

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 indikaciju za koji entitet se prosleđuje dati podatak.

- **Ko može da koristi program?** Aplikacija se razvija za jednu od pet ciljnih grupa:
 - **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 obavljenju akciju - *Undo*. Iz oblasti upotrebe aplikacije imaju visoko znanje. *(Dodatne funkcionalnosti će biti navedene kasnije u tekstu)*
- **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).
 - Za brisanje objekta potrebno je uz tabelu na interfejsu dodati dugme koje će brisati entitet koji je označen u tabeli.
 - Korisnik mora imati mogućnost da pretražuje ili filtrira prikaz u tabeli:

- **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).

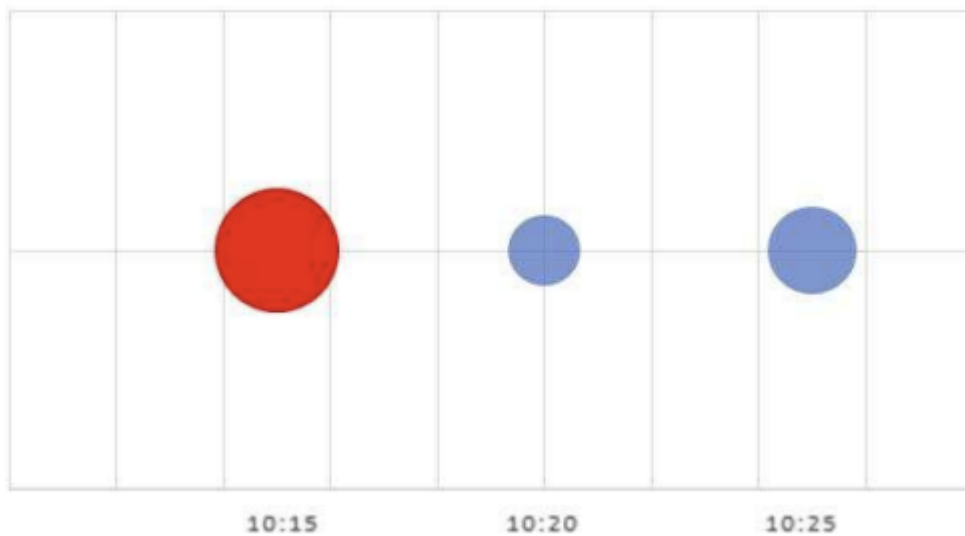
2. CG4 - Neophodna je manipulacija podacima kod tabelarnog prikaza u vidu dodatnog filtera koji pokazuje objekte čije je poslednje izmereno stanje bilo van/unutar opsega očekivanih vrednosti. Neophodno je omogućiti kombinovanje ovog dodatnog filtera sa osnovnim filterom/pretragom, tako da se prikazuju samo oni rezultati koji zadovoljavaju sve unete kriterijume.

Nakon dodavanja ili brisanja objekata, simulatorska aplikacija mora da se restartuje (može se realizovati automatski ili ručno) 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*.

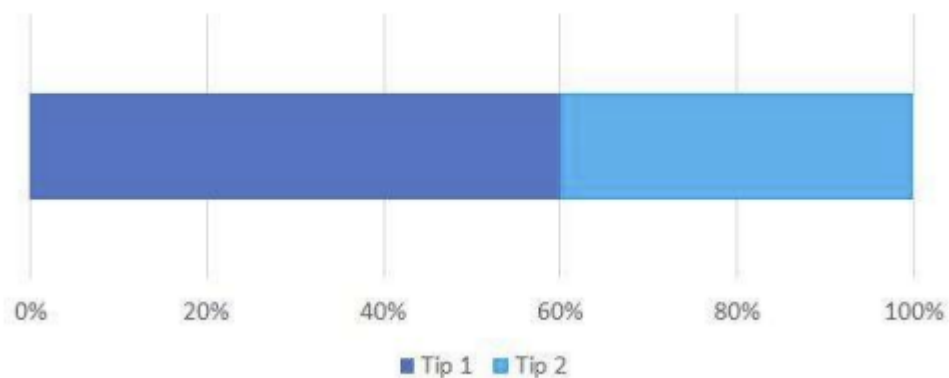
- Takođe, moguće je prevlačiti entitete koji su već smešteni na *Drag&Drop* mreži u neko drugo slobodno polje na njoj.
 - Pored svega ovoga, treba omogućiti da se entiteti mogu povezivati linijama (na autoru ostaje da smisli način kako će se iscrtati linija), koje će morati da se pomeraju pri pomeranju 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 listu, 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:
- **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.

G3



2. **CG4** - Iscrutati 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.

CG4



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.

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 **ponišćavanje efekta poslednje realizovane komande**.

2. CG4 - 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).

- 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 - 12. nedelja vežbi**, svaka grupa u svom terminu, dok će za stare studente termin odbrane biti blagovremeno objavljen na sajtu predmeta.
 - **Za kompletiran PZ2 - 15. nedelja vežbi**, svaka grupa u svom terminu, dok će za stare studente termin odbrane biti blagovremeno objavljen na sajtu predmeta.