

## Projektovanje i implementacija softverske arhitekture sistema za nadzor temperature polova rotora u hidroelektrani Đerdap 2

Blagoje Babić<sup>1</sup>, Saša Milić<sup>1</sup>, Miša Kožicić<sup>2</sup>, Nikola Miladinović<sup>1</sup>

<sup>1</sup>Univerzitet u Beogradu, Elektrotehnički institut Nikola Tesla, Koste Glavinića 8a, 11000 Beograd, Srbija

[blagoje.babic@ieent.org](mailto:blagoje.babic@ieent.org)

[s-milic@ieent.org](mailto:s-milic@ieent.org)

[nidzo@ieent.org](mailto:nidzo@ieent.org)

<sup>2</sup>EPS AD Beograd, ogranak HE Đerdap, Srbija

[misa.kozicic@eps.rs](mailto:misa.kozicic@eps.rs)

**Kratak sadržaj:** U radu je prikazana arhitektura softverskog sistema za nadzor temperature polova rotora u hidroelektrani Đerdap 2. Sistem je realizovan korišćenjem .NET 10 platforme i zasnovan je na slojevitoj arhitekturi sa primenom *repository pattern*-a, čime je omogućena jasna separacija odgovornosti između komponenti sistema. Prikupljeni podaci sa senzora se skladište u SQL bazu korišćenjem ADO.NET tehnologije. Poseban fokus stavljen je na sigurnost, skalabilnost i mogućnost daljeg razvoja sistema u realnim industrijskim uslovima. Doprinos rada ogleda se u definisanju prilagođene slojevite arhitekture za industrijski sistem nadzora, koja omogućava efikasnu obradu velikog broja mernih podataka u realnom vremenu, uz primenu Model-View-ViewModel (MVVM) obrasca za organizaciju korisničkog interfejsa i bolju separaciju prezentacionog sloja.

**Ključne reči:** Slojevita arhitektura, Model-View-ViewModel (MVVM), *repository pattern*, ADO.NET, SQL, merenje temperature, polovi rotora, hidrogenerator

### 1. Uvod

U radu je prikazana značajno unapređena arhitektura softverskog sistema za merenje i nadzor temperature polova rotora u hidroelektrani Đerdap 2, kao jednog od bitnih parametara za praćenje ispravnosti i

bezbednog rada hidrogeneratora [1]-[7]. Stabilan i precizan nadzor ovih vrednosti od suštinskog je značaja za prevenciju potencijalnih kvarova i optimizaciju rada postrojenja, posebno u uslovima kontinuiranog rada generatora.

Sistem je realizovan korišćenjem .NET 10 platforme, koja omogućava visok nivo performansi, skalabilnosti i podršku za savremene obrasce razvoja softvera. Arhitektura sistema je zasnovana na slojevitoj organizaciji [8], čime je postignuta jasna podela odgovornosti između različitih delova sistema i unapređena održivost i proširivost celokupnog rešenja. U okviru arhitekture primenjen je *repository pattern* [9], koji omogućava apstrakciju pristupa podacima i smanjuje direktnu zavisnost upravljačke (poslovne) logike od baze podataka.

Prikupljeni podaci sa senzora, kao i alarmna stanja, skladište se u SQL bazu podataka korišćenjem ADO.NET tehnologije [10][11], čime je obezbeđena efikasna i direktna komunikacija sa bazom uz kontrolu nad izvršavanjem upita i obradom podataka. Na ovaj način je omogućeno pouzdano čuvanje velikog broja mernih vrednosti u realnom vremenu, što je od ključne važnosti za industrijske monitoring sisteme. Pored podataka sa senzora i definisanih alarmnih stanja, u bazu se takođe skladište i podaci koje softver dobija putem Modbus protokola od eksternih višekanalnih elektronskih mernih sklopova, koji u realnom vremenu šalju merne vrednosti (temperature) ka delu za obradu podataka. Na ovaj način je obezbeđena integracija različitih izvora podataka i centralizovano čuvanje svih relevantnih informacija u okviru jedinstvenog sistema za nadzor.

U okviru sistema je implementiran osnovni mehanizam autentifikacije (*Basic* autentifikacija) koji omogućava identifikaciju korisnika i kontrolu pristupa funkcionalnostima aplikacije. Nakon uspešne prijave, korisniku se dodeljuje odgovarajuća uloga, na osnovu koje se definišu njegova prava pristupa. Podaci u bazi su dodatno zaštićeni tako što se čuvaju u hešovanom obliku, čime se onemogućava njihovo direktno čitanje i zloupotreba. Hešovanje predstavlja proces pretvaranja originalnih podataka u niz znakova koji se ne može jednostavno vratiti u prvobitni oblik. Umesto toga, prilikom provere (npr. lozinke), ulazni podatak se ponovo hešuje i dobijeni rezultat se poredi sa već sačuvanim hešom.

Poseban fokus je stavljen na sigurnost, skalabilnost i mogućnost daljeg razvoja sistema u realnim industrijskim uslovima. Dizajn sistema omogućava jednostavno proširenje funkcionalnosti, integraciju novih senzora i prilagođavanje različitim industrijskim scenarijima bez potrebe za značajnim izmenama postojeće arhitekture.

Dodatno, u korisničkom delu sistema primenjen je MVVM (Model-View-ViewModel) obrazac [12],[12], koji omogućava jasnu separaciju prezentacionog sloja od poslovne logike i olakšava održavanje i testiranje korisničkog interfejsa. Ovaj pristup doprinosi boljoj organizaciji koda i većoj fleksibilnosti u razvoju klijentskih aplikacija.

Doprinos rada ogleda se u definisanju i implementaciji prilagođene slojevite arhitekture za industrijski sistem nadzora, koja omogućava efikasnu obradu velikog broja mernih podataka u realnom vremenu, uz primenu savremenih softverskih principa i obrazaca dizajna. Time se obezbeđuje stabilna osnova za dalji razvoj i unapređenje sistema u skladu sa zahtevima modernih industrijskih okruženja

## **2. Zahtevi i opis sistema**

### **2.1. Funkcionalni zahtevi**

Softverski sistem za nadzor temperature polova rotora u hidroelektrani Đerdap 2 dizajniran je tako da obezbedi kontinuirano prikupljanje, obradu i prikaz mernih podataka u realnom vremenu. Funkcionalni zahtevi sistema obuhvataju prikupljanje podataka sa različitih izvora:

- Pirometar Optris CT 4ML [14]
- Podaci dobijeni putem Modbus protokola od eksternih mernih podsklopova, koje u realnom vremenu šalju izmerne vrednosti temperatura sa stacionarnih davača generatora ka sistemu.
- Istorijski podaci o merenju.

Svi podaci o temperaturama se potom obrađuju, validiraju i prikazuju u grafičkom korisničkom interfejsu (graphical user interface – GUI). Obradom dobijenih vrednosti se, po potrebi, generiše prikaz upozoravajućih i havarijskih alarmnih stanja, skladište sirovi merni podaci i alarmni događaji u SQL bazu podataka. Dodatne funkcionalnosti programske aplikacije omogućavaju upravljanje podacima, uključujući brisanje određenih zapisa iz baze u skladu sa korisničkim ovlašćenjima, konfiguraciju sistemskih parametara od strane administratora (npr. intervali snimanja podataka, pragovi alarma), itd.

Navedene funkcionalnosti omogućavaju da sistem funkcioniše kao centralizovani nadzorni program koji je razvijen za industrijsko okruženje sa visokim zahtevima za pouzdanost i dostupnost podataka.

### **2.2. Monitoring zahtevi**

Pored funkcionalnih zahteva, sistem mora da zadovolji i određene monitoring zahteve koji su ključni za industrijsku primenu datog sistema, kao što su: rad u realnom vremenu – sistem mora obezbediti minimalno kašnjenje u prikupljanju i obradi podataka, pouzdanost – obezbeđivanje stabilnog rada u

kontinuiranom režimu rada bez gubitka podataka, skalabilnost – mogućnost proširenja sistema dodatnim senzorima i izvorima podataka bez značajnih izmena arhitekture, bezbednost – kontrola pristupa sistemu i zaštita osetljivih podataka, održivost i proširivost – modularna arhitektura koja omogućava jednostavno održavanje i nadogradnju sistema, performanse – efikasna obrada velikog broja mernih podataka u kratkom vremenskom intervalu.

### 2.3. Izvori podataka

Sistem integriše više različitih izvora podataka (stacionarni davači sa temperaturnih senzora i infracrveni merni uređaji) kako bi se obezbedio kompletan uvid u stanje nadziranog hidrogeneratora. Na slici 1 je prikazan dijagram očitavanja i obrade temperature u softveru sa Optris CT 4ML pirometra.

Komunikacija sa pirometrom realizovana je putem serijskog (COM) interfejsa i zasniva se na binarnom protokolu razmene podataka. Za razliku od ASCII komunikacije, kod koje se podaci prenose u obliku tekstualnih stringova, u ovom slučaju razmena podataka se odvija na nivou sirovih bajtova, čime se postiže veća brzina i efikasnost komunikacije, što je od posebnog značaja u sistemima koji rade u realnom vremenu.

Proces komunikacije započinje slanjem komandnog bajta vrednosti 0x01, kojim se inicira očitavanje trenutne procesne temperature. Nakon prijema komande, uređaj vraća odgovor u formi dva bajta, gde prvi bajt predstavlja viši bajt (MSB – Most Significant Byte), dok drugi predstavlja niži bajt (LSB – Least Significant Byte) mernog podatka.

Primljeni bajtovi se zatim kombinuju u jedinstvenu celobrojnu vrednost prema izrazu (1):

$$raw = (MSB \ll 8) \mid LSB \quad (1)$$

Dobijena vrednost se potom konvertuje u stvarnu temperaturnu vrednost primenom transformacione funkcije definisane u tehničkoj dokumentaciji proizvođača (2):

$$T = (raw - 1000) / 10 \quad (2)$$

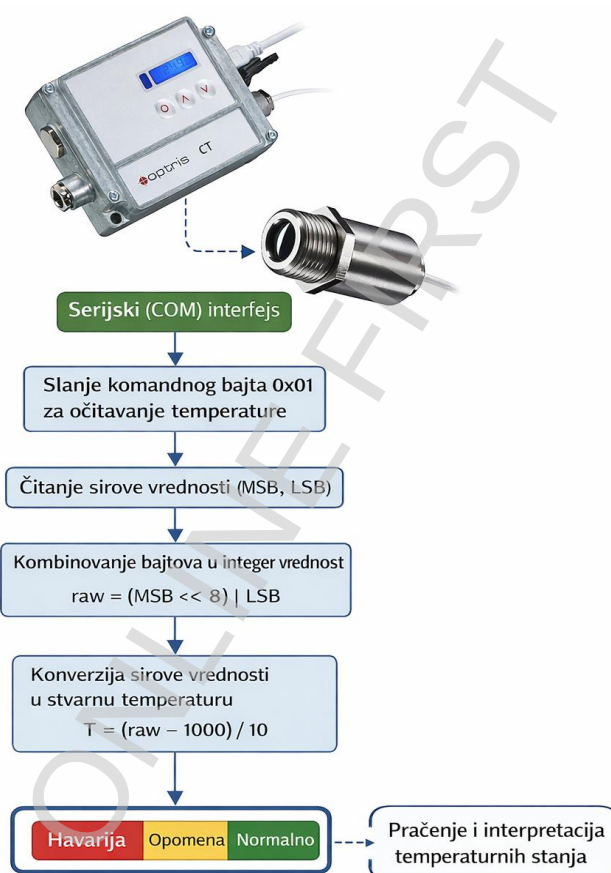
Na ovaj način se obezbeđuje precizno preslikavanje sirovih podataka u temperaturne vrednosti izražene u stepenima Celzijusa.

Pored pirometra, sistem koristi i podatke temperatura sa stacionarnih davača koje dobija putem Modbus protokola. U zavisnosti od opsega u kojem se nalazi izmerena temperatura, sistem definiše različita stanja:

- temperatura havarije
- temperatura opomene

- izmerena radna “normalna” temperatura
- stanje sonde u prekidu.

Na ovaj način je omogućena brza vizuelna interpretacija trenutnog stanja mernih tačaka, što doprinosi efikasnijem nadzoru i pravovremenom reagovanju u slučaju odstupanja od definisanih graničnih vrednosti ili kvara merne sonde.



Slika 1. Dijagram očitavanja i obrade temperature

## 2.4. Bezbednost, upravljanje sistemom i brisanje podataka

Sistem omogućava administrativno upravljanje podacima koji se nalaze u bazi. U okviru definisanih korisničkih privilegija, administrator ima mogućnost

da vrši brisanje istorijskih podataka, alarmnih zapisa ili konfiguracionih informacija. Ova funkcionalnost je posebno značajna sa aspekta održavanja baze podataka i upravljanja količinom skladištenih podataka tokom dužeg vremenskog perioda. Brisanje podataka je kontrolisano i ograničeno na korisnike sa odgovarajućim ovlašćenjima, čime se obezbeđuje sigurnost i integritet sistema.

Sistem podržava dve osnovne korisničke uloge (administrator i korisnik). Administrator ima proširena ovlašćenja i jedini ima pravo da vrši izmene ključnih parametara sistema. To uključuje podešavanje: pragova temperature za nivoe alarma (opomena i havarija), dužine snimanja signala, vremenskog intervala između uzorkovanja, komunikacionih parametara (npr. COM port), kao i drugih konfiguracionih vrednosti koje utiču na rad sistema.

Sa druge strane, korisnik bez administratorskih privilegija ima ograničen pristup sistemu. On može da prati trenutne vrednosti, pregledava istorijske podatke i koristi osnovne funkcionalnosti (npr. prikaz grafika i tabela), ali nema mogućnost izmene sistemskih parametara.

### **3. Arhitektura softverskog sistema**

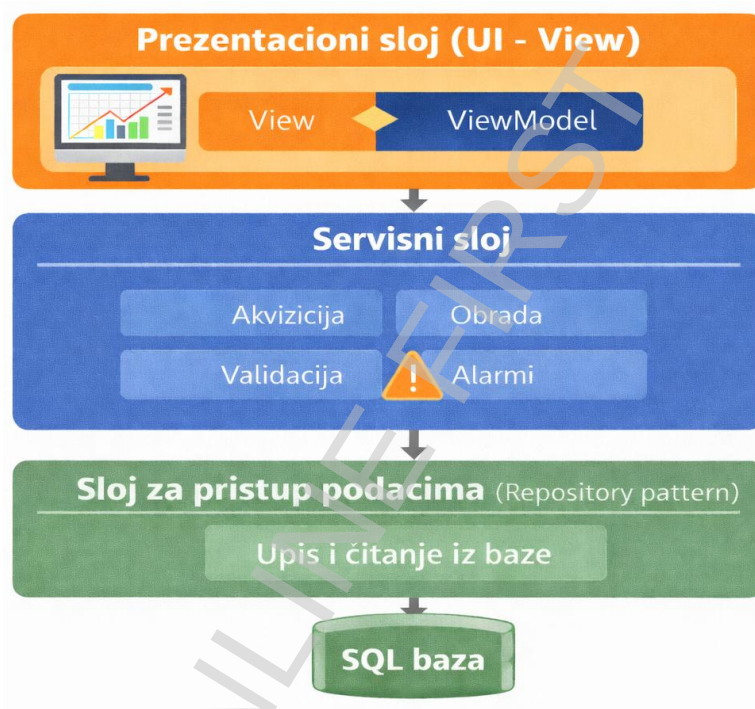
#### **3.1 Slojevita arhitektura sistema**

Softverski podsistem za nadzor temperature polova rotora projektovan je primenom slojevite arhitekture (*layered architecture*), koja omogućava jasnu podelu sistema na logički odvojene celine (slika 2). Osnovni slojevi sistema obuhvataju prezentacioni sloj, poslovni (servisni) sloj i sloj za pristup podacima.

Prezentacioni sloj je zadužen za prikaz podataka korisniku, najčešće u obliku grafika i tabelarnih prikaza, i interakciju sa sistemom, poslovni sloj implementira logiku akvizicije, obrade podataka i upravljanje alarmnim stanjima, dok sloj za pristup podacima obezbeđuje komunikaciju sa SQL bazom podataka (korišćenjem ADO.NET tehnologije). Ovakva organizacija omogućava veću preglednost sistema, kao i lakše održavanje i proširenje funkcionalnosti.

Dodatno, ovakva arhitektura omogućava nezavisni razvoj i testiranje pojedinačnih slojeva sistema. Na primer, izmene u prezentacionom sloju, poput unapređenja korisničkog interfejsa ili promene načina vizualizacije podataka, ne utiču direktno na poslovnu logiku niti na način skladištenja podataka. Slično tome, promene u bazi podataka ili načinu pristupa podacima mogu se realizovati bez potrebe za izmenama u višim slojevima, čime se smanjuje rizik od grešaka i povećava stabilnost sistema.

Pored toga, slojevita arhitektura omogućava lakšu integraciju sa spoljnim sistemima i buduća proširenja, kao što su dodavanje novih izvora podataka, implementacija naprednijih algoritama obrade ili uvođenje dodatnih modula za analizu i izveštavanje. Zahvaljujući jasno definisanim granicama između slojeva, sistem ostaje fleksibilan i prilagodljiv novim zahtevima, što je posebno značajno u industrijskim okruženjima gde se potrebe mogu menjati tokom vremena.



Slika 2. Prikaz slojevite arhitekture sistema

### 3.2 Princip razdvajanja odgovornosti (*Separation of Concerns*)

U okviru projektovanja sistema primenjen je princip razdvajanja odgovornosti (*Separation of Concerns*), koji podrazumeva da svaki deo sistema ima jasno definisanu ulogu i odgovornost. Time se smanjuje međuzavisnost komponenti i omogućava nezavisno razvijanje i testiranje pojedinačnih delova sistema.

Ovaj princip je dodatno unapređen primenom repository pattern-a u sloju za pristup podacima. Repository pattern omogućava apstrakciju rada sa

bazom podataka, čime se poslovna logika (servisni sloj) oslobađa direktne zavisnosti od konkretne implementacije skladištenja i čitanja podataka iz baze podataka. Na taj način se povećava fleksibilnost sistema i olakšava eventualna promena tehnologije baze podataka u budućnosti.

### 3.3 Tok podataka

Tok podataka kroz IC sistem započinje na nivou senzora i mernih uređaja, gde se temperature prikupljaju sa Optris pirometra putem serijske komunikacije, kao i sa sistema stacionarnih mernih davača (STK) korišćenjem Modbus protokola. Ovi podaci predstavljaju ulazne merne vrednosti koje se dalje prosleđuju servisnom sloju programa.

U servisnom sloju se vrši obrada, validacija i analiza prikupljenih podataka. U okviru ovog sloja implementirana je logika za detekciju alarmnih stanja, pri čemu se na osnovu unapred definisanih pragova identifikuju normalna, upozoravajuća i havarijska stanja sistema. Servisni sloj predstavlja centralnu komponentu sistema, jer objedinjuje komunikaciju sa izvorima podataka, obradu merenih vrednosti temperatura i upravljanje poslovnom logikom. Servisni sloj dakle prima podatke od senzora i eksternih uređaja, obrađuje ih, proverava pragove alarma i odlučuje šta treba sačuvati od podataka.

Nakon obrade, servisni sloj koristi repository pattern za komunikaciju sa slojem za pristup podacima. Repository komponenta omogućava apstrakciju rada sa bazom podataka i zadužena je isključivo za izvršavanje operacija nad bazom, kao što su upis, izmena, brisanje i čitanje podataka. Na ovaj način poslovna logika ostaje nezavisna od konkretne implementacije pristupa podacima. Skladištenje podataka realizovano je korišćenjem ADO.NET tehnologije. Sloj za pristup podacima nema poslovnu logiku (npr. ne odlučuje da li je havarijski alarm, opomena ili normalno stanje), već samo izvršava pristup podacima (read, update, insert, delete). Njegova uloga je da apstrahuje izvršavanje operacija nad bazom, čime se poslovna logika oslobađa direktne zavisnosti od SQL upita i konkretne implementacije pristupa podacima.

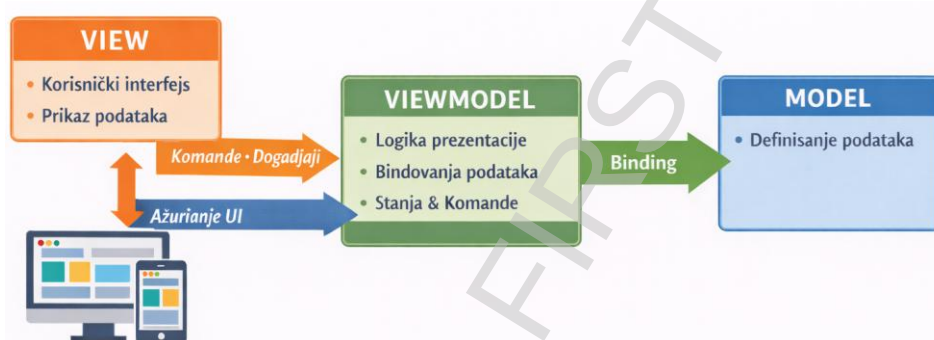
Podaci se u bazu upisuju u definisanim vremenskim intervalima, u zavisnosti od izvora. Na primer, sirovi podaci sa IC pirometra mogu se skladištiti u dužim vremenskim intervalima (npr. na svakih sat vremena), dok se podaci sa eksternih uređaja putem Modbus komunikacije beleže češće (npr. na svaka dva minuta), u skladu sa zahtevima sistema i dinamikom promena mernih veličina.

U prezentacionom sloju, koji je realizovan primenom MVVM obrasca, *ViewModel* komponenta prima obrađene podatke iz servisnog sloja i priprema ih za prikaz u korisničkom interfejsu. Korisniku se podaci prikazuju u realnom



vremenu, najčešće u obliku grafika i tabelarnih prikaza, čime je omogućen kontinuiran nadzor i praćenje stanja sistema.

MVVM (Model-View-ViewModel, slika 3) je arhitekturni obrazac koji odvaja korisnički interfejs od logike aplikacije i podataka. View predstavlja ono što korisnik vidi na ekranu. Model sadrži strukturu i definiciju podataka, dok *ViewModel* povezuje ta dva sloja tako što priprema podatke za prikaz, obrađuje komande i upravlja stanjem interfejsa. Zahvaljujući data binding-u, promene u *ViewModel*-u mogu automatski da se prikažu u *View* delu, što razvoj aplikacije čini preglednijim, lakšim za održavanje i jednostavnijim za testiranje.



Slika 3. MVVM arhitektura u .NET aplikacijama

*View* je fokusiran isključivo na prikaz i ne sadrži poslovnu logiku, dok *ViewModel* preuzima kompletnu komunikaciju sa Modelom i upravlja svim interakcijama koje dolaze iz korisničkog interfejsa. Ovakav pristup smanjuje zavisnosti između komponenti i olakšava ponovno korišćenje koda. U praksi, MVVM se često koristi u kombinaciji sa mehanizmima kao što su *ICommand* i *INotifyPropertyChanged*, koji omogućavaju efikasno upravljanje događajima i automatsko osvežavanje UI-a. Korisnički interfejs (UI) predstavlja skup vizuelnih elemenata aplikacije koji omogućavaju prikaz podataka i interakciju korisnika sa sistemom. Zahvaljujući svemu tome, aplikacije postaju fleksibilnije, skalabilnije i spremnije za dalji razvoj.

Pored prikaza trenutnih vrednosti, sistem omogućava i učitavanje arhiviranih podataka iz baze, što omogućava analizu istorijskih vrednosti, praćenje trendova i donošenje zaključaka o radu sistema u dužem vremenskom periodu. Ovakva organizacija toka podataka obezbeđuje jasno razdvajanje odgovornosti između slojeva sistema, efikasnu obradu podataka u realnom vremenu i pouzdano skladištenje mernih vrednosti, što je od ključnog značaja za industrijske sisteme nadzora.

### 3.4 Integracija više izvora podataka

Jedna od ključnih karakteristika sistema je mogućnost integracije više različitih izvora podataka u jedinstvenu arhitekturu. Sistem istovremeno obrađuje podatke sa pirometra, koji koristi binarni serijski protokol, kao i podatke sa eksternih uređaja koji komuniciraju putem Modbus protokola.

Ovakav pristup omogućava objedinjavanje heterogenih izvora podataka u jedinstveni sistem nadzora, čime se postiže potpuniji uvid u stanje nadziranog sistema i povećava pouzdanost analize.

### 3.5 Primena MVVM obrasca

U prezentacionom sloju sistema primenjen je MVVM (Model-View-ViewModel) obrazac, koji obezbeđuje jasno razdvajanje korisničkog interfejsa od logike koja upravlja prikazom i obradom podataka. Primena ovog obrasca doprinosi boljoj organizaciji koda, lakšem održavanju aplikacije i jednostavnijem testiranju komponenti korisničkog interfejsa.

U okviru MVVM obrasca, View predstavlja vizuelni deo aplikacije, odnosno forme, stranice, grafikone i tabelarne prikaze koje korisnik vidi i sa kojima ostvaruje interakciju. ViewModel predstavlja posredni sloj između korisničkog interfejsa i servisnog sloja sistema. Njegova uloga je da preuzima obrađene podatke iz servisnog sloja, priprema ih za prikaz i prosleđuje ih View komponenti u odgovarajućem formatu. Model predstavlja strukture podataka koje se koriste za prenos, prikaz i obradu informacija unutar sistema.

Važno je naglasiti da ViewModel ne komunicira direktno sa bazom podataka niti sa hardverskim uređajima. Komunikacija sa Optris pirometrom i eksternim uređajima putem Modbus protokola realizuje se u servisnom sloju, gde se obavlja i obrada, validacija i analiza podataka. Nakon toga se obrađeni podaci prosleđuju ViewModel komponenti, koja ih dalje prikazuje u korisničkom interfejsu. Na isti način, pristup arhiviranim podacima iz baze odvija se preko servisnog sloja i repository komponente, a ne direktno iz ViewModel-a.

Primena MVVM obrasca u ovom sistemu omogućava da se korisnički interfejs razvija nezavisno od poslovne logike i sloja za pristup podacima. Time se postiže jasna separacija odgovornosti: servisni sloj je zadužen za komunikaciju sa izvorima podataka i obradu informacija, repository sloj za pristup bazi, dok je ViewModel zadužen isključivo za pripremu podataka za prikaz u korisničkom interfejsu.

Na ovaj način MVVM obrazac je prirodno uklopljen u slojevit arhitekturu sistema i predstavlja važan mehanizam za organizaciju prezentacionog sloja, bez preklapanja sa odgovornostima servisnog sloja i *repository pattern*-a.

### 3.6 Prednosti primenjene arhitekture

Primena slojevite arhitekture, zajedno sa MVVM obrascem i repository pattern-om, donosi niz značajnih prednosti:

- **modularnost sistema** – jasno razdvojeni slojevi omogućavaju lakše razumevanje i održavanje sistema,
- **skalabilnost** – sistem se može proširivati dodavanjem novih izvora podataka ili funkcionalnosti bez značajnih izmena postojećeg koda,
- **održivost** – izmene u jednom sloju ne utiču direktno na ostale slojeve,
- **paralelan razvoj** – različiti timovi mogu raditi nezavisno na pojedinačnim delovima sistema (npr. jedan tim na korisničkom interfejsu, drugi na backend logici),
- **fleksibilnost** – omogućena je zamena ili unapređenje pojedinih komponenti sistema bez potrebe za kompletnom rekonstrukcijom,
- **testabilnost** – odvojeni slojevi i jasno definisane odgovornosti olakšavaju testiranje sistema.

Zahvaljujući ovakvoj arhitekturi, sistem predstavlja stabilnu i fleksibilnu osnovu za dalji razvoj i primenu u kompleksnim industrijskim okruženjima.

## 4. Prezantacioni sloj – UI

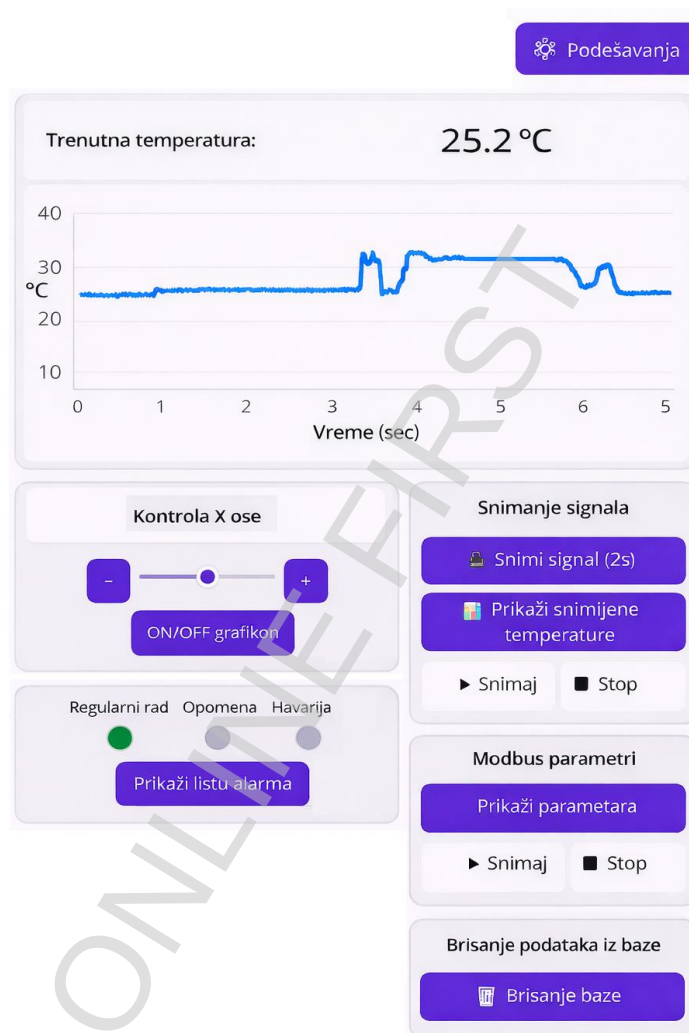
Prezantacioni sloj sistema realizovan je kao korisnički interfejs koji omogućava vizuelni prikaz mernih podataka i interakciju korisnika sa sistemom. Ovaj sloj implementiran je primenom MVVM (Model-View-ViewModel) obrasca, čime je obezbeđeno jasno razdvajanje logike prikaza od samog vizuelnog dela aplikacije. Na taj način omogućeno je lakše održavanje, proširenje i testiranje korisničkog interfejsa.

Na slici 4 prikazana je glavna stranica sistema sa grafičkim prikazom temperature.

Osnovna uloga prezentacionog sloja jeste prikaz podataka u realnom vremenu, kao i mogućnost pregleda istorijskih merenja i alarmnih stanja. Podaci koji se prikazuju u korisničkom interfejsu dolaze iz servisnog sloja, gde su prethodno obrađeni, validirani i klasifikovani u odgovarajuće kategorije.

Na glavnoj stranici aplikacije realizovan je grafički prikaz temperature dobijene pomoću Optris CT 4ML pirometra. Ovaj prikaz realizovan je u formi grafikona koji omogućava korisniku da u realnom vremenu prati promene temperature, kao i da uoči eventualne nagle skokove ili odstupanja od normalnih vrednosti. Grafikon pruža jasan vizuelni uvid u trend kretanja

temperature, što je od posebnog značaja za pravovremeno reagovanje u slučaju potencijalnih problema.



Slika 4. Deo izgled glavne UI stranice

Pored grafičkog prikaza podataka sa Optris senzora, sistem omogućava i prikaz podataka dobijenih putem Modbus protokola (slika 5). Ovi podaci se odnose na temperature sa stacionarnih davača (STK merni podsistem) i prikazani su u tabelarnom obliku, što omogućava jasan i pregledan uvid u stanje svih mernih tačaka.

## Merni kanali



Slika 5. Temperature sa stacionarnih davača preuzete pomocu Modbus protokola (simulirane vrednosti)

Temperaturne vrednosti u tabeli klasifikovane su prema unapred definisanim opsezima, pri čemu svaka kategorija ima odgovarajuću boju. Na ovaj način omogućena je brza vizuelna interpretacija stanja sistema. U zavisnosti od izmerene vrednosti, mogu se razlikovati sledeća stanja: temperatura havarije, temperatura opomene, radna "normalna" izmerena temperature i stanje sonde u prekidu.

Ispod tabele su dodatno prikazana značenja korišćenih boja, čime se korisniku olakšava interpretacija podataka i donošenje odluka na osnovu prikazanih vrednosti. Ovakav pristup omogućava brzo prepoznavanje kritičnih stanja bez potrebe za detaljnom analizom numeričkih vrednosti.

Zahvaljujući primeni MVVM obrasca, svi podaci prikazani u korisničkom interfejsu automatski se ažuriraju putem data binding mehanizma. Promene koje nastaju u ViewModel komponenti reflektuju se direktno na View, bez potrebe za dodatnom logikom u samom korisničkom interfejsu. Na ovaj način obezbeđen je efikasan i odgovarajući prikaz podataka u realnom vremenu.

Prezentacioni sloj je dizajniran tako da bude intuitivan i jednostavan za korišćenje, što je od posebnog značaja u industrijskim sistemima gde je brzina reakcije i jasnoća prikaza od ključne važnosti. Kombinacijom grafičkih i

tabelarnih prikaza omogućeno je sveobuhvatno praćenje sistema, kako kroz pregled trendova, tako i kroz detaljan uvid u pojedinačne merne vrednosti.

## 5. Zaključak

Primena slojevite arhitekture, u kombinaciji sa MVVM obrascem u prezentacionom sloju i repository pattern-om u sloju za pristup podacima, donosi niz značajnih prednosti u razvoju i eksploataciji sistema za nadzor temperature.

Pre svega, ovakav pristup omogućava modularnost sistema, jer su pojedinačni slojevi jasno razdvojeni i imaju precizno definisane odgovornosti. Prezentacioni sloj je zadužen za interakciju sa korisnikom, servisni sloj za obradu podataka i poslovnu logiku, dok sloj za pristup podacima upravlja komunikacijom sa bazom. Ovakva podela olakšava razumevanje sistema i njegovo dalje unapređenje.

Značajna prednost je i skalabilnost sistema, budući da je moguće proširivati funkcionalnosti bez značajnih izmena postojećeg koda. Na primer, dodavanje novih senzora ili izvora podataka može se realizovati unutar servisnog sloja, bez potrebe za izmenama u prezentacionom sloju ili načinu skladištenja podataka.

Primena repository pattern-a omogućava fleksibilnost u radu sa podacima, jer poslovna logika nije direktno vezana za konkretne SQL upite ili tehnologiju baze podataka. Time se omogućava lakša promena ili nadogradnja sistema za skladištenje podataka u budućnosti, uz minimalan uticaj na ostatak aplikacije.

Korišćenje MVVM obrasca doprinosi održivosti i preglednosti korisničkog interfejsa, jer omogućava razdvajanje logike prikaza od samog vizuelnog dela aplikacije. Ovakav pristup olakšava izmene u korisničkom interfejsu bez uticaja na poslovnu logiku sistema, kao i obrnuto.

Dodatno, ovakva arhitektura omogućava paralelan razvoj, gde različiti timovi mogu raditi nezavisno na pojedinačnim slojevima sistema. Na primer, jedan tim može razvijati korisnički interfejs, dok drugi radi na implementaciji servisnog sloja ili optimizaciji pristupa podacima.

Jedna od važnih prednosti je i povećana testabilnost sistema. Jasno razdvojeni slojevi omogućavaju jednostavnije testiranje pojedinačnih komponenti, kao što su servisne metode ili ViewModel logika, bez potrebe za direktnim povezivanjem sa bazom ili korisničkim interfejsom.

Na kraju, ovakav pristup obezbeđuje pouzdanost i stabilnost sistema, što je od posebnog značaja u industrijskim okruženjima. Kontrolisana obrada podataka u servisnom sloju, zajedno sa pouzdanim skladištenjem putem

ADO.NET tehnologije, omogućava kontinuiran rad sistema i sigurno čuvanje mernih vrednosti.

Zahvaljujući ovim karakteristikama, primenjena arhitektura predstavlja čvrstu i fleksibilnu osnovu za dalji razvoj sistema, kao i za njegovu primenu u složenim industrijskim uslovima gde su pouzdanost, brzina obrade i skalabilnost od ključnog značaja.

Mogućnost daljeg unapređenja sistema dodatno potvrđuje kvalitet izabrane arhitekture. Jedan od logičnih sledećih koraka jeste razvoj web aplikacije koja bi omogućila udaljeni pristup sistemu, pregled podataka u realnom vremenu i upravljanje parametrima preko mreže. Zahvaljujući jasno razdvojenim slojevima i primeni MVVM i repository pattern-a, postojeći backend može se relativno lako proširiti, bez značajnih izmena u postojećoj logici sistema.

Takođe, unapređenje bezbednosti može se postići prelaskom sa osnovne (basic) autorizacije na moderniji i sigurniji JWT (JSON Web Token) mehanizam. Time se omogućava skalabilniji i fleksibilniji sistem autentifikacije, posebno u kontekstu web i mobilnih aplikacija, kao i jednostavnije upravljanje korisničkim sesijama i pravima pristupa.

Integracija dodatne opreme, kao što je termovizijska kamera, može se realizovati prvenstveno u servisnom sloju, gde bi se obradila komunikacija sa kamerom, preuzimanje i obrada temperaturnih podataka. Ipak, da bi korisnik mogao da vidi rezultate merenja, neophodno je izvršiti i određena proširenja u prezentacionom sloju, poput dodavanja novih prikaza, tabela, grafikona ili vizuelnih elemenata za prikaz termovizijskih podataka. Prednost slojevite arhitekture je u tome što te izmene ostaju lokalizovane i kontrolisane: servisni sloj se proširuje radi podrške novom uređaju, dok se prezentacioni sloj dopunjuje samo u meri potrebnoj za prikaz novih informacija, bez narušavanja ostatka sistema.

Sve ove mogućnosti unapređenja dodatno potvrđuju da je primenjena arhitektura dugoročno održiva i spremna za proširenje u skladu sa rastućim zahtevima sistema i potrebama korisnika.

## **Zahvalnica**

Ovaj rad je podržalo Ministarstvo nauke, tehnološkog razvoja i inovacija Republike Srbije kroz ugovor o realizaciji i finansiranju naučnoistraživačkog rada NIO u 2026. godini.

## Literatura

- [1] Varduca, Dan F., "Monitoring system for over temperature detection of water cooled rotor poles of hydro electrical generators installed at the Danube River Iron Gates Power Plant", in *Proceedings of the Intersociety Energy Conversion Engineering Conference*, vol. 2, 1996, pp 1459-1464
- [2] S. Milić, A. Žigić, M. Ponjavić, "Online temperature monitoring, fault detection, and a novel heat run test of a water-cooled rotor of a hydrogenerator," *IEEE Transaction on Energy Conversion*, vol. 28, issue 3, pp. 698-706, Sep. 2013. <https://doi.org/10.1109/TEC.2013.2265262>
- [3] S. Milić, N. Miladinović "Daljinski monitoring temperature u elektroenergetskim sistemima", *Međunarodni naučno-stručni simpozijum Informacione tehnologije - Infoteh 2009*, zbornik radova, Vol. 8, Ref. D-11, strane 333-336, Jahorina, Republika Srpska, mart 2009.
- [4] S. Milić, N. Miladinović, A. Žigić, N. Karanović, Z. Kršenković, M. Kožicić, "Merni sistem za praćenje temperatura polova rotora hidrogeneratora na hidroelektrani Đerdap II", *CIGRE - Srbija 29. savetovanje*, zbornik radova R B5 - 08, Zlatibor, Srbija, 31. maj - 06. jun 2009.
- [5] S. Milić, D. Kovačević, N. Karanović, Z. Kršenković, M. Kožicić, "Sistem za temperaturni monitoring namotaja rotora hidrogeneratora", *Elektroprivreda* broj 2, April-Jun 2009.
- [6] S. Milić, "Daljinski temperaturni nadzor polova rotora hidrogeneratora", *Zbornik radova, Institut "Nikola Tesla"*, knjiga 19, strane 271-283, Beograd, 2008-2009. <https://scindeks-clanci.ceon.rs/data/pdf/0350-8528/2009/0350-85280919271M.pdf>
- [7] S. Milić, M. Srećković, N. Karanović, Z. Kršenković, J. Marković-Petrović, "On-line Remote Monitoring of Generators and Transformers in Energy Power Systems ", *Powerplants 2010, Elektrane 2010, međunarodna konferencija*, Vrnjačka Banja, Srbija, 26-29. oktobar 2010.
- [8] R. C. Martin, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Boston, MA: Prentice Hall, 2017.
- [9] M. Fowler, *Patterns of Enterprise Application Architecture*. Boston, MA, USA: Addison-Wesley, 2002.
- [10] M. Microsoft, "ADO.NET Overview," *Microsoft Learn*. <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/>
- [11] A. Freeman, *Pro .NET Data Access with ADO.NET*. New York, NY: Apress, 2019. pp. 155-167.



- [12] J. Smith, "WPF Apps with the Model-View-ViewModel Design Pattern," *MSDN Magazine*, Feb. 2009. [Online]. Available: <https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern>
- [13] Microsoft, "Model-View-ViewModel (MVVM)," *Microsoft Learn*. <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>
- [14] <https://optris.com/support-service/videos/product-videos/ct-4ml-high-speed-pyrometer-for-fast-moving-objects/>

**Abstract.** The paper presents the architecture of a software system for monitoring rotor pole temperature in the Đerdap 2 hydropower plant. The system is implemented using the .NET 10 platform and is based on a layered architecture with the application of the repository pattern, enabling clear separation of responsibilities between system components. Data collected from sensors are stored in an SQL database using ADO.NET technology. Special emphasis is placed on security, scalability, and the potential for further system development in real industrial environments. The contribution of this work lies in defining a customized layered architecture for an industrial monitoring system, which enables efficient processing of a large amount of measurement data in real time, along with the application of the MVVM pattern for organizing the user interface and improving the separation of the presentation layer.

**Keywords:** Layered architecture, MVVM, repository pattern, ADO.NET, SQL, temperature monitoring, rotor poles, hydrogenerator.

## Design and Implementation of the Software Architecture for a Rotor Pole Temperature Monitoring System at the Đerdap 2 Hydroelectric Power Plant

Blagoje Babić<sup>id</sup>, Saša Milić<sup>id</sup>, Miša Kožicić, Nikola Miladinović<sup>id</sup>

Rad primljen u uredništvo: 06.05.2026. godine.

Rad prihvaćen: 27.06.2026. godine.