VideoActivity

5. Rezultati

U ovom poglavlju je dat opis rezultata programskog rešenja. Rešenje je testirano na mobilnim telefonima sa verzijama Android operativnog sistema 2.3.3 Gingerbread i 4.0.3 Ice Cream Sandwich. Pri pokretanju aplikacije dobija se izgled prikazan na slici 11.



Slika 11 – Izgled aplikacije pri pokretanju programa

Od korisnika se zahteva da izabere kom sadržaju želi da pristupi, da li želi da pretražuje multimedijalni sadržaj sa svog mobilnog uređaja ili želi da pristupi DLNA

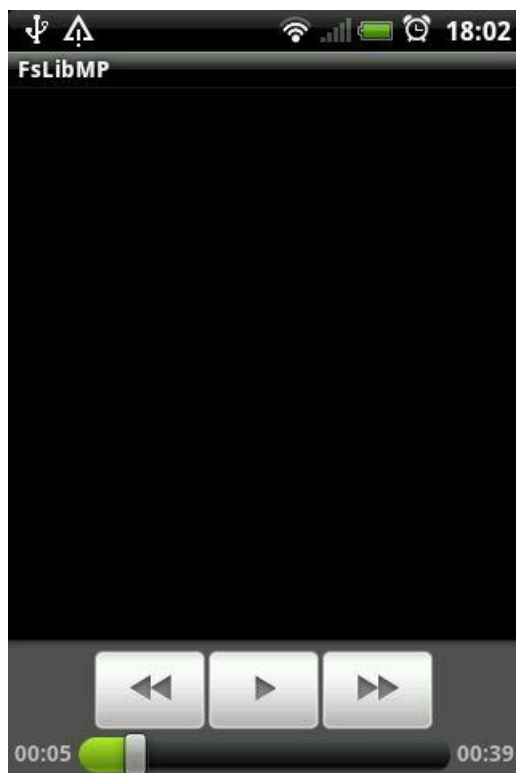
poslužiocima. Pritiskom na dugme *Go to dlna root* dobiće listu svih DLNA poslužioca koji se nalaze u mreži kojoj je pristupio. Pritiskom na dugme *Go to local root* korisnik će dobiti koreni direktorijum svog mobilnog uređaja.

Tokom ispitivanja je utvrđeno da se za dovoljno velike slike program gasi zato što virtuelna memorija nije dovoljno velika da dekoduje sliku. Ovaj problem je rešen smanjenjem broja uzoraka koji se dekoduju, i skaliranjem slike ako se slika smanjila usled smanjenja broja uzoraka koji se dekoduju. Na slici 12. prikazana je reprodukovana slika na mobilnom uređaju.



Slika 12 – Izgled aplikacije pri reprodukciji slike

Na kraju je bilo potrebno ispitati funkcionalnost pri reprodukciji audio i video sadržaja. Pri reprodukciji audio sadržaja ekran će imati crnu podlogu za vreme trajanja reprodukcije, a u slučaju video datoteke na ekranu će se prikazati njen sadržaj. Na slikama 13. i 14. respektivno je prikazana reprodukcija audio i video sadržaja.



Slika 13 - Izgled aplikacije pri reprodukciji audio sadržaja



Slika 14 - Izgled aplikacije pri reprodukciji video sadržaja

Na slikama 13. i 14. se može uočiti razlika u kontroleru za reprodukciju multimedijalnih sadržaja. Ta razlika proizilazi iz činjenice da su različiti kontroleri za različite verzije Android operativnog sistema. Sadržaj na slici 13. je reprodukovano u verziji 2.3.3 Gingerbread, dok se na slici 14. reprodukcija vrši na verziji 4.0.3 Ice Cream Sandwich Android operativnog sistema.

6. Zaključak

Ovaj rad predstavlja jedno rešenje realizacije sistema za reprodukciju multimedijalnog sadržaja u DLNA mreži na mobilnim uređajima baziranim na Android operativnom sistemu.

Rešenje je univerzalno za sve verzije Android operativnog sistema počevši od verzije 2.3 Gingerbread.

Testiranje aplikacije je izvršeno na mobilnim telefonima i tabletima različitih proizvođača sa verzijama operativnog sistema 2.3, 2.3.3, 2.3.4 i 4.0.3 (Ice Cream Sandwich). Na svim ovim uređajima uspešno je reprodukovan multimedijalni sadržaj koji se nalazi na DLNA poslužiocima kao i sadržaj koji se nalazi na tom uređaju.

Realizacijom ovog sistema i demonstracijom njegovih mogućnosti, uočava se i postojanje prostora za dalji napredak.

Glavni akcenat pri realizaciji ovog zadatka je bio omogućavanje rada DLNA servisa na ciljnoj platformi. Usled ove činjenice aplikacija služi za proveru rada sistema, jednostavna je za korišćenje bez nekih naprednih funkcija.

U daljem razvoju ove aplikacije treba se bazirati na grafičkom delu da bi potencijalnom korisniku, kupcu, bila prijatnija za korišćenje. Od naprednijih funkcija koje bi se u nekoj budućoj verziji mogle dodati su istorija reprodukovanog sadržaja i dodavanje omiljenog sadržaja u listu.

Još jedna bitna stvar koja bi unapredila aplikaciju je omogućavanje korisniku da stavi aplikaciju u pozadinu, pri čemu bi se sadržaj i dalje reprodukovao dok korisnik obavlja druge poslove na svom uređaju na primer sluša muziku sa DLNA poslužioca i pretražuje internet.

7. Literatura

- [1] Stephan Haugsrud: *Developing Android Applications with Arctis*, Master's thesis
Norwegian University of Science and Technology Trondheim, June 2009
- [2] Rato Meier: *Professional Android Application Development*, Wrox, November 2008
- [3] Sajt Wikipedia: www.wikipedia.com , učitano dana 26.06.2012.
- [4] Sajt Stackoverflow, www.stackoverflow.com, korišćen od 01.05.2012. do
15.06.2012.
- [5] M.Schmalohr and F.Kozamernik: *Home networks*, EBU Technical Review, Q2 2009