

Gimnazija „Veljko Petrović“

Sombor, Dositeja Obradovića 2

Maturski rad iz objektno orijentisanog programiranja

Upravljanje računarima u učionici

Mentor:

prof. Diósy Csaba

Učenik:

Đorđe Mančić

Sombor, maj 2023. god.

Sadržaj

1. Uvod.....	3
2. Dizajn programa.....	5
2.1. Ime i logotip.....	5
2.2. Korisnički interfejs, konfiguracija programa.....	6
3. Sistem učionica i uloga	11
3.1. IP protokol.....	11
3.2. TCP protokol.....	12
3.3. UDP protokol	13
3.4. C# .NET implementacija TCP i UDP	13
3.5. Koncept učionice	14
3.6. „Sava protokol“.....	15
3.7. Traženje učiteljskog računara	15
3.8. Proveravanje učeničkog računara	16
4. Glavne funkcije programa.....	17
4.1. Deljenje ekrana	17
4.2. Slanje poruka	18
4.3. Isključivanje računara	18
5. Zaključak.....	19
6. Literatura	20

1. Uvod

Informacione tehnologije se veoma brzo razvijaju i sve su više prisutne, a i potrebne, u našoj svakodnevnicima – od dopisivanja sa našim najbližima, igranja video-igara, gledanja filmova i zabavnih snimaka na internetu, sve do kucanja dokumenata za posao, plaćanja raznoraznih računa i preciznih računanja vrlo bitnih podataka. Od ogromnog su značaja za napredak čovečanstva i danas nije moguće više zamisliti kakav bi svet bio bez njih.

Naravno, kako bi neko iskoristio pun potencijal informacionih tehnologija, potrebno je da zna da ih koristi.

Ranije su ljudi učili kako da koriste svoj nov računar, pametni telefon, tablet ili neki program na njima od svojih, često mlađih, ukućana, prijatelja i kolega sa posla, ili bi upisali kurs kako bi učili na tradicionalniji način. Međutim, kako je informatika napredovala i za decu se pretvorila iz „ono što stariji koriste“ u nešto što se pojavljuje svakog dana u njihovim životima, prvenstveno za učenje i zabavu, nastala je potreba da i deca znaju ne samo kako da koriste informacione tehnologije, već i da ih pravilno i bezbedno koriste. Tu potrebu ne može da zadovolji baš svako, već izvanredni učitelji koji znaju kako da rade sa decom. Kao što je rekao Bil Gejts 12. oktobra 1997. godine u intervjuu za „Independent on Sunday“ – *„Tehnologija je samo alat. Da bi deca radila zajedno i bila motivisana, učitelj je najbitniji“*. Na našu sreću, u poslednje vreme su se ti učitelji našli i u učionicama svakog osnovca, omogućivši svakom detetu da ima način da bude informatički pismeno.

U jednom prosečnom informatičkom kabinetu, svaki učenik, učitelj, nastavnik i profesor sedi za svojim računarom ili klijentom povezanom na server tj. najčešće računar na sred stola. Ipak, znajući kakve je informatika prirode, svaki alat znači kako bi se olakšalo predavanje gradiva učenicima. Činjenica da svaki učenik ima svoj računar znatno olakšava posao, ali samo to nije dovoljno. Zašto bi profesor trebao da traži fleš drajv kako bi preneo datoteke na učeničke računare, uključivao projektor ili crtao po tabli kako bi pokazao učenicima kako se koristi program? Svi ti računari su već povezani na zajedničku računarsku mrežu, pa čak i internet – trebalo bi da koristimo te prednosti.

Srećom, postoji već brdo softvera za upravljanje računarima u učionici, kao što su, na primer, „Google učionica“ ili „LanSchool“. Problem je u tome što Google učionica, iako besplatan softver, služi isključivo za deljenje datoteka i ostalih informacija sa učenicima, a takođe je potreban i veb pregledač koji sporo rade na starijim školskim računarima. LanSchool rešava praktično sve probleme na koje učitelj može da naiđe, ali problem je u tome što ne samo što

treba da se plati licenca za program, već je ta licenca i skupa, a veliki broj škola u Srbiji nema novca da priušti sebi tako nešto na godišnjem nivou. Naravno, postoji besplatna demo verzija tog programa starija nego neki učenici, ali sa time i dolaze moguće sigurnosne slabe tačke u softveru. Posle dugog vremena koje sam proveo tražeći besplatne alternative za taj program, došao sam do zaključka da tako nešto ne postoji. Motivisan da pomognem školama u Srbiji da najzad imaju softver koji bi olakšao predavanje informatike svim učiteljima, nastavnicima i profesorima, odlučio sam da izaberem baš to kao temu za moj maturski rad – softver za upravljanje računarima u učionici koji je, po ideji, besplatan i otvorenog koda, tako da svako ko želi i ima volju može da ga poboljša i podeli ista poboljšanja sa ostalima. Program je napisan u C# programskom jeziku i .NET 7.0 radnom okviru, unutar razvojnog okruženja Visual Studio 2022.

2. Dizajn programa

Dizajn bilo kakvog programa, kao i ovog, je vrlo bitan stvar. Pored ulaganja dosta vremena praveći korisnički interfejs, odlučio sam i da dam programu ime i logotip.

2.1. Ime i logotip

Program sam odlučio da nazovem „Sava Monitor“, te se Sava Monitor može pronaći u daljem tekstu umesto reči program, aplikacija itd. Odabrao sam ime Sava jer sam želeo da ga imenujem po Svetom Savi – prvom srpskom prosvetitelju, a i znajući da je Sveti Sava širom Srbije i školska slava. Monitor sam dodao na ime kako ne bi došlo do zabune, s obzirom na to da dosta marki imaju reč „Sava“ u sebi, a i kako bi se lako znalo da je program namenjen kao monitor za učionicu.

Logotip na slici 1 sam, slično inspiraciji za ime, dizajnirao da bude kombinacija oreola i vizualizacija signala, kako bih istakao još više namenu programa i poreklo imena.



Slika 1. Logotip Sava Monitora

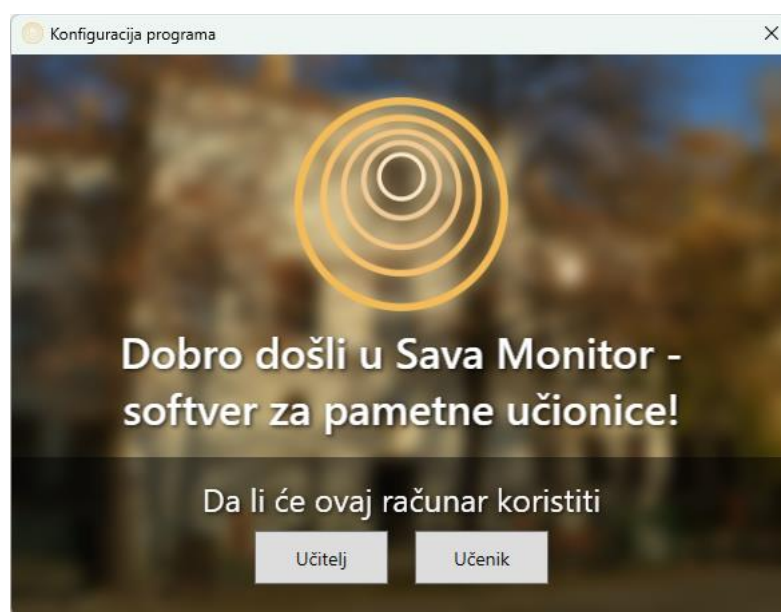
Logotip je dizajniran u programu „Adobe Illustrator“.

2.2. Korisnički interfejs, konfiguracija programa

Korisnički interfejs Sava Monitora je dizajniran da bude pregledan i lak za korišćenje.

Odlučio sam da koristim WPF (engl. Windows Presentation Foundation) radni okvir za korisnički interfejs, s obzirom na to da je dosta fleksibilan i dalje održavan, čak i posle skoro 2 decenije od prvog izdanja.

Kada korisnik otvori Sava Monitor po prvi put, otvara se prozor za konfiguraciju programa, kao što je prikazano na slici 2.



Slika 2. Početni prozor

Preporučeno je da učitelj, nastavnik ili profesor uradi konfiguraciju svakog računara, bez obzira na to da li je učiteljski ili učenički. Potrebno je prvo konfigurisati učiteljski računar, a time i učionicu. Klikom na dugme „Učitelj“ se pojavljuje forma prikazana na slici 3, u koju se unose podaci o učionici, kao što su ime, ID i, ukoliko je potrebno promeniti, port učionice, kao i statičku IP adresu računara.

Konfiguracija programa

Konfiguracija učiteljskog računara i učionice

Ime učionice

ID učionice

Port učionice

Nemojte menjati port osim ako je sigurno zauzet.

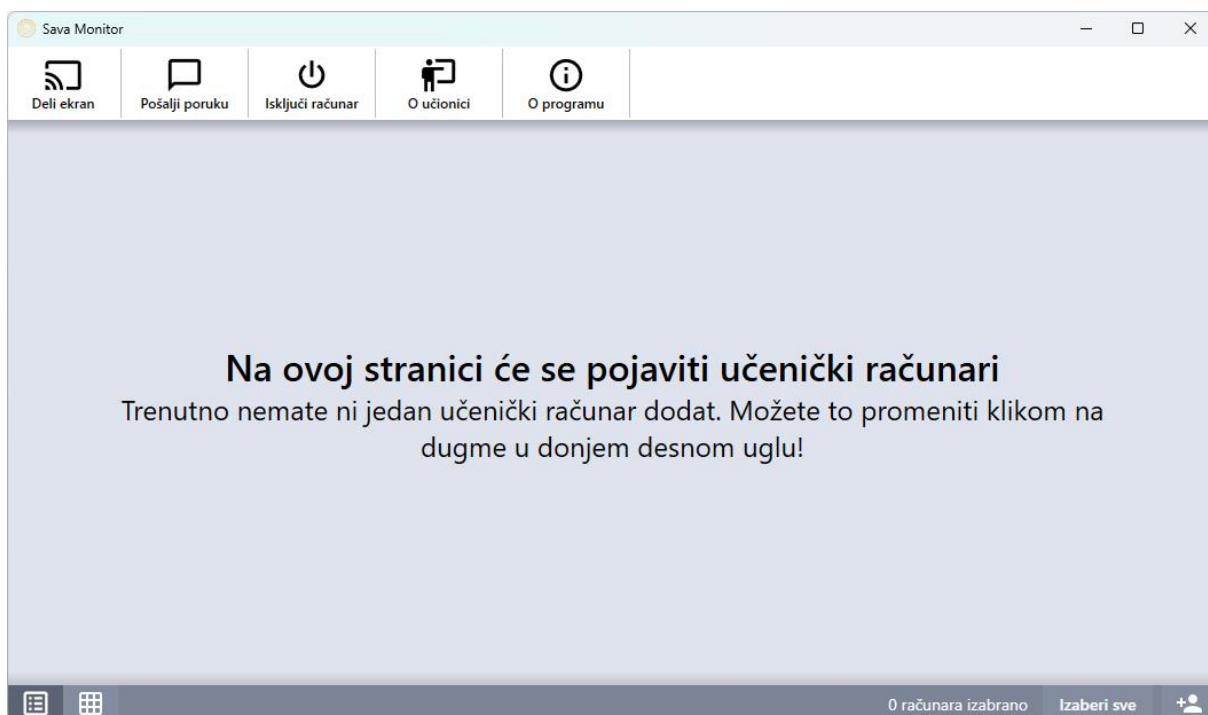
☐ Računaru je dodeljena statička IP adresa (ubrzava povezivanje)

ID računara

Nazad Sledeće

Slika 3. Forma za konfiguraciju učiteljskog računara

Klikom na dugme „Sledeće“ se pravi konfiguracija. Prilikom završavanja konfiguracije učiteljskog računara, pojavljuje se glavna konzola učionice, prikazana na slici 4.



Slika 4. Glavna konzola učionice

Na glavnoj konzoli možemo videti dugmad, kao što su glavne funkcije na gornjoj traci:

- **Deli ekran** – Omogućava deljenje ekrana sa učiteljskog računara

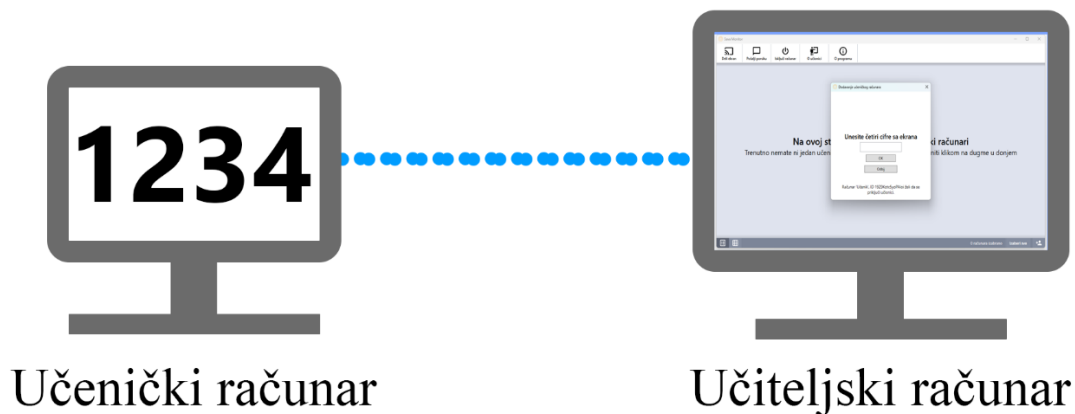
- **Pošalji poruku** – Šalje poruku učeničkim računarima
- **Isključi računar** – Isključuje računare na daljinu
- **O učionici** – Sadrži sve podatke o učionici
- **O programu** – Osnovni podaci o programu

Na donjoj traci uglavnom sporedne funkcije – dugmad za promenu prikaza učeničkih računara u donjem levom uglu, kao i informacije o broju učeničkih računara izabrano i dugme za odabir svih računara u donjem desnom uglu. Međutim, ništa od toga neće funkcionisati ako se ne priključi bar jedan učenički računar na našu učionicu. Učeničkim računarima je zabranjeno da se priključe učionici sve dok se na učiteljskom računaru ne otvori prozor za dodavanje novog računara. To možemo uraditi klikom na dugme za dodavanje učeničkih računara u donjem desnom uglu.

Isto kao što smo konfigurisali učiteljski računar, na drugom računaru ćemo, kada nas program pita ko će koristiti računar, kliknuti na „Učenik“. Time će se pojaviti forma prikazana na slici 5, u koju se unose podaci kao što su ime računara (npr. „Učenik 1“, „Učenik 2“ itd.), ID učionice, port učionice, u slučaju da je izmenjen, i mogućnost da se navede statička IP adresa učiteljskog računara. Ukoliko ste zaboravili podatke o učionici, možete ih ponovo proveriti klikom na dugme „O učionici“ na vrhu glavne konzole.

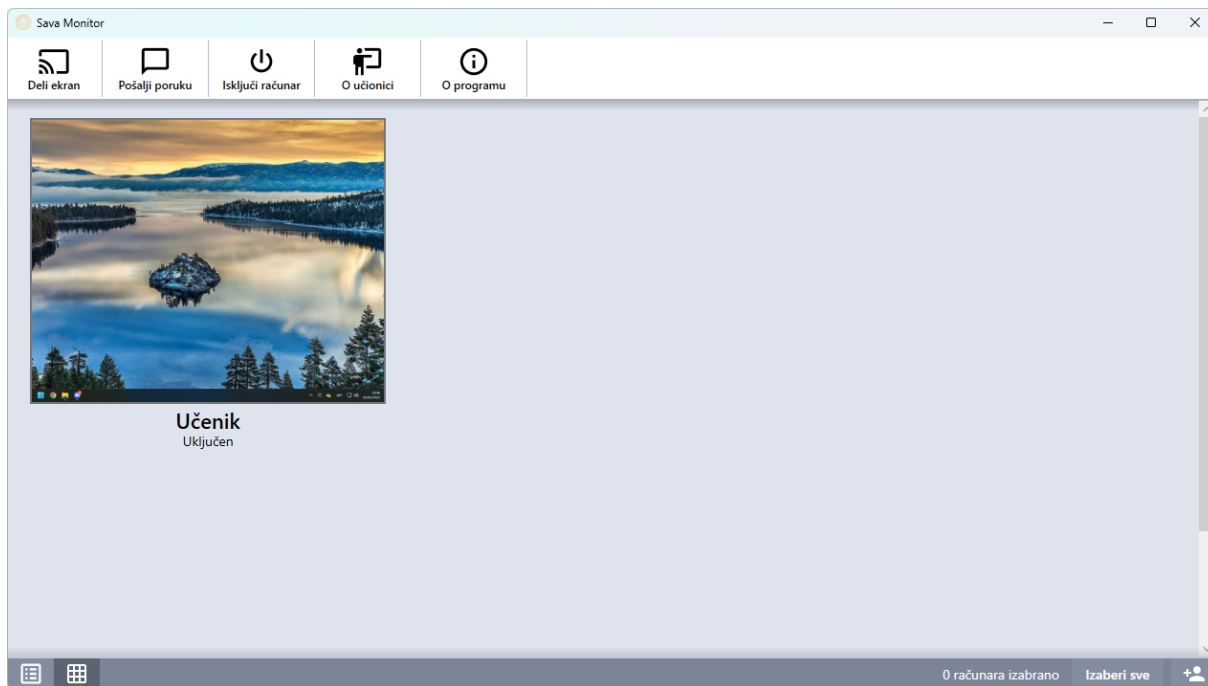
Slika 5. Forma za konfiguraciju učenikog računara

Kada učenički računar nađe učiteljski računar, kreće proces uparivanja, gde učenički računar prikazuje četvorocifreni broj koji se mora uneti na učiteljskom računaru. Vizualizacija je prikazana na slici 6.



Slika 6. Vizualizacija procesa priključivanja računara na učionicu

Nakon pravilnog unošenja četvorocifrenog broja, učenički računar je dodeljen u učionicu, kao što je prikazano na slici 7. Tada možete menjati prikaz da prikazujete računare u listi ili kao ikonice, pregledati šta je na ekranu učenika na svakih par sekundi, kao i koristiti glavne funkcije programa, kao što su deljenje ekrana ili slanje poruke, nakon što izaberete prvo računare za koje želite da izvršite radnju.



Slika 7. Glavna konzola nakon dodavanja učeničkog računara

Uzmite u obzir da se Sava Monitor ne uključuje pri prijavljivanju na računar, jer je instalacioni program van opsega ove teme. Međutim, možete ručno postaviti da se otvara pri svakom prijavljivanju tako što ćete napraviti prečicu za „SavaMonitor.exe“ datoteku i postaviti je u **C:\Users\[ime korisnika]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup** direktorijum.

3. Sistem učionica i uloga

Kao početna tačka tehničkog dela programa, prvo ćemo razjasniti na koji način Sava Monitor omogućava računarima da komuniciraju međusobno, kako funkcioniše koncept učionica, kako računar zna koju ulogu ima i u kojoj je učionici...

Međutim, kako bi daljnji tekst bio jasniji, moramo znati karakteristike IP, TCP i UDP protokola, kao i njihove implementacije u C# i .NET radnom okviru.

3.1. IP protokol

Računarska mreža je mreža u kojoj dva ili više računara mogu da komuniciraju međusobno. Po toj definiciji, ako povežemo dva računara kablom, bez ikakvog posrednika, imamo računarsku mrežu. Takva jedna situacija je vrlo jednostavna – ukoliko prvi računar šalje podatke, zna se da su ti podaci namenjeni drugom računaru i obrnuto. Ipak, računarske mreže najčešće nisu tako jednostavne, i sastoje se od mnogo više računara. Ako su, na primer, 3 ili više računara povezana „magičnim kablom“ koji ih povezuje sve međusobno kao digitalna verzija telefona od konzervi i kanapa, nastaju razni problemi prilikom slanja i primanja podataka.

Najočigledniji problem jeste taj da računari nemaju predstavu o tome ko šalje podatke i do kog računara će njihovi podaci stići. Drugi problem je to što bilo kakve podatke da računar pošalje, ukoliko još neko šalje podatke preko tog kabla u isto vreme, podaci će se preplitati i postati u potpunosti nejasni.

Prvi problem možemo rešiti nekim identifikatorom koji je dodeljen svakom računaru, poput JMBG-a za računare. Usput ćemo postaviti i posrednika u našoj mreži – još jedan računar kojem ćemo, pored podataka, slati i identifikator računara kojem želimo da pošaljemo podatke. Identifikator će posrednik dodeliti svakom računaru svojevolejno i obavestiti računare o ostalim računarima na mreži, kao i njihovim identifikatorima.

Drugi problem i dalje trebamo rešiti jer bi nastali problemi kada bi dva ili više računara krenula da šalju podatke u isto vreme istom računaru, te bi se ponovo podaci preplitali. Možemo pustiti prvi računar da pošalje sve podatke, dok drugi čeka, i tek kada završi prvi sa slanjem podataka onda drugi može krenuti sa slanjem svojih podataka. Takvo rešenje radi samo kada su manji podaci u pitanju, ali kada su veći, mogu nastati ogromni zastoji u mreži, a naš cilj je da omogućavamo da računar ipak može da prima u isto vreme podatke od više

računara. Taj problem je takođe rešiv – baš zato što prošlo rešenje radi dobro kada su mali podaci u pitanju, zašto ne bismo „cepkali“ podatke od jednog i drugog računara, te ih slali u komadićima, tj. paketima? Na taj način bismo rešili i prvi i drugi problem, bez stvaranja novih.

Rešenja navedena za prvi i drugi problem zapravo i jesu deo **IP** protokola (engl. **Internet Protocol**), i temelj su današnjih računarskih mreža, jer omogućavaju da podaci dospeju do željenog računara u kraćem vremenskom periodu. Posrednik, u prethodno navedenom tekstu, je ruter, a identifikator je IP adresa.

IP adresa, po najčešće korišćenom IP verzija 4 protokolu, se sastoji od četiri bajtova i predstavljene su u formatu **[prvi bajt].[drugi bajt].[treći bajt].[četvrti bajt]** – što znači da postoji ukupno preko 4 milijardi IP adresa koje mogu biti korišćene. Naravno, kako postoji mnogo više od 4 milijardi računara na ovome svetu povezanih na internet, jedna adresa na internetu ne znači jedan računar, već jedna mreža unutar internacionalne mreže. Ta mreža, zatim, može da ima svoje adrese preko kojih će identifikovati, ili računare, ili podmreže. Nas lično zanimaju lokalne mreže, koje imaju svoj rezervisan opseg, od 192.168.0.0 do 192.168.255.255

Nama su potrebne lokalne IP adrese kako bismo znali kom računaru u kabinetu tačno šaljemo podatke, ali i dalje se paketi mogu slučajno poslati u pogrešnom redosledu ukoliko računarima šaljemo veće podatke, te nam je potreban još jedan protokol pored IP protokola koji bi te pakete doveo u red – TCP protokol.

3.2. TCP protokol

TCP (engl. **Transmission Control Protocol**) je protokol koji ima ulogu da obezbedi slanje podataka u pravilnom redosledu i bez izgubljenih paketa. Putem TCP protokola, dva računara otvaraju vezu kako bi slali podatke. Zatim, kada jedan računar šalje podatke drugom, uz te podatke se šalju i dodatni TCP podaci o redosledu, kao i ostali podaci u okviru TCP protokola.

Paketi nekada mogu da se „izgube“ prilikom slanja, te u sklopu TCP protokola je uključena i provera da li je paket zaista došao do željenog računara. Ukoliko željeni računar se ne odaziva da je primio paket, on se ponovo šalje njemu.

Zbog svih tih razloga, TCP se koristi kada želimo da budemo sigurni da su podaci koje želimo da pošaljemo stigli na željeno mesto, ali to znači da će podaci sporije stići zbog svih provera koje su u sklopu TCP protokola.

Bitno je napomenuti i da se TCP nikako ne koristi sam, već skoro uvek uz kombinaciju sa IP protokolom, te se ta kombinacija zove TCP/IP.

TCP/IP zato ima primenu kod elektronske pošte, preuzimanja datoteka ili učitavanja stranica preko veb pregledača, ali ne i za razgovor preko interneta, video-časkanje i sličnog, već se koristi protokol koji je sušta suprotnost TCP-a – UDP protokol.

3.3. UDP protokol

UDP (engl. User Datagram Protocol) je protokol koji je dosta jednostavniji od TCP-a. Za razliku od njega, ne otvara vezu za slanje podataka, već ih šalje u istom trenutku ka željenom računaru. Nema provere kao u sklopu TCP protokola, te podaci, iako brzo stignu do željenog cilja, ne postoji ista pouzdanost, te paketi mogu da uopšte ne dođu do računara ili u pogrešnom redosledu.

Možda se pitate – „Ko bi uopšte koristio ovo?“. UDP ima primene kod video-časkanja i usluga digitalne telefonije baš zato što omogućava da podaci brzo stignu do cilja, a podataka ima toliko puno da nije potrebno brinuti o gubljenju paketa. UDP je takođe savršen za manje podatke, o kojima ćemo pričati kasnije u tekstu.

3.4. C# .NET implementacija TCP i UDP

.NET radni okvir nudi, kako sirov pristup mrežnim soketima, tako i srede klase za primanje i slanje podataka putem TCP i UDP protokola.

Za TCP protokol postoje klase **TcpListener**, koja služi kao TCP server, i **TcpClient**, koja služi kao TCP klijent. **TcpListener** se konstruiše sa potrebnim parametrima kao što su IP adresa (koja može biti postavljena na **IPAddress.Any** za slušanje konekcija na bilo kojoj IP adresi vezanoj za računar) i port. Zatim, posle pokretanja **TcpListener** uz pomoć **TcpListener.Start()** metode, može da čeka **TcpClient** uz pomoć metode **TcpListener.AcceptTcpClient()** koja zaustavlja kod sve dok ne dobije konekciju, a **TcpClient** se može povezati na njega uz pomoć **TcpClient.Connect()** ili asinhrono **TcpClient.ConnectAsync()** metode. **TcpClient** ima svojstvo koje se zove **NetworkStream**, koji se koriste za čitanje podataka sa prenosa u niz bajtova pomoću **NetworkStream.Read()** metode, i slanje podataka tj. niza bajtova uz pomoć **NetworkStream.Write()** metode.

Slede primeri koda za TCP server (slika 8.) i klijent (slika 9.) u C# u .NET radnom okviru.

```

TcpListener tcpListener = new TcpListener(IPAddress.Any, 3212);
tcpListener.Start();
Byte[] bytes = new Byte[256];
while (true)
{
    TcpClient tcpClient = tcpListener.AcceptTcpClient();
    NetworkStream networkStream = tcpClient.GetStream();

    while (tcpClient.Connected == true)
    {
        int bytesReceived = networkStream.Read(bytes, 0, bytes.Length);
        Debug.WriteLine("Dobijena poruka: " + Encoding.UTF8.GetString(bytes));
        if (bytesReceived == 0) break; //Kraj konekcije
        networkStream.Write(Encoding.UTF8.GetBytes("Priljeno"));
    }
}

```

Slika 8. Primer koda za TCP server

```

TcpClient tcpClient = new TcpClient();
tcpClient.Connect(IPAddress.Parse("192.168.0.1"), 3212);
NetworkStream networkStream = tcpClient.GetStream();
networkStream.Write(Encoding.UTF8.GetBytes("Pozdrav od klijenta!"));
Byte[] bytes = new Byte[256];
while (tcpClient.Connected == true)
{
    int bytesReceived = networkStream.Read(bytes, 0, bytes.Length);
    Debug.WriteLine("Dobijena poruka: " + Encoding.UTF8.GetString(bytes));
    if (bytesReceived == 0) break; //Kraj konekcije
    networkStream.Write(Encoding.UTF8.GetBytes("Prijem."));
}

```

Slika 9. Primer koda za TCP klijent

Kako je UDP dosta jednostavniji protokol, postoji samo **UdpClient** klasa putem koje se šalju i čitaju podaci na vrlo sličan način kao za TCP protokol.

Sledi primer koda za UDP implementaciju u C# u .NET radnom okviru na slici 10.

```

UdpClient udpClient = new UdpClient(3212);
while (true)
{
    IPEndPoint remoteEndPoint = new IPEndPoint(IPAddress.Any, 0);
    Byte[] receivedBytes = udpClient.Receive(ref remoteEndPoint);
    //remoteEndPoint će posle prijema biti postavljen na IP adresu
    //pošaljioca podatka
    Debug.WriteLine("Dobijena poruka: " + Encoding.UTF8.GetString(receivedBytes));
    udpClient.Send(Encoding.UTF8.GetBytes("Zdravo UDP klijentu!"));
}

```

Slika 10. Primer koda za UDP klijent

3.5. Koncept učionice

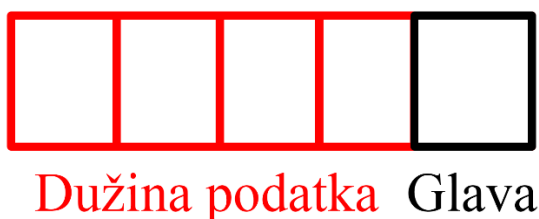
Učionice su grupe čiji su vlasnici učiteljski računari. Kada se učenički računar tj. klijent poveže na učionicu, on se zapravo povezuje na učiteljski računar tj. server koji mu pruža sve usluge učionice, a učenički računar sve podatke primljene od strane učiteljskog računara primenjuje na svoj objekat učionice.

Iz sigurnosnih razloga, kako bi se učenički računar povezao na učionicu, mora prvo da prosledi lozinku koju je dobio od učiteljskog računara prilikom procesa priključivanja na učionicu.

Međutim, učenički računar nema odmah IP adresu učiteljskog računara, kako uspeva da se ipak poveže?

3.6. „Sava protokol“

Pre započinjanja tema o kodu za umrežavanje, bitno je spomenuti da pre slanja bilo kakvih podataka, Sava Monitor šalje i pet bajtova o podatku koji bi trebalo da pošalje, vizualizovano na slici 11.



Slika 11. Vizualizacija početnih bajtova podataka

Prva četiri bajta su 32-bitni broj i opisuju koliko bajtova računar može još da očekuje da mu se pošalju, tj. **Dužina podatka**.

Peti bajt je **Glava**, i označava koji tip podatka se šalje – da li je možda zahtev za ulazak u učionicu, ili slanje slike ekrana?

3.7. Traženje učiteljskog računara

Učenički računar, dakle, ima samo ID učionice i pretpostavimo da nema zapisanu statičku IP adresu učiteljskog računara. Kako ipak uspe da nađe učiteljski računar i poveže se na njega?

Prvo, uzima sve podatke o svojim mrežnim interfejsima, kao što su lokalna IP adresa i IPv4 maska interfejsa. Zatim, računa adresu za emitovanje UDP paketa širom mreže tako što izvrši ILI operaciju između lokalne IP adrese i inverznih bajtova IPv4 maske, te emituje UDP paket za traženje učionica. Svaki učiteljski računar se pritom odazove na taj paket, i šalje nazad informaciju da je primio paket. Učenički računar beleži svaku IP adresu koja se odazove i, posle 3 sekunde, pokušava da se poveže putem TCP protokola na svaku IP adresu koja se odazvala, te daje zahtev učiteljskom računaru da pošalje informacije o učionici. Učiteljski računar zatim šalje nazad osnovne podatke o učionici u JSON formatu i, ukoliko se poklapa

ID učionice sa traženim ID-om, nastavlja po potrebi (povezuje se na učionicu, šalje zahtev za priključenje, itd.). Ukoliko se ne poklapa, nastavlja dalje, a ako ne nađe učionicu, u zavisnosti od razloga za traženje učiteljskog računara, ili pokušava ponovo, ili odustaje i obaveštava korisnika.

3.8. Proveravanje učeničkog računara

Na svakih par sekundi, učiteljski računar šalje TCP zahtev učeničkom računaru kako bi proverio da li se učenički računar odaziva. Ukoliko se uopšte ne odaziva, učiteljski računar označava učenički da nije povezan na učionicu. Učenički računar se na zahtev, pored povratne poruke, odaziva i snimkom svog ekrana, kako bi korisnik učiteljskog računara imao uvid u to šta učenik radi.

4. Glavne funkcije programa

Sava Monitor nudi par glavnih funkcija kao što su pomenute u poglavlju „Dizajn programa“. U ovom poglavlju ćemo preći preko tehničkih karakteristika svake funkcije.

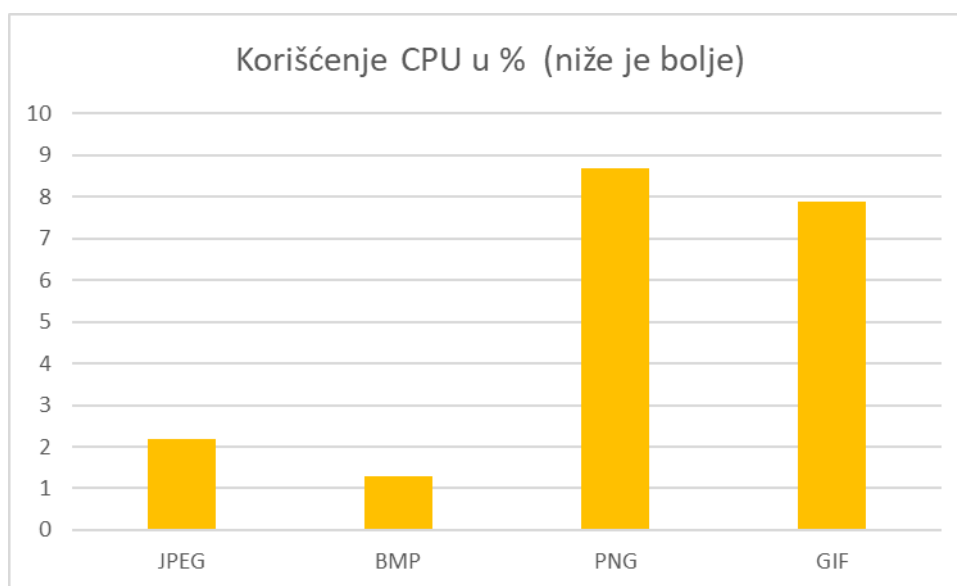
4.1. Deljenje ekrana

Pošto WPF radni okvir sam po sebi ne podržava pravljenje slika ekrana, morao sam koristiti par metoda iz Windows Forms radnog okvira.

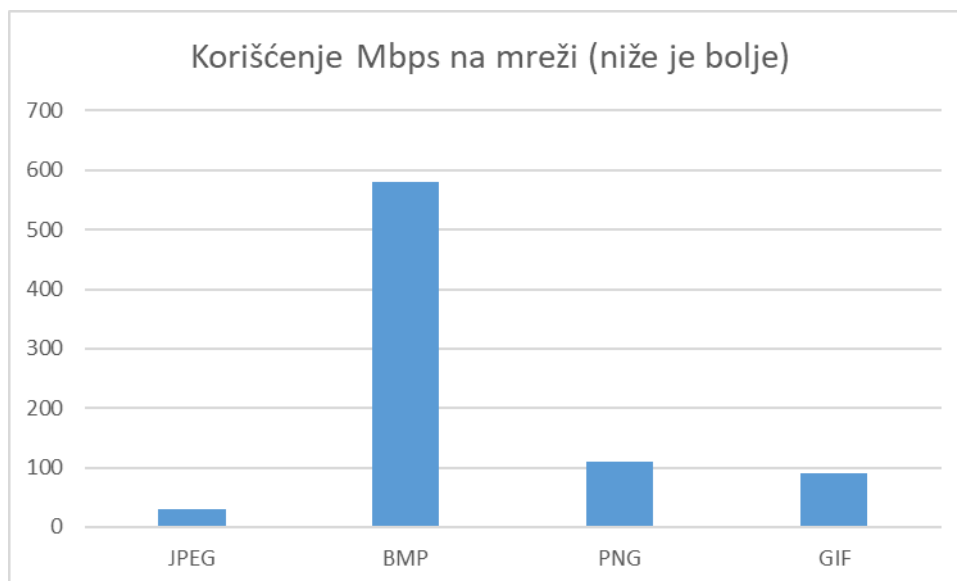
Program prvo pravi bitmapu, te uzima **Graphics** objekat od te bitmape. Na **Graphics** objekat poziva metodu **Graphics.CopyFromScreen()** koja kopira ceo ekran i lepi ga na bitmapu. Zatim, bitmapa se čuva u **JPEG** (engl. **Joint Photographic Experts Group**) formatu u **MemoryStream** kako bi se dobili sirovi bajtovi slike ekrana. Ti bajtovi se zatim šalju preko TCP protokola ka učeničkim računarima, koji čitaju bajtove i prave sliku od njih, te ih čuvaju u promenljivu koju može prozor za deljenje ekrana da očitava. To radi u krug 5 puta svake sekunde sve dok učiteljski računar ne prekine deljenje.

Ovaj proces se primenjuje i kod slanja slike ekrana sa učeničkog računara na učiteljski računar.

JPEG format je izabran posle dužeg testiranja na više računara kao format sa najboljim odnosom korišćenja što manje procesorske snage (slika 12.), ali i što manje veličine slike (slika 13.).



Slika 12. Prosečni procenat korišćenja procesorske snage za formate slika



Slika 13. Prosečno Mbps poslato pri deljenju ekrana sa formatima slika

Za TCP protokol sam se ipak opredelio jer program šalje celu sliku, a ne izmene u slici, te bi UDP protokol za deljenje ekrana bio loš izbor zbog svoje nepouzdanosti i male količine podataka koje bi program mogao da pošalje u jednom paketu.

4.2. Slanje poruka

Kada korisnik za učiteljskim računarom napiše poruku i pritisne „Pošalji“, ta poruka je zatim pretvorena u Unikod bajtove i poslata TCP protokolom izabranim učeničkim računarima.

Učenički računari zatim dobijaju podatak, pretvaraju ga u Unikod tekst i prikazuju putem **MessageBox.Show()** metode.

4.3. Isključivanje računara

Klikom na dugme za isključivanje, učiteljski računar šalje TCP protokolom praznu poruku sa glavom poruke koja signalizira na isključivanje. Učenički računari zatim pokreću ugrađenu Windows komandu „shutdown“ sa parametrima „shutdown /s /t 0“.

/s – Zahtev da se isključi računar

/t 0 – Odmah se isključuje računar, bez čekanja

5. Zaključak

Umrežavanje računara zaista nije bauk kao što neki misle da jeste, a veoma je korisna i interesantna oblast informatike koja uči čoveka da razmišlja na malo drugačiji način. C# i .NET radni okvir znatno olakšavaju posao pravljenja, kako aplikacija koje omogućavaju računarima da komuniciraju preko mreže, tako i kompleksnih aplikacija koje su raznovrsne po pitanju mogućnosti koje pružaju.

Obrazovanje je izuzetno bitno i smatram da bi trebao veći broj ljudi da bude angažovan u prosveti, makar indirektno, po pitanju rada na nečemu što bi olakšalo osobama koje obrazuju i koje se obrazuju, jer onaj ko ulaže bilo koji deo svog života u obrazovanje ostalih ulaže u bolju budućnost. Takođe sam vrlo zadovoljan temom koju sam odabrao jer, pored svega što sam naučio razrađujući ovu temu i uspeha da napravim nešto kao što je softver za upravljanje računarima u učionici, nadam se da ću moći svojim radom na neki način i da pomognem olakšavanju posla učiteljima i učenja učenicima u manjim školama u Srbiji.

6. Literatura

- <https://www.cloudflare.com/learning/network-layer/internet-protocol/>
- <https://datatracker.ietf.org/doc/html/rfc6890>
- <https://learn.microsoft.com/en-us/dotnet/>
- <https://mycomputernotes.com/network-address-and-broadcast-address/>