Sustav za nadzor i upravljanje PLC uređaja Siemens S7-1200

Filjak, Mario

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:225:891566

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

ALGEBRA VISOKO UČILIŠTE

Repository / Repozitorij:

Algebra Univerity College - Repository of Algebra Univerity College

zir.nsk.hr

dabar DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

Sustav za nadzor i upravljanje PLC uređaja Siemens S7-1200

Mario Filjak

Zagreb, listopad 2017.

Student vlastoručno potpisuje Završni rad na prvoj stranici ispred Predgovora s datumom i oznakom mjesta završetka rada te naznakom:

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada“.

U Zagrebu, datum.

Predgovor

S ovim radom ujedno je i stigao kraj jednog lijepog perioda mog života obilježenog studijem. Koristim priliku kako bih zahvalio osobama koje su mi pružile podršku i omogućile da proživim taj period.

Zahvaljujem mentoru dr. sc. Goranu Đambiću što je mentorirao ovaj rad te što je svojom angažiranošću i korektnim pristupom usmjeravao nas studente u području računarstva.

Mojim roditeljima, htio bih zahvaliti na svemu što su činili i još uvijek čine za mene. Bez njihove podrške, ništa od ovoga ne bi bilo moguće.

Ujaku Frani Juriću, želim zahvaliti na podršci koju mi je pružio tijekom ove tri godine studija te što je bio ujak tokom cijelog života.

Bratu Marku zahvaljujem na pomoći prilikom izrade ovog rada te na ulozi starijeg brata koju je uvijek nosio kada je trebalo.

I na kraju zahvaljujem svojoj Ivani što je učinila ovaj lijep period života još ljepšim. Bila je moja inspiracija i najveća podrška tijekom studija. I još uvijek je.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

Sadržaj rada prikazuje praktični razvoj sustava za nadzor i upravljanje PLC uređaja Siemens S7-1200. Svrha praktičnog rada je testiranje mogućnosti implementacije sustava. Razvoj prototipa je implementiran simulacijom modela pivovare koja koristi PLC i periferne uređaje u svrhu simulacije proizvodnog procesa. Poslovna domena sustava se sastoji od izravnog nadzora uređaja putem *desktop* aplikacije i administrativne web aplikacije čija je svrha pregled izvještaja o proizvodnji te slanje obavijesti u slučaju identifikacije incidenta. Praktični dio rada je nadopunjen teorijskim cjelinama koje objašnjavaju PLC uređaj, postojeće sustave nadzora, Snap7 API za komunikaciju s uređajem te oblikovne obrasce i tehnologije za razvoj web i *desktop* aplikacija u .NET okruženju.

Ključne riječi: PLC, Siemens S7-1200, Snap7, ASP.NET MVC, WPF

Summary

The Control and monitoring system of Siemens S7-1200 was developed as prototype which purpose is the proof of concept. Prototype design model represents simulation of brewery. The Brewery model was built using PLC with peripheral devices that simulates manufacturing process. The business domain consists of two parts. First part represents on site device control implemented by desktop application and the second part represents administrative web application. The purpose of web application is production report and incident notification. The prototype build is backed up with theoretical research which involves PLC device, existing solutions of PLC control, Snap7 API and design patterns used for web and desktop development in .NET framework.

Key words: PLC, Siemens S7-1200, Snap7, ASP.NET MVC, WPF

Sadržaj

1. Uvod 1

2. Poslovna prilika 2

2.1. SCADA sustavi 2

2.2. Prednosti i nedostaci rješenja 3

3. PLC Siemens S7-1200 5

3.1. Rad PLC uređaja 5

3.1.1. Ciklus rada PLC uređaja 6

3.2. Arhitektura i komponente 6

3.2.1. Postolje i utor 7

3.2.2. Digitalni i analogni priključci 8

3.3. Struktura memorije 9

3.3.1. Organizacijski blokovi 9

3.3.2. Funkcijski blokovi 9

3.3.3. Funkcije 9

3.3.4. Podatkovni blokovi 10

3.4. Programiranje 10

3.4.1. Ladder dijagram 10

3.4.2. Dijagram funkcijskih blokova 11

3.4.3. SCL jezik 11

3.4.4. Optimizirani i neoptimizirani podatkovni blokovi 12

3.5. Komunikacija 13

4. Snap7 14

4.1. Primjena 14

4.2. S7 protokol 14
4.3. S7 Arhitektura (Klijent, Poslužitelj, Partner) 15
4.3.1. Arhitektura Klijent-Poslužitelj 15
4.3.2. Partner 16
4.4. Snap7 implementacija arhitekture 16
4.5. Struktura podatkovnih tipova 17
4.6. Snap7 API konvencija 18
4.6.1. Konvencija naziva funkcija 19
4.7. Snap7 API komponente korištene u radu 19
5. Poslovni model sustava 20
5.1. Desktop aplikacija PS Operator 20
5.1.1. Autentikacija 20
5.1.2. Glavni prozor 20
5.1.3. Prikaz proizvodnih procesa 21
5.1.4. Prikaz trenutne temperature 22
5.1.5. Prikaz podataka o uređaju 22
5.1.6. Prikaz korisničkog računa i odjava 23
5.1.7. Postavke 24
5.2. Web aplikacija PS Cloud 24
5.2.1. Upravljačka ploča 25
5.2.2. Poruke 25
5.2.3. Izvještaj o proizvodnji 26
5.2.4. Izvještaji o incidentima 27
5.2.5. PLC uređaji 28
5.2.6. Postavke 29
5.2.7. Korisničke role 30

5.2.8. Servis za obavještavanje incidenata 30
5.3. Prednosti poslovnog modela 31
6. Model simulacije pivovare 32
6.1. Popis funkcionalnosti 32
6.2. Kontrola toka 34
6.2.1. Postupak autentikacije 34
6.2.2. Odjava korisnika 35
6.2.3. Pokretanje proizvodne linije 35
6.2.4. Zaustavljanje proizvodne linije 36
6.2.5. Generiranje izvještaja o proizvodnji 36
6.2.6. Identificiranje incidenata 36
6.3. Hardver komponente 36
6.3.1. SIMATIC S7-1200 adapter za PT100 temperaturnu sondu 36
6.3.2. Temperaturna sonda PT100 37
6.3.3. Ultrazvučni senzor 37
6.3.4. Optički senzor 37
6.3.5. Induktivni senzor 37
7. Desktop aplikacija 38
7.1. Primjena 38
7.2. WPF 38
7.3. MVVM oblikovni obrazac 38
7.4. Struktura koda 39
7.4.1. PSOperator.MVVM 40
7.4.2. PSOperator.PLC 40
7.4.3. PSOperator.Network 40
7.4.4. PSOperator.Common 40

8. Web aplikacija 41
8.1. Model troslojne arhitekture 41
8.2. ASP.NET MVC 42
8.3. REST protokol i Web API 2 42
8.4. Arhitektura web aplikacije 43
8.5. Servis za primanje izvještaja 43
8.6. Baza podataka 44
Zaključak 45
Popis kratica 46
Popis slika 47
Popis tablica 49
Popis kôdova 50
Literatura 51

1. Uvod

Završni rad obrađuje temu razvoja sustava nadzora i upravljanja industrijskim procesima upravljanim Siemensovim S7-1200 PLC (programibilni logički upravljač) uređajem. Sustav je osmišljen kao alternativa skupim i nefleksibilnim postojećim sustavima. Poslovna domena sustava se može podijeliti na dva segmenta od kojih je jedan vezan uz izravno upravljanje uređajem, dok je drugi orijentiran na administrativni dio prikupljanja podataka i generiranja izvještaja.

Segment upravljanja uređajem se odnosi na vizualizaciju stanja proizvodnog procesa te korisničko sučelje za upravljanje. Spomenuti segment omogućuje pravovremeni uvid u stanje procesa te je namijenjen operaterima u proizvodnji. Drugi administrativni segment je namijenjen menadžerima, tehnološkom odjelu i odjelu održavanja. Sustav omogućuje spomenutim poslovnim funkcijama pregled izvještaja te obavještavanje o zabilježenim incidentima u proizvodnom procesu.

Rad sustava je testiran na modelu simulacije pivovare. Virtualna pivovara se sastoji od spremnika tekućine čija su temperatura i razina tekućine promatrane vrijednosti, pumpe koja određuje razinu tekućine u spremniku i brojača proizvedenih boca. *Desktop* aplikacija (*desktop* – radna površina, aplikacija koja se izvodi na radnoj površini operacijskog sustava) predstavlja korisničko sučelje tj. segment upravljanja uređajem, dok je administrativni segment implementiran u obliku web aplikacije.

Sadržaj rada je semantički podijeljen u tri dijela. Prvi dio objašnjava poslovnu priliku kao motivaciju za razvoj sustava. Drugi dio se odnosi na tehničku domenu koja objašnjava PLC uređaj i rješenje otvorenog koda koje omogućuje tehničku izvedbu rada. Treći dio objašnjava poslovnu domenu i praktičnu izvedbu idejnog rješenja.

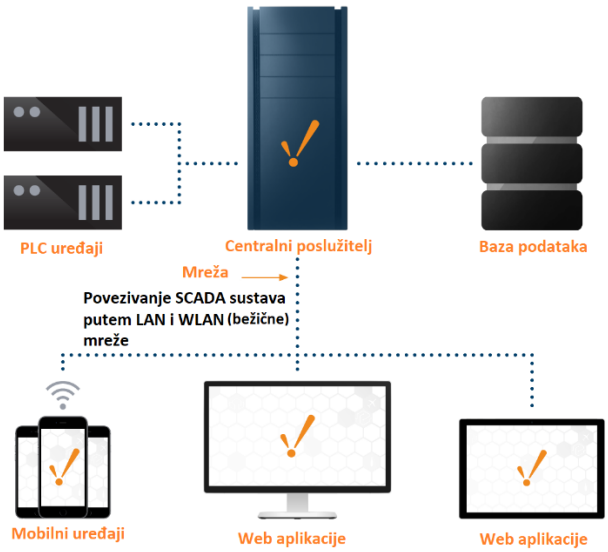
2. Poslovna prilika

Cilj završnog rada je testirati mogućnost implementacije sustava za nadzor i upravljanje industrijskih postrojenja automatiziranih i upravljanih Siemensovim S7-1200 PLC[1] uređajem. Nadzor i upravljanje se izvršava korištenjem PC (osobno računalo) računala povezanog s PLC uređajem putem LAN (lokalna računalna mreža) mreže. Poslovna primjena sustava:

- Nadzor stanja proizvodnih procesa
- Prikupljanje podataka iz proizvodnih procesa
- Kreiranje izvještaja o rezultatima proizvodnih procesa
- Obavještavanje nadležnih osoba o incidentima

2.1. SCADA sustavi

Postojeći sustavi koji implementiraju definiranu poslovnu primjenu se nazivaju SCADA (nadzor upravljanja i prikupljanje podataka) sustavi. Slika 2.1 prikazuje model SCADA sustava koji se sastoji od PLC uređaja, centralnog poslužitelja i baze podataka. Klijenti pristupaju poslužiteljskoj aplikaciji putem LAN mreže.



Slika 2.1 Model SCADA sustava[2]

[1] Siemens S7-1200 PLC je programibilni logički upravljač proizvođača Siemens. S7-1200 označava tip procesora. PLC je objašnjen u poglavlju 3. PLC Siemens S7-1200 (S7-1200 *system manual*, *Preface*, str. 3)
[2] Izvor: https://inductiveautomation.com/static/images/ignition-scada-architecture.png, listopad 2017.

SCADA sustav se sastoji od PLC uređaja spojenih na HMI (sučelje za interakciju korisnika i uređaja) panele i računala te centralnog poslužitelja sa bazom podataka i poslužiteljskom aplikacijom.[3] HMI paneli su uređaji koji implementiraju korisničko sučelje i omogućuju interakciju operatera s PLC uređajem.[4] Slika 2.2 prikazuje primjere Siemensovih HMI panela. U testnom sustavu PC računalo se koristiti kao HMI uređaj i kao poslužitelj centralne aplikacije.

Slika 2.2 Primjer Siemensovog HMI panela[5]

2.2. Prednosti i nedostaci rješenja

S obzirom da sustav koji ispunjava zahtjeve spomenute poslovne primjene već postoji, postavlja se pitanje zašto uopće razvijati novi sustav. U nastavku su navedene prednosti i nedostaci alternativnog rješenja.

Prednosti:

- Rješenje otvorenog koda
- Nisu potrebna specifična znanja o industrijskoj automatizaciji
- Korišteni PC moguće je prenamijeniti za razliku od HMI panela

[3] Izvor: https://inductiveautomation.com/what-is-scada, listopad 2017
[4] Izvor: http://w3.siemens.com/mcms/human-machine-interface/en/operator-devices/basichmi/basic-panels/Pages/Default.aspx, listopad 2017.
[5] Izvor: http://w3.siemens.com/mcms/human-machine-interface/de/bediengeraete/basic-hmi/basic-panels/tabcardseiten/PublishingImages/basic-panels-2nd-gen-all-800.jpg, listopad 2017.

- Veća sloboda pri kreiranju vizualnog rješenja

Nedostaci:

- Rješenje nije testirano te su upitne performanse u zahtjevnom okruženju
- Nedostatak tehničke podrške i jamstva biblioteke otvorenog koda

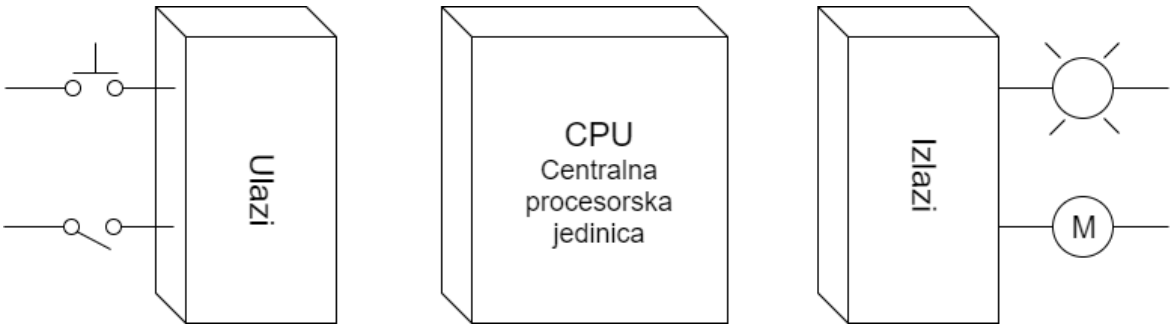
Zatvorenost i ograničenja sustava od strane proizvođača se žele izbjeći korištenjem rješenja otvorenog koda i PC računala. Ovo je i ujedno razlog razvoja sustava koji je tema rada.

3. PLC Siemens S7-1200

PLC je programibilni uređaj čija je namjena upravljanje industrijskim procesima. Rad uređaja se zasniva na praćenju stanja perifernih uređaja spojenih na ulaze, odnosno izlaze te izvršavanju programa koji donosi odluke na temelju ulaznih vrijednosti. S7-1200 označava tip procesora PLC uređaja proizvođača Siemens (S7-1200 *system manual*, *Introducing the* S7-1200 PLC, str. 19) koji je i korišten za tehničku izvedbu simulacije pivovare (6. model simulacije pivovare).

3.1. Rad PLC uređaja

Rad PLC uređaja se zasniva na programu koji donosi odluke na temelju ulaznih vrijednosti. Slika 3.1 prikazuje tri glavne komponente od kojih se sastoji svaki PLC uređaj, a to su ulazi, CPU (centralna procesorska jedinica) i izlazi. CPU predstavlja procesor koji izvršava instrukcije programa koji se nalaze unutar memorije. Ulazi su priključci za periferne uređaje kao što su npr. senzori, tipkala, prekidači i ostale industrijske komponente. Spomenuto praćenje stanja ulaznih vrijednosti se odnosi na čitanje vrijednosti stanja uređaja spojenih na ulaze i njihovo spremanje u memoriju. Izlazi su priključci za periferne uređaje čiji su rad i aktivnost upravljani programom PLC uređaja. Izlazni uređaji mogu biti sirene, elektromotori, pneumatski uređaji i slične komponente koje se koriste u industriji .[6]



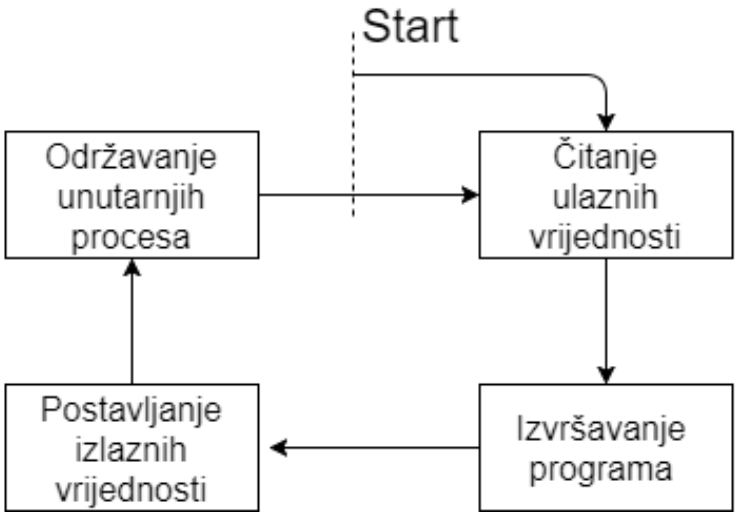
Slika 3.1 Osnovne komponente PLC uređaja[7]

[6] Izvor: https://www.amci.com/industrial-automation-resources/plc-automationtutorials/what-plc, rujan 2017.
[7] Izvor: https://www.amci.com/index.php/download_file/view_inline/2422, rujan 2017.

3.1.1. Ciklus rada PLC uređaja

Proces izvršavanja se odvija kroz četiri faze koje se kontinuirano izvršavaju slijedom prikazanim na slici 3.2. U nastavku su redom navedene i objašnjene faze izvršavanja.[8]

- Čitanje ulaznih vrijednosti – Čitanje stanja ulaznih perifernih uređaja i spremanje stanja u memoriju.
- Izvršavanje programa – Izvršavanje programske logike korisnički definiranog programa.
- Postavljanje izlaznih vrijednosti – Reguliranjem vrijednosti napona na izlaznim priključcima upravlja se stanjem spojenih uređaja.
- Održavanje unutarnjih procesa – Unutarnji procesi predstavljaju procese operacijskog sustava kao što su dijagnostika, upravljanje vremenom i slično.

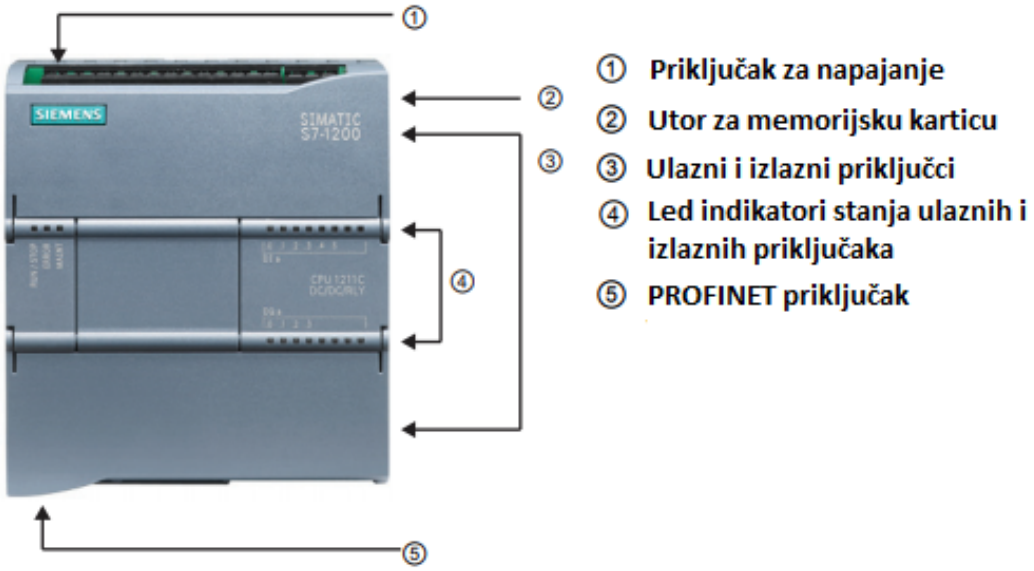


Slika 3.2 Ciklus rada PLC uređaja[9]

3.2. Arhitektura i komponente

Uz tri glavna elementa navedena u poglavlju 3.1., arhitektura PLC uređaja varira ovisno o tipu CPU-a. Slika 3.4 prikazuje komponente S7-1200 uređaja. Primjer arhitekture uređaja objašnjen je na primjeru Siemens S7-1200, model CPU 1212C. Spomenuti uređaj je korišten u praktičnoj izvedbi ovoga rada. Tablica 3.1 prikazuje izdvojene tehničke karakteristike CPU 1212C uređaja.

[8] Izvor: https://www.amci.com/industrial-automation-resources/plc-automationtutorials/what-plc, rujan 2017.
[9] Izvor: https://www.amci.com/index.php/download_file/view_inline/2423, rujan 2017.



Slika 3.3 Uređaj i periferni priključci[10]

Tablica 3.1 Izdvojene tehničke karakteristike CPU 1212C[11]

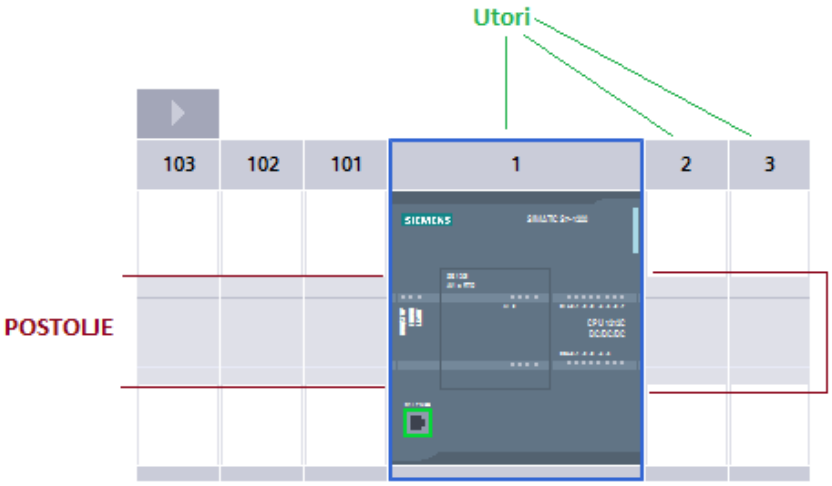
Dimenzije (mm)	90 x 100 x 75
Radna memorija	50 kB
Podatkovna memorija	1 MB
Trajna memorija	10 kB
Digitalni ulazi	8
Digitalni izlazi	6
Analogni ulazi	2
Broj *ethernet* priključaka	1

3.2.1. Postolje i utor

Model CPU 1212C ima osam digitalnih i dva analogna ulaza te šest digitalnih izlaza. Međutim, nekada ovaj broj izlaza i ulaza nije dovaljan za potrebe industrijskog procesa.

[10] Izvor: S7-1200 *system manual*, str. 19
[11] Izvor: S7-1200 *system manual*, str. 20, tablica 1-1 usporedba CPU modela

Izlazi i ulazi se nalaze na komponenti koja se naziva utor (engl. *slot*). Mogućnost dodavanja novih utora omogućuje komponenta postolje (engl. *rack*). Postolje nije dio samog PLC uređaja već predstavlja komponentu na koju se može povezati više uređaja i utora (S7-1200 *system manual*, 10.2.3.2 *Configuring the devices*, str. 461). Komponente postolje i utor su važne kako bi se u *desktop* aplikaciji moglo definirati iz kojeg se djela memorije čitaju podaci. Slika 3.3 ilustrira odnos postolja i utora. Postavljanje postavki *desktop* aplikacije opisano je u poglavlju 5.1.7. Postavke.



Slika 3.4 Postolje i utor (snimka zaslona iz programskog okruženja TIA portal[12])

3.2.2. Digitalni i analogni priključci

Osnovna razlika između analognog i digitalnog signala u primjeni je periodičnost mjerenja te prikaz vrijednosti. Analogni signali neprekidno prikazuju rezultate mjerenja u obliku brojčanih vrijednosti. Mjerenje temperature primjer je analognog mjerenja s obzirom na neprekidno očitavanje temperature. Konkretan primjer ove vrste mjerenja bila bi živa u termometru ili otpor PT100 temperaturne sonde[13] u kontekstu modela simulacije pivovare. Digitalni signal predstavlja stanje prikazano nizom nula i jedinica. Jedinica i nula su ekvivalent stanjima uključeno i isključeno (Goldberg; Ng, 2010). Primjer mjerenja digitalnim signalom je induktivni senzor[14] čija je namjena detekcija prisutnosti zatvarača

[12] TIA (potpuno integrirana automatizacija) portal – programsko okruženje za programiranje Siemensovih PLC uređaja, https://www.siemens.com/global/en/home/products/automation/industry-software/automation-software/tia-portal.html#, listopad 2017.
[13] Temperaturna sonda korištena u modelu simulacije pivovare. Više u 6.3.2. Temperaturna sonda PT100
[14] Induktivni senzor detektira prisutnost metala. Više u 6.3.5. Induktivni senzor

boce u modelu simulacije pivovare. Stanje jedan signalizira prisutnost zatvarača, dok se izostanak zatvarača signalizira nulom.

3.3. Struktura memorije

Memoriju unutar PLC uređaja možemo promatrati sa dva različita stajališta: prema blokovima namijenjenima za izvršavanje programa te prema namjeni. Blokovi memorije se dijele na organizacijske, funkcijske, podatkovne blokove te funkcije (*Programming Guideline for* S7-1200/1500, 3. *General Programming*, str. 27). Prema namjeni, memorija se dijeli na radnu, podatkovnu i trajnu memoriju. Namjena podatkovne memorije jednaka je namjeni RAM (radna memorija) memorije u PC računalima. Podatkovna memorija služi za pohranu podataka samo tijekom rada uređaja, za razliku od trajne memorije koja čuva vrijednosti i nakon prestanka napajanja (S7-1200 *system manual*, 4.1.6 CPU *memmory*, str. 82).

3.3.1. Organizacijski blokovi

Organizacijski blokovi (engl. *organization block*) predstavljaju sučelje između korisničkog programa i operacijskog sustava. OB (organizacijski blok) blokovi sadrže FB (funkcijske blokove) blokove i FC (funkcije) blokove te određuju tijek izvršavanja funkcija i instrukcija unutar programa (*Programming Guideline for* S7-1200/1500, 3.2.1 *Organization blocks* (OB), str. 37). Operacijski sustav izvršava operacijske blokove u drugoj fazi ciklusa rada (3.1.1. Ciklus rada PLC uređaja).

3.3.2. Funkcijski blokovi

Funkcijski blokovi (engl. *function block*) predstavljaju dijelove programa koji sadrže FB blokove, FC blokove i reference na varijable koje predstavljaju ulazne, odnosno izlazne parametre funkcija. Varijable se nalaze u DB (podatkovnim blokovima) blokovima koje referencira funkcijski blok (*Programming Guideline for* S7-1200/1500, 3.2.3 *Function blocks* (FB), str. 42).

3.3.3. Funkcije

FC blok predstavlja funkcije (engl. *function*) koje nemaju dodijeljen trajni memorijski prostor za varijable. Sve varijable unutar funkcije se brišu iz privremene memorije nakon što

se funkcija prestane izvršavati (*Programming Guideline for* S7-1200/1500, 3.2.2 *Functions* (FC), str. 40).

3.3.4. Podatkovni blokovi

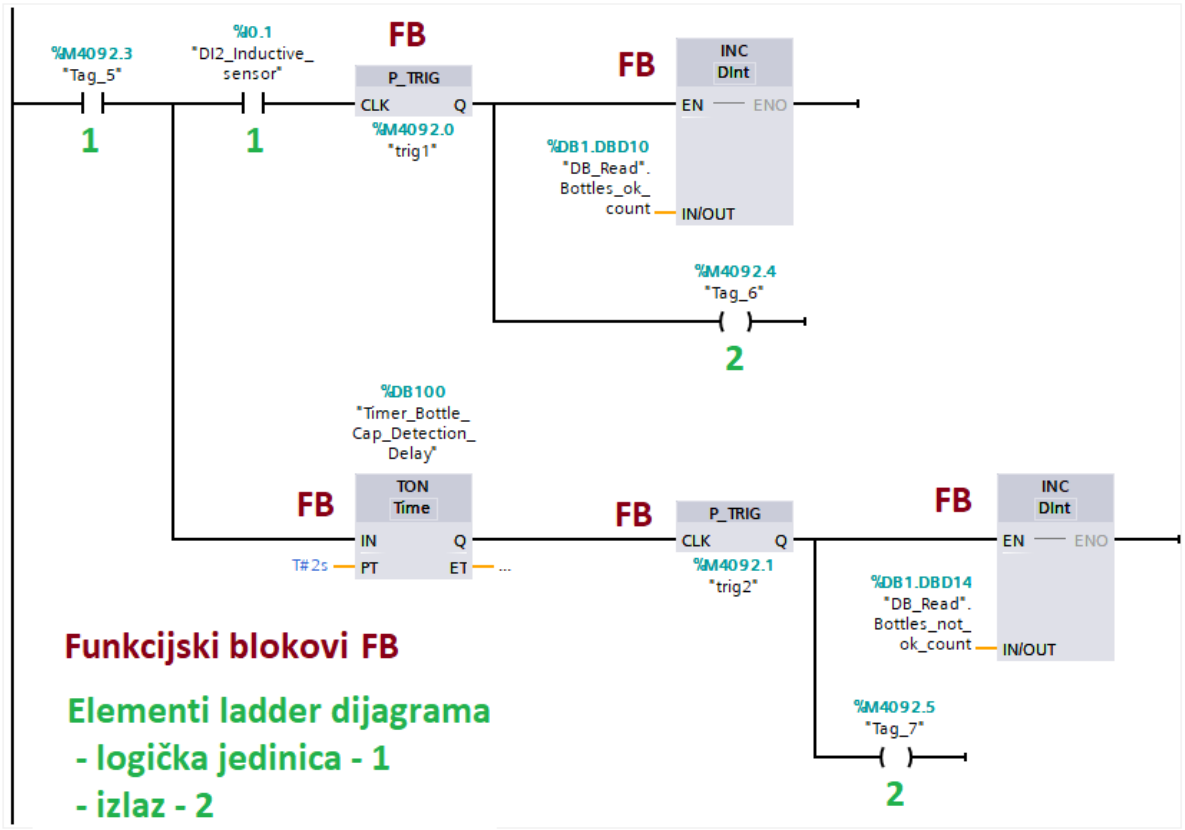
DB blokovi su dijelovi memorije koji pohranjuju varijable potrebne za izvršavanje programa. Postoje globalni podatkovni blokovi (engl. *data block*) čijim varijablama imaju pristup svi blokovi te instancirani blokovi koji mogu biti dodijeljeni određenim funkcijama i funkcijskim blokovima. (*Programming Guideline for* S7-1200/1500, 3.2.6 *Global data blocks* (DB), str. 45).

3.4. Programiranje

Uređaje tipa S7-1200 je moguće programirati u tri različita programska jezika: *Ladder* dijagram (engl. *ladder diagram*), Dijagram funkcijskih blokova (engl. *function block diagram*) i SCL (strukturirani upravljački jezik, engl. *structured programming language*). Tri navedena jezika se međusobno kombiniraju. *Ladder* dijagram određuje tok izvršavanja funkcijskih blokova. Veza između funkcijskih blokova je definirana dijagramom funkcijskih blokova, dok je logika unutar funkcijskih blokova definirana SCL jezikom (*Programming Guideline for* S7-1200/1500, 2.3 *Programming languages*, str. 8). Okruženje korišteno za programiranje PLC uređaja u praktičnom dijelu ovog rada je TIA portal.

3.4.1. *Ladder* dijagram

Ladder dijagram je programski jezik sastavljen od dijagrama koji predstavljaju relejnu logiku tj. povezanost između sklopki i potrošača. Sklopke mogu biti u stanju 0 ili 1, što određuje stanje zatvorenosti strujnog kruga. Dijagramom se ostvaruju logičke funkcije iz *booleove* algebre (Bolton, 2009). U kontekstu PLC uređaja, sklopke su bitovi u memoriji koji predstavljaju stanje otvoren odnosno zatvoren, dok su potrošači izlazni priključci uređaja. Slika 3.5 prikazuje primjer *Ladder* dijagrama.



Slika 3.5 Primjer funkcije koja koristi kombinaciju *Ladder* dijagrama i funkcijskih blokova[15]

3.4.2. Dijagram funkcijskih blokova

Dijagram funkcijskih blokova je programski jezik koji u obliku dijagrama prikazuje međusobnu povezanost funkcijskih blokova. Svaki funkcijski blok predstavlja funkciju i veze sa ostalim funkcijama. Navedene veze predstavljaju ulazne i izlazne parametre funkcija (Bolton, 2009).

3.4.3. SCL jezik

SCL je proceduralni programski jezik baziran na programskom jeziku PASCAL, prilagođen programiranju PLC uređaja. Ovaj programski jezik se koristi za definiranje programske logike unutar funkcijskih blokova i funkcija (*Structured Control Language* (SCL) *for* S7-300/S7-400 *Programming*). Primjer kôda 3.1 prikazuje funkciju za skaliranje ultrazvučnog senzora koristeći jednadžbu pravca kroz dvije točke (1) napisane u SCL jeziku. Varijabla y iz formule predstavlja skaliranu vrijednost, dok varijabla x predstavlja izmjerenu vrijednost.

[15] Slika je nastala kao snimka zaslona iz programskog okruženja TIA portal.

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} \cdot x - x_1 \quad (1)$$

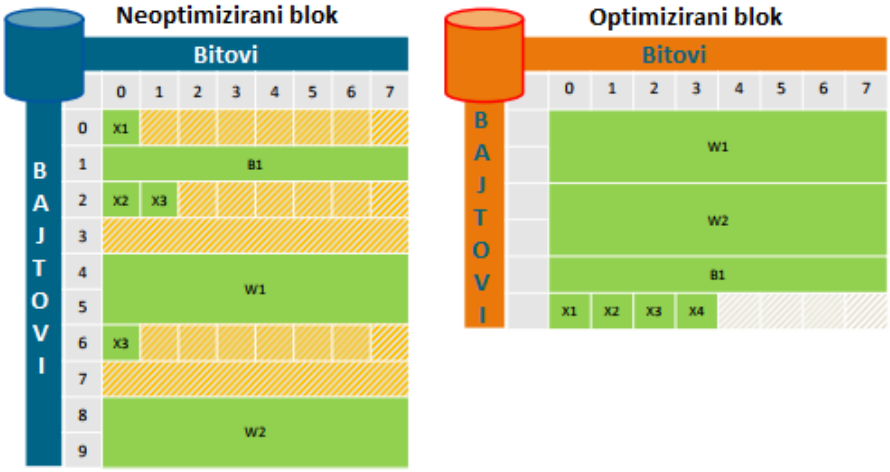
```
#Input_Real := INT_TO_REAL(#Input_int);
#x_min_Real := INT_TO_REAL(#x_min_int);
#x_max_Real := INT_TO_REAL(#x_max_int);

#Out_Real := ((#y_max_Real - #y_min_Real) / (#x_max_Real -
#x_min_Real)) * (#Input_Real - #x_min_Real) + #offset_Real +
#y_min_Real;
```

Kôd 3.1 Skaliranje vrijednosti ultrazvučnog senzora jednadžbom pravca kroz dvije točke

3.4.4. Optimizirani i neoptimizirani podatkovni blokovi

Memorija uređaja je podijeljena na nizove bajtova. Najmanji podatkovni tip u PLC uređaju je bit koji ima namjenu kao i podatkovni tip *bool* iz C# programskog jezika (više o podatkovnim tipovima u poglavlju 4.5. Struktura podatkovnih tipova). Za pohranu varijable veličine n bitova, gdje je n <= 8, potrebno je iskoristiti cijeli bajt u memoriji. S7-1200 ima opciju korištenja optimiziranih podatkovnih blokova, što znači da se varijable istog tipa grupiraju u memoriji kako bi se izgubilo što manje prostora (*Programming Guideline for* S7-1200/1500, 2.6 *Optimized blocks*, str. 10). Slika 3.6 ilustrira problematiku optimiziranih i neoptimiziranih blokova. Za isti broj varijabli neoptimizirani blok iskoristi 10 bajtova, dok optimizirani blok iskoristi 6 bajtova. Međutim, Snap7 API[16] (aplikacijsko programsko sučelje) za komunikaciju računala i PLC uređaja, na kojem se bazira sustav koji je ujedno i praktična izvedba rada, ne može upravljati s uređajima koji koriste optimizirane blokove (Nardella, 2015). Ovo je jedan od nedostataka sustava koji je tematika rada. Programer PLC uređaja je primoran koristiti neoptimizirane blokove što može imati nepovoljan utjecaj na automatizirani sustav.

[16] Snap7 je biblioteka otvorenog koda namijenjena za komunikaciju računala i PLC uređaja. Više u poglavlju 4. Snap7.



Slika 3.6 Optimizirani i neoptimizirani podatkovni blokovi[17]

3.5. Komunikacija

Standard za komunikaciju industrijskih uređaja naziva se *Profinet* koji se zasniva na TCP/IP[18] (protokol kontrole prijenosa podataka) protokolu isto kao i *ethernet*[19] s kojim je kompatibilan i može komunicirati. Razlika između *Profineta* i *etherneta* je u brzini razmjene podataka i stabilnosti. Za razliku od *etherneta*, *Profinet* može ostvariti komunikaciju bez odgode u stvarnom vremenu.[20]

[17] Izvor: *Programming Guideline for* S7-1200/1500, slika 2-5, stranica 10
[18] TCP/IP je protokol za prijenos podataka koji enkapsulira aplikacijski, prezentacijski, sesijski, prijenosni, mrežni, podatkovni i fizički sloj, https://technet.microsoft.com/en-us/library/cc958821.aspx, listopad 2017.
[19] *Ethenret* je vrsta protokola korištenih u LAN mrežnim tehnologijama, http://searchnetworking.techtarget.com/definition/Ethernet, rujan 2017.
[20] http://us.profinet.com/technology/profinet, rujan 2017.

4. Snap7

Snap7 je biblioteka otvorenog koda čija je namjena ostvarivanje komunikacije sa Siemensovim PLC uređajima serije S7. Objavljena je i licencirana od strane GNU projekta pod licencom naziva *Lesser General Public Licence version 3.0* (Lesser javna licenca verzija 3.0). Licenca omogućuje besplatno korištenje biblioteke u komercijalne svrhe, ali bez promjene izvornog kôda.[21] GNU je skupina biblioteka koda korištenih za Linux operacijske sustave napisana u C programskom jeziku.[22] Snap7 je kompatibilan sa C++ programskim jezikom te nudi mnoge omotače (engl. *wrappers*) za više programskih jezika kao što su C#, Java i Python (Nardella, 2015).

4.1. Primjena

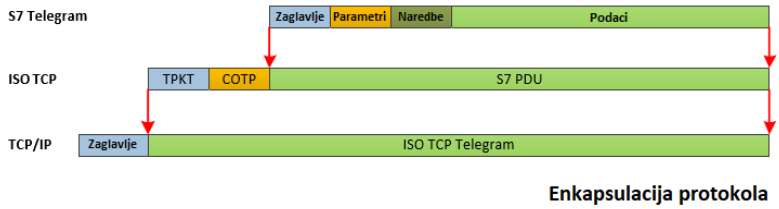
Snap7 omogućuje komunikaciju PLC uređaja i računala putem *ethernet* protokola. Uobičajeni protokol prijenosa podataka u industriji je *Profibus*, međutim *ethernet* ima određenih prednosti kao što su veća brzina prijenosa podataka, laka detekcija grešaka slanjem *ping* poruke te jeftinija i jednostavnija izvedba. Snap7 API omogućuje funkcionalnosti čitanja i pisanja u memoriju uređaja tj. u DB blokove, čitanje informacija iz operacijskog sustava uređaja te funkcionalnosti izmjene OB, FC i FB blokova. Što se tiče kompatibilnosti, API je kompatibilan sa više operacijskih sustava među kojima su najznačajniji Windows i Linux operacijski sustavi (Nardella, 2015).

4.2. S7 protokol

S7 protokol se bazira na TCP/IP protokolu prijenosa blokova podataka. Blokovi podataka koji se prenose označeni su mjernom jedinicom PDU (jedinica prijenosa podatkovnog protokola, engl. *protocol data unit*) koja označava maksimalnu duljinu bloka bajtova. S7 protokol je funkcijski orijentiran protokol, što znači da se razmjena podataka zasniva na određenoj naredbi ili odgovoru na naredbu. Svaka naredba se sastoji od zaglavlja, seta naredbi, parametara i podataka. Zaglavlje i setovi naredbi se uvijek koriste dok su ostala dva elementa opcionalna (Nardella, 2015). Slika 4.1 prikazuje međusobnu enkapsulaciju

[21] Izvor: https://www.gnu.org/licenses/lgpl-3.0.en.html, rujan 2017.
[22] Izvor: https://www.gnu.org/software/libc, rujan 2017.

podataka između protokola S7, ISO TCP (industrijski protokol kontrole prijenosa podataka) i TCP/IP. Naredbe i klase koje ih implementiraju su opisane u poglavlju 4.6. Snap7 API konvencija.



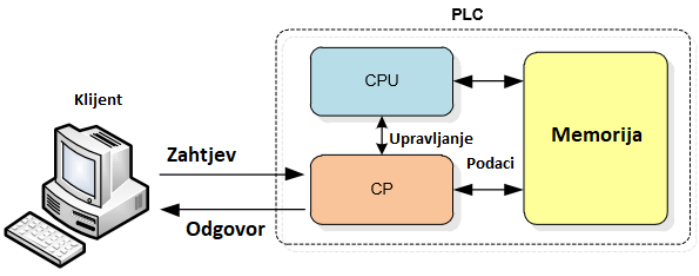
Slika 4.1 Enkapsulacija protokola S7, ISO-TCP, TCP/IP[23]

4.3. S7 Arhitektura (Klijent, Poslužitelj, Partner)

Jedna od namjena i prednosti PLC uređaja je mogućnost međusobne komunikacije sa drugim PLC uređajima. Siemens definira arhitekturu određenu elementima klijent (engl. *client*), poslužitelj (engl. *server*) i partner. Klijent je element sustava koji šalje upite, odnosno zahtjeve, dok je poslužitelj element koji obrađuje zahtjev i šalje odgovor. Partneri su elementi sustava koji mogu međusobno komunicirati, odnosno razmjenjivati poruke (Nardella, 2015).

4.3.1. Arhitektura Klijent-Poslužitelj

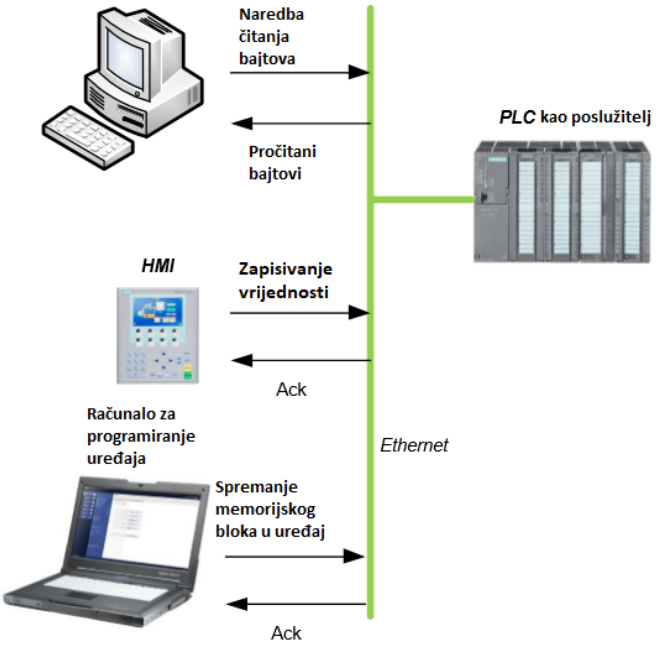
Slika 2.1 prikazuje vezu klijenta i komunikacijskog procesora PLC uređaja u ulozi poslužitelja. CP (komunikacijski procesor) prosljeđuje zahtjeve centralnoj procesorskoj jedinici te preuzima od nje odgovor koji prosljeđuje klijentu.



Slika 4.2 Veza računalo klijent i PLC uređaja[24]

[23] Izvor: Nardella D.Snap7 *Reference manual*, str. 18
[24] Izvor: Nardella D.Snap7 *Reference manual*, str. 19

Slika 4.3 prikazuje arhitekturu klijent-poslužitelj gdje je PLC uređaj poslužitelj, dok su računala i HMI uređaj klijenti (Nardella, 2015). Sustav simulacije pivovare implementira arhitekturu klijent-poslužitelj u kojoj PC računalo zamjenjuje HMI panel.



Slika 4.3 Arhitektura klijent – poslužitelj[25]

4.3.2. Partner

Partneri su elementi sustava koji predstavljaju pristupne točke koje međusobno komuniciraju. Razlikujemo aktivnog i pasivnog partnera. Aktivni partner je onaj koji šalje zahtjev, dok je pasivni onaj koji prima zahtjev. Ovakva arhitektura se naziva klijent-klijent arhitekturom. Namjena ovakve arhitekture je međusobno povezivanje više PLC uređaja (Nardella, 2015).

4.4. Snap7 implementacija arhitekture

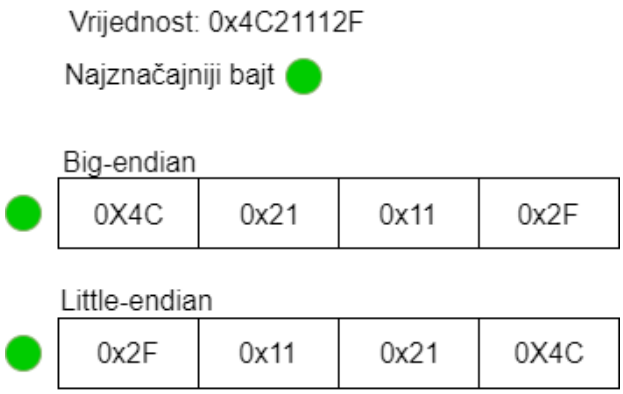
Snap7 API omogućuje kreiranje sustava u kojem računalo može biti klijent, poslužitelj i partner istovremeno. Klase koje to omogućuju su `Snap7Client`, `Snap7Partner` i `Snap7Server`. Jedna aplikacija može istovremeno predstavljati sve tri navedene uloge.

[25] Izvor: Nardella D.Snap7 *Reference manual*, str. 20

Aplikacija može slati zahtjev PLC uređaju i tako predstavljati klijenta, predstavljati vezu između više PLC uređaja i tako izvršavati ulogu partnera te odgovarati na zahtjeve PLC uređaja i tako obavljati ulogu poslužitelja (Nardella, 2015).

4.5. Struktura podatkovnih tipova

Podaci koje PLC pohranjuje u svojoj memoriji su zapisani u formatu *big-endian* (veliki *endian*), dok operacijski sustav Windows podatke čuva u *little-endian* (mali *endian*) formatu (Nardella, 2015). Razlika između dva spomenuta formata čuvanja podataka je u položaju najznačajnijeg bajta. U *big-endian* formatu je prvi bajt ujedno i najznačajniji bajt dok je u *little-endian* formatu obrnuto te se najznačajniji bajt nalazi na posljednjem mjestu.[26] Slika 4.4 prikazuje zapis cjelobrojne varijable veličine 4 bajta u oba formata.



Slika 4.4 Zapis cjelobrojne varijable u *big* i *little endian* formatu[27]

Prilikom čitanja varijabli u aplikaciji pisanoj u C# programskom jeziku, polje bajtova je potrebno preokrenuti kako bi se dobila točna vrijednost. Kôd 4.1 prikazuje primjer čitanja i konverzije varijable.

```
byte[] tempData = new byte[parameter.SizeInBytes];

// Pretvaranje iz big u little endian
Array.Copy(parameters, parameter.Offset, tempData, 0,
parameter.SizeInBytes);
Array.Reverse(tempData);
```

Kôd 4.1 Čitanje i konverzija varijable iz PLC uređaja

[26] Izvor: https://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Data/endian.html, rujan 2017.
[27] Izvor: Nardella D. Snap7 *Reference manual*, str. 22

Svi podatkovni tipovi iz Siemens S7-1200 uređaja imaju podudarajući podatkovni tip u programskom jeziku C#, osim podatkovnog tipa bit (Tablica 4.1). Bit je najbliži podatkovnom tipu *bool* iz C# jezika, međutim nije odgovarajuća veličina bajtova s obzirom da u C# programskom jeziku nije moguće instancirati varijablu manju od jednog bajta (Nardella, 2015).

Tablica 4.1 Podatkovni tipovi Siemens S7-1200 i odgovarajući podatkovni tipovi iz programskog jezika C#[28]

Siemens S7-1200	C# programski jezik	Veličina u bajtovima
BIT	Ne postoji odgovarajući podatkovni tip	1 bit
WORD	*ushort*	2
INT	*short*	2
DWORD	*uint*	4
DINT	*int*	4
REAL	*float*	4
S7 *strings*	*string*	1 bajt po znaku + 2 bajta početak i kraj
CHAR	*Byte*	1

4.6. Snap7 API konvencija

S obzirom da je API napisan u programskom jeziku C koji nije objektno orijentiran, funkcije nisu enkapsulirane u klase i strukture u izvornom kodu. Funkcije se enkapsuliraju u objekte, odnosno klase za više programske jezike kao C++ te C#, Python ili Java. Tri osnovne komponente koje čine arhitekturu sustava su Klijent, Partner i Poslužitelj te su stoga tri osnovne klase Snap7 bliblioteke: `S7Client`, `S7Server` i `S7Partner` (Nardella, 2015).

[28] Izvor: Nardella D., Snap7 Reference manual, str. 23

4.6.1. Konvencija naziva funkcija

Svi objekti se kreiraju pozivom naredbe *xxx*_`Create` gdje prefiks „xxx“ označava element iz arhitekture. Tako primjerice funkcije vezane uz klijenta počinju s prefiksom „Cli“, za partnera „Par“ te za poslužitelja „Srv“. Funkcija `Create` vraća pokazivač na memorijsku lokaciju kreiranog objekta preko kojeg se pristupa metodama. Poziv funkcije `Create`, odnosno kreiranje objekta je skriveno od korisnika zahvaljujući klasama omotačima (Nardella, 2015).

4.7. Snap7 API komponente korištene u radu

U radu je korištena samo klasa `S7Client` i svega par funkcija za povezivanje s uređajem te funkcija za čitanje i pisanje blokova memorije. S obzirom na funkcionalnosti modela simulacije pivovare (tema obrađena u poglavlju 6. Model simulacije pivovare) nije postojala potreba za manipulacijom ostalih elemenata PLC uređaja. Funkcije korištene u radu:

- `DBRead` – čitanje DB blokova
- `DBWrite` – pisanje u DB blokove
- `ConnectTo` – povezivanje s uređajem
- `Disconnect` – prekid veze s uređajem

5. Poslovni model sustava

Sustav se zasniva na povezivanju računala i PLC uređaja putem *etherneta*. Svrha poslovnog sustava je povezivanje s proizvodnim procesima te ubrzavanje obavještavanja u slučaju incidenata. PLC uređaj nije ovisan o radu *desktop* aplikacije, dok je ona ovisna o stanju uređaja. To znači da *desktop* aplikacija ne smije utjecati na proizvodnju, osim izmjenom parametara. *Desktop* aplikacija izravno komunicira s uređajem te služi kao sloj između web aplikacije i uređaja. Web aplikacija je dostupna unutar *intraneta*[29] te omogućuje generiranje izvještaja o proizvodnji te obavijesti u slučaju incidenata. Njezin primarni zadatak je omogućiti primanje izvještaja od klijenta, odnosno *desktop* aplikacije, pohrana podataka u bazu te obavještavanje nadležnih u slučaju incidenata.

5.1. *Desktop* aplikacija PS Operator

PS je skraćenica od *Production supervisor*, što bi u prijevodu značilo nadglednik u proizvodnji. Ostatak naziva *Operator* (operater) indicira da je korisnik aplikacije operater u proizvodnji. Aplikacija se sastoji od početnog prozora za autentikaciju te glavnog prozora koji sadrži pet različitih pogleda.

5.1.1. Autentikacija

Prilikom povezivanja s PLC uređajem, PS Operator izvodi autentikaciju. Za uspješan proces autentikacije, aplikacija čita jedinstveni identifikacijski broj uređaja spremljen u njegovu trajnu memoriju. Uz jedinstveni identifikacijski broj uređaja, korisnik unosi svoje korisničko ime i lozinku. Postupak autentikacije opisan je u poglavlju 6.2.1. Postupak autentikacije.

5.1.2. Glavni prozor

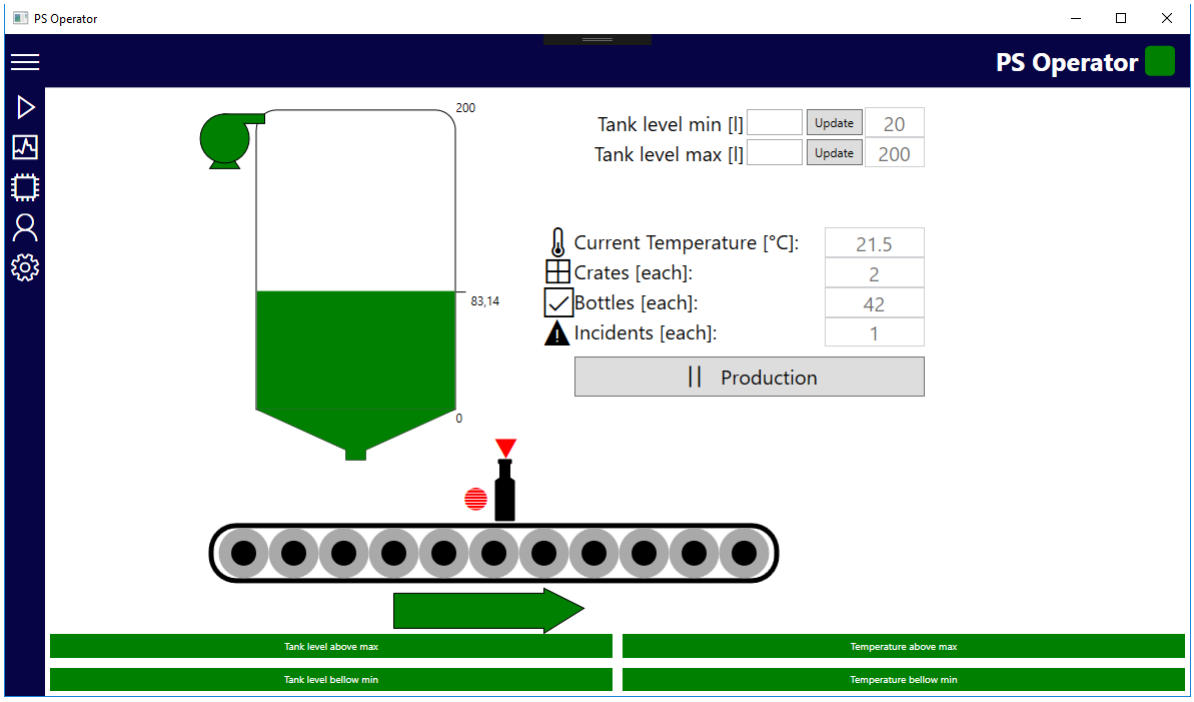
Glavni prozor se sastoji od glavnog izbornika koji sadrži poveznice na poglede. U gornjem desnom kutu glavnog prozora se nalazi indikator koji prikazuje stanje povezanosti uređaja i aplikacije. Zelena boja označava povezanost uređaja i aplikacije, dok crvena boja označava stanje nepovezanosti. Na dnu prozora se nalaze četiri indikatora pogrešaka u procesu. U

[29] *Intranet*: Privatna mreža računala kojoj mogu pristupiti samo autorizirani korisnici. Obično se odnosi na mrežu unutar organizacije. (Kunštek; Grgić, 2010)

slučaju normalnog stanja, pozadina im je zelene boje, dok se prilikom detekcije pogreške, odnosno incidenta, pozadina mijenja u crvenu boju. Pogreške u proizvodnji se odnose na odstupanje od dozvoljene razine tekućine u spremniku i odstupanje od dozvoljene temperature spremnika.

5.1.3. Prikaz proizvodnih procesa

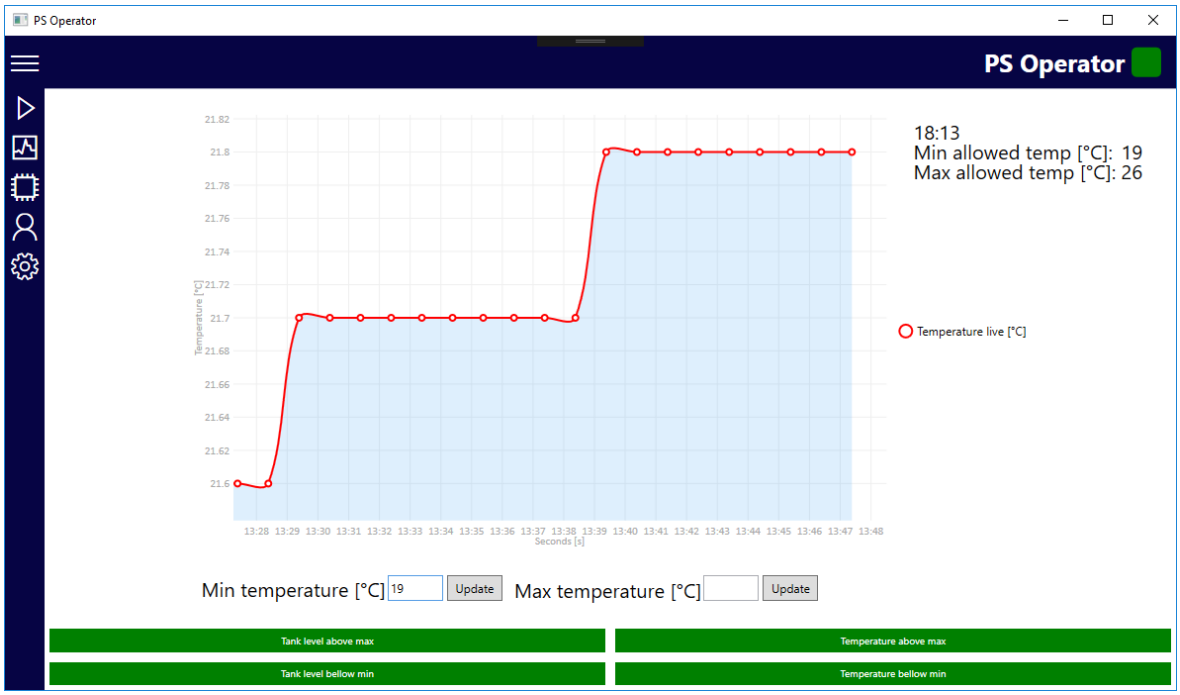
Pogled proizvodnja (engl. *production*) prikazuje trenutno stanje proizvodnog procesa. Slika 5.1 prikazuje stanje proizvodnog procesa. Gumb *Production* pokreće proizvodnu traku. Spremnik prikazuje razinu tekućine te mijenja boju iz zelene u crvenu u slučaju odstupanja od dozvoljene razine. Ukoliko je razina tekućine manja od 90 % razine spremnika, uključuje se pumpa. Uz prikaz boce nalaze se prikaz induktivnog i optičkog senzora. Stanje senzora se prikazuje crvenom i zelenom bojom ovisno o detekciji predviđenog objekta. Optički senzor detektira bocu, dok induktivni senzor detektira prisutnost zatvarača na boci. U slučaju detekcije boce bez zatvarača, dojavljuje se novi incident. Prikaz trenutne temperature, broj proizvedenih boca i sanduka te broj incidenata nalaze se na desnoj strani pogleda. Pogled također omogućuje određivanje željene minimalne i maksimalne razine spremnika tekućine.



Slika 5.1 Prikaz stanja proizvodnog procesa

5.1.4. Prikaz trenutne temperature

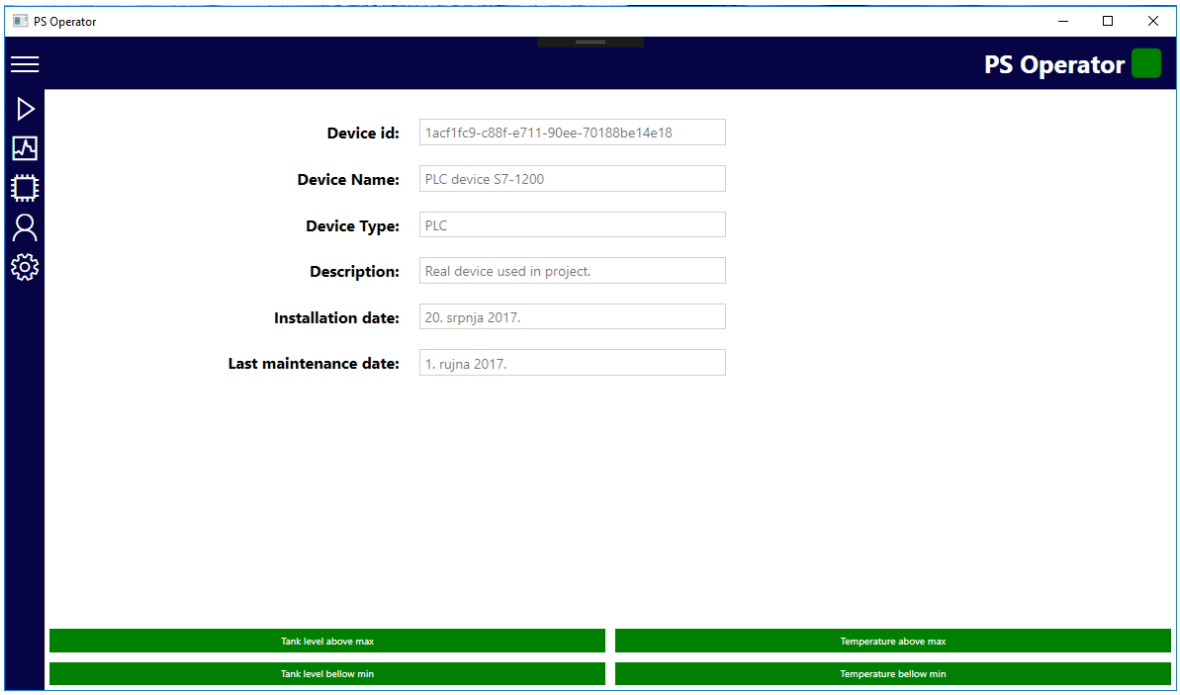
Zadaća pogleda dijagnostika (engl. *diagnostic*) je prikaz trenutne temperature u obliku linijskog dijagrama. Apscisna os prikazuje vrijeme mjerenja temperature, dok ordinatna os prikazuje izmjerenu temperaturu. Graf prikazuje interval od dvadeset sekundi i ažurira se svake sekunde. Na dnu pogleda nalaze se kontrole koje omogućuju izmjenu vrijednosti parametara maksimalne i minimalne dozvoljene temperature spremnika tekućine. Slika 5.2 prikazuje mjerenje temperature u intervalu od 13:28 do 13:48 (13 minuta i 48 sekundi) uz minimalnu dozvoljenu temperaturu 19 °C i maksimalnu 26 °C.



Slika 5.2 Prikaz trenutne temperature linijskim dijagramom

5.1.5. Prikaz podataka o uređaju

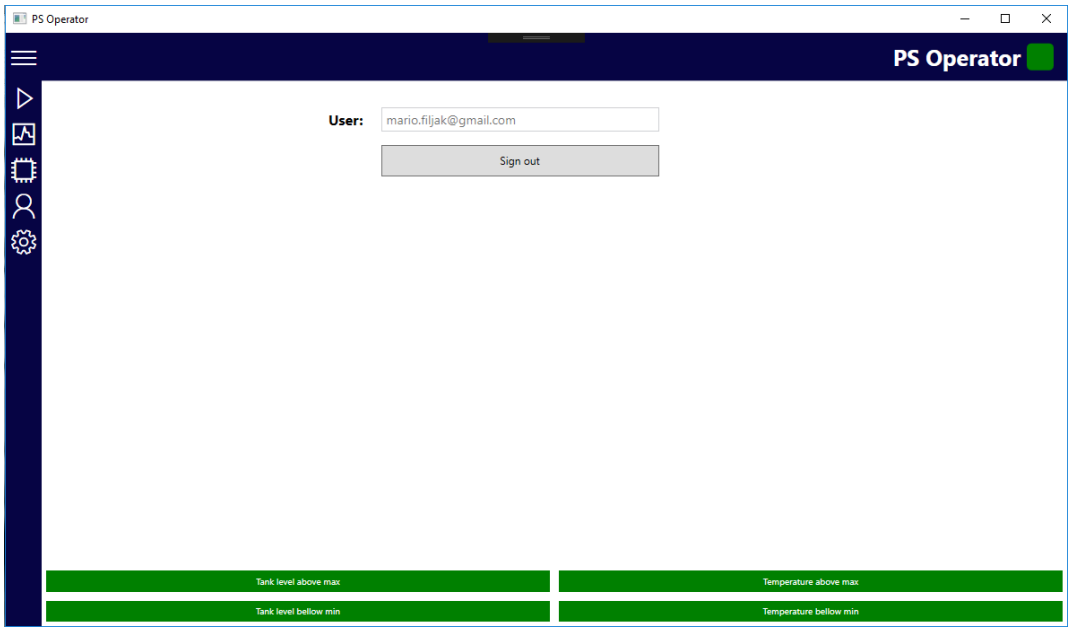
Pogled uređaj (engl. *device*) prikazuje podatke o uređaju s kojim je povezana aplikacija. Podaci dolaze s web aplikacije nakon uspješne autentikacije. Podaci prikazuju jedinstveni identifikacijski broj uređaja, naziv, tip, opis, datum instalacije uređaja i datum zadnjeg održavanja. Slika 5.3 prikazuje podatke dobivene s web aplikacije nakon uspješne autentikacije.



Slika 5.3 Prikaz podataka o povezanom uređaju

5.1.6. Prikaz korisničkog računa i odjava

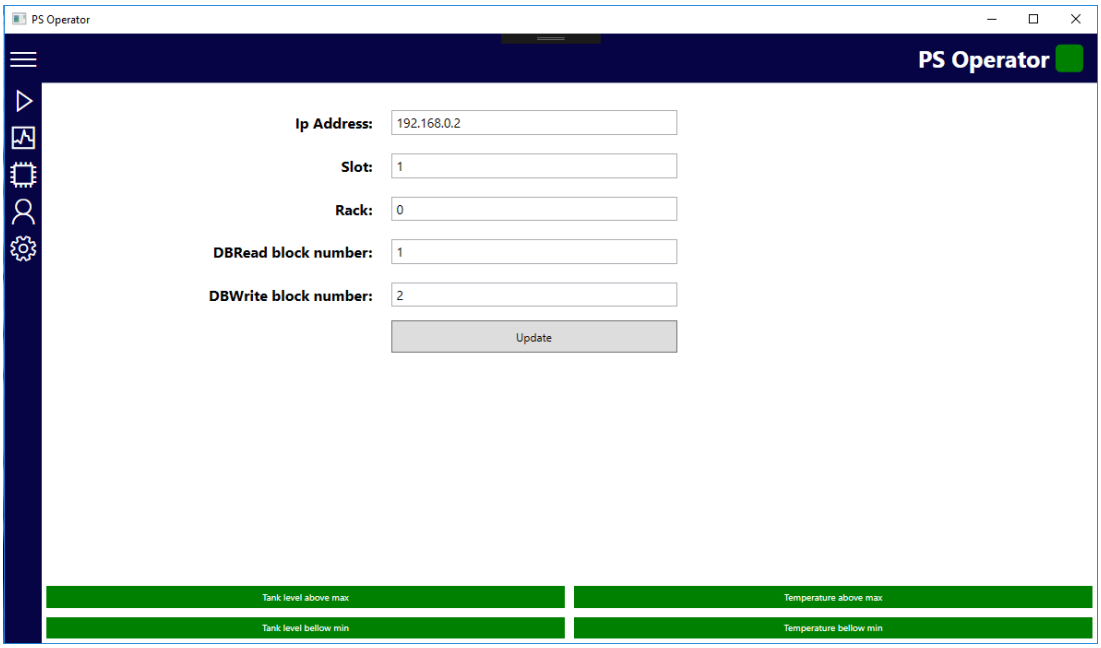
Pogled korisnički račun (engl. *account*) prikazuje korisničko ime prijavljenog korisnika te sadrži gumb za odjavu korisnika. Slika 5.4 prikazuje prijavljenog korisnika s korisničkim imenom mario.filjak@gmail.com.



Slika 5.4 Prikaz korisničkog računa i odjava

5.1.7. Postavke

Pogled postavke omogućuje izmjenu postavki potrebnih za povezivanje i komuniciranje s uređajem. Slika 5.5 prikazuje postavke uređaja koje čita iz datoteke *Settings.xml* prilikom pokretanja aplikacije.



Slika 5.5 Postavke uređaja

Postavke su IP adresa uređaja, utor i postolje uređaja, broj podatkovnog bloka iz kojeg se čitaju varijable i broj podatkovnog bloka u koji se pohranjuju trajne vrijednosti. Trajne vrijednosti su one vrijednosti koje ostaju pohranjene nakon isključivanja uređaja.

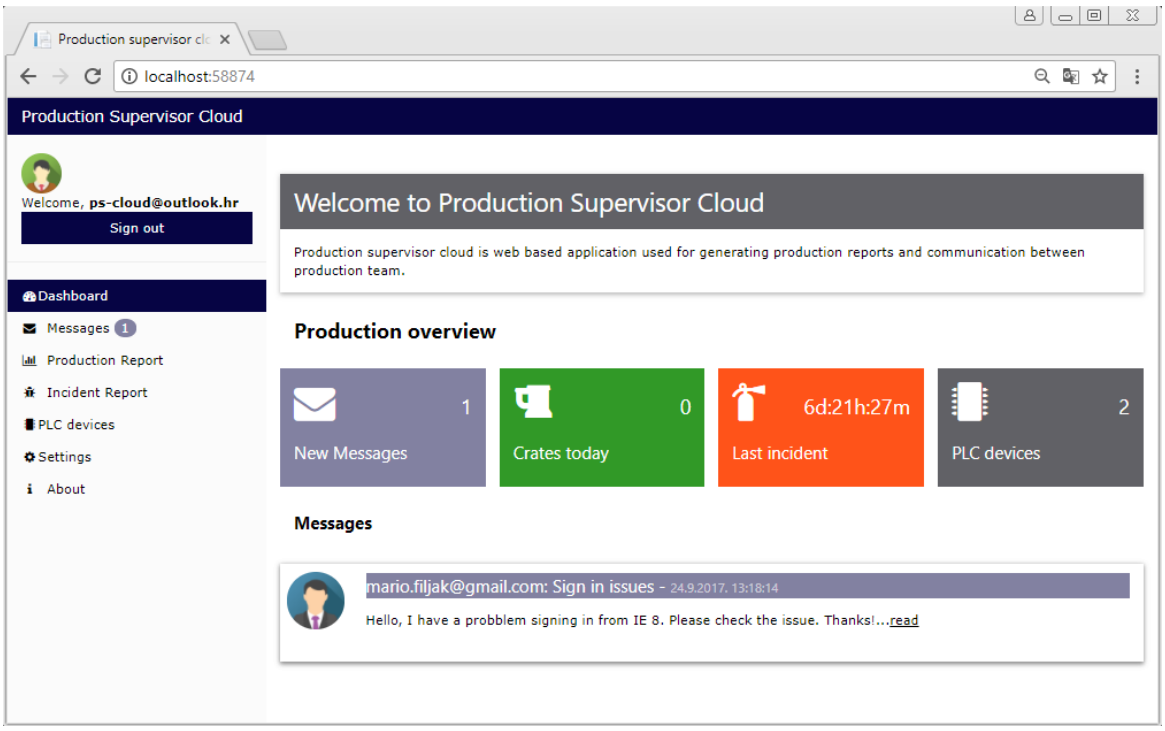
5.2. Web aplikacija PS Cloud

PS označava *Production supervisor* (nadglednik u proizvodnji), kao i kod PS Operator *desktop* aplikacije. *Cloud* (oblak) označava da se aplikacija izvršava u oblaku. Web aplikacija ima tri primarne zadaće: prikaz izvještaja i međusobna komunikacija korisnika, REST[30] (reprezentacijski prijenos stanja) poslužitelj za *desktop* aplikaciju te servis za obavještavanje o incidentima.

[30] REST je protokol koji omogućuje interoperabilnost među informacijskim sustavima i aplikacijama. Više u 8.3. REST protokol i Web API 2

5.2.1. Upravljačka ploča

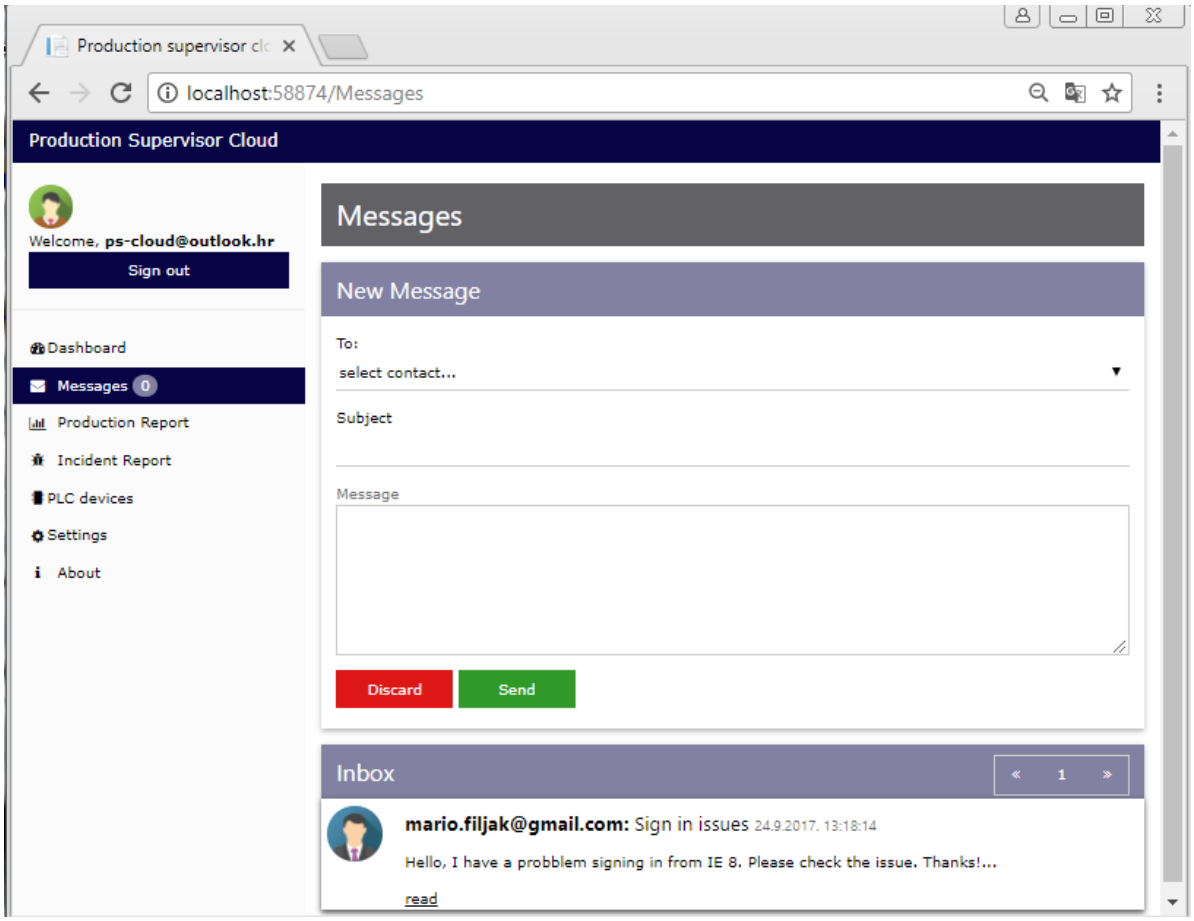
Upravljačka ploča (engl. *dashboard*) prikazuje broj novih poruka, broj proizvedenih sanduka, vrijeme zadnjeg incidenta te broj uređaja u sustavu. Na dnu pogleda upravljačka ploča se nalazi pet najnovijih poruka koje je korisnik primio. Slika 5.6 prikazuje upravljačku ploču za korisnika ps-cloud@outlook.com.



Slika 5.6 Pogled upravljačka ploča

5.2.2. Poruke

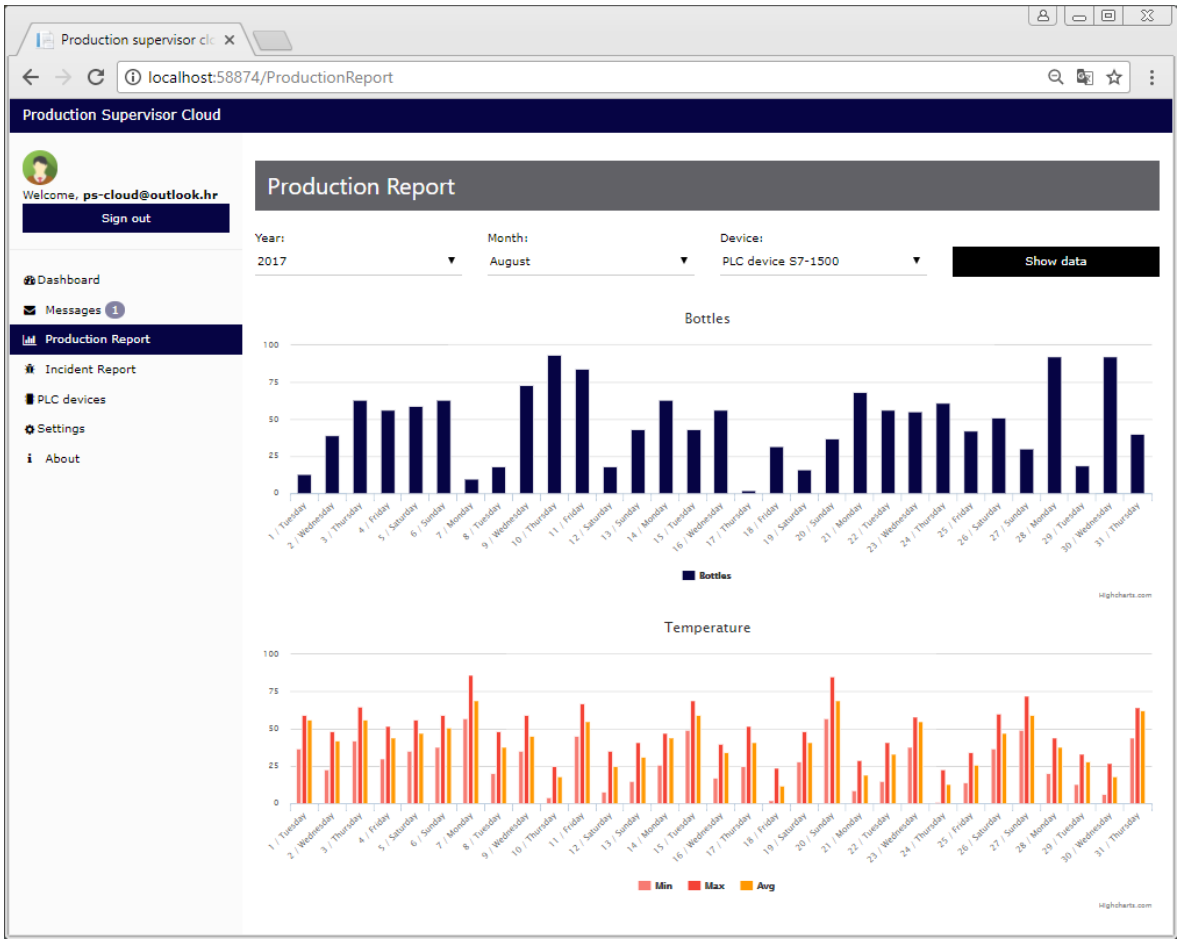
Pogled poruke (engl. *messages*) omogućuje slanje poruka registriranim korisnicima unutar sustava. Namjena poruka je dodjeljivanje zadataka, slanje podsjetnika ili traženje korisničke podrške. Slika 5.7 prikazuje formu za slanje poruke korisnicima unutar sustava. Korisnik odabire kontakt iz padajućeg izbornika. Na dnu stranice se nalaze primljene poruke od drugih korisnika.



Slika 5.7 Forma za slanje poruka i prikaz pristigle pošte

5.2.3. Izvještaj o proizvodnji

Pogled izvještaj o proizvodnji (engl. *production report*) prikazuje podatke o količini proizvedenih boca te minimalnoj, maksimalnoj i prosječnoj zabilježenoj temperaturi spremnika tekućine. Izvještaji su podijeljeni prema mjesečnom periodu i uređajima. Korisnik odabire godinu i mjesec u kojem želi pogledati izvještaj te uređaj za koji želi vidjeti podatke. Slika 5.8 prikazuje izvještaj za virtualni uređaj S7-1500 u kolovozu 2017. godine. Stupčasti grafovi prikazuju podatke za svaki dan unutar odabranog mjeseca. Graf na vrhu pogleda prikazuje broj proizvedenih boca u određenom danu. Graf na dnu pogleda prikazuje zabilježenu minimalnu, maksimalnu i prosječnu temperaturu spremnika tekućine.

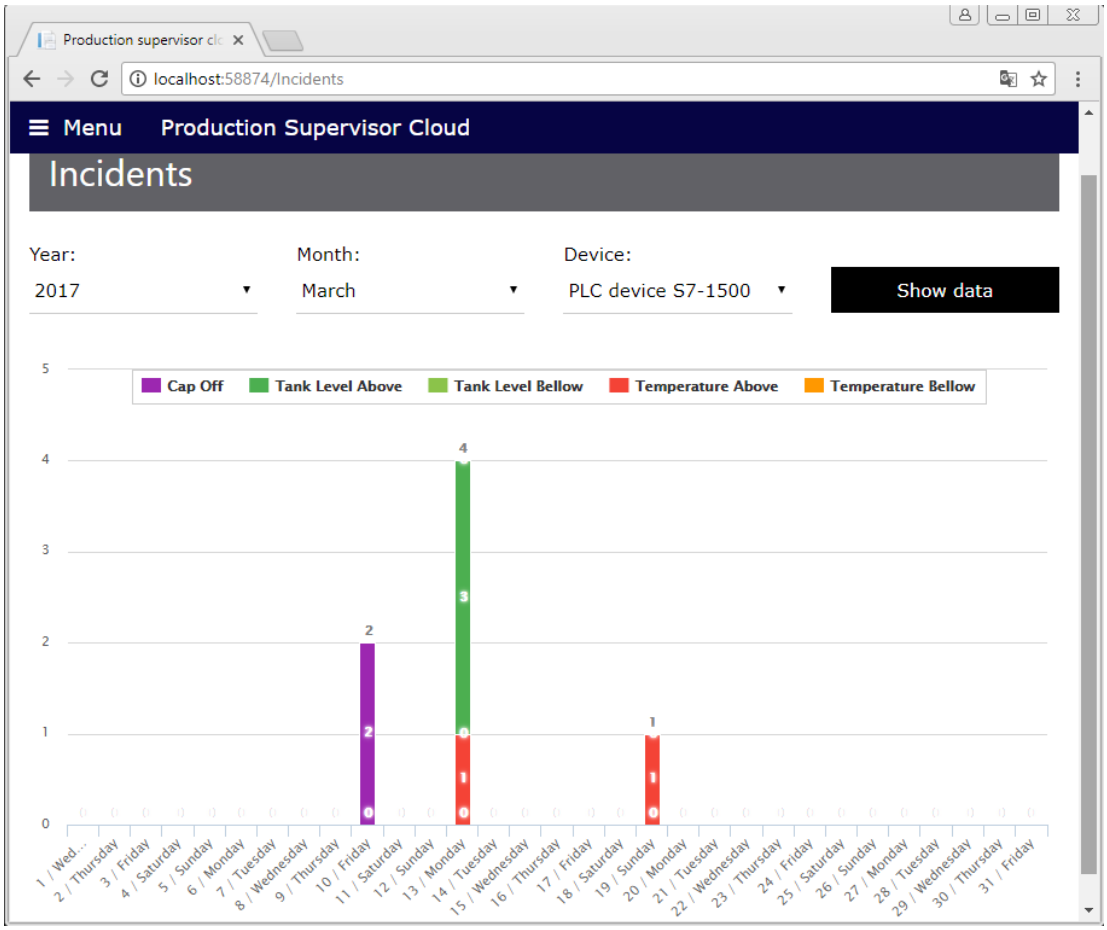


Slika 5.8 Prikaz izvještaja o proizvodnji za virtualni uređaj S7-1500 u kolovozu 2017.[31]

5.2.4. Izvještaji o incidentima

Pogled izvještaji o incidentima (engl. *incident report*) omogućuje pregled incidenata u odabranom mjesecu. Korisnik odabire godinu, mjesec i uređaj za koji želi pogledati podatke o incidentima. Incidenti koji se prate i bilježe jesu razina spremnika tekućine ispod ili iznad dozvoljene, temperatura spremnika tekućine ispod, odnosno iznad dozvoljene i detektirana boca bez zatvarača. Slika 5.9 prikazuje izvještaj za virtualni uređaj S7-1500 u ožujku 2017. godine. Podaci su prikazani u obliku stupčastog grafa te prikazuju broj incidenata za svaki dan u odabranom mjesecu.

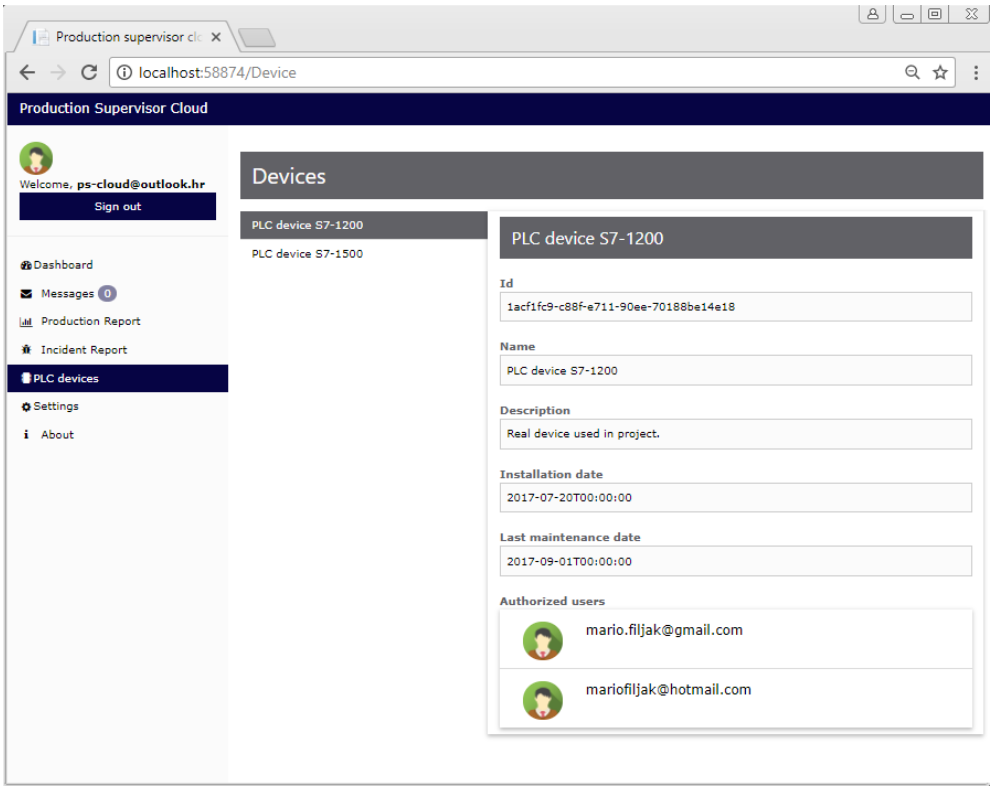
[31] S7-1500 je tip uređaja iz serije S7. Stvarni uređaj tipa S7-1500 unutar sustava ne postoji. Virtualni uređaj s nasumičnim podacima je namijenjen za prezentacijske svrhe.



Slika 5.9 Prikaz izvještaja o incidentima za virtualni uređaj S7-1500 u ožujku 2017.

5.2.5. PLC uređaji

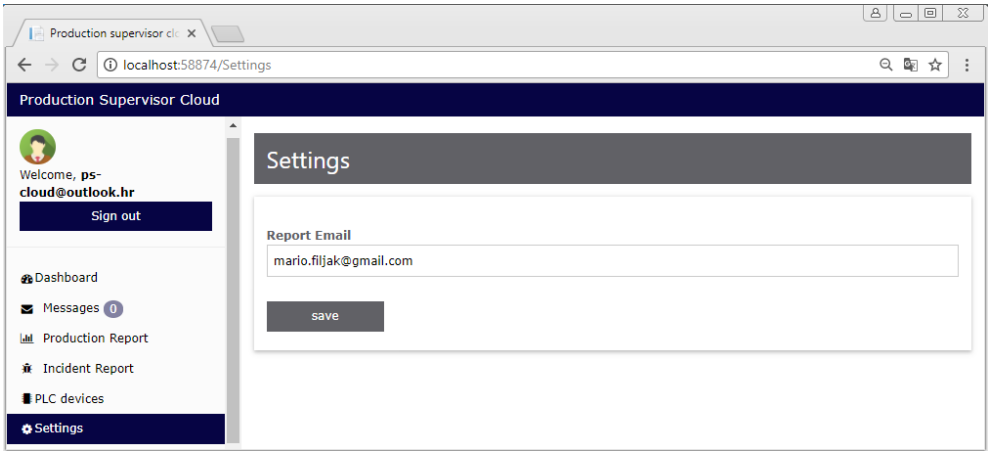
Pogled PLC uređaji (engl. *devices*) prikazuje podatke o uređajima u sustavu i korisnike koji su autorizirani za upravljanje. Slika 5.10 prikazuje podatke za registrirani uređaj S7-1200 te autorizirane korisnike mario.filjak@gmail.com i mariofiljak@hotmail.com.



Slika 5.10 Prikaz uređaja s autoriziranim korisnicima

5.2.6. Postavke

Pogled postavke (engl. *settings*) omogućuje unos *email* adrese korisnika koji će biti obaviješten prilikom incidenta. Slika 5.11 prikazuje postavke u kojima je nadležni korisnik za primanje obavijesti mario.filjak@gmail.com.



Slika 5.11 Postavke kontakta za slanje obavijesti o incidentu

5.2.7. Korisničke role

Super user (super korisnik) je rola koja može pristupiti svim pogledima aplikacije, a svrha ove role je pružiti programerima cjeloviti pristup prilikom razvoja i održavanja aplikacije. *Engineer* (inženjer) je rola namijenjena proizvodnim inženjerima i tehnolozima. Oni mogu pregledavati podatke o uređajima i određivati koji korisnici će primati obavijesti u slučaju incidenata. *Maintenance* (održavanje) je rola namijenjena operaterima i radnicima u održavanju koji mogu pregledavati podatke o uređajima. *Manager* (menadžer) je rola namijenjena menadžerima koji mogu pregledavati izvještaje o proizvodnji i incidentima. Tablica 5.1 prikazuje poglede i pravo pristupa prema rolama. Poglede upravljačka ploča i poruke mogu vidjeti sve role.

Tablica 5.1 Korisničke role i pravo pristupa

Korisnička rola	Izvještaji o proizvodnji	Izvještaji o incidentima	Postavke	PLC uređaji
Super user	O	O	O	O
Manager	O	O	X	X
Engineer	X	X	O	O
Maintenance	X	X	O	O

5.2.8. Servis za obavještavanje incidenata

U slučaju prijave incidenta od strane *desktop* aplikacije, web aplikacija šalje obavijest putem elektroničke pošte definiranom korisniku. Aplikacija se spaja na poslužitelja elektroničke pošte putem SMTP (protokol za slanje poruka)[32] protokola.

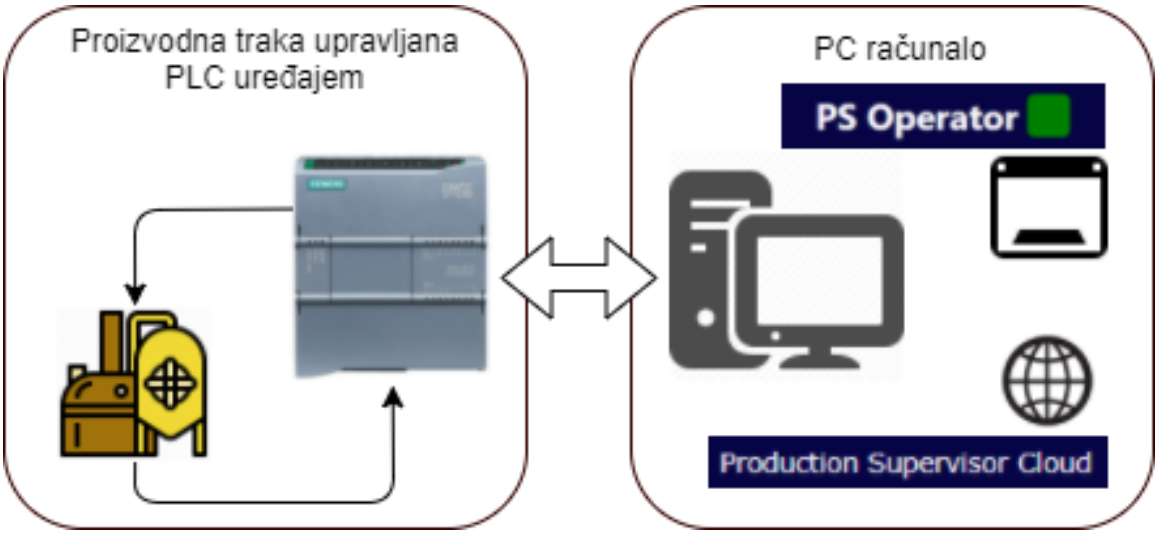
[32] SMTP (engl. *Simple Message Transfer Protocol*) je komunikacijski protokol čija je osnovna namjena siguran i pouzdan prijenos poruka elektroničke pošte (eng. *email*) bez obzira na tehnologiju i sustav koji se koristi za ostvarenje samog prijenosa. (Izvor: http://www.cis.hr/www.edicija/SMTPprotokol.html, listopad 2017.)

5.3. Prednosti poslovnog modela

Mogućnost vizualnog prikaza stanja proizvodnog procesa u stvarnom vremenu glavna je prednost sustava. Time se operateru omogućuje brzo otkrivanje pogreške te pravovremena reakcija. Ukoliko se radi o proizvodnom procesu u kojem operater nije stalno prisutan uz stroj, brzo otkrivanje pogreške omogućuje obavijest putem elektroničke pošte. Zaposlenici koji nisu operateri i koji nemaju izravan pristup stroju, rezultate proizvodnog procesa mogu pratiti putem web aplikacije. S obzirom da je prikupljanje podataka automatizirano, sustav omogućuje uštedu vremena.

6. Model simulacije pivovare

Model simulacije pivovare je osmišljen s ciljem prikaza praktične primjene sustava koji je tema ovog rada te predstavlja simulaciju praćenja proizvodnog procesa u virtualnoj pivovari. Slika 6.1 prikazuje sustav pivovare koji se sastoji od tri glavna elementa: Proizvodna traka upravljana PLC uređajem, *desktop* aplikacija PS Operator i web aplikacija PS Cloud na PC računalu. Proizvodna traka se sastoji od spremnika tekućine, pumpe za doziranje tekućine, pokretne trake i senzora za detekciju boce i zatvarača.



Slika 6.1 Elementi sustava pivovare

6.1. Popis funkcionalnosti

Svi elementi sustava su međusobno povezani te svaki od njih ima određene zadaće i funkcionalnosti. Jedini element u sustavu koji može raditi samostalno je PLC uređaj.

Lista funkcionalnosti PLC uređaja:

- Pokretanje proizvodnje - Postavljanjem varijable `Line_running` na vrijednost 1, pokreće se proces proizvodnje.
- Pokretanje pumpe - Ukoliko je razina tekućine ispod 90% definirane razine, pumpa se uključuje. Ukoliko razina naraste iznad definirane vrijednosti, pumpa se isključuje.
- Mjerenje razine tekućine u spremniku - Koristeći ultrazvučni senzor, koji je opisan u poglavlju 6.3.3. Ultrazvučni senzor, uređaj mjeri razinu tekućine u spremniku.

Skalirana vrijednost se računa jednadžbom pravca kroz dvije točke. Postupak skaliranja pojašnjen je u poglavlju 3.4.3. SCL jezik te primjeru kôda 3.1 Skaliranje vrijednosti ultrazvučnog senzora jednadžbom pravca kroz dvije točke.

- Mjerenje temperature spremnika tekućine - Za mjerenje temperature sustav koristi PT100 temperaturnu sondu koja je opisana u poglavlju 6.3.2. Temperaturna sonda PT100.
- Brojač ispravnih i neispravnih boca – Optički senzor detektira prisutnost boce. Ukoliko induktivni senzor ne detektira prisutnost zatvarača, varijabla broja neispravnih boca se povećava za jedan, dok se u kod detekcije prisutnosti zatvarača varijabla broja ispravnih boca povećava za jedan.

Lista funkcionalnosti *desktop* aplikacije:

- Autentikacija – Postupak autentikcije opisan je u poglavlju 6.2.1. Postupak autentikacije
- Pokretanje proizvodne linije
- Određivanje maksimalne i minimalne dozvoljene razine tekućine u spremniku
- Prikaz razine tekućine spremnika
- Prikaz stanja pumpe
- Određivanje maksimalne i minimalne dozvoljene temperature
- Prikaz temperature linijskim grafom koji se ažurira u stvarnom vremenu
- Izmjena postavki povezivanja s PLC uređajem
- Prikaz podataka o uređaju
- Praćenje minimalne, maksimalne i prosječne izmjerene temperature
- Obavještavanje operatera u slučaju odstupanja od dozvoljene razine tekućine u spremniku
- Obavještavanje operatera u slučaju odstupanja od dozvoljene temperature spremnika

Lista funkcionalnosti web aplikacije:

- Pregled podataka o proizvedenoj količini boca – Stupčasti graf prikazuje broj proizvedenih boca u danu za odabrani uređaj u odabranom mjesecu.
- Pregled podataka o temperaturi spremnika – Stupčasti graf prikazuje minimalnu, maksimalnu i prosječnu izmjerenu temperaturu spremnika za odabrani uređaj u odabranom mjesecu.

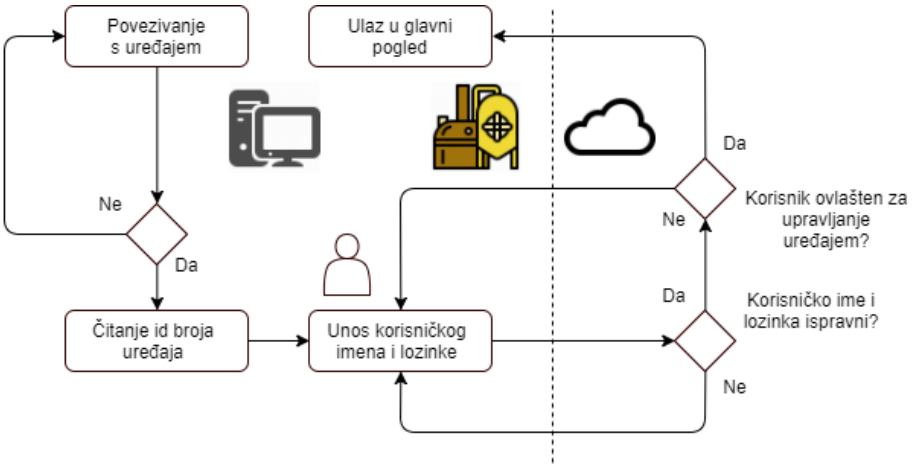
- Pregled podataka o zabilježenim incidentima – Stupčasti graf prikazuje broj zabilježenih incidenata u određenom danu za odabrani uređaj. Incidenti su grupirani prema tipu incidenta.
- Pregled podataka o uređajima u sustavu – Uz pregled podataka prikazani su korisnici autorizirani za korištenje uređaja.
- Unos *email* adrese korisnika koji će biti obaviješten o incidentima.
- Obavještavanje nadležne osobe o incidentu putem elektroničke pošte.
- Autentikacija korisnika i uređaja – provjera da li je korisnik autoriziran za upravljanje uređajem.
- Zaprimanje i zapisivanje izvještaja o proizvodnji.
- Zaprimanje i zapisivanje incidenata.

6.2. Kontrola toka

U narednim poglavljima su objašnjeni autentikacija i odjava, pokretanje i zaustavljanje proizvodne linije te generiranje izvještaja o proizvodnji i identifikaciji incidenata. Svrha poglavlja je detaljnije objasniti navedene postupke kako bi se podržale teze i objašnjenja iz prethodnih poglavlja i cjelina.

6.2.1. Postupak autentikacije

Prilikom pokretanja *desktop* aplikacije započinje postupak povezivanja uređaja s aplikacijom. Ukoliko je veza ostvarena, *desktop* aplikacija čita jedinstveni identifikacijski broj uređaja. Korisnik unosi korisničko ime i lozinku te *desktop* aplikacija šalje zahtjev web aplikaciji. Ukoliko su korisničko ime i lozinka ispravni, a korisnik je autoriziran za upravljanje uređajem, web aplikacija šalje odgovor s podacima o uređaju i korisniku. U protivnom, web aplikacija vraća poruku o pogrešci prilikom autentikacije. Ukoliko je autentikacija uspješna, *desktop* aplikacija ulazi u glavni pogled. Slika 6.2 prikazuje tijek postupka autentikacije.



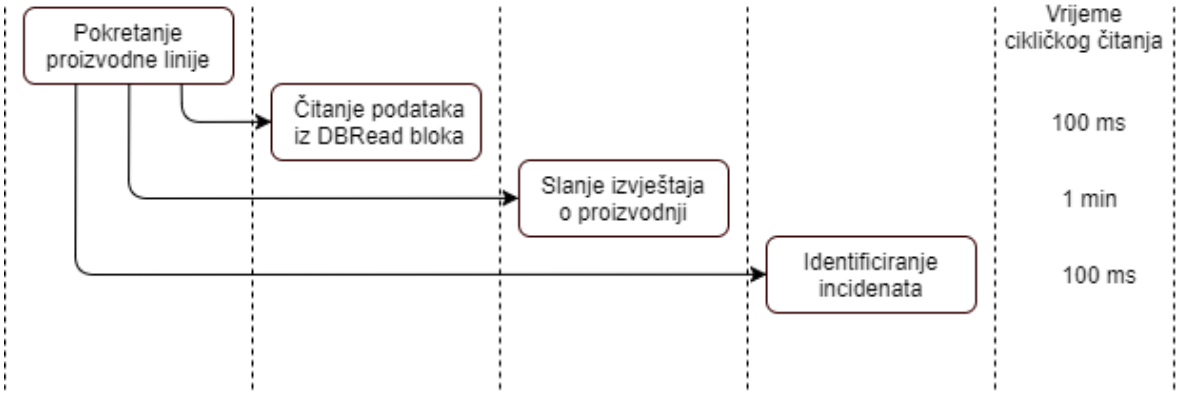
Slika 6.2 Postupak autentikacije

6.2.2. Odjava korisnika

Pritiskom na gumb *Sign out* korisnik se odjavljuje iz *desktop* aplikacije koja ga preusmjerava na početni prozor za prijavu. PLC uređaj nastavlja s radom ukoliko korisnik nije zaustavio proizvodnu liniju.

6.2.3. Pokretanje proizvodne linije

Pritiskom na gumb *Production* unutar *desktop* aplikacije, varijabla `Line_Running` unutar PLC uređaja se postavlja u 1. Navedenom operacijom započinje cikličko čitanje podataka iz bloka *DBRead*. Čitanje se izvršava u intervalima od 100 ms. Paralelno uz čitanje podataka započinje postupak cikličkog izvještavanja web aplikacije koje se izvršava u intervalima od jedne minute. Slika 6.3 prikazuje vremenski dijagram cikličkih radnji.



Slika 6.3 Pokretanje proizvodne linije

6.2.4. Zaustavljanje proizvodne linije

Pritiskom na gumb *Production* unutar *desktop* aplikacije, varijabala `Line_Running` unutar PLC uređaja se postavlja u 0. Ovime se zaustavljaju ciklički zadaci pokrenuti prilikom pokretanja proizvodne linije.

6.2.5. Generiranje izvještaja o proizvodnji

Desktop aplikacija svake minute kreira izvještaj o količini proizvedenih boca te minimalnoj, prosječnoj i maksimalnoj temperaturi. Izvještaje šalje web aplikaciji u intervalima od jedne minute. Web aplikacija pohranjuje izvještaje u bazu podataka.

6.2.6. Identificiranje incidenata

Desktop aplikacija čita podatke iz uređaja, vrši provjeru o odstupanju temperature spremnika od dozvoljene vrijednosti te odstupanje tekućine u spremniku od dozvoljene razine. Ukoliko *desktop* aplikacija detektira incident, šalje obavijest web aplikaciji. Web aplikacija pohranjuje podatke o incidentu u bazu podataka i šalje obavijest u obliku elektroničke pošte nadležnom korisniku.

6.3. Hardver komponente

Sustav se sastoji od četiri senzora, PLC uređaja s adapterom za PT100 temperaturnu sondu te PC računala. U ovom poglavlju opisane su sve glavne komponente osim računala i PLC uređaja.

6.3.1. SIMATIC S7-1200 adapter za PT100 temperaturnu sondu

SIMATIC S7-1200 adapter je ulazna analogna kartica za prihvat signala sa otporničkih temperaturnih senzora koja omogućuje mjerenje temperature PT100 temperaturnom sondom.[33] Razlika između analognog i digitalnog ulaza odnosno izlaza opisana je u poglavlju 3.2.2. Digitalni i analogni priključci.

[33] Izvor:https://mall.industry.siemens.com/mall/en/ww/Catalog/Product/6ES7231-5PA30-0XB0, listopad 2017.

6.3.2. Temperaturna sonda PT100

Temperaturna sonda sadrži otpornik koji mijenja vrijednost otpora s obzirom na promjenu temperature. Pri vrijednosti temperature od 0 °C otpor iznosi 100 Ω, dok na temperaturi od 100 °C otpor iznosi 138,4 Ω. Raspon mjerenja temperature se kreće od -200 °C do 850 °C.[34]

6.3.3. Ultrazvučni senzor

Ultrazvučni senzori mjere udaljenost koristeći ultrazvučnu zraku. Vrijeme između registriranja odbijene zrake proporcionalno je udaljenosti predmeta. Što je dulje vrijeme potrebno da se poslana zraka vrati, to je promatrani predmet udaljeniji od senzora.[35] Karakteristike senzora korištenog u modelu simulacije pivovare[36]:

- Mjerno područje: 0-200 mm
- Napajanje: 24 VDC
- Izlazni signal: 0-10 V

6.3.4. Optički senzor

Optički senzori se koriste za detekciju prisustva predmeta, a za detekciju koriste svjetlosnu zraku. Difuzni optički senzor detektira predmete putem slanja svjetlosne zrake (infracrvena, LED, laserska) i registriranja ukoliko se ona odbija od predmeta (Ifm *electronic Photoelectric sensors*, 1.1 *Photoelectric sensors in industrial applications*, str. 6).

6.3.5. Induktivni senzor

Induktivni senzor služi za detekciju prisustva metalnih predmeta. Induktivni senzor na izlazu daje signal ukoliko je u osjetnom području senzora prisutan metalni predmet. Signal se koristi za daljnju logiku procesa. U modelu simulacije pivovare induktivni senzor detektira prisutnost zatvarača boce (Ifm *electronic Inductive sensors*, 1.1 *Proximity switches in industrial processes*, str. 5).

---

[34] Izvor: https://www.picotech.com/library/application-note/pt100-platinum-resistance-thermometers, listopad 2017.
[35] Izvor: https://www.baumer.com/ch/en/functionality-and-technology-of-ultrasonic-sensors/a/know-how_function_ultrasonic-sensors, listopad 2017.
[36] Izvor: https://www.baumer.com/ch/en/product-overview/distance-measurement/ultrasonic-sensors/miniature/cylindrical/unam-12u9914-s14/p/27806, rujan 2017.

7. *Desktop* aplikacija

PS Operator je *desktop* aplikacija koja izravno komunicira i upravlja s PLC uređajem. Napravljena je u WPF (*windows* prezentacijske forme) tehnologiji te implementira MVVM (model, pogled, model za pogled) obrazac.

7.1. Primjena

PS Operator se izvršava na računalu koje direktno komunicira sa PLC uređajem putem TCP/IP protokola. Za komunikaciju s PLC uređajem *desktop* aplikaciji koristi Snap7 API sučelje opisano u poglavlju 4. Snap7. Zadaća aplikacije je prikaz trenutnog stanja u proizvodnom procesu te slanje izvještaja poslužitelju u određenim intervalima.

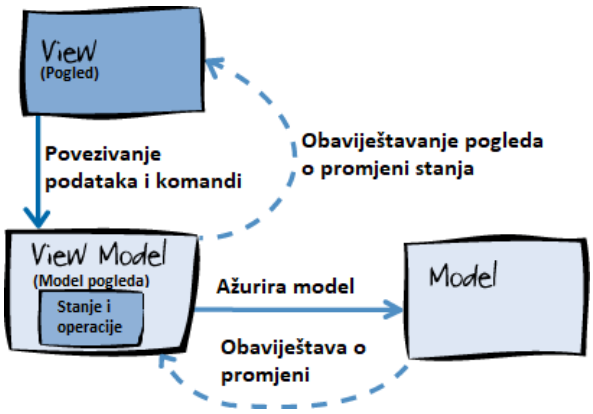
7.2. WPF

WPF je grafički sustav namijenjen renderiranju korisničkog sučelja (eng. *user interface*). Dio je .NET radnog okvira (engl. *framework*) te je podržan na Windows operacijskom sustavu. Odvajanje korisničkog sučelja od poslovne logike jedna je od najbitnijih značajki WPF sustava. Najznačajniji koncepti sustava su povezivanje podataka između poslovne logike i korisničkog sučelja (engl. *data binding*), animacije, renderiranje dvodimenzionalne i trodimenzionalne grafike te neovisnost o rezoluciji zaslona (Andrade et al., 2007).

7.3. MVVM oblikovni obrazac

MVVM omogućava odvajanje korisničkog sučelja od poslovne logike u kôdu. Komponente od kojih se obrazac sastoji su model, *view* (pogled) i *view model* (model pogleda). Model predstavlja klase iz poslovne domene aplikacije. Direktan pristup modelu ima element *view model*. Njegova namjena je priprema podataka koji će biti prikazani na pogledu (engl. *view*), obavještavanje pogleda o promjeni vrijednosti modela te ažuriranje podataka u modelu. Pogled odnosno korisničko sučelje je povezano s *view modelom* putem koncepta povezivanja podataka. Prikaz podataka na korisničkom sučelju se mijenja kada *view model* obavijesti pogled o promjeni. Pogled i *view model* su povezani komandama (engl. *commands*).

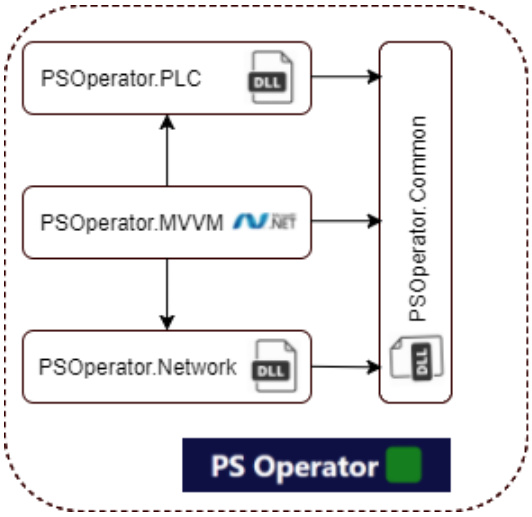
Pozivanjem određene komande, *view model* izvršava operacije te ažurira vrijednost modela.[37] Slika 7.1 prikazuje odnos elemenata unutar MVVM obrasca.



Slika 7.1 MVVM oblikovni obrazac[38]

7.4. Struktura koda

Aplikacija se sastoji od četiri projekta (Visual studio[39] projekt). Slika 7.2 prikazuje odnos projekata unutar aplikacije. Početni projekt je PSOperator.MVVM, koji je tipa WPF. Ostali projekti PSOperator.Common, PSOperator.Network i PSOperator.PLC su tipa biblioteka klasa (engl. *class library*). Početni projekt referencira sve projekte, dok zajednički sloj PSOperator.Common referenciraju svi projekti.



Slika 7.2 Arhitektura PS Operator aplikacije

[37] Izvor: https://msdn.microsoft.com/en-us/library/hh848246.aspx, listopad 2017.
[38] Izvor: https://i-msdn.sec.s-msft.com/dynimg/IC564167.png, listopad 2017.
[39] Visual studio je Microsoftova razvojna okolina za .NET okruženje (https://www.visualstudio.com, listopad 2017.)

7.4.1. PSOperator.MVVM

PSOperator.MVVM projekt je strukturiran prema MVVM oblikovnom obrascu te sadrži prezentacijski sloj aplikacije. Imenski prostor (engl. *namespace*) `Models` sadrži modele iz poslovne logike. *View modeli* su organizirani prema načelu u kojem svaki pogled ima svoj odgovarajući *view model*. Aplikacija sadrži dva pogleda `MainWindow` i `LoginWindow`, koji predstavljaju korisničko sučelje. Pogledi `Production`, `Diagnostic`, `Device`, `Account` i `Settings`, opisani u poglavlju 5.1. *Desktop* aplikacija PS Operator, su implementacije korisničkih kontrola (engl. *user control*). Navedeni pogledi nasljeđuju klasu `UserControl` koja im omogućuje vlastitu implementaciju prikaza na prezentacijskom sloju te vlastitu pozadinsku logiku.

7.4.2. PSOperator.PLC

PSOperator.PLC projekt predstavlja sloj komunikacije s PLC uređajem koji se zasniva na Snap7 aplikacijskom programskom sučelju. Klasa `SiemensPLC` predstavlja sučelje između `S7Client` klase, opisane u poglavlju 4.6. Snap7 API konvencija, i poslovne logike aplikacije.

7.4.3. PSOperator.Network

PSOperator.Network projekt predstavlja sloj za komunikaciju *desktop* aplikacije sa web aplikacijom putem REST protokola. Osnovna klasa je `PSCloudClient` čija je zadaća slanje izvještaja na PS Cloud aplikaciju. Imenski prostor `Models` sadrži modele koji su parametri WEB API[40] metoda.

7.4.4. PSOperator.Common

PSOperator.Common projekt je sloj kojeg referenciraju svi projekti iz rješenja. Sadrži enumeracije, korisnički definirane iznimke te pomoćne klase.

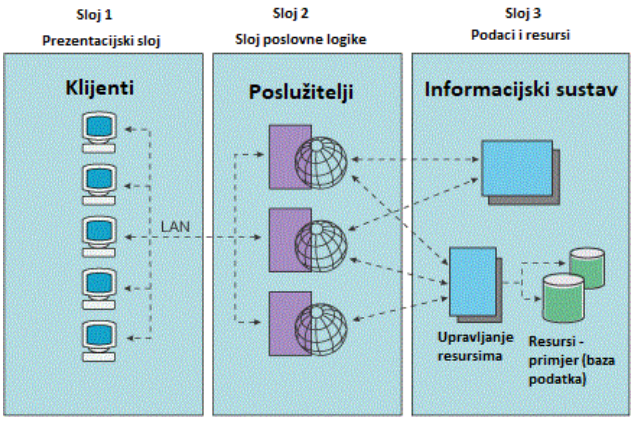
[40] Više o WEB API programskom sučelju u 8.3. REST protokol i Web API 2

8. Web aplikacija

Aplikacija PS Cloud predstavlja servis za prikupljanje i prikaz izvještaja. Izvještaji se prikupljaju iz *desktop* aplikacije putem REST protokola. Korisnici aplikacije su inženjeri, članovi odjela održavanja te menadžeri. Struktura aplikacije je oblikovana prema troslojnoj arhitekturi. Tehnologija korištena za implementaciju PSCloud aplikacije je ASP.NET MVC, dok je za bazu podataka korišten SQL Server.

8.1. Model troslojne arhitekture

Svrha modela troslojne arhitekture (engl. *three tier architecture model*) je odvajanje segmenata aplikacije u tri sloja. Slika 8.1 prikazuje prezentacijski sloj, sloj poslovne logike i podatkovni sloj. Prezentacijski sloj je sučelje prema korisniku te omogućuje prikaz i manipulaciju podacima. Sloj poslovne logike se zasniva na servisima. Servisi su klase koje imaju zadaću obrade i distribucije podataka prema poslovnim pravilima. Podatkovni sloj ima pristup resursima (baze podataka, datotečni sustavi) informacijskog sustava.[41] Zadaća podatkovnog sloja je upravljanje resursima, odnosno u slučaju PS Cloud aplikacije, izvršavanje CRUD (kreiranje, dohvaćanje, ažuriranje, brisanje podataka) operacije nad bazom podataka.



Slika 8.1 Model troslojne arhitekture[42]

[41] Izvor: https://www.ibm.com/support/knowledgecenter/en/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/covr_3-tier.html, listopad 2017.
[42] Izvor: https://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/images/thtrcs.gif, listopad 2017.

8.2. ASP.NET MVC

ASP.NET je okruženje za razvoj web aplikacija razvijeno od strane Microsofta.[43] ASP.NET MVC je okruženje u kojem je implementirana MVC (model pogled kontroler) arhitektura. MVC predstavlja oblikovni obrazac koji se sastoji od tri komponente navedene i objašnjene u nastavku.[44]

- Model - Predstavlja podatke iz poslovne logike aplikacije, odnosno podatke koji će biti prikazani na pogledu.
- *View* (pogled) – Predstavlja korisničko sučelje, odnosno u slučaju web aplikacije, predstavlja web stranicu.
- *Controller* (kontroler) – Komponenta čija je namjena dohvaćanje modela i prosljeđivanje podatka za prikaz pogledu. Kontroler obrađuje zahtjeve korisnika te generira i šalje odgovor korisniku. Pod pojmovima zahtjev i odgovor podrazumijevaju se HTTP zahtjev (engl. *request*) i HTTP odgovor (engl. *response*)[45].

8.3. REST protokol i Web API 2

REST protokol omogućuje interoperabilnost među informacijskim sustavima i aplikacijama. Prijenos podataka je omogućen HTTP (protokol prijenosa informacija) protokolom (Lazić; Šimec, 2014). HTTP metode korištene u REST protokolu:

- *Get* (dohvati) - Dohvaćanje podataka.
- *Post* (objavi) - Izmjena podataka.
- *Put* (postavi) - Unos novog podatka.
- *Delete* (obriši) - Brisanje podataka.

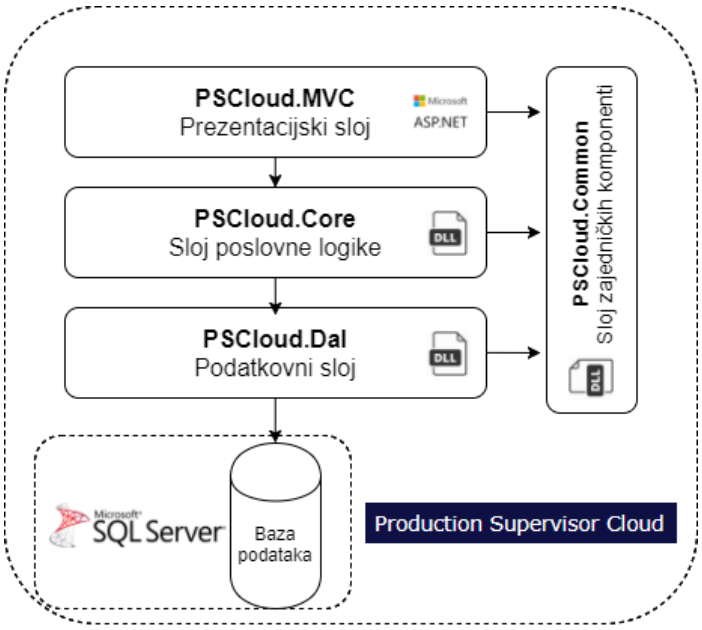
Web API 2 je aplikacijsko programsko sučelje unutar ASP.NET okruženja koje omogućuje komunikaciju s vanjskim aplikacijama putem REST protokola. Uvjet za ostvarivanje komunikacije je implementacija HTTP protokola od strane vanjske aplikacije.[46]

---

[43] Izvor: https://www.tutorialspoint.com/asp.net/asp.net_introduction.htm, listopad 2017.
[44] Izvor: https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx, listopad 2017.
[45] HTTP protokol za razmjenu podataka putem weba. Zasniva se na principu zahtjev-odgovor. (Izvor: https://tools.ietf.org/html/rfc2616, listopad 2017.)
[46] Izvor: https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api, listopad 2017.

8.4. Arhitektura web aplikacije

Web aplikacija je strukturirana prema modelu troslojne arhitekture. Sastoji se od četiri projekta (Visual studio projekt) čiji je međusobni odnos prikazan na slici 8.2. Početni projekt je PSCloud.MVC koji predstavlja prezentacijski sloj. Prezentacijski sloj ima pristup projektu PSCloud.Core koji predstavlja sloj poslovne logike. Projekt PSCloud.Core ima direktan pristup PSCloud.Dal projektu koji ima ulogu podatkovnog sloja. Podatkovni sloj predstavlja baza podataka. Četvrti projekt je PSCloud.Common kojem pristupaju svi slojevi aplikacije. PSCloud.MVC projekt je tipa ASP.NET MVC, dok su ostali projekti tipa biblioteka klasa.



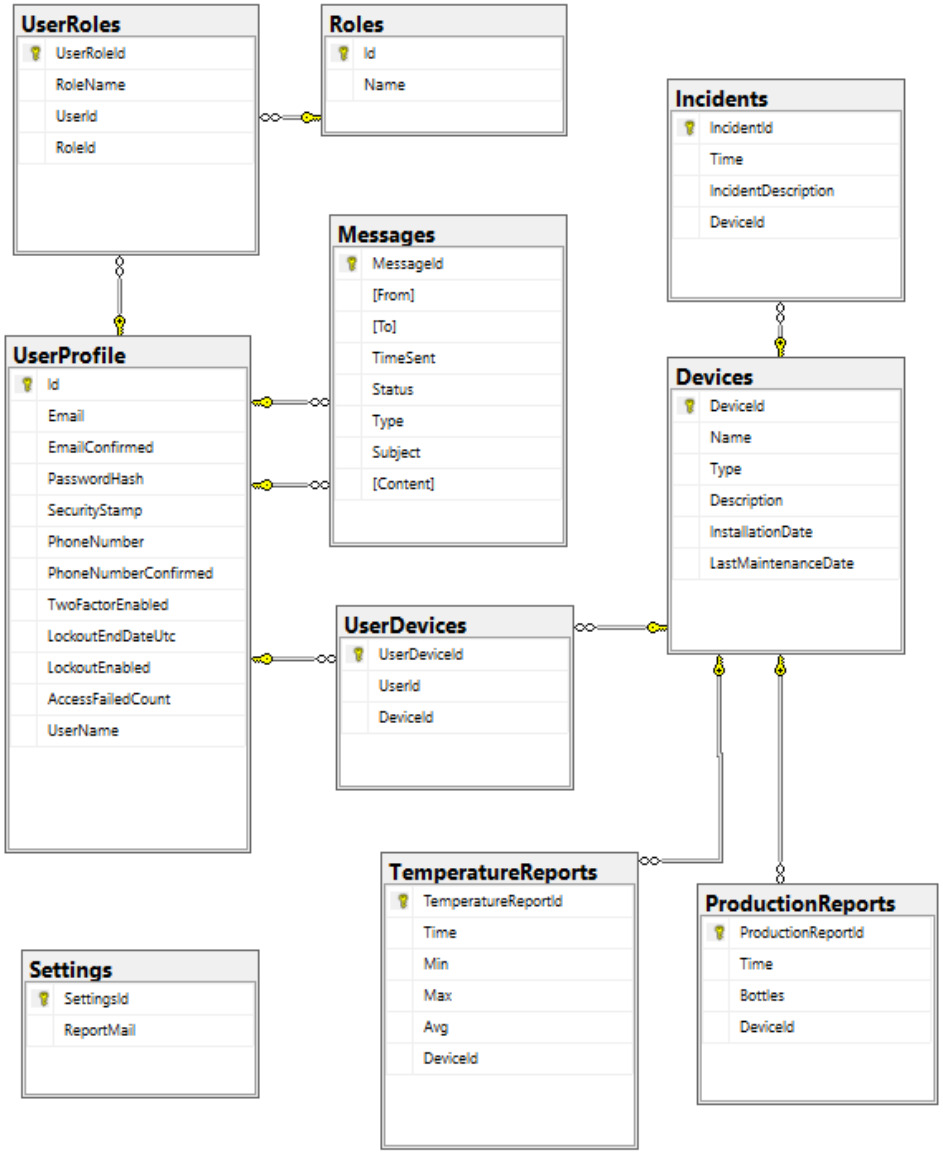
Slika 8.2 Arhitektura PS Cloud web aplikacije

8.5. Servis za primanje izvještaja

Servis za primanje izvještaja iz *desktop* aplikacije implementiran je na prezentacijskom sloju aplikacije tj. na PSCloud.MVC. HTTP metoda korištena za primanje izvještaja je *Post*. Servis za zaprimanje i obradu izvještaja se nalazi u sloju poslovne logike. U slučaju uspješno obavljene transakcije izvještaja, kontroler šalje odgovor s HTTP statusnim kodom (engl. *status code*) 200 OK (u redu). U slučaju neispravne transakcije, HTTP statusni kod je 204 *No content* (nema sadržaja). Svrha HTTP statusnog koda je povrat informacije o uspješnosti transakcije.

8.6. Baza podataka

Baza podataka je implementirana u Microsoftovoj tehnologiji SQL Server. SQL Server predstavlja relacijsku bazu podataka.[47] Relacijske baze podataka su one baze koje čuvaju podatke u tablicama. Veze među tablicama se nazivaju relacije te su definirane stranim ključevima (engl. *foreign key*).[48] Slika 8.3 prikazuje relacijski model baze podataka PS Cloud. Linije između tablica predstavljaju veze entiteta poslovne domene.



Slika 8.3 Relacijski model PS Cloud baze podataka

[47] Izvor: http://searchsqlserver.techtarget.com/definition/SQL-Server, listopad 2017.
[48] Izvor: http://searchsqlserver.techtarget.com/definition/relational-database, listopad 2017.

Zaključak

Svrha praktičnog dijela rada je ispitivanje mogućnosti implementacije sustava za nadzor i upravljanje Siemens S7-1200 PLC uređaja s ciljem pronalaska alternativnog rješenja postojećim SCADA sustavima. Implementacija je izvedena na modelu simulacije pivovare. Sve funkcionalnosti simulacije pivovare su uspješno implementirane u praktičnoj izvedbi, što ukazuje na mogućnost upotrebe Snap7 biblioteke otvorenog koda u izgradnji poslovnog sustava koji je alternativa postojećim SCADA rješenjima.

Nedostatak alternativnog rješenja implementiranog u radu je primoranost programera PLC uređaja na korištenje neoptimiziranih podatkovnih blokova kako bi se mogla ostvariti komunikacija s uređajem putem Snap7 API sučelja. Pitanje koje proizlazi iz ovog zaključka je koliko bi ova nepovoljna okolnost djelovala na performanse složenih industrijskih procesa automatiziranih S7-1200 uređajem.

Još jedan uočeni nedostatak implementiranog sustava je izvedba funkcionalnosti praćenja stanja minimalne, maksimalne i prosječne temperature. Prilikom odjave korisnika iz *desktop* aplikacije gube se podaci o stanju te prilikom nove prijave korisnika, podaci u bazi se izmjenjuju s novim nepotpunim podacima čije mjerenje započinje novom prijavom. Prilikom izgradnje ovakvih sustava potrebno je odgovornost čuvanja stanja implementirati u program PLC uređaja.

S aspekta procjene performansi *desktop* i web aplikacije, sustav se pokazao stabilnim te ispunjava sve funkcionalnosti poslovne domene. *Desktop* aplikacija prikazuje stanje proizvodnog procesa, čiji se prikaz ažurira svakih 100 ms, što daje privid praćenja stanja u stvarnom vremenu.

Rezultati ispitivanja mogućnosti implementacije sustava su pozitivni te potvrđuju tezu o mogućnosti izgradnje alternative postojećim SCADA sustavima. U budućoj izgradnji sustava u obzir treba uzeti ograničenje zbog primoranosti korištenja neoptimiziranih podatkovnih blokova. Na kraju dolazimo do vrijedne spoznaje da čuvanje podataka iz industrijskih procesa mora biti implementirano u programu PLC uređaja, dok *desktop* aplikacija mora služiti isključivo kao korisničko sučelje.

Popis kratica

PLC	*Programmable logic controller*	programabilni logički upravljač
PC	*Personal computer*	osobno računalo
LAN	*Local area network*	lokalna računalna mreža
SCADA	*Supervisory control and data acquisition*	nadzor upravljanja i prikupljanje podataka
HMI	*Human machine interface*	sučelje za interakciju korisnika i uređaja
CPU	*Central processing unit*	centralna procesorska jedinica
TIA	*Totally Integrated Automation*	potpuno integrirana automatizacija
RAM	*Random-access memory*	radna memorija
OB	*Organisation block*	organizacijski blok
FB	*Function block*	funkcijski blok
FC	*Function*	funkcija
DB	*Data block*	podatkovni blok
SCL	*Structured programming language*	strukturirani upravljački jezik
TCP/IP	*Transmission control Protocol*	protokol kontrole prijenosa podataka
ISO TCP	Industrial *Transmission control Protocol*	industrijski protokol kontrole prijenosa podataka
API	*Application programming interface*	aplikacijsko programsko sučelje
PDU	*Protocol Data Unit*	jedinica prijenosa podatkovnog protokola
CP	*Communication processor*	komunikacijski procesor
REST	*Representational state transfer*	reprezentacijski prijenos stanja
SMTP	*Simple Message Transfer Protocol*	protokol za slanje poruka
WPF	*Windows presentation foundation*	windows prezentacijske forme
MVVM	*Model view view model*	model, pogled, model za pogled
MVC	*Model View Controller*	model pogled kontroler
HTTP	*Hypertext transfer protocol*	protokol prijenosa informacija
CRUD	*Create, Retrieve, Update, Delete*	kreiranje, dohvaćanje, ažuriranje, brisanje
OK	*okay*	u redu

Popis slika

Slika 2.1 Model SCADA sustava 2

Slika 2.2 Primjer Siemensovog HMI panela 3

Slika 3.1 Osnovne komponente PLC uređaja 5

Slika 3.2 Ciklus rada PLC uređaja 6

Slika 3.4 Uređaj i periferni priključci 7

Slika 3.3 Postolje i utor (snimka zaslona iz programskog okruženja TIA portal) 8

Slika 3.5 Primjer funkcije koja koristi kombinaciju Ladder dijagrama i funkcijskih blokova 11

Slika 3.6 Optimizirani i neoptimizirani podatkovni blokovi 13

Slika 4.1 Enkapsulacija protokola S7, ISO-TCP, TCP/IP 15

Slika 4.2 Veza računalo klijent i PLC uređaja 15

Slika 4.3 Arhitektura klijent – poslužitelj 16

Slika 4.4 Zapis cjelobrojne varijable u big i little endian formatu 17

Slika 5.1 Prikaz stanja proizvodnog procesa 21

Slika 5.2 Prikaz trenutne temperature linijskim dijagramom 22

Slika 5.3 Prikaz podataka o povezanom uređaju 23

Slika 5.4 Prikaz korisničkog računa i odjava 23

Slika 5.5 Postavke uređaja 24

Slika 5.6 Pogled upravljačka ploča 25

Slika 5.7 Forma za slanje poruka i prikaz pristigle pošte 26

Slika 5.8 Prikaz izvještaja o proizvodnji za virtualni uređaj S7-1500 u kolovozu 2017. 27

Slika 5.9 Prikaz izvještaja o incidentima za virtualni uređaj S7-1500 u ožujku 2017. 28

Slika 5.10 Prikaz uređaja s autoriziranim korisnicima 29

Slika 5.11 Postavke kontakta za slanje obavijesti o incidentu 29

Slika 6.1 Elementi sustava pivovare 32
Slika 6.2 Postupak autentikacije 35
Slika 6.3 Pokretanje proizvodne linije 35
Slika 7.1 MVVM oblikovni obrazac 39
Slika 7.2 Arhitektura PS Operator aplikacije 39
Slika 8.1 Model troslojne arhitekture 41
Slika 8.2 Arhitektura PS Cloud web aplikacije 43
Slika 8.3 Relacijski model PS Cloud baze podataka 44

Popis tablica

Tablica 3.1 Izdvojene tehničke karakteristike CPU 1212C 7

Tablica 4.1 Podatkovni tipovi Siemens S7-1200 i odgovarajući podatkovni tipovi iz programskog jezika C# 18

Tablica 5.1 Korisničke role i pravo pristupa 30

Popis kôdova

Kôd 3.1 Skaliranje vrijednosti ultrazvučnog senzora jednadžbom pravca kroz dvije točke12

Kôd 4.1 Čitanje i konverzija varijable iz PLC uređaja..... 17

Literatura

[1] S7-1200 *system manual*, 04/2012, https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf, rujan 2017.

[2] https://inductiveautomation.com/what-is-scada, listopad 2017.

[3] http://w3.siemens.com/mcms/human-machine-interface/en/operator-devices/basic-hmi/basic-panels/Pages/Default.aspx, listopad 2017.

[4] https://www.amci.com/industrial-automation-resources/plc-automation-tutorials/what-plc, rujan 2017.

[5] GOLDBERG J., NG. JASON, *Practical Signal and Image Processing in Clinical Cardiology*: *Analog and Digital Signals* 9,15, ISBN 978-1-84882-515-4, 2010

[6] *Programming Guideline for* S7-1200/1500, 03/2014, http://www.industry.siemens.nl/automation/nl/nl/industriele-automatisering/industrial-automation/simatic-controller/modulaire-controllers/simatic-s7-1500/documents/81318674_programming_guideline_doku_v12_en.pdf, rujan 2017.

[7] BOLTON W. *Ladder and Functional Block Programming*, *Programmable Logic Controllers*, ISBN: 9780750681124, 2009, https://booksite.elsevier.com/9781856176217/appendices/01~Ch11.pdf , rujan 2017.

[8] *Structured Control Language* (SCL) for S7-300/S7-400 *Programming*, https://cache.industry.siemens.com/dl/files/188/1137188/att_27471/v1/SCLV4_e.pdf , rujan 2017.

[9] http://us.profinet.com/technology/profinet, rujan 2017.

[10] https://technet.microsoft.com/en-us/library/cc958821.aspx, listopad 2017.

[11] NARDELLA D. Snap7 *Reference manual*, rev. 6, 2.6.2015.

[12] https://www.gnu.org/licenses/lgpl-3.0.en.html, rujan 2017.

[13] https://www.gnu.org/software/libc, rujan 2017.

[14] https://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Data/endian.html, rujan 2017.

[15] KUNŠTEK Z., GRGIĆ S. Uvod u računalne mreže, Algebra, Zagreb, 2010

[16] http://www.cis.hr/www.edicija/SMTPprotokol.html, listopad 2017.

[17] https://mall.industry.siemens.com/mall/en/ww/Catalog/Product/6ES7231-5PA30-0XB0, listopad 2017.

[18] https://www.picotech.com/library/application-note/pt100-platinum-resistance-thermometers, listopad 2017.

[19] https://www.baumer.com/ch/en/functionality-and-technology-of-ultrasonic-sensors/a/know-how_function_ultrasonic-sensors, listopad 2017.

[20] https://www.baumer.com/ch/en/product-overview/distance-measurement/ultrasonic-sensors/miniature/cylindrical/unam-12u9914-s14/p/27806, rujan 2017.

[21] Ifm *electronic Photoelectric sensors*, http://www.ifm.com/obj/s200e.pdf, listopad 2017.

[22] Ifm *electronic Inductive sensors*, http://www.ifm.com/obj/s100e.pdf, listopad 2017.

[23] ANDRADE C., LIVERMORE S., MEYERS M., VAN VILET S. *Professional* WPF *Programming*:.NET *Development with the Windows® Presentation Foundation, Wiley Publishing, Inc*., ISBN: 978-0-470-04180-2, 2007

[24] https://msdn.microsoft.com/en-us/library/hh848246.aspx, listopad 2017.

[25] https://www.ibm.com/support/knowledgecenter/en/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/covr_3-tier.html, listopad 2017.

[26] https://www.tutorialspoint.com/asp.net/asp.net_introduction.htm, listopad 2017.

[27] https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx, listopad 2017.

[28] LAZIĆ D., ŠIMEC A. Pametna komunikacija na Internetu preko REST protokola, https://bib.irb.hr/datoteka/713376.2014_MIPRO_Pametna_komunikacija_na_Internetu_preko_REST_protokola.pdf, listopad 2017.

[29] https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api, listopad 2017.

[30] https://tools.ietf.org/html/rfc2616, listopad 2017.

[31] http://searchsqlserver.techtarget.com/definition/SQL-Server, listopad 2017.

[32] http://searchsqlserver.techtarget.com/definition/relational-database, listopad 2017.