

Mobilna aplikacija za terapiju tinitusa zvukom

Škrlj, Luka

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:099501>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-01-26**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni diplomski studij računarstva

Diplomski rad

**Mobilna aplikacija za terapiju tinitusa
zvukom**

Rijeka, rujan 2024.

Luka Škrlj
0069088022

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni diplomski studij računarstva

Diplomski rad

**Mobilna aplikacija za terapiju tinitusa
zvukom**

Mentor: izv. prof. dr. sc. Sandi Ljubić

Rijeka, rujan 2024.

Luka Škrlj
0069088022

Rijeka, 06.03.2024.

Zavod: Zavod za računarstvo
Predmet: Razvoj mobilnih aplikacija

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Luka Škrlj (0069088022)**
Studij: Sveučilišni diplomski studij računarstva (1400)
Modul: Programsko inženjerstvo (1441)

Zadatak: **Mobilna aplikacija za terapiju tinitusa zvukom / Mobile Application for Tinnitus Sound Therapy**

Opis zadatka:

Upoznati se s dijagnostičkim protokolima i specifičnostima terapije zvukom kod tinitusa. U skladu s relevantnim saznanjima, razviti mobilnu aplikaciju s tri glavna modula koji omogućavaju: (1) audiometrijsko testiranje s prikazom audiograma, (2) detekciju frekvencije i intenziteta tinitusa, te (3) transformaciju audio zapisa u format pogodan za terapiju tinitusa zvukom. Implementirati i poslužiteljski dio sustava koji čini osnovu za podršku upravljanja podacima o korisnicima, rezultatima mjerenja te standardiziranim upitnicima koji se administriraju putem mobilne aplikacije.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Mentor:
izv. prof. dr. sc. Sandi Ljubić

Komentor:
pred. dr. sc. Andrea Andrijašević

Predsjednik povjerenstva za
diplomski ispit:
prof. dr. sc. Miroslav Joler

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2024.

Ime Prezime

Zahvala

Zahvaljujem svom mentoru, izv. prof. dr. sc. Sandiju Ljubiću, na stručnom vodstvu, podršci i savjetima tijekom izrade ovog diplomskog rada. Također, zahvaljujem se svojoj komentorici, pred. dr. sc. Andrei Andrijašević, na korisnim savjetima i uloženom vremenu.

Posebnu zahvalu dugujem svojoj djevojci Petri S. na neiscrpnj podršci, razumijevanju i motivaciji tijekom cijelog ovog procesa. Njena prisutnost i ohrabrenje bili su neprocjenjivi u trenucima izazova.

Sadržaj

Popis slika	viii
1 Uvod	1
2 Osjetljivost sluha i šum u uhu	3
2.1 Audiometrija	4
2.1.1 Audiogram	5
2.2 Tinitus	8
2.3 Terapija tinitusa zvukom	10
3 Razvoj mobilne aplikacije za terapiju tinitusa zvukom	12
3.1 Arhitektura sustava	12
3.2 Tehnološki stog	14
3.2.1 Android	14
3.2.2 Kotlin	15
3.2.3 Firebase	16
3.2.4 Django	17
3.2.5 MATLAB	18
3.2.6 Angular	19
3.3 Poslužiteljski dio sustava	22

Sadržaj

3.3.1	Kolekcija <i>audiometry</i>	23
3.3.2	Kolekcija <i>questionnaire</i>	23
3.3.3	Kolekcija <i>tinnitus</i>	24
3.3.4	Kolekcija <i>user</i>	24
3.3.5	Obrada zvuka	25
3.4	Klijentski dio sustava	28
3.4.1	Razred <i>FrequencyGenerator</i>	30
3.5	Administracijsko sučelje u Angularu	34
4	Karakteristični slučajevi korištenja	36
4.1	Android aplikacija	36
4.1.1	Mjerenje osjetljivosti sluha	37
4.1.2	Detekcija odrednica tinitusa	39
4.1.3	Terapija zasnovana na filtriranim zvučnim zapisima	40
4.1.4	Upitnici za samoprocjenu učinaka tinitusa	42
4.2	Administracijsko web sučelje	44
4.2.1	Upravljanje korisnicima	45
4.2.2	Pregled podataka koje generiraju korisnici	45
5	Zaključak	48
	Bibliografija	49
	Pojmovnik	52
	Sažetak	53

Popis slika

2.1	Audiogram koji prikazuje normalnu razinu sluha u desnom uhu	7
2.2	Gubitak sluha u desnom uhu	8
3.1	Shema arhitekture sustava	14
4.1	Ekran za audiometriju	37
4.2	Ekran s prikazom rezultata audiometrije	39
4.3	Ekran za detekciju odrednica tinitusa	40
4.4	Ekran za prikaz pohranjenih rezultata s frekvencijom tinitusa	41
4.5	Ekran za reprodukciju audio datoteka	42
4.6	Ekran s listom ispunjenih upitnika	43
4.7	Ekran s pitanjem iz Tinnitus Handicap Inventory (THI)	44
4.8	Ekran s listom korisnika	45
4.9	Ekran s prikazom podataka o audiometriji	46
4.10	Ekran s prikazom podataka o tinitusu	47
4.11	Ekran s listom upitnika	47

Popis ispisa

3.3.1	Isječak koda u Python programskom jeziku	26
3.4.2	Isječak koda u Kotlin programskom jeziku	32

Poglavlje 1

Uvod

Tinitus je medicinsko stanje koje karakterizira percepcija zvuka u ušima ili glavi bez prisutnosti vanjskog izvora zvuka. Ovaj poremećaj može imati značajan utjecaj na kvalitetu života oboljelih, uključujući probleme sa spavanjem, koncentracijom, pa čak i mentalnim zdravljem. Iako točan uzrok tinitusa često ostaje nejasan, različite terapije, uključujući terapiju zvukom, pokazuju obećavajuće rezultate u smanjenju simptoma. Terapija zvukom koristi različite zvučne stimulacije kako bi maskirala ili reducirala percepciju tinitusa, pružajući olakšanje pacijentima.

Razvojem modernih tehnologija, mobilne aplikacije mogu postati podloge u pružanju zdravstvenih usluga, uključujući i terapije za tinitus. Mobilne aplikacije nude mogućnost personaliziranog pristupa terapiji, kontinuiranog praćenja simptoma te prilagodbu terapije prema specifičnim potrebama korisnika. Takav pristup omogućava bolju dostupnost terapije, smanjenje troškova te povećanu učinkovitost liječenja.

Ovaj diplomski rad ima za cilj razviti mobilnu aplikaciju za terapiju tinitusa zvukom, koja će se temeljiti na suvremenim dijagnostičkim protokolima i metodama liječenja tinitusa. Aplikacija obuhvaća tri glavna modula: (1) audiometrijsko testiranje s prikazom audiograma, (2) detekcija frekvencije i intenziteta tinitusa, te (3) transformacija audio zapisa u format pogodan za terapiju tinitusa zvukom. Implementacija ovih modula omogućit će sveobuhvatan pristup terapiji tinitusa, od dijagnoze do personaliziranog tretmana.

Osim mobilne aplikacije, rad uključuje i razvoj poslužiteljskog dijela sustava, koji

Poglavlje 1. Uvod

omogućava upravljanje podacima o korisnicima, prikaz rezultata mjerenja te administraciju standardiziranih upitnika. Ovaj sveobuhvatan pristup osigurava ne samo individualiziranu terapiju za svakog korisnika, već i sustav za kontinuirano praćenje i evaluaciju učinkovitosti terapije, što je ključno za daljnji razvoj i unaprjeđenje metoda liječenja tinitusa.

Poglavlje 2

Osjetljivost sluha i šum u uhu

Osjetljivost sluha je sposobnost uha da detektira zvukove u okruženju, a ovisi o mnogim fiziološkim i psihološkim faktorima. S druge strane, šum u uhu, ili tinitus, subjektivni je fenomen koji uključuje percepciju zvuka u ušima ili glavi bez prisutnosti vanjskog zvučnog podražaja. Ove dvije teme međusobno su povezane, budući da promjene u osjetljivosti sluha mogu utjecati na percepciju tinitusa i obrnuto.

Standardna mjera osjetljivosti sluha je prag sluha, koji se definira kao najniža razina zvuka koja može biti detektirana u više od 50% pokušaja. Uši ljudi najosjetljivije su na frekvencijama između donje granice od 1000 Hz do 5000 Hz, što odgovara području govora [1].

Osjetljivost sluha može varirati zbog različitih čimbenika:

- Dob: Slušna osjetljivost smanjuje se s dobi, osobito na višim frekvencijama, što se naziva prezbiakuzija [2];
- Buka: Duga izloženost glasnim zvukovima može uzrokovati oštećenje slušnih stanica u unutarnjem uhu, što vodi do gubitka sluha;
- Genetika: Genetski čimbenici mogu igrati ulogu u individualnoj osjetljivosti sluha, uključujući sklonost prezbiakuziji i osjetljivost na oštećenje bukom;
- Zdravstveni status: Zdravstveni problemi, poput dijabetesa ili kardiovaskularnih bolesti, mogu utjecati na slušnu osjetljivost kroz smanjenu cirkulaciju krvi u unutarnjem uhu.

Poglavlje 2. Osjetljivost sluha i šum u uhu

Osjetljivost sluha i šum u uhu (tinitus) ključni su elementi audiologije (grana otorinolaringologije čija je zadaća ispitivanje i liječenje postojećih smetnji sluha) koji zahtijevaju daljnja istraživanja zbog njihove složenosti i utjecaja na kvalitetu života. Razumijevanje njihovih uzroka, međusobne povezanosti, te mogućnosti liječenja može značajno poboljšati pristupe u liječenju i skrbi za osobe koje se suočavaju s problemima sluha i tinitusom.

Sljedeća potpoglavlja su usredotočena na audiometriju, standardnu metodu za procjenu osjetljivosti sluha, koja omogućava precizno mjerenje i identifikaciju različitih tipova oštećenja sluha (potpoglavlje 2.1). Zatim se detaljnije razmatra tinitus, analizirajući njegove uzroke i učinke na pojedince (potpoglavlje 2.2). Treće potpoglavlje opisuje različite pristupe terapiji tinitusa zvukom, uključujući kako određene vrste zvukova mogu pomoći u ublažavanju simptoma tinitusa i poboljšanju kvalitete života oboljelih (potpoglavlje 2.3).

2.1 Audiometrija

Gubitak sluha varira s godinama, pogađajući najmanje 25% ljudi starijih od 50 godina i više od 50% onih starijih od 80 godina [3]. Jedna od metoda otkrivanja problema sa sluhom je audiometrija.

Audiometrija je neinvazivna dijagnostička metoda kojom se ispituje sluh u svrhu otkrivanja oštećenja sluha, odnosno sposobnost osobe da čuje zvukove različite jačine i frekvencije. Osnovni alat u audiometriji je audiometar, uređaj koji generira tonove različitih frekvencija i intenziteta.

Audiometrijski testovi obuhvaćaju nekoliko metoda za procjenu sluha. Tonska audiometrija mjeri zračnu provodnost tj. prijenos zvuka zrakom kroz zvukovod i bubnjić do unutrašnjosti uha, te koštanu provodnost koja se odnosi na prijenos zvuka kroz kost do unutrašnjeg uha, koji zaobilazi zvukovod, bubnjić i slušne kosti. Osim tonske audiometrije postoji i govorna audiometrija koja procjenjuje sposobnost razumijevanja govora. U tonskoj audiometriji frekvencije mjerenja su u razinama spektra govora i sluha, od 250 Hz do 8000 Hz. Tijekom testiranja pacijent nosi slušalice i signalizira kad čuje zvuk, čime se određuje njegov prag sluha na različitim frekven-

Poglavlje 2. Osjetljivost sluha i šum u uhu

cijama.

Testiranje uglavnom započinje na uhu za koje ispitanik tvrdi da bolje čuje frekvencijom od 1000 Hz. Zatim se nastavlja redosljedom: 2000 Hz, 4000 Hz, 8000 Hz, 500 Hz i 250 Hz. Nakon završetka testiranja na prvom uhu, ponovno se testira na 1000 Hz. Ako se vrijednost praga sluha u ponovljenom testu ne razlikuje za više od 5 dB od originalne vrijednosti, kao konačna vrijednost se uzima ona osjetljivija. Ako je razlika veća od 5 dB, potrebno je pronaći uzrok odstupanja. U tom slučaju ispitaniku možda treba ponovno objasniti upute, a cijeli test za uho ponoviti. Neoobične vrijednosti rezultata će biti zabilježene na audiogramu. Ukoliko je potrebno i moguće, testiranje će biti provedeno i na međufrekvencijama od 750 Hz, 1500 Hz, 3000 Hz i 6000 Hz. Drugo uho će se testirati u istom redosljedu, s time da se ponovno testiranje na 1000 Hz obično ne provodi, osim ako u prvom uhu nisu otkrivene značajne varijacije [4].

Audiometrija je ključna za rano otkrivanje problema sa sluhom, što omogućava pravodobnu intervenciju i primjenu odgovarajućih terapijskih mjera. Ovaj postupak je posebno važan u provjeravanju sluha kod novorođenčadi, djece i starijih osoba, kao i u praćenju učinaka izloženosti buci.

2.1.1 Audiogram

Audiogram je grafički prikaz rezultata audiometrijskog testa koji pruža vizualnu reprezentaciju osjetljivosti sluha osobe na različitim frekvencijama i intenzitetima zvuka. Ovaj dijagram se koristi za dijagnosticiranje vrsta i stupnjeva gubitka sluha te je ključan alat u audiologiji. Na osi apscisa (x-osi) audiograma prikazane su frekvencije zvuka izražene u hertzima (Hz), koje obično variraju od 125 Hz do 8000 Hz, dok je na osi ordinata (y-osi) prikazan intenzitet zvuka u decibelima (dB), koji se kreće od -10 dB do 120 dB (od jako tihog do jako glasnog) [5]. Zvukovi niskih decibela uključuju šapat, dok zvukovi visokih decibela uključuju intenzivnije izvore, poput udaranja čekićem. Za nisko frekventne zvukove se mogu uzeti zvukovi poput grmljavine, a za visokofrekventne zvukove zvukovi koje, primjerice, proizvodi cvrčak ili zviždanje.

Intenzitet zvuka na y-osi se može grupirati u kategorije koje će označavati razinu

Poglavlje 2. Osjetljivost sluha i šum u uhu

gubitka sluha. Standardne kategorije su:

- Normalan sluh: mogu se čuti tihi zvukovi između 0 dB i 20 dB;
- Blagi gubitak sluha: gubitak sluha se događa između 21 dB i 40 dB;
- Umjereni gubitak sluha: gubitak sluha između 41 dB i 70 dB;
- Teški gubitak sluha: gubitak sluha između 71 dB i 95 dB;
- Duboki gubitak sluha: gubitak sluha iznad 95 dB [5].

Povećanjem frekvencije dobiva se krivulja na temelju čijeg oblika se može dobiti puna slika stanja sluha. Ako je krivulja:

- stabilna, bez velikih varijacija i nalazi se na vrhu audiograma, ukazivati će na normalan sluh;
- stabilna, nema velike varijacije i ne nalazi se na vrhu audiograma tj. nalazi se niže od 20 dB, ukazivati će na jednolik gubitak sluha;
- varijabilna s porastima i padovima, ukazivati će na gubitak sluha za određene frekvencije;
- padajuća s porastom frekvencije, ukazivati će na uobičajeni gubitak sluha povezanoga sa starenjem;
- u cijelosti niže na audiogramu, ukazivati će na ekstremniji gubitak sluha [6].

Ovo su osnovne smjernice za tumačenje audiograma, ali postoje i složeniji gubitci sluha koji mogu zahtijevati dodatne analize.

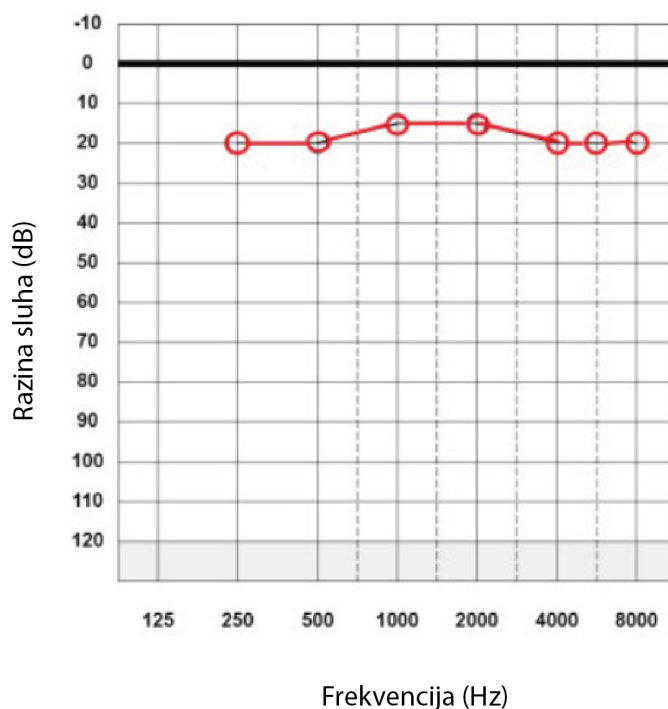
Na slici 2.1 [7] prikazan je audiogram normalne razine sluha za desno uho, dok je na slici audiograma 2.2 [7] prikazan progresivni gubitak sluha s porastom frekvencije.

U ovom diplomskom radu, na audiogramima će se koristiti sljedeći simboli za zračnu provodnost:

- ○ - desno uho;
- X - lijevo uho.

Postoje i drugi simboli koji se koriste u audiometriji, poput ispunjenih simbola koji ukazuju na primjenu maskiranja tijekom testiranja, te simboli za koštanu provod-

Poglavlje 2. Osjetljivost sluha i šum u uhu

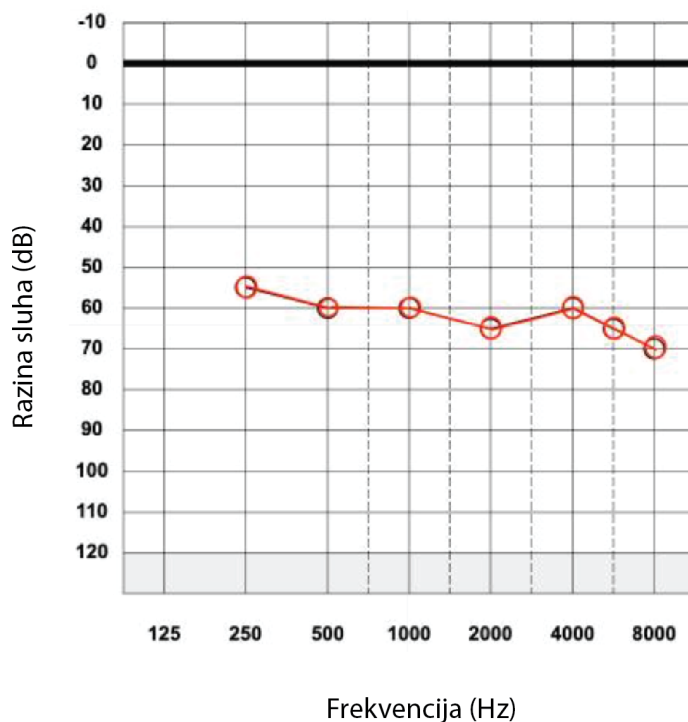


Slika 2.1 Audiogram koji prikazuje normalnu razinu sluha u desnom uhu

nost. Međutim, u ovom radu korišteni su samo gore navedeni simboli, s obzirom na to da se radi o mobilnoj aplikaciji pri čemu korisnici nemaju pristup opremi za testiranje koštane provodnosti. Stoga će se daljnja analiza fokusirati isključivo na zračnu provodnost.

Audiogram je ključan alat u dijagnostici različitih poremećaja sluha, uključujući i tinitusa. Iako tinitus može biti prisutan bez značajnog gubitka sluha, u mnogim slučajevima osobe koje pate od tinitusa imaju određeni stupanj gubitka sluha koji se može prikazati na audiogramu. Osobe s tinitusom često primjećuju da im je sluh oslabljen upravo u rasponu frekvencija gdje je tinitus najizraženiji. Na temelju tih podataka stručnjaci mogu prilagoditi terapiju, poput upotrebe slušnih pomagala koja ciljano pojačavaju određene frekvencije, čime se može smanjiti intenzitet tinitusa.

Poglavlje 2. Osjetljivost sluha i šum u uhu



Slika 2.2 Gubitak sluha u desnom uhu

2.2 Tinitus

Tinitus je percepcija zvuka u ušima ili glavi bez vanjskog zvučnog podražaja. Može se manifestirati kao zujanje, brujanje, šuštanje, ili čak u formi melodije. Postoje dvije glavne vrste tinitusa:

- Subjektivni tinitus: Najčešći oblik, koji samo osoba koja ga doživljava može čuti i pritom je uzrokovan promjenama u slušnim putovima;
- Objektivni tinitus: Rjeđi oblik, koji mogu čuti i druge osobe, a obično je uzrokovan stvarnim fizičkim zvukom, poput turbulentnog protoka krvi.

Uzroci tinitusa su brojni i mogu uključivati:

- Oštećenje sluha: Najčešći uzrok tinitusa je oštećenje slušnih stanica, često zbog izloženosti glasnim zvukovima ili starenju;

Poglavlje 2. Osjetljivost sluha i šum u uhu

- Infekcije i bolesti uha: Stanja poput *otitis media*, otoskleroze, i Meniereove bolesti mogu uzrokovati tinitus;
- Stres i anksioznost: Psihološki stres može pogoršati simptome tinitusa;
- Lijekovi: Neki lijekovi, kao što su aspirin i određeni antibiotici, mogu uzrokovati tinitus kao nuspojavu.

Dijagnoza tinitusa uključuje temeljitu anamnezu, audiometriju, i druge testove kako bi se isključili specifični uzroci. Liječenje tinitusa može biti izazovno, s obzirom na to da ne postoji univerzalni lijek.

Liječenje može uključivati:

- Liječenje osnovnog uzroka: Ako je tinitus posljedica stanja poput infekcije uha, liječenje tog stanja može smanjiti ili ukloniti simptome;
- Zvukovna terapija: Korištenje bijelog šuma ili specijaliziranih uređaja za tinitus može pomoći u maskiranju neugodnih zvukova;
- Kognitivno-bihevioralna terapija (KBT): KBT može pomoći pacijentima u razvijanju strategija za suočavanje s emocionalnim aspektima tinitusa.

Cilj liječenja je obično smanjenje percepcije tinitusa i poboljšanje kvalitete života pacijenta, a ne potpuni nestanak simptoma. Novija istraživanja usmjerena su na razumijevanje neuroplastičnosti i mogućnost primjene ciljanih neurostimulatora za modificiranje aktivnosti u auditivnom korteksu i smanjenje simptoma tinitusa.

Procjenjuje se da čak 10-15% svjetske populacije pati od tinitusa u nekom obliku, dok se u oko 1-2% slučajeva javljaju ozbiljni simptomi koji značajno narušavaju kvalitetu života [8]. Fiziološka osnova tinitusa još uvijek nije u potpunosti razjašnjena, ali se smatra da je rezultat abnormalne aktivnosti u auditivnom sustavu. Oštećenje unutarnjeg uha ili slušnog živca može uzrokovati nepravilne signale koji se prenose u mozak, gdje se pogrešno interpretiraju kao zvuk [9]. Nadalje, istraživanja su pokazala da tinitus može biti povezan s promjenama u središnjem živčanom sustavu, uključujući promjene u aktivaciji auditivnog korteksa i drugih dijelova mozga odgovornih za obradu zvuka [10].

Tinitus je često povezan s gubitkom sluha, posebno visokih frekvencija, što uka-

zuje na to da može biti uzrokovan smanjenom osjetljivošću na određene zvučne frekvencije. Kada određeni zvukovi više ne dopiru do mozga zbog oštećenja sluha, mozak može pokušati nadoknaditi tišinu stvaranjem vlastitih zvukova, što rezultira tinitusom. Ova teorija podržava činjenicu da su osobe koje su izložene glasnim zvukovima, poput glazbenika ili radnika u industriji, podložnije razvoju tinitusa [11].

Zaključno, tinitus je složeno stanje koje može imati različite uzroke i manifestacije. Iako je fiziološka osnova tinitusa još uvijek predmet istraživanja, napredak u razumijevanju neuroloških mehanizama koji stoje iza ovog fenomena pruža nadu za razvoj učinkovitijih terapija u budućnosti.

2.3 Terapija tinitusa zvukom

Terapija zvukom jedna je od najčešće korištenih metoda za ublažavanje simptoma tinitusa. Temeljna ideja ove terapije jest korištenje specifičnih zvukova ili glazbe kako bi se maskirali ili smanjili neugodni zvukovi tinitusa. U kontekstu terapije tinitusa, zvukovi ili glazba se prilagođavaju individualnim potrebama pacijenta, uzimajući u obzir frekvenciju i intenzitet tinitusa te osobne preferencije pacijenta. Korištenjem ove metode, pacijent može postići značajno smanjenje percepcije tinitusa, a time i poboljšanje kvalitete života.

Jedan od glavnih ciljeva terapije glazbom jest ometanje moždane percepcije tinitusa, stvaranjem ugodnih i smirujućih zvukovnih okruženja. Istraživanja pokazuju da personalizirana glazbena terapija može pomoći u reprogramiranju neuronskih putova u mozgu, smanjujući percepciju tinitusa. Konkretno, odabir glazbe koja se kreće unutar ili izvan frekvencijskog raspona tinitusa može modulirati aktivnost slušnih područja mozga, što rezultira smanjenom percepcijom neugodnih zvukova.

Individualizirana glazbena terapija može biti prilagođena tako da uključuje frekvencije koje su najslabije onima koje pacijent doživljava kao tinitus. Takva prilagodba omogućava ciljanje specifičnih područja u mozgu koja su preaktivna zbog tinitusa, čime se smanjuje njihova osjetljivost i potiče opuštanje. Istraživanja ukazuju na to da ovakva terapija ne samo da može smanjiti intenzitet tinitusa, već može i poboljšati emocionalno stanje pacijenta, smanjujući anksioznost i stres povezan s

Poglavlje 2. Osjetljivost sluha i šum u uhu

tinitusom.

Terapija zvukom, osobito glazbom, nudi obećavajuću opciju za smanjenje simptoma tinitusa. Ovaj pristup može se integrirati u svakodnevni život pacijenata, nudeći im alate za kontinuirano upravljanje simptomima. Daljnje istraživanje i razvoj ove terapijske metode mogu pridonijeti razumijevanju tinitusa i otvaranju novih putova za učinkovito liječenje ovog složenog stanja [12][13].

Poglavlje 3

Razvoj mobilne aplikacije za terapiju tinitusa zvukom

Razvoj mobilne aplikacije za terapiju tinitusa zvukom obuhvaća nekoliko ključnih aspekata koji osiguravaju njezinu funkcionalnost i učinkovitost. Potpoglavlje "Arhitektura sustava" pruža detaljan pregled strukture aplikacije i načina na koji različite komponente međusobno djeluju. Arhitektura je osmišljena na način kako bi omogućila integraciju klijentskog i poslužiteljskog dijela sustava, osiguravajući pritom pouzdanu obradu podataka i responzivno korisničko sučelje.

Potpoglavlje "Tehnološki stog" detaljno opisuje specifične tehnologije korištene za izradu i integraciju svih komponenti sustava. Ovdje se ulazi u tehničke detalje svakog dijela sustava, od klijentske aplikacije i njezinih razvojnih alata do poslužiteljskog okruženja i njegovih funkcionalnosti. Poznavanje i razumijevanje ovih tehnologija ključno je za daljnji razvoj, održavanje i nadogradnju sustava, a samim time i za poboljšanje terapijskih učinaka aplikacije na korisnike.

3.1 Arhitektura sustava

Sustav je dizajniran s više povezanih tehnologija koje zajedno čine cjelovitu aplikaciju. Za razvoj mobilne aplikacije korišten je Android operacijski sustav, pri čemu su korisničko sučelje i povezana interakcija definirani pomoću Extensible Markup Lan-

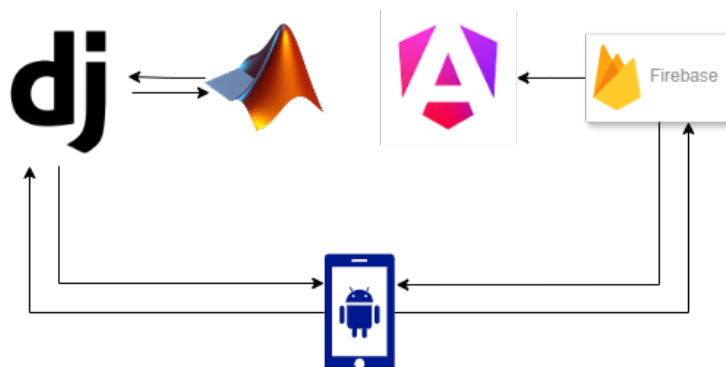
Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

guage (XML) predložaka, a poslovna logika je implementirana u programskom jeziku Kotlin. Ova mobilna aplikacija omogućava korisnicima dohvat i pohranu podataka na platformu Firebase, koja, osim baze podataka, omogućava i sustav autentikacije.

Podaci spremljeni na Firebase mogu se pregledati i upravljati njima putem administracijskog sučelja izrađenog u Angularu, radnom okviru zasnovanome na programskom jeziku TypeScript koji omogućava izradu responzivnih web aplikacija. Ovo sučelje omogućava administratorima pregled i upravljanje korisničkim podacima.

Za obradu zvuka, Android aplikacija koristi OkHttp knjižnicu za slanje Hypertext Transfer Protocol (HTTP) zahtjeva prema poslužitelju koji je sagrađen na Django razvojnom okviru temeljenom na Python programskom jeziku. Django poslužitelj prima zahtjeve za obradu zvučnih datoteka te ih prosljeđuje MATLAB okruženju putem MATLAB Engine Application Programming Interface (API), omogućavajući naprednu analizu i obradu zvuka. Nakon što MATLAB završi obradu zvučne datoteke, rezultati se vraćaju natrag na Django poslužitelj.

Kada je obrada završena, Django poslužitelj vraća odgovor Android aplikaciji, a obrađena zvučna datoteka preuzima se na korisnički uređaj. Ovaj proces omogućava korisnicima jednostavan i efikasan način obrade zvuka putem mobilne aplikacije. Takva arhitektura sustava osigurava skalabilnost, fleksibilnost i održivost aplikacije u budućem razvoju i nadogradnjama. Shema sustava je prikazana na slici 3.1 gdje strelice ukazuju na komunikaciju između različitih dijelova sustava i tokove podataka. Na dnu sheme je Android uređaj, a na vrhu su redom prikazane tehnologije, odnosno sustavi Django, MATLAB, Angular i Firebase.



Slika 3.1 Shema arhitekture sustava

3.2 Tehnološki stog

U razvoju suvremenih programskih rješenja, izbor odgovarajućeg tehnološkog stoga od ključne je važnosti za uspješnu implementaciju i održavanje aplikacije. Tehnološki stog predstavlja skup tehnologija, alata, programskih jezika i okvira koji se koriste za razvoj, testiranje, implementaciju i održavanje programskog sustava. U ovom diplomskom radu odabran je tehnološki stog koji uključuje različite tehnologije kako bi se osiguralo da su svi aspekti aplikacije adekvatno pokriveni i integrirani.

Svaka od tehnologija navedenih u arhitekturi sustava bit će detaljno obrađena zajedno sa svojim pripadajućim programskim jezicima i knjižnicama kroz zasebne podulomke, u kojima će se istražiti njihova uloga, prednosti i specifična upotreba u kontekstu razvijanog sustava.

3.2.1 Android

Android se temelji na Linux jezgri te je otvorenog koda, što omogućava široku prilagodbu i modifikaciju od strane razvojnog programera. Kao operacijski sustav, Android pokreće milijarde uređaja diljem svijeta, uključujući pametne telefone, tablete, pametne satove i druge povezane uređaje. Za razvojne programere, Android predstavlja bogatu platformu s brojnim mogućnostima, ali i izazovima.

Razvoj Android aplikacija primarno se odvija u integriranom razvojnom okru-

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

ženju (IDE) poznatim pod imenom Android Studio, koje je službeno podržano od strane Googlea. Android Studio pruža alate za razvoj, testiranje, detekciju i otkrivanje pogrešaka i optimizaciju Android aplikacija. Pored standardnog Java programskog jezika, podržava se i Kotlin, jezik koji je postao službeni jezik za razvoj Android aplikacija.

Uz Android Studio, programeri koriste Android Software Development Kit (SDK), koji uključuje osnovne alate i API-je potrebne za razvoj aplikacija. SDK nudi bogat set knjižnica i alata za rad s raznim sklopovskim i programskim komponentama uređaja, uključujući senzore, kamere, Globalni položajni sustav (GPS) i mrežne komunikacije.

Android aplikacije temelje se na modularnoj arhitekturi koja uključuje aktivnosti (engl. activities), fragmente (engl. fragments), servise (engl. services) i prijemnike emitiranja (engl. broadcast receivers). Aktivnosti predstavljaju pojedinačne ekrane s kojima korisnik interaktivno komunicira. Fragmenti omogućavaju modularnost unutar aktivnosti, uz ponovnu upotrebu korisničkih sučelja u različitim dijelovima aplikacije [14].

3.2.2 Kotlin

Kotlin je suvremeni programski jezik koji je u kratkom vremenu stekao veliku popularnost, posebno među Android razvojnim programerima. Kotlin se temelji na Java Virtual Machine (JVM), što omogućava njegovu kompatibilnost s postojećim Java kodom, ali istovremeno nudi brojne prednosti.

Jedna od ključnih prednosti Kotlina je njegova jednostavnost i čitljivost. Kotlin sintaksa je koncizna, a mogućnosti kao što su inferiranje tipova (engl. type inference) i pametno ukalupljanje (engl. smart casts) smanjuju količinu koda potrebnog za obavljanje uobičajenih zadataka. Time se značajno smanjuje mogućnost pojave pogrešaka, a kod postaje čitljiviji i lakši za održavanje.

Kotlin također nudi ugrađenu sigurnost od `null` vrijednosti (engl. null safety), čime se izbjegavaju uobičajeni problemi s kojima se programeri susreću u Javi, poput `NullPointerException` iznimke. U Kotlinu, varijable prema zadanim postav-

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

kama ne mogu sadržavati `null` vrijednosti, osim ako nisu izričito deklarirane kao `nullable`, što povećava pouzdanost koda.

Pored toga, Kotlin podržava korutine (engl. *coroutines*), koje omogućavaju jednostavno upravljanje asinkronim operacijama. Kroz korutine, moguće je pisati kod koji izgleda sinkrono, ali se izvršava asinkrono, što znatno pojednostavljuje rad s dugotrajnim operacijama kao što su mrežni pozivi ili rad s bazama podataka [15].

3.2.3 Firebase

Firebase je platforma za razvoj mobilnih i web aplikacija. Svrha platforme je omogućiti brzu i jednostavnu izradu aplikacija uz smanjenje opterećenja koje dolazi s razvojem poslužitelja i upravljanja infrastrukturom. Firebase nudi niz usluga, uključujući bazu podataka, autentikaciju, posluživanje stranica, analitiku, pohranu datoteka i još mnogo toga. Među najvažnijim komponentama Firebasa su Firestore i autentikacija, koje su ključne za pohranu podataka i sigurnost.

Kombinacijom Firestorea i Firebase autentikacije, mogu se brzo izgraditi sigurne, responzivne i skalabilne aplikacije. Ove usluge omogućavaju brzu pohranu i dohvaćanje podataka te pouzdanu autentifikaciju korisnika, što su ključne komponente modernih mobilnih i web aplikacija. Firebase nudi sveobuhvatno rješenje koje pojednostavljuje razvoj, održavanje i skaliranje aplikacija, omogućavajući da se pažnja usmjeri na razvoj funkcionalnosti i korisničkog iskustva.

Cloud Firestore

Firebase Firestore je skalabilna, fleksibilna NoSQL baza podataka u oblaku (engl. *cloud*) koja omogućava pohranu i sinkronizaciju podataka između korisničkih aplikacija u stvarnom vremenu. Podaci unutar Firestore-a su organizirani u obliku kolekcija i dokumenata. Kolekcije su osnovne jedinice organizacije podataka i sadrže skup povezanih dokumenata. S druge strane, dokumenti su zapisi unutar kolekcija koji sadrže podatke u obliku JavaScript Object Notation (JSON) objekta. Svaki dokument u Firestoreu ima jedinstveni identifikator unutar svoje kolekcije i može sadržavati raznovrsne podatke kao što su nizovi, brojevi, datumi, pa čak i ugniježdene kolekcije.

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

Ovo omogućava fleksibilnu i dinamičku strukturu podataka koja se može prilagoditi različitim potrebama aplikacije. Firestore je dokumentno-orijentirana baza podataka, što znači da pohranjuje podatke u obliku dokumenata unutar zbirki. Prednost Firestorea leži u sposobnosti da se lako integrira s ostalim Firebase uslugama, poput autentifikacije i funkcija u oblaku, te u podršci za izvanmrežni način rada, što znači da korisnici mogu nastaviti koristiti aplikaciju čak i kad nisu povezani na internet. Kada se ponovno povežu, Firestore automatski sinkronizira podatke [16].

Firestore Authentication

Firestore autentifikacija pruža jednostavno rješenje za upravljanje autentifikacijom korisnika u aplikacijama. Podržava različite metode prijave, uključujući prijavu putem e-pošte i lozinke, prijavu putem trećih strana poput Googlea, Facebooka i Twittera, te prijavu putem telefonskog broja. Firestore autentifikacija također omogućava jednostavnu integraciju s vlastitim sustavima autentifikacije, pružajući fleksibilnost u implementaciji složenijih sigurnosnih zahtjeva. Kroz konzolu, programeri mogu jednostavno upravljati korisničkim računima, resetirati lozinke, provjeravati e-mail adrese i još mnogo toga, što olakšava održavanje sigurnosti i privatnosti korisničkih podataka [17].

3.2.4 Django

Django je visoko produktivan i skalabilan web okvir (engl. web framework) za Python, koji omogućava brzo i efikasno stvaranje složenih web aplikacija. Web okvir se pridržava filozofije "ne ponavljaj se" (engl. Don't Repeat Yourself) i "konvencija nad konfiguracijom" (engl. Convention over Configuration), što omogućava brži razvoj aplikacija i jednostavnije održavanje koda.

Jedna od ključnih značajki Django je njegov Object-Relational Mapping (ORM) sustav, koji omogućava rad s bazama podataka pomoću Python objekata umjesto pisanja Structured Query Language (SQL) upita. Ovo značajno pojednostavljuje rad s bazama podataka i smanjuje mogućnost pojave pogrešaka, omogućavajući programerima da se usmjere na logiku aplikacije umjesto na složenosti manipulacije

podacima.

Django je poznat po svojoj snažnoj sigurnosnoj podršci, jer automatski štiti aplikacije od uobičajenih web sigurnosnih prijetnji poput SQL injekcija, Cross Site Scripting (XSS) i Cross-Site Request Forgery (CSRF). Ove ugrađene sigurnosne značajke omogućavaju programerima da s povjerenjem razvijaju sigurne web aplikacije.

Ističe se i po svojoj opširnoj dokumentaciji i velikoj zajednici koja pruža podršku. Ova zajednica aktivno doprinosi razvoju novih značajki, rješavanju problema i stvaranju dodatnih alata i paketa koji proširuju funkcionalnosti Djanga [18].

3.2.5 MATLAB

MATLAB (skraćenica od Matrix Laboratory) je visokorazinski programski jezik i računalni alat široko korišten u znanstvenim, inženjerskim i matematičkim zajednicama za numeričke proračune, simulacije, analize podataka i vizualizacije. MATLAB pruža interaktivno okruženje koje kombinira računalnu moć sa sveobuhvatnim knjižnicama funkcija, omogućavajući korisnicima rješavanje složenih problema.

Jedna od glavnih prednosti MATLAB-a je mogućnost rada s matricama i vektorima kao osnovnim podacima. Ovaj pristup omogućava brzo izvođenje matematičkih operacija, što je posebno korisno u aplikacijama koje zahtijevaju obradu velikih skupova podataka ili složene postupke iz linearne algebre. MATLAB nudi širok spektar ugrađenih funkcija za linearnu algebru, statistiku, Fourierovu analizu, optimizaciju i numeričku integraciju.

Osim toga, MATLAB nudi opsežan set alata za vizualizaciju podataka. Korisnici mogu jednostavno generirati 2D i 3D grafove, dijagrame, površinske prikaze i animacije, što je korisno za interpretaciju rezultata i izradu vizualno privlačnih izvješća. Ove mogućnosti vizualizacije omogućavaju istraživačima i inženjerima da na intuitivan način analiziraju i prezentiraju svoje podatke.

Alat podržava skriptiranje i automatizaciju zadataka. Korištenjem skripti i funkcija, korisnici mogu automatizirati ponavljajuće zadatke, smanjujući potrebu za ručnim unosom i minimizirajući mogućnost pogrešaka. Ova mogućnost čini MATLAB izuzetno korisnim u industriji i akademiji, gdje je potrebna visoka razina točnosti i

učinkovitosti.

MATLAB također nudi mogućnost integracije s drugim programskim jezicima poput C, C++, Java i Python, što omogućava proširenje funkcionalnosti MATLAB-a ili integraciju MATLAB skripte u druge sustave i aplikacije.

Sve ove značajke čine MATLAB neizostavnim alatom u brojnim industrijama i akademskim disciplinama, omogućavajući inženjerima i znanstvenicima da učinkovito rješavaju složene probleme, analiziraju podatke i razvijaju inovativna rješenja [19].

MATLAB Engine za Python

MATLAB Engine za Python je alat koji omogućava integraciju MATLAB-a s Python-om. Ova integracija pruža korisnicima mogućnost korištenja MATLAB funkcija i skripti izravno iz Python okruženja, čime se kombiniraju snage oba jezika: napredne matematičke i grafičke mogućnosti MATLAB-a te fleksibilnost i svestranost Python-a.

Jedna od ključnih prednosti MATLAB Engine-a za Python je mogućnost korištenja postojećih MATLAB funkcija bez potrebe za ponovnim pisanjem koda u Python-u. Ovo je osobito korisno u industrijama i akademskim krugovima gdje je MATLAB standardni alat, a prelazak na Python bi zahtijevao značajno ulaganje vremena i resursa. Alat omogućava pozivanje MATLAB funkcije izravno iz Python skripte, prosljeđivanje podataka između dva jezika i dohvaćanje povratnih rezultata u Python za daljnju analizu ili vizualizaciju.

Osim toga, MATLAB Engine omogućava interaktivnu analizu podataka i razvoj algoritama. Moguće je, primjerice, koristiti Python za obradu velikih skupova podataka, a zatim iskoristiti MATLAB za napredne matematičke analize ili vizualizaciju. Ovaj pristup omogućava iskorištavanje najboljeg od oba svijeta, čime se ubrzavaju istraživački i razvojni procesi [20].

3.2.6 Angular

Angular je otvoreni radni okvir (engl. framework) za razvoj aplikacija na klijentskoj strani. Temeljen na TypeScriptu, Angular omogućava izradu složenih i responzivnih

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

web aplikacija koje su jednostavne za održavanje i testiranje. Njegova modularna struktura omogućava razdvajanje logike aplikacije na komponente, što doprinosi boljoj organizaciji i ponovnoj upotrebi koda.

Svaka komponenta u Angularu sadrži logiku, predložak i stilove, što omogućava jasan i organiziran pristup razvoju aplikacija. Ovakva arhitektura pojednostavljuje održavanje i testiranje aplikacija, te omogućava lakše upravljanje složenim projektima.

Osim toga, Angular nudi snažan sustav upravljanja stanjima, usmjeravanje (engl. routing), kao i alatne značajke za rad s obrascima (engl. forms) i HTTP zahtjevima. Zahvaljujući tome, Angular se koristi za razvoj širokog spektra aplikacija, od jednostavnih web stranica do složenih poslovnih rješenja.

Predlošci u Angularu

U Angularu, HyperText Markup Language (HTML) se koristi kao temelj za izradu predložaka (engl. templates) koji definiraju strukturu korisničkog sučelja. Predlošci su obogaćeni Angularovim jezikom predložaka, koji omogućava dinamičko vezanje podataka (engl. data binding) te interakciju s komponentama kroz direktive i izraze (engl. expressions). Korištenjem Angularovih izraza poput `@if` i `@for`, moguće je manipulirati prikazom elemenata na temelju stanja aplikacije.

Svaki predložak sadrži i stilove koji omogućavaju modificiranje izgleda predloška. U radu se koristio Sassy Cascading Style Sheets (SCSS) za oblikovanje predložaka. SCSS je preprocesor za Cascading Style Sheets (CSS) koji se često koristi u Angular aplikacijama zbog svojih naprednih značajki poput varijabli, ugniježdivanja (engl. nesting) i funkcija. U Angularu, SCSS stilovi se obično vežu uz pojedine komponente, omogućavajući im da budu inkapsulirani i specifični za određenu komponentu. Ovakav pristup doprinosi modularnosti i održivosti koda jer se stilovi ne prelijevaju između komponenti.

TypeScript

TypeScript je programski jezik otvorenog koda koji je razvijen s ciljem poboljšanja JavaScripta. Kao nadskup JavaScripta, TypeScript dodaje statičko tipiziranje (engl. static typing) i napredne razvojne alate, omogućavajući ranije otkrivanje pogrešaka i povećanje produktivnosti programera. Podržava značajke objektnog orijentiranog programiranja, poput razreda i sučelja, što olakšava organizaciju i održavanje koda. Kompatibilan je s JavaScriptom, omogućavajući njegovu postepenu integraciju u postojeće projekte. Svojom rastućom popularnošću i podrškom od strane alata i okvira, TypeScript postaje ključna tehnologija za razvoj skalabilnih i pouzdanih web aplikacija [21].

ApexCharts

ApexCharts je moderna JavaScript knjižnica za stvaranje interaktivnih i responzivnih grafova koji se lako mogu integrirati u web aplikacije. Razvijen s ciljem pružanja jednostavnog i intuitivnog alata za vizualizaciju podataka. Knjižnica podržava širok spektar različitih tipova grafova, uključujući linijske grafove, stupčaste grafove, tortne dijagrame, toplinske karte, i mnoge druge. Zahvaljujući svojoj prilagodljivosti, korisnici mogu lako konfigurirati stil i funkcije grafova putem jednostavnog API-a, čime se omogućava prikaz složenih podataka na vizualno privlačan način. Kompatibilan je s popularnim JavaScript okvirima kao što su React, Angular, i Vue.js, što ga čini idealnim izborom za ovaj rad [22].

Angular material

Angular Material je službeni skup UI komponenti razvijen od strane Angular tima, temeljen na Googleovim principima dizajna poznatima kao Material Design. Ova knjižnica omogućava da se brzo i jednostavno integriraju responzivne i vizualno ujednačene komponente u Angular aplikacije. Angular Material pruža širok spektar unaprijed izrađenih komponenti kao što su gumbi, navigacijske trake, obrasci, kartice, dijalozi i druge, koje su optimizirane za mobilne uređaje i desktop preglednike. Sve komponente su dizajnirane kako bi slijedile najbolje prakse u pogledu

pristupačnosti i performansi, što rezultira boljim korisničkim iskustvom. Uz lakoću korištenja, Angular Material nudi i snažnu prilagodljivost, omogućavajući prilagodbu izgleda i ponašanje komponenti kako bi se zadovoljile specifične potrebe projekata. Osim što ubrzava razvojni proces, Angular Material osigurava i konzistentan izgled aplikacija, čime se olakšava održavanje i daljnji razvoj [23].

3.3 Poslužiteljski dio sustava

U kontekstu razvoja modernih aplikacija, kao što su Android i Angular aplikacije, Firestore se ističe svojom sposobnošću da rukuje strukturama podataka koje uključuju hijerarhijske i relacijske odnose, omogućavajući da se jednostavno upravlja složenim strukturama podataka.

Implementacija Firestore baze podataka u ovoj aplikaciji omogućava efikasno upravljanje velikim količinama podataka i pruža fleksibilnost potrebnu za skaliranje i prilagodbu promjenjivim potrebama aplikacije. Firestore struktura koja se temelji na kolekcijama i dokumentima savršeno odgovara potrebama moderne aplikacije, omogućavajući jednostavnu integraciju i upravljanje podacima u stvarnom vremenu.

Za pohranu i rukovanje svim vremenskim oznakama u aplikaciji koristi se razred `Timestamp` koji pruža `Firestore`. `Timestamp` se može stvoriti korištenjem statičke metode `Timestamp.now()`, koja vraća trenutni datum i vrijeme, ili korištenjem `Timestamp.fromDate(date)`, gdje se kao argument prosljeđuje `JavaScript Date` objekt. Ovaj razred omogućava i izvođenje raznih operacija, kao što su usporedba dvaju vremenskih oznaka ili pretvorba u različite formate vremena, kao što su Unix vrijeme ili `JavaScript Date` objekti. Koristeći `Timestamp` razred, `Firestore` osigurava dosljedno rukovanje vremenskim podacima.

Baza podataka aplikacije sastoji se od četiri glavne kolekcije:

- *audiometry*,
- *questionnaire*,
- *tinnitus*,
- *user*.

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

U nastavku poglavlja se nalazi detaljni opis svake kolekcije te sadržaj dokumenata unutar tih kolekcija.

3.3.1 Kolekcija *audiometry*

Ova kolekcija pohranjuje podatke vezane uz audiometrijska testiranja korisnika. Svaki dokument unutar ove kolekcije sadrži sljedeće podatke:

- `createdAt`: `Timestamp` instanca koja predstavlja datum i vrijeme stvaranja ovog zapisa;
- `userId`: jedinstveni identifikator korisnika koji je stvorio ovaj zapis;
- `amplitude`: cjelobrojna vrijednost koja sadrži amplituda zvuka;
- `frequency`: cjelobrojna vrijednost koja sadrži frekvenciju zvuka u hercima;
- `ear`: znakovni niz koji predstavlja uho na kojem je obavljeno mjerenje;
- `audiogramId`: jedinstveni identifikator audiograma na kojem će podatak biti prikazan;
- `phoneModel`: model korisničkog Android uređaja;
- `systemAmplitude`: cjelobrojna vrijednost koja predstavlja razinu intenziteta zvuka podešenog sa strane korisnika na njegovom Android uređaju.

3.3.2 Kolekcija *questionnaire*

Kolekcija `questionnaire` sadrži odgovore korisnika na upitnike vezane uz njihove slušne sposobnosti i opće zdravstveno stanje. Svaki dokument unutar ove kolekcije uključuje:

- `createdAt`: `Timestamp` instanca koja predstavlja datum i vrijeme stvaranja upitnika;
- `scores`: polje cijelih brojeva koje predstavlja rezultat odgovora na svako pitanje;
- `userId`: jedinstveni identifikator korisnika koji je ispunio upitnik.

3.3.3 Kolekcija *tinnitus*

Ova kolekcija fokusira se na specifične podatke vezane uz stanje tinitusa kod korisnika. Svaka kolekcija unutar ovog dokumenta sadržava:

- `createdAt`: Timestamp instanca koja predstavlja datum i vrijeme kreiranja ovog zapisa;
- `userId`: jedinstveni identifikator korisnika;
- `amplitude`: cjelobrojna vrijednost koja sadrži amplitudu zvuka;
- `frequency`: cjelobrojna vrijednost koja sadrži frekvenciju zvuka na Musical Instrument Digital Interface (MIDI) skali;
- `ear`: znakovni niz koji predstavlja uho na kojem je obavljeno mjerenje;
- `phoneModel`: model korisničkog Android uređaja;
- `systemAmplitude`: cjelobrojna vrijednost koja predstavlja razinu intenziteta zvuka podešenog sa strane korisnika na njegovom Android uređaju;
- `error`: znakovni niz koji sadrži informaciju o tome je li korisnik pojačao zvuk do maksimuma na uređaju i je li korisnik uključio slušalice prilikom identifikiranja frekvencije tinitusa.

3.3.4 Kolekcija *user*

Ovo je kolekcija koja pohranjuje osnovne informacije o korisnicima aplikacije. Dokumenti u ovoj kolekciji služe kao centralizirano mjesto za sve informacije vezane uz korisnike, omogućavajući jednostavan pristup i upravljanje korisničkim podacima. Primjer dokumenta u kolekciji `user` može izgledati ovako:

- `name`: puno ime korisnika;
- `email`: korisnikova adresa elektroničke pošte.

3.3.5 Obrada zvuka

Poslužiteljski dio sustava za obradu audio zapisa izrađen je u Djangu. Poslužitelj se pokreće naredbom:

```
python manage.py runserver 0.0.0.0:8000.
```

IP adresa 0.0.0.0 je posebna adresa koja govori poslužitelju da sluša sva dostupna mrežna sučelja. Drugim riječima, omogućava pristup poslužitelju s bilo koje IP adrese glavnog računala. Prilikom lokalnog pokretanja, drugi uređaji na istoj mreži mogu pristupiti pomoću IP adrese glavnog računala. Kako bi uspjeli pristupiti poslužitelju preko lokalne mreže, potrebno je u polje `ALLOWED_HOSTS` dodati znakovni niz lokalne IP adrese računala. Varijabla `ALLOWED_HOSTS` unutar datoteke `settings.py` služi kao važna sigurnosna zaštita koja pomaže u sprječavanju neovlaštenog pristupa web aplikaciji i osigurava da samo zahtjevi sa specifičnih, odobrenih domena ili IP adresa budu obrađeni. Po pokretanju poslužitelja, unutar varijable `eng` se pohranjuje instanca `MatlabEngine` koja omogućava komunikaciju između Python-a i MATLAB-a.

Unutar datoteke `urls.py` nalazi se ruta `modify-audio` koja poziva funkciju `modify_audio(request)` iz datoteke `views.py`. Metoda je anotirana s `@csrf_exempt` kako bi se moglo pristupiti tom pogledu preko mobilne aplikacije i pritom metoda prima jedan parametar `request`. Varijabla `request` sadrži sve podatke o zahtjevu, a za uspješno izvršavanje funkcije, unutar objekta `request.FILES['file']` se mora nalaziti datoteka koju želimo obraditi, a unutar `request.POST.get('midi')` se mora nalaziti MIDI broj s kojim se audio datoteka filtrira. U nastavku je prikazan isječak koda koji služi za filtriranje audio datoteke:

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

```
1 audio = AudioSegment.from_file(uploaded_file, format="wav")
2 matlab_audio = eng.oktavnoFiltriranje(
3     audio_to_double_array(audio), audio.frame_rate, midi)
4 modified_audio = create_audio_segment(matlab_audio, audio)
5 buffer = io.BytesIO()
6 modified_audio.export(buffer, format="wav")
7 buffer.seek(0)
8 response = HttpResponse(buffer, content_type='audio/wav')
9 response['Content-Disposition'] = f'attachment;
10 filename="{uploaded_file.name} - filtered"
```

Ispis 3.3.1 Isječak koda u Python programskom jeziku

Prva linija koda koristi `pydub` knjižnicu za učitavanje audio datoteke u Python programskom jeziku. Funkcija `AudioSegment.from_file` otvara audio datoteku i sprema ju u varijablu `audio`.

Funkcija `oktavnoFiltriranje` se nalazi unutar MATLAB datoteke, a njen je zadatak filtrirati zvuk s obzirom na danu frekvenciju i pritom prima sljedeće parametre:

- `audio_to_double_array(audio)`: podaci iz zaglavlja Waveform Audio File Format (WAV) datoteke na temelju kojih se transformiraju uzorci zvuka u matricu decimalnih brojeva koja je pogodna za obradu unutar MATLAB-a;
- `audio.frame_rate`: brzina uzorkovanja (engl. frame rate) audio datoteke;
- `midi`: MIDI broj dobiven putem zahtjeva.

Filtrirani zvuk će biti vraćen iz MATLAB funkcije u obliku matrice s decimalnim brojevima pa je potrebno napraviti konverziju u novi `AudioSegment` s filtriranim podacima. Prvi red u matrici predstavlja uzorke lijevog kanala, a drugi red predstavlja uzorke desnog kanala. Iz instance novog `AudioSegment`-a stvara se WAV datoteka koja se nalazi unutar međumemorije poslužitelja kako se audiozapisi ne bi zadržali na poslužitelju. Držanje korisničkih audiozapisa na poslužitelju može biti problematično zbog problema povezanih s privatnošću podataka i usklađenošću s regulativama.

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

Kako bi se osigurala ispravna konverzija sirovih podataka WAV datoteke, bitno je prvo razumjeti format datoteke. WAV je standardni format za pohranu audio podataka na računalima. Ovaj format je postao uobičajen za nekomprimirani audio zapis, koristeći Pulse-Code Modulation (PCM) za kodiranje zvučnih valova. WAV datoteke su široko korištene zbog svoje jednostavnosti i visokokvalitetnog zvuka, što ih čini idealnim za profesionalnu obradu zvuka i arhiviranje.

Datoteka se sastoji od niza blokova podataka, od kojih je najvažnije zaglavlje datoteke. Zaglavlje sadrži podatke koji opisuju karakteristike audio sadržaja, kao što su broj kanala, brzina uzorkovanja, dubina bita i veličina audio podataka. Nakon zaglavlja, slijede stvarni audio podaci koji predstavljaju zvučni signal.

Zaglavlje WAV datoteke je strukturirano na sljedeći način:

- Resource Interchange File Format (RIFF) oznaka (ChunkID): Ova oznaka identificira datoteku kao RIFF datoteku, koja je osnovni format za multimedijalne datoteke korištene u WAV formatu;
- Veličina datoteke (ChunkSize): Ovaj broj označava veličinu cijele datoteke u oktetima, umanjeno za 8 okteta koji se koriste za ChunkID i ChunkSize;
- Format (Format): Vrijednost "57 41 56 45", koje u ASCII kodu predstavljaju znakovni niz "WAVE";
- Subchunk1 identifikator (Subchunk1ID): Vrijednost "66 6d 74 20", koja u ASCII kodu predstavlja znakovni niz "fmt", označava početak bloka koji sadrži informacije o audio formatu;
- Veličina Subchunk1 (Subchunk1Size): 4-oktetni cijeli broj koji definira veličinu ovog pod-dijela zaglavlja, tipično je 16 za PCM, što je najčešći audio format
- Audio format (AudioFormat): 2-oktetni cijeli broj (sadrži vrijednost 1 za PCM - nekomprimirani format) koji označava format zvučnih podataka;
- Broj kanala (NumChannels): vrijednost sadrži 2-oktetni cijeli broj koja pokazuje koliko kanala sadrži audio (1 za mono, 2 za stereo);
- Brzina uzorkovanja (SampleRate): vrijednost je 4-oktetni cijeli broj koji označava broj uzoraka po sekundi (npr. 44100 Hz);

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

- Brzina prijenosa okteta (`ByteRate`): 4-oktetni cijeli broj koji označava broj okteta koji se obrađuju svake sekunde, što pomaže u aplikacijama s izravnim prijenosom zvuka za procjenu brzine prijenosa podataka;
- Veličina bloka (`BlockAlign`): 2-oktetni cijeli broj koji označava broj okteta po uzorku svih kanala i pritom omogućava pravilno razdvajanje uzoraka u višekanalnim zapisima;
- Dubina bita (`BitsPerSample`): 2-oktetni cijeli broj koji označava broj bitova korištenih za pohranu svake uzorkovane vrijednosti pri čemu veća dubina bita omogućava veću preciznost zvuka;
- Subchunk2 ID (`Subchunk2ID`): Vrijednost "64 61 74 61", koja u ASCII kodu predstavlja znakovni niz "data", označava početak stvarnih audio podataka;
- Veličina Subchunk2 (`Subchunk2Size`): 4-oktetni cijeli broj koji označava veličinu stvarnih audio podataka u oktetima;
- Audio podaci: Niz podataka sa audio uzorcima koje WAV datoteka reproducira.

Po ponovnoj konstrukciji WAV datoteke, ona se postavlja unutar "Content-Disposition" zaglavlja u HTTP odgovoru, što određuje kako pošiljatelj zahtjeva treba rukovati sadržajem odgovora, posebno kod preuzimanja datoteka. Vrijednost "attachment" omogućavajućuje spremanje datoteke umjesto direktno prikazivanje datoteke, a "filename" predstavlja ime datoteke.

3.4 Klijentski dio sustava

Kako bi se omogućila komunikacija između poslužiteljskog dijela i korisničkog sučelja, potrebno je konfigurirati mobilnu aplikaciju i administracijsko sučelje. Firebase konfiguracija za Android i Angular aplikacije uključuje nekoliko ključnih koraka kako bi se omogućila integracija s Firebase servisima. Prvo, potrebno je stvoriti projekt u Firebase konzoli i dodati aplikaciju, unoseći osnovne informacije poput naziva paketa za Android ili generirajući konfiguracijske ključeve za Angular. Za Android aplikaciju, nakon što je aplikacija dodana u Firebase projekt, preuzima se datoteka `google-services.json`, koja se smješta u `app` direktorij projekta. U

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

`build.gradle` datotekama projekta zatim se dodaju potrebne ovisnosti i utičnice za Firebase, te se Firebase inicijalizira u glavnoj aktivnosti aplikacije. Za Angular aplikaciju, Firebase se integrira instalacijom Firebase i AngularFire modula putem Node Package Manager-a, nakon čega se konfiguracijski podaci dodaju u datoteku okruženja. Firebase se inicijalizira unutar Angular aplikacije putem modula, čime se omogućava pristup različitim Firebase servisima unutar komponenti i servisa aplikacije. Ova konfiguracija omogućava jednostavnu i sigurnu povezanost aplikacija s Firebaseom.

Komunikacija između Django poslužitelja i Android aplikacije je ostvarena putem HTTP zahtjeva. Sa strane Android aplikacije koristi se `OkHttp` knjižnica koja omogućava jednostavno i efikasno slanje HTTP zahtjeva i primanje odgovora. Prije slanja zahtjeva generira se sadržaj zahtjeva pomoću konstruktora `Builder()`. Tip zahtjeva se postavlja na `MultipartBody.FORM` koji zahtjevu daje oblik `multipart/form-data`. To je format koji omogućava da se različiti dijelovi podataka (tekst, datoteke) šalju kao zasebni dijelovi u jednom HTTP zahtjevu. Nakon toga se metodom `addFormDataPart` dodaje datoteka koja se mora formatirati pomoću metode `asRequestBody("audio/wav".toMediaTypeOrNull())` kako bi se mogla pravilno interpretirati na poslužitelju. Na kraju se metodom `addFormDataPart` dodaje i MIDI broj i po završetku izgradnje sadržaja zahtjeva poziva se metoda `build`. Uz pomoć `Request.Builder()` metode gradi se zahtjev koji sadrži metodu `url` za definiranje putanje do poslužitelja i metodu `post` koja prima prijašnje generiran sadržaj zahtjeva te definira da se radi o POST zahtjevu.

Odgovor koji se očekuje je modificirana datoteka. Po odgovoru, poziva se metoda `onResponse`. Unutar te metode datoteka se sprema na odgovarajuće mjesto unutar datotečnog sustava Android uređaja. Za spremanje podataka koristi se razred `ContentResolver` koji omogućava pristup različitim vrstama podataka pohranjenih u aplikaciji ili na uređaju. To je središnji mehanizam za pristup podacima koji su pohranjeni putem različitih pružatelja sadržaja, kao što su kontakti, SMS poruke, medijski sadržaji, ali i druge vrste podataka. `ContentResolver` omogućava da se dohvaćaju i spremaju podaci bez izravnog pristupa datotekama ili bazama podataka i pritom pruža apstrakciju koja olakšava siguran pristup i manipulaciju podacima. Parametri za spremanje datoteke su dobiveni iz objekta

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

`MediaStore.MediaColumns` gdje je:

- `DISPLAY_NAME`: postavljen na ime datoteke s nastavkom "- filtrirano";
- `IS_MUSIC`: postavljen na 1 što označava da se radi o audiozapisu;
- `MIME_TYPE`: postavljen na "wav" što znači da je format datoteke WAV;
- `RELATIVE_PATH`: postavljen na direktorij "Music" kako bi se svi filtrirani audio zapisi nalazili na istom mjestu i bili dostupni korisniku ne samo na Android uređaju već i za potencijalnu uporabu na drugim uređajima.

Osim komunikacije s poslužiteljima, aplikacija mora korisniku omogućiti generiranje sinusnih zvukova za izvođenje testa audiometrije i ispitivanje frekvencije tinitusa. Sva logika povezana uz generiranje zvuka se nalazi unutar `FrequencyGenerator` razreda.

3.4.1 Razred *FrequencyGenerator*

Za upravljanje zvukom koriste se razredi `AudioManager` i `AudioTrack`. Ovi razredi omogućavaju kontrolu različitih aspekata audio izlaza, kao što su glasnoća, reproduciranje zvukova, te upravljanje audio fokusom. Dok `AudioManager` upravlja globalnim audio postavkama i fokusom, `AudioTrack` je razred koji omogućava nisku razinu kontrole nad audio izlazom. `AudioTrack` omogućava aplikacijama direktno slanje audio podataka u audio izlazni uređaj, što je korisno za reprodukciju prilagođenih zvukova, glazbe ili drugih audio podataka koji nisu unaprijed snimljeni ili pohranjeni.

Razred `AudioTrack` isto tako omogućava različite načine reprodukcije, uključujući `streaming` i `static` načine radova. U `streaming` modu, audio podaci se kontinuirano šalju u audio izlaz, što je korisno za duže audio tokove, poput glazbe ili govora. U `static` modu, svi audio podaci učitavaju se odjednom u memoriju, što je idealno za kratke zvučne efekte koji se brzo reproduciraju.

Metode i parametri unutar razreda `AudioTrack` pružaju mogućnost kontrole nad različitim aspektima audio reprodukcije, uključujući frekvenciju uzorkovanja, broj kanala, format audio podataka, te način rukovanja kašnjenjem. Ove opcije

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

omogućavaju optimizaciju audio performansi aplikacija i osiguravaju visoku kvalitetu zvuka.

Zvukovi koji se reproduciraju korisniku kod audiometrije i identifikacije frekvencije tinitusa su čisti sinusni valovi. Kako bi se ti sinusni valovi generirali, potrebno je prvo popuniti polje `buffer` unutar kojeg su pohranjeni uzorci koji se preko razreda `AudioTrack` puštaju na zvučnu karticu uređaja. Veličina polja `buffer` je jednaka frekvenciji uzorkovanja koja se dobiva pomoću `getNativeOutputSampleRate` metode iz razreda `AudioTrack`. Vrijednosti u polju `buffer` se popunjavaju po sljedećoj formuli:

$$buffer_i = \sin(2\pi * F * i/S) * A \quad (3.1)$$

Članovi u formuli su:

- $buffer_i$: vrijednost unutar polja `buffer` na indeksu i ;
- F : frekvencija tona u hercima;
- i : indeks polja `buffer` koje se popunjuje s tom vrijednošću;
- S : frekvencija uzorkovanja dobivena s `getNativeOutputSampleRate` metodom;
- A : amplituda signala prikazana kao 16-bitni broj koji može sadržavati vrijednosti od 0 do 32767.

Kada se popune vrijednosti unutar polja `buffer`, potrebno je pustiti ton samo na određeno uho za što se koristi metoda `setStereoVolume`. Metoda kao prvi parametar prima decimalnu vrijednost intenziteta zvuka za lijevo uho, a kao drugi parametar decimalnu vrijednost intenziteta zvuka za desno uho. Za odabrati na koje uho se pušta zvuk, dovoljno je postaviti intenzitet zvuka za željeno uho na vrijednost dobivenu metodom `getMaxVolume` iz razreda `AudioTrack`, a za drugo uho postaviti vrijednost na nulu.

Nakon odabira uha, važno je osigurati da se reprodukcija zvuka odvija u pozadini kako bi se izbjeglo blokiranje glavne niti (engl. `thread`) aplikacije. Pravilno upravljanje radnom niti je ključno za održavanje glatkog korisničkog iskustva i izbjegavanje

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

zamrzavanja aplikacije. U tu svrhu, Kotlin korutine zajedno s `Dispatchers.IO` omogućavaju jednostavno upravljanje radnom niti za pozadinske operacije. Sljedeći isječak koda (ispis 3.4.2) prikazuje kako je ostvarena kontinuirana reprodukcija zvuka u pozadini:

```
1 lifecycleScope.launch(Dispatchers.IO) {  
2     while (isPlaying) {  
3         audioTrack!!.write(buffer, 0, sampleRate)  
4     }  
5 }
```

Ispis 3.4.2 Isječak koda u Kotlin programskom jeziku

U programskom kodu prisutnom na ispisu 3.4.2, petlja `while (isPlaying)` koristi se za kontinuiranu reprodukciju zvuka sve dok varijabla `isPlaying` ima vrijednost `true`. Unutar petlje `write` metoda piše audio podatke iz međuspremnik (engl. `buffer`) u `AudioTrack` objekt. Prvi parametar je prijašnje popunjeno polje uzoraka, drugi parametar je pomak uzoraka u odnosu na prijašnje vrijednosti zapisane na zvučnu karticu, a treći parametar je veličina polja, odnosno frekvencija uzorkovanja. Budući da je korutina pokrenuta unutar `lifecycleScope`, ona će automatski biti otkazana kada se životni ciklus aktivnosti ili fragmenta završava. Ovo sprječava curenje memorije (engl. `memory leaks`) i osigurava oslobađanje resursa kada korutina završi.

Prilikom puštanja zvuka u svrhu audiometrije bitno je osigurati da korisnik ima uključene slušalice. Funkcija `initEarphoneReceiver` koja je definirana unutar razreda `FrequencyGenerator` inicijalizira prijemnik (engl. `receiver`) za praćenje stanja slušalica u Android uređaju. Funkcija prima argument `listener` tipa `OnEarphonePluggedListener`, koji je sučelje za slušanje promjena u stanju slušalica. Unutar funkcije stvara se anonimni objekt `BroadcastReceiver`, koji nadjačava metodu `onReceive`. Metoda `onReceive` poziva se svaki put kada sustav pošalje događaj u vezi s priključivanjem slušalica. `OnEarphonePluggedListener` sučelje sadrži metodu `onEarphonePlugged` koju aktivnosti ili fragmenti sustava

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

moгу nadjačati kako bi dobili informaciju o tome je li korisnik priključio slušalice.

Kako bi se osigurao konzistentni intenzitet zvuka sa strane mobilnog Android uređaja, po pokretanju aplikacije je potrebno osigurati da je korisniku zvuk uređaja pojačan na maksimum i nije modificiran prilikom testiranja. U tu svrhu je generiran razred `AppVolumeObserver` koji nasljeđuje `ContentObserver` i pritom omogućava praćenje promjena glasnoće na uređaju. Konstruktor ovog razreda prima `context` i `handler`, pri čemu `context` omogućava pristup sustavnim uslugama, dok `handler` omogućava rukovanje asinkronim zadacima. Unutar razreda, `AudioManager` se koristi za dobivanje i upravljanje razinom glasnoće. Razred također definira varijablu `volumeChangeListener` koja prima implementaciju sučelja `OnVolumeChangeListener`. Ovo sučelje definira metodu `onVolumeChanged` koja se poziva kada se promijeni glasnoća. Kada se dogodi promjena u glasnoći, metoda `onChange` dobiva trenutnu razinu glasnoće pomoću `AudioManager` i prosljeđuje tu vrijednost metodi `onVolumeChanged`. Time se omogućava aplikaciji da dinamički reagira na promjene glasnoće.

Osim kontrole nad glasnoćom i frekvencijom, razred `FrequencyGenerator` sadrži funkciju koja pretvara MIDI frekvencije u herce putem izvoda:

$$F = 440.0 \times 2.0^{\frac{m-69}{12.0}} \quad (3.2)$$

Unutar formule 3.2 varijabla m označava MIDI frekvenciju tona. Broj 440 označuje frekvenciju tona A4 koji je standardno definiran kao 440 Hz u modernom muzikalnom sustavu. Ova standardizacija omogućava dosljednost u muzičkom notiranju i reprodukciji. U MIDI sustavu, A4 je dodijeljen broju 69. Ova povezanost služi kao referentna točka za sve druge note. Razlika između nota u MIDI sustavu izražava se u polutonovima. Svaka promjena za jedan poluton uzrokuje promjenu frekvencije za faktor od $\sqrt[12]{2}$.

Nakon spremanja ovih podataka prilikom identifikacije frekvencije tinitusa, vrijednosti se prikazuju na raspršenom grafikonu koji je generiran uz pomoć knjižnice `MPAndroidChart`. `MPAndroidChart` nudi širok raspon funkcionalnosti za prikaz raznih vrsta grafova s naglaskom na interaktivnost i lakoću korištenja. Knjižnica omogućava jednostavnu i brzu integraciju grafičkih prikaza u Android aplikacijama

bez potrebe za razvijanjem složenih grafičkih sustava od nule. Za razliku od prikaza rezultata mjerenja frekvencije tinitusa, rezultati audiometrije su prikazani na audiogramu koji je kreiran od nule u razredu `AudiogramView` koja nasljeđuje razred `View` i sadrži `Canvas`.

Android `Canvas` je razred unutar Android grafičkog sustava koja omogućava crtanje. Ona pruža bogat skup za crtanje osnovnih oblika, teksta, slika, pa čak i složenijih grafičkih elemenata, omogućavajući tako stvaranje prilagođenih prikaza i grafičkih elemenata poput audiograma.

3.5 Administracijsko sučelje u Angularu

Administracijsko sučelje u Angularu je sastavljeno od nekoliko komponenta:

- `AppComponent`: Glavna komponenta u Angularu koja služi za prikaz drugih komponenta;
- `LoginComponent`: Služi za Google autentikaciju korisnika sučelja;
- `HomeComponent`: Prikazuje listu korisnika;
- `UserComponent`: Prikazuje detalje o korisniku;
- `AudiogramComponent`: Nalazi se unutar `UserComponent` za prikaz audiograma i liste svih audiograma;
- `TinnitusGraphComponent`: Služi za prikaz podataka o tinitusu;
- `QuestionnaireComponent`: Prikazuje listu upitnika i korisnikove odgovore na njih.

Komponente se prikazuju ovisno o tome na kojoj ruti se korisnik administracijskog sučelja nalazi:

- ruta `/login` prikazuje komponentu `LoginComponent` i na nju je korisnik usmjeren po inicijalizaciji aplikacije;
- ruta `/home` prikazuje `HomeComponent`;
- ruta `/user/{userId}` prikazuje `UserComponent` gdje parametar `userId`

Poglavlje 3. Razvoj mobilne aplikacije za terapiju tinitusa zvukom

predstavlja identifikacijski broj korisnika te pritom služi za dohvat informacija koje će se prikazati unutar komponente.

U slučaju da korisnik ručno unese Uniform Resource Locator (URL) koji nije definiran unutar datoteke `app.routes.ts`, automatski je preusmjeren na rutu `/login`. Rute `/home` i `/user` su zaštićene s `AuthGuard` razredom iz Firebase knjižnice koja onemogućuje prikaz komponenta u slučaju da korisnik trenutno nije prijavljen unutar administracijskog sučelja.

Unutar predložaka su korištene sljedeće Angular Material komponente:

- `<mat-toolbar>`: za prikaz alatne trake;
- `<mat-form-field>`: za pretraživanje po imenu korisnika;
- `<mat-selection-list>`: za prikaz i mogućnost odabira liste korisnika i audiograma;
- `<mat-expansion-panel>`: za prikaz liste upitnika koja se proširuje pri odabiru;
- `<mat-icon>`: za prikaz ikona (npr. ikona za odjavu administratora);
- `MatSnackBar`: za prikaz notifikacija kod pogrešaka u dohvat podataka.

Kako bi logika za učitavanje podataka s Firebasea bila objedinjena i mogla se ponovno iskoristiti, izrađen je servis `FirestoreService`. Razred sadrži metodu u koju je dovoljno proslijediti kolekciju i upit za dohvat podataka o listi ili pojedinom korisniku. Podaci o audiometriji i tinitusu su prikazani uz pomoć raspršenog grafikona iz `ApexCharts` knjižnice.

Poglavlje 4

Karakteristični slučajevi korištenja

U ovom poglavlju opisane su ključne funkcionalnosti implementirane unutar sustava za terapiju tinitusa zvukom, s naglaskom na opcije koje sustav omogućava korisnicima i administratorima. Prikazan je način na koji način različiti dijelovi sustava međusobno djeluju kako bi pružili cjelovito rješenje za dijagnozu i terapiju tinitusa.

U prvom dijelu poglavlja, fokus je na funkcionalnostima Android mobilne aplikacije koja je dizajnirana za pacijente. Opisano je kako aplikacija omogućava mjerenje osjetljivosti sluha, detekciju ključnih parametara tinitusa te provođenje terapija temeljenih na filtriranim zvučnim zapisima. Uz to, prikazat će se dio aplikacije s upitnicima za samoprocjenu koji pomažu pacijentima pratiti učinke tinitusa na njihovu svakodnevnicu.

Drugi dio poglavlja posvećen je administracijskom sučelju, koje omogućava upravljanje korisnicima i pregled podataka koje generiraju korisnici aplikacije. Opisano je kako administratori mogu koristiti ovo sučelje za praćenje napretka pacijenata, analizu podataka i prilagodbu terapijskih protokola.

4.1 Android aplikacija

Aplikacija je dizajnirana s nekoliko aktivnosti i fragmenata. Prva aktivnost koja se prikazuje na ekranu prilikom otvaranja aplikacije omogućava prijavu u sustav. Korisnik se može prijaviti putem Google računa ili anonimno. Ako se korisnik odluči

Poglavlje 4. Karakteristični slučajevi korištenja

na Google autentikaciju, u pozadini mu je stvoren račun i njegove podatke može pregledavati administrator. Prilikom anonimne autentikacije ne spremaju se korisnički podaci na poslužitelj, već se unutar Android uređaja pohranjuje nasumično generiran korisnički identifikator. Podaci anonimnog korisnika nisu vidljivi administratoru, ali su pohranjeni na poslužitelju kako bi ih korisnik mogao dohvatiti.

4.1.1 Mjerenje osjetljivosti sluha

Mjerenje osjetljivosti sluha je omogućeno na prvom fragmentu aplikacije prikazanom na slici 4.1. Ovaj fragment se pojavljuje nakon što se korisnik uspješno autentificira.



Slika 4.1 Ekran za audiometriju

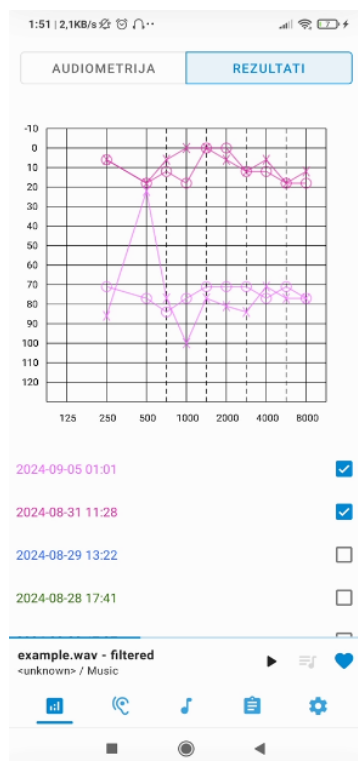
Pri vrhu fragmenta se nalaze gumbi "AUDIOMETRIJA" i "REZULTATI", koji omogućavaju prikaz sučelja za mjerenje osjetljivosti sluha i prikaz rezultata (automatski je odabrana vrijednost "AUDIOMETRIJA"). Prije početka audiometrije korisnik mora osigurati da su slušalice priključene i da je glasnoća Android uređaja pojačana

Poglavlje 4. Karakteristični slučajevi korištenja

do maksimuma. Pojačanjem zvuka uređaja do maksimuma osigurava se da će tokom procesa audiometrije zvučni signali imati ispravnu amplitudu. U slučaju da jedan od uvjeta nije zadovoljen, korisniku se na dnu ekrana prikazuje odgovarajuća poruka i gumbi za pokretanje testiranja se deaktiviraju. Prvi gumb "ZAPOČNI AUDIOMETRIJU" služi za pokretanje audiometrije. Po početku audiometrije, na lijevo uho se pušta zvučni signal s frekvencijom od 1000 Hz i tekst gumba se mijenja u "SLJEDEĆI ZVUK". Nakon što korisnik poslušava zvuk, omogućeni su mu gumbi za pojačanje zvuka ili za ponavljanje zvuka. U slučaju da korisnik ne čuje zvuk, mora kliknuti "POVEĆAJ GLASNOĆU" što reproducira istu frekvenciju zvuka s većom glasnoćom. U slučaju da je korisnik nesiguran je li čuo zvuk može kliknuti na "PONOVNI ZVUK" čime se ponavlja zvuk bez ikakvih promjena. Klikom na "SLJEDEĆI ZVUK" korisniku se pušta zvuk s frekvencijom od 2000 Hz i dalje redom 4000 Hz, 8000 Hz, 500 Hz, 250 Hz, 750 Hz, 1500 Hz, 3000 Hz, 6000 Hz, i na kraju, ponovno 1000 Hz kako bi se odbacila moguća pogreška pri prvom testiranju sluha sa frekvencijom od 1000 Hz. Kada korisnik uspije percipirati sve navedene frekvencije na lijevom uhu, te iste frekvencije se puštaju na desno uho. Prilikom završetka procesa, rezultati obavljene audiometrije se automatski spremaju na poslužitelj i dostupne su klikom na "REZULTATI" kao što je prikazano na slici 4.2.

Na ekranu se prikazuje prazan audiogram ispod kojeg se prikazuje sortirana lista audiograma počevši od najnovijeg. Svaki element u listi prikazuje datum i vrijeme izvođenja audiometrije te sadrži potvrdni okvir. Iz liste je moguće odabrati više audiograma nakon čega se odgovarajući audiogrami prikazuju u istoj boji kao datum i vrijeme elementa u listi. Kako bi se audiogrami mogli prikazati unutar jednog okvira, svaki je audiogram prikazan drugom bojom za lakše razlikovanje. Korisnik pritom može uočiti je li mu se osjetljivost sluha promijenila unutar nekog vremenskog perioda. Nakon mjerenja, korisnik se može savjetovati s audiologom ili istražiti na internetu informacije o njegovom audiogramu u slučaju očitih nepravilnosti prikazanih na audiogramu.

Poglavlje 4. Karakteristični slučajevi korištenja



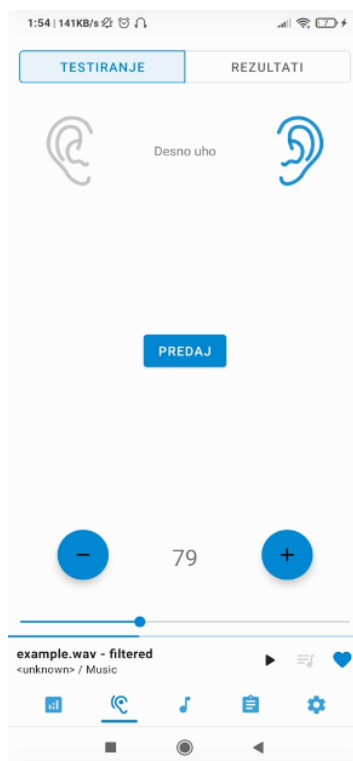
Slika 4.2 Ekran s prikazom rezultata audiometrije

4.1.2 Detekcija odrednica tinitusa

Drugi fragment korisniku omogućava identifikaciju frekvencije tinitusa. Slika ekrana po odabiru dotičnog fragmenta je prikazana na slici 4.3. Pri vrhu fragmenta se ponovo nalaze gumbi "TESTIRANJE" i "REZULTATI". Odabirom gumba za analizu korisniku su pri vrhu ekrana prikazani gumbi za odabir uha na kojem želi identificirati frekvenciju tinitusa. Po odabiru uha, gumb promijeni boju u plavu te se prikazuje tekst koji obavještava korisnika na koje uho se pušta zvuk. Pri dnu ekrana se nalaze gumbi s ikonama za pojačavanje i smanjivanje frekvencije. Između ta dva gumba je prikazana frekvencija u obliku MIDI tona. Ispod gumba nalazi se klizač koji korisniku omogućava podešavanje intenziteta zvuka. Kada korisnik ustanovi da se frekvencija koju čuje na odgovarajuće uho poklapa s frekvencijom tinitusa može pritisnuti na gumb "POHRANI" koji pohranjuje podatke o tinitusu na poslužitelj.

Korisniku je omogućen prikaz rezultata koji su učitani s poslužitelja pritiskom na

Poglavlje 4. Karakteristični slučajevi korištenja



Slika 4.3 Ekran za detekciju odrednica tinitusa

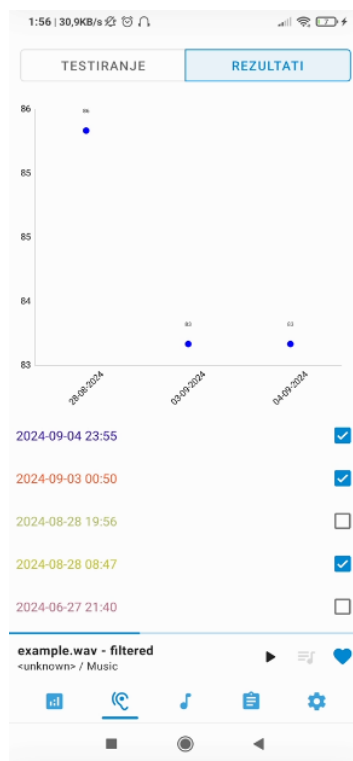
gumb "REZULTATI" (slika 4.4). Ispod gumba je prikazan raspršeni grafikon koji na ordinati prikazuje frekvencije tinitusa, a na apscisi prikazuje datume u kojima je ta frekvencija zabilježena. Ispod grafa se nalazi lista poredana po datumu stvaranja zapisa o frekvenciji tinitusa. Na listi je moguće odabrati više elemenata pa korisnik ima uvid u informaciju je li se frekvencija tinitusa promijenila i treba li pritom ponovo filtrirati zvučne datoteke kojima vrši terapiju.

4.1.3 Terapija zasnovana na filtriranim zvučnim zapisima

Terapija tinitusa zasnovana na filtriranim zvučnim zapisima je izvedena na način da korisnik sam odabire glazbu koju želi slušati tijekom terapije. Pri vrhu trećeg fragmenta je prikazana alatna traka s gumbima za pretragu glazbe, mjerača vremena, gumba za konvertiranje glazbe i gumba za sortiranje (slika 4.5).

Kako bi se u listi prikazale audio datoteke, korisnik mora odabrati gumb za

Poglavlje 4. Karakteristični slučajevi korištenja



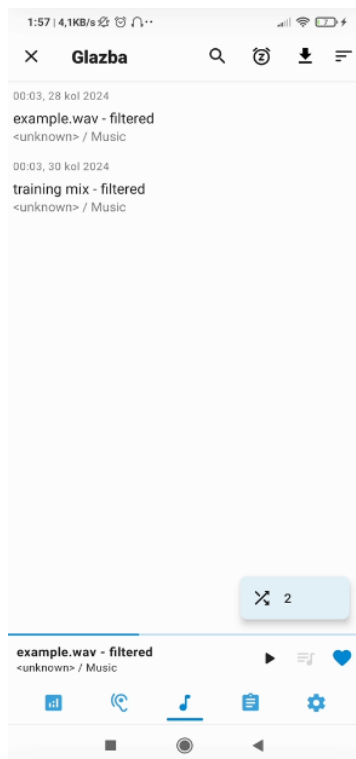
Slika 4.4 Ekran za prikaz pohranjenih rezultata s frekvencijom tinitusa

konvertiranje glazbe. Pritiskom na gumb, otvara se aplikacija Android sustava za odabir datoteka unutar koje korisnik može odabrati samo audio datoteke. Odabirom željene datoteke, vraća se prikaz liste s zvučnim datotekama i datoteka se u pozadini šalje i obrađuje na poslužitelju. Nakon obrade zvuka, obrađena zvučna datoteka se dohvaća i prikazuje u listi.

Kako bi korisniku omogućili praktično reproduciranje glazbe i kontrolu nad trenutno slušanom glazbom, iznad donje navigacijske trake se uvijek nalazi prikaz glazbe koja se trenutno reproducira. Klikom na naziv pjesme otvara se dijaloški okvir koji sadrži sljedeće funkcionalnosti:

- pokretanje ili zaustavljanje reprodukcije zvuka;
- ubrzavanje ili usporavanje zvuka;
- spremanje favorita;

Poglavlje 4. Karakteristični slučajevi korištenja



Slika 4.5 Ekran za reprodukciju audio datoteka

- *equalizer*;
- odabir načina ponavljanja audio datoteka;
- mogućnost pauziranja nakon što završi reprodukcija trenutne audio datoteke;
- pojačavanje ili smanjenje intenziteta zvuka.

4.1.4 Upitnici za samoprocjenu učinaka tinitusa

Na četvrtom fragmentu, prikazanom na slici 4.6, vizualizirana je lista upitnika s datumima i rezultatima na pojedini datum. Korisnik može pratiti svoje stanje po rezultatima i ispuniti novi upitnik klikom na gumb u donjem desnom kutu fragmenta. Kada korisnik klikne na gumb, otvara se nova aktivnost s pitanjima iz THI upitnika (slika 4.7).

THI je standardizirani upitnik koji se koristi za procjenu utjecaja tinitusa na

Poglavlje 4. Karakteristični slučajevi korištenja



Upitnici	
27/08/2024 15:37	Rezultat: 34
27/08/2024 15:30	Rezultat: 2
27/08/2024 15:28	Rezultat: 2
27/08/2024 15:25	Rezultat: 12
27/08/2024 15:23	Rezultat: 6
27/08/2024 15:09	Rezultat: 30
17/03/2024 23:16	Rezultat: 6

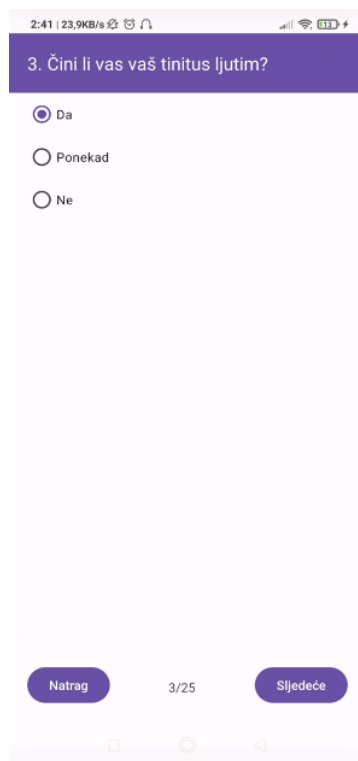
Slika 4.6 Ekran s listom ispunjenih upitnika

kvalitetu života pojedinca [24]. Razvijen je s ciljem objektivnog mjerenja stupnja smetnji i emocionalnog opterećenja koje tinitus može uzrokovati. THI upitnik se sastoji od 25 pitanja koja su strukturirana tako da pokrivaju tri glavne domene: funkcionalnu, emocionalnu i katastrofičnu. Funkcionalna domena obuhvaća pitanja vezana uz svakodnevne aktivnosti i koncentraciju, emocionalna domena se fokusira na psihološke učinke, kao što su stres i depresija, dok katastrofična domena procjenjuje strahove i osjećaj beznada povezan s tinitusom.

Svako pitanje u THI upitniku može se ocijeniti s "Da", "Ponekad" ili "Ne", pri čemu odgovor "Da" nosi četiri boda, odgovor "Ponekad" nosi dva boda, a odgovor "Ne" nosi nula bodova. Odgovori se zatim zbrajaju kako bi se dobio ukupni rezultat. Rezultati se kreću od 0 do 100, pri čemu viši rezultat označava veći stupanj smetnji uzrokovanih tinitusom. Na temelju ukupnog rezultata, pacijenti se mogu svrstati u jednu od pet kategorija: blagi, umjereni, ozbiljni, jaki ili vrlo jaki učinci tinitusa na kvalitetu života.

Poglavlje 4. Karakteristični slučajevi korištenja

THI je alat koji omogućava kliničarima da razumiju ozbiljnost problema i učinkovitost terapija tijekom vremena, te se koristi kao standardna metoda u istraživanjima i kliničkim procjenama tinitusa. Njegova jednostavnost, pouzdanost i validiranost čine ga ključnim instrumentom za praćenje pacijenata s tinitusom i planiranje adekvatnih terapijskih intervencija [25].



Slika 4.7 Ekran s pitanjem iz THI

4.2 Administracijsko web sučelje

Administracijsko sučelje se sastoji od tri glavna modula koji omogućavaju:

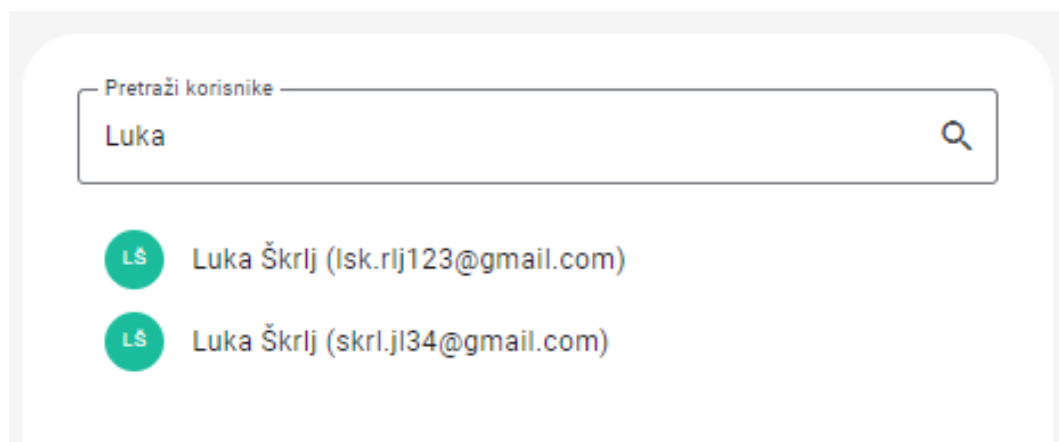
- autentikaciju;
- prikaz liste i odabir korisnika;
- prikaz detalja o svakom korisniku.

Poglavlje 4. Karakteristični slučajevi korištenja

U autentikacijskom prozoru korisniku se automatski otvara dijaloški okvir unutar kojeg može odabrati Google račun s kojim se želi prijaviti u sustav. U slučaju da slučajno zatvori prozor, može kliknuti na gumb "Prijavi se" za prijavu u sustav.

4.2.1 Upravljanje korisnicima

Nakon uspješne autentikacije, korisnik sučelja je preusmjeren na rutu `/home`. Preusmjerenje na tu rutu učitava sve korisnike iz kolekcije `users`. Na vrhu komponente je prikazana alatna traka s opcijom za odjavu administratora iz sučelja. Korisnike je moguće pretražiti po imenu kao u primjeru sa slike 4.8.



Slika 4.8 Ekran s listom korisnika

Na svakoj stavci liste nalazi se slika s inicijalima, puno ime korisnika i korisnikova adresa e-pošte. Klikom na stavku, aplikacija preusmjerava administratora na detalje o korisniku.

4.2.2 Pregled podataka koje generiraju korisnici

Pregled podataka o korisniku se sastoji od:

- liste s rezultatima audiometrije;
- grafa koji prikazuje podatke o tinitusu;

Poglavlje 4. Karakteristični slučajevi korištenja

- liste upitnika s rezultatima.

Svaki element u listi prikazuje datum na koji je obavljena audiometrija. Moguće je odabrati više audiograma nakon čega će se na desnoj strani prikazati graf kao što je prikazano na slici 4.9.

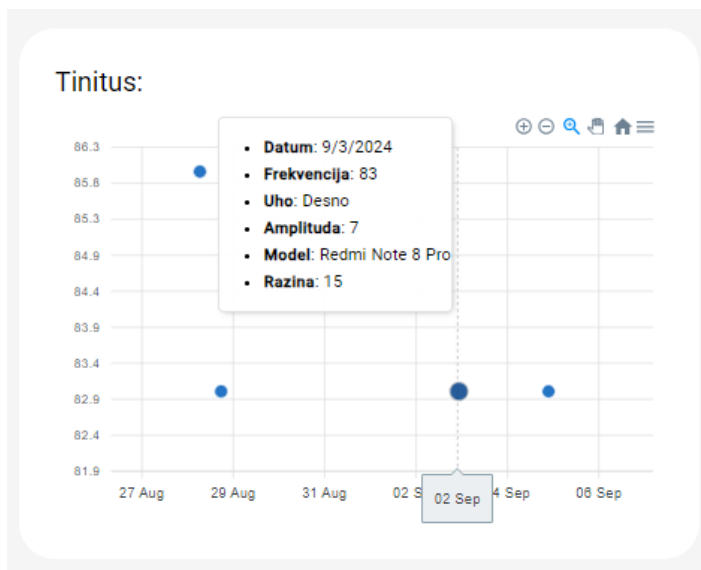


Slika 4.9 Ekran s prikazom podataka o audiometriji

Ispod prikaza audiograma se nalazi raspršeni grafikon koji prikazuje podatke o frekvenciji tinitusa kroz vremenski period (slika 4.10). Svaki grafikon nudi mogućnost uvećavanja, umanjivanja, pomicanja, povratka na početni prikaz i izvoza podataka. Izvoz podataka je moguć u .csv, .svg i .png formatu.

Na dnu prikaza korisničkih detalja prikazana je lista ispunjenih upitnika kao što je prikazano na slici 4.11. Svaki element u listi prikazuje datum na koji je upitnik popunjen zajedno s informacijama o ukupnom broju bodova. Odabirom upitnika proširuje se prikaz u kojem se nalaze pitanja s odgovorima i ostvarenim bodovima na tom pitanju.

Poglavlje 4. Karakteristični slučajevi korištenja



Slika 4.10 Ekran s prikazom podataka o tinitusu

Upitnici:

8/27/2024	Bodovi: 34 Da(6) Ponekad(5) Ne(14)	▼
8/27/2024	Bodovi: 2 Da(0) Ponekad(1) Ne(24)	▼
8/27/2024	Bodovi: 2 Da(0) Ponekad(1) Ne(24)	▼
8/27/2024	Bodovi: 12 Da(2) Ponekad(2) Ne(21)	▼
8/27/2024	Bodovi: 6 Da(1) Ponekad(1) Ne(23)	▼
8/27/2024	Bodovi: 30 Da(5) Ponekad(5) Ne(15)	▼
3/17/2024	Bodovi: 6 Da(1) Ponekad(1) Ne(23)	▼

Slika 4.11 Ekran s listom upitnika

Poglavlje 5

Zaključak

Cilj ovog diplomskog rada bio je razviti mobilnu aplikaciju za terapiju tinitusa zvukom koja će olakšati dijagnostiku i liječenje ovog sve učestalijeg problema. U skladu s relevantnim znanstvenim spoznajama i dijagnostičkim procesima, aplikacija je implementirana s tri glavna modula: audiometrijsko testiranje s prikazom audiograma, detekcija frekvencije tinitusa te transformacija audio zapisa u format prikladan za terapiju tinitusa zvukom. Ovi moduli omogućavaju korisnicima precizno praćenje njihovog sluha i omogućavaju prilagodbu terapije prema individualnim potrebama.

Uz razvoj mobilne aplikacije, implementiran je i poslužiteljski dio sustava koji omogućava učinkovito upravljanje podacima o korisnicima, pohranu rezultata mjerenja i administraciju standardiziranih upitnika. Time se osigurava sigurno i centralizirano upravljanje ključnim informacijama koje su od značaja pacijentima i audiolozima za praćenje napretka terapije.

Ovaj rad doprinosi u području digitalne zdravstvene tehnologije, s naglaskom na individualizirani pristup liječenju tinitusa. Razvijena mobilna aplikacija i pripadajući sustav pružaju praktičan alat koji može pomoći pacijentima u svakodnevnom životu. Daljnji rad može uključivati proširenje funkcionalnosti aplikacije te dodatna istraživanja kako bi se optimizirali i poboljšali procesi za terapiju tinitusa zvukom.

Bibliografija

- [1] PRAG SLUHA I TONALNA AUDIOMETRIJA. , s Interneta, <https://fonet.ffzg.unizg.hr/djelatnici/bakran/psi/PRAG%20SLUHA%20I%20TONALNA%20AUDIOMETRIJA.html> , kolovoz 2024. .
- [2] M. J. P. L.-. d. S.-.-. P. . Gates GA, “Presbycusis,” *PubMed*, 2005.
- [3] M. L. M. C. A.-J. L. D. A. JENNIFER JUNNILA WALKER, MD and A. JENNIFER S. SEALES, “Audiometry screening and interpretation,” *Am Fam Physician*, 2013.
- [4] BSA Professional Guidance Group , *Pure-tone air-conduction and boneconduction threshold audiometry with and without masking*, kolovoz 2018.
- [5] D. D. Jaikaransingh, “Audiogram interpretation – osce guide,” 2024.
- [6] N. Silver, “How to read your audiogram,” *healthline*, 2022.
- [7] C. J. P. S. Mitchell, “Audiology clinical masking,” *National Library of Medicine*, 2022.
- [8] Zvonjenje ili zujanje. , s Interneta, <https://hemed.hr/Default.aspx?sid=18236> , kolovoz 2024. .
- [9] Eggermont JJ, Roberts LE., *The neuroscience of tinnitus. Trends Neurosci.* 2004 Nov;27(11):676-82. doi: 10.1016/j.tins.2004.08.010. PMID: 15474168., kolovoz 2018.
- [10] Henry JA, Dennis KC, Schechter MA , *General review of tinnitus: prevalence, mechanisms, effects, and management. J Speech Lang Hear Res.* 2005 Oct;48(5):1204-35. doi: 10.1044/1092-4388(2005/084). PMID: 16411806., kolovoz 2018.
- [11] D. McFadden, “Tinnitus: facts, theories, and treatments,” *National Academy Press*, 1982.

Bibliografija

- [12] S. G. D. R. A. E.-. H. J. A. Hoare, D. J., “Sound therapy for tinnitus management: Practicable options,” *Journal of the American Academy of Audiology*, 2014.
- [13] S. H. S. W. . P. C. Okamoto, H., “Listening to tailor-made notched music reduces tinnitus loudness and tinnitus-related auditory cortex activity,” *Proceedings of the National Academy of Sciences*, 2010.
- [14] Developer Guides. , s Interneta, <https://developer.android.com/guide> , kolovoz 2024. .
- [15] Kotlin Documentation. , s Interneta, <https://kotlinlang.org/docs/home.html> , kolovoz 2024. .
- [16] Cloud Firestore. , s Interneta, <https://firebase.google.com/products/firestore> , kolovoz 2024. .
- [17] Firebase Authentication. , s Interneta, <https://firebase.google.com/products/auth> , kolovoz 2024. .
- [18] Django Documentation. , s Interneta, <https://docs.djangoproject.com/en/5.1/> , kolovoz 2024. .
- [19] MATLAB Documentation. , s Interneta, <https://www.mathworks.com/help/matlab/> , kolovoz 2024. .
- [20] Call MATLAB from Python. , s Interneta, <https://www.mathworks.com/help/matlab/matlab-engine-for-python.html> , kolovoz 2024. .
- [21] TypeScript is JavaScript with syntax for types. , s Interneta, <https://docs.djangoproject.com/en/5.1/> , kolovoz 2024. .
- [22] APEXCHARTS.JS. , s Interneta, <https://apexcharts.com/> , kolovoz 2024. .
- [23] Angular Material. , s Interneta, <https://material.angular.io/> , kolovoz 2024. .
- [24] Tinnitus Handicap Inventory. , s Interneta, https://order.starkeypro.com/pdfs/THI_Questionnaire.pdf , kolovoz 2024. .
- [25] Newman CW, Jacobson GP, Spitzer JB. , *Development of the Tinnitus Handicap Inventory. Arch Otolaryngol Head Neck Surg. 1996 Feb;122(2):143-8. doi: 10.1001/archotol.1996.01890140029007. PMID: 8630207.*, August 2018.

Pojmovnik

- API** Application Programming Interface. 13, 21
- CSRF** Cross-Site Request Forgery. 18
- CSS** Cascading Style Sheets. 20
- GPS** Globalni položajni sustav. 15
- HTML** HyperText Markup Language. 20
- HTTP** Hypertext Transfer Protocol. 13, 20, 28, 29
- IDE** integriranom razvojnom okruženju. 14
- JSON** JavaScript Object Notation. 16
- JVM** Java Virtual Machine. 15
- KBT** Kognitivno-bihevioralna terapija. 9
- MIDI** Musical Instrument Digital Interface. 24–26, 29, 33, 39
- ORM** Object–Relational Mapping. 17
- PCM** Pulse-Code Modulation. 27
- RIFF** Resource Interchange File Format. 27
- SCSS** Sassy Cascading Style Sheets. 20

SDK Android Software Development Kit. 15

SQL Structured Query Language. 17, 18

THI Tinnitus Handicap Inventory. viii, 42–44

URL Uniform Resource Locator. 35

WAV Waveform Audio File Format. 26–28, 30

XML Extensible Markup Language. 12

XSS Cross Site Scripting. 18

Sažetak

U ovome radu predstavljena je izvorna mobilna aplikacija za terapiju tinitusa zvukom, koja se zasniva na standardnim dijagnostičkim protokolima i priznatoj metodi liječenja. Aplikacija obuhvaća tri ključna modula: audiometrijsko testiranje s prikazom audiograma, detekciju odrednica tinitusa, te transformaciju audio zapisa u format pogodan za terapiju tinitusa zvukom. Dodatno je, pomoću standardiziranih upitnika, omogućeno i kontinuirano praćenje utjecaja tinitusa na kvalitetu života pacijenta. Razvijen je i poslužiteljski dio sustava koji omogućava prikupljanje i pregled podataka o korisnicima, kao i svim rezultatima mjerenja koji se generiraju putem mobilne aplikacije.

Ključne riječi — tinitus, terapija zvukom, mobilna aplikacija, audiometrija

Abstract

This thesis presents an original mobile application for tinnitus sound therapy based on standard diagnostic protocols and a recognized treatment method. The application includes three key modules: audiometric testing with audiogram display, detecting tinnitus determinants, and transforming audio files into a format suitable for tinnitus sound therapy. In addition, using standardized questionnaires, continuous monitoring of the impact of tinnitus on the patient's quality of life is supported. The server part of the system was also developed, which enables the collection and presentation of user data, as well as all measurement results generated through the mobile application.

Keywords — tinnitus, sound therapy, mobile application, audiometry