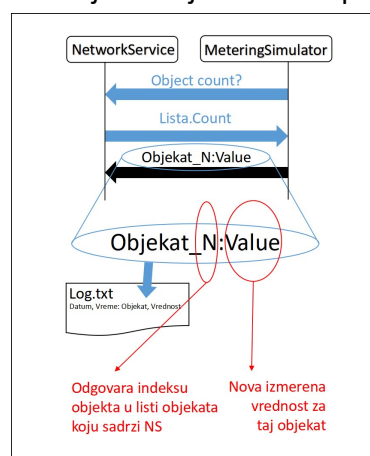


Predmetni zadatak 2 (PZ2) - 30 poena

- **Ime projekta:** Simulator infrastrukturnog sistema
- **Opis projekta:** WPF MVVM aplikacija koja predstavlja servis (**NetworkService**) za monitoring izmerenih vrednosti nad entitetima u sistemu. Merenja se dobijaju od već implementirane aplikacije **MeteringSimulator**.

NetworkService omogućava svom korisniku tabelarni prikaz entiteta u sistemu, njihov raspored na *Drag&Drop* mreži, kao i prikaz podataka pomoću grafikona. **MeteringSimulator** će u nasumičnim vremenskim trenucima slati podatke ka **NetworkService** aplikaciji uz indicaciju za koji entitet se prosleđuje dati podatak.



opis komunikacije

- **Ko može da koristi program?** Aplikacija se razvija za jednu od pet ciljnih grupa:
 - **CG1 - Dispečerski, kontrolni sistem** - kontrolni sistemi služe da se prate i kontrolišu industrijske mašine i procesi i često su automatizovane. Korisnici dispečerskih sistema imaju visoko znanje iz upotrebe aplikacije i potrebni su im načini da se brže kreću kroz aplikaciju i da se neke akcije kompletiraju mnogo brže. Važno je da aplikacija bude pregledna (elementi interfejsa moraju biti krupni i pregledno raspoređeni) i da nema nejasnoća šta koji deo aplikacije ima za cilj. Paleta boja koja se koristi mora biti ograničena, tako da postoji jak kontrast između boja, ali da one ne umaraju vid korisnika, budući da će puno vremena provoditi gledajući u korisnički interfejs kontrolnog sistema. Korisnički interfejs aplikacije treba da bude organizovan tako da nalikuje na kontrolnu tablu u okviru koje će korisniku konstantno biti prikazana mreža za prevlačenje entiteta, a na preostalom delu prozora će se moći navigirati između ostalih *View*-ova aplikacije. *(Dodatne funkcionalnosti će biti navedene kasnije u tekstu)*
 - **CG2 - Linux korisnici** - aktivno koriste *Linux* operativne sisteme i većina njih je navikla na upotrebu terminala prilikom zadavanja komandi računaru. Dosta njih više koristi tastaturu u svom radu, sa što manje upotrebe miša zbog vremenske efikasnosti. Omogućiti korisnicima da mogu i pomoću jednog ili više mini-

terminala zadavati komande (mora postojati mogućnost i redovnog korišćenja aplikacije upotrebom standardnih kontrola, bez terminala). Raditi sa pretpostavkom da će se aktivno koristiti prečice za izvršavanje komandi (npr. *Tab* dugme) i navigaciju, a miš samo u nužnoj potrebi. Raditi i sa pretpostavkom da korisnici koriste aplikaciju sa smanjenom pažnjom, a da iz oblasti upotrebe aplikacije imaju visoko znanje. Potrebno je omogućiti da se u svakom trenutku vrate na prethodni *View*, da potvrde svoju akciju i da im se omogući poništavanje poslednje obavljene akcije - *Undo*.

- **CG3 - Korisnici mobilnih telefona** - Oslanjaju se na "pointerski" uređaj, a za unos teksta treba implementirati "virtuelnu" tastaturu (pojavljuje se na ekranu i pomoću miša se pritiskaju dugmad na njoj - ne može biti virtuelna tastatura koja je deo operativnog sistema i ne može se koristiti *NuGet* paket). Kretanje kroz aplikaciju se obavlja pomoću menija, a sve vreme tokom korišćenja aplikacije korisnici imaju dostupno "*Home*" dugme da se vrate na početni *View* aplikacije, kao i dugme koje omogućava poništavanje poslednje obavljene akcije - *Undo* (omogućava takođe i povratak na prethodni prikaz). Funkcionalnost *Undo* treba da bude implementirana tako da je moguće pojedinačno poništavati jednu po jednu akciju prolazeći kroz istoriju svih izvršenih akcija. Aplikacija će se ponašati kao emulator za mobilni telefon, te će dimenzije prozora odgovarati dimenzijama ekrana mobilnog uređaja (u obliku uspravnog pravougaonika - *portrait mode*).
- **CG4 - Službenici** - Vrlo dugo koriste računar. Naviknuti su na informacione sisteme sa upotrebom u oblasti knjigovodstva. Potrebni su im jasni i precizni podaci u vidu brojeva, procenata i grafikona. Raditi sa pretpostavkom da im je vrlo bitno da ne naprave grešku u radu i da će koristiti prečice za najčešće korišćene akcije, kao i da poništavaju poslednje obavljenu akciju - *Undo*. Na *Network Entities* i *Network Display View*-ovima mora postojati mogućnost za poništavanje svih do tog trenutka izvršenih akcija odjednom. Na posebnom mestu u okviru prikaza izlistati sve izvršene akcije (npr. kao "*History*" paleta). Iz oblasti upotrebe aplikacije imaju visoko znanje. (*Dodatne funkcionalnosti će biti navedene kasnije u tekstu*)
- **CG5 - John Smith** - Srednje iskusan u radu sa računarom. Zna obaviti svakodnevne aktivnosti, poput gledanja filma i surfovanja internetom. Iz oblasti upotrebe aplikacije zna vrlo malo, ali je zaposlen u novoj firmi i sada mora da nauči. Prvi put vidi sličnu aplikaciju i najviše će se osloniti na *ToolTip*-ove i druge korisne informacije u toku rada. Za kompleksnije akcije, ponuditi *Help* (mora biti jasno naznačena sekcija u kojoj se nalazi, a sadržaj mora biti detaljan, konkretan, jasno i precizno formulisan za korišćenje funkcionalnosti *View*-a koji je trenutno prikazan). Kako vreme bude prolazilo biće efikasniji, potražiće prečice i brže načine izvršavanja zadataka. Nakon što savlada upotrebu aplikacije, neophodna je mogućnost da se isključi prikaz pomoći i *ToolTip*-ova.

[Date pretpostavke u opisima služe za kreiranje raznolikosti zadataka i ne trebaju biti shvaćene kao realne predstave konkretnih grupa korisnika.]

- **Feature list (Zahtevi):**

- Svaki put kada pristigne nova vrednost merenja, potrebno ju je upisati u *Log* datoteku (.txt) na disku sistema. Upisuje se vremenski trenutak kada je stiglo merenje, kao i vrednost koja je pristigla, uz referencu za koji objekat je pristiglo dato merenje.
- **Network Entities View:** Potrebno je omogućiti dodavanje i brisanje entiteta za monitoring i potrebno je osnovne podatke o njima čuvati u vidu tabele (novodobijena brojeva vrednost merenja se prikazuje u jednoj od kolona tabele).
 - Prilikom dodavanja, potrebno je dodeliti vrednosti svim svojstvima entiteta. Uz prikaz tabele treba da postoji forma za dodavanje novog entiteta u skladu sa njegovim atributima (navedeno kasnije u tekstu).
 1. **CG1** - Dodavanje entiteta u sistem će se za ovu ciljnu grupu realizovati kroz klik na dugme i podaci o njemu će biti programski kreirani i dodati u sistem, tako što će korisnik birati samo tip entiteta koji se dodaje. Prilikom navigacije, ovaj View će biti zamenjen sa View-om za prikaz grafikona.
 - Za brisanje objekta potrebno je uz tabelu na interfejsu dodati dugme, klikom na koje će se ukloniti entitet koji je označen u tabeli.
 - Korisnik mora imati mogućnost da pretražuje ili filtrira prikaz u tabeli:
 - **Pretraga (P1)** se realizuje preko dva *RadioButton*-a, koji daju opcije „naziv” i „tip”, i *TextBox*-a, u koji se unosi kriterijum pretrage. Izborom neke od opcija i unosom (dela) imena, u tabeli se prikazuju samo entiteti čiji se naziv ili ime tipa delom ili kompletno poklapaju sa unesenim tekstom.
 - **Filter (P2)** se realizuje preko *ComboBox*-a, u kojem su izlistani tipovi i tri *RadioButton*-a koji daju opcije „<”, „>” ili „=” uz *TextBox* u koji se unosi broj, pa je rezultat filter funkcionalnosti ili samo izlistavanje entiteta izabranog tipa, ili objekata čiji je ID veći ili manji od unete vrednosti zavisno od označenog *RadioButton*-a ili kombinacija obe opcije (mogućnost prikaza objekata koji su i odgovarajućeg tipa i čiji ID zadovoljava uneti uslov).
 1. **CG1** - Potrebno je omogućiti da se filteri odnosno pretrage sa svojim parametrima mogu sačuvati (na primer, u *ComboBox*-u) da bi se mogli ponovo koristiti bez ponovnog popunjavanja parametara.
 2. **CG4** - Neophodna je manipulacija podacima kod tabelarnog prikaza u vidu dodatnog filtera koji pokazuje objekte čije je poslednje izmereno stanje bilo van/unutar opsega očekivanih vrednosti. Neophodno je omogućiti kombinovanje ovog dodatnog filtera sa osnovnim filterom/pretragom, tako da se prikazuju samo oni rezultati koji zadovoljavaju sve unete kriterijume.

Za svaki od primenjenih filtera/pretraga mora da postoji mogućnost da se prikaz rezultata vrati na prikaz svih elemenata (poništanje filtera/pretrage).

Nakon dodavanja ili brisanja objekata, simulatorska aplikacija mora da se restartuje (realizovati automatski) i ona će tada sama da prikupi podatke koliko ukupno postoji objekata u (**NetworkService**) aplikaciji, za koje treba da daje podatke o stanju.

- **Network Display View:** Sadrži prostor gde će se nalaziti vizuelni prikazi entiteta i “simulirati” njihovo mesto u sistemu/mreži (minimum 12 *canvas* kontrola). Raspored entiteta nije bitan i njega određuje korisnik tako što pomera njihove prikaze (slike) *Drag&Drop* tehnikom.
 - Obavezno je da uz prikaz slike u okviru *canvas*-a pišu i podaci o entitetu koji je prevučen na taj *canvas* (ID, poslednja izmerena vrednost).
 - Ako simulirana novoizmerena vrednost za dati entitet bude ispod ili iznad zadate granice potrebno je omogućiti upadljivu vizuelnu izmenu prikaza, koja može biti u vidu notifikacije, promene boje, ili promene statusne slike. Autoru ostaje da odluči kako će izgledati vizuelne predstave, simboli normalnog stanja i stanja opasnosti.
 - *Drag&Drop* akcija će funkcionisati tako što su pored mreže sa poljima za prevlačenje, u okviru *TreeView* kontrole, grupisani po tipu, izlistani svi entiteti koji kada se prevuku na *Drag&Drop* mrežu više nisu izlistani u *TreeView* kontroli dok se ne uklone sa *Drag&Drop* mreže čime se vraćaju u *TreeView*.
 1. **CG1** - treba omogućiti da, pored „ručnog“ prevlačenja entiteta postoji dugme klikom na koje će se svi izlistani entiteti rasporediti na slobodna mesta na *Drag&Drop* mreži. Ovaj *View* će biti prikazan konstantno tokom izvršavanja aplikacije.
 - Takođe, moguće je prevlačiti entitete koji su već smešteni na *Drag&Drop* mreži u neko drugo slobodno polje na njoj.
 - Pored svega ovoga, treba omogućiti da se entiteti mogu povezivati linijama (na autoru ostaje da smisli način kako će se iscrtati linija), koje će morati da se pomeraju pri pomeranju entiteta na *Drag&Drop* mreži, sa ciljem da se predstavi veza između njih. Takođe, ako se entitet obriše iz liste svih entiteta biće uklonjen sa prikaza zajedno sa svim linijama sa kojima je povezan. Slično, ako se entitet ukloni sa *Drag&Drop* mreže i vrati u *TreeView*, sve linije sa kojima je povezan se uklanjaju.
 - Sprečiti iscrtavanje više linija između istih entiteta.
 - Čuvati kompletan sadržaj ovog *View*-a, tako da se prilikom navigacije on ne gubi.

- **Measurement Graph View:** Na osnovu podataka zapisanih u *Log* datoteci, treba prikazati istoriju promena stanja pomoću grafikona. Grafikoni će se konstantno menjati (u "realnom vremenu", bez potrebe da korisnik ručno osveži prikaz), jer se konstantno dobijaju nove informacije. Grafikon prikazuje poslednjih pet primljenih vrednosti za izabrani objekat. Potrebno je omogućiti pomoću *ComboBox*-a biranje entiteta za koji se iscrtava grafikon (istorija merenja). Prilikom iscrtavanja grafikona, izabrati boje koje će se koristiti za prikaz validnih i nevalidnih (ispod ili iznad zadate granice) izmerenih vrednosti i adekvatno ih upotrebiti. **Grafikoni se programski crtaju, bez upotrebe gotovih *Chart* kontrola.** Grafikoni se realizuju na jedan od tri načina:

- **G1 - Graf u vidu kružnih markera, povezanih linijama, prema vremenu** - Graf se iscrtava kao markeri u obliku krugova, povezanih linijama, koji predstavljaju menjanje vrednosti (po Y-osi) i unutar samih krugova su navedene vrednosti, pa nema potrebe za označavanjem podeoka Y-ose, već samo vremenskih trenutaka za date vrednosti na X-osi.
- **G2 - Graf u vidu barova prema vremenu** - Graf se iscrtava kao pravougaonici koji rastu u visinu koja odgovara izmerenim vrednostima. Obe ose grafikona trebaju biti označene.
- **G3 - Graf u vidu krugova različitih poluprečnika prema vremenu** - Graf se iscrtava kao krugovi poređani na vremenskoj osi, tako da su im centri poravnati paralelno po X-osi, a oni su pomereni za najveći poluprečnik po Y-osi. Poluprečnik svakog kruga odgovara vrednosti merenja koja se prikazuje. Vremenski trenuci moraju biti označeni na X-osi.

1. **CG1** - Prilikom navigacije, ovaj *View* će biti zamenjen sa *Network Entities View*-om.
2. **CG4** - Iscrtati dodatni grafikon koji pokazuje koliko entiteta kog tipa postoji u sistemu - grafikon se ponaša kao pravougaonik podeljen na dve (ili tri, zavisno od broja tipova) celine koje odgovaraju procentu zastupljenosti datog tipa.



- **Način modelovanja entiteta:**

- **T1 - Pritisak u ventilima** - entiteti se modeluju kao merači pritiska sa atributima ID, naziv i tip (može biti Kablovski senzor i Digitalni manometar). Kada je u pitanju vrednost koja se meri, ona treba da bude između 5 i 16 Mega Paskala (MP). Sve osim toga predstavlja nevalidnu vrednost.
- **T2 - Potrošnja struje** - entiteti se modeluju kao merači potrošnje struje sa atributima ID, naziv i tip (može biti *Interval Meter* i *Smart Meter*). Kada je u pitanju vrednost koja se meri, ona treba da bude između 0,34 i 2,73 kWh. Sve osim toga predstavlja nevalidnu vrednost.
- **T3 - Dnevni saobraćaj** - entiteti se modeluju kao putevi sa atributima ID, naziv i tip (koji može biti IA i IB). Kada je u pitanju vrednost koja se meri, ona treba da ima vrednost do 15000 na putevima IA reda, a do 7000 na putevima IB reda. Sve vrednosti iznad predstavljaju nevalidnu vrednost.
- **T4 - Proizvodnja vetro i solarnih generatora** - entiteti se modeluju kao distribuirani energetske resursi (DER) sa atributima ID, naziv i tip (može biti Solarni panel i Vetrogenerator). Kada je u pitanju vrednost koja se meri, ona treba da bude između 1 i 5 MW. Sve osim toga predstavlja nevalidnu vrednost.
- **T5 - Protok vode** - entiteti se modeluju kao merači protoka vode sa atributima ID, naziv i tip (može biti Zapreminski, Turbinski i Elektronski). Kada je u pitanju vrednost koja se meri, ona treba da bude između 670 i 735 litara. Sve osim toga predstavlja nevalidnu vrednost.
- **T6 - Opterećenje servera** - entiteti se modeluju kao serveri sa atributima ID, naziv, IP adresa i tip (može biti *Web server*, *File server*, *Database server*). Kada je u pitanju vrednost koja se meri, ona treba da bude između 45 i 75%. Sve osim toga predstavlja nevalidnu vrednost.
- **T7 - Temperatura reaktora** - entiteti se modeluju kao oprema za merenje temperature sa atributima ID, naziv i tip (može biti RTD i TermoSprega). Kada je u pitanju vrednost koja se meri, ona treba da bude između 250 i 350 stepeni Celzijusa. Sve osim toga predstavlja nevalidnu vrednost.

ID treba da bude ceo broj (int) i on mora biti jedinstven.

Tip entiteta se formira kao dodatna klasa sa poljem ime i slika (putanja do slike) koji bliže određuju tip. Tipovi su unapred definisani i kada se kreira objekat, u formi postoji izbor tipa iz *ComboBox*-a, zavisno od teme. Za svaki tip treba da postoji unapred definisana slika.

- **Napomene:**

- Aplikacija treba da bude implementirana u skladu sa MVVM obrascem i da sadrži mehanizme *DataBinding*-a.
- Moraju postojati prečice za navigaciju kroz prikaze aplikacije, a ako postoje *Expander* ili slične kontrole čiji sadržaj može biti sakriven, za prikaz njihovog sadržaja treba da se implementira prečica.

- Svaka ciljna grupa koja ima kao traženu funkcionalnost *Undo*, će je implementirati kao **poništanje efekta poslednje realizovane komande**.
- Estetika (raspored, poravnanja, konzistentnost) i prilagođenost temi (palette boja, slike, fontovi, ikonice, animacije itd.) se takođe boduju. Svi prikazi aplikacije moraju biti konzistentni i prilagođeni temi. Potrebno je da budu zastupljeni svi navedeni aspekti (palette boja, ikonice, fontovi) dizajna korisničkog interfejsa aplikacije.
- Na svim prikazima gde se unosi bilo koja vrednost mora postojati validacija svih polja i izbora. Validacija mora biti implementirana za svako polje pojedinačno (poruka o grešci se ispisuje ispod konkretnog polja). Validacione poruke ne mogu biti prikazivane kroz *MessageBox*, moraju biti smislene i prilagođene svakom polju pojedinačno. **Upotreba poruke "Polja nisu ispravno popunjena" nije adekvatna za različite tipove grešaka.**
- Prilikom brisanja objekta, zatražiti od korisnika potvrdu akcije.
- Za svaku uspešno kompletiranu akciju (dodavanje, brisanje) korisniku prikazati povratnu informaciju (može biti *MessageBox* ili *Toast* notifikacija). Ako se u izradi projekta izabere *MessageBox*, on se mora konzistentno koristiti, uz navođenje **svih parametra** (sadržaj, naslov, ponuđena dugmad i ikonice).
- Davati korisniku aplikacije povratne informacije u vidu promena kursora miša i prikazivanja *Tooltip*-ova sa adekvatnim sadržajem.
- Prilikom pisanja koda, poštovati *C#* konvencije za imenovanje, i pisati što „čistiji“ kod (nazivi kontrola, promenljivih, funkcija, klasa itd). Jezik na kome se piše kod mora biti **konzistentan kroz čitavu aplikaciju** (poželjno engleski). Jezik korisničkog interfejsa takođe mora biti **konzistentan kroz čitavu aplikaciju**.
- Na odbranu je potrebno doneti projekat sa minimum tri već kreirana entiteta popunjena validnim podacima koji su prilagođeni temi (relevantni za temu), kao i slikama prilagođenim temi.
- Neophodno je znati do najsitnijih detalja šta i kako svaka linija koda u rešenju radi i čemu služi. Za neadekvatna objašnjenja, nepoznavanje i nerazumevanje sopstvenog koda **će se gubiti bodovi**. Isto važi i **za prepisivanje i plagijarizam**.
- **Konkurencija/Inspiracije:** Različiti SCADA softveri
- **Wireframe:** Potrebno je napraviti *low-fidelity* prototip rešenja u skladu sa dobijenom kombinacijom zadatka, slično primeru iz 4. nedelje vežbi.
- **Rok:**
 - **Za kompletiran Wireframe - 9. nedelja vežbi**, svaka grupa u dodeljenom terminu odbrane.
 - **Za kompletiran PZ2 - 10. nedelja vežbi**, svaka grupa u dodeljenom terminu odbrane.