

SISTEM ZA SINHRONIZOVANU REPRODUKCIJU AUDIO SADRŽAJA GRUPAMA KORISNIKA**THE SYSTEM FOR SYNCHRONIZED PLAYBACK OF AUDIO CONTENT TO USER GROUPS**Marko Jevtović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu je predstavljen sistem za sinhronizovanu grupnu reprodukciju audio sadržaja. Aplikaciju čine server i klijent aplikacije. Server vrši audio striming ka klijentima preko RTSP protokola. Komunikacija između klijenata i servera se odvija i upotrebom REST i WebSocket standarda. Integracija pomenutih tehnologija je izvršena uz pomoć biblioteka otvorenog izvornog koda. Sprovedena je evaluacija ove aplikacije i performanse su upoređene sa sličnim rješenjima. Na osnovu toga su izvučena zaključna razmatranja ovog rada.

Ključne reči: sinhronizacija, reprodukcija, audio sadržaj, RTSP

Abstract – This paper presents the system for synchronized group playback of audio content. The application consists of the server application and the client application. The server performs audio streaming to the clients via RTSP protocol. Communication between clients and the server also takes place using the REST and WebSocket standards. The integration of these technologies is performed using the open source libraries. Evaluation of this application is conducted and performance is compared to similar solutions. Based on the evaluation results concluding considerations of this paper are drawn.

Keywords: synchronization, playback, audio content, RTSP

1. UVOD

Aplikacije koje rade sa multimedijalnim sadržajem u većini slučajeva podržavaju samo pojedinačnog korisnika i podređuju sve funkcionalnosti njemu. Aplikacija koja je opisana u ovom radu i njoj slične omogućavaju da u jednom trenutku bude prisutno više korisnika kojima je potrebno pružiti iste usluge – slično kao kod radio stanica, gdje korisnici koji slušaju istu stanicu slušaju i isti sadržaj u isto vrijeme. Osim toga, u rješenju predstavljenom u ovom radu korisnici mogu direktno da utiču na sadržaj koji se reprodukuje, pa i da oni sami budu autori pjesama i plejliste. Sinhronizacija reprodukcije u realnom vremenu predstavlja izazov jer je potrebno podržati više korisnika,

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Marko Marković, docent.

tačnije pružiti im isto iskustvo bez obzira u kom trenutku reprodukcije su se priključili. Rješavanje ovog problema stvara doživljaj da više osoba zajedno sluša muziku, sa različitih lokacija.

2. PREGLED SLIČNIH APLIKACIJA

Postojeća rješenja u ovom domenu uglavnom predstavljaju aplikacije za mobilne uređaje. Njima se postiže da više osoba sluša isti audio sadržaj istovremeno, dok neke od njih imaju podršku i za video.

JOBX [1] je desktop i mobilna aplikacija. Oslanja se na platformu Spotify, s obzirom da koristi njen API što je dobra strana budući da je u pitanju najpoznatiji muzički servis ali sa druge strane znači da ne postoji druga opcija za reprodukciju. Radi na principu muzičkih soba koje mogu biti javne ili privatne. Unutar sobe postoje i dodatne opcije kao što su pravljenje plejliste, dopisivanje korisnika, ocjenjivanje trenutne pjesme i druge.

SoundSeeder [2] je Android aplikacija koja reprodukuje zvuk na više uređaja povezanih na istu mrežu. Podržava HTTP stream, DLNA, UPnP, ali i lokalne fajlove. Takođe, postoji opcija da se sluša neka od preko 25 000 radio stanica širom svijeta. Ova aplikacija koristi jedan uređaj kao orkestrator, i sa njega emituje sadržaj prema drugim uređajima. Na taj način se stvara sistem u kojem se glavni uređaj ponaša kao virtuelni server, a ostali uređaji kao zvučnici. Putem glavnog uređaja je moguće kontrolisati reprodukciju ali i jačinu zvuka na svim ostalim. Ova aplikacija se može pokrenuti i na desktop platformama, ali samo u režimu zvučnika, dakle bez opcije da takav sistem ima ulogu orkestratora.

AmpMe [3] je desktop i mobilna aplikacija koja je kao kombinacija dvije prethodno navedene, s obzirom na to da omogućava reprodukciju putem Spotify servisa, lokalnih fajlova ali ima podršku i za YouTube. Samim tim, pored audia, dostupan je i video sadržaj. Moguće je pridružiti se sobama u blizini zavisno od lokacije uređaja, kao i drugim javno dostupnim sobama u čitavom svijetu. Postoji i opcija povezivanja naloga sa socijalnih mreža, kako bi se olakšalo pronalaženje drugih osoba, čije profile je moguće zapratiti i tako dobijati obavještenja kada neko od njih napravi sobu.

Rave [4] je aplikacija za više platformi namjenjena korisnicima koji žele da zajedno slušaju ili gledaju sadržaj. Podržava različite striming servise, od kojih su najpopularniji YouTube, Netflix i Vimeo. Moguće je

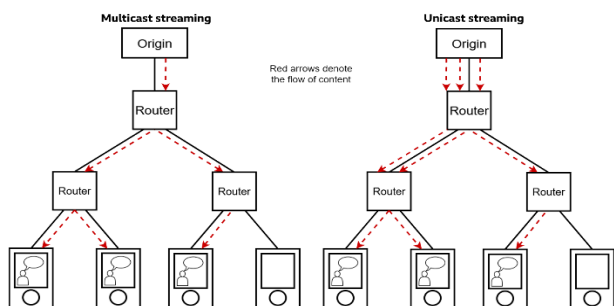
pristupiti i reprodukovati sadržaj sa DropBox ili Google Drive servisa za skladištenje. Korisnik koji napravi sobu, može da podesi da ona bude dostupna javno, samo korisnicima u blizini, samo prijateljima ili samo korisnicima koji uđu preko pozivnice tj. linka. Takođe, postoji opcija za način izbora sljedeće stavke za reprodukciju. Moguće je da korisnici o tome glasaju, može da bude automatski, ili da vlasnik sam izabere.

3. PREGLED KORIŠĆENIH TEHNOLOGIJA

RTP (Real Time Transport Protocol) [5] je transportni protokol dizajniran za saobraćaj u realnom vremenu, npr. za audio i video, prvi put objavljen 1996. godine. Obimno se koristi u sistemima za komunikaciju i zabavu koji uključuju multimedijalni striming, video pozive i konferencije, VoIP, itd. Oslanja se na UDP protokol, pa samim tim bolje toleriše manje gubitke paketa nego njihovo kašnjenje. Može da se koristi i sa TCP protokolom ali takvi slučajevi nisu česti budući da TCP mehanizmi za kontrolu grešaka mogu da uspore saobraćaj i utiču na vrijeme isporuke paketa. Može da se koristi zajedno sa RTCP protokolom.

RTCP (Real-time Transport Control Protocol) [6] je kontrolni protokol koji nadgleda strim i pruža povratne informacije o njegovom kvalitetu. Dok RTP prenosi pakete sa podacima, RTCP šalje kontrolne pakete učesnicima komunikacije. Oni obuhvataju transportne statistike i informacije kao što su brojači paketa, devijacija signala, vrijeme putovanja, i druge. Aplikacija može da koristi navedene informacije i na osnovu njih promijeni parametre strima ako je potrebno. RTCP ne obezbeđuje enkripciju ni autentikaciju, ali je to moguće implementirati upotrebom sigurnog RTCP protokola (SRTCP), odnosno, kad je u pitanju RTP, njemu ekvivalentnog sigurnog RTP (SRTP) protokola.

RTSP (Real Time Streaming Protocol) [7] je striming protokol čija je uloga da kontroliše audio ili video strim bez potrebe za preuzimanjem fajla. Podržava operacije kao što su: Options, Setup, Teardown, Play, Pause, Record, i druge. Pritom, klijent može prije slanja neke od funkcionalnih operacija da pošalje upit serveru kako bi utvrdio koje od njih su moguće. Koristi kombinaciju protokola kao što su TCP, UDP i RTP da bi uspostavio i održao sesiju između klijenta i servera, vršio prenos podataka, itd. Može da se upotrebljava za unicast striming gdje se prenos podataka odvija između jednog klijenta i servera, kao i za multicast striming gdje se paketi sa servera propagiraju do više klijenta, kao na slici 1.



Slika 1. Multicast i unicast striming [8]

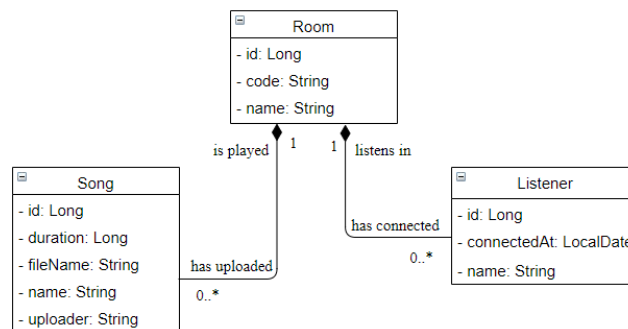
VLC (VideoLAN Client) [9] je program otvorenog koda za prikazivanje i emitovanje multimedijalnog sadržaja. Podržava veliki broj formata i protokola. Započet je kao studentski projekat 1996. godine i u početku je bio samo klijent za reprodukciju sadržaja. Danas posjeduje i značajne serverske mogućnosti kada je u pitanju striming u realnom vremenu. Formiran je od velikog broja modula, što čini lakšim dodavanje novih interfejsa, striming opcija i drugih funkcionalnosti. Podržava veliki broj operativnih sistema, na desktop i mobilnim uređajima.

VLCJ [10] je frejmwork otvorenog koda, koji omogućava integrisanje Java aplikacije sa VLC programom. Upotrebljiv je i za klijent i za server aplikacije. Nudi pristup većini VLC funkcionalnosti, od reprodukcije lokalnih fajlova do potpunog VOD servera. Enkapsulira rukovanje osnovnom bibliotekom, pri čemu štiti korisnika od nepravilne upotrebe komponenti, a i omogućuje mu proširivanje funkcionalnosti zavisno od potrebe. Npr. kod rukovanja asinhronim događajima, moguće je registrovati nove obrađivače koji će izvršiti korisničku logiku.

LibVLC [11] je biblioteka koja sadrži VLC funkcionalnosti podjeljene po modulima. Predstavlja jezgro i interfejs za multimedijalnu platformu na kojoj se VLC bazira. Postoji veliki broj verzija za desktop i mobilne platforme, pa je moguće integrisati je u različite aplikacije.

4. SPECIFIKACIJA SISTEMA

Na slici 2 je prikazan dijagram klasa koji prikazuje odnos između entiteta u implementiranom sistemu. Klasa *Room* je apstrakcija sobe i njenih podataka. Sadrži jake veze ka druge dvije klase, što je predstavljeno kompozicijom, s obzirom na to da instance ovih klasa ne mogu da postoje bez sobe. Klasa *Song* predstavlja pjesmu unutar plejliste sobe. Može da bude reprodukovana u samo jednoj sobi, dok u jednoj sobi može da bude nula ili više pjesama. Klasa *Listener* je reprezentacija korisnika, tj. slušaoca u sobi. U jednom trenutku može da bude prisutan u samo jednoj sobi, dok jedna soba ne mora da ima, ali i može, više konektovanih slušalaca.



Slika 2. Dijagram klasa

Svi entiteti imaju polje za jedinstveni identifikator. U osnovne podatke sobe spadaju njen kôd i ime. Slušalac takođe ima svoje ime ali i vrijeme konektovanja u sobu. Svaka pjesma pored imena ima trajanje i ime fajla u lokalnoj memoriji servera, ali i ime slušaoca koji ju je aploudovao. U ovu svrhu nije upotrebljena direktna veza na entitet *Listener* da bi se izbjegla ciklična veza između entiteta.

5. IMPLEMENTACIJA SISTEMA

U ovom poglavlju je predstavljena implementacija sistema za sinhronizovano slušanje audio fajlova na više uređaja. Čine ga klijentska i serverska aplikacija, tako da su one i njihovi moduli odvojeno objašnjeni.

5.1 Server

Ovaj dio sistema čini Spring Boot aplikacija, koja uz pomoć različitih tehnologija i protokola ispunjava zahtjeve navedene u specifikaciji sistema. Osim standardnih Spring biblioteka za veb aplikaciju, koristi još i VLCJ. Server je napisan u Kotlin programskom jeziku koji je dosta sličan programskom jeziku Java i takođe se može izvršavati pomoću JVM, ali sadrži brojna unaprijeđenja. Za čuvanje podataka se koristi relaciona baza podataka, u ovom slučaju MySQL.

REST interfejs servera omogućava neke od funkcionalnosti za klijente kao npr. dobijanje osnovnih informacija o muzičkoj sobi kojoj žele da se priključe, kreiranje nove sobe, upload fajla na server, itd.

Kada je u pitanju WebSocket komunikacija, u sistemu postoje dva tipa poruka koje se razmjenjuju. Prvi, koji predstavlja stvarne poruke korisnika u dijelu aplikacije za dopisivanje koje iniciraju klijenti i drugi koji označava događaje u realnom vremenu, koje inicira server. Za implementaciju je iskorišćena Spring WebSocket podrška koja se bazira na STOMP protokolu.

U cilju ispunjavanja svoje glavne funkcionalnosti, sistem je u mogućnosti da fajlove koje mu pošalju klijenti strimuje ka svima koji se nalaze u datoj sobi. Pošto je moguće da postoji više soba, to znači da u jednom trenutku više strimova može da bude aktivno. Implementacija to podržava na način da se svaki strim veže za sobu, tačnije za njen jedinstven kod pa je tako i njegov URL jedinstven. Takođe je potrebno voditi računa o plejlisti za svaku sobu i o objektima koji učestvuju u strimingu. Kao osnova za audio strim je upotrebljen VLC media player, tačnije njegova implementacija u Javi u okviru VLCJ paketa.

5.2 Klijent

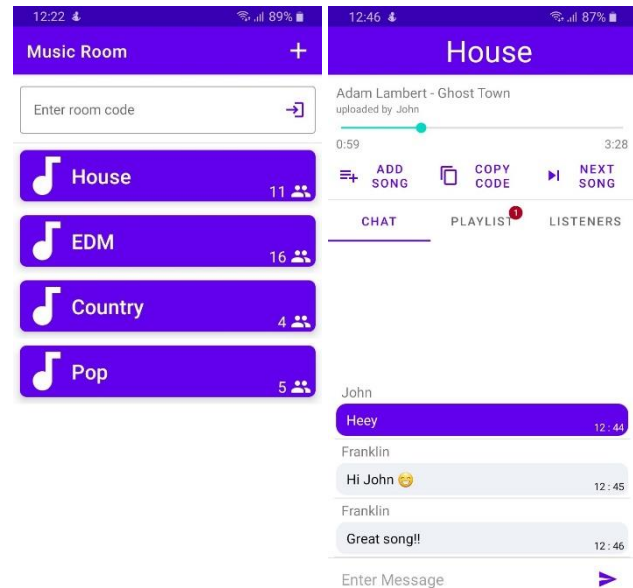
Klijentska strana je implementirana kao Android mobilna aplikacija. Takođe je, kao i server, napisana u Kotlin programskom jeziku. Neke od njegovih prednosti su povećana produktivnost, sigurniji i stabilniji kod, manje šablonskog koda, kompatibilnost sa Javom, i druge. Minimalna Android SDK verzija koju aplikacija podržava je 26, odnosno verzija sistema 8.0 Oreo što po trenutno aktuelnoj zvaničnoj statistici obuhvata oko 60% mobilnih uređaja.

Za potrebe poziva na REST API servera koristi se Retrofit klijent. On omogućava da se deklariraju metode unutar interfejsa sa posebnim anotacijama, koje pri pozivu šalju zahtjev na server. Moguće ih je izvršiti sinhrono, tj. u blokirajućem načinu tako da se čeka na odgovor, a moguće je i podesiti logiku koja će se izvršiti kada server odgovori, dakle asinhrono. Za podešavanje klijenta je potrebno napraviti Retrofit instancu koja ima definisan URL servera i koja za zadati interfejs kreira njegovu implementaciju.

Za WebSocket komunikaciju upotrebljena je STOMP Android biblioteka. Slično kao i za Retrofit i ovde je potrebno napraviti instancu klijenta koja će se koristiti za slanje i primanje poruka. Parametri koji joj se prosljeđuju su naziv klase koja služi za konekciju i URL servera.

Za konektovanje na RTSP strim i reprodukciju pjesama upotrebljena je LibVLC Android biblioteka. Komponente koje je potrebno napraviti su slične kao i za VLCJ na serveru, samo što ovde ne postoji pojam plejliste, već samo jedan strim na koji se klijent konektuje, a sadržaj koji dobije od njega reprodukuje preko zvučnika uređaja.

Na slici 3 se vidi kako izgleda početni ekran aplikacije i ekran kada se korisnik nalazi u sobi.



Slika 3. Izgled ekrana aplikacije

6. EVALUACIJA

Da bi se performanse ove i sličnih aplikacija uporedile u realnom okruženju, sprovedeno je testiranje na dva fizička Android uređaja, Samsung Galaxy A52 i Samsung Galaxy J6. Od navedenih aplikacija u drugom poglavlju, u obzir su uzete SoundSeeder i AmpMe. Evaluacija performansi Music Room i SoundSeeder aplikacija je sprovedena reprodukcijom fajlova sa uređaja, dok je za AmpMe u pitanju bio sadržaj sa YouTube platforme, pošto ista nije omogućavala korišćenje lokalnih fajlova. U nastavku je objašnjeno poređenje rada ovih aplikacija na navedenim uređajima u smislu sinhronizacije i kvaliteta reprodukcije sadržaja i količine generisanog mrežnog saobraćaja.

Sihronizacija je testirana analiziranjem zvuka koji uređaji emituju dok su konektovani na strim, prilikom reprodukcije kratkog audio snimka koji ima jednako raspoređene signale. Zvuk je zabilježen trećim uređajem, u ovom slučaju laptop računarnom koji posjeduje mikrofona, a uz pomoć Audacity softvera.

Proces je tekao tako da se prvo na jednom uređaju zvuk pojača a na drugom utiša, a onda se u toku snimka izvor zvuka obrne tako što se na prvom uređaju utiša a na drugom pojača.

Na kraju se na snimku mjeri rastojanje od posljednjeg signala sa prvog uređaja do prvog signala sa drugog

uređaja. To rastojanje se zatim upoređi sa fiksnim rastojanjem koje je poznato iz originalnog snimka. Da bi se reprodukcija smatrala sinhronizovanom, potrebno je da rastojanja budu približno podudarna. Što je manja razlika između njih, to je bolja sinhronizacija. Rezultati mjerenja su prikazani u tabeli 1.

Tabela 1. Odstupanje od sinhronizacije po aplikacijama

Aplikacija	Odstupanje u sinhronizaciji (ms)
Music Room	78,33
SoundSeeder	31,25
AmpMe	17,67

Usaglašenost kvaliteta reprodukcije je izmjerena poređenjem zvučnih talasa na audio izlazima iz uređaja. Dakle, uređaji se konektuju na strim i svaki uređaj snima svoj zvuk. Kada se strim završi, audio zapisi se uporede tako da se vidi koji uređaj je kakav zvuk reprodukovao u kom trenutku. Rezultati mjerenja su prikazani u tabeli 2.

Tabela 2. Razlika reprodukovano zvuka po aplikacijama

Aplikacija	Razlika u zvuku sa uređaja (%)
Music Room	7,71
SoundSeeder	10,19
AmpMe	5,22

Za mjerenje mrežnog saobraćaja koji aplikacija koristi na uređaju sprovedeni su testovi sa istom pjesmom u tri prethodno evaluirane aplikacije. Mjerenje protoka na mreži je specifično kada postoji više konektovanih uređaja, jer je teško izdvojiti samo podatke od interesa. Android aplikacije i servisi često u pozadini vrše određene operacije koje zahtjevaju pristup internetu, tako da je u ovom slučaju upotrebljena GlassWire aplikacija koja mjeri potrošnju za svaku aplikaciju posebno, u određenim vremenskim intervalima. Na taj način je obezbjeđeno da se prikupljaju podaci samo od aplikacija koje su predmet ove evaluacije, i to u momentu reprodukcije pjesme. Rezultati mjerenja su prikazani u tabeli 3.

Tabela 3. Količina mrežnog saobraćaja po aplikacijama

Aplikacija	Dolazni Saobraćaj (MB)	Odlazni Saobraćaj (MB)	Ukupno (MB)
Music Room	11,8	0,0122	11,8
SoundSeeder	35,8	1,1	36,8
AmpMe	11,2	2,3	13,5

7. ZAKLJUČAK

U ovom radu je predstavljen primjer implementacije sistema za sinhronizovano slušanje pjesama preko više uređaja koji ne moraju biti na istoj mreži. Prednosti ovakvog rješenja su što pruža brz i jednostavan pristup funkcionalnostima sistema. Nije potrebna registracija ili

prijavljanje unutar aplikacije, niti povezivanje na naloge eksternih sistema. Samim tim, ne postoje posebni preduslovi za upotrebu. U poređenju sa srodnim rješenjima, dobra strana ovog rješenja je što ne postoje ograničenja u vidu broja korisnika ili trajanja reprodukcije, kao i manja količina mrežnog saobraćaja uz kvalitet reprodukcije koji je konkurentan komercijalnim aplikacijama. Same performanse sistema zavise od hardverskih kapaciteta servera i mobilnih uređaja, ali i kvaliteta komunikacione mreže koju koriste.

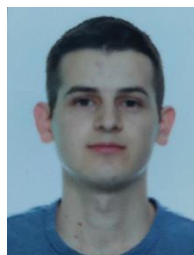
Kao mane se mogu pripisati relativno zastarjeli način reprodukcije muzike kao jedina opcija, odnosno nedostatak drugih izvora sadržaja. Danas se sve više koriste muzički servisi kao što su Spotify, YouTube, Deezer, SoundCloud i drugi, koji nude bogatu kolekciju pjesama uz dosta pratećih podataka i raznih funkcionalnosti.

Reprodukcija audio fajlova polako postaje prošlost uzimajući u obzir dostupnost interneta i sadržaja na njemu. Osim toga, neka od sličnih postojećih rješenja za razliku od ovog nude i raznovrsnost u vidu protokola i tehnologija preko kojih se odvija strim, kao i platformi za koje su aplikacije dostupne.

8. LITERATURA

- [1] JQBX, <https://www.jqbx.fm> (pristupljeno u avgustu 2021.)
- [2] SoundSeeder, <https://soundseeder.com> (pristupljeno u avgustu 2021.)
- [3] AmpMe, <https://www.ampme.com> (pristupljeno u avgustu 2021.)
- [4] Rave, <https://rave.io> (pristupljeno u avgustu 2021.)
- [5] What is RTP?, <https://www.3cx.com/pbx/rtp> (pristupljeno u avgustu 2021.)
- [6] What is RTCP?, <https://www.3cx.com/pbx/rtcp> (pristupljeno u avgustu 2021.)
- [7] Real Time Streaming Protocol, <https://www.avsilc.com/real-time-streaming-protocol> (pristupljeno u avgustu 2021.)
- [8] Multicast and unicast streaming, <https://www.bbc.co.uk/rd/blog/2019-09-forecaster-5g-mobile-interactive-content-experience> (pristupljeno u avgustu 2021.)
- [9] VLC media player, https://wiki.videolan.org/VLC_media_player (pristupljeno u avgustu 2021.)
- [10] VLCJ, <https://capricasoftware.co.uk/projects/vlcj> (pristupljeno u avgustu 2021.)
- [11] LibVLC, <https://www.videolan.org/vlc/libvlc.html> (pristupljeno u avgustu 2021.)

Kratka biografija:



Marko Jevtović rođen je u Istočnom Sarajevu 1996. god. Osnovne akademske studije završio je 2019. god. na Fakultetu tehničkih nauka na smjeru Računarstvo i automatika. Iste godine upisuje master akademske studije na programu Primenjene računarske nauke i informatika - Elektronsko poslovanje. kontakt: markojevtovic25@yahoo.com