



ELEKTRONSKI FAKULTET NIŠ

STRUČNA PRAKSA

## **Izveštaj o realizaciji stručne prakse**

Jovan Jovanović 17673

Niš, April 2024.

## Sadržaj

|                |    |
|----------------|----|
| Uvod.....      | 3  |
| Izveštaj.....  | 4  |
| Zaključak..... | 12 |

## Uvod

Praksa je održana u firmi DMV u Nišu, i to u periodu od 15.04.2024. do 30.04.2024.

Firma je osnovana 1991. godine i bavi se izradom industrijskih kontrolnih sistema i delova LED ekrana uglavnom za svrhu saobraćajne signalizacije. Broji preko 200 zaposlenih a partneri su klijenti širom sveta iz preko 80 zemalja. U sklopu firme postoji 4000m<sup>2</sup> prostora za proizvodni pogon gde se ugrađuju svi tehnološki delovi za proizvodnju LED ekrana. To su u potpunosti mehaničke, elektronske, spremne za testiranje i montažu divizije, kao i jedinice za razvoj hardvera i softvera.

Kada je reč o razvoju softvera, postoji zaseban sektor za tu namenu. Koriste se tehnologije kako za WEB tako i za desktop aplikacije. Neke od tehnologija koje koriste za razvoj WEB aplikacija su PHP, Laravel, React.. Dok za desktop aplikacije koriste C#, C++, QT.

## Projekat

Kako sam se opredelio za rad u C# tehnologiji, dobio sam zadatak da izradim desktop aplikaciju. Koristio sam okruženje Visual Studio 2019 na kome sam kreirao Windows Forms aplikaciju. Kako se oni bave mahom izradom kontrolera za gotove uređaje, imao sam zadatak da pretražim lokalnu mrežu korišćenjem broadcast-a kako bih pronašao sve trenutno dostupne uređaje. Potom da izvršim filtriranje tih uređaja na osnovu imena parametara i imena grupe parametara koje unosi korisnik putem odgovarajućeg korisničkog interfejsa koji sam takođe trebao da napravim. Rezultat pretrage bi trebao korisniku da prikaže informacije o pronađenim uređajima na mreži (IP i MAC adresu uređaja), status o konekciji (da li je bilo moguće povezati se na uređaj i da li je pronađen odgovarajući rezultat) i vrednost traženog parametra.

Projekat je zamišljen da bude kao dodatna funkcionalnost aplikacije "Display Studio 2020". To je interna aplikacija firme DMV koja se koristi za proveru mreže, povezivanje sa uređajima na mreži i upravljanje istim. Tako da sam na raspolaganju imao njihovu C# biblioteku sa API-jem koji koristi aplikacija Display Studio.

# Izveštaj

## Dan 1 - Obilazak firme

Pre početka rada na projektu, mentor me je poveo u obilazak firme. Kako se na jednom mestu nalazi ceo pogon za projektovanje, izradu, programiranje, testiranje i transport proizvoda, obilazak je podrazumevao obilaženje sektora, kancelarija i laboratorija koji se koriste u navedene svrhe. Zanimljivost je da ova firma ima i veliku galeriju slika na zadnjem spratu sa sigurno preko stotinak različitih umetničkih dela.

## Dan 2 - Upoznavanje sa okruženjem Display Studio

Da bih bolje shvatio način na koji trebam da uradim svoj projekat, potrebno je bilo da se prvo upoznam sa samom aplikacijom.

Aplikacija je vrlo jednostavnog izgleda realizovana kao Windows Forms aplikacija. Meni su od koristi bile funkcionalnosti pretraga mrežnih kartica mog uređaja na kome sam radio, zatim pretraga mreže koja pokazuje sve dostupne uređaje, uspostavljanje konekcije sa nekim uređajem na mreži i prikaz svih parametara koje poseduje konkretan uređaj sa kojim se uspostavlja konekcija.

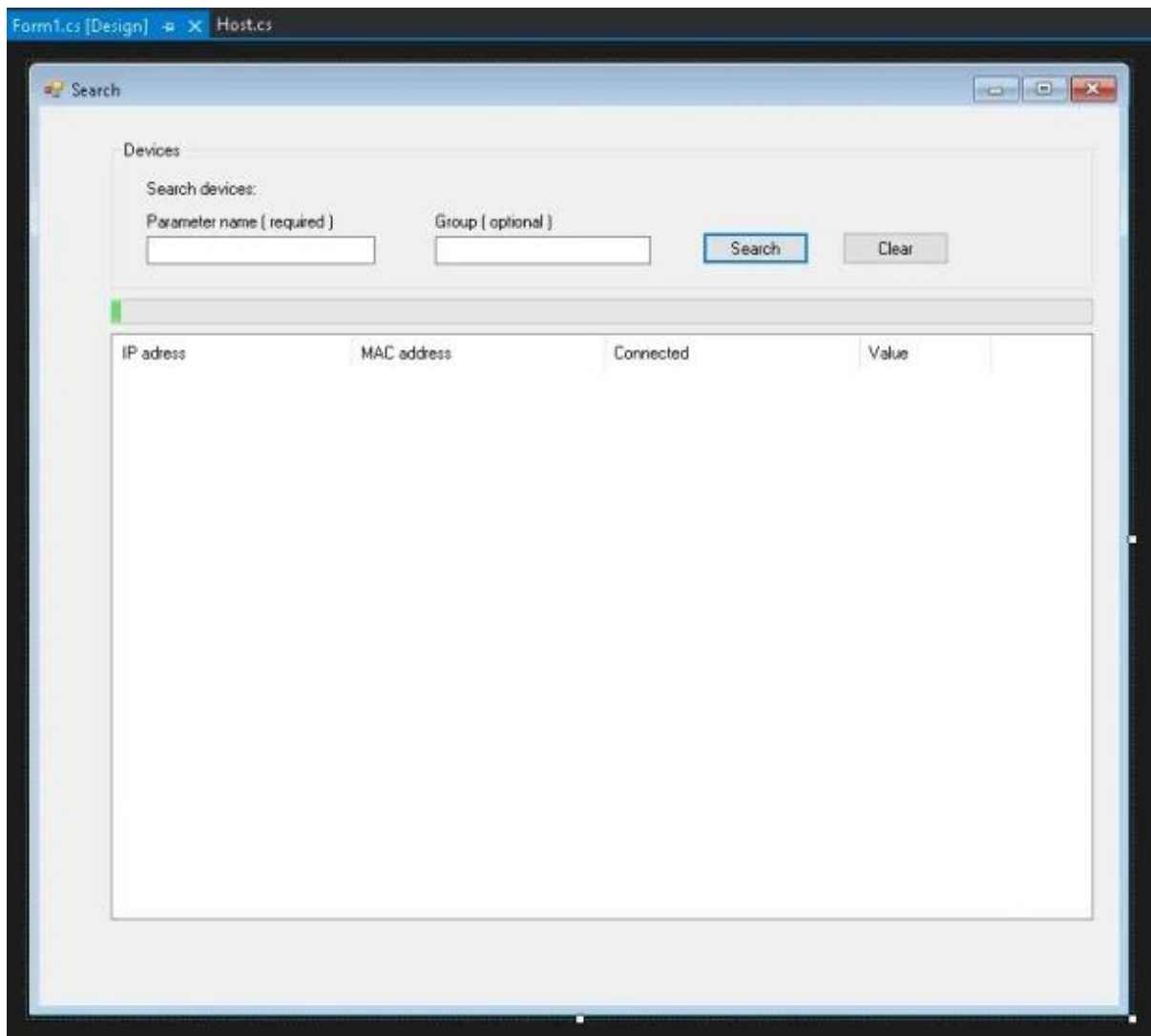
Kad sam identifikovao sve ove funkcije koje su mi potrebne za izradu zadatka, sledeći korak je bio da iste te funkcije pronađem u njihovoj C# biblioteci čiji API sadrži sve te funkcije za rad.

## Dan 3 - Upoznavanje sa DMV\_API bibliotekom

Ova biblioteka nije ni malo mala.. Samo klasa koja se koristi kao kontroler imala je preko 40.000 linija koda. Ova biblioteka je razvijana unazad godinama i puno ljudi je radilo na njoj. Uvideo sam da veliki udeo u radu programera čini upravo razumevanje i korišćenje već napisanog koda. Za ovaj deo, potrebno je vreme i iskustvo jer ako nisu poštovani principi *clean code*, kao i ako nisu pisani komentari, programeru koji je nov na projektu potrebno je znatno više vremena da razume kod. U toku prvih nekoliko dana, uglavnom sam istraživao ovu biblioteku u potrazi za funkcijama koje su mi potrebne da realizujem svoj projekat. Kada bih ih pronašao onda bi usledio dobar deo vremena gde sam se upoznavao sa kodom kako bih znao da ih koristim u svom projektu.

## Dan 4 - Izrada korisničkog interfejsa

Korisnički interfejs za ovaj projekat nije komplikovan, sastoji se od polja za unos parametara po kojima se vrši pretraga, dugmeta za pretragu i kontrole za prikaz liste rezultata. Uz to postoji i progress bar koji pokazuje napredak tokom pretrage.



## Dan 5 - Planiranje strukture programa

U početku sam imao samo jednu klasu **Devices** koja je sadržala sva polja i funkcije ali sam uvideo da je moguće refaktorirati kod i uraditi to na bolji način. Primetio sam da postoje polja i funkcije koje koristi samo uređaj sa kog se vrši pretraga dok ostala polja i funkcije koriste uređaji koji su pronađeni na mreži. Stoga sam napravio dve zasebne klase **Host** i **NetworkDevice**. Klasa **Host** služi da predstavi uređaj na kome je korisnik dok klasa **NetworkDevice** služi da predstavi uređaj na mreži. Tako da je struktura programa dve klase koje komuniciraju međusobno. Pored ove dve klase bilo je potrebno da se u projekat uključi i projekat DMV\_API koje su obe klase koristile za pozivanje funkcija API-ja.

## Dan 6 - Delegati (delegates) i događaji (events)

Prilikom pretrage pokušava se uspostavljanje komunikacije sa puno različitih uređaja na mreži. Svaki uređaj je priča za sebe, neki nije podešen i komunikacija sa njim neće biti uspostavljena, s nekim će komunikacija biti uspostavljena ali tek nakon nekog perioda dok će s nekim komunikacija odmah biti uspostavljena. Takođe zahtev za parametre koji se šalje svakom uređaju može da ima različit ishod. Može doći do greške ako s uređajem nije uspostavljena konekcija ili mogu biti vraćeni svi parametri uređaja. U ovakvoj situaciji nije moguće napisati sinhroni kod koji će moći da zadovolji zahteve projekta jer postoji previše stvari koje vremenski ne znamo kada će da se izvrše. Zbog toga sam se opredelio za asinhrono programiranje koristeći koncept **delegata** i **događaja**.

**Delegat** je pokazivač ali za razliku od običnih pokazivača na promenljive, delegati su pokazivači na funkcije!

**Događaj (event)** je signal koji može da se emituje u kodu, koristi se da se odradi sinhronizacija. Kada se desi događaj onda se aktivira određeni delegat kog tipa je sam događaj. Potom se izvršava funkcija na koju pokazuje delegat. Na ovaj način se postiže asinhrono izvršavanje.

U biblioteci DMV\_API postoji funkcija kojom se zahteva od uređaja da pošalje sve svoje parametre. Kada uređaj pošalje svoje parametre i oni budu obrađeni, dolazi do događaja *ParametersProcessed*. Kako bih ja mogao da dobijem te parametre u svom projektu, potrebno je da u klasi **NetworkDevice** kreiram delegat koji će da pokazuje na funkciju koja će da se izvršava kada dođe do prijema parametara. Takođe potrebno je izvršiti pretplatu na događaj *ParametersProcessed*.

Primer korišćenja delegata i događaja:

U klasi **NetworkDevice** postoji funkcija `device_ParametersProcessed` koja služi da pokupi parametre koji su poslali od strane uređaja na mreži.

```
public void device_ParametersProcessed(Parametri_v2 processedParams)
{
    if (processedParams != null)
    {
        deviceParameters = processedParams;
        status = SearchFunction();
    }
    Host.currentDeviceIndex++;
    IndexChanged(this);
}
```

I da bi se ova funkcija izvršila, moramo da izvršimo pretplatu na događaj *ParametersProcessed* koja se nalazi u `DMV_APIController` instanci *controller* na sledeći način:

```
controller.ParametersProcessed +=
devicesOnNetwork[currentDeviceIndex].device_ParametersProcessed;
```

Posle ove linije koda, kad god dođe do događaja *ParametersProcessed* u kontroleru, onda će se tada i izvršiti moja funkcija *device\_ParametersProcessed*. **Tada pribavljam sve parametre jednog uređaja.**

## Dan 7 - Pretraga mreže

Za pretragu mreže koristio sam UDP(User Datagram Protocol) protokol. Za ovakvu pretragu gde je potrebno pretražiti **sve** uređaje na mreži, koristi se broadcast. To je kad ne šaljemo paket konkretnom uređaju nego šaljemo na adresu 255.255.255.255. Na ovaj način se paket šalje svim uređajima na mreži nakon kojeg svi odgovaraju sa svojim mrežnim parametrima (IP adresa, MAC adresa...).

```
private void Broadcast()
{
    devicesOnNetwork = new List<NetworkDevice>();
    int port = 1898;
    UdpClient client = new UdpClient();
    client.EnableBroadcast = true;
    client.Client.ReceiveTimeout = 3000;
    byte[] data = new byte[20];
    client.Send(data, data.Length, IPAddress.Broadcast.ToString(), port);
    var from = new IPEndPoint(IPAddress.Any, 0);
    var sw = new Stopwatch();
    sw.Start();
    while (sw.ElapsedMilliseconds < 2000)
    {
        try
        {
            Byte[] receiveBytes = client.Receive(ref from);
            //check header value
            if (!(receiveBytes[0] == 0 && receiveBytes[1] == 3))
            {
                Console.WriteLine("Recieved message did not have correct header [00 03].");
                continue;
            }
            //get MAC address from recieved bytes
        }
    }
}
```



```

string macAddress = "MAC:" + BitConverter.ToString(receiveBytes, 2, 6);
//get IP address from recieved bytes
byte[] ipBytes = new byte[4];
Buffer.BlockCopy(receiveBytes, 8, ipBytes, 0, 4);
IPAddress ipAddress = new IPAddress(ipBytes);
//Add device to the listNetworkDevice device = new
NetworkDevice(ipAddress, macAddress, NetworkDevice.EStatus.NotConnected);
devicesOnNetwork.Add(device);
}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
}
}
sw.Stop();
devicesOnNetwork.Sort((x, y) => -x.HostName.CompareTo(y.HostName));
}

```

## Dan 8 - Pretraga parametara uređaja

Kada dobijem sve parametre uređaja, potrebno je da izvršim pretragu po nazivu parametra i po grupi. Parameter name je obavezan parametar pretrage dok je ime grupe Group name opcioni parametar.

```

public EStatus SearchFunction()
{
    if (deviceParameters == null Host.searchParamName == null)
        return EStatus.NotConnected;
    foreach (Parametar_v2 parameter in deviceParameters.ParamList)
    {
        if (parameter.Name != Host.searchParamName)
            continue;
        if (Host.searchGroup == "" Host.searchGroup == parameter.Group)
        {
            parameterValue = parameter.ParamValue;

```

```

        return EStatus.Connected_Found;
    }

    return EStatus.Connected_NotFound;
}

```

Nakon pretrage dobijam status koji može da bude **Connected\_Found** za uređaje na kojima je pronađen odgovarajući parametar sa odgovarajućom grupom, **Connected\_NotFound** za uređaje sa kojima je uspostavljena konekcija ali traženi parametar nije nađen i **NotConnected** ako konekcija nije uspostavljena sa uređajem.

Devices

Search devices:

Parameter name (required )

CFG

Group ( optional )

General

Search

Clear

| IP address      | MAC address           | Connected       | Value         |
|-----------------|-----------------------|-----------------|---------------|
| 172.16.17.35    | MAC:00-06-55-01-58-23 | Connected_Found | 3092          |
| 172.16.17.34    | MAC:00-06-55-01-58-22 | Connected_Found | 3092          |
| 172.16.17.236   | MAC:00-06-55-01-33-EC | Connected_Found | 2400          |
| 172.16.17.231   | MAC:00-06-55-01-2E-E7 | Connected_Found | 825           |
| 172.16.17.228   | MAC:00-06-55-01-48-E4 | Connected_Found | 967           |
| 172.16.17.221   | MAC:00-06-55-01-46-63 | Connected_Found | 1828          |
| 172.16.17.21    | MAC:00-06-55-01-33-D2 | Connected_Found | 3070          |
| 172.16.17.199   | MAC:00-06-55-01-2B-62 | Connected_Found | 967           |
| 172.16.17.19    | MAC:00-06-55-01-44-0D | Connected_Found | 967           |
| 172.16.17.185   | MAC:00-06-55-01-4C-B9 | Connected_Found | 3090          |
| 172.16.17.172   | MAC:00-06-55-01-2E-38 | Connected_Found | 1119          |
| 172.16.17.101   | MAC:00-06-55-01-2C-64 | Connected_Found | 743           |
| 172.16.17.100   | MAC:00-06-55-01-58-68 | Connected_Found | 3105          |
| 172.16.15.174   | MAC:00-06-55-01-50-4A | Connected_Found | 2532          |
| 172.16.1.68     | MAC:F4-B5-20-35-ED-A5 | Connected_Found | 20020         |
| 172.16.1.56     | MAC:3C-7C-3F-24-CB-BF | NotConnected    | not available |
| 172.16.1.121    | MAC:00-09-0F-FE-00-01 | NotConnected    | not available |
| 172.16.0.105    | MAC:18-C0-4D-4B-9B-4F | Connected_Found | 20020         |
| 159.159.159.164 | MAC:00-06-55-01-51-5C | NotConnected    | not available |
| 159.159.159.163 | MAC:00-06-55-01-4C-78 | NotConnected    | not available |
| 159.159.159.162 | MAC:00-06-55-01-4C-6B | NotConnected    | not available |
| 159.159.159.161 | MAC:00-06-55-01-4C-58 | NotConnected    | not available |

Ovako izgleda korisnički interfejs kada je pretraga u toku.

## **Dan 9 - Refaktorisanje koda**

Kada su sve funkcionalnosti bile gotove, uz pomoć saveta mentora, koji mi je pomagao i nadgledao me tokom projekta, krenuo sam u refaktorisanje koda. Konkretno stvari koje sam ovde radio bile su: efikasnije i čitljivije pisanje petlji i funkcija, razdvajanje koda u dve zasebne klase koje predstavljaju dve različite celine, poboljšavanje izgleda i funkcionalnosti korisničkog interfejsa.

## **Zaključak**

Na samom kraju, po završetku izrade projekta, šef sektora je došao da pogleda projekat na kome sam radio i bio je zadovoljan rezultatom mog rada. Kroz praktičnu primenu postojećeg znanja i sticanjem novih, stekao sam vredno iskustvo u oblasti C#-a i Windows formi. Takođe sam stekao iskustvo rada u kancelariji i radnu naviku, odnosno navikavanje na radno vreme. Kancelarija je dobro opremljena i radna atmosfera u kancelariji je vrlo prijatna. Rad je bio uvek bez pritiska, mogao sam da izaberem koliko vremena ću da provedem na poslu i kada da napravim pauzu. Mentor je bio odličan i uvek dostupan. Uz njegovu pomoć sam završio sve sa lakoćom, stekao nova znanja i iskustvo.