



# 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: Марко Лончаревић

Број индекса: 12941

Тема рада: Развој ДЛНА ДМС апликације за таблет/телефоне на бази  
Андроид оперативног система

Ментор рада: Проф. др Јелена Ковачевић

Нови Сад, 2012



## 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:	Марко Лончаревић		
Ментор, МН:	др Јелена Ковачевић		
Наслов рада, НР:	Развој ДЛНА апликације за таблет/телефоне на бази Андроид оперативног система		
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 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:			
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	Др Никола Теслић	
	Члан:	Др Иштван Пап	Потпис ментора
	Члан, ментор:	Др Јелена Ковачевић	



## KEY WORDS DOCUMENTATION

Accession number, <b>ANO:</b>		
Identification number, <b>INO:</b>		
Document type, <b>DT:</b>	Monographic publication	
Type of record, <b>TR:</b>	Textual printed material	
Contents code, <b>CC:</b>	Bachelor Thesis	
Author, <b>AU:</b>	<b>Marko Loncarevic</b>	
Mentor, <b>MN:</b>	<b>PhD Jelena Kovacevic</b>	
Title, <b>TI:</b>	<b>Development DLNA application for tablet/phones based on Android operating systems</b>	
Language of text, <b>LT:</b>	Serbian	
Language of abstract, <b>LA:</b>	Serbian	
Country of publication, <b>CP:</b>	Republic of Serbia	
Locality of publication, <b>LP:</b>	Vojvodina	
Publication year, <b>PY:</b>	2012	
Publisher, <b>PB:</b>	Author's reprint	
Publication place, <b>PP:</b>	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, <b>PD:</b> (chapters/pages/ref./tables/pictures/graphs/appendices)		
Scientific field, <b>SF:</b>	Electrical Engineering	
Scientific discipline, <b>SD:</b>	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, <b>S/KW:</b>		
<b>UC</b>		
Holding data, <b>HD:</b>	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, <b>N:</b>		
Abstract, <b>AB:</b>		
Accepted by the Scientific Board on, <b>ASB:</b>		
Defended on, <b>DE:</b>		
Defended Board, <b>DB:</b>	President:	PhD Nikola Teslić
	Member:	PhD Istvan Papp
	Member, Mentor:	PhD Jelena Kovačević
		Menthor's sign

## Zahvalnost

Zahvaljujem stručnim saradnicima Tomislavu Maruni i Nikoli Crvenkoviću na pruženoj pomoći tokom izrade završnog (*bachelor*) rada.

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

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

## Sadržaj

1. Uvod .....	1
2. Teorijske osnove .....	2
2.1 DLNA .....	2
2.2 UPnP.....	5
2.3 Android operativni sistem .....	5
2.4 Bionic libc .....	7
3. Koncept rešenja .....	8
4. Programsко rešenje .....	9
4.1 Građenje izvornog koda .....	9
4.2 JNI.....	11
4.3 Opis funkcionalnosti pozivanog izvornog koda.....	12
5. Rezultati .....	13
6. Zaključak .....	17
7. Literatura .....	18

## Spisak slika

Slika 1.0. Vizuelna ilustracija primera jedne DLNA mreže.....	2
Slika 1.1. Vizuelna ilustracija primera jednog hardverskog DMS.....	3
Slika 1.2. Primer rada jednog softverskog DMS.....	4
Slika 1.3. Arhitektura Android operativnog sistema.....	6
Slika 2.0. Jasan prikaz pozicije JNI sloja.....	8
Slika 3.1. Prikaz korišćenja JNI sloja u procesu povezivanja Java i C/C++.....	11
Slika 4.0. Izgled aplikacije pri pokretanju programa.....	13
Slika 4.1. Izgled aplikacije nakon pritiska dugmeta <i>START</i> .....	14
Slika 4.2. Prikaz ekrana prilikom preuzimanja slike.....	15
Slika 4.3. Prikaz slika dostupnih klijentima.....	15

## Skraćenice

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

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

**DMS** - *Digital Media Server*, Modul za deljenje sadržaja klijentima u lokalnoj mreži

**DMC** - *Digital Media Controller*, Modul za kontrolu rada DMP i DMS u lokalnoj mreži

**DMR** - *Digital Media Renderer*, Modul koji vrši reprodukciju sadržaja primljenog od DMC

**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*, Sloj koji obezbeđuje komunikaciju između aplikativnog sloja pisanih u Java programskom jeziku i sloja pisanih u programskom jeziku C

**URL** - *Uniform Resource Locator*, Jedinstvena adresa resursa

**NAS** - *Network Attached Storage* je uređaj koji u sebi sadrži hard disk

**IP** - *Internet Protocol*, Internet protokol

**LAN** - *Local area network*, Lokalna računarska mreža

**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 deljenje multimedijalnog (slika, audio, video) sadržaja, prilagođenog namenskim sistemima, tabletima i mobilnim telefonima, zasnovanim na Android operativnom sistemu. Multimedijalni sadržaj je pohranjen u memoriji DLNA(*Digital Living Network Alliance*) servera koji ovaj sadržaj deli klijentima dostupnim u lokalnoj mreži.

Rad je sačinjen od sedam poglavlja.

Prvi deo rada se odnosi na građenje izvornog (*native*) koda biblioteka libupnp, libdlna i libmedia za Android operativni sistem za mobilne telefone verzije 2.2 (Froyo). Za testiranje i verifikaciju potrebno je napraviti jednostavnu aplikaciju u Java programskom jeziku koja inicira pokretanje rada DLNA servera.

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

U trećem poglavlju se nalazi koncept rešenja sistema za deljenje multimedijalnog sadržaja u lokalnoj mreži.

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 prilikom izrade ovog rada.

## 2. Teorijske osnove

U ovom poglavlju dat je opis DLNA(sa tipovima uređaja) i UPnP protokola, opis Android operativnog sistema za mobilne telefone i tablete na kojem aplikacija radi i Android sistemske biblioteke Bionic libc.

### 2.1 DLNA

DLNA (*Digital Living Network Alliance*) je neprofitna organizacija koja je definisala pravila za razmenu digitalnih multimedijalnih sadržaja između korisničkih uređaja kao što su računari, mobilni telefoni, kamere i ostali multimedijalni uređaji. Uređaje koje DLNA podržava i mehanizam pristupa multimediji putem mreže definisan je UPnP protocolom. DLNA uređaji su podeljeni u tri klase: kućne mrežne uređaje, ručne mobilne uređaje i kućne infrastrukturne uređaje.



Slika 1.0. Vizuelna ilustracija primera jedne DLNA mreže

---

Tipovi uređaja:

**DIGITAL MEDIA SERVER(DMS)** je uređaj za skladištenje podataka (kao na primer muzike, videa i drugih fajlova), koji moraju biti dostupni klijentima za upotrebu. Server mora biti u mogućnosti da konvertuje fajlove, pre slanja, na zahtev klijenta. DMS može biti:

- Hardverski
- Softverski

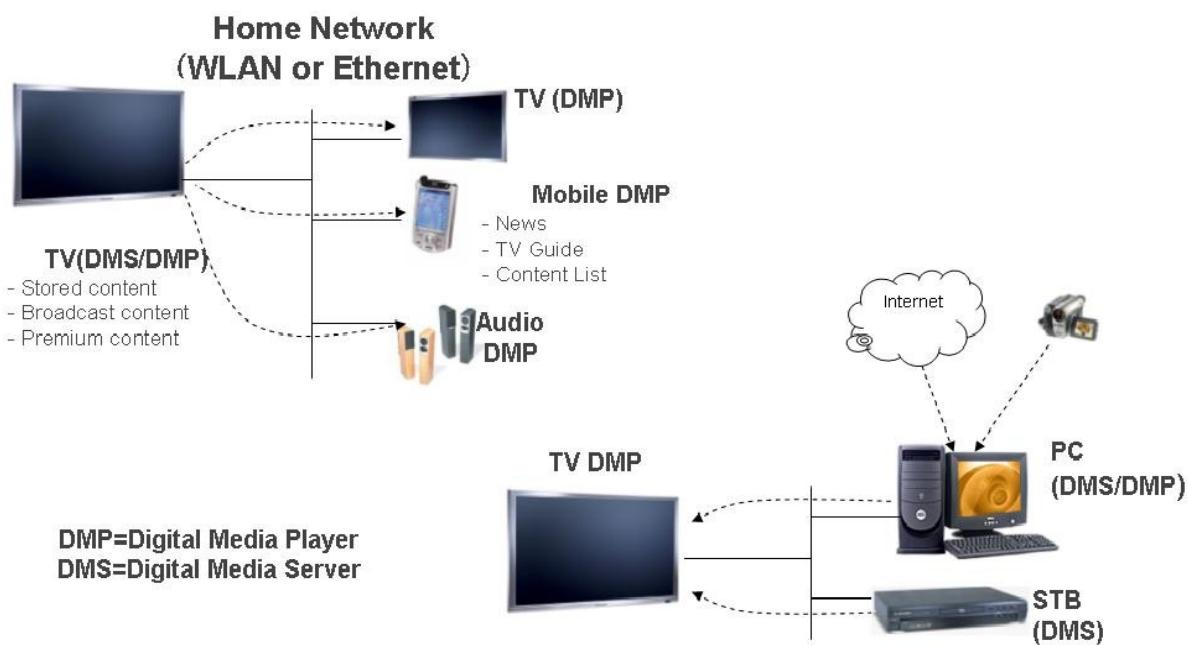
Hardverski media server podrazumeva posedovanje NAS (*Network Attached Storage* uređaja). Network Attached Storage je uređaj koji faktički u sebi sadrži hard disk (na kojem skladištite sav vaš materijal) i poseduje LAN priključak. Direktno se povezuje na ruter, i automatski postaje dostupan uređajima na mreži.



Slika 1.1. Vizuelna ilustracija primera jednog hardverskog DMS

Nasuprot hardverskom, softverski media server ne zahteva kupovinu bilo kakvog uređaja, dovoljno je samo na svoj računar instalirati program za softverski media server. Samim tim softverski media server je znatno isplatljiviji u finansiskom pogledu.

Ukoliko je u mreži prisutno više računara neophodno je dana svakom bude instaliran prethodno pomenuti program za softverski media server.



Slika 1.2. Primer rada jednog softverskog DMS

**DIGITAL MEDIA PLAYER(DMP)** je uređaj koji prikazuje sadržaj koji mu je poslat od strane servera.

**DIGITAL MEDIA KONTROLER(DMC)** nalazi sadžaje na DMS i prikazuje ih na DMP. Kao na primer, zahtevanje od servera da prikaže video na određenom TV-u.

#### **DIGITAL MEDIA RENDERER(DMR)**

To su uređaji koji vrše reprodukciju sadržaja primljenog od DMC, koji nalazi sadržaj na (DMS). Primeri uključuju televizore, audio / video prijemnike, zvučnike za muziku i slično. Važno je napomenuti mogućnost da jedan uređaj funkcioniše kao DMP i DMR.

## 2. 2 UPnP

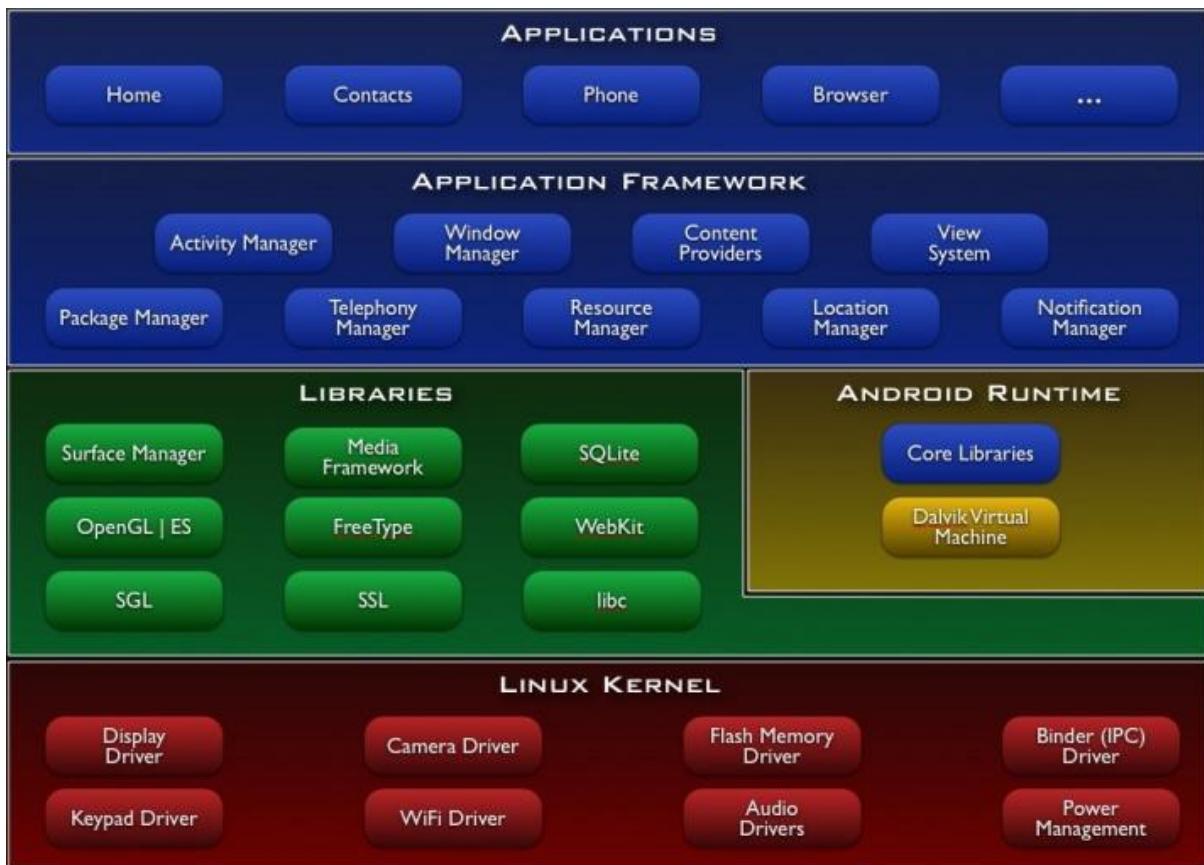
UPnP (*Universal Plug and Play*) je skup mrežnih protokola koji omogućavaju umreženim uređajima kao što su računari, štampači, bežične pristupne tačke, mobilni uređaji da bez problema otkriju prisustvo drugih uređaja u mreži i da uspostave mrežne usluge za deljenje podataka, komunikaciju i zabavu. Ovaj 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 *plug and play* (pokreni i koristi) što znači da korisnik ne mora ništa ručno da konfiguriše. Ovim se uređajima omogućuje dinamički pristup mreži, dobijanje IP adrese i stavljanje na raspolaganje svoje sposobnosti drugima u mreži. Uredaj može napustiti mrežu u svakom trenutku. UPnP tehnologija je nezavisna od operativnog sistema i programskog jezika i izgradena je na internet baziranim tehnologijama kao što su IP, TCP, UDP, HTTP i XML.

## 2.3 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 1.3. prikazana je arhitektura Android operativnog sistema:

- Linuks kernel nalazi se na dnu Android arhitekture i deluje kao sloj apstrakcije između 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
- 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 biblioteka za rad sa bazama podataka dostupna svim aplikacijama. SGL biblioteka se u osnovi koristi za prikazivanje dvodimenzionalne grafike, dok se OpenGL ES koristi za trodimenzionalnu grafiku. Biblioteka Libc biće detaljno opisana u sledećem poglavljju.



Slika 1.3 – 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 (kontroliše 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.

## 2.4 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 dostupnih 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 učitavanje biblioteke mora biti brzo. To se postiže sa brzim kodovima putanja, kao i malom i brzom prilagođenom implementacijom POSIX niti (u daljem tekstu *pthread*).

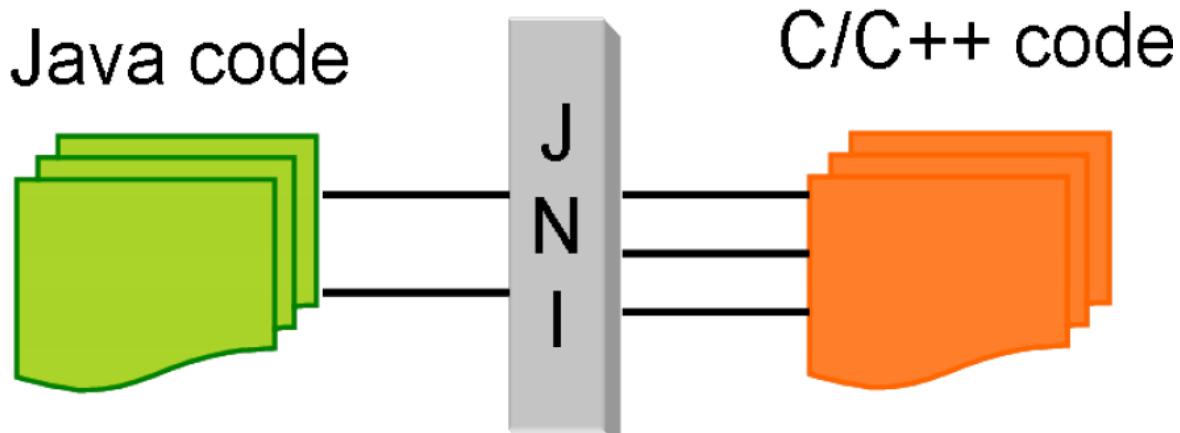
Bionic poseduje podršku za važne tipične Android servise kao što su svojstva sistema i prijavljivanje.

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 delimično izmenjena za Android operativni sistem. Neke *pthread* funkcije iz osnovne implementacije nije moguće koristiti kao što je *pthread\_cancel()* koja ne postoji, te 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.

### 3. Koncept rešenja

Integracija DMS-a u Android operativni sistem zahteva postojanje više slojeva programske podrške. Svaki od slojeva komunicira sa susednim slojevima. Na najvišem aplikativnom nivou se nalazi Android aplikacija koja omogućuje pozivanje JNI sloja.

*Java Native Interface (JNI)* sloj obezbeđuje komunikaciju između aplikativnog sloja pisanog u Java programskom jeziku i sloja pisanog u programskom jeziku C.



Slika 2.0. Prikaz pozicije JNI sloja

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

## 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. Sledi primer Android.mk:

```
LOCAL_PATH:= $(call my-dir)
include $(CLEAR_VARS)
LOCAL_CFLAGS :=-DDEFAULT_IFACE=\"eth0\" \
               -DTMP_SETTINGS_PATH=\"/data/data/android.dtv\"
LOCAL_CFLAGS+=-DANDROID_LIBDLNA_BUILD

LOCAL_SRC_FILES:= \
    /src/adapt/android_dtv_media_explorer_MediaExplorerControlNative.cpp \
    /src/common/sqlite3.c \
    /src/common/didl.c \
    /src/adapt/filelib.c \
    /src/common/dlna_profile.c \
    /src/common/av_mpeg1.c \
    /src/common/av_mpeg2.c \
    /src/common/av_mpeg4_part10.c \
    /src/common/av_mpeg4_part2.c \
    /src/common/av_wmv9.c \
```

```
LOCAL_C_INCLUDES:= \
$(LOCAL_PATH)/../libfdlna-rel/src/adapt/\
$(LOCAL_PATH)/../fs-lib/src/\
$(LOCAL_PATH)/../libfdlna-rel/src/dmc/\
$(LOCAL_PATH)/../libfdlna-rel/src/dmr/\
  
LOCAL_MODULE := libdtv_media_explorer
LOCAL_LDLIBS := -llog
LOCAL_WHOLE_STATIC_LIBRARIES:=libupnp libmedia
LOCAL_MODULE_TAGS:=optional
LOCAL_PRELINK_MODULE:=false
include $(BUILD_SHARED_LIBRARY)
```

LOCAL\_PATH - promenjivom počinje svaki Android.mk i ona pronalazi izvorne kodove u razvojnog 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.

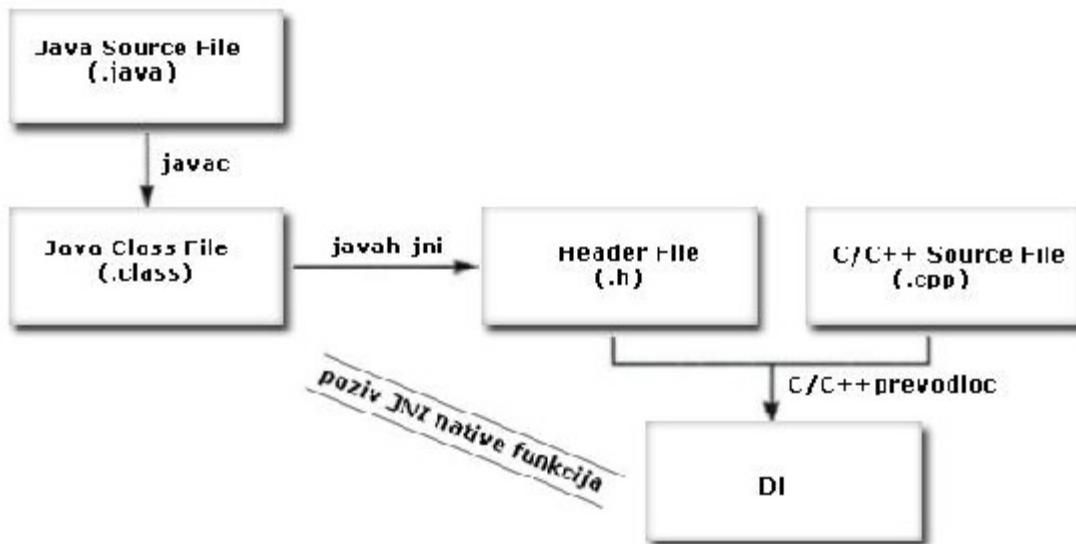
LOCAL\_PRELINK\_MODULE - prevencija greške koje se mogu javiti pri povezivanju.

BUILD\_SHARED\_LIBRARY - građenje deljene biblioteke.

Na sličan način se grade i ostale biblioteke libxml, libthreadutil, libupnp i libmedia..

## 4.2 JNI(Java Native Interface)

Ukoliko je gradjenje datoteka uspešno izvršeno, pristupamo kreiranju JNI sloja i pozivamo željene funkcije za pokretanje i zaustavljanje rada DLNA servera. JNI sloj se kreira ukoliko je neophodno pozvati funkcije napisane u drugim programskim jezicima(C/C++).



Slika 3.1. prikaz korišćenja JNI sloja u procesu povezivanja Jave i C/C++

Poziv JNI native funkcija se odvija unutar Java sloja. JNI funkcije pozivaju DL (*Dinamic Library*) u kojem su zapisane prevedene metode korišćene na C/C++ sloju. Na taj način je ostvarena veza između Java sloja i C/C++ sloja.

Prostupak na slici 3.1. možemo zapisati:

1. Implementacija JNI native metoda na Java sloju
2. Generisanje JNI klasa
3. Kreiranje DL na temelju metoda napisanih na C/C++ sloju
4. Prevođenje programskog koda napisanog u Javi i njegovo izvršavanje

Funkcije koje se pozivaju putem JNI sloja u Javi moraju biti deklarisane kao prirodne(*native*). Primer deklaracije prirodne funkcije:

```
public native int dlna_server_init();
```

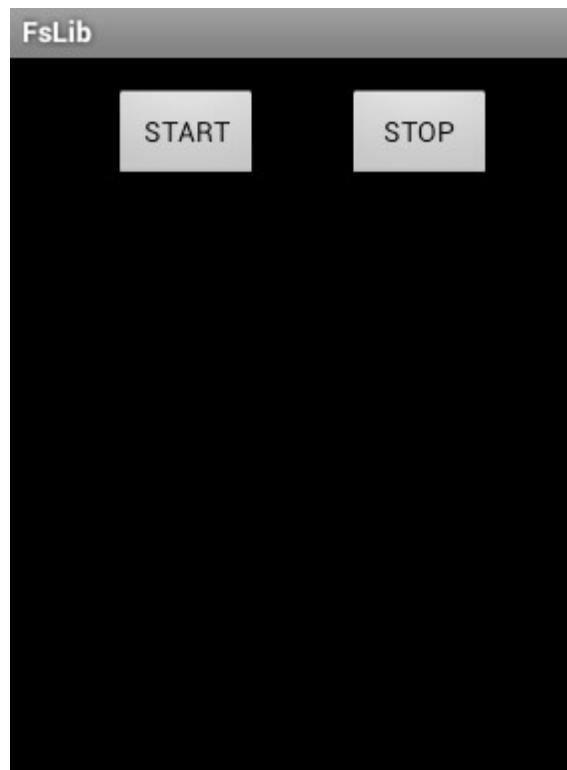
## 4.3 Opis funkcionalnosti pozivanog izvornog koda

U telu funkcije *dlna\_server\_init* koju smo pozvali putem JNI sloja se nalaze pozivi funkcija *server\_init*, *add\_init\_folders*, *content\_db\_add\_dir*. Funkcijom *add\_init\_folders*, kreiramo bazu i u nju dodajemo fascikle. Kreiramo posebne fascikle za slike, video i muziku. U okviru svake od navedenih fascikli kreirane su podfascikle u kojima će biti rasporedjen sadržaj u zavisnosti od izvođača, albuma, žanra i sl.

Funkciji *content\_db\_add\_dir* prosleđujemo putanju do direktorijuma u kom se nalazi sadržaj koji server deli klijentima. U okviru ove funkcije poziva se funkcija *ftw\_callback*. Ovom funkcijom se vrši provera da li je sadržaj na navedenoj putanji predstavlja sliku, muziku ili video. Zatim se nakon ove provere, nastavlja sa proverama i sortiranjem sadržaja po datumu kreiranja, vrsti žanra(ukoliko je u pitanju muzika), odnosno albuma ukoliko se radi o slici. Navedene funkcije su deo gradjene deljene biblioteke libdlna.

## 5.Rezultati

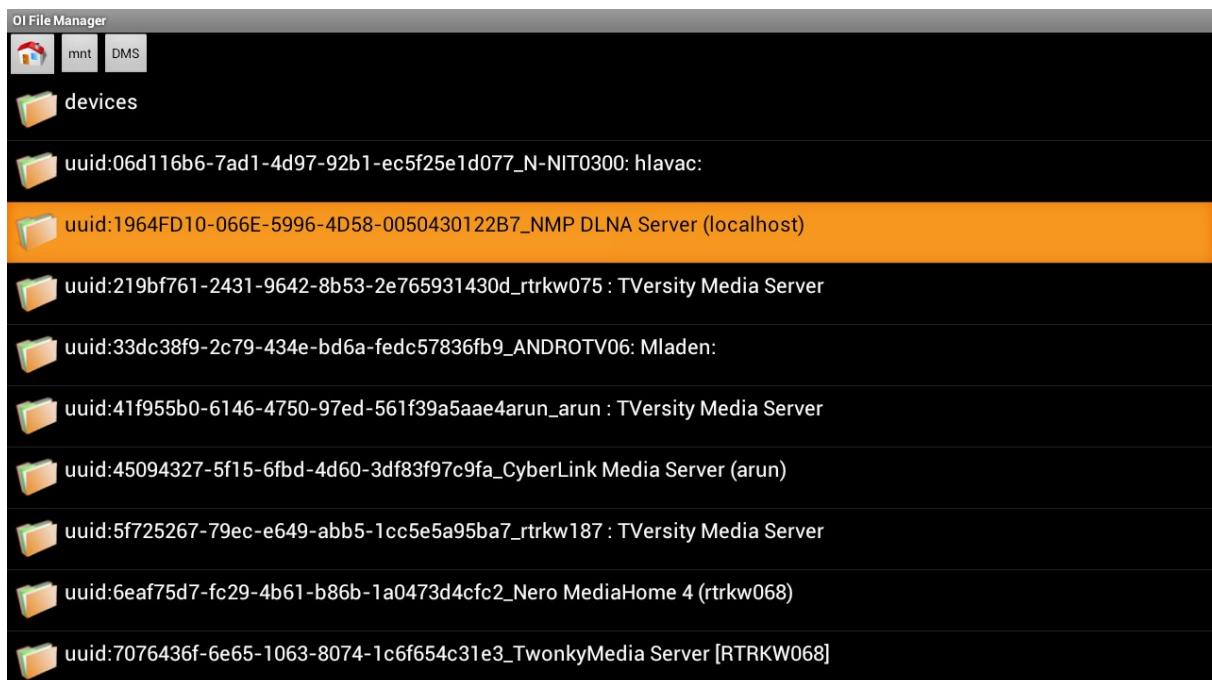
U ovom poglavlju je dat opis rezultata programskog rešenja. Rešenje je ispitivano na Visio mobilnim telefonima sa verzijama Android operativnog sistema 2.2. Froyo. Pri pokretanju aplikacije dobija se izgled prikazan na slici 11.



Slika 4.0 Izgled aplikacije kod pokretanja programa

Od korisnika se zahteva da izabere da li želi da pritisne na dugme *START*, i tako automatski pokrene server i učini njegov sadržaj deljenim ili da izabere dugme *STOP* i tako zaustavi rad servera. Samim tim deljeni sadržaj postaje nedostupan klijentima.

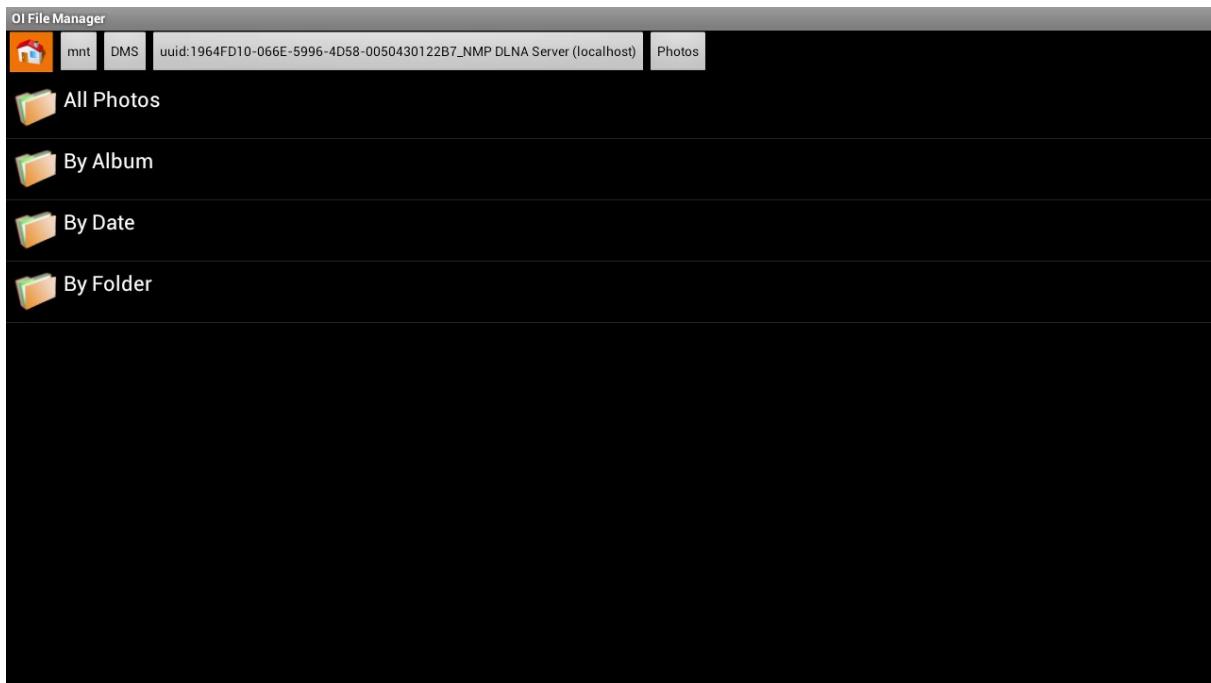
Ukoliko je izabrana prva opcija i pritisnuto dugme *START* server će se startovati i aplikacija će imati sledeći izgled:



Slika 4.1 Izgled aplikacije nakon pritiska dugmeta *START*

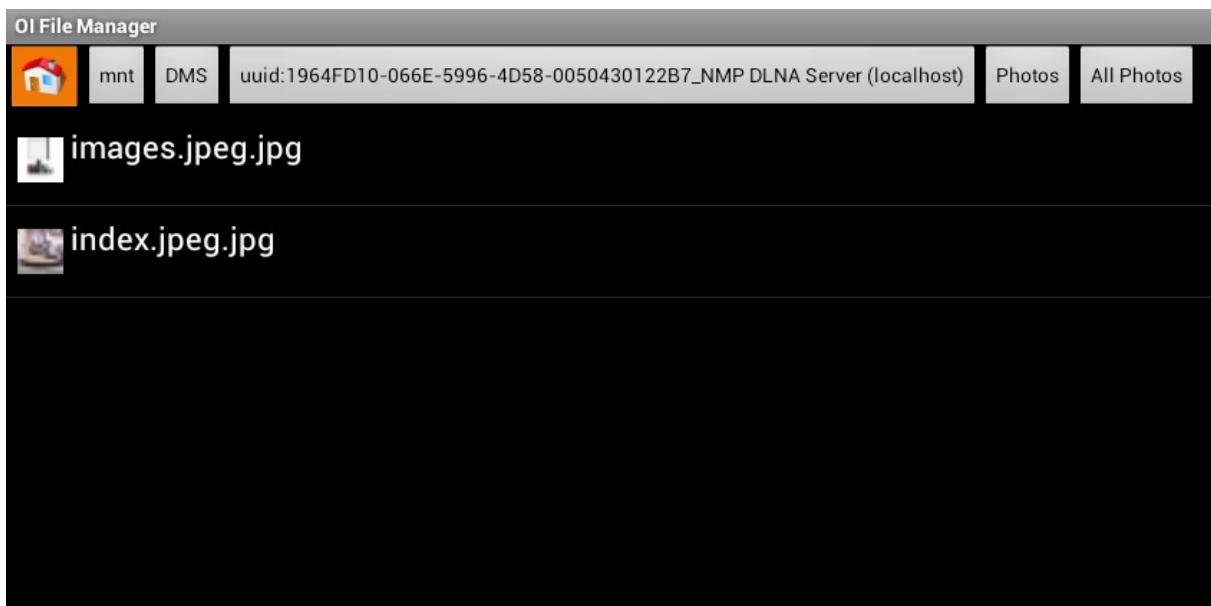
Sa slike se jasno može videti da je server pokrenut i dostupan klijentima za korišćenje. Izaberemo ga u listi servera koji su nam dostupni. Zatim je neophodno izabrati da li želimo da preuzmemos muziku, video ili sliku.

Ukoliko je izabrano da želimo preuzeti sliku na ekranu će se pojaviti sledeći prozor:



Slika 4.2 Prikaz ekrana prilikom preuzimanja slika

Slike na serveru rasporedjene su u sledećim fasciklama: Fascikla All Photos sadrži sve slike koje server deli klijentima, bez obzira na album, datum ili fasciklu. U tri fascikle ispod slike su raspoređene u zavisnosti od albuma, datuma i fascikle, u odgovarajuće fascikle.



Slika 4.3.Prikaz slika dostupnih klijentima

Postupak preuzimanja videa i muzike je u potpunosti identičan preuzimanju slika s tim da muzika pored toga što je sortirana po fasciklama i albumima, dodatno sortirana i po žanrovima odnosno izvođačima.

## 6. Zaključak

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

Rešenje je realizovano za Visio tablete sa 2.2 Froyo verzijom Android operativnog sistema.

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.

U daljem razvoju ove aplikacije treba se bazirati na grafičkom delu da bi potencijalnom kupcu bila prijatnija za korišćenje. Od naprednijih funkcija koje se u nekoj budućoj verziji mogu dodati su mogućnost da DLNA server klijentima deli i druge sadržaje osim muzike, videa ili slike (kao što su tekstualne datoteke i slično).

## 7.Literatura

- [1] Stephan Haugsrud: *Developing Android Applications with Arctis*, Master's thesis Norwegian Norwegian University of Science and Technology Trondheim, June 2009
- [2] Marvell, *88DE3010 Software API User Manual*
- [3] Rato Meier: *Professional Android Application Development*, Wrox, November 2008
- [4] Vladimir Kovačević, Miroslav Popović: *Sistemska programska podrška u realnom vremenu*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2002
- [5] M.Schmalohr and F.Kozamernik: Home networks, EBU Technical Review, Q2 2009