



Srednja šola za kemijo, elektrotehniko in računalništvo

ODDALJENO NADZIRANJE RAČUNALNIKA

Raziskovalna naloga

Mentor:

mag. Boštjan Resinovič , univ. dipl. inž. rač. inf.

Avtor:

Timotej Kroflič, R4-b

Mestna občina Celje, Mladi za Celje,

Celje, 2021

Zahvala

Zahvaljujem se mentorju mag. Boštjanu Resinoviču, za pomoč in vodenje pri delu, in vsem drugim, ki so pomagali pri nastanku te raziskovalne naloge.

Kazalo vsebine

Kazalo slik.....	II
Kazalo grafov.....	III
Povzetek.....	IV
Abstract.....	IV
1 Uvod.....	1
1.1 Cilji raziskovalne naloge	1
1.2 Hipoteze	2
1.3 Raziskovalne metode	2
2 Delovno okolje	3
2.1 Programski jeziki in knjižnice.....	3
2.1.1 Windows Core Audio APIs	3
2.1.2 WinSock in Socket API.....	4
2.1.3 Bluetooth API	4
2.1.4 Android SDK	4
3 Teoretični del	5
3.1 Nizko nivojsko delovanje zvočnih sistemov	5
3.2 TCP/IP	6
3.3 Omrežne vtičnice in prenos podatkov	7
3.4 Bluetooth.....	9
3.5 Opis miške	9
3.6 Opis tipkovnice	9
4 Potek raziskovalnega dela.....	10
4.1 Upravljanje glasnosti v sistemu Windows.....	10
4.1.1 Sistemski dogodki naprav	10
4.1.2 Vmesniki avdio naprav in upravljanje zvoka.....	11
4.1.3 Vmesniki avdio sej in njihovo upravljanje.....	11
4.1.4 Sprememba privzete naprave.....	13
4.1.5 Sprememba avdio naprave seje.....	13
4.2 Miška in tipkovnica.....	13
4.3 Omrežni protokol	13

4.3.1	Snovanje omrežnega protokola	13
4.3.2	Koraki protokola na strežniku	14
4.3.3	Izgradnja paketa protokola	15
4.3.4	Serializacija.....	15
4.3.5	Operacije protokola	15
4.3.6	Težave, ki so se pojavile	16
4.4	Izgled in funkcionalnost daljinca na pametnem telefonu	16
5	Analiza ankete.....	19
5.1	Ugotovitve ankete	24
6	Analiza hipotez.....	24
7	Zaključek	25
8	Viri in literatura	26
9	Priloga	28
9.1	Anketni vprašalnik.....	28

Kazalo slik

Slika 1:	Diagram nizko nivojske zgradbe.	5
Slika 2:	Enkapsulacija podatkov.	7
Slika 3:	Trostopenjsko usklajevanje (3-way handshake).	8
Slika 4:	Funkcije IMMNotificationClient.	10
Slika 5:	Volume Mixer (SndVol.exe).....	12
Slika 6:	Funkcije IAudioSessionEvents.....	12
Slika 7:	Diagram poteka delovanja strežnika	14
Slika 8:	Skica izgleda prijavnne strani.....	16
Slika 9:	Skica podmenija za upravljanje glasnosti	17
Slika 10:	Skica podmenija sledilne tablice.....	18
Slika 11:	Skica podmenija mešalnika glasnosti.	18

Kazalo grafov

Graf 1: Želja po upravljanju računalnika preko pametnega telefona.....	19
Graf 2: Želje po funkcijah aplikacije.....	20
Graf 3: Želja po centraliziranem upravljanju.....	20
Graf 4: Upravljanje preko Bluetooth povezave.....	21
Graf 5: Operacijski sistem na računalniku.....	21
Graf 6: Operacijski sistem na pametnem telefonu.....	22
Graf 7: Poznavanje IP naslovov in vrat.....	22
Graf 8: Spol vprašanih.....	23
Graf 9: Starost vprašanih.....	23

Povzetek

Raziskovalna naloga je pot do razvoja dveh kosov programske opreme, ki skupaj tvorita en izdelek. Ta izdelek je daljinec za računalnik z operacijskim sistemom Windows, na katerega naložiš program, ki omogoča aplikaciji na pametnem telefonu s platformo Android, da se nanj poveže in preko njega upravlja računalnik. Glavnina raziskave se usmerja v komunikacijo med dvema napravama po lokalnem omrežju in Bluetooth povezavi, del pa še na delovanje knjižnic operacijskega sistema Windows, ki omogočajo upravljanje računalnikov s tem operacijskim sistemom. Daljinec omogoča upravljanje z glasnostjo zvočnih naprav, premikanje miške, vnos preko tipkovnice na telefonu in še nekatera manjša opravila, ki povečajo funkcionalnost.

Ključne besede: daljinec, oddaljeno nadziranje, omrežje, Bluetooth

Abstract

This research paper is about the development of two separate parts of software, which together represent one product. That product is a remote control for a computer running the Windows operating system that can be controlled through an app on a smartphone running the Android operating system. This app connects to the program on the computer and provides remote control capabilities. Most of the research focuses on the communication between two devices on the same local network and communication through a Bluetooth connection. Other research is focused on the functioning of Windows operating system libraries that provide programmatic control of the computer. The remote control provides control of audio devices connected to the computer, mouse movement and keyboard input all on the smartphone app. It also includes some smaller features that enhance functionality.

Keywords: remote, remote control, network, Bluetooth

1 Uvod

Vedno več ljudi uporablja računalnik za ogled filmov in poslušanje glasbe. Nekaterim, predvsem uporabnikom prenosnih računalnikov, je lahko njihov računalniški zaslon premajhen, nima dovolj dobre ločljivosti, barvnega razpona ali česa drugega v čemer je boljša televizija, zato se odločijo gledati vsebine na televiziji, na katero priklopijo računalnik. To lahko onemogoči priročen in hiter dostop do računalnika in posledično do nadzora predvajanja, kot so pavza/predvajanje in glasnost. V tej situaciji sem se našel sam in sem si zaželel nekakšen daljinec za računalnik – podobno kot ga imajo televizije in tako imenovani set-top boxi –, ki bi bil kot aplikacija na telefonu, ker ga sedaj imamo vsi. Ko sem pobrskal po internetu, nisem nikjer našel rešitve, ki je takšna, kot sem si jo želel, zato sem se odločil narediti to aplikacijo sam.

Čeprav obstaja veliko programov, ki omogočajo oddaljeni nadzor računalnika preko pametnega telefona, nimajo preprostega uporabniškega vmesnika, saj za to niso bili narejeni. Večina jih, kadar se povežeš na računalnik, zatemni zaslon in ga preusmeri na mobilni telefon, kar ni zaželeno, kadar na računalniku gledaš vsebino.

Nekateri predvajalniki videa in glasbe imajo svoje aplikacije, ki to omogočajo, ampak je mogoče upravljati samo ta program in ne celotnega sistema. Nekateri programi sploh nimajo takšnih aplikacij. Uporabniku v takšni situaciji preostane samo še upravljanje celotnega sistema, kar želi ta aplikacija doseči.

1.1 Cilji raziskovalne naloge

Cilj te raziskovalne naloge je narediti aplikacijo, ki bi rešila vse probleme, ki so bili naštet, in dodati še nekaj funkcij, kot so premikanje miške, vnos teksta preko tipkovnice na telefonu, upravljanje glasnosti posameznega programa, če bo to mogoče, in še upravljanje preko Bluetooth povezave, ker računalnik ali telefon nista vedno povezana ali v istem omrežju ali pa sta povezana v javno omrežje, ki tipično izolira vsako napravo od drugih.

Za izvedbo teh zahtev je bilo potrebno raziskati delovanje upravljanja glasnosti v sistemu Windows, da bo mogoče upravljati glasnost sistema in avdio naprav, delovanje miške in tipkovnice za oddaljeno premikanje miške in vnos teksta, raziskati delovanje omrežja, omrežnih vtičnic (angl. socket) in prenosa podatkov med napravami z lastnim aplikacijskim omrežnim protokolom za komunikacijo, ker omrežnega protokola, ki bi bil narejen za takšen tip komunikacije, ni. Želel bi, da ta protokol deluje tudi preko Bluetooth povezave. Na koncu sem še želel poizvedeti od javnosti, če je ta aplikacija sploh zaželeno in uporabna v življenju drugih.

1.2 Hipoteze

Hipoteze so usmerjale razmišljanje in pot pri razvoju aplikacije, ker so postavile cilje, zahteve, način dela in razmišljanja o delovanju aplikacije. Postavljene so bile naslednje hipoteze:

1. Implementacija in razvoj lastnega aplikacijskega omrežnega protokola za komunikacijo je zahtevna
2. Večina anketirancev uporablja Windows in android operacijska sistema, zato bi bilo najbolj smiselno razviti android aplikacijo za nadzorovanje računalnika z operacijskim sistemom Windows.
3. Upravljanje računalnika preko pametnega telefona je zaželeno.
4. Večina anketirancev se ne spozna na računalnike in omrežja dovolj dobro, da bi vedeli, kaj so IP naslovi in vrata.

1.3 Raziskovalne metode

Za raziskovanje sem uporabil več raziskovalnih metod, najbolj pomembno je bilo raziskovanje po internetu, ker literature v fizični obliki ni. Glavni viri so bili dokumentacija podjetja Microsoft o njihovih knjižnicah operacijskega sistema Windows, dokumentacija Android operacijskega sistema in Android SDK-ja za razvoj aplikacije na pametnem telefonu, poleg tega pa še dokumentacija Socket API za delovanje funkcij, povezanih z omrežnimi vtičnicami. Za razumevanje delovanja transporta podatkov po omrežju in kako obdelati podatke, preden jih pošljemo ali prejmemo iz omrežja, je zelo pomagal priročnik Beej's Guide to Network Programming.

Potrebno je bilo preštudirati in prebrati dokumentacijo sistema Windows za upravljanje zvoka, ki se ji reče Windows Core Audio APIs, da sem razumel, kako stvari delujejo v sistemu Windows in na kakšen način. Potrebno je bilo tudi preštudirati delovanje omrežnih vtičnic za prenos podatkov med sistemi na omrežju in preko Bluetooth povezave. Med programiranjem sem moral še prebrskati dokumentacijo API sistema Windows, WinSocket in Socket API ter Bluetooth API. Po programiranju je bilo potrebno testirati delovanje in uloviti ter popraviti nepravilnosti v delovanju. Na koncu razvoja osnovne funkcionalnosti programa sem izvedel anketo o potencialih dodatnih funkcijah, ki si jih želijo ali jih potrebujejo uporabniki.

Največ časa je vzela analiza in študij delovanja drugih podobnih protokolov, ki so vplivali na načrtovanje izgradnje in delovanje lastnega omreženega protokola za komunikacijo, saj mora biti robusten, zanesljiv in lahko razširljiv.

2 Delovno okolje

Za razvoj backenda (del na računalniku) sem uporabil Visual Studio 2019, saj je najboljši za razvijanje programov, ki ciljajo operacijske sisteme Windows. Uporabil sem tudi knjižnice operacijskega sistema Windows

Za razvoj Android aplikacije (frontend) sem uporabil Android Studio, ker je najbolj uporabljen in najboljši za razvijanje aplikacij za Android in Android SDK, ki ponuja razrede in funkcije za razvoj Android aplikacij.

2.1 Programski jeziki in knjižnice

Za backend sem uporabil C++, čeprav je bil C# favorit in zaželena opcija, ker ponuja Garbage Collector, ki omogoča avtomatsko čiščenje pomnilnika, ki ni več v uporabi (v C++ mora to programer narediti sam). Ponuja tudi obsežen framework, ki je zelo stabilen in omogoča visoko kompatibilnost z operacijskim sistemom Windows.

C# na koncu ni bil izbran, ker so se v preizkusni uporabi pojavili veliki problemi z marshallingom, nedovoljenim dostopom do pomnilnika in še razne druge funkcionalnosti, ki omogočajo delovanje z nizko nivojskimi rutinami operacijskega sistema. Ti problemi se pri C++ niso pojavili, ker so omenjeni APIji namenjeni in prilagojeni za uporabo z jezikom C++.

2.1.1 Windows Core Audio APIs

Je zbirka API-jev za upravljanje z vhodnimi in izhodnimi zvočnimi napravami ter avdio sejami. Vmesniki in funkcije v teh zbirkah knjižnic omogočajo:

- enumeracijo skozi vse zvočnice naprave,
- prijavo na dogodke, povezane s temi napravami,
- prijavo na dogodke sprememb glasnosti na napravah,
- upravljanje glasnosti teh naprav,
- pridobitev podatkov o zvočnih napravah,
- zaznavanje novo ustvarjenih zvočnih sej,
- prijavo na dogodke zvočnih sej,
- pridobitev podatkov o zvočnih sejah,
- upravljanje glasnosti zvočnih sej.

To je samo nekaj najpomembnejših funkcij, ki so hkrati uporabljene v aplikaciji, pa tudi najbolj raziskane, ker so najbolj pomembne za načrtovano delovanje aplikacije.

Knjižnice, ki so uporabljene iz te zbirke:

- MMDevice API (za dogodke naprav)

Ta API omogoča odkritje avdio naprav, kot so zvočniki in slušalke, in prijavo na sistemske dogodke, povezane z avdio napravami. Poleg tega omogoča še enumeracijo skozi naprave in dostop do njihovih lastnosti.

- EndpointVolume API (za upravljanje nivoja zvoka na napravi)

Ponuja vmesnike in funkcije za upravljanje z glasnostmi zvočnih naprav in vsemi lastnostmi, povezanimi z njimi.

- Windows Audio Session API (za upravljanje glasnosti avdio sej in njihovi dogodki)

Vsebuje vmesnike za dogodke avdio sej in njihove lastnosti, možnost upravljanja s sejami in informacije o njih. Ponuja tudi ustvarjanje novih sej na napravah, ampak to nikoli ni bil cilj tega projekta.

2.1.2 WinSock in Socket API

Zbirka funkcij, ki se uporabljajo, kadar želi aplikacija delati na omrežju. Ta knjižnica omogoča razne stvari, ampak najpomembnejše so: ustvarjanje omrežnih vtičnic in njihovo uničenje, pošiljanje in sprejemanje podatkov, spreminjanje njihovih lastnosti delovanja in pretvarjanje številskih vrednosti, ki so na različnih računalniških arhitekturah drugače predstavljene. V sistemu Windows se Socket API imenuje WinSock API, čeprav ima enako delovanje kot Socket API, vsebuje še ogromno dodatnih funkcij, ki so specifične za Windows operacijski sistem.

2.1.3 Bluetooth API

Knjižnica, ki omogoča upravljanje z Bluetooth adapterjem na računalniku. Ponuja veliko funkcij in vmesnikov, ki omogočajo vzpostavitev osebnega omrežja PAN preko tehnologije Bluetooth in vzpostavitev povezave z drugimi napravami. Omogoča tudi prenos podatkov kar preko omrežnih vtičnic (Socket API).

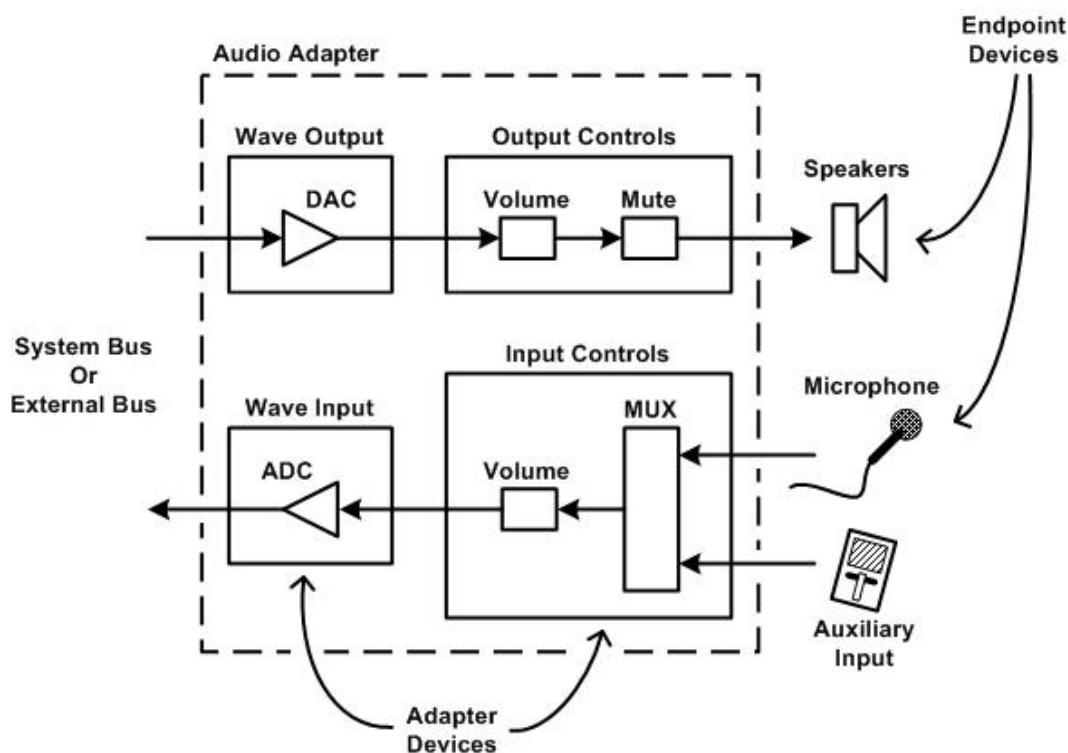
2.1.4 Android SDK

Je framework, ki se uporablja za razvijanje Android aplikacij z programskim jezikom Java ali Kotlin. Podpira vse verzije Android operacijskega sistema in ponuja ogromno funkcij, ki naredijo razvijanje aplikacij lažje. V skladu z Android Studiom ponuja tudi močno platformo in orodje za ustvarjanje grafičnih vmesnikov.

3 Teoretični del

3.1 Nizko nivojsko delovanje zvočnih sistemov

Za razumevanje delovanja vmesnikov in funkcij za upravljanje z zvočnimi napravami je potrebno najprej razčistiti, kako sploh delujejo ti sistemi v fizični obliki, katere komponente jih sestavljajo, čemu služijo, kako so povezane med sabo in kako jih upravljati.



Slika 1: Diagram nizko nivojske zgradbe (vir: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/images/devices.jpg>).

Zgornji diagram prikazuje, kako delujejo zvočni sistemi na nizkem nivoju in kako so povezani med sabo. Ta diagram velja pri vseh digitalnih napravah, ki imajo opravka z vhomom ali izhodom zvoka. Zvok je namreč analogni signal, ki ga je treba pretvoriti v digitalno obliko, da lahko računalnik z njim upravlja. Če gre za vhodne naprave, kot so mikrofoni, to naredimo z ADC (Analog-to-Digital Converter) pretvornikom. Kadar gre za izhodne naprave pa moramo pretvoriti signal iz digitalne oblike v analogno, kar se naredi z DAC (Digital-to-Analog Converter) pretvornikom. Najpomembnejše komponente zvočnega sistema so:

- **Endpoint devices** (končne naprave)
Končne naprave, ki pretvorijo električne signale v zvočno valovanje ali obratno
- **Input or Output Controls** (vhodne ali izhodne kontrole)
Ta vezja prilagajajo glasnost in se odločijo, ali se naj analogni signal sploh predvaja na končni napravi, kadar gre za izhodne kontrole.
Vhodne kontrole imajo tudi možnost prilagajanja glasnosti, imajo pa tudi MUX vezje, ki lahko združi signale več vhodnih naprav v enega, odloča tudi, katere signale naprav bo skupaj združil v glavni signal.

- **Audio adapter** (avdio adapter)

To je vezje, ki upravlja z električnim signalom, da pride do ali iz naprave takšen, kot želi uporabnik; gre za glasnost, mešanje več signalov, če je priklopljenih več naprav, utišano ali ne. Te upravljajo vezja, ki so na sliki poimenovana Adapter Devices ali po slovensko naprave adapterja.

- **System or External Bus** (sistemsko ali zunanje vodilo)

Sistemsko vodilo služi za prenos podatkov do vezij avdio adapterjev, ali iz njih. Po njem se prenašajo podatki v digitalni obliki.

3.2 TCP/IP

Je internetni protokolni sklad, ki predstavlja konceptualni model in komunikacijske protokole interneta. Je de facto standard, kar pomeni, da je uporabljen v praksi in ni uradno standardiziran. Standardiziran pa je ISO/OSI model, ki je TCP/IP-ju zelo podoben. Največja razlika je, da je ISO/OSI bolj specifičen in ima zato tudi več plasti. Te plasti ISO/OSI modela TCP/IP združi v štiri bolj logično povezane. Tako se imenuje po dveh najbolj pomembnih protokolih TCP (Transmission Control Protocol) in IP (Internet Protocol). Razdeljen je v štiri plasti, vsaka služi svojemu namenu in ima svoje vzdrževalce in razvijalce. Te štiri plasti so:

- **Aplikacijska plast**

Vzdržuje jo razvijalec aplikacije. Pridobi podatke od uporabnika in jih v svoji zaželeni obliki pošlje naprej na transportno plast.

- **Transportna plast**

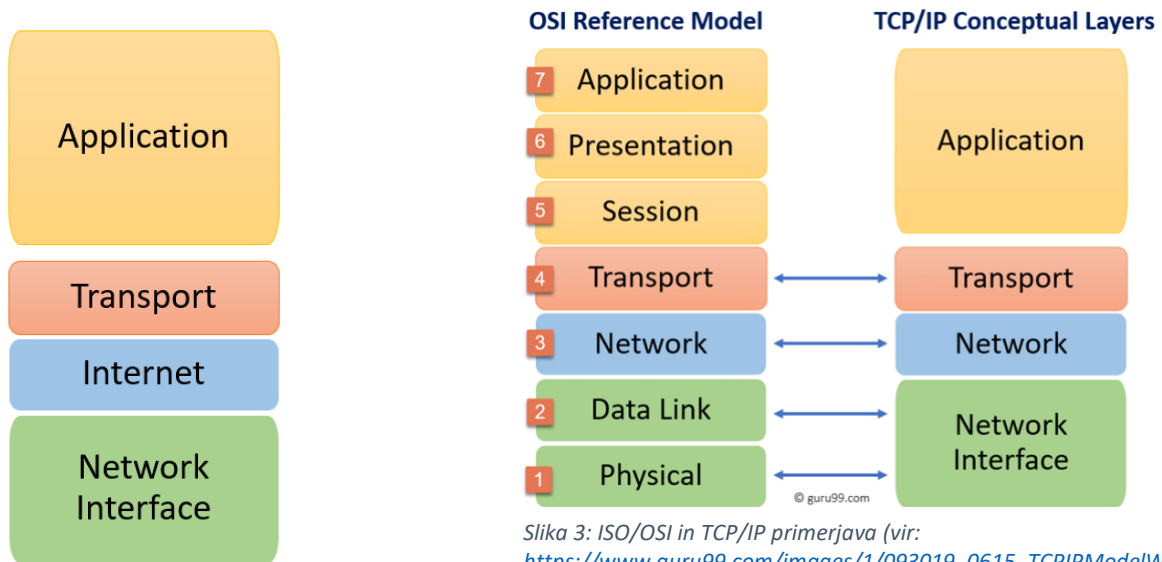
Vzdržuje jo razvijalec operacijskega sistema. Odgovorna je za zanesljiv transport podatkov in nadzor transporta. Svoje podatke, vključno s podatki aplikacijske plasti, pošlje internetni plasti.

- **Internetna plast**

Vzdržuje jo razvijalec operacijskega sistema. Odgovarja za premik paketov od izvira do cilja in se povezuje z logičnim internetnim omrežjem.

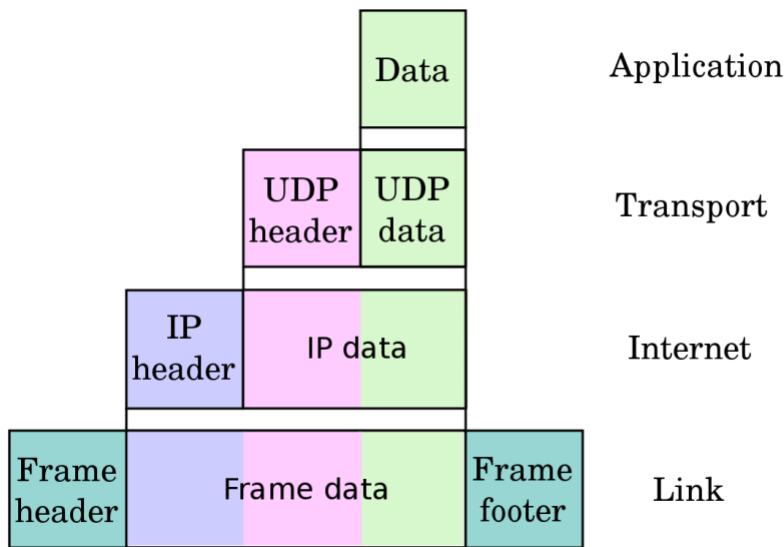
- **Fizična plast**

Vzdržujejo jo proizvajalci omrežnih kartic. Pristojna je za fizični prenos podatkov po mediju med dvema napravama.



Slika 2: TCP/IP plasti (vir: https://www.guru99.com/images/1/093019_0615_TCPIPModelW1.png)

Vsaka nižja plast doda višjim plastem še svoje podatke, s katerimi si pomaga pri delu. Temu principu se reče enkapsulacija in v praksi izgleda tako (spodnja slika):



Slika 2: Enkapsulacija podatkov (vir: https://upload.wikimedia.org/wikipedia/commons/3/3b/UDP_encapsulation.svg)

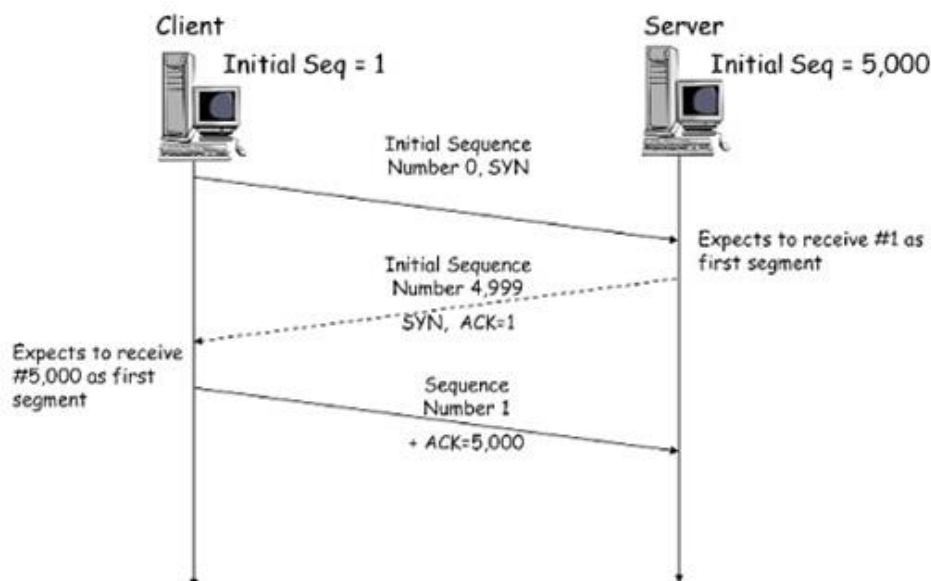
3.3 Omrežne vtičnice in prenos podatkov

Berkley Socket, kot se jim običajno reče, je API za internetne 'vtičnice', ki se uporabljajo za interprocesijsko komunikacijo, ki je lahko tudi med različnimi sistemi. Največkrat je implementirano kot knjižnica. Služi za prenos podatkov med procesi in različnimi fizičnimi napravami. Omrežna vtičnica, ki je številčni identifikator, predstavlja lokalno končno točko ali omrežno komunikacijsko pot, ki služi za dostop do tokov (angl. stream), ki vsebujejo vhodne in izhodne podatke.

Lokalna točka predstavlja kar nekaj stvari, na primer svoj naslov in vrata, naslov in vrata omrežne vtičnice na drugi strani, svoje lastnosti za komunikacijo, tip transportnega protokola, lastnosti IP naslova in še veliko drugih.

Ta API prvotno izvira iz BSD operacijskega sistema, ki so ga razvili na Univerzi v Kaliforniji kot odprtokodno različico omrežne implementacije Unix sistema. Ta API sedaj uporabljajo vsi moderni operacijski sistemi; tudi tisti, ki ga uradno ne, mu sledijo.

Pri vseh operacijskih sistemih sta zagotovljeni dve implementaciji protokolov na transportni plasti, to sta TCP (Transmission Control Protocol) in UDP (User Datagram Protocol). Prvi je pretočni in zagotavlja, da bodo podatki vedno prišli na cilj, tudi če se na poti izgubijo, in prihajali v vrstnem redu, tako kot so bili poslani. Je povezavni protokol, kar pomeni, da se je potrebno povezati z oddaljenim računalnikom in vzdržuje navidezno povezavo med njima. Nato je potrebno še izvesti tri-smerno usklajevanje, po angleško handshake, preden se lahko začnejo pošiljati podatki. To usklajevanje poteka v treh korakih in izgleda tako (spodnja slika).



Slika 3: Trostopenjsko usklajevanje (3-way handshake) (vir:

https://www.open.edu/openlearncreate/pluginfile.php/258370/mod_oucontent/oucontent/35227/9b0bce6c/594a9486/cn_black_fiq3.jpg)

Pri prvem koraku pošlje stranka strežniku SYN zahtevo, z njo pove, da se želi povezati in na katerih vratih posluša. V drugem koraku ji strežnik odgovori s SYN-ACK, s čimer potrdi, da je dobil zahtevo. Pri zadnjem koraku stranka pošlje odgovor, da je prejela sporočilo. Na koncu sta oba pripravljena za pošiljanje podatkov.

UDP pa nima povezave in ne zagotavlja vsega, kar zagotavlja TCP, je pa veliko hitrejši in lažji za obravnavo sporočil, saj vsak datagram sprejeme posebej in ni potrebno imeti sporočilnih ločil, ki povejo, kje se končajo podatki. Ta protokol služi predvsem za pošiljanje podatkov, pri katerih si lahko privoščimo občasne izgube, napačno dostavo, podvojene podatke in izven podatke izven vrste pošiljanja.

3.4 Bluetooth

Je brezžična tehnologija, ki je prvič izšla leta 1989. Omogoča povezavo ter prenos podatkov preko radijskih valov med različnimi napravami. Omogoča postavitve PAN (Personal Area Network) omrežij, ki imajo doseg od 10 do 100 metrov, odvisno od verzije Bluetooth vezja na napravah. Deluje v frekvenčnem območju od 2,402 GHz do 2,48 GHz, ki ga uporablja za pošiljanje podatkov preko kanalov, ki so v tem območju.

Bluetooth je industrijsko-standardni protokol, ki ga uporablja veliko naprav, ki potrebujejo poceni, nizkoenergijsko in svetovno sprejeto možnost brezžične povezave. Ta je na voljo za ogromno naprav, kot so mobilni telefoni, računalniki, tiskalniki, zvočniki in še mnogo drugih. Ima tudi dobro definirano dokumentacijo in programske vmesnike, ki so že znani večini razvijalcev programske opreme, kar jim omogoča hitro razvijanje aplikacij.

Razvija ga skupina Bluetooth SIG, ki je sestavljena iz podjetij, univerz in različnih ustanov, ki se ukvarjajo z razvojem brezžičnih tehnologij. Ta skupina redno posodablja in izboljšuje tehnologijo.

Za Bluetooth povezavo potrebuješ vsaj dve napravi, ki ga podpirata. Ker Bluetooth omogoča način skrivanja naprave, mora biti vsaj ena naprava vidna, da se druga naprava lahko z njo poveže. Tista naprava, ki je skrita, mora nato poslati vidni napravi zahtevo za povezavo, ki nato zahteva potrditev povezave od uporabnika te naprave, če se res želi povezati z drugo napravo. Če pritisne DA, se napravi povežeta in je med njima vzpostavljen komunikacijski kanal. Če pritisne NE, se ne povežeta.

3.5 Opis miške

Miška je kazalna vhodna naprava za računalnik, ki zaznava svoj premik v 2-dimenzionalnem prostoru. Poleg tega vsebuje še gumbe in kolesček, ki razširijo njeno funkcionalnost. Ti gumbi se uporabljajo za klikanje, medtem ko se kolesček uporablja za pomikanje, kjer je potreben linearni enodimenzionalni premik.

Miška na nizkem nivoju zaznava svoje premike in jih pošlje računalniku, ti premiki pa so predstavljeni kot dve pozitivni ali negativni številki, ki povesta, kako hitro se je miška prestavila v prostoru. Hkrati se še vedno pošlje status gumbov in ali so pritisnjeni ali ne.

3.6 Opis tipkovnice

Tipkovnica je vhodna naprava, ki računalniku pošilja kode tipk, ki jih je pritisnil uporabnik. Njena največja uporabnost je vnosa teksta in ukazov, saj pritisk določene tipke pomeni en znak ali ukaz.

Na nižjem nivoju deluje tako: uporabnik pritisne tipko, katere kodo prebere mikrokontroler in številko tipke pošlje računalniku, ki jo interpretira tako, kot je sprogramiran. Tak način delovanja omogoča veliko fleksibilnost in prilagajanje posebnim namenom.

4 Potek raziskovalnega dela

Preden sem začel z delom, je bilo potrebno raziskati, kako sploh deluje upravljanje in nadzor glasnosti v sistemu Windows, potem je bilo potrebno raziskati, kako deluje prenos po omrežju in Bluetooth povezavi in ugotoviti, kako deluje prenos podatkov. To so bile najpomembnejše raziskave za osnovno idejo aplikacije.

4.1 Upravljanje glasnosti v sistemu Windows

4.1.1 Sistemski dogodki naprav

Za prijavo na dogodke je potrebno implementirati vmesnik IMMNotificationClient. Eden najlažjih načinov je, da razred deduje od njega. Tako je narejeno tudi v razredu MMEvents (spodnja slika):

```
virtual HRESULT __stdcall OnDeviceStateChanged(LPCWSTR pwstrDeviceId, DWORD dwNewState) override;
virtual HRESULT __stdcall OnDeviceAdded(LPCWSTR pwstrDeviceId) override;
virtual HRESULT __stdcall OnDeviceRemoved(LPCWSTR pwstrDeviceId) override;
virtual HRESULT __stdcall OnDefaultDeviceChanged(EDataFlow flow, ERole role, LPCWSTR pwstrDefaultDeviceId) override;
virtual HRESULT __stdcall OnPropertyValueChanged(LPCWSTR pwstrDeviceId, const PROPERTYKEY key) override;
```

Slika 4: Funkcije IMMNotificationClient.

To so funkcije vmesnika IMMNotificationClient. Te funkcije je potrebno pozoviti in napisati svojo implementacijo.

OnDeviceStateChanged

Se sproži, kadar se spremeni status naprave. Vsaka naprava ima namreč več možnih stanj: aktivna, onemogočena, neaktivna in izklopljena.

OnDeviceAdded in OnDeviceRemoved

Sprožena je kadar je naprava dodana (added) ali odstranjena (removed) iz sistema.

OnDefaultDeviceChanged

Poklicana je kadar se spremeni privzeta naprava predvajanja ali snemanja. Privzeta naprava je tista, na kateri se bo privzeto predvajal ali snemal zvok brez predhodne konfiguracije.

OnPropertyValueChanged

Sprožena je kadar se spremeni lastnost naprave. Teh lastnosti je ogromno in nimajo vse naprave istih, nekatere dodajo tudi svoje.

Za želeno delovanje aplikacije potrebujemo vse funkcije razen OnPropertyValueChanged, ker nas ne zanimajo spremembe lastnosti na napravah. Potrebne so, ker je zahteva za aplikacijo zaznavanje spremembe aktivnosti ali neaktivnosti, saj upravljamo samo aktivne naprave. Zanima nas tudi, kdaj so dodane in odstranjene, da ne poskusimo upravljati z napravo, ki ne obstaja več.

Posebna pozornost je posvečena privzeti napravi, ker je ta največkrat uporabljena, uporabniki je tipično ne spreminjajo in sistem jo avtomatsko nastavi, kakor misli, da je najbolj zaželeno za uporabnika.

Dejanska prijava na dogodke pa zahteva klic funkcije, ki je v vmesniku IMMDeviceEnumerator. Ta vmesnik poleg registriranja dogodkov naprav omogoča tudi enumeracijo skozi avdio naprave določenega tipa.

4.1.2 Vmesniki avdio naprav in upravljanje zvoka

Avdio naprave imajo vmesnik IMMDevice, ki vsebuje funkcije za upravljanje raznih lastnosti, pridobitev unikatnega identifikatorja in status naprave, indirektno tudi upravljanje glasnosti. Dobimo ga iz IMMDeviceEnumerator na dva načina. Prvi način je možen, če poznamo unikatni identifikator te naprave; takrat uporabimo funkcijo GetDevice, ki vrne vmesnik naprave. Drugi pa je preko enumeracije naprav s funkcijo EnumAudioEndpoints in IMMDeviceCollection (zbirka naprav), ki jo vrne enumeracija. Ta zbirka ima funkcijo GetItem, ki vrne IMMDevice.

Sintaksa Activate funkcije, ki je uporabljena v kodi:

```
IMMDevice.Activate(__uuidof(/*ime vmesnika*/), CLSCTX_ALL, 0, (void**)&spremenljivkaVmesnika)
```

Za upravljanje glasnosti naprave moramo od funkcije Activate pridobiti vmesnik IAudioEndpointVolume, ki omogoča obširno upravljanje glasnosti na napravi. Ta vmesnik vsebuje tudi pomembno funkcijo RegisterControlChangeNotify, ki omogoča prijavo vmesnika IAudioEndpointVolumeCallback, ki vsebuje funkcijo OnNotify. Prejšnja funkcija, kadar je registrirana, se sproži, kadar se spremeni glasnost na točno tej napravi, zraven pa sporoči, na katero stopnjo se je spremenila glasnost in tudi, ali je bilo nastavljeno utišanje.

IAudioEndpointVolume vsebuje še ostale za to aplikacijo pomembne funkcije:

- SetMasterVolumeLevelScalar (nastavi glasnost med 1 in 100),
- GetMasterVolumeLevelScalar (pridobi glasnost med 1 in 100),
- SetMute (nastavi utišanje),
- GetMute (pridobi stanje utišanja),
- VolumeStepDown (zniža glasnost za korak, odvisen od naprave),
- VolumeStepUp (zviša glasnost za korak, odvisen od naprave).

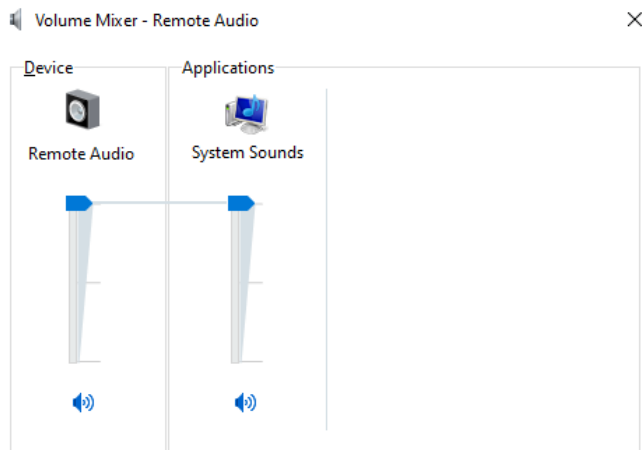
Preko teh funkcij se bo glasnost naprav programsko spreminjala in pridobivala.

Vsi ti vmesniki so implementirani v razredu MMDevice, ki samostojno skrbi za vse potrebne zadeve, povezane z napravo.

4.1.3 Vmesniki avdio sej in njihovo upravljanje

Avdio seja je skupina sorodnih avdio pretokov, s katerimi upravlja neka stranka. Ta stranka mora ustvariti avdio sejo na napravi. Na avdio seji se lahko upravlja glasnost in utišanje. Vsaka avdio seja pripada procesu, ki jo ustvari in z njo upravlja. Do teh sej lahko dostopamo kar v

sistemu Windows in pri vsaki posebej upravljamo z glasnostjo in utišanjem. Temu programu se reče Audio Mixer (SndVol.exe) in izgleda tako:



Slika 5: Volume Mixer (SndVol.exe) (vir: <https://strontic.github.io/xcyclopedia/library/screenshots/SndVol.exe-7D7D5466FCDCD28976A004B5B08864E3-1.png>).

Za dostop do seje je potrebnih več vmesnih korakov in vmesnikov. Najprej je potrebno pridobiti vse seje, ki so bile ustvarjene na napravi. Za to je potreben vmesnik `IAudioSessionManager2` (menedžer avdio seje), ki poleg enumeracije avdio seje na napravi omogoča tudi prijavo na dogodek ustvarjene nove avdio seje. Ta vmesnik se pridobi s klicem na funkcijo `Activate` v vmesniku `IMMDevice`, saj ima vsaka napravo samo enega menedžerja avdio seje, ki upravlja z njimi. Avdio manager je implementiran v razredu `AudioSessionManager`. Ta razred ima tudi polje avdio seje, ki temu menedžerju pripadajo.

Dejansko upravljanje z avdio sejami in njihovimi dogodki je izvedeno preko vmesnika `IAudioSessionControl2`, ki omogoča nastavitve lastnosti avdio seje in registracijo vmesnika `IAudioSessionEvents` na dogodke seje, preko funkcije `RegisterAudioSessionNotification`. `IAudioSessionEvents` vsebuje naslednje funkcije:

```
virtual HRESULT __stdcall OnDisplayNameChanged(LPCWSTR NewDisplayName, LPCGUID EventContext) override;
virtual HRESULT __stdcall OnIconPathChanged(LPCWSTR NewIconPath, LPCGUID EventContext) override;
virtual HRESULT __stdcall OnSimpleVolumeChanged(float NewVolume, BOOL NewMute, LPCGUID EventContext) override;
virtual HRESULT __stdcall OnChannelVolumeChanged(DWORD ChannelCount, float NewChannelVolumeArray[], DWORD ChangedChannel, LPCGUID EventContext) override;
virtual HRESULT __stdcall OnGroupingParamChanged(LPCGUID NewGroupingParam, LPCGUID EventContext) override;
virtual HRESULT __stdcall OnStateChanged(AudioSessionState NewState) override;
virtual HRESULT __stdcall OnSessionDisconnected(AudioSessionDisconnectReason DisconnectReason) override;
```

Slika 6: Funkcije `IAudioSessionEvents`.

- `OnDisplayNameChanged` (spremenilo se je ime)
- `OnIconPathChanged` (spremenila se je ikona)
- `OnSimpleVolumeChanged` (spremenila se je glasnost)
- `OnChannelVolumeChanged` (spremenila se je glasnost na enem kanalu)
- `OnGroupingParamChanged` (spremenil se je skupinski parameter)
- `OnStateChanged` (spremenilo se je stanje seje)
- `OnSessionDisconnected` (seja je bila uničena)

Od teh so tukaj implementirane samo `OnSimpleVolumeChanged`, `OnGroupingParamChanged`, `OnStateChanged` in `OnSessionDisconnected`. Druge niso zanimive in jih bomo ignorirali.

4.1.4 Sprememba privzete naprave

Včasih si uporabnik zaželi spremeniti napravo, na kateri želi predvajati zvok. To je lahko na primer, kadar se sistem odloči, da bo zvok pošiljal na televizijo namesto na AUX konektor, na katerega je povezano boljše ozvočenje.

Ko sem raziskoval dokumentacijo `CoreAudioAPIs` nisem nikjer našel vmesnika ali funkcije, ki bi spremenila privzeto napravo, ampak sem videl narejeno v praksi. Tako je bilo potrebno veliko raziskovanja po internetu in izvirni kodi programa `EarTrumpet`, ki to omogoča, da sta sem našel vmesnika `IPolicyConfig` (Windows 7 in naprej) in `IPolicyConfigVista` (Windows Vista), ki sploh nista nikjer dokumentirana in jih je zato potrebno deklarirati lastnoročno. Oba pa vsebujeta potrebno funkcijo `SetDefaultEndpoint`, ki spremeni privzeto napravo na želeno.

4.1.5 Sprememba avdio naprave seje

Po veliko raziskovanja po internetu in izvirni kodi programa `EarTrumpet`, sem našel vmesnik `IAudioPolicyConfig` in funkcijo `SetPersistedDefaultAudioEndpoint`, ampak ni nikjer definiran in dokumentiran, zato implementacije tega v tej aplikaciji ni in s tem ni funkcije spreminjanja avdio naprave seje.

4.2 Miška in tipkovnica

Miška in tipkovnica sta zelo pomembni orodji za komunikacijo z računalnikom. Dovolita nam, da računalniku dajemo ukaze in vnašamo podatke, zato je zaželeno, da lahko to delamo tudi na daljavo. Delovanje obeh je opisano že v teoretičnem delu. Potrebno je bilo raziskati, kako bi lahko programsko vnesli dogodke, za katere sicer rabimo fizično napravo. Izkazalo se je, da ima `WindowsAPI` več načinov. Dva najbolj uporabljena sta funkciji `mouse_event` (za miško) in `keybd_event` (za tipkovnico). Obe funkciji sta sedaj zastareli, nadomestila ju je funkcija `SendInput`, ki omogoča pošiljanje dogodkov vhodnih naprav in vnos več podrobnosti kot prejšnji dve funkciji, hkrati tudi ne zablokira programa, če trenutno ni prostora za pošiljanje dogodkov.

4.3 Omrežni protokol

4.3.1 Snovanje omrežnega protokola

Preden se je lahko začelo programiranje protokola, sem si postavil naslednje cilje, ki jih mora protokol zagotoviti:

- Podatki so lahko kadarkoli na voljo za pošiljanje ali sprejemanje na strežniku ali na stranki. Ta pogoj je najpomembnejši za delovanje tega protokola, ker lahko uporabnik kadarkoli na primer spremeni glasnost, izključi napravo, začne novo sejo predvajanja in podobno.

- Podpira več operacij in dejanj, ki jih stranka lahko pošlje strežniku.
- Dogodki morajo prihajati od enega k drugemu tako, kot so bili sproženi.
- Komunikacija mora biti obojesmerna in vedno na voljo.
- Vsi poslani podatki morajo priti na cilj, preden se povezava načrtovano zapre.
- Povezana je lahko samo ena stranka naenkrat, vse ostale zavrne.

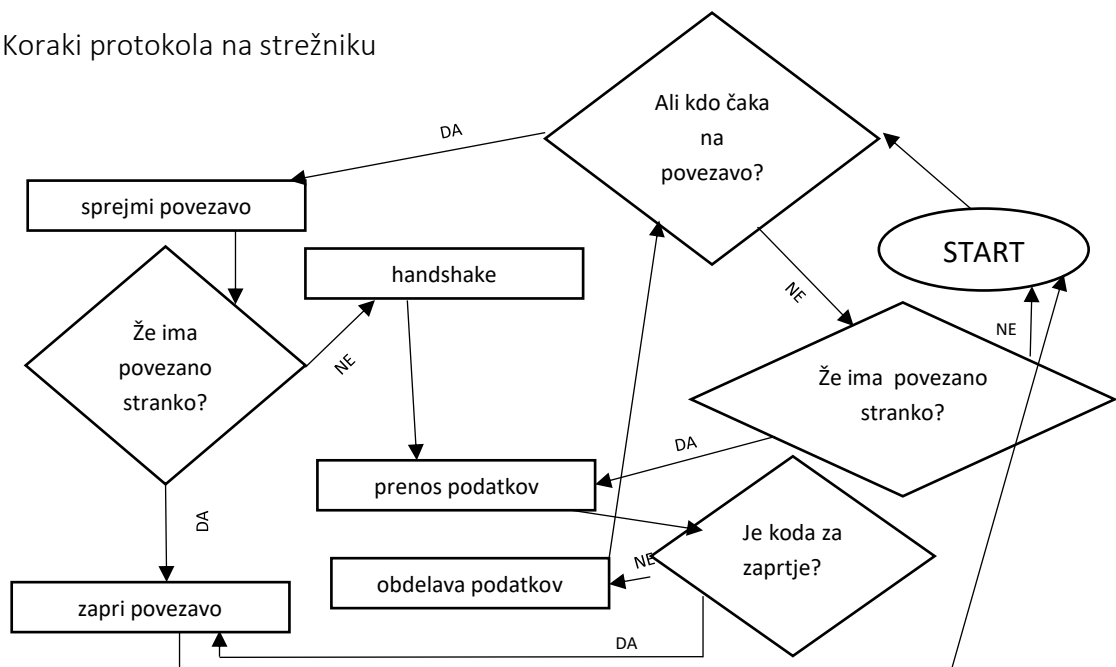
Iz teh zahtev lahko takoj razberemo, da je potreben transportni protokol, ki je zanesljiv in prenaša podatke po vrsti, zato bo izbran TCP protokol, ki omogoča te stvari.

Prvotno je bil načrt uporabiti UDP, ker ga je lažje implementirati kot sporočevalno orientiran protokol, ampak sem ga razmisleku opustil, ker ne zagotavlja, da pridejo podatki na cilj, in tudi ne, da pridejo po vrsti.

Protokol je binarnega in ne tekstovnega tipa, kot je HTTP. Razlogov za to je več:

- delo s stringi in tekstom zahteva več procesorskega časa, kar tukaj ni zaželeno;
- vsi sistemi nimajo istih znakovnih naborov, saj je za pravilno delovanje potreben črkovni nabor Unicode;
- velikokrat so podatki preveliki, ker en znak zasede 1 ali 2 bajta in jih je potrebno več, da povejo isto, kot bi na primer 1-bajtna številka, ki ima 256 različnih vrednosti. To je pomembno, ker mora ta protokol delovati s čim manjšimi količinami podatkov.

4.3.2 Koraki protokola na strežniku



Slika 7: Diagram poteka delovanja strežnika.

Zgornji diagram prikazuje, kako poteka omrežni del v strežniku.

4.3.3 Izgradnja paketa protokola

koda dejanja	velikost podatkov	podatki
--------------	-------------------	---------

Vsak paket protokola vsebuje operacijsko kodo (dejanje), ki je velikosti 1 B, kar omogoča 256 različnih dejanj ali operacij, nato sledi velikost podatkov, ki je 2 B število, ki pove, koliko bajtov veliki so podatki tega paketa. Na koncu so podatki posredovani še serializatorju, ki jih deserializira in vrne ustrezno podatkovno strukturo, ki vsebuje vse obdelane podatke.

Ker je protokol enkapsulirane narave, pomeni, da če želimo, lahko imamo še dodatno enkapsulacijo med podatki, s katero se ukvarjajo serializatorske funkcije.

4.3.4 Serializacija

Je namenjena večji organiziranosti, lažjemu dodajanju novih operacij v prihodnosti in zagotovi, da je vsa koda združena s podatki, za katere je odgovorna. Vsak razred, ki vsebuje podatke, je tudi odgovoren za svojo serializacijo in deserializacijo. Da je zagotovljeno vsebovanje teh dveh funkcij, morajo vsi razredi dedovati od razreda `network_serializable`. Ta razred vsebuje še dodatno funkcijo, ki pove, katero operacijsko kodo predstavlja podrazred tega razreda.

Kadar podatki pridejo v serializiranje, pridejo kot razred `network_serializable`, v katerem pokličemo ustrezno funkcijo. Funkcija za serializacijo vrne polje bajtov, ki vsebuje podatke, pripravljene za pošiljanje v omrežje, medtem ko funkcija za deserializacijo popularizira vse spremenljivke v razredu in je nato na voljo za uporabo drugje.

4.3.5 Operacije protokola

Vse možnosti, ki jih lahko upravljamo na napravi, morajo biti tudi prenesljive preko omrežja, kar pomeni, da mora vsaka pomembna operacija mora dobiti svojo kodo in podatke. Trenutno so najbolj pomembne naslednje operacije:

- zaprtje povezave,
- usklajevanje,
- javljanje napak,
- ping in odgovor nanj,
- podatki o napravah,
- dogodki, ki se zgodijo na strežniku,
- upravljanje glasnosti,
- premik miške,
- vnos preko tipkovnice.

To so posplošene operacije, ki bodo vedno na voljo z razvojem protokola. Ta se namreč spreminja iz verzije v verzijo, ampak te vedno ostanejo nespremenjene. Vse operacije, njihove kode in dokumentacija se nahajajo v dokumentaciji protokola, ki je sproti posodobljena.

4.3.6 Težave, ki so se pojavile

Pri razvoju protokola se je pojavilo ogromno težav. Največ jih je bilo pri sočasnem prenosu in pošiljanju podatkov. Najprej sem ustvaril novo nit (angl. thread) za sprejemanje strank, pošiljanje in sprejemanje podatkov. Vsako dejanje je imelo svojo nit. S takšno implementacijo se je pojavilo kup težav. Vsako nit je potrebno pravilno zaustaviti, kadar je to potrebno, kar pomeni, da je morala obstajati spremenljivka, ki je zahtevala pregled pravilnosti, preden se zanka ponovi. Če je izvajanje zaželeno, nadaljuje, drugače pa izvede pravilni zaključek. Če se nit ne zaključi pravilno (v primerih, kadar jo 'ubijemo'), povzroča veliko problemov z nesproščenim pomnilnikom in nepravilno zaprtimi sistemskimi viri. S to implementacijo je bilo tudi potrebno zagotoviti sinhroniziran dostop, sicer se pojavijo težave s prepisovanjem podatkov. Sinhronizacija zagotovi, da lahko samo ena nit naenkrat dostopa do bloka kode, druge morajo medtem čakati. S tem zagotovimo, da jih hkrati več ne prepisuje ene spremenljivke.

Po raziskovanju in testiranju se je izkazalo, da je bila taka rešitev prezapletena in čisto nepotrebna, saj se to lahko dovolj hitro izvede v samo eni niti, kar zelo olajša delo in zniža kompleksnost delovanja.

4.4 Izgled in funkcionalnost daljinca na pametnem telefonu

Kadar se aplikacija zažene, se pojavi prijavno okno, ki zahteva vnos IP naslova in vrat, v formatu *IP naslov: številka vrat*, na katerih posluša strežnik na računalniku. Nato je potrebno vnesti še geslo in pritisniti na gumb prijava (na sliki 'Log In')

IP address : port

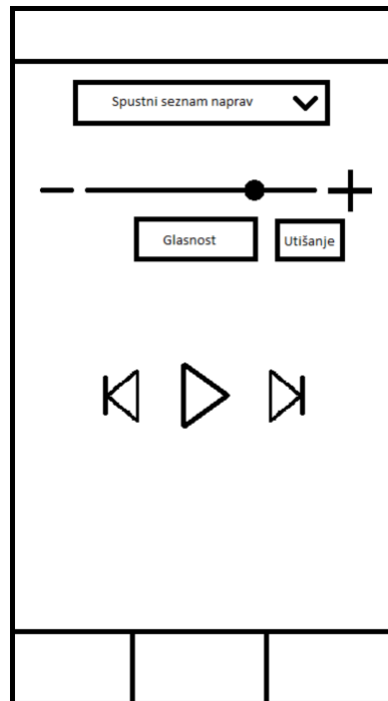
Password

LOG IN

Slika 8: Skica izgleda prijavnne strani.

Če uporabnik vse vnese pravilno, je prijavljen v aplikacijo in se mu bo prikazal prvi podmeni z vsemi drugimi podmeniji spodaj (spodnja slika).

Aplikacija na pametnem telefonu ima 3 podmenije. Glavni vsebuje predele za upravljanje z glasnostjo, privzeto napravo in stanje predvajanj. Na vrhu v opravilni vrstici je gumb za dostop do nastavitvev aplikacije, gumb za odjavo iz aplikacije in gumb za izklop računalnika.

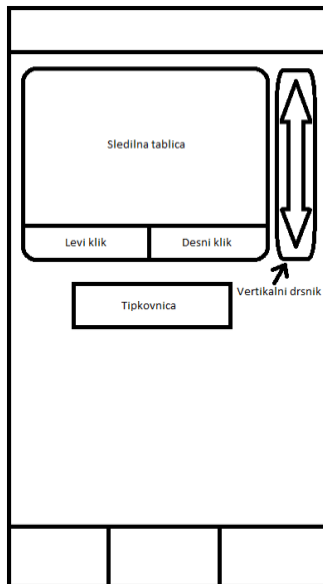


Slika 9: Skica podmenija za upravljanje glasnosti.

Sprememba privzete naprave se upravlja preko spustnega seznama, ki vsebuje vse naprave. Uporabnik samo izbere, katero napravo želi spremeniti v privzeto. Vedno je prikazano ime trenutne privzete naprave.

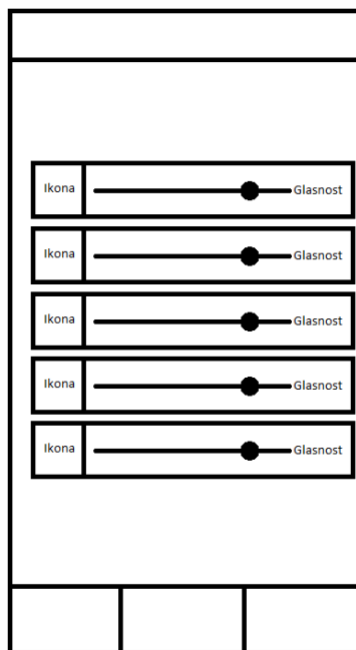
Glasnost se lahko upravlja na dva načina. Prvi način je upravljanje skozi gumba plus in minus. Plus poveča glasnost za en korak (sistemsko definiran), medtem ko minus zmanjša glasnost za korak. Drugi način je preko drsnika, ki ima mejo od 0 do 100, kar predstavlja skalarno glasnost v odstotkih. Trenutna glasnost je vedno izpisana v predelu, kjer piše glasnost, in je sinhronizirana z drsnikom. Ob glasnosti je gumb, ki upravlja s stanjem utišanja naprave.

Na sredini so kontrole za nadzor predvajanja. Te so predvajaj ali pavza, naprej in nazaj. Pod temi kontrolami je prikazano, kaj se trenutno predvaja na računalniku in kateri program to predvaja.



Slika 10: Skica podmenija sledilne tablice.

Drugi podmeni (zgornja slika) posnema sledilno tablico, ki deluje kot miška. Na spodnji strani ima še prostor razdeljen, na dva dela, levi služi kot levi klik, desni pa kot desni klik. Poleg tega je na desni strani še prostor, ki služi kot kolesček za vertikalno premikanje. Pod to tablico je še gumb, ki ob pritisku odpre okno, kamor se vpiše tekst, ki ga želimo poslati na računalnik.



Slika 11: Skica podmenija mešalnika glasnosti.

Tretji podmeni je namenjen upravljanju glasnosti posameznih avdio sej na privzeti napravi. V seznamu bodo nanizane vse seje, njihove ikone in glasnost od 0 do 100, ki se lahko spremenijo.

5 Analiza ankete

Anketa je potekala od 6. aprila 2021 do 13. aprila 2021 na spletni strani 1ka.arnes.si, anketo sem poslal med sošolce, prijatelje in družino. Anketo je izpolnilo 22 anketirancev, od tega je bilo pravilno izpolnjenih 13 anket.

Anketo sem izvedel, ker me je zanimalo, ali je takšna aplikacija sploh zaželena med uporabniki. Če je zaželena, me je zanimalo, kakšne so njihove zahteve in kakšen je nivo njihovega znanja, potrebnega za uporabo te aplikacije. S temi rezultati sem si pomagal pri izgledu in funkcionalnosti aplikacije na pametnem telefonu in računalniku.

Anketa je vsebovala 10 vprašanj, od tega 8 obveznih.

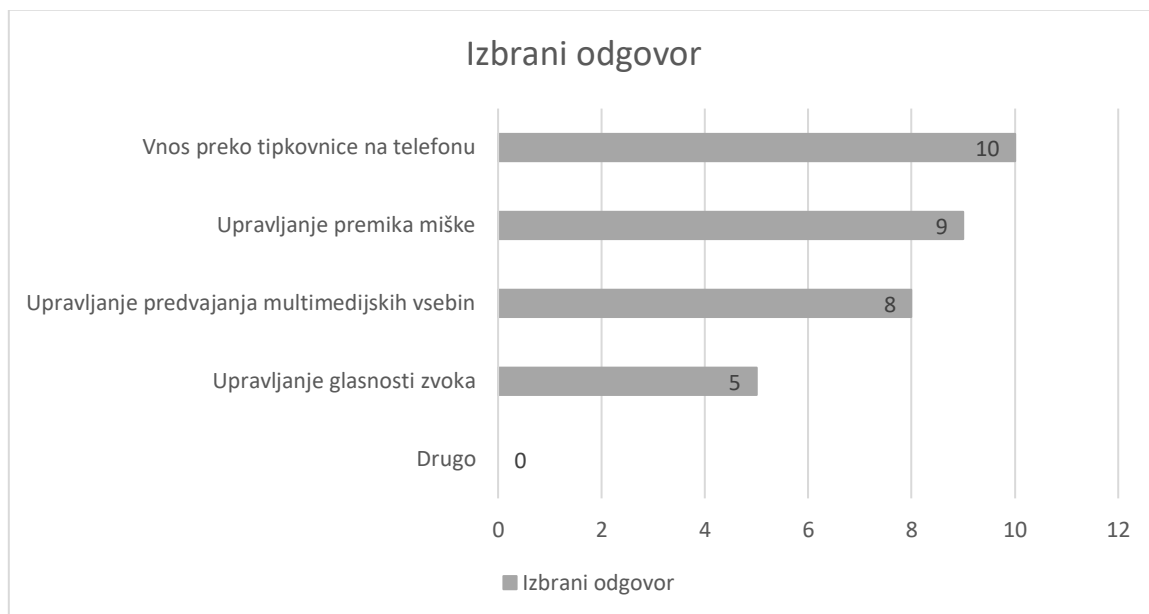
1. Ali ste si kdaj želeli upravljati računalnik preko pametnega telefona?



Graf 1: Želja po upravljanju računalnika preko pametnega telefona.

S tem vprašanjem sem želel izvedeti ali je takšna aplikacija sploh zaželena v realnem svetu. Izkazalo se je, da je zaželena pri 81 % vprašanih.

2. Kaj se vam zdi najbolj pomembno pri upravljanju računalnika na daljavo? (Možnih je več odgovorov)



Graf 2: Želje po funkcijah aplikacije.

Pri tem vprašanju sem povprašal po funkcijah, ki si jih želijo uporabniki. Očitno je, da si jih veliko želi vnos teksta na računalniku preko tipkovnice pametnega telefona.

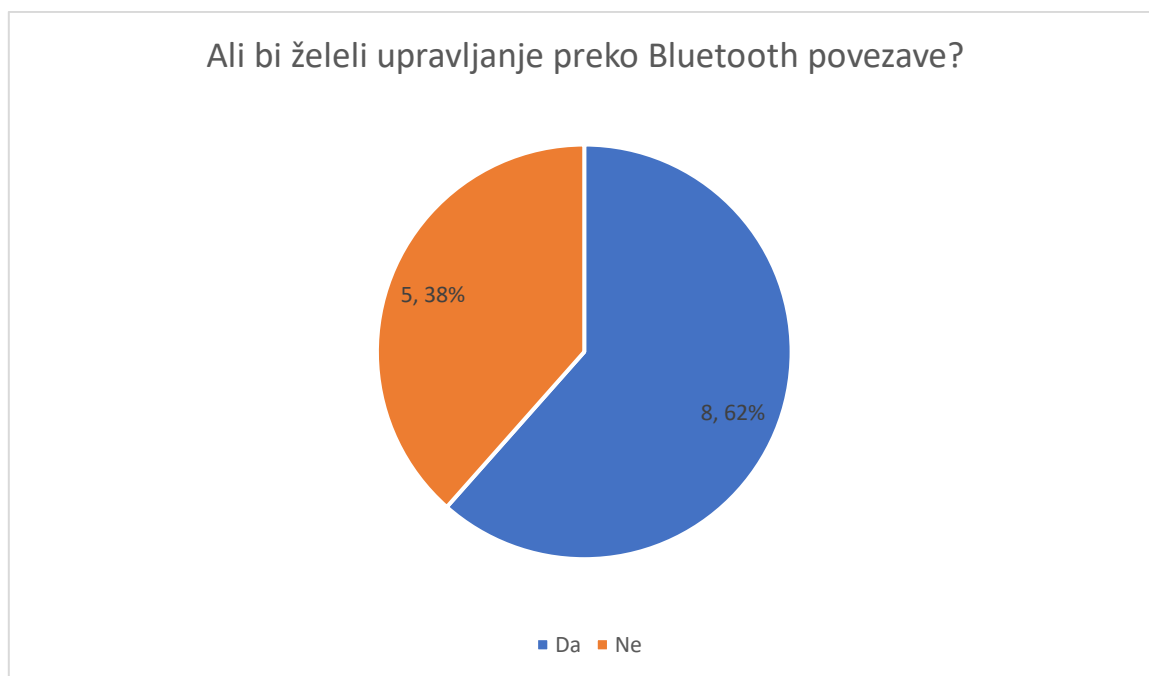
3. Ali bi si želeli centralizirano možnost upravljanja glasnosti posameznih programov?



Graf 3: Želja po centraliziranem upravljanju.

To vprašanje je bilo pomembno za implementacijo mešalnika glasnosti v aplikaciji za mobilni telefon. Iz grafa lahko razberemo, da je ta funkcija zaželena.

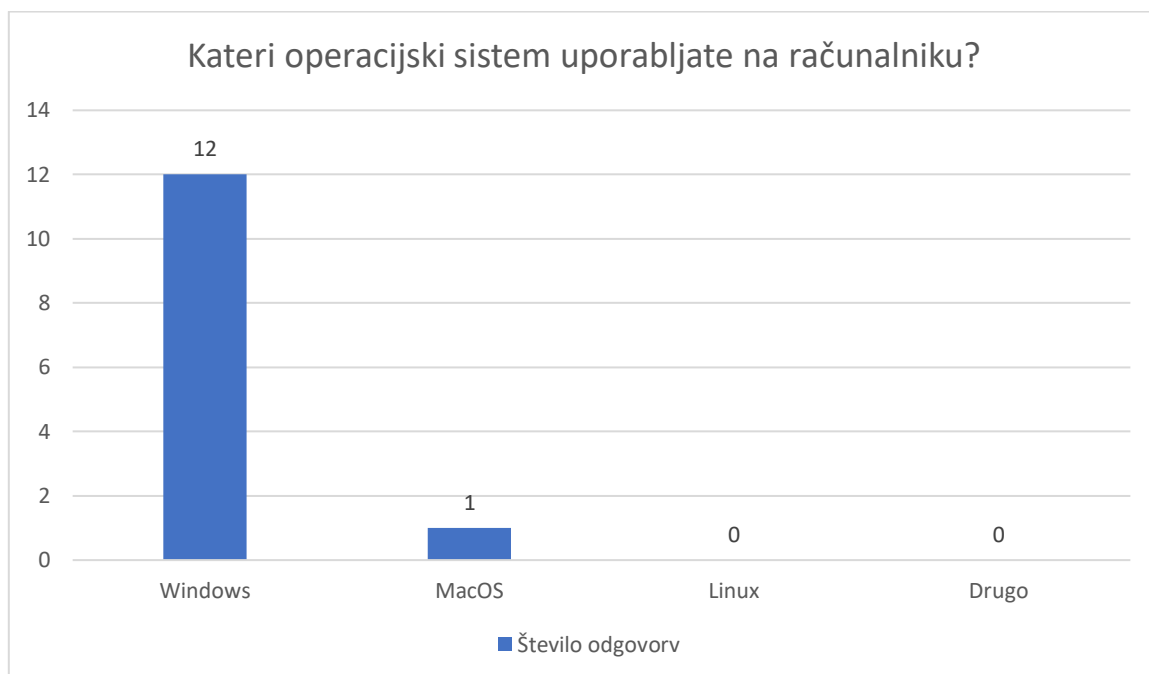
4. Ali bi želeli upravljanje preko Bluetooth povezave?



Graf 4: Upravljanje preko Bluetooth povezave.

Iz tega vprašanja je bilo ugotovljeno, da si večina želi upravljanja preko Bluetooth povezave.

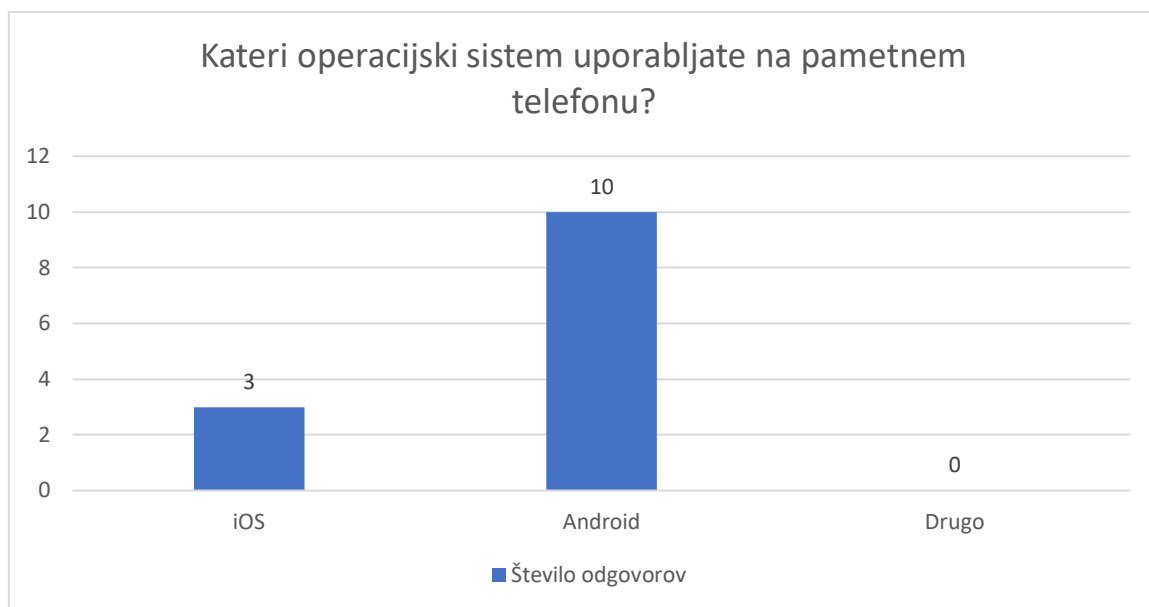
5. Kateri operacijski sistem uporabljate na računalniku?



Graf 5: Operacijski sistem na računalniku.

Večina vprašanih uporablja operacijski sistem Windows. To je dobro vedeti, ker če se kdaj odločim prilagoditi to aplikacijo za drugi operacijski sistem, me zanima, ali bi se to sploh izplačalo.

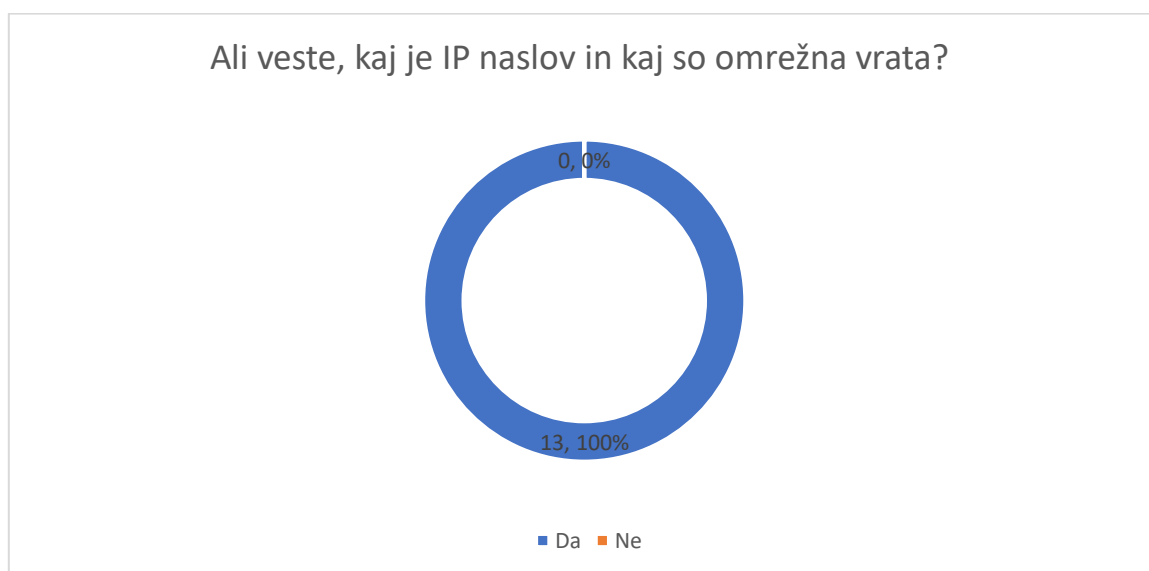
6. Kateri operacijski sistem uporabljate na pametnem telefonu?



Graf 6: Operacijski sistem na pametnem telefonu.

Želel sem izvedeti podobne stvari kot pri vprašanju 6, samo, da gre tukaj za pametni telefon.

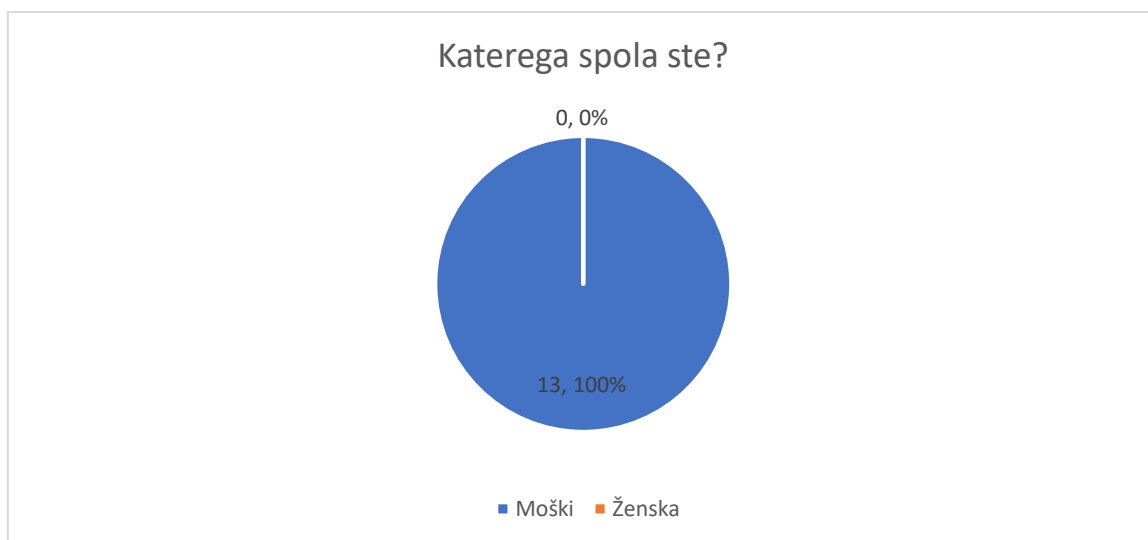
7. Ali veste, kaj je IP naslov in kaj so omrežna vrata?



Graf 7: Poznavanje IP naslovov in vrat.

Vsi vprašani vedo, kaj je IP naslov in kaj so vrata. Odgovori povejo, ali je potrebno implementirati takšen sistem povezovanja, da bodo pridobili povezavo tudi tisti, ki ne poznajo stroke.

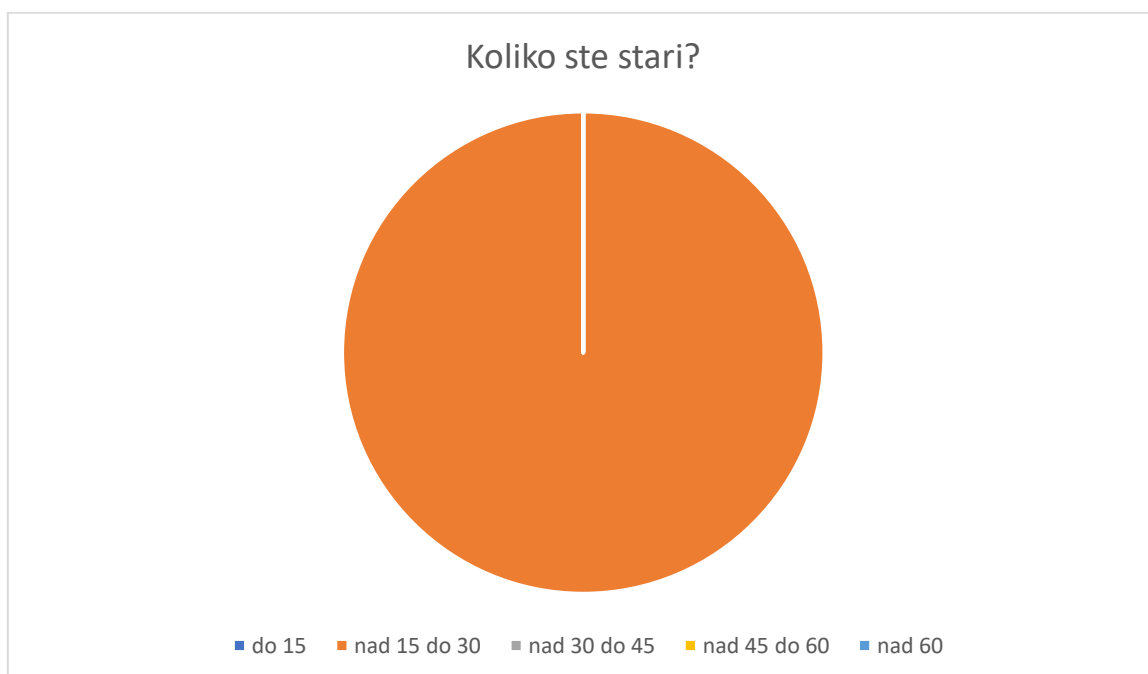
8. Katerega spola ste?



Graf 8: Spol vprašanih.

Zanimalo me je kdo so vprašani, in ali so kakšne zahteve bolj specifične za spol.

9. Koliko ste stari?



Graf 9: Starost vprašanih.

Vsi, ki so izpolnili vprašalnik, so stari med 15 in 30 let. Ta podatek sem potreboval za bolj natančno analizo, kaj si katera starostna skupina bolj želi in kaj ta skupina uporablja.

5.1 Ugotovitve ankete

Iz ankete sem veliko izvedel o dejanski uporabnosti aplikacije, njenih zaželenih funkcij in strukturi sestavi možnih uporabnikov.

Razčistila je tudi veljavnost druge, tretje in četrte hipoteze. Druga je bila potrjena pri odgovorih na vprašanja pet in šest. Tretja hipoteza je bila potrjena z večino pritrdilnih odgovorov na prvo vprašanje. Četrta hipoteza ni bila ne potrjena ne ovržena, ker je bil vzorec premajhen za pravilno sklepanje.

Glede na to, da je bilo izpolnjenih malo vprašalnikov, ta anketa ne more predstavljati realnega stanja v svetu. Videti je, da je vprašalnik izpolnila ista starostna skupina iste izobrazbene smeri. Vprašanja dva, tri in štiri so podala vpogled o željah vprašanih o funkcijah, ki si jih želijo imeti v aplikaciji.

6 Analiza hipotez

S prvo hipotezo sem želel potrditi ali ovreči svoje mnenje o razvoju omrežnih protokolov: ali je razvoj lastnega aplikacijskega omrežnega protokola za komunikacijo res tako zahteven, kot si mislim. Izkazala se je kot potrjena, saj je razvoj in implementacija lastnega omrežnega protokola za komunikacijo res zahtevno opravilo, kar lahko sklepamo po vseh problemih, ki so se pojavili med razvojem in programiranjem.

Druga hipoteza je potrjena, ker je anketa pokazala, da večina ljudi na računalniku uporablja operacijski sistem Windows, na pametnem telefonu pa operacijski sistem Android.

Tretjo hipotezo potrjuje odgovori na prvo vprašanje, ki je povprašalo po želji za to aplikacijo, saj je 81 % vprašanih pritrdilo vprašanju, da si želijo takšno aplikacijo.

S četrto hipotezo sem želel preveriti svoje mnenje o večini anketirancev in njihovem poznavanju osnov računalniških omrežij. Te hipoteze ni možno ne potrditi ne ovreči, ker je bil vzorec premajhen in je predstavljal sam eno izobrazbeno smer in starostno skupino.

7 Zaključek

Med raziskovanjem sem se naučil veliko o tem, kako deluje prenos podatkov po omrežju in kako se morajo ti obdelati na računalnikih, da iz njih dobiš v pravilni obliki. Naučil sem se naučil tudi jezika C++, ki ga je bilo potrebno uporabiti. Kakor je bilo že omenjeno, ni bil favorit in ga ne bi več izbral za takšen tip programa, ampak bi raje izbral C#, s katerim je programiranje lažje.

Če bi se še enkrat odločil delati isto raziskovalno nalogo, bi spremenil trenutno pot. Bolj bi se posvetil prebiranju virov in iskanju informacij po internetu, kot sem se, in bi zato imel manj težav pri razvijanju programske opreme in ozadju delovanja.

Čeprav sem z aplikacijo dosegel večino zastavljenih ciljev, je še vedno odprtih veliko možnosti za nove funkcije in izboljšave, ki bi omogočale še nadaljnje raziskovanje. Ena teh možnih je, da se ta daljinec spremeni v fizično obliko.

8 Viri in literatura

guru99.com. (11. April 2021). *TCP 3-Way Handshake (SYN, SYN-ACK,ACK)*. Pridobljeno iz Guru99: <https://www.guru99.com/tcp-3-way-handshake.html>

guru99.com. (13. April 2021). *TCP/IP Model: Layers & Protocol | What is TCP IP Stack?* Pridobljeno iz guru99: <https://www.guru99.com/tcp-ip-model.html>

Hall, B. “. (13. April 2021). *Beej's Guide to Network Programming*. Pridobljeno iz <https://beej.us/guide/bgnet/html/>

Microsoft Corporation. (6. April 2021). *About the Windows Core Audio APIs*. Pridobljeno iz Microsoft Docs: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/about-the-windows-core-audio-apis>

Microsoft Corporation. (12. April 2021). *Audio Endpoint Devices*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/audio-endpoint-devices>

Microsoft Corporation. (12. April 2021). *Audio Session Events*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/audio-session-events>

Microsoft Corporation. (12. April 2021). *Audio Sessions*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/audio-sessions>

Microsoft Corporation. (9. April 2021). *Device Events*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/device-events>

Microsoft Corporation. (9. April 2021). *Device Roles*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/device-roles>

Microsoft Corporation. (10. April 2021). *Endpoint Volume Controls*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/endpoint-volume-controls>

Microsoft Corporation. (9. April 2021). *EndpointVolume API*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/endpointvolume-api>

Microsoft Corporation. (9. April 2021). *Enumerating Audio Devices*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/enumerating-audio-devices>

- Microsoft Corporation. (9. April 2021). *MMDevice API*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/mmdevice-api>
- Microsoft Corporation. (10. April 2021). *Session Volume Controls*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/session-volume-controls>
- Microsoft Corporation. (10. April 2021). *Volume Controls*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/volume-controls>
- Microsoft Corporation. (9. April 2021). *WASAPI*. Pridobljeno iz Microsoft Documetation: <https://docs.microsoft.com/en-us/windows/win32/coreaudio/wasapi>
- Wikimedia Foundation. (11. April 2021). *Berkely sockets*. Pridobljeno iz Wikipedia The Free Encyclopedia: https://en.wikipedia.org/wiki/Berkeley_sockets
- Wikimedia Foundation. (11. April 2021). *Bluetooth*. Pridobljeno iz Wikipedia The Free Encyclopedia: <https://en.wikipedia.org/wiki/Bluetooth>
- Wikimedia Foundation. (8. April 2021). *C++*. Pridobljeno iz Wikipedia The Free Encyclopeida: <https://en.wikipedia.org/wiki/C%2B%2B>
- Wikimedia Foundation. (12. April 2021). *Computer Mouse*. Pridobljeno iz Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Computer_mouse
- Wikimedia Foundation. (13. April 2021). *Internet Protocol Suite*. Pridobljeno iz Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Internet_protocol_suite
- Wikimedia Foundation. (13. April 2021). *Keyboard Technology*. Pridobljeno iz Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Keyboard_technology
- Wikimedia Foundation. (11. April 2021). *User Datagram Protocol*. Pridobljeno iz Wikipedia The Free Encyclopedia: https://en.wikipedia.org/wiki/User_Datagram_Protocol

9 Priloge

9.1 Anketni vprašalnik

***1) Ali ste si kdaj želeli upravljati računalnik preko pametnega telefona?**

Da

Ne

***2) Kaj se vam zdi najbolj pomembno pri upravljanju računalnika na daljavo?**

Možnih je več odgovorov

Upravljanje predvajanja multimedijskih vsebin

Upravljanje glasnosti zvoka

Upravljanje premika miške

Vnos preko tipkovnice na telefonu

Drugo:

***3) Ali bi si želeli centralizirano možnost upravljanja glasnosti posameznih programov?**

Da

Ne

***4) Ali bi želeli upravljanje preko Bluetooth povezave?**

Da

Ne

***5) Kateri operacijski sistem uporabljate na računalniku?**

Windows

MacOS

Linux

Drugo:

***6) Kateri operacijski sistem uporabljate na pametnem telefonu?**

- iOS (iPhone)
 - android (na primer Samsung, Huawei ...)
 - Drugo:
-

***7) Ali veste, kaj je IP naslov in kaj so omrežna vrata?**

- Da
- Ne

9) Hvala vam za vnesene odgovore. Sedaj, če želite, bi vas prosil, da izpolnite še dve kratki vprašanji o sebi.

10) Katerega spola ste?

- Moški
 - Ženska
-

11) Koliko ste stari?

- do 15
- nad 15 do 30
- nad 30 do 45
- nad 45 do 60
- nad 60

9.2 Izjava

Mentor/-ica _____ v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Oddaljeno nadziranje računalnika, katere avtor/-ica je Timotej Kroflič:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, _____

žig šole

Podpis mentorja

Podpis odgovorne osebe

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

9.2 Izjava

Mentor/-ica mag. Boštjan Lesinar v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Oddaljeno nadziranje računalnika, katere avtor/-ica je Timotej Kroflič:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 12.5.2021



Podpis mentorja

za [signature]

Podpis odgovorne osebe

[signature]

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.