



**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: Ковачевић Милан
Број индекса: E12775

Тема рада: Симулација даљинског управљача са подршком за препознавање и могућност управљања гласом на Андроид платформи

Ментор рада: Др Јелена Ковачевић

Нови Сад, Јул, 2012.



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

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 :	Милан Ковачевић
Ментор, МН :	Проф. др Јелена Ковачевић
Наслов рада, НР :	СИМУЛАЦИЈА РАДА ДАЉИНСКОГ УПРАВЉАЧА СА ПОДРШКОМ ЗА ПРЕПОЗНАВАЊЕ И МОГУЋНОСТ УПРАВЉАЊА ГЛАСОМ НА АНДРОИД ПЛАТФОРМИ
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 : (поглавља/страна/ цитата/табела/слика/графика/прилога)	7/33/0/0/14/0/0
Научна област, НО :	Електротехника и рачунарство
Научна дисциплина, НД :	Рачунарска техника
Предметна одредница/Кључне речи, ПО :	Технологија претварања гласа у текстуални облик, Андроид, симулација даљинског управљача
УДК	
Чува се, ЧУ :	У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, ВН :	
Извод, ИЗ :	У овом раду приказана је једна од могућности коришћења технологије претварања гласа у текстуални облик помоћу Google Speech to Text технологије на Андроид платформи. Апликација симулира рад даљинског управљача.
Датум прихватања теме, ДП :	
Датум одбране, ДО :	
Чланови комисије, КО :	Председник: Др Мирослав Поповић
	Члан: Др Драган Самарџија
	Члан, ментор: Др Јелена Ковачевић
	Потпис ментора



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 :	Milan Kovačević
Mentor, MN :	PhD Jelena Kovačević
Title, TI :	Simulation of remote control application based on Google Speech to Text technology on Android platform.
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 : <small>(chapters/pages/ref./tables/pictures/graphs/appendixes)</small>	7/33/0/0/14/0/0
Scientific field, SF :	Electrical Engineering
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, S/KW :	Android, Google Speech to Text, simulation of remote control
UC	
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :	
Abstract, AB :	This paper presents a solution of application based on Google Speech to Text technology which simulates remote control on Android platform.
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	
Defended Board, DB :	President: Miroslav Popović, PhD
	Member: Dragan Samardžija, PhD
	Member, Mentor: Jelena Kovačević, PhD
	Mentor's sign

Zahvalnost

Zahvaljujem se svom mentoru Dr Jeleni Kovačević na stručnoj pomoći i strpljenju tokom izrade završnog (*bachelor*) rada.

Srdačno se zahvaljujem asistentima Nenadu Četiću i Ivanu Letvenčuku na podsticanju za nastanak završnog rada i za savete i podršku tokom njegove izrade.

Na kraju se zahvaljujem svima onima koji su na bilo koji način doprineli izradi ovog završnog rada.

SADRŽAJ

1. Uvod	1
2. Teorijske osnove.....	3
2.1 Speech To Text tehnologija	3
2.2 Android operativni sistem.....	4
2.3 TCP/IP.....	6
3. Koncept rešenja	7
3.1 Sprega sa korisnikom.....	8
3.2 Speech To Text algoritam.....	10
3.3 Uspostava veze.....	11
4. Programsko rešenje.....	13
4.1 Modul daljinskog upravljača (paket Control)	14
4.2 Modul uspostave veze (paket Connection)	19
4.3 Modul unapređenja algoritma (paket Artificial Intelligence)	20
5. Rezultati.....	22
6. Zaključak	25
7. Literatura	26

SPISAK SLIKA

Slika 1 – Arhitektura Android operativnog sistema	5
Slika 2 – Tok slanja, prijema i obrade podataka.....	8
Slika 3 – Izgled glavnog prozora	9
Slika 4 – Stavke menija	10
Slika 5 – Aktiviran STT u vodoravnom modu	11
Slika 6 – Izgled prozora za uspostavu veze	12
Slika 7 – Životni vek aktivnosti.....	14
Slika 8 – Klasni dijagram paketa Control.....	15
Slika 9 – Klasni dijagram paketa Control.....	16
Slika 10 – Klasni dijagram paketa Control.....	17
Slika 11 – Klasni dijagram paketa Connection.....	19
Slika 12 – Klasni dijagram paketa Artificial Intelligence.....	20
Slika 13 – Ispitivanje efikasnosti algoritma za unapređenje.....	23
Slika 14 – Rezultati prepoznavanja komandi	24

SKRAĆENICE

IP	- <i>Internet Protocol</i> , Internet Protokol
TCP	- <i>Transmission Control Protocol</i> , međumrežni protokol
API	- <i>Application Programming Interface</i> , programska sprega
OS	- <i>Operating System</i> , Operativni Sistem
STT	- <i>Speech to Text</i> , tehnologija koja pretvara govor u tekst
TV	- <i>Television</i> , Televizija
ADT Plugin	- <i>Android Development Tools plugin</i> , Android razvojni alat
XML	- <i>Extensible Markup Language</i> , jezik za opisivanje podataka
SDK Tool	- <i>Software Development Kit Tool</i> , skup alata za razvoj aplikacija

1. Uvod

Izrada ovog diplomskog rada podrazumevala je integrisanje Google tehnologije pretvaranja govora u tekst, na platformi sa Android operativnim sistemom. Za potrebe verifikacije, bilo je potrebno izraditi klijentsku aplikaciju koja oponaša rad daljinskog upravljača na uređaju koji koristi Android operativni sistem, bilo na tablet računaru ili mobilnom telefonu. Aplikacija je bazirana na Google tehnologiji prepoznavanja govora (eng. STT - *Speech To Text* tehnologija). Cilj je omogućiti govorom upravljaju aplikaciju, kako bi uređaj reagovao na odgovarajuće komande. Skup komandi je tipičan za upravljanje TV prijemnikom.

Akcent istraživanja je STT tehnologija, odnosno funkcionisanje samog algoritma, kao i rukovanje sa rezultatima dobijenih od Google servisa. Ideja rada je da se istraži koliko efikasno je moguće iskoristiti sam STT algoritam, i da li ga je moguće poboljšati zarad još preciznijih rezultata.

Ovaj rad je sačinjen od sedam poglavlja.

Drugo poglavlje opisuje prednosti i razloge korišćenja Java servisa u odnosu na trenutno rešenje programske podrške televizijskog prijemnika.

U trećem poglavlju dato je idejno rešenje kao i opis ključnih komponenti sistema.

Četvrto poglavlje sadrži detaljan opis programskog rešenja aplikacije, prikaz korišćenih funkcija i datoteka.

U petom poglavlju su opisani načini ispitivanja i verifikovanja tehnologije pretvaranja govora u tekst.

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 dat je opis STT tehnologije, Android operativnog sistema i međumrežnog protokola na kojima se zasniva sam rad.

2.1 Speech To Text tehnologija

Prepoznavanje govora (*Speech Recognition*, u mnogim kontekstima takođe poznato i kao automatsko prepoznavanje govora, računarsko prepoznavanje govora, ili kao što se pogrešno naziva, prepoznavanje glasova) je proces preobraćanja govornih signala u niz reči, uz pomoć algoritma implementiranog kao računarski program (Speech to text). Primena prepoznavanja govora koja se pojavila u poslednjih nekoliko godina uključuje glasovno biranje ili glasovno pozivanje (voice dialing, npr. *Pozovi kuću*), usmeravanje poziva (npr. *Hteo bih da ostvarim poziv na račun sagovornika*), unos jednostavnih podataka (npr. unos broja kreditne kartice), priprema strukturiranih dokumenata (npr. radiološki izveštaj) i govorna audio pretraga zasnovana na sadržaju (npr. naći podkast gde su izgovorene određene reči). Vremenom, aplikacije koje koriste ovu tehnologiju dobijaju sve više na značaju. Tako je ova tehnologija pronašla primenu u medicini, vojnoj industriji, telekomunikacijama, robotici, domaćinstvu...

Takvi moderni sistemi prepoznavanja govora opšte namene obično su zasnovani na skrivenim *Markovljevim modelima*. Ovo je statistički model koji proizvodi niz simbola ili kvantiteta. Jedan mogući razlog zbog kojeg se skriveni Markovljevi modeli upotrebljavaju u prepoznavanju govora jeste to što se govorni signal može posmatrati kao po delovima stacionarni signal ili kratkotrajan stacionarni signal. To jest, može se pretpostaviti da, u kratkom vremenskom periodu od 10 milisekundi, govor može biti shvaćen kao stacionaran

proces. O govoru se, prema tome, može misliti kao o Markovljevom modelu za mnoge stohastičke procese (poznate kao pretpostavke). Drugi razlog zbog kojeg su skriveni Markovljevi modeli popularni je zato što mogu biti obučeni automatski i zato što su jednostavni i praktični za računarsku upotrebu. U prepoznavanju govora, da bismo proizveli najjednostavniju moguću postavku, skriveni Markovljev model bi trebalo da proizvede niz n -dimenzionalnih vektora od prave vrednosti, pri čemu je n , recimo, oko 13, proizvodeći po jedan na svakih 10 milisekundi. Vektori, opet u najjednostavnijem slučaju, bi se sastojali od kepstalnih (cepstral) koeficijenta, koji se dobijaju korišćenjem Furijeove transformacije kratkotrajnog prozora govora i dekorelacije spektra korišćenjem kosinusne transformacije i onda uzimanjem prvih (najznačajnijih) koeficijenata. Skriveni Markovljev model će težiti da ima, u svakom stanju, statističku distribuciju koja predstavlja mešavinu Gausovskih raspodela verovatnoće koje imaju dijagonalne kovarijacione matrice i koje će dati verovatnoću za svaki posmatran vektor. Svaka reč, ili (za opštije sisteme prepoznavanja govora) svaka fonema, imaće različit proizvod distribucije; skriveni Markovljev model za niz reči ili fonema je napravljen spajanjem individualno obučenih skrivenih Markovljevih modela za odvojene reči i foneme.

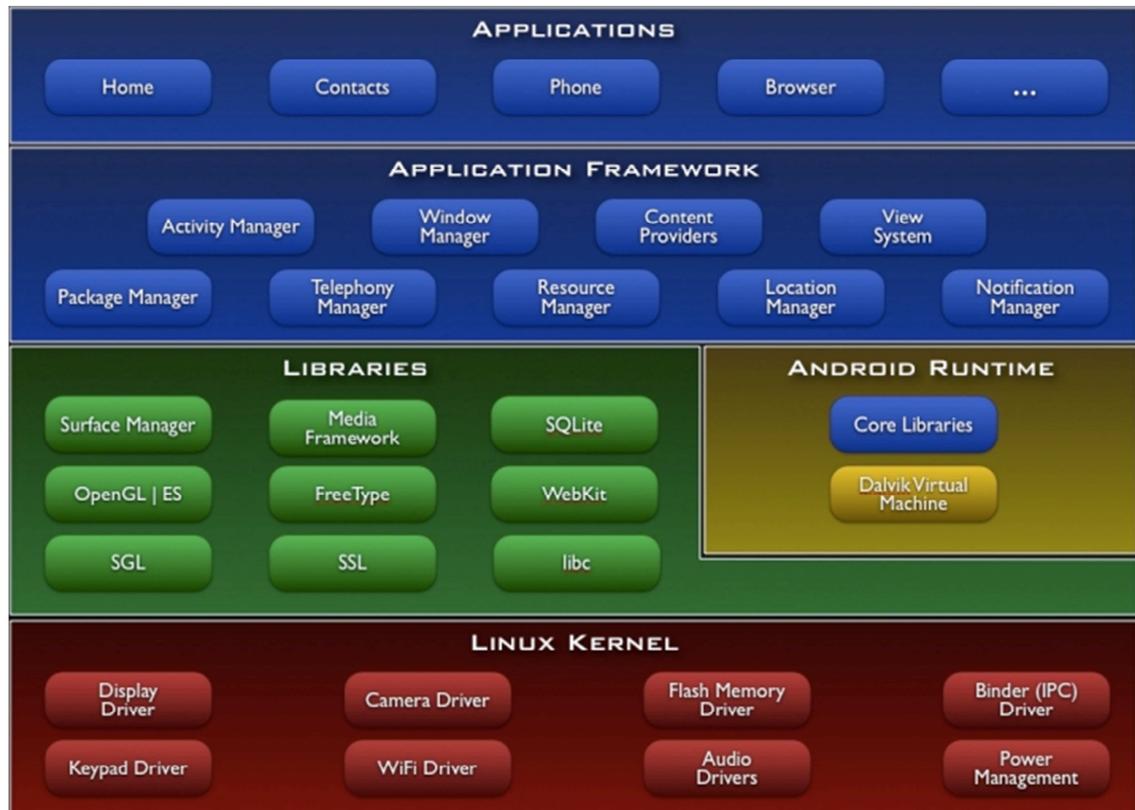
2.2 Android operativni sistem

Android operativni sistem je trenutno najrasprostranjeniji operativni sistem za mobilne telefone, zasnovan je na Linux kernelu i prilagođen je tako da se može koristiti na većini mobilnih uređaja, uključujući pored mobilnih telefona i tablet računare, laptop računare, netbook računare, smartbook računare, čitače elektronskih knjiga, pa čak i ručne satove.

Iako je Android Linux distribucija, on po nekim stvarima odstupa od većine standardnih Linux distribucija, npr. Android nema standardni X Window System ili standardni skup GNU biblioteka pa tako nije u mogućnosti da pokreće aplikacije razvijene za druge standardne Linux sisteme. Pokretanje samih aplikacija se, kod Android operativnog sistema, ne vrši direktno, već se aplikacije pokreću u okruženju odvojenom od ostatka sistema. Aplikacije dobijaju samo određeni deo sistemskih resursa, pa tako nemaju pristup delovima sistema koji su im nepotrebni. To donekle poboljšava sigurnost i stabilnost sistema, takođe pri instalaciji aplikacija korisnik dobija listu svih dozvola koje jedna aplikacija zahteva da bi se instalirala, što korisniku daje mogućnost da uoči potencijalno štetne aplikacije i obustavi njihovu instalaciju pre nego što dođe do oštećenja.

Sa tehničke strane Android predstavlja Linux operativni sistem razvijen za ARM (eng. Acorn RISC Machine) i x86 arhitekturu i sastoji se od modifikovanog monolitnog Linux kernela zaduženog za podršku fizičkoj arhitekturi i funkcija niskog nivoa, skupa biblioteka

zaduženih za dodatne podrške kao što su iscrtavanje grafike, podrška za dekodovanje video snimaka itd... Na višem nivou od biblioteka su systemske aplikacije neophodne za upotrebu sistema od strane korisnika i tu se nalaze, window manager, menadžer resursa, menadžer instalacionih paketa, kao i aplikacije zadužene za obavljanje osnovnih funkcija vezanih za mobilne telefone ili uređaj na kom je instaliran Android operativni sistem, na najvišem nivou se nalaze krajnje korisničke aplikacije, odnosno aplikacije koje direktno koristi korisnik. Arhitektura Android operativnog sistema može se prikazati na sledećoj slici.



Slika 1 – Arhitektura Android operativnog sistema

Arhitektura Android operativnog sistema predstavlja standardnu arhitekturu Linux sistema gde su segmenti sistema razdvojeni po nivoima na kojim rade. Za crtanje 3D grafike Android koristi biblioteku zasnovanu na OpenGL ES 2.0 specifikaciji, što ovom sistemu daje mnoge napredne grafičke sposobnosti. Android poseduje i ugrađenu podršku za multitasking.

Distribucija Android operativnog sistema počinje 5. novembra 2007., sa osnivanjem "Open Handset" alijanse, konzorcijuma sastavljenog od 71 proizvođača fizičke arhitekture, programske podrške, kao i telekom kompanija, posvećenih unapređenju otvorenih standarda

za mobilne uređaje. Google je napravio većinu koda za Android pod *Apache* licencom, slobodnu programsku podršku sa licencom otvorenog koda.

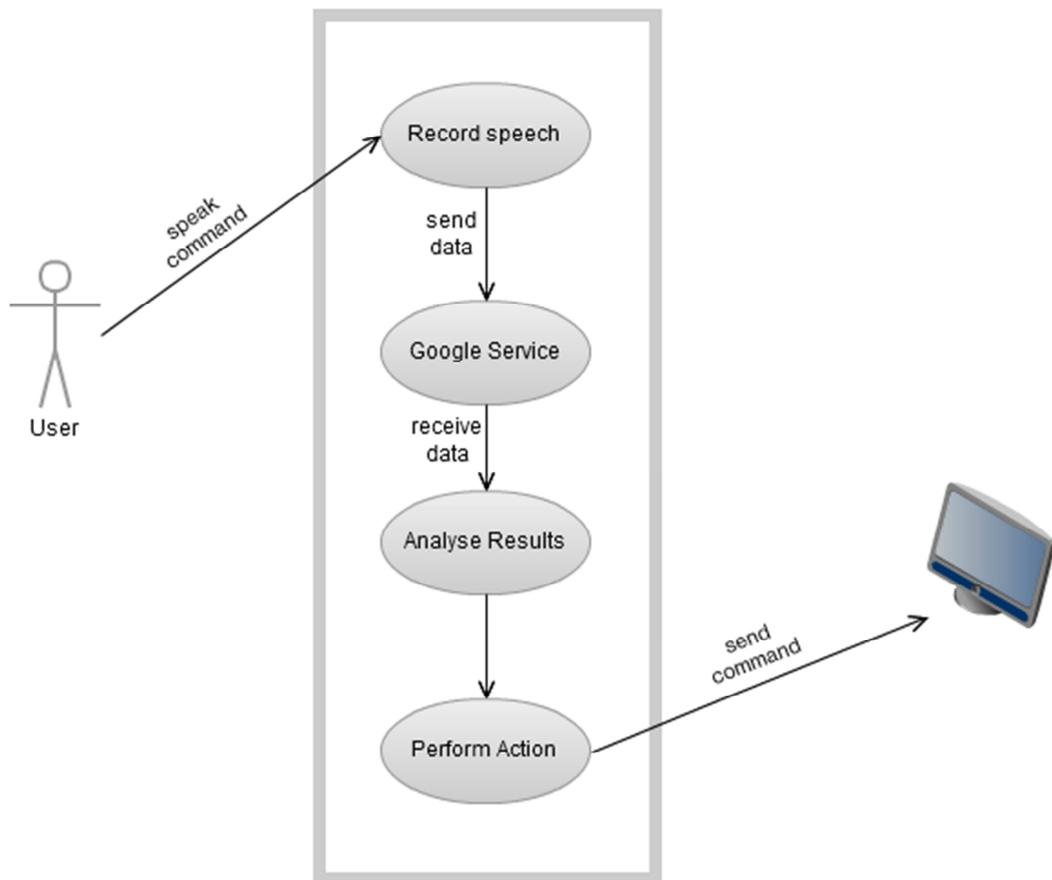
Programska podrška bazirana na Android operativnom sistemu se sastoji od Java aplikacija koje rade na Java baziranom, objektno orijentisanom aplikacijskom okviru povrh temeljnih Java biblioteka, koje rade na *Dalvik* virtualnoj mašini, čineći *JIT* (eng. Just-In-Time) kompilaciju. Android operativni sistem sadrži 12 miliona linija koda, uključujući 3 miliona linija XML-a, 2.8 miliona linija C-a, 2.1 miliona linija u Java, i 1.75 miliona linija u C++.

2.3 TCP/IP

U okviru ovog rada smo samo oponašali slanje komandi na drugi uređaj (koji je oponašao TV prijemnik), i veza koja se ostvaruje između klijenta koji koristi našu aplikaciju, i servera koji prima komande je TCP veza. **TCP/IP** protokol stek je skup protokola razvijen da omogući umreženim računarima da dele resurse putem mreže. Razvijen je od strane agencije DARPA u okviru ARPANET-a ranih 1970ih. U periodu od juna 1987. do juna 1998. više od 300 različitih proizvođača imalo je proizvode koji su podržavali TCP/IP protokole, a postojalo je na desetine hiljada mreža, različitih veličina i tipova, koji su ih koristili. Njihov broj se iz dana u dan povećava, što je najbolji primer značaja TCP/IP-a u računarskim telekomunikacijama.

3. Koncept rešenja

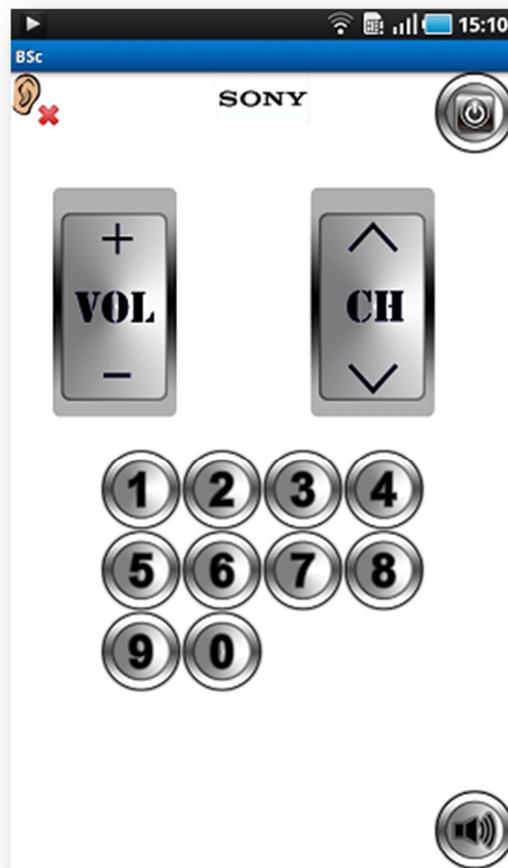
Tokom izrade ovog rada, rešenje je razvijano u tri pravca. Prva stvar koju je bilo potrebno uraditi jeste da se napravi sprega sa korisnikom, odnosno izgled aplikacije. Zatim je bilo potrebno da se implementira algoritam STT na način koji bi bio logičan za upravljanje daljinskim upravljačem pomoću glasovnih komandi. To podrazumeva mogućnost promene kanala, promena intenziteta zvuka, paljenje i gašenje uređaja. Korisnik može da koristi daljinski upravljač i kada je osluškivanje ugašeno, ali mora da bude povezan sa drugim uređajem (TV prijemnikom). Poslednji segment izrade predstavlja uspostava veze sa željenim uređajem. Osnovna zamisao načina rada i funkcionisanja ove aplikacije, može se videti na slici 2. U daljem delu teksta ćemo reći nešto više o gore navedenim stavkama.



Slika 2 – Tok slanja, prijema i obrade podataka

3.1 Sprega sa korisnikom

Cilj ove aplikacije je da, pored funkcionalnosti, ima i prikladan korisnički podsistem. Na samom početku izrade, izgledu aplikacije je posvećeno najviše vremena. Aplikacija je zamišljena tako da se prilikom njenog pokretanja otvori pozadina na kojoj su osnovna dugmad daljinskog upravljača. Kao i kod svake druge aplikacije sa Android operativnim sistemom, potrebno je obezbediti prikladan grafički podsistem. To podrazumeva da je aplikacija interaktivna, odnosno da sadrži pojedine efekte aktiviranja, deaktiviranja dugmeta, promene pozadine u odnosu na orijentaciju uređaja, otvaranja raznih opcija pritiskom na *menu* dugme itd., i oni su prikazani na slikama koje slede kroz ovaj dokument.



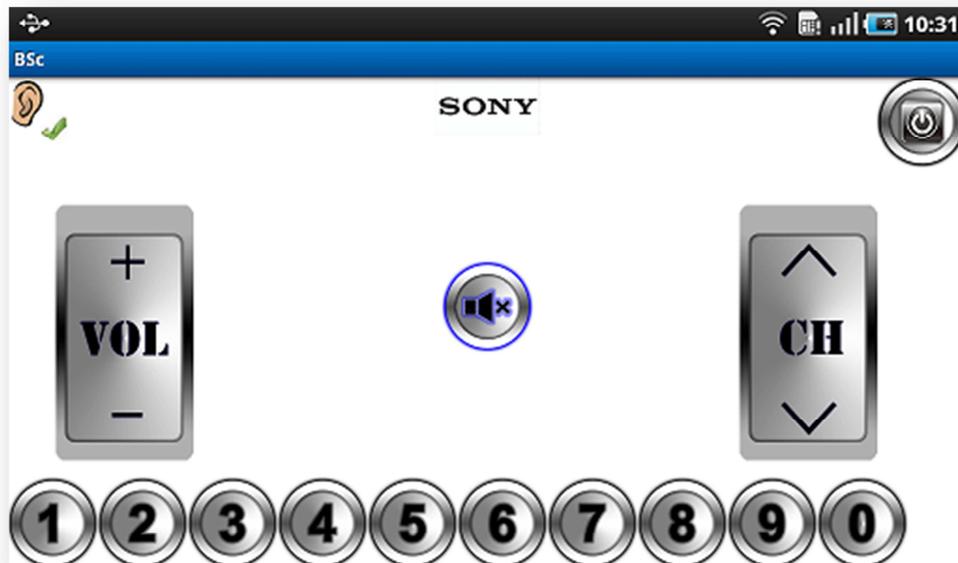
Slika 3 – Izgled glavnog prozora



Slika 4 – Stavke menija

3.2 Speech To Text algoritam

Kako je cilj ovog rada bilo upoznavanje sa *Speech To Text* tehnologijom, aplikacijom je pokrivena osnovna funkcionalnost daljinskog upravljača pomoću glasovnih komandi. Način funkcionisanja ove tehnologije je prost. Govor se snima pomoću mikrofona na korišćenom uređaju, i šalje na odgovarajuću internet adresu (eng.URL - Uniform Resource Locator). Tada Google servis obradi dobijene podatke u vidu audio datoteke, i vrati korisniku datoteku koja sadrži niz karaktera koje je servis prepoznao kao izgovorene. U ovoj aplikaciji osluškivanje glasa se aktivira pritiskom na specifično dugme, uho, i tada se komande snimaju sve dok korisnik ne završi. To vreme može da traje maksimalno petnaest sekundi, a ukoliko korisnik prestane da govori, automatski se prekida snimanje i snimljeni govor se šalje na servis. Ono što je bilo potrebno rešiti to je da se nakon slanja prvog snimljenog govora ne prestaje sa snimanjem komandi, već da uređaj stalno osluškuje i reaguje na samo odgovarajuće komande. Registrovane komande su „Volume up“, „Volume down“, „Channel up“, „Channel down“ (kao i komande Channel 0-9), „Mute“, „Turn off“, „Stop“.

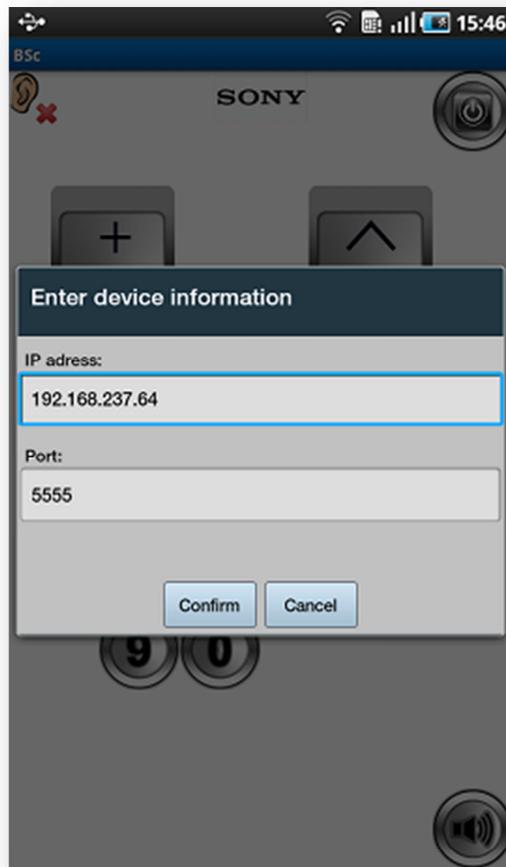


Slika 5 – Aktiviran STT u vodoravnom modu

Kao što se može videti na slici, pritiskom na dugme *uha* aktivirano je osluškivanje, a deaktivirati se može ili na isto dugme ili glasovnom komandom “Stop”. Više o STT algoritmu u programskom rešenju.

3.3 Uspostava veze

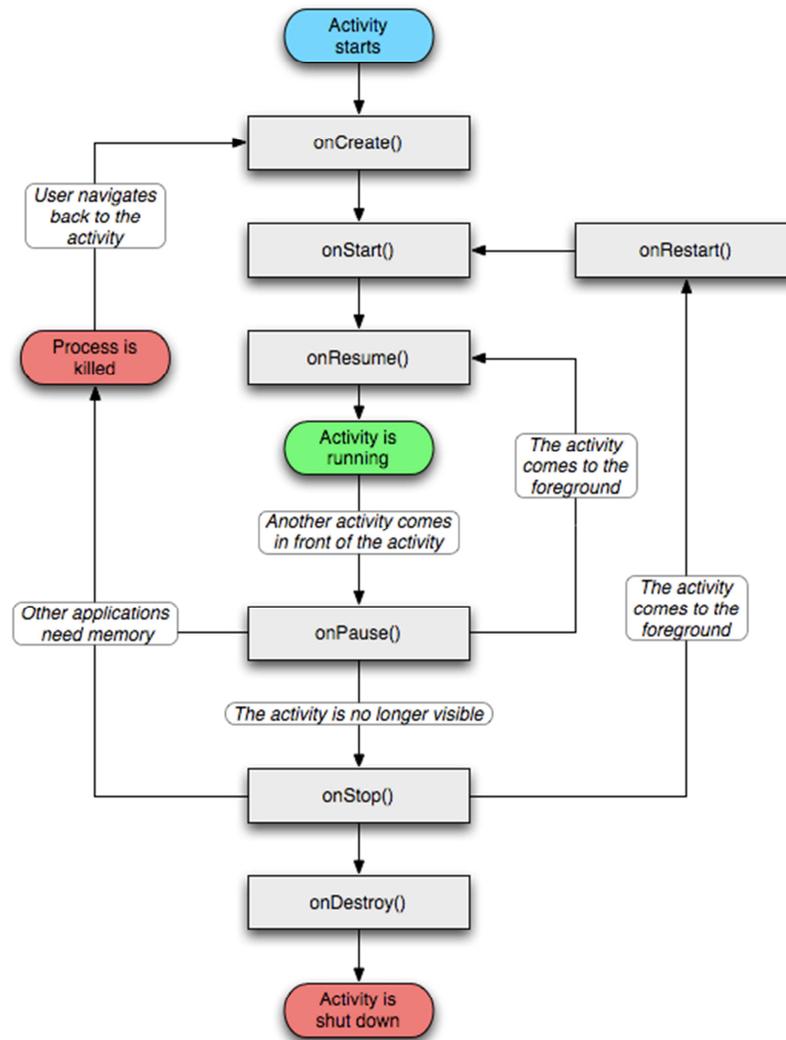
Veza između uređaja i TV prijemnika je klasična TCP klijent-server veza. Sve dok se korisnik uspešno ne poveže sa željenim uređajem, pritiskom na bilo koji od dugmadi biva obavešten da nije uspostavljena veza sa TV prijemnikom.



Slika 6 – Izgled prozora za uspostavu veze

4. Programsko rešenje

Za realizaciju ovog rada korišćen je programski jezik Java, *SpringSource* razvojno okruženje sa *ADT Plugin* i *Android SDK alatom*. Rešenje je zasnovano na korišćenju Android sistemskih klasa, nasleđivanjem, menjanjem postojećih i dodavanjem novih. Sam koncept programskog rešenja se može podeliti na tri modula gde je u svakom od njih realizovan odgovarajući algoritam, počevši od interakcije sa korisnikom, razmene podataka sa Google servisom, do slanja podataka ka željenom uređaju, bilo da je to TV prijemnik, android uređaj ili slično. Najvažniji programski element aplikacije je svakako *Activity* klasa, u daljem tekstu aktivnost. To je grafički element koji odgovara jednom ekranu. Svaki bitniji segment ove aplikacije je posebna aktivnost. Određeni podaci su potrebni u više različitih aktivnosti, te je bilo potrebno implementirati da se nekako te poruke prosleđuju od jedne ka drugoj. Te poruke se nazivaju *Intentovi*, i iniciraju aktiviranje aktivnosti. Sistem sam uparuje aktivnosti na osnovu zadate poruke. Životni vek aktivnosti prikazan je na sledećoj slici.

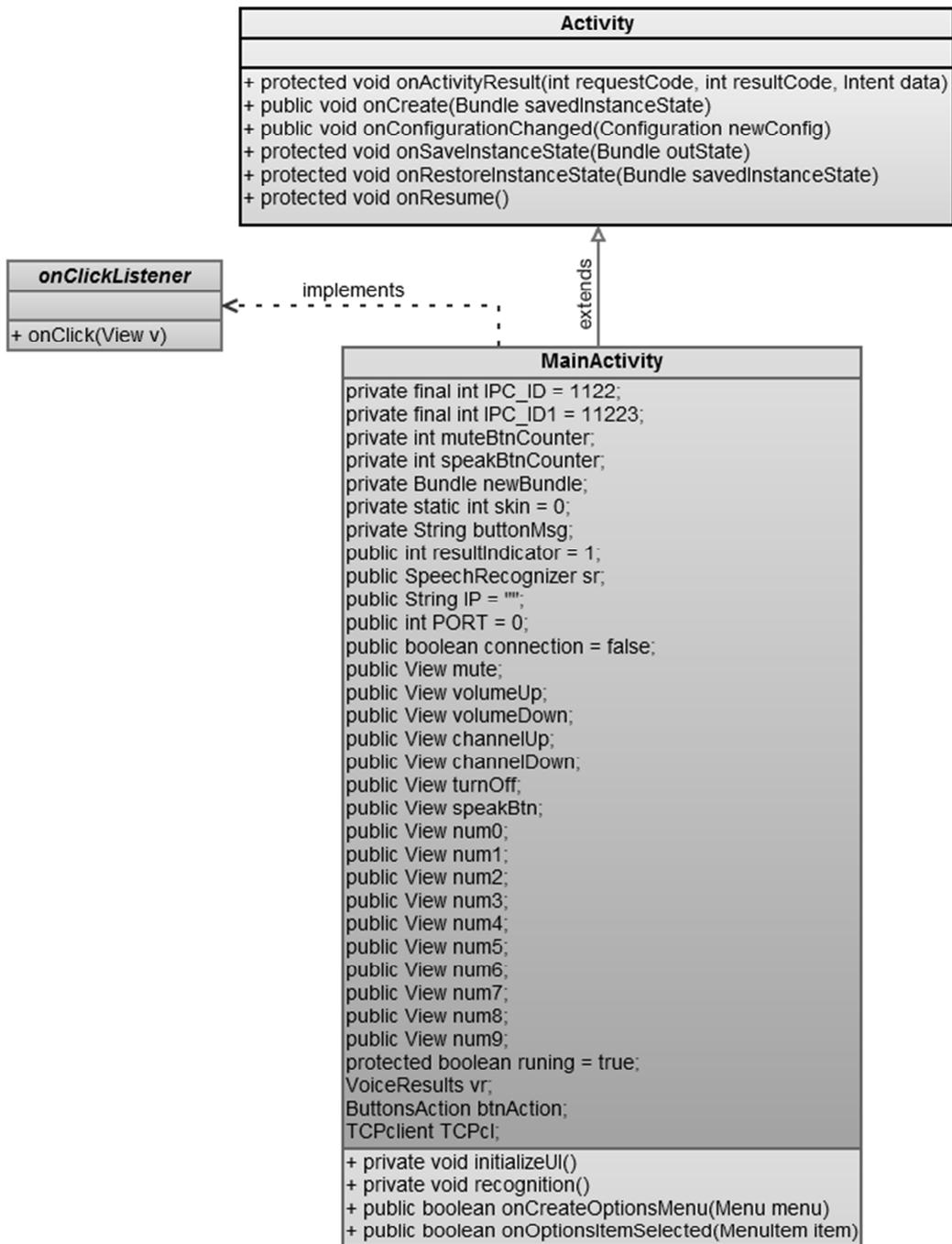


Slika 7 – Životni vek aktivnosti

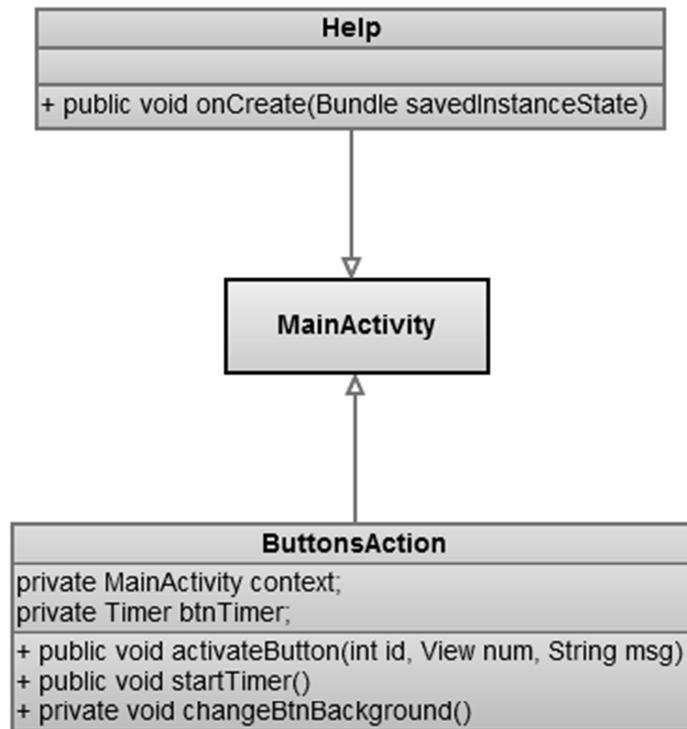
Sada ćemo prikazati programsku strukturu svakog modula, i ukratko objasniti funkcionalnost.

4.1 Modul daljinskog upravljača (paket Control)

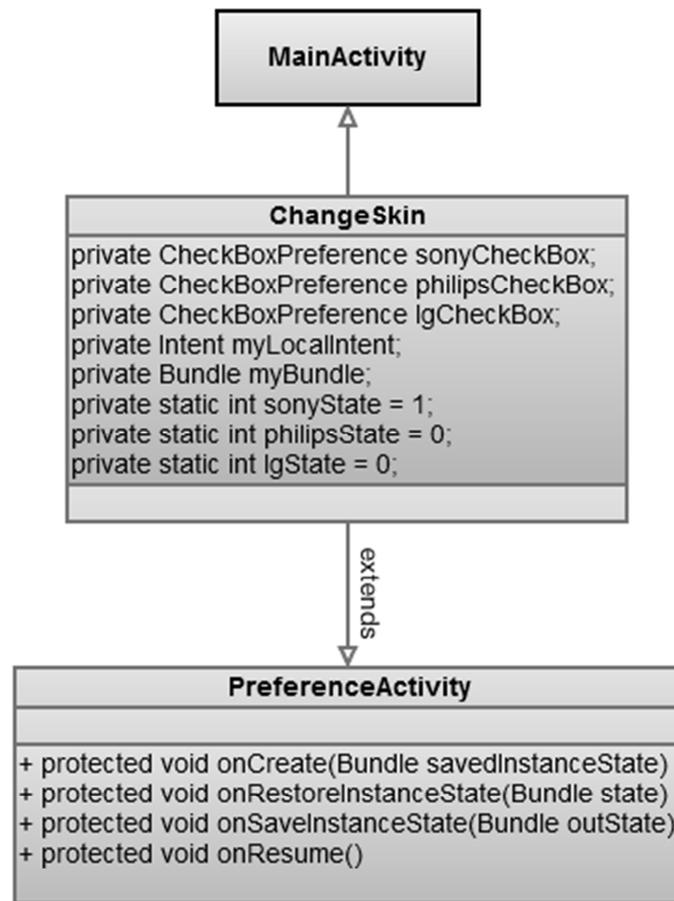
Ovaj programski paket sadrži klase čija se funkcionalnost odnosi na sam daljinski upravljač, odnosno na izgled aplikacije, kao i razne mogućnosti koje su dostupne, promena tipa upravljača za različite TV uređaje, upustvo za korišćenje, kao i uspostava veze sa željenim uređajem. Struktura ovog paketa može se prikazati pomoću sledećih dijagrama.



Slika 8 – Klasni dijagram paketa Control



Slika 9 – Klasni dijagram paketa Control

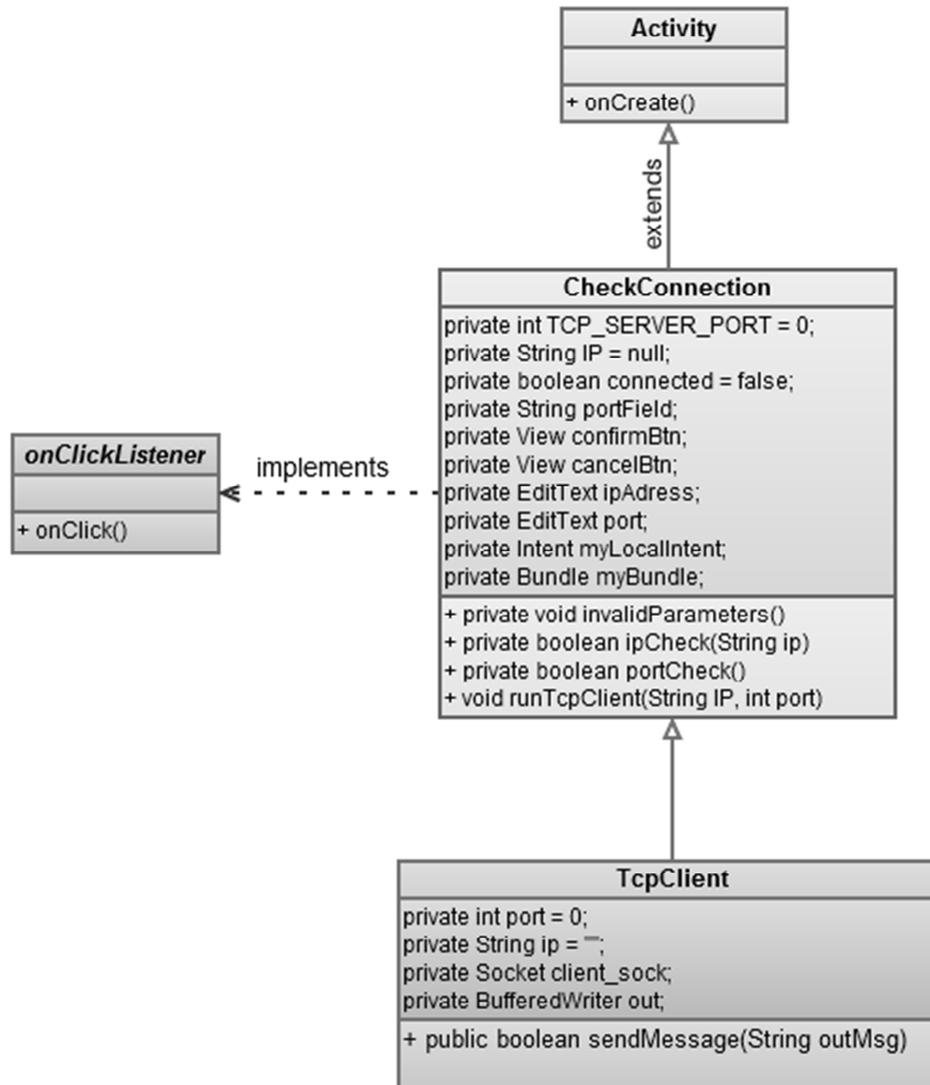


Slika 10 – Klasni dijagram paketa Control

Kao što se može primetiti na dijagramima, klasa *MainActivity* predstavlja srž samog programa. Prilikom pokretanja aplikacije, u okviru te klase se nalazi funkcija *onCreate()*, gde se definiše i sam izgled aplikacije. Raspored komponenti na ekranu se definiše pomoću .xml fajlova, tako da je svaki prozor u aplikaciji određena pozadina (engl. *layout*) definisan u .xml. Pozadine se razlikuju kada je uređaj u horizontalnom (eng. *portrait*) i u vertikalnom (eng. *landscape*) modu, da bi se izbegli problemi da pojedine komponente nestaju sa ekrana. Takođe, pozivaju se funkcije iz ostalih klasa, u zavisnosti koja su dugmad daljinskog upravljača pritisnuta. Tako na primer, pritiskom na dugme za “oslušivanje” glasovnih komandi, kreira se objekat klase *SpeechRecognizer*, nad kojim se izvršavaju funkcije potrebne za slanje i prijem podataka ka i od *Google* servisa. Stalno oslušivanje se vrši pomoću niti koja radi u pozadini, i čiji je zadatak da ponovo kreira objekat *SpeechRecognizer* svaki put

kad servis vrati rezultate, jer bi se inače osluškivanje deaktiviralo. Dobile podatke od strane Google servisa bilo je potrebno analizirati i obraditi na odgovarajući način i više reči o tome će biti u modelu za unapređenje algoritma. Pritiskom na dugme *menu*, otvaraju se kartice sa ponuđenim stavkama menija, od uspostave veze ka TV prijemnikom, promene “*skina*” daljinskog upravljača, upustva za korišćenje, i izlaska iz aplikacije. Pritiskom bilo koju od stavki, pokreće se nova aktivnost. Tako za pritisak na stavku “*establish connection*” otvara se prozor za unos *IP* adrese i prolaza (eng. port) uređaja na koji želimo da se povežemo. Pritiskom na stavku “*change skin*” otvara nam se prozor u kojem možemo da izaberemo tip TV prijemnika, odnosno da prilagodimo naš daljinski upravljač određenom TV prijemniku.

4.2 Modul uspostave veze (paket Connection)

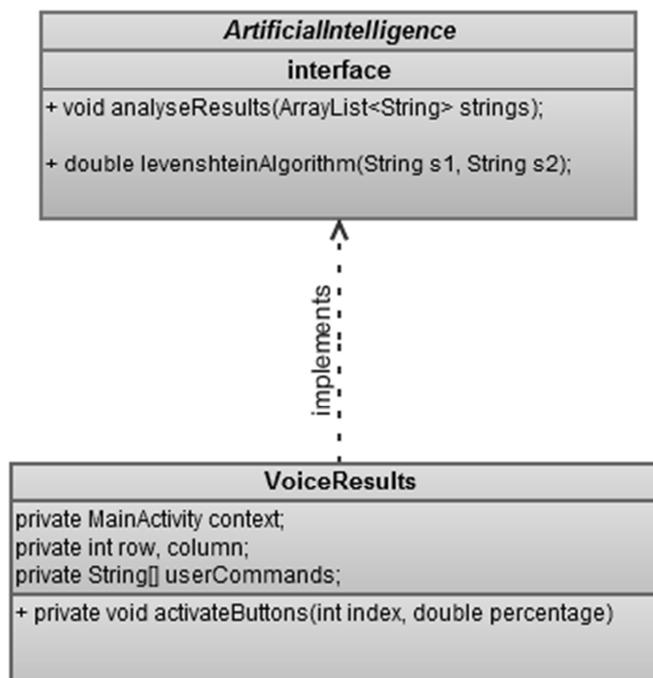


Slika 11 – Klasni dijagram paketa Connection

Paket *Connection* se sastoji iz dve klase, *TcpClient* i *CheckConnection*. U okviru ovih klasa su realizovane funkcije za uspostavljanje *TCP* veze između android uređaja (klijenta, osobe koja koristi aplikaciju) i TV prijemnika (servera). Jednom kad se uspostavi, veza je stalno otvorena, i svaka glasovna komanda i pritisak dugmeta na daljinskom upravljaču šalje odgovarajuće podatke na IP adresu servera, na odgovarajućem portu. Pri svakom unosu IP

adrese i porta, proverava se njihova validnost. Oni se unose u tekstualna polja, i prilikom pritiska na dugme *Confirm* zapamte se i obrade podaci koji su uneseni. Ukoliko uređaj sa zadatom IP adresom nije u okolini daljinskog upravljača, korisnik biva obavešten o tome. Sve dok se veza ne uspostavi korisnik će dobiti poruku “*No connection with other device!*” pritiskom na bilo koji od dugmadi.

4.3 Modul unapređenja algoritma (paket Artificial Intelligence)



Slika 12 – Klasni dijagram paketa Artificial Intelligence

Ovaj paket se odnosi na tretiranje rezultata dobijenih od strane *Google* servisa nakon što snimimo glasovnu komandu i pošaljemo servisu. Čest je slučaj da se rezultati u obliku niza karaktera ne poklapaju sa izgovorenim komandama. Tada te rezultate analiziramo i na osnovu različitih algoritama proveravamo da li rezultat koji je dobijen od servisa predstavlja neku od

glasovnih komandi. Naime, kada se izgovori glasovna komanda, snimljen govor se šalje na google servis, i kao povratna vrednost se dobijaju nizovi karaktera koji su, prema servisu, najpribližniji izrečenom. Tada, nizove karaktera šaljemo na obradu, i proveravamo da li u okviru tog niza koji se sastoji od četiri niza karaktera, postoji niz karaktera identičan registrovanim komandama za daljinski upravljač. Ukoliko ne postoji, primenjujemo našu programsku logiku. Algoritam za poboljšanje se zasniva na Levenštajnovom rastojanju (*Levenshtein distance*) [5]. Primer računanja Levenštajnovog rastojanja najlakše je uočiti iz sledećeg:

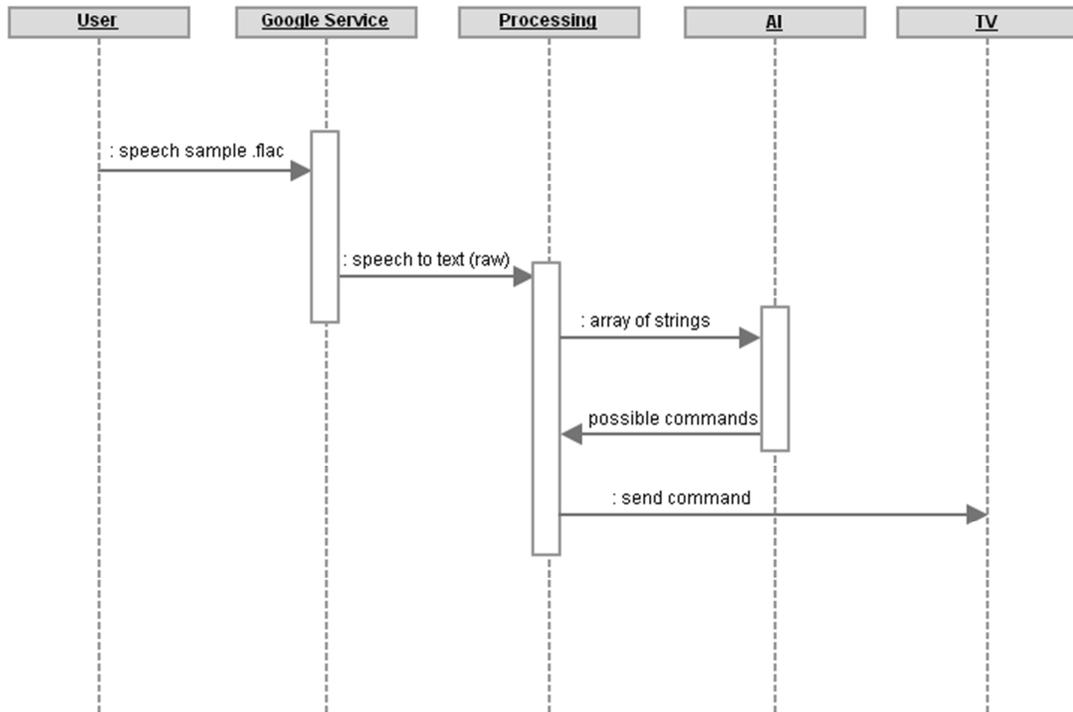
Levenštajново rastojanje reči "kitten" i "sitting" je 3, jer su potrebne najmanje tri operacije izmene da se jedna reč transformiše u drugu:

1. kitten → sitten (zamena 's' sa 'k')
2. sitten → sittin (zamena 'i' sa 'e')
3. sittin → sitting (umetanje 'g' na kraj)

Nizovi karaktera dobijenih od google servisa se šalje u funkciju *analyseResults* gde se svaki od njih pojedinačno propušta kroz Levenštajnov algoritam, i meri kolika je sličnost tog niza karaktera sa svakom od komandi. To znači da, pošto ima sedam memorisanih komandi, a četiri povratna niza karaktera, dobija se matrica od četiri vrste i sedam kolona. Kada se završi sa analizom niza karaktera, proverava se najveći član u matrici, odnosno kojoj koloni pripada. Tada, ukoliko je sličnost niza karaktera veća od određenog procenta, npr od 50% (mute – nuke), izvršava se data komanda.

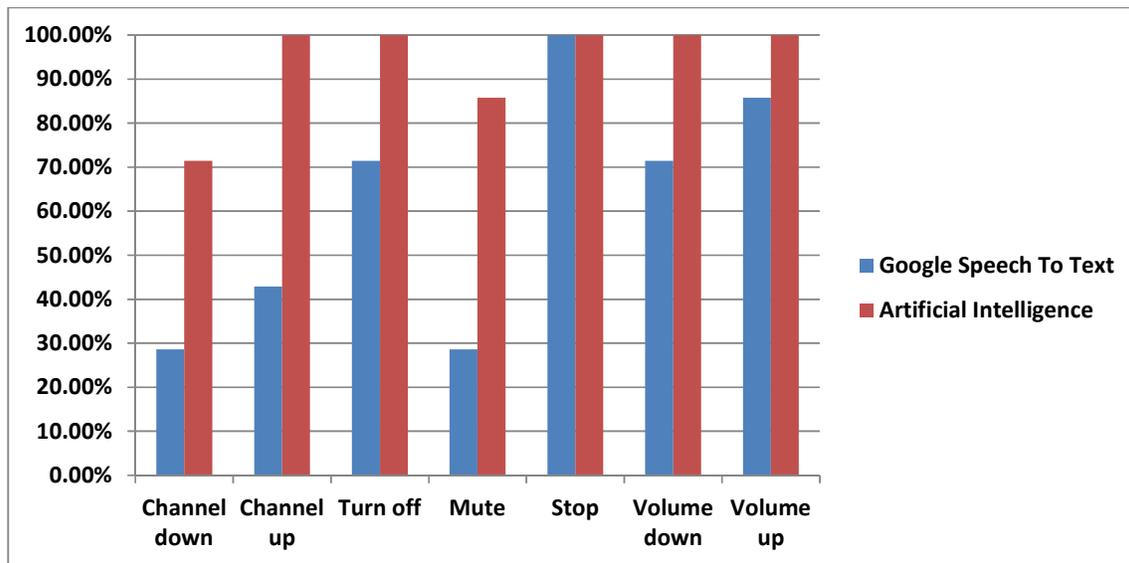
5. Rezultati

Često se upotreba *Speech to Text* tehnologije osporava zbog toga što preciznost varira u različitim slučajevima. Naime, što je duža rečenica koju korisnik izgovara, verovatnoća za pojavu greške raste. Konkretno u ovoj aplikaciji, izgovoreni brojevi između 0 i 9, prepoznati su bez greške, dok kod brojeva većih od 10 preciznost varira od osobe do osobe, odnosno načina izgovora. Način rada ove tehnologije je specifičan. Primera radi, izgovorenu rečenicu „The apple is red“, ukoliko Google servis ne prepozna, servis će sopstvenim algoritmom probati da prepozna gramatički najpribližniju rečenicu koju ima u svojoj bazi fraza, te sigurno neće vratiti rezultat „The apple is the“. To znači da će svaki rezultat koji servis vrati imati određen smisao. Upravo iz tog razloga, prilikom spontanog govora, servis će vratiti rezultate sa velikim brojem grešaka. Uzrok greške može biti i eho u prostoriji, ili određeni nivo buke koji dolazi spolja. S obzirom da je ova tehnologija za sad moguća na engleskom, francuskom, nemačkom, mandarinskom (kineskom) jeziku za očekivati je da korisnici sa govornih područja različitih od pomenutih jezika, ne izgovaraju komande pravilno, odnosno da verovatnoća za grešku bude veća. Upravo iz ovih razloga, u aplikaciji je napravljena programska logika koja će dodatno obraditi dobijene nizove karaktera, čiji je koncept i programsko rešenje opisano. Cilj je da i kad servis ne prepozna registrovanu komandu, obrada tih rezultata izvrši na što bolji način i na osnovu algoritma ustanovi koju komandu je potrebno aktivirati. U cilju ispitivanja efikasnosti razvijenog algoritma, snimljeno je nekoliko osoba koje različito izgovaraju komande, i na osnovu tih uzoraka izvršeno merenje tačnosti prepoznavanja komandi razvijenog algoritma naspram Google servisa. Način funkcionisanja prikazan je na sledećem dijagramu.



Slika 13 – Ispitivanje efikasnosti algoritma za unapređenje

Svi rezultati uzoraka dobijeni od strane servisa i razvijenog algoritma za unapređenje, su zapamćeni i upoređeni, te je dobijena statistika koja prikazuje uspešnost prepoznavanja glasovnih komandi.



Slika 14 – Rezultati prepoznavanja komandi

Kao što se može videti na grafiku uspešnost prepoznavanja komandi od strane Google servisa varira od reči do reči. Npr, za komandu “stop” se vidi da je svaki put izvršena bez potrebe da se analizira pomoću algoritma za unapređenje. Za komandu “mute”, koju zbog načina izgovora mnogi ne izgovore kako treba, servis je prepoznao na vrlo malom broju uzoraka. Obradom tih podataka i primenom razvijene programske logike, komanda je prepoznata u većem broju slučajeva. Posmatrajući ostale komande, vidi se da je nakon unapređenja algoritma, najslabiji procenat prepoznavanja za komandu “channel down” i iznosi oko 71%.

6. Zaključak

U ovom radu je opisano jedno rešenje korišćenja Google tehnologije pretvaranja govora u tekst. Osnovni koncept diplomskog rada, govorom upravljana aplikacija je uspešno realizovan. Uređaj stalno osluškuje komande i šalje ih u vidu niza karaktera ka uređaju koji se nalazi na serverskoj strani TCP veze. U zavisnosti od načina izgovora komandi koje su na engleskom jeziku aplikacija se različito ponaša, ali je procenat prepoznavanja komandi na visokom nivou. Da bi dobili precizniju statistiku potrebno je snimiti mnogo više uzoraka sa izgovorenim komandama. Tako bi imali jasniju sliku kako da algoritam bude unapređen, jer bi znali koje komande Google servis najmanje prepoznaje. Pokazano je da postojeći algoritam koji koristi Google servis nije dovoljno efikasan, jer bez algoritma za unapređenje korišćenje ovakvog tipa aplikacija nije praktično.

Izgled same aplikacije će se minimalnim izmenama poboljšati, bez izmena postojećeg koda, već zamenom postojećih resursa koji se odnose na grafičke stavke (slike, efekti).

Dalji razvoj ove aplikacije podrazumevao bi da se uređaj zapravo poveže sa TV prijemnikom. Da bi se taj cilj ostvario, potrebno je da se detaljnije analizira način povezivanja Android uređaja sa *smart* TV prijemnicima, podatke koje razmenjuju između itd...

7. Literatura

- [1] Vladimir Kovačević, Miroslav Popović: Sistemska programska podrška u realnom vremenu, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2002.
- [2] Sajt Android podrške za razvoj, *www.developer.android.com*, učitano 21.06.2012.
- [3] Sajt programske implementacije Levenshteinovog algoritma, *http://www.merriampark.com/ldjava.htm*, učitano 21.06.2012.
- [4] Sajt StackOverflow, programerski forum, *http://stackoverflow.com/*, učitano 21.06.2012.
- [5] Vladimir Levenshtein: Binary codes capable of correcting deletions, insertions, and reversals, 1966.