

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Domen Stanič

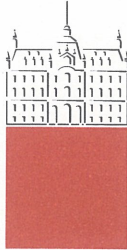
**Programska rešitev za poročanje v nadzornem sistemu
SCADA**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Igor Rožanc

Ljubljana, 2012



Št. naloge: 00223/2012

Datum: 03.04.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DOMEN STANIČ**

Naslov: **PROGRAMSKA REŠITEV ZA POROČANJE V NADZORNEM SISTEMU SCADA**

A SOFTWARE SOLUTION FOR REPORTING IN THE SCADA CONTROL SYSTEM

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Podjetje Salonit Anhovo d.d. uporablja za nadzor, spremljanje in vodenje svojega proizvodnega procesa nadzorni sistem SCADA. V diplomski nalogi najprej kratko predstavite ta sistem, nato pa zasnujete in izdelajte učinkovito programsko rešitev za nazorno poročanje o alarmih in drugih pomembnih dogodkih, ki jih SCADA zazna in beleži v svoji podatkovni bazi. V okviru tega preverite več rešitev, izbrano rešitev pa predstavite tako s tehničnega kot uporabniškega stališča.

Mentor:

viš. pred. dr. Igor Rožanc



Dekan:

prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU
diplomskega dela

Spodaj podpisani **Domen Stanič,**

z vpisno številko **63070323,**

sem avtor diplomskega dela z naslovom:

Programska rešitev za poročanje v nadzornem sistemu SCADA

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom (naziv, ime in priimek)
viš. pred. dr. Igorja Rožanca
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 5.7.2012

Podpis avtorja:

Zahvala

Zahvaljujem se mentorju viš. pred. dr. Igorju Rožancu za vso pomoč in strokovno vodenje pri izdelavi diplomske naloge. Zahvaljujem se tudi Mitji Berlotu, mentorju v podjetju, za vso tehnično pomoč in usmerjanje pri izdelavi aplikacije.

Posebno zahvalo pa namenjam staršem, ki so mi v času študija vedno stali ob strani ter me moralno podpirali.

Kazalo vsebine

Povzetek	1
Abstract.....	3
1 Uvod	5
2 Predstavitev problematike	7
2.1 Predstavitev podjetja.....	7
2.1.1 Predstavitev proizvodnega procesa	8
2.2 Nadzor proizvodnega sistema	9
2.2.1 Sistemi SCADA.....	10
2.2.2 Procesna podatkovna baza.....	11
2.3 Idejna rešitev problema.....	12
3 Analiza zahtev in izbire orodij.....	14
3.1 Zajem uporabniških zahtev	14
3.2 Orodja za interakcijo s podatkovno bazo.....	16
3.3 Urejevalniki kode.....	17
3.3.1 Notepad++	17
3.3.2 Adobe Dreamweaver	18
3.4 Možni načini in orodja za izpis podatkov in izdelavo poročil	19
3.4.1 Nadgradnja obstoječe aplikacije.....	19
3.4.2 Tiskanje podatkov iz html dokumenta.....	20
3.4.3 Izpis poročila s pomočjo orodja PHP Report Maker.....	21
3.4.4 Izdelava poročil s pomočjo orodja iReport.....	21
4 Predstavitev rešitve.....	23
4.1 Izbira orodja za izdelavo poročil.....	23
4.2 Vzpostavitev povezave PHP/Java Bridge.....	24
4.3 Postavitev podatkovne baze in povezave z SCADA nadzornim sistemom.....	26
4.4 Postopek izdelave aplikacije	28
4.4.1 Izdelava vmesnika	28
4.4.2 Programiranje logike aplikacije.....	30
4.4.3 Način izpisa podatkov	33
4.4.4 Uporaba predlog poročil v aplikaciji.....	38
4.5 Odprava ponavljajočih alarmov	39

4.6	Integracija aplikacije v SCADA nadzorni sistem.....	44
5	Pregled in uporaba aplikacije.....	47
6	Sklepne ugotovitve.....	51
	Literatura in viri.....	53

Kazalo slik

Slika 1:	Logotip podjetja Salonit Anhovo, d.d.....	7
Slika 2:	Shematski prikaz proizvodnega procesa v podjetju Salonit Anhovo, d.d.	9
Slika 3:	Shematski prikaz delovanja sistemov SCADA.	10
Slika 4:	Slika področja Gorilec iz nadzornega sistema SCADA.	11
Slika 5:	Prikaz podatkovnih blokov iz procesne podatkovne baze.....	12
Slika 6:	Shematski prikaz izbirnih področij vmesnika.....	15
Slika 7:	Smernice pri izpisu različnih tipov podatkov.	16
Slika 8:	Izvedba poizvedovanja v programu Microsoft SQL Server Management Studio Express.....	16
Slika 9:	Analiza podatkov z orodjem Excel.....	17
Slika 10:	Uporaba programa Notepad++.	18
Slika 11:	Vmesnik orodja Dreamweaver.	19
Slika 12:	Izpis alarmov s pomočjo obstoječe aplikacije.	20
Slika 13:	Izdelovanje poročila s pomočjo programa PHP Report Maker.	21
Slika 14:	Sestavljanje predloge poročila v orodju iReport Designer.	22
Slika 15:	Prikaz delovanja povezave med programskima jezika Java in PHP.	24
Slika 16:	Rezultat programa <code>zbirka.php</code>	26
Slika 17:	Diagram uporabe ODBC povezave.	27
Slika 18:	Nastavitve pošiljanja alarmov preko ODBC vmesnika.	27
Slika 19:	Prva verzija uporabniškega vmesnika.	29
Slika 20:	Izdelovanje novega uporabniškega vmesnika v orodju Dreamweaver.....	29
Slika 21:	Rezultat oblikovanja.	35
Slika 22:	Rezultat zgornjega oblikovanja glave izpisa alarmov.	36
Slika 23:	Uporaba orodja Report Query.	37
Slika 24:	Delovno območje programa iReport Designer in izdelava predloge poročil alarmov.	38
Slika 25:	Podvojeni alarmi.....	40
Slika 26:	Prikaz odpravljenih alarmov.....	41
Slika 27:	Prikaz nepotrjenih alarmov.....	42
Slika 28:	Prikaz vseh alarmov.....	43
Slika 29:	Diagram delovanje aplikacije.	43
Slika 30:	Uvodna stran nadzornega sistema SCADA.....	44
Slika 31:	VBA urejevalnik.....	45
Slika 32:	Prikaz delovanje aplikacije v nadzornem sistemu.	45
Slika 33:	Končna verzija uporabniškega vmesnika.	47

Slika 34: Izbira področja v vmesniku.	48
Slika 35: Izpis alarmov.	49
Slika 36: Prikaz poročila alarmov v pdf dokumentu.	49
Slika 37: Prikaz poročila alarmov v html dokumentu.	50

Povzetek

V diplomski nalogi prikazujemo rešitev problematike zajemanja in prikazovanja podatkov iz proizvodnega procesa podjetja Salonit Anhovo, d.d.. Glede na uporabniške zahteve smo izdelali orodje, ki uporabniku omogoča učinkovit pregled podatkov in izdelavo poročil. Osredotočili smo se na izdelavo takšnega tipa aplikacije, ki omogoča vgraditev v nadzorni sistem SCADA. SCADA je programsko orodje, ki služi za nadzor, spremljanje in vodenje proizvodnega procesa v podjetju. Brez takšnega orodja bi si težko zamislili obvladovanje tako velikega sklopa naprav, kot je proizvodna linija cementarne. Ker aplikacija pripomore pri lažjemu obvladovanju procesa, jo je smiselno vgraditi v nadzorni sistem SCADA.

V prvem delu naloge smo predstavili podjetje, nadzorni sistem SCADA in problematiko, s katero se bomo v nadaljevanju srečali. Analizirali smo podatke iz podatkovne baze, nadaljevali z zajemanjem uporabniških zahtev ter predstavitev orodij, ki smo jih uporabljali. Po skrbnem premisleku in načrtovanju smo v drugem delu naloge aplikacijo tudi izdelali. Najprej smo poskrbeli za pravilno delovanje podatkovne baze, ki služi za shranjevanje podatkov iz nadzornega sistema in predstavlja vir informacij za aplikacijo, nato pa smo podrobno predstavili izdelavo aplikacije. Med programiranjem smo naleteli na mnogo ovir, ki smo jih tudi predstavili. Po končani izdelavi smo aplikacijo testirali in predstavili njeno delovanje. Na koncu naloge smo predstavili sklepne ugotovitve.

Ključne besede: spletna aplikacija, relacijska podatkovna baza, nadzorni sistem SCADA, izdelava poročil.

Abstract

In this thesis we present a solution for the problem of collecting and presenting data in the manufacturing process in Salonit Anhovo, d.d.. According to user requirements we developed a tool that allows to users to effectively review the data and generate reports. We focused on constructing a suitable type of application that enables integration into the SCADA control system. SCADA is a software tool that is used for controlling, monitoring and managing the production process within the company. Without such tools it is hard to imagine managing such a large set of devices, built into the cement plant production line. Since the application helps in facilitating the production process, it is wise to integrate it into the SCADA control system.

The first part of the thesis is devoted to the presentation of the company, the SCADA control system, and the main problem which we will encounter further on with. We analyzed the data from the database, gathered the user requirements and presented the tools, we used. After careful consideration and planning, we devoted our time to the making of the application. The constructing of the application was the second part of the thesis. The first task was to achieve the proper functioning of the database that is used to store data from the control system which is the source of information for the application. With special emphasis we presented the details of making applications. During the programming we encountered many problems, for which we found and presented suitable solutions. When we completed the application, we tested and presented its performance. The last chapter of the thesis is devoted to the presentation of the final findings.

Keywords: web application, relation database, SCADA control system, production of reports.

1 Uvod

V proizvodnem obratu cementarne Salanit Anhovo, d.d. se nahaja veliko naprav. Vsaka naprava potrebuje za svoje delovanje izpolnjene določene pogoje. Pogosto se zgodi, da pogoji niso izpolnjeni in naprava neha delovati. Pri tem se napaka in vzrok napake beležita v nadzornem sistemu SCADA. Pri širjenju proizvodnje se je število naprav povečalo, posledično pa tudi število napak oziroma alarmov naprav. Zaradi tega je prišlo do oteženega diagnosticiranja nepravilnega ali prekinjenega delovanja naprav. Celotna proizvodnja cementa je razdeljena na več sklopov. Vsak računalnik, ki je v sklopu nadzornega sistema, pa beleži alarme naprav, ki mu pripadajo. To pripelje tudi do decentralizacije beleženja alarmov. Posledica teh težav je otežen pregled zgodovine alarmov in dogodkov. Rešitev je bila postavitvev podatkovne baze, ki nam omogoča centralno zbiranje podatkov. Ker je podatkovna baza brez primerne orodja za prikaz podatkov neuporabna, smo v okviru diplomske naloge izdelali aplikacijo, ki omogoča prikaz in filtriranje podatkov o alarmih, dogodkih in ponovitvah alarmov.

Glavni namen diplomske naloge je prikazati, na kakšen način bi lahko takšen sistem izdelali. Naša naloga je bila izbira primernih orodij, zajem uporabniških zahteve, postavitvev okolja za shranjevanje podatkov ter izdelava aplikacije. Da bi prikazali kar se da kvalitetne informacije, smo morali podatke iz podatkovne baze tudi skrbno analizirati. Ker bi aplikacija lahko bila del nadzornega sistema, smo jo razvili na takšen način, da bi jo bilo mogoče vključiti v nadzorni sistem SCADA. Zaradi boljše dostopnosti in integracije z nadzornim sistemom smo si aplikacijo zamislili kot spletno storitev. Za razvoj aplikacije smo uporabljali različne tehnologije. Eden glavnih problemov je bil način izpisa podatkov, da bodo ti primerni za tiskanje. Ker je html dokument neprimeren za tiskanje, smo za generiranje poročil morali poiskati primerno orodje.

Začeli bomo s predstavitvijo problematike, kjer bomo najprej predstavili podjetje, nadzorni sistem SCADA in bralca spoznali s problematiko. V naslednjem poglavju bomo podrobneje analizirali primerna orodja. Tukaj se bomo prvič srečali z problemom primerne izpisa za tiskanje. Analizirali bomo nekaj orodij za izdelavo poročil in predstavili njihove prednosti in slabosti. Sledi poglavje, v katerem bomo podrobneje predstavili rešitev. Pogledali si bomo postavitvev podatkovne baze, kreiranje tabel in vzpostavitev povezave med podatkovno bazo in nadzornim sistemom SCADA. Poglobili se bomo tudi v izdelavo same aplikacije, izdelavo uporabniškega vmesnika ter predstavili logiko aplikacije, ki teče v ozadju. Izpostavili bomo glavne težave, na katere smo naleteli, in predstavili rešitve zanje. V poglavju Pregled in uporaba aplikacije bomo predstavili izgled aplikacije in njene funkcionalnosti. V zadnjem poglavju pa bomo namenili nekaj besed sklepnim ugotovitvam.

2 Predstavitev problematike

V poglavju predstavitev problematike bomo predstavili podjetje in njegov proizvodni proces. Srečali se bomo s sistemom SCADA, ki ga bomo podrobneje pojasnili v podpoglavju Nadzor proizvodnega sistema. Na koncu pa bomo predstavili problem, ki smo ga v nalogi reševali, ter predstavili idejno rešitev.

2.1 Predstavitev podjetja

Salonit Anhovo, d.d. (v nadaljevanju Salonit Anhovo) je proizvajalec in dobavitelj različnih tipov cementa [1]. Zgodovina Salonita Anhovo sega v leto 1921, ko je bila ustanovljena družba Cementi Isonzo S.p.A.. Sedaj je znan po znanjih na področju proizvodnje, trženja in aplikacije kakovostnih cementov ter cementnih kompozitov v slovenskem in širšem evropskem prostoru. V Salonitu Anhovo ohranjajo in krepijo status vodilnega slovenskega proizvajalca cementa, z bogato, več kot 90 letno zgodovino in tradicijo. Uporabljajo naravne vire surovin in danosti, ob sočasni skrbi za okolje in prostor, v katerem poslujejo in živijo. Njihovi proizvodi so visoke kakovosti, skladni z veljavno slovensko in evropsko tehnično regulativo. Gradbeni materiali so prilagojeni sodobnim načinom gradnje in sistemom uporabe ter so okolju in zdravju prijazni. Salonit Anhovo je tudi med prvimi podjetji na področju industrije gradbenih materialov v Sloveniji, ki je pridobil certifikat kakovosti ISO 9001 in certifikat okoljske odgovornosti ISO 14001. Na sliki 1 je prikazan logotip podjetja Salonit Anhovo.



Slika 1: Logotip podjetja Salonit Anhovo, d.d.

Salonit Anhovo, se je oblikoval v skupino Salonit, ki je v Sloveniji center znanj in izkušenj na področju proizvodnje, razvoja, aplikacije in oskrbovanja porabnikov z visokokakovostnimi, okolju in zdravju prijaznimi gradbenimi materiali na osnovi mineralnih surovin. Skupina Salonit, katera je sestavljena iz desetih družb, je vodilni slovenski proizvajalec v panogi industrije gradbenega materiala ter vodilni dobavitelj proizvodov za gradbeno industrijo in druge porabnike. Glavni cilj skupine Salonit je stalno iskanje in izboljševanje medsebojnih sinergij, doseganje optimalne organiziranosti procesov ter optimalne alokacije in izrabe vseh virov (materialnih, finančnih, energetskih itd.). Na prvem mestu pa so znanja in veščine, razvoj zaposlenih ter razvoj karier. Sonaravni razvoj, visoko postavljeni okoljski standardi, izboljšave tehnologij in vseh poslovnih procesov kažejo na dosledno spoštovanje teh vrednot, tako v delniški družbi kot družbah v skupini Salonit, ki s svojimi dejanji in načrti tudi sledijo obvladujoči družbi.

Proizvodna veriga skupine Salanit obsega ponudbo osnovnih (kot so kamnolomski agregati in peski, cementi, betonski izdelki, apno, malte, apnene barve itd.) ter specialnih in drugih gradbenih proizvodov, ki skupaj s programi pridruženih družb, predvsem s področja betonov, vlaknocementnih plošč in specialnih proizvodov, oblikujejo široko ponudbo za gradbeno in drugo industrijo. Poleg proizvodov nekatere izmed družb v skupini Salanit nudijo tudi storitve vzdrževanja in servisiranja matični družbi ter tudi zunanjim poslovnim partnerjem.

2.1.1 Predstavitev proizvodnega procesa

Naša aplikacija bo omogočala lažji nadzor proizvodnega procesa (slika 2), zato bomo nekaj besed namenili opisu le tega. Bralcu bomo tako lažje predstavili okolje iz katerega aplikacija črpa informacije [2].

Osnovna dejavnost podjetja Salanit Anhovo je proizvodnja različnih tipov cementa za različne namene gradnje. Poglavitna surovina za proizvodnjo cementa je naravni lapor, kateremu zaradi omejenih virov dodajamo apnene in glinene komponente. Te surovine se pridobijo v lastnem kamnolomu ob cementarni. Sledi drobljenje in homogeniziranje surovin. Tako pripravljen material se nato skladišči v predhomogenizacijski hali, kjer sta vedno dva kupa; enega se uporablja za namene proizvodnje, drugi je v pripravi. Po transportnem traku potuje pripravljena mešanica v mline surovin, kjer se zmelje v laporno moko. Laporna moka nato nadaljuje pot skozi izmenjevalnik toplote, kjer dimni plini material predhodno segrejejo. Material nato potuje v peč, kjer se speče v klinker. Klinker se z različnimi dodatki (kot so sadra, žlindra, pucolani, ipd.), zmelje v sivo zelen prah – cement. Cement se nato skladišči v silosih. Del cementa se v avtomatski pakirni liniji zapakira v vreče po 25 kg. Večji del skladiščnih količin pa se odpremi v razsutem stanju s kamionskimi oziroma vagonскими cisternami. Poleg kamionskega in železniškega prevoza se (predvsem za namene izvoza na Bližnji Vzhod) uporablja tudi ladijski transport.

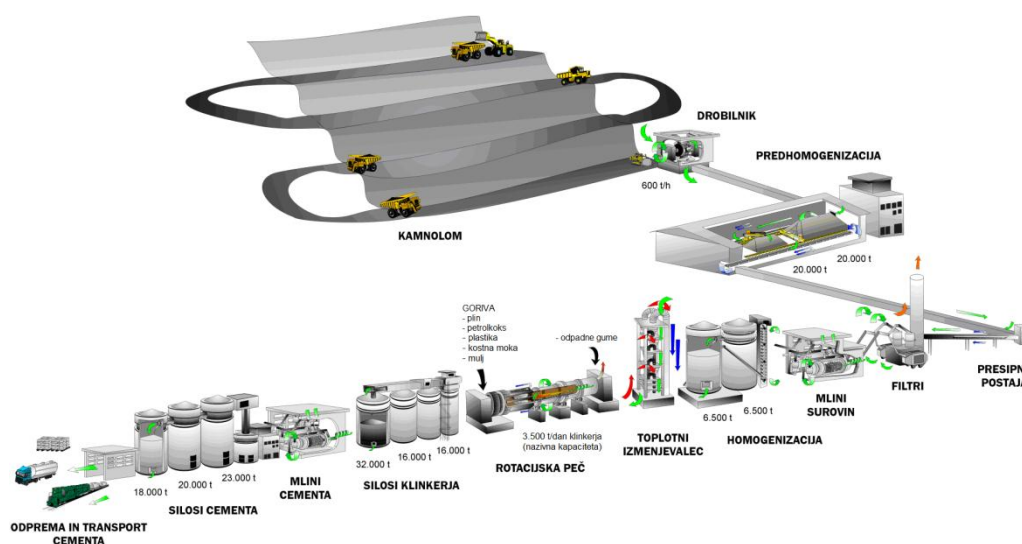
Z realizacijo zadnjega investicijskega ciklusa je postala cementarna ena najsodobnejših v evropskem merilu. Na liniji za proizvodnjo klinkerja bo v bodočnosti možno proizvajati maksimalno 3.500 ton klinkerja dnevno, kar pomeni preko 4.000 ton cementa dnevno. Trenutno pa proizvajajo s kapaciteto 3.100 ton klinkerja dnevno [3].

Eden najpomembnejših ciljev Salonita Anhovo je povečevanje energetske učinkovitosti. Energetika je pomembno področje, saj je proizvodnja cementa proces z visoko porabo energije. V procesu poteka proizvodnja cementnega klinkerja v rotacijski peči s ciklonskim izmenjevalnikom toplote. Toplotna energija predstavlja okrog 85% vse porabe energije v proizvodnji cementa. Zaradi te specifik je energetiki posvečena posebna pozornost. Toploto, potrebno za pečenje, dovajamo v sistem preko glavnega gorilnika ter preko doziranja na izmenjevalniku toplote, ki ima za to posebej pripravljena kurišča. Istočasno je mogoče uporabljati različna goriva, ki morajo biti posebej pripravljena in prilagojena sistemu za

doziranje. Izvedene tehnološke posodobitve so bistveno izboljšanje tudi energetske učinkovitost procesa [9].

Primerno moramo obvladovati vse odpadke, ki jih uporabljamo kot alternativna goriva. Cementne peči so primeren medij za sosežig izbranih vrst odpadkov, za katere je zahteva, da nimajo škodljivih primesi (npr. za emisije, proces ali proizvode) ter imajo dovolj visoko kalorično vrednost. Posebnih tehnologij za sosežig odpadkov ni, saj v cementnih pečeh uporabljamo alternativna goriva (odpadna olja, odpadne gume, plastika itd.) na enak način kot fosilna (zemeljski plin, premog in mazut). Običajno je uporaba v cementarni povezana le s prilagoditvijo sistemov doziranja in skladiščenja teh goriv.

Tehnologija proizvodnje cementa v Salonitu Anhovo je prilagojena načelom najboljših razpoložljivih tehnik in z njo dosegamo takšne oziroma nižje vplive na okolje, kot jih zahteva zakonodaja ter okoljevarstveno (IPPC) dovoljenje. Izbira goriv in njihova kakovost mora biti vedno v okviru teh zahtev. V ta namen je potrebno zagotavljati ustrezen nadzor nad uporabo vseh (primarnih in alternativnih) goriv.

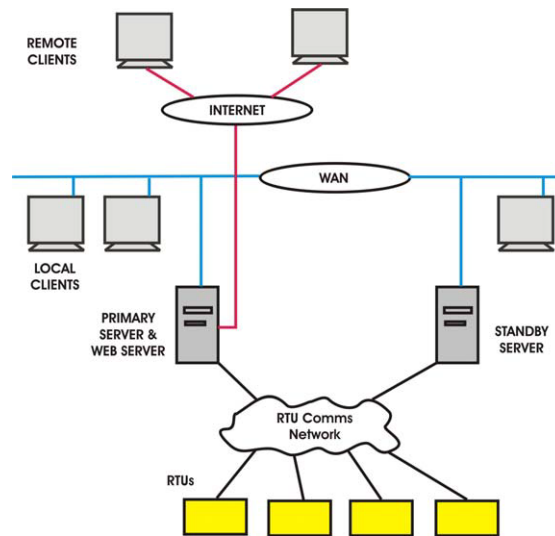


Slika 2: Shematski prikaz proizvodnega procesa v podjetju Salonit Anhovo, d.d.

2.2 Nadzor proizvodnega sistema

V zadnjih treh desetletjih se je v avtomatizaciji proizvodnje mnogo spremenilo. To je posledica stalno naraščajočih potreb po bolj kompleksnem in natančnem vodenju proizvodnih procesov. Razvoj se je začel s pojavom mikroelektronskih komponent (z iznajdbo polprevodnikov ter uporabo tranzistorja), nato pa s pojavom in s stalno izpopolnitvijo mikroprocesorske tehnike nadaljeval do nastanka danes znanih programirljivih logičnih krmilnikov PLK (ang. Programmable Logic Controller - PLC). Nadzor nad procesom je prevzela posebna programska oprema SCADA, ki skupaj z izjemno zmogljivimi PLK

omogoča popoln nadzor nad avtomatizirano proizvodnjo. SCADA je običajno nameščena na računalnikih ali strežnikih v kontrolni sobi, kjer imamo nadzor nad celotnim procesom, samo krmiljenje pa izvajajo PLC nameščeni v proizvodnji.



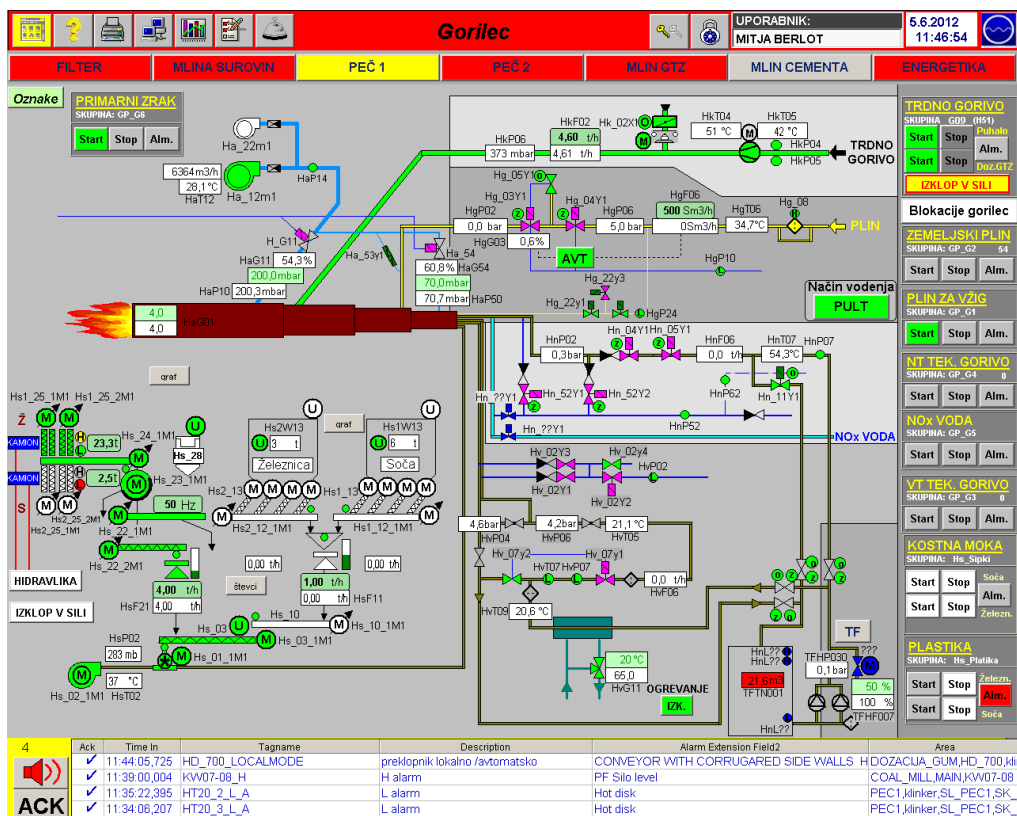
Slika 3: Shematski prikaz delovanja sistemov SCADA.

2.2.1 Sistemi SCADA

Sistemi SCADA (ang. Supervisory Control And Data Acquisition systems) [10] se uporabljajo za spremljanje in nadzor naprav, ki se tipično nahajajo v drugem okolju. Sistem nadzoruje delovanje krmilnih enot in preko tega nadzora vpliva na proizvodni proces (slika 3). Glavna naloga sistemov SCADA je prenos trenutnih informacij iz/do različnih virov ali lokacij, pri tem pa mora zagotavljati pravilnost podatkov in ustrezen cikel posodabljanja podatkov. V podjetju katerega namen je proizvodnje izdelkov je to tipično nadzor, spremljanje in vodenje proizvodnje. Sistemi SCADA omogočajo veliko odprtost sistema, saj se lahko podatki zajemajo iz različnih virov, ustrezno obdelajo in nato koristijo kot navodila za izvedbo proizvodnega procesa.

Na tržišču sistemov SCADA sta prisotni dve glavni skupini sistemov: splošno uporabni sistemi in namensko uporabni sistemi. Razlika med tema dvema skupinama je predvsem v področjih uporabe sistemov SCADA in povezanosti s krmilnimi enotami. V podjetju Saloniit Anhovo uporabljajo programsko rešitev Proficy iFIX 3.5 [11] proizvajalca GE Intelligent Platforms. Nadzorni sistem posreduje le spremembe parametrov, omejitve in nastavitve, nima pa funkcije samega krmiljenja in regulacije oziroma so le te omejene. Za shranjevanje zgodovine meritev podjetje uporablja strežnik Proficy Historian verzije 3.1 prav tako proizvajalca GE Intelligent Platforms. Shranjene meritve se uporabljajo predvsem za izrisovanje grafov (ang. trend). Grafe se prikazuje v slikah nadzornega sistema s pomočjo vmesnika in programske opreme, ki jo zagotavlja proizvajalec.

Za nemoteno delovanje nadzornega sistema skrbi služba za avtomatizacijo proizvodnih procesov. Naloga službe je oskrba industrijskega omrežja, strojne in programske opreme, ki zajema del krmilniškega ter nadzornega sistema. Služba za avtomatizacijo strmi k izgradnji takšnega sistema, ki bo proizvajal izdelke neprekinjeno in z največjo možno kakovostjo, obenem pa bo za to potrebno čim manj osebja in inventarja. Za zgradbo takega sistema je potrebno neposredno odločanje in stalno optimiziranje poteka proizvodnje, kar pa zaradi človeške počasnosti lahko izvaja le stroj. V zadnjem času je to računalnik in govorimo o računalniško podprti proizvodnji (ang. Computer Integrated Manufacturing - CIM). Model CIM je najsodobnejše orodje v vodenju proizvodnje in se danes uporablja v vseh razvitih industrijskih panogah. Na sliki 4 je prikaz nadzornega okna področja Gorilec v sistemu SCADA.



Slika 4: Slika področja Gorilec iz nadzornega sistema SCADA.

2.2.2 Procesna podatkovna baza

Procesna podatkovna baza (ang. Process Database - PDB) ima ključno nalogo pri zagotavljanju pravilnosti podatkov v sistemu SCADA [4]. PDB je podatkovna baza, ki v realnem času hrani podatke iz proizvodnega procesa. Namenjena je komuniciranju med krmilnikom in grafično postajo oziroma odjemalcem. Je osnova nadzornega dela, ki skrbi za proženje alarmov in ustvarjanje grafov. Vsebina podatkovne baze se nenehno spreminja, saj je vsaka sprememba na krmilniku vidna tudi v procesni podatkovni bazi. Tekom izdelave

aplikacije smo se pogosto srečevali z procesno podatkovno bazo, saj je predstavljala vir vseh podatkov, ki smo jih mi s pomočjo aplikacije prikazovali. Na sliki 5 je prikazano orodje Proficy iFIX Database Manager, ki omogoča prikaz in urejanje podatkovnih blokov procesne podatkovne baze.

Tag Name	Type	Description	Scan Time	I/O Dev	I/O Addr	Curr Value	Alarm Area 1	Alarm Area 4	Alarm
1 CR_10_A5_OUTENG	AI	CR.10 -A5. Temperatura transportn	1	SL4	CR.DB:105.0	????	PAK1		
2 CR_10_A6_OUTENG	AI	CR.10 -A6. Tlak transportnega zrak	1	SL4	CR.DB:105.64	????	PAK1		
3 CR_10_B30_OUTENG	AI	CR.10 -B30. Temperatura v silosu-	1	SL4	CR.DB:105.32	????	PAK1		
4 CR_10_B31_OUTENG	AI	CR.10 -B31. Temperatura v silosu-	1	SL4	CR.DB:105.36	????	PAK1		
5 CR_10_D11_STATUS	AI	Disoccont tehtanje silosa	1	SL4	CR.DB:123.16	????	PAK1		
6 CR_10_F60_STATUS	AI	Gotwald	1	SL4	CR.DB:123.18	????	PAK1		
7 CR_10_STATUS	AI	Status skupine silosa	1	SL4	CR.DB:123.106	????	PAK1		
8 CR_10_TEZA_OUTENG	AI	teža silosa gotwald - kg	1	SL4	CR.DB:105.36	????	PAK1		
9 CR_11_ABSBATCHSP_OUTENG	AI	Količina v prehopperju [kg]	1	SL4	CR.DB:105.324	????	PAK1		
10 CR_11_ABSOLUTESP_OUTENG	AI	Setpoint dozacije - kg/h	1	SL4	CR.DB:105.368	????	PAK1		
11 CR_11_ACTUALSP_OUTENG	AI	Trenutni setpoint - kg/h	1	SL4	CR.DB:105.392	????	PAK1		
12 CR_11_BATCHFEEDRA_OUTENG	AI	Števec 1 [kg]	1	SL4	CR.DB:105.362	????	PAK1		
13 CR_11_BATCHRESAM_OUTENG	AI	Števec 2 [kg]	1	SL4	CR.DB:105.356	????	PAK1		
14 CR_11_D11_STATUS	AI	Disoccont tehtanje dozime polževke	1	SL4	CR.DB:123.26	????	PAK1		
15 CR_11_FEEDRATE_OUTENG	AI	Trenutna količina doziranja - kg/h	1	SL4	CR.DB:105.376	????	PAK1		
16 CR_11_FILLWEIGHT_OUTENG	AI	Količina v prehopperju - kg	1	SL4	CR.DB:105.384	????	PAK1		
17 CR_11_M50_STATUS	AI	Dozime polževka	1	SL4	CR.DB:123.20	????	PAK1		
18 CR_11_M51_STATUS	AI	Dozime polževka	1	SL4	CR.DB:123.22	????	PAK1		
19 CR_11_STEVECVCERAJ_OUTENG	AI	Števec doziranega materiala včere	1	SL4	CR.DB:105.476	????	PAK1		
20 CR_11_SURGEOPPER_OUTENG	AI	Zdozirana količina šarže [kg]	1	SL4	CR.DB:105.340	????	PAK1		
21 CR_11_TOTALIZER1_OUTENG	AI	Števec 1 - kg	1	SL4	CR.DB:105.380	????	PAK1		
22 CR_11_TOTALIZER2_OUTENG	AI	Števec 2 - kg	1	SL4	CR.DB:105.396	????	PAK1		
23 CR_12_ABSBATCHSP_OUTENG	AI	Rezerve	1	SL4	CR.DB:105.364	????	PAK1		
24 CR_12_ABSOLUTESP_OUTENG	AI	Setpoint dozacije - kg/h	1	SL4	CR.DB:105.408	????	PAK1		
25 CR_12_ACTUALSP_OUTENG	AI	Trenutni setpoint - kg/h	1	SL4	CR.DB:105.432	????	PAK1		
26 CR_12_BATCHFEEDRA_OUTENG	AI	Trenutni setpoint [kg/h]	1	SL4	CR.DB:105.392	????	PAK1		
27 CR_12_BATCHRESAM_OUTENG	AI	Števec 2 [kg]	1	SL4	CR.DB:105.396	????	PAK1		
28 CR_12_D11_STATUS	AI	Disoccont tehtanje dozime polževke	1	SL4	CR.DB:123.28	????	PAK1		
29 CR_12_FEEDRATE_OUTENG	AI	Trenutna količina doziranja - kg/h	1	SL4	CR.DB:105.416	????	PAK1		
30 CR_12_FILLWEIGHT_OUTENG	AI	Količina v prehopperju - kg	1	SL4	CR.DB:105.424	????	PAK1		
31 CR_12_M50_STATUS	AI	Dozime polževka	1	SL4	CR.DB:123.24	????	PAK1		
32 CR_12_STEVECVCERAJ_OUTENG	AI	Števec doziranega materiala včere	1	SL4	CR.DB:105.480	????	PAK1		
33 CR_12_SURGEOPPER_OUTENG	AI	Števec 1 [kg]	1	SL4	CR.DB:105.380	????	PAK1		
34 CR_12_TOTALIZER1_OUTENG	AI	Števec 1 - kg	1	SL4	CR.DB:105.420	????	PAK1		
35 CR_12_TOTALIZER2_OUTENG	AI	Števec 2 - kg	1	SL4	CR.DB:105.436	????	PAK1		
36 CR_51C_ABSBATCHSP_OUTENG	AI	Setpoint šarže [kg]	1	SL4	CR.DB:105.72	????	PAK1		
37 CR_51C_ABSOLUTESP_OUTENG	AI	Setpoint dozacije - kg/h	1	SL4	CR.DB:105.108	????	PAK1		
38 CR_51C_ACTUALSP_OUTENG	AI	Trenutni setpoint - kg/h	1	SL4	CR.DB:105.132	????	PAK1		
39 CR_51C_BATCHFEEDRA_OUTENG	AI	Zdozirana količina šarže [kg]	1	SL4	CR.DB:105.100	????	PAK1		
40 CR_51C_BATCHRESAM_OUTENG	AI	Preostanek šarže [kg]	1	SL4	CR.DB:105.104	????	PAK1		
41 CR_51C_FEEDRATE_OUTENG	AI	Trenutna količina doziranja - kg/h	1	SL4	CR.DB:105.116	????	PAK1		
42 CR_51C_FILLWEIGHT_OUTENG	AI	Količina v prehopperju - kg	1	SL4	CR.DB:105.124	????	PAK1		
43 CR_51C_STEVECVCERAJ_OUTENG	AI	Števec doziranega materiala včere	1	SL4	CR.DB:105.448	????	PAK1		
44 CR_51C_SURGEOPPER_OUTENG	AI	?	1	SL4	CR.DB:105.88	????	PAK1		
45 CR_51C_TOTALIZER1_OUTENG	AI	Števec 1 - kg	1	SL4	CR.DB:105.120	????	PAK1		
46 CR_51C_TOTALIZER2_OUTENG	AI	Števec 2 - kg	1	SL4	CR.DB:105.136	????	PAK1		

Slika 5: Prikaz podatkovnih blokov iz procesne podatkovne baze.

2.3 Idejna rešitev problema

Vsaka naprava, ki je pod okriljem nadzornega sistema, ob prekoračitvi določene vrednosti sproži alarm. Ker je naprav v proizvodnem procesu veliko in bo v bodoče število še raslo, je prišlo do velikega števila alarmov in dogodkov. Posledica velikega števila tovrstnih zapisov je veliko podatkov, ki uporabniku ne prinašajo uporabnega vpogleda. Zato želimo uporabniku dati orodje, ki mu bo omogočalo prikaz kvalitetnih informacij na pregleden način. Poleg velikega števila zapisov, imamo problem tudi pri podvajanju zapisa istega alarma. Ker je format podatkov, ki jih nadzorni sistem pošilja podatkovni bazi težko berljiv, je tudi težko izluščiti pravilne podatke, ki jih uporabnik zahteva.

Poleg samih alarmov imamo v sistemu tudi dogodke, ki nosijo informacijo o akciji, ki jo sproži operater na komandnem pultu. Taki dogodki so recimo pritiski na gumb (npr. zasilna zaustavitev naprave) ali sprememba vrednosti parametra neke naprave. Dogodki so enako pomembni kot alarmi, saj je lahko v določenih primerih njihova posledica alarm.

Prvi korak je podatke, ki so razpršeni po nadzornih računalnikih, hraniti v centralni podatkovni bazi. Podatki, ki smo jih črpali iz nadzornega sistema, ne obsegajo celotne proizvodnje podjetja, za namen predstavitve rešitve problema pa so popolnoma zadoščali. Pri izbiri podatkovne baze moramo upoštevati možno uporabo te na strežnikih podjetja. Po vzpostavitvi podatkovne baze smo načrtovali izdelavo aplikacijo, ki bo črpala podatke iz naše podatkovne baze in omogočala izdelavo poročil. Pri izdelavi smo morali upoštevati tudi možnost vgraditve aplikacije v nadzorni sistem, ter ostale želje uporabnika. Danes opažamo porast tako imenovanih oblačnih storitev, ki omogočajo dostopnost vsebin ne glede na uporabnikovo lokacijo. Tudi naša ideja je narediti aplikacijo v obliki spletne storitve. S tem bi omogočili enostavno vgradnjo v SCADA sistem ter lažjo dostopnost aplikacije uporabnikom v internem omrežju. Pri problemu izdelave poročil, smo možno rešitev videli v uporabi orodja, ki je namenjeno izdelovanju poročil.

3 Analiza zahtev in izbire orodij

V tem poglavju bomo zbrali uporabnikove zahteve glede aplikacije, uporabniškega vmesnika in oblike poročil. Predstavili bomo orodja, s katerimi smo se srečali in katera so nam olajšala delo. Pri orodjih za izdelavo poročil bomo predstavili prednosti in slabosti vsakega orodja in se glede na njihove značilnosti odločili za najbolj ustreznega.

3.1 Zajem uporabniških zahtev

Uporabniške zahteve obsegajo zahteve glede aplikacije in strojne opreme, uporabniškega vmesnika in izgleda poročil. Pri izgledu poročil mislimo predvsem na podatke, kateri bodo v poročilu predstavljeni in obliko poročila.

Aplikacija mora ustrezati naslednjim zahtevam:

- končna oblika aplikacije naj bo takšna, da bo omogočala integracijo z SCADA nadzornim sistemom,
- dostopnost aplikacije naj bo mogoča iz poljubnega računalnika v istem omrežju,
- aplikacija naj zahteva uporabo čim manj dodatne programske opreme na odjemalčevem računalniku,
- aplikacija naj uporablja Microsoftov SQL strežnik.

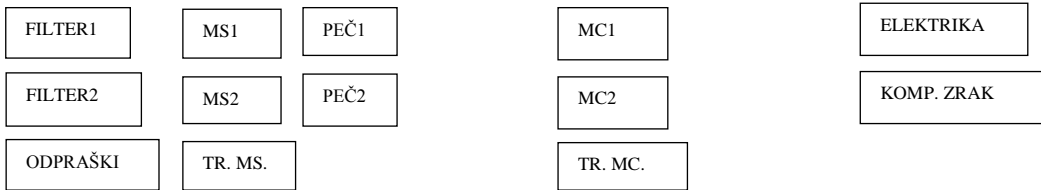
Uporabniški vmesnik naj omogoča:

- pri sklopu možnost izbire več elementov hkrati,
- barvanje izbranih elementov,
- zagon aplikacije z privzetimi vrednosti,
- izbiro med pripadajočimi področji, v kolikor uporabnik izbere en sklop,
- upoštevanje vseh pripadajočih področij, v kolikor je izbran več kot en sklop,
- postavitve izbirnih območij naj bo v vrstnem redu kot ga prikazuje slika 6.

1. Sklop



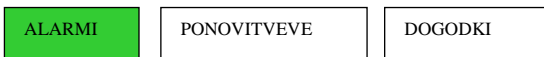
2. Področje



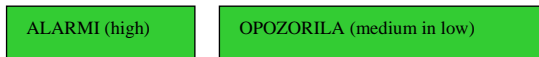
3. Izbira jezika



4. Izbira: alarmi, ponovitve alarmov ali dogodki



5. Izbira prioritete alarmov



6. Tip alarma



7. Vnos datuma in časa

8. Možnost dodatnega filtriranja

- a. po imenu alarma,
- b. po opisu napake,
- c. po opisu naprave.

Slika 6: Shematski prikaz izbirnih področij vmesnika.

Izpis naj omogoča:

- izpis alarmov, ki naj ne vsebuje duplikatov,
- izpis naj bo prilagojen za tiskanje,
- izpis naj bo pregleden,
- zaradi varčevanja z barvami naj se uporablja zgolj odtenke sive barve, a tudi te naj se uporablja le za pomembnejše dele poročila kot so glava ali noga,
- pri izpisu poročila naj se upošteva smernice, ki so prikazane na sliki 7.

1. Alarmi

Začetek		Konec		Alarm (tag)	Opis napake (opis 2)	Opis naprave (opis 1)
29.3.2012	15:51	29.3.2012	15:53	HT15_L_A	L alarm	Ciklon 5 - vrh
29.3.2012	15:51			VF10001_ETOTAL	Skupna napaka	Merilec pretoka zraka - skupno

2. Ponovitve alarmov

Ponovitve	Alarm	Opis
164	HP58_GRTDRV2_OILPH_A	Rešetke hladilnika S
70	HD_0_LACK_OF_TYRES_SEPARACION	LACK OF TYRES SEPARATION
		pomanjkanje gum za separacijo

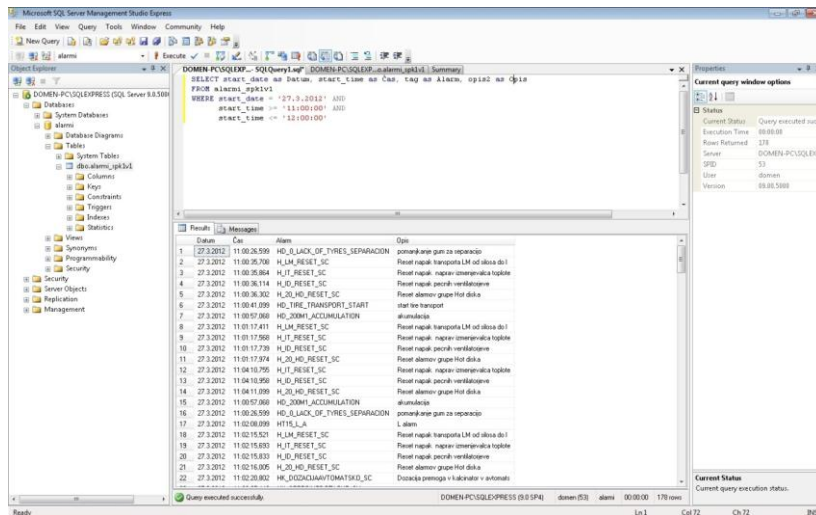
3. Dogodki

Pojavitev	Dogodek	
28.3.2012	13:57:07	Reset alarmov grupe Hot diska
28.3.2012	13:57:08	Reset napak transporta LM od silosa do I

Slika 7: Smernice pri izpisu različnih tipov podatkov.

3.2 Orodja za interakcijo s podatkovno bazo

Pri izdelavi aplikacije so nam veliko pomagala orodja za rokovanje z podatkovno bazo. Velikokrat smo morali pogledati določen zapis v bazi in preverjati točnost izpisa aplikacije. Za dodajanje uporabnikov in kreiranje tabel smo uporabljali orodje Microsoft SQL Server Management Studio Express [5], katero smo dobili v paketu s podatkovno bazo. Orodje ponuja uporabniku prijazen in pregleden vmesnik, ki omogoča uporabo vseh glavnih funkcionalnosti podatkovne baze. Microsoft SQL Server Management Studio smo uporabljali tudi za preverjanje pravilnosti poizvedb (slika 8).



Slika 8: Izvedba poizvedovanja v programu Microsoft SQL Server Management Studio Express.

V nekaterih primerih pa nam program Microsoft SQL Server Management Studio Express ni zadostoval našim potrebam. Pri pregledovanju večjega števila vrstic v podatkovni bazi, postane brskanje po rezultatih oteženo in počasno. Zato smo v ta namen uporabljali programsko orodje Microsoft Excel (v nadaljevanju Excel) [12]. V programu Excel je enostavneje pregledovati večje število podatkov, saj omogoča boljšo interakcijo s podatki in obilo funkcij, katere lahko uporabimo nad njimi. Po končanem delu, lahko rezultat tudi shranimo.

V našem primeru smo si s programom pomagali pri analiziranju shranjenih podatkov in iskanju vzorcev, katere smo nato uporabili pri prepoznavanju določenih tipov podatkov. S pomočjo orodja Excel smo ob analizi podatkov ugotovili, da različne različice klientov nadzornega sistema SCADA (primerjali smo različici 3.5 in 4.5) na različen način shranjujejo podatke. Nekateri tipi zapisov se pri novejši različici ne zapisujejo v podatkovno bazo, zato smo za vir shranjevanja uporabili starejšo, bolj preverjeno različico. Uporaba programa je prikazana na sliki 9.

last_date	last_time	tag	opis2	alm_status	msg_type	priority	area
		_OF_TYRES	pomanjkanje gum	CFN	ALARM	LOW	DOZACIJA_GUMAHD_0.klink
		ACCUMULATION	akumulacija	CFN	ALARM	LOW	DOZACIJA_GUMAHD_700.kh
		ACCUMULATION	akumulacija	OK	ALARM	LOW	DOZACIJA_GUMAHD_700.kh
		_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	OK	ALARM	LOW	DOZACIJA_GUMAHD_0.klink
		_OF_TYRES	pomanjkanje gum	OK	ALARM	LOW	DOZACIJA_GUMAHD_0.klink
		ICCOM_UPPER_SHAFT	zalagovnik zaprt	CFN	ALARM	LOW	DOZACIJA_GUMAHD_1300.k
		ICCOM_UPPER_SHAFT	zalagovnik zaprt	OK	ALARM	LOW	DOZACIJA_GUMAHD_1300.k
		_OF_TYRES	pomanjkanje gum	CFN	ALARM	LOW	DOZACIJA_GUMAHD_0.klink
		A	L.alarm	CFN	ALARM	LOW	PECLIKlinkerSL_PECCLSK_I2
		_OF_TYRES	pomanjkanje gum	OK	ALARM	LOW	DOZACIJA_GUMAHD_0.klink
		SOKA_GUMA	Previsoka guma na HD 435	CFN	ALARM	LOW	DOZACIJA_GUMAHD_435.kh
		_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	CFN	ALARM	LOW	DOZACIJA_GUMAHD_0.klink
		A	L.alarm	CFN	ALARM	LOW	PECLIKlinkerSL_PECCLSK_I2
		A	L.alarm	CFN	ALARM	LOW	PECLIKlinkerSL_PECCLSK_I2
		SRPSTART	[MS] Fik32.MS.HMICONTROL_MixLimeDirection_F_2 set to 1 by PECV9:IZMENA	OPERATOR	LOW	ALL	
		Group start		EVENT	LOW	ALL	
		IMS	[Fik32.MS.MixFeed_GRPSTART_F_CV set to 1 by PECV9:IZMENA	OPERATOR	LOW	ALL	
		_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	OK	ALARM	LOW	DOZACIJA_GUMAHD_0.klink
		Previsoka guma na HD 435		OK	ALARM	LOW	DOZACIJA_GUMAHD_435.kh
		IMS	[Fik32.MS.HMICONTROL_Mix1Res_F_CV set to 1 by PECV10:IZMENA	OPERATOR	LOW	ALL	
		IMS	[Fik32.MS.HMICONTROL_Lime2Res_F_CV set to 1 by PECV10:IZMENA	OPERATOR	LOW	ALL	
		IMS	[Fik32.MS.HMICONTROL_Fe1Res_F_CV set to 1 by PECV10:IZMENA	OPERATOR	LOW	ALL	
		IMS	[Fik32.MS.HMICONTROL_Fe1Res_F_CV set to 1 by PECV10:IZMENA	CFN	ALARM	LOW	DOZACIJA_GUMAHD_0.klink
		IMS	[Fik32.MS.HMICONTROL_Fe1Res_F_CV set to 1 by PECV10:IZMENA	OPERATOR	LOW	ALL	
		IMS	[Fik32.MS.HMICONTROL_AddDirection_F_1 set to 1 by PECV9:IZMENA	CFN	ALARM	LOW	DOZACIJA_GUMAHD_0.klink
		IMS	[Fik32.MS.HMICONTROL_AddDirection_F_1 set to 1 by PECV9:IZMENA	OPERATOR	LOW	ALL	
		ADDFFED_GRPSTART	Group start	EVENT	LOW	ALL	
		IMS	[Fik32.MS.AddFeed_GRPSTART_F_CV set to 1 by PECV9:IZMENA	OPERATOR	LOW	ALL	
		IMS	[Fik32.MS.HMICONTROL_MeatTosIsel_F_4 set to 1 by PECV10:IZMENA	OPERATOR	LOW	ALL	
		IMS	[Fik32.MS.HMICONTROL_MeatTosIsel_F_5 set to 1 by PECV10:IZMENA	OPERATOR	LOW	ALL	

Slika 9: Analiza podatkov z orodjem Excel.

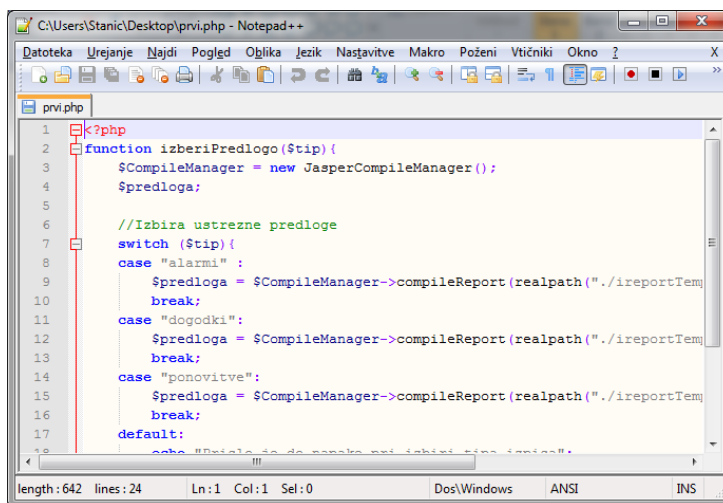
3.3 Urejevalniki kode

Za uspešno programiranje potrebujemo dobro razvojno okolje. Pri izdelavi aplikacije smo uporabljali dve različni orodji za urejevanje kode.

3.3.1 Notepad++

Prvo orodje je Notepad++ [13] (slika 10). Gre za izboljšano verzijo beležnice, ki omogoča barvanje kode, ter obilo dodatkov, ki jih lahko vključimo. Program pozna tudi veliko

programskih jezikov ter tako omogoča širok spekter uporabe. Novejše različice programa pa podpirajo tudi funkcijo samodejnega zaključevanja ukazov, ki programerju omogoča hitrejše programiranje. Orodje smo uporabljali predvsem zaradi njegove enostavnosti in učinkovitosti ter brezplačne uporabe. Na sliki 10 vidimo urejanje kode v programu Notepad++.

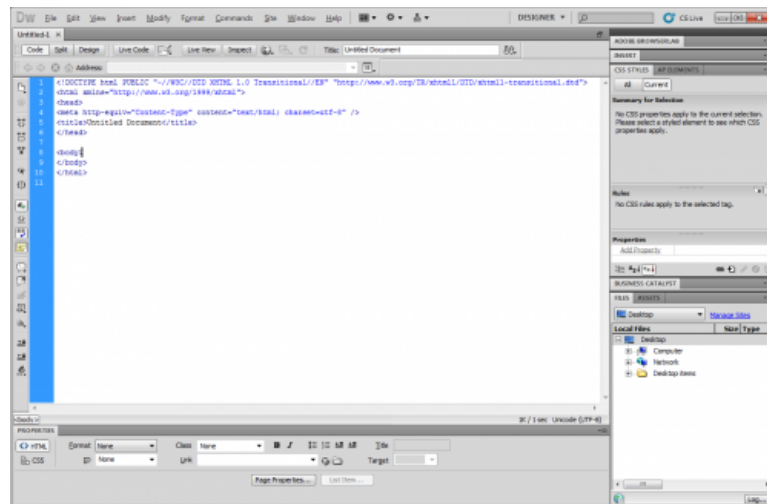


Slika 10: Uporaba programa Notepad++.

3.3.2 Adobe Dreamweaver

Drugo orodje je Dreamweaver proizvajalca Adobe [14]. Gre za orodje, ki ponuja okolje za razvoj spletnih strani in aplikacij (ang. Integrated development enviroment - IDE). Uporablja se predvsem za razvoj v html in CSS okolju ter programiranja v jezikih, ki se izvršujejo na strani strežnika (PHP, JavaScript, JSP, ASP itd.). Dreamweaver spada med tako imenovano skupino orodij, ki odražajo takojšne spremembe kode v izgledu oziroma funkcionalnosti programa (ang. What You See Is What You Get - WYSIWYG). Ta funkcionalnost nam je prišla prav predvsem pri izdelavi vmesnika, saj smo tako lažje spremljali njegov izgled. Pri programiranju v jeziku PHP in JavaScript, pa nam je prav prišla funkcionalnost nudenja namigov (ang. code hinting), ki programerju omogoča hitrejše programiranje.

Uporabniški vmesnik programa (slika 11) je dobro zasnovan in uporabniku omogoča funkcionalen pregled nad projektom. Z področji (kot so drevesna struktura projekta, nastavitve in informacije o gradnikih) omogoča hiter razvoj spletne strani oziroma aplikacije. Orodje Dreamweaver omogoča tudi vizualno programiranje, pri katerem uporabnik na delovno površino dodaja in ureja gradnike, pri tem pa orodje v ozadju generira kodo. Tega načina smo se pri našem delu raje izognili, saj je generirana koda včasih težko berljiva in razumljiva. Ker je program plačljiv smo v namen izdelave naše aplikacije uporabili testno različico programa.



Slika 11: Vmesnik orodja Dreamweaver.

3.4 Možni načini in orodja za izpis podatkov in izdelavo poročil

V tem poglavju bomo iskali primerno orodje in način za izpis podatkov ter izdelavo poročil. Pogledali si bomo nekaj orodij, ki omogočajo izdelavo poročil ter predstavili njihove prednosti in slabosti.

3.4.1 Nadgradnja obstoječe aplikacije

Podjetje je v preteklosti uporabljalo aplikacijo, napisano v programskem jeziku Delphi, ki je omogočala izdelavo preprostih poročil (slika 12). Do sedaj je podjetju uspešno služila. Po določenem času pa so na dan prišle nove zahteve, ki jih dosedanja aplikacija ni zmožna izpolnjevati. Pri iskanju rešitve problema smo zato imeli v mislih tudi nadgradnjo obstoječe aplikacije, a smo to možnost po premisleku ovrgli, saj programski jezik, v katerem je bila aplikacija napisana, ni omogočal izdelave aplikacije po naših željah.

Prednosti:

- prihranili bi veliko dela pri izdelavi aplikacije,
- problem z izdelovanjem poročil, bi bil rešen na podoben način kot doslej.

Slabosti:

- potrebno je učenje novega programskega jezika,
- veliko časa bi porabili za analizo obstoječe aplikacije,
- avtorske pravice so lahko potrebne,
- kljub temu, da aplikacija omogoča izdelavo poročil, so ta v nekaterih primerih nepopolno (robovi se prekrivajo z vsebino),
- otežena bi bila integracija z nadzornim sistemom,
- potrebna namestitvev aplikacije in orodja za branje poročil na vsak računalnik.

Alarmi za področje GORILEC		od	05.6.2012 10:55	SALONIT ANHOVO	
		do	05.6.2012 11:55		
ZACETEK	KONEC	OPIS NAPAKE	TIP	ALARM	
05.06.2012 11:17:08	06.06.2012 11:18:09	OPIS NAPAKE	ODPRAVLJENA	HST02_H_A	OPIS NAPAKE
Minimalen nivo					Pogon požarnik - S/transport
05.06.2012 11:21:42	06.06.2012 11:21:43		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:21:42	06.06.2012 11:21:49		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:21:54	06.06.2012 11:21:56		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:21:57	06.06.2012 11:21:58		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:21:59	06.06.2012 11:22:04		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:22:05	06.06.2012 11:22:08		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:22:05	06.06.2012 11:22:10		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:22:05	06.06.2012 11:22:13		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:22:15	06.06.2012 11:22:16		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:22:17	06.06.2012 11:22:19		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:22:17	06.06.2012 11:22:23		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:22:24	06.06.2012 11:22:25		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:22:33	06.06.2012 11:22:36		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura
05.06.2012 11:22:33	06.06.2012 11:22:38		ODPRAVLJENA	HST02_H_A	Upih v gorilec temperatura

Datum in ura izpisa: 5.6.2012 11:55:44 Strani: 1

Slika 12: Izpis alarmov s pomočjo obstoječe aplikacije.

3.4.2 Tiskanje podatkov iz html dokumenta

Izpis podatkov v obliki tabele s pomočjo funkcije napisane v programskem jeziku PHP [7] je zelo enostavna rešitev. Sprehodimo se skozi rezultate poizvedbe SQL in izpisujemo vrstice, pri tem pa oblikujemo morebitne robove tabel in pisavo besedila.

Prednosti:

- hiter izpis,
- enostavna izvedba,
- dobra integracija s spletno aplikacijo,
- nepotrebna uporaba dodatnih programov.

Slabosti:

- še vedno slab izpis pri tiskanju,
- problemi pri določanju začetka in konca strani,
- robovi pri tiskanju se popačijo,
- nekonsistenten izpis pri različnih brskalnikih,
- nekateri brskalniki imajo težave z izpisom robov,
- skoraj nemogoč izpis glave in noge za vsako stran dokumenta,
- potrebno znanje iz programskega jezika PHP, html in oblikovanja CSS,
- veliko časa porabimo za izdelavo in oblikovanje tabel.

3.4.3 Izpis poročila s pomočjo orodja PHP Report Maker

PHP Report Maker [15] je namensko orodje za izdelavo spletnih poročil. Omogoča izbiro podatkovne baze, predloge poročila (barve ozadja in tabel) ter izbiro načina izvoza (html, pdf, excel). Orodje je zasnovano tako, da uporabnik z uporabo čarovnika izbere najprej izvor podatkov, nato oblikuje izgled poročila, na koncu pa izbere med možnostmi izvoza poročila. Program generira projekt z datotekami PHP, katere nato naložimo na strežnik.

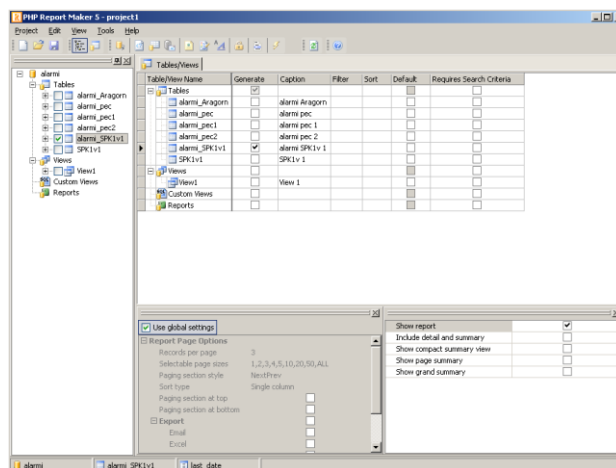
Prednosti:

- hitra izdelava poročila,
- enostaven čarovnik za izdelavo poročil,
- v relativno hitrem času imamo narejeno spletno stran s podatki iz podatkovne baze, ki jo lahko takoj objavimo.

Slabosti:

- ko je enkrat projekt generiran, ne moremo spreminjati poizvedbe,
- program je plačljiv,
- težave pri izvozu v pdf dokument, večje datoteke porabijo veliko časa za izvoz.

S tem orodjem še zmeraj ne rešimo problema izpisa podatkov na takšen način, da bi bil primeren za tiskanje. Izpis je žal primeren zgolj za ogled na zaslonu. Primer uporabe programa je na sliki 13.

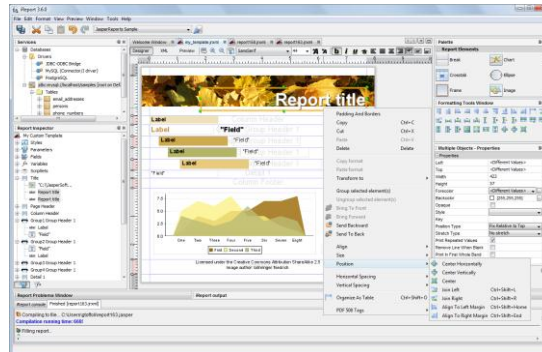


Slika 13: Izdelovanje poročila s pomočjo programa PHP Report Maker.

3.4.4 Izdelava poročil s pomočjo orodja iReport

Orodje iReport [8, 16] je prav tako namensko orodje za izdelavo poročil. Napisano je v Javi in za razliko od drugih orodij omogoča uporabo funkcij za izdelavo poročil tudi v Java programih. Tako nam daje proste roke pri izdelavi poročil. Uporabnik lahko s pomočjo

programa iReport Designer oblikuje predlogo poročila in določi tip podatkov, katere bo poročilo zajemalo. Na predlogo se lahko dodaja tako statične kot dinamične elemente, ki se glede na podatke iz podatkovne baze spreminjajo. Primer izdelave poročila z orodjem iReport Designer je prikazan na sliki 14.



Slika 14: Sestavljanje predloge poročila v orodju iReport Designer.

Prednosti:

- veliko možnosti pri izdelavi predloge poročila,
- enostavna izdelava predlog z orodjem iReport Designer,
- podprtost z različnimi grafi,
- možnost različnih načinov vnosa podatkov (xml, sql, Java collection),
- brezplačnost programa,
- enostavna uporaba orodja iReport Designer.

Slabosti:

- klici funkcij orodja možni samo v okolju Java,
- potrebno programersko znanje, če hočemo klicati funkcije iz drugega programa,
- zahtevna integracija z drugimi okolji (v našem primeru PHP).

Pri izbiri orodja smo imeli veliko težav. Najprej zato, ker nismo vedeli, ali katero orodje omogoča želene funkcionalnosti. Druga težava je način, kako bi lahko to orodje vgradili v našo aplikacijo.

Če bi šlo samo za izpis podatkov na zaslon, bi bila rešitev dokaj enostavna. Podatke bi enostavno predstavili v obliki tabelarnega zapisa v html dokumentu. Ne bi potrebovali nastavljanja glave in noge, ter drugih nastavitev, katere so nujne pri izpisu na list papirja. Če hočemo poročilo predstaviti tako, da bo primerno tudi za tiskanje, moramo izbrati namensko orodje, ki nam bo to omogočilo.

4 Predstavitev rešitve

Nobeno orodje samo po sebi ne omogoča uporabniku, da bi na enostaven način izbral kriterije povpraševanja. Od uporabnika se zahteva, da navede način dostopa do podatkovne baze in vpiše poizvedbo SQL. Poleg tega je treba paziti na pravilno sklicevanje stolpcev v tabelah podatkovne baze. Glede na to, da povprečen uporabnik ne pozna stavkov SQL, je to zanj prevelik zalogaj.

Kot rešitev smo si omislili izdelavo uporabniškega vmesnika, ki bi omogočal izbiro kriterijev povpraševanja. Glede na izbrane kriterije, bi v poročilu izpisali pravilne podatke. Aplikacijo smo si zamislili kot spletno storitev, do katere, bi lahko dostopali iz vsakega računalnika v istem omrežju. Prednost spletne aplikacije je podprtost na vseh sistemih, ne glede na operacijski sistem, saj ni potrebe po inštalaciji dodatnih programov, razen brskalnika in pregledovalnika pdf [23] vsebin. Takšen tip aplikacije bi bil primeren tudi za vgradnjo v SCADA nadzorni sistem.

4.1 Izbira orodja za izdelavo poročil

Pri izbiri orodja za izdelavo poročil smo imeli veliko težav. Na prvi pogled optimalne rešitve ni bilo na vidiku. Poskušali smo izdelati html dokument tako, da bi prepoznal začetek in konec strani, ter omogočil izpis glave in noge. Pri tem poskusu smo naleteli na naslednje težave:

- oblika izpisa poročila se je razlikovala pri različnih brskalnikih (npr. velikost pisave),
- robovi tabel so se pri določenih brskalnikih popačili (npr. debelejši rob glave in noge ni bil enakomeren),
- za pravilno določanje začetka in konca strani smo morali določiti fiksno število vrstic na stran.

Pri fiksnem določanju števila vrstic je nastala težava, če se je izpis katerega izmed podatkov zaradi svoje dolžine prelomil v naslednjo vrstico. Rezultat je bil zamik celotnega poročila. Kmalu smo prišli do spoznanja, da takšna rešitev ne pride v poštev. Takšen način izpisa nam bi prišel v poštev zgolj kot nekakšen predogled podatkov preden jih natisnemo na list papirja.

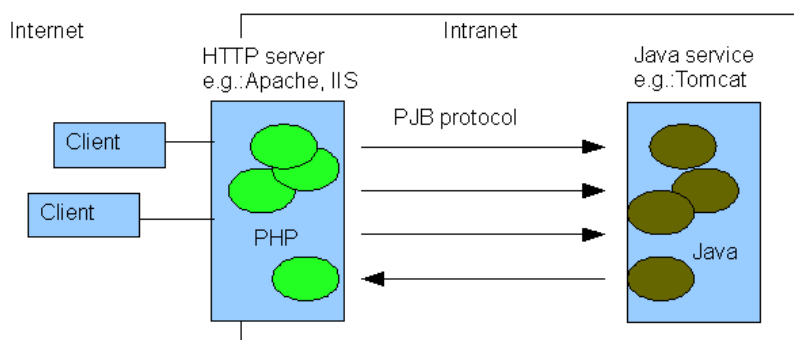
Naš naslednji kandidat je bilo orodje iReport. Orodje ima veliko funkcionalnosti. Pri zagonu izdelave poročil omogoča poganjanje skript in izvajanje regularnih izrazov. Poleg tega ima pregleden urejevalnik poročil, v katerem lahko na poročilo dodajamo različne komponente, kot so tabele, podatke iz podatkovne baze, različne spremenljivke, grafi, slike, itd.

Njegova največja prednost pred ostalimi orodji je omogočanje klicev funkcij za izgradnjo poročila iz programa napisanega v Javi. To bi nam omogočilo izgradnjo poročil iz naše aplikacije. Problem pa je nastal v tem, kako bi iz naše PHP aplikacije klicali Javanske

funkcije. Vprašanje je ali je to sploh mogoče. Možno rešitev bi verjetno opustili, če ne bi našli rešitve v tako imenovani povezavi PHP/Java Bridge. V nadaljevanju bomo predstavili kakšen tip povezave smo uporabili za povezavo med programskima jezika PHP in Javo. Predstavili bomo tudi dva primera uporabe.

4.2 Vzpostavitev povezave PHP/Java Bridge

PHP/Java Bridge [17] je implementacija pretočne povezave, ki bazira na XML (ang. eXtensible Markup Language) omrežnem protokolu. Takšna povezava se uporablja za povezavo glavnih funkcionalnosti med PHP-jem in Javinim virtualnim računalnikom. Takšen način komunikacije je dosti hitrejši od ostalih alternativ (ang. Remote procedure call via Simple object access protocol - RPC via SOAP) [13], ter porabi manj resursov na strani strežnika. Poleg tega pa ne potrebuje nobenih dodatnih komponent za klice Java procedur iz PHP-ja ali PHP procedur iz Jave (slika 15). V primeru, da je naša aplikacija napisana v čistem PHP-ju, je to najboljša rešitev implementacije Javanskih metod.



Slika 15: Prikaz delovanja povezave med programskima jezika Java in PHP.

Naša aplikacija živi na strežniku Apache, ki ga dobimo v kompletu programa WAMP [18]. Za delovanje povezave med programskima jezika Java in PHP pa smo potrebovali strežnik Apache Tomcat. Na njem se nahajajo tudi vsi potrebni razredi za klice metod iReporta. V datoteki (v našem primeru `iReportPrint.php`) kličemo Javine metode in iReporta, moramo uvoziti tudi datoteko `java.inc`, ki nam omogoča povezavo do strežnika Apache Tomcat in klice metod.

V naslednjem programu bomo pokazali uporabo povezave. Namen PHP programa je izpis enostavnega niza s pomočjo Javanskih objektov.

```
<?php
//uvoz datoteke java.inc
include("java.inc");

//razredi Jave
```

```

use java\lang\String as String;

//deklaracija novega razreda
$pozdrav = new String("Hello World!");
//izpis
echo $pozdrav;
?>

```

Rezultat zgornjega programa je izpis besedila »Pozdravljen Svet« (ang. »Hello World!«). Nekoliko zahtevnejši program prikazuje uporabo Javinih zbirk.

```

<?php
include("java.inc");

//javini razredi
use java\util\HashMap as HashMap;
use java\util\ArrayList as ArrayList;
use java\lang\String as String;
use java\lang\Integer as Integer;

//nova zbirka oseb
$listOseb = new ArrayList();

//ustvarimo osebo1 in ji dodamo lastnosti
$oseba1 = new HashMap();
$oseba1->put("ime", "Martin");
$oseba1->put("priimek", "Krpan");
$oseba1->put("starost", 30);

//osebo1 dodamo v zbirko oseb
$listOseb->add(0,$oseba1);

//ustvarimo osebo2
$oseba2 = new HashMap();
$oseba2->put("ime", "Peter");
$oseba2->put("priimek", "Klepec");
$oseba2->put("starost", 20);

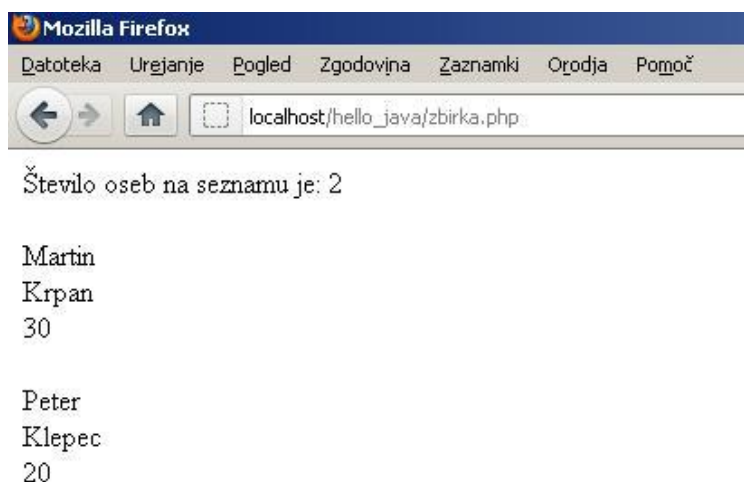
//osebo2 dodamo v zbirko oseb
$listOseb->add(1,$oseba2);

//prikaz zanimivega tipiziranja
$n = (integer)(string)(new String($listOseb->size()));
echo "tevilo oseb na seznamu je: " . $n . "</br></br>";

//izpis oseb iz zbirke
for($i = 0; $i < $n; $i++){
    $osebaTemp = $listOseb->get($i);
    echo $osebaTemp->get("ime") . "</br>";
    echo $osebaTemp->get("priimek") . "</br>";
    echo $osebaTemp->get("starost") . "</br>";
    echo "</br>";
}
?>

```

Rezultat zgornjega programa je izpis podatkov o dveh osebah v html dokumentu (slika 16).



Slika 16: Rezultat programa `zbirka.php`.

4.3 Postavitev podatkovne baze in povezave z SCADA nadzornim sistemom

Podatkovna baza je namenjena shranjevanju podatkov. Vsi podatki o alarmih in dogodkih iz nadzornega sistema se shranjujejo na podatkovni bazi.

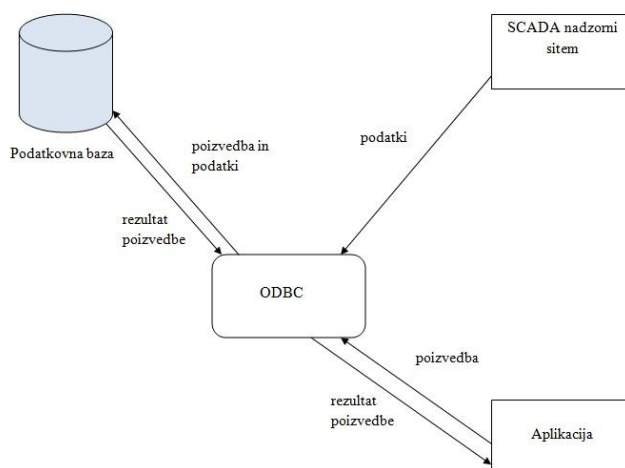
Zahtevana je bila uporaba Microsoftovega SQL strežnika, saj je v prihodnosti načrtovana selitev aplikacije in podatkovne baze na strežnik z Microsoftovo programsko opremo. Za naše testiranje smo uporabili brezplačni strežnik Microsoft SQL server express [5]. Strežnik podpira velikosti podatkovnih baz do velikosti 4GB, kar je za naše potrebe popolnoma zadoščalo. Uporabljali smo starejšo verzijo strežnika SQL server express 2005. Razlog pa je bil kompatibilnost ODBC gonilnika z nadzornim sistemom SCADA. Izkazalo se je, da pri novejših verzijah strežnika povezava med strežnikom in nadzornim sistemom ni delovala.

Na voljo nam je več paketov. Izbiramo lahko od osnovnega (v katerem je zgolj gola podatkovna baza), do paketov za zahtevnejše uporabnike v katerih dobimo dodatna programska orodja za manipulacijo z podatkovno bazo. Izbrali smo paket Advanced, v katerem dobimo grafično orodje za upravljanje z bazo, ter dodatna programska orodja za napredno iskanje nizov. Še kako nam je prišlo prav orodje Microsoft SQL Server Management Studio Express v katerem smo enostavno ustvarjali nove podatkovne baze in tabele ter dodajali nove uporabnike. Uporabljali smo ga tudi kot pripomoček za testiranje pravilne izvedbe povpraševanj.

Pri vzpostavljanju povezave z nadzornim sistemom smo uporabili vmesnik ODBC [19]. ODBC (ang. Open Database Connectivity) je vmesnik, napisan v programskem jeziku C, za dostop do podatkovnih baz. Razvijalci so pri tem hoteli doseči, da bi bil sistem neodvisen od podatkovne baze in operacijskega sistema. To so dosegli tako, da so ODBC gonilnik uporabili

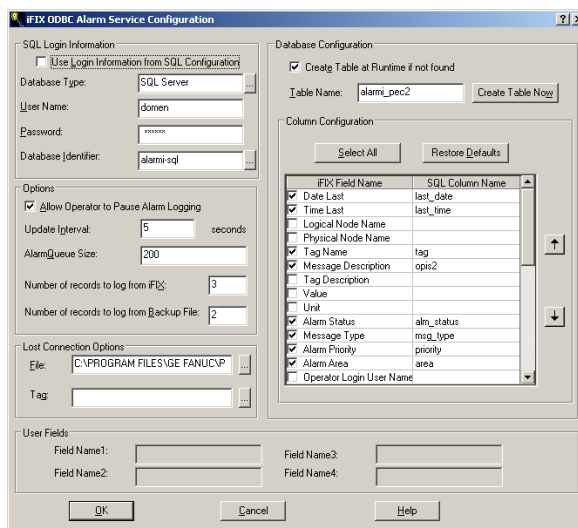
kot sloj za prevajanje med aplikacijo in podatkovno bazo. Poizvedba SQL tako potuje do ODBC gonilnika, ki ga ta nato pošlje do podatkovne baze.

ODBC povezavo smo uporabljali za povezavo z nadzornim sistemom in kot vmesnik za povezavo naše aplikacije z podatkovno bazo. Za lažjo predstavo uporabe ODBC-ja smo njegovo uporabo predstavili v diagramu, ki je prikazan na sliki 17.



Slika 17: Diagram uporabe ODBC povezave.

Podatkovni bazi smo morali zagotoviti vir informacij. V nastavitvah nadzornega sistema smo morali nastaviti še pošiljanje alarmov in dogodkov preko ODBC povezave. Na sliki 17 lahko vidimo nastavitveno okno in možnosti, ki nam jih to ponuja.



Slika 18: Nastavitve pošiljanja alarmov preko ODBC vmesnika.

V nastavitvah moramo navesti ime ODBC povezave ter uporabniško ime in geslo, s katerim dostopamo do podatkovne baze. Pri tem lahko sami definiramo tabelo v podatkovni bazi ali pa to prepustimo nadzornemu sistemu. V območju `Column Configuration` določimo, katere parametre bomo shranjevali in kako jih bomo poimenovali.

Glede na želje uporabnika smo izbrali ustrezne:

- `last_date` – prikazuje datum odprave alarma ali dogodka,
- `last_time` – prikazuje čas odprave alarma ali dogodka,
- `tag` – ime alarma oziroma naprave, ki je prožila alarm,
- `opis`, `opis1` in `opis2` – opis naprave, alarma ali dogodka - v prihodnosti nameravamo dva atributa uporabiti za prikaz slovenskega in angleškega prevoda opisa,
- `alm_status` – status alarma - s pomočjo tega atributa izvemo ali je bil alarm potrjen ali ne,
- `msg_type` – vrednost atributa določa izvor in tip podatka - pomaga nam pri določanju alarmov in dogodkov.
- `priority` – parameter `priority` nosi informacijo o prioriteti alarma - zasede lahko vrednosti `high`, `medium` in `low`. V kolikor je prioriteta `medium` ali `low` gre za opozorilo, drugače gre za alarm,
- `area` – iz atributa `area` izvemo področje v katerem je alarm ali dogodek nastal,
- `start_time` – čas nastanka alarma ali dogodka,
- `start_date` – datum nastanka alarma ali dogodka,
- `start` – datum in čas nastanka alarma ali dogodka,
- `finish` – datum in čas odprave alarma ali dogodka.

V območju `Lost Connection Options` smo nastavili tudi lokalno beleženje podatkov, v kolikor bi prišlo do izpada povezave.

4.4 Postopek izdelave aplikacije

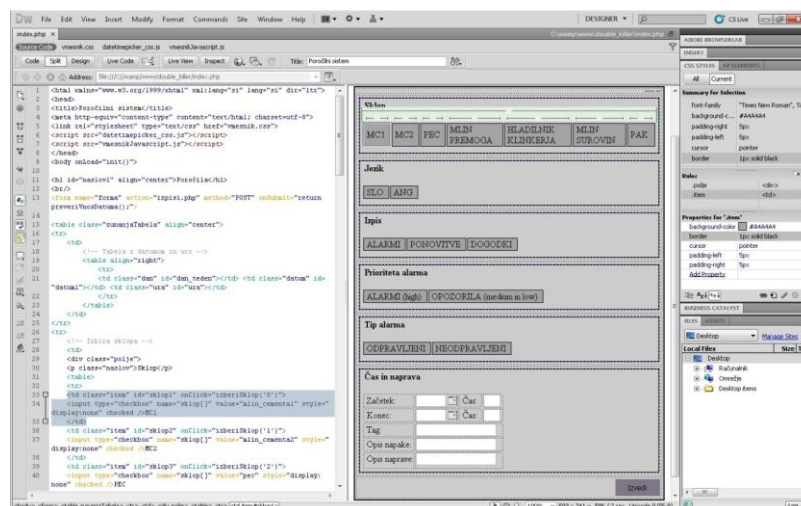
Prvi korak pri razvoju aplikacije je bil izdelava uporabniškega vmesnika. Sprva smo vanj vgradili le osnovne funkcionalnosti kot so iskanje po datumu in času. S temi parametri smo se nato poskušali povezati s podatkovno bazo in izvršiti prva poizvedovanja. Ko smo uspeli odpraviti določene začetniške napake, smo vmesnik nadgradili z ostalimi funkcionalnostmi, kot so izbira sklopa in področja ter tipa in prioritete alarma. Povpraševanja so tako postala zahtevnejša, kar je povzročilo, da smo morali logiko povpraševanja dobro načrtovati.

4.4.1 Izdelava vmesnika

Začeli smo z načrtovanjem in izgradnjo vmesnika. Za poskusno delovanje aplikacije smo poskušali sestaviti vmesnik, ki bi bil podoben že obstoječi aplikaciji. Vmesnik smo izdelali v html kodi. Za vizualno urejevanje in pregled smo si pomagali z preizkusno različico orodja Dreamweaver. Področja vmesnika smo razdelili s pomočjo uporabe tabel. Takšen način pozicioniranja je učinkovit tudi pri spreminjanju velikosti vmesnika in pri uporabi različnih brskalnikov. Za oblikovanje pisav, barv gradnikov ter ozadja smo uporabili stilske predloge CSS (ang. Cascading Style Sheets) [20]. Naš prvi uporabniški vmesnik je prikazan na sliki 19.

Slika 19: Prva verzija uporabniškega vmesnika.

S prvim uporabniškim vmesnikom smo testirali delovanje aplikacije za vse vhodne parametre. Po zadovoljivem delovanju testne aplikacije smo začeli z izdelavo novega vmesnika, pri katerem smo upoštevali vse zahteve uporabnika.



Slika 20: Izdelovanje novega uporabniškega vmesnika v orodju Dreamweaver.

Na sliki 20 lahko vidimo izdelavo nove verzije uporabniškega vmesnika. Tudi v tem vmesniku smo za postavitev elementov uporabili tabelo. V primerjavi s prejšnjo verzijo vmesnika ima nov vmesnik namesto izbirnih področij, gumbe. Tako smo izboljšali usklajenost z ostalimi deli aplikacije in preglednost med gradniki vmesnika. V primeru pritiska na gumb smo ta gumb obarvali. Z JavaScript [21] funkcijo smo sprogramirali vizualne spremembe html gradnikov. Napisali smo tudi kodo, ki ob zagonu aplikacije obarva privzeto izbrane elemente.

Več lastnosti vmesnika bomo predstavili v poglavju Pregled in uporaba aplikacije, v katerem bomo predstavili celotno delovanje aplikacije.

4.4.2 Programiranje logike aplikacije

Podatki iz uporabniškega vmesnika se preko metode `$_POST` pošljejo datoteki `izpisi.php`. Glede na to, kateri tip podatkov je bil izbran (alarmi, dogodki ali ponovitve), določimo tip izpisa. Ker je za vsak tip izpisa drugačen način povpraševanja, imamo povpraševanja ločena v datotekah. Ko vemo kateri izpis bomo izbrali, uvozimo ustrezno datoteko.

V vsaki datoteki enolično oblikujemo stavek SQL [6]. Vrstici `select in from`, ter dodatni pogoji ostanejo že vnaprej določeni, saj niso odvisni od parametrov iz vmesnik. Glavne pogoje iz vrstice `where` generira funkcija `sestaviPogoj($selectAndFrom)`. Naloga funkcije je, da glede na parametre, ki so bili izbrani v vmesniku, sestavi pogoje in jih doda obstoječi `select in from` vrstici. Na koncu smo dodali še dodatne pogoje, kateri so enolični za vsak tip izpisa. Funkcijo `sestaviPogoj($selectAndFrom)` smo sestavili tako, da je njen rezultat uporaben tako pri izpisu alarmov, dogodkov ali pri izpisu ponovitev.

```
SELECT podatki
FROM tabela
WHERE
+
sestaviPogoj($niz)
+
dodatni pogoji
```

Če hočemo izpisati alarme, uvozimo datoteko `izpisAlarmov.php`, če dogodke uporabimo datoteko `izpisDogodkov.php`, v primeru, da želimo pregledati ponovitve alarmov uvozimo datoteko `izpisPonovitev.php`.

a) Izpis alarmov

Izpis alarmov izpiše alarme, ki zadostujejo kriterijem poizvedbe. Alarm je dogodek, ki se zapiše v podatkovno bazo, če določena naprava neha delovati ali deluje izven mej normalnega obratovanja (npr. temperatura olja v motorju je višja od dovoljene). Iz podatkovne baze

razlikujemo alarme od drugih podatkov s pomočjo vrednosti atributa `msg_type`, ki je enak `alarm`.

Pri tem se v bazo zapiše:

- datum in ura nastalega alarma,
- datum in ura potrditve alarma,
- ime naprave,
- opis napake,
- opis naprave,
- tip alarma,
- tip sporočila,
- prioriteta alarma,
- območje v katero naprava spada.

V kolikor alarm še ni bil potrjen, sta datum in ura nastanka ter datum in ura potrditve alarma enaka.

b) Izpis dogodkov

Poleg alarmov se v podatkovno bazo beležijo tudi dogodki. Dogodki so vse spremembe, ki jih operater izvede na SCADA klient računalniku. Poleg tega se kot dogodek beleži tudi sprememba stanja sistema SCADA. Dogodki se od alarmov razlikujejo po atributu `msg_type`, ki je enak `operator`, ter samem formatu zapisa atributa `opis2`. Dogodki nimajo posebej definiranega alarmnega področja, zato imajo alarmno področje enako `all`. Ta spoznanja, so nam pomagala pri ločevanju dogodkov od drugih zapisov v podatkovni bazi. Kjer se potrjuje alarme celotnega področja, se podatek o področju zapiše v atribut `opis2` v oglatih oklepajih (npr. `[HLAJENJE] Fix32.HLAJENJE.H_60_TRANS_STOP_SC.F_CV set to 1 by MC1_3::IZMENA`).

c) Izpis ponovitev

Ponovitve so število enakih alarmov, ki se vsaj enkrat ponovijo v obdobju, ki ga določi uporabnik. Prikaz ponovitev alarmov je smotno, saj nam pri diagnosticiranju napak zelo pomaga podatek, kolikokrat se je ponovil alarm določene naprave. Ponovitve pri izpisu sortiramo po padajočem vrstnem redu, tako, da so najpogosteje ponovljeni alarmi na začetku. Število ponovljenih alarmov izračunamo s pomočjo stavka SQL.

```
SELECT COUNT(tag) as skupaj, tag, opis1, opis2
FROM alarmi_pec2
WHERE start >= '19.3.2012 11:45:25' AND
      finish <= '19.3.2012 12:45:25' AND
      tag != ''
GROUP BY tag, opis1, opis2
HAVING COUNT(tag) > 1
ORDER BY skupaj DESC
```

Ko imamo sestavljeno ogrodje stavka SQL in vemo po katerem tipu podatkov bomo povpraševali, nam ostane samo še izgradnja pogoja. Postopek za izgradnjo pogoja v stavku SQL smo si zamislili nekako takole. V datoteki `funkcije.php` smo najprej preverili, po čemu bomo vse povpraševali. Glede na to, kateri podatki so bili izbrani v uporabniškem vmesniku, smo za sestavo pogoja klicali ustrezna funkcije. Tako smo ustvarili popoln sistem pogojev.

Funkcij v datoteki `funkcije.php`, ki skrbijo za izgradnjo stavka SQL:

- `sestaviCas()` – sestavi pogoje glede na izbran čas,
- `sestaviSklop()` – sestavi pogoje glede na izbran sklop (izbranih sklopov je lahko tudi več),
- `sestaviPodrocje()` – stavku SQL doda pogoj za izbiro področja,
- `sestaviPrioriteta()` – omogoča sestavo pravilnega povpraševanja, glede na izbrano prioriteto.

Aplikacija je zasnovana tako, da mora uporabnik vnesti vsaj podatke o začetnem času. V kolikor to polje pusti prazno, se povpraševanje ne izvrši, uporabnik pa dobi obvestilo o napaki.

Ko imamo stavek SQL sestavljen, nam preostane še povezava z bazo in izvršitev povpraševanja. S podatkovno bazo smo se povezali preko ODBC vmesnika.

```
<?php
//Podatki o podatkovni bazi
$povezavaODBC = "alarmi-sql";
$upIme = "uporabnik";
$geslo = "geslo";

//Vzpostavljanje povezave
$povezava = odbc_connect($povezavaODBC, $upIme, $geslo);
if(!$povezava){
    exit("Povezava ni uspela");
}

//Izvedba povpraševanja
$rez = odbc_exec($povezava, $sql);
if(!$rez){
    exit("Prislo je do napake v poizvedovanju");
}
?>
```

Rezultati povpraševanja se shranijo v spremenljivko `$rez`, ki je nato na voljo funkcijam za izpis podatkov.

4.4.3 Način izpisa podatkov

Uporabniku smo omogočili dva načina izpisa. Prvi način je izpis podatkov v html tabeli. Ta način je primeren za hiter pregled. Uporabnik lahko uporablja funkcionalnost brskalnika, ki mu omogoča iskanje nizov v dokumentu. Tako lahko na primer hitro poišče ime naprave ali čas nastopa alarma oziroma dogodka. Za izbiro takega načina izpisa smo se odločili z zavedanjem, da bo uporabnik večinoma izpis podatkov uporabil zgolj za pregled ali iskanje neke vsebine.

Kot drugo možnost smo omogočili izpis podatkov v obliki poročila, ki je primerno za tiskanje in nato arhiviranje. Za izvedbo te opcije smo uporabili orodje iReport za izdelavo predlog poročil, ter povezavo PHP/Java Bridge za klice funkcij iReport-a iz PHP projekta.

a) Izpis podatkov v html tabeli

Vsak tip izpisa (alarmi, dogodki, ponovitve) smo sprogramirali v svoji funkciji. Vsaka funkcija kot argument prejme rezultat povpraševanja. Izbiro ustrezne funkcije dosežemo z naslednjo kodo.

```
switch ($mode) {
    case "alarmi": izpisAlarmov($rez);
        break;
    case "dogodki": izpisDogodkov($rez);
        break;
    case "ponovitve": izpisPonovitev($rez);
        break;
    default:
        echo "Napaka pri izpisu.";
}
```

Po izbiri ustrezne funkcije izpišemo podatke. V spodnji kodi bomo prikazali izpis alarmov v html tabeli. Poleg alarmov imamo še funkcijo `izpisDogodkov($rez)`, ki izpisuje dogodke in funkcijo `izpisPonovitev($rez)`, ki izpisuje ponovitve alarmov. Funkcije se razlikujejo po podatkih v glavi poročila in izpisanih atributih. Oblikovanje pa je za vse tipe izpisa enako.

```
<?php
function izpisAlarmov($rez) {
    $zacetek = $_POST["zacetek"] . " " . $_POST["cas_zacetek"];
    $konec;
    $podrocje;

    if($_POST["konec"] == "") {
        $konec = $_POST["zacetek"] . " " . $_POST["cas_konec"];
    }
    else {
        $konec = $_POST["konec"] . " " . $_POST["cas_konec"];
    }

    if(isset($_POST["podrocje"])) {
```

```

    $podrocje = $_POST["podrocje"];
}
else{
    $podrocje = "PEC";
}

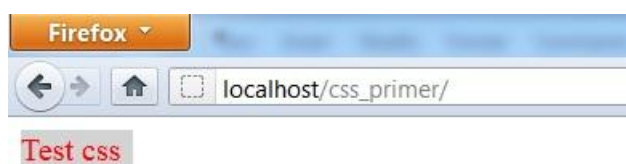
//Izpis glave
echo "
<table align='center'>
<tr>
    <td colspan='4' class='glavaL'>Alarmi za Podrocje:</td>
    <td colspan='2' class='glavaD' id='zacetek'>OD: " . $zacetek . "</td>
</tr>
<tr>
    <td colspan='4' class='glavaL' id='podrocje'>" . $podrocje . "</td>
    <td colspan='2' class='glavaD' id='konec'>DO: " . $konec . "</td>
</tr>
<tr>
    <td class='glavaL'>Zacetek</td>
    <td class='glava'>Konec</td>
    <td class='glava'>Tip</td>
    <td class='glava'>Alarm</td>
    <td class='glava'>Opis napake</td>
    <td class='glavaD'>Opis naprave</td>
</tr>";

//Izpis podatkov
while(odbc_fetch_row($rez)){
    //Vrstica s podatki
    echo"
    <tr>
    <td class='leviZgoraj'>" . odbc_result($rez, "start_date") . "</td>
    <td class='zgoraj'>" . odbc_result($rez, "last_date") . "</td>";
    $status = odbc_result($rez, "alm_status");
    if (substr($status, 0, 3) == "CFN"){
        echo "<td class='sredina' rowspan='2'>nov</td>";
    }
    else if (substr($status, 0, 2) == "OK"){
        echo "<td class='sredina' rowspan='2'>odpravljen</td>";
    }
    else{
        echo "<td class='sredina' rowspan='2'></td>";
    }
    echo "
    <td class='sredina' rowspan='2'>" . odbc_result($rez, "tag") . "</td>
    <td class='sredina' rowspan='2'>" . odbc_result($rez, "opis2") . "</td>
    <td class='desni' rowspan='2'>" . odbc_result($rez, "opis1") . "</td>
    </tr>
    <tr>
        <td class='levi' id='krepko1'>" . substr(odbc_result($rez,
"start_time"),0,-7) . "</td>
        <td class='spodaj' id='krepko2'>" . substr(odbc_result($rez,
"last_time"),0,-7) . "</td>
    </tr>";
}
echo "</table>";
}
?>

```

Podatki so sedaj zapisani v tabeli a še vedno nismo zadovoljni z obliko izpisa (besedilo še ni oblikovano, ter robovi tabel niso vidni). Težavo rešimo z uporabo stilskih predlog CSS, ki opisujejo prikaz elementov html dokumenta. CSS je možno uporabiti v html dokumentu, ali v ločeni datoteki tipa `.css`. V naslednjem primeru bomo demonstrirali enostavno oblikovanje besedila v dokumentu html. Rezultat spodnje kode je prikazan na sliki 21.

```
<html>
<body>
  <p style="font-size:20px; color:red; background-color:lightgrey;
    width:70px">
    Test css
  </p>
</body>
</html>
```



Slika 21: Rezultat oblikovanja.

V naslednjem primeru pa bomo predstavili oblikovanje v ločeni datoteki. Oblikovanje v ločeni datoteki je še posebej primerno, če se oblikovanje navezuje na več strani. Z zunanjim oblikovanjem lahko dosežemo, da z spremembo ene datoteke spremenimo izgled celotne spletne strani. V našem primeru smo zaradi boljše berljivosti kode in lažje preglednosti aplikacije uporabili oblikovanje v ločeni datoteki. Html dokumentu povemo da bomo oblikovanje uporabili v drugi datoteki tako, da glavi dokumenta dodamo povezavo do datoteke CSS (v našem primeru `porocilo.css`) in navedemo, da hočemo to datoteko uporabiti za oblikovanje.

```
<head>
<link rel="stylesheet" type="text/css" href="porocilo.css"/>
</head>
```

Sintaksa jezika CSS je enostavna. Za vsak element navedemo oblikovanje v bloku, omejenem z zavrtimi oklepaji. Za sklicevanje na element lahko uporabimo razred (`class`) ali enolični identifikator (`id`). Razred uporabljamo če se več elementov sklicuje na isto oblikovanje. To nam omogoča, da z enim oblikovanjem oblikujemo več elementov istega razreda. Razredno naslavljanje uporablja atribut `class` v html dokumentu in je definiran s simbolom `».«`. Id uporabljamo ko se sklicujemo na en unikatni element dokumenta. Id naslavljanje uporablja atribut `id` v html dokumentu in je določen s simbolom `»#«`.

```
.glavaL{
  border-top:2px solid #000;
  border-bottom:2px solid #000;
```

```

border-left:2px solid #000;
font-size:14px;
font-weight:bold;
background-color:#69F;
}
.glava{
border-top:2px solid #000;
border-bottom:2px solid #000;
font-size:14px;
font-weight:bold;
background-color:#69F;
}
.glavaD{
border-top:2px solid #000;
border-bottom:2px solid #000;
border-right:2px solid #000;
font-size:14px;
font-weight:bold;
background-color:#69F;
}

```

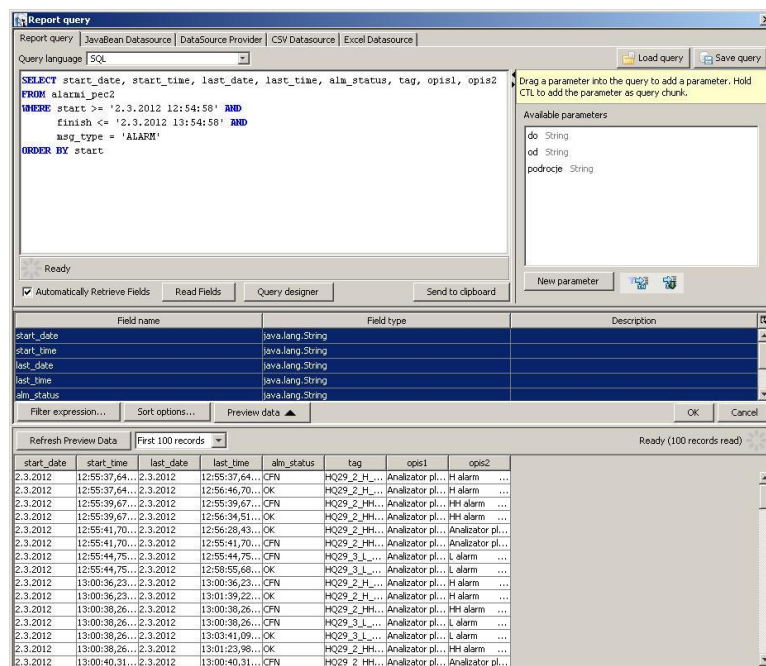
Zgornja koda prikazuje oblikovanje glave izpisanih podatkov. Rezultat oblikovanja je viden na sliki 22.

Alarmi za Področje:		OD: 21.3.2012 12:53:21	DO: 11.4.2012 13:53:21
PEC		Tip alarmov: vsi	
Začetek	Konec	Alarm	Opis napake
			Opis naprave

Slika 22: Rezultat zgornjega oblikovanja glave izpisa alarmov.

b) Izdelava predlog poročil z orodjem iReport Designer

V tem razdelku bomo predstavili izdelavo predlog poročil. Predloge poročil smo izdelali s programom iReport Designer. Orodje je dobro pregledno in omogoča uporabo številnih orodij pri izdelavi predlog. S tem orodjem smo lahko enostavno oblikovali glavo in nogo poročila, kar je bilo z html tabelami skoraj nemogoče. Program nam na začetku ponudi že izdelane predloge, kar bi prišlo prav za hitro izdelavo poročil. Ponujena nam je možnost, da z orodjem Report Query (slika 23) vnesemo poizvedbo SQL, ter tako na enostaven način pridobimo podatke, ki jih želimo prikazati v poročilu.

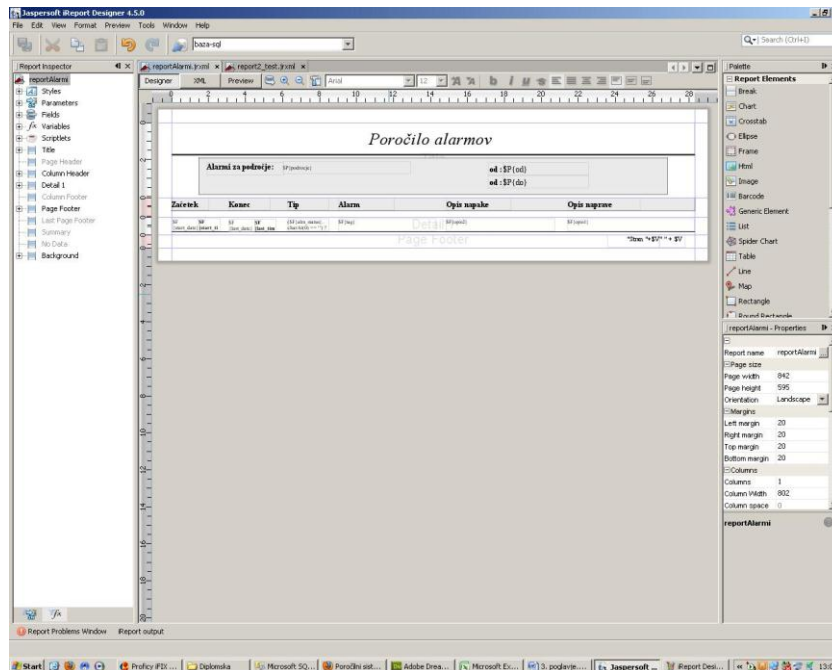


Slika 23: Uporaba orodja Report Query.

V našem primeru ta postopek žal ni prišel v poštev. Poizvedbo SQL smo morali izvršiti na nivoju aplikacije. Tako smo program iReport Designer uporabljali zgolj za izdelavo predlog poročil.

Pri sestavi predlog nam pomaga pregledno in funkcionalno delovno okolje (slika 24). Na levi strani imamo drevo strukture projekta, na desni pa paletu z gradniki in okno z nastavitvami. V sredini imamo predlogo razdeljeno na območja. Območja lahko po želji dodajamo ali odstranjujemo odvisno od naših želja in potreb. Uporabili smo naslednja:

- Title - vanj smo vnesli naslov glavo poročila (območje title se pojavi na začetku poročila),
- Column Header - tukaj smo določili naslove atributov ki smo jih nato prikazali v območju detail,
- Detail - v tem območju smo izpisovali podatke iz podatkovne baze,
- Page Footer - območje smo uporabili za nogo poročila, v kateri smo izpisovali trenutno število strani.



Slika 24: Delovno območje programa iReport Designer in izdelava predloge poročil alarmov.

4.4.4 Uporaba predlog poročil v aplikaciji

Našo spletno aplikacijo smo morali nekako povezati z iReport poročili, da smo lahko uporabili funkcionalnosti, ki jih orodje omogoča. Kot že rečeno naša aplikacija živi na Apache strežniku. Žal strežnik ne podpira Javanskih aplikacij, zato smo zanje uporabili strežnik Apache Tomcat. Nanj smo naložili datoteke potrebne za vzpostavitev povezave med programskima jezicoma Java in PHP. To smo storili tako, da smo v mapo `webapps` strežnika Apache Tomcat uvozili datoteko `JavaBridge.war`. War (ang. Web application ARchive) datoteka je Java arhiv za spletne aplikacije v katerem so razredi in ostale datoteke za poganjanje spletnih aplikacij. Poleg datotek potrebnih za povezavo smo v mapo `lib` uvozili arhive (JAR datoteke) iReport-a. S tem smo se izognili morebitnih težav pri definiciji razrednih poti. S tem je naš strežnik Tomcat pripravljen na gradnjo poročil.

V aplikaciji smo klice metod iReport-a uporabili v datoteki `ireportPrint.php`. Vse potrebne podatke kot so stavek SQL in ostali parametri smo pridobili z pomočjo forme iz strani `izpisi.php`. Za vzpostavitev povezave med strežnikoma smo v datoteko `ireportPrint.php` morali uvoziti datoteko `java.inc`. Glede na izbrane podatke iz forme smo izbrali ustrezno predlogo poročil in ustrezne parametre. V spodnjem izseku kode prikazujemo izbiro primerne predloge poročila, glede na izbran tip izpisa.

```
<?php
function izberiPredlogo($tip){
    $CompilerManager = new JasperCompilerManager();
    $predloga;
```

```

//Izbira ustrezne predloge
switch ($tip){
case "alarmi" :
    $predloga = $CompileManager->compileReport
    (realpath("./ireportTemplates/reportAlarmi.jrxml"));
    break;
case "dogodki":
    $predloga = $CompileManager->compileReport
    (realpath("./ireportTemplates/reportDogodki.jrxml"));
    break;
case "ponovitve":
    $predloga = $CompileManager->compileReport
    (realpath("./ireportTemplates/reportPonovitve.jrxml"));
    break;
default:
    echo "Prislo je do napake pri izbiri tipa izpisa";
    break;
}
return $predloga;
}
?>

```

Pri izpisovanju poročila smo uporabljali naslednje razrede:

- `JasperCompileManager()` – razred skrbi za prevajanje poročil,
- `JasperFillManager()` – z uporabo tega razreda napolnimo podatke v predlogo,
- `JasperExportManager()` – razred nam omogoča izvoz poročila v različnih formatih,
- `HashMap()` – Javina zbirka podatkov. Uporabili smo jo za hranjenje parametrov, ki smo jih prikazali v poročilu,
- `DriverManager()` – s pomočjo tega razreda smo izbrali ustrezen gonilnik za povezavo z našo podatkovno bazo in izvedli povpraševanje,
- `JRResultSetDataSource()` – v tej zbirki smo hranili rezultate povpraševanja.

Za konec smo si pustili reševanje enega izmed težjih problemov. Še zmeraj je bilo najtežje izbrati in sestaviti najprimernejšo rešitev za izdelavo poročil, tako, da bi lahko naš slednji problem postavili na drugo mesto težavnosti. Spoprijeti smo se morali z problemom podvajanja alarmov.

4.5 Odprava ponavljajočih alarmov

V sistemu imamo odpravljene in nove ali neodpravljene alarme. Odpravljene alarmi so tisti alarmi, ki jih je operater potrdil in se takoj po potrditvi izbrišejo iz sistema. Novi alarmi pa so alarmi, ki še niso potrjeni in čakajo, da jih operater potrdi (so še prisotni v sistemu). Problem pa je nastal pri potrjevanju alarma, saj se v podatkovno bazo zapišeta dva podatka. Prvi podatek je nepotrjen, drugi pa potrjen alarm. To pripelje do podvajanja nepotrjenih alarmov, saj se isti podatek zapiše takrat, ko je alarm nastal in takrat, ko se je alarm potrdil (slika 25). Težava je v samem nadzornem sistemu in načinu pošiljanja podatkov v podatkovno bazo. Podvajanje smo sprva hoteli rešiti z nastavitvami pošiljanja v nadzornem sistemu, a so

nastavitve zelo omejene in ne omogočajo filtriranja prometa. Ostalo nam je samo filtriranje podatkov v podatkovni bazi.

	A	B	C	D	E
1	start	finish	tag	opis2	alm_status
89	23.3.2012 1:28:38	23.3.2012 1:28:38	HP25_2_HH_A	HH alarm	CFN
103	23.3.2012 1:28:38	23.3.2012 1:28:51	HP25_2_HH_A	HH alarm	OK
104	23.3.2012 1:34:30	23.3.2012 1:34:30	HQ11_1_H_A	H alarm	CFN
107	23.3.2012 1:34:30	23.3.2012 1:34:53	HQ11_1_H_A	H alarm	OK
108	23.3.2012 1:37:07	23.3.2012 1:37:07	HD_200M2_SPEED_MONITOR	kontrolnik vrtljajev	CFN
118	23.3.2012 1:37:07	23.3.2012 1:39:13	HD_200M2_SPEED_MONITOR	kontrolnik vrtljajev	OK
120	23.3.2012 1:38:01	23.3.2012 1:38:01	HF63_FLOWH_A	Flow H	CFN
130	23.3.2012 1:38:01	23.3.2012 1:38:55	HF63_FLOWH_A	Flow H	OK
131	23.3.2012 1:40:38	23.3.2012 1:40:38	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	CFN
133	23.3.2012 1:40:38	23.3.2012 1:42:49	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	OK
134	23.3.2012 1:41:07	23.3.2012 1:41:07	HS1L25_1S1_A	Minimalen nivo	CFN
147	23.3.2012 1:41:07	23.3.2012 1:41:55	HS1L25_1S1_A	Minimalen nivo	OK
148	23.3.2012 1:41:51	23.3.2012 1:41:51	HD_200M2_LOCALMODE	preklopnik lokalno /avtomatsko	CFN
149	23.3.2012 1:41:51	23.3.2012 1:42:37	HD_200M2_LOCALMODE	preklopnik lokalno /avtomatsko	OK
150	23.3.2012 1:46:29	23.3.2012 1:46:29	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	CFN
151	23.3.2012 1:46:29	23.3.2012 1:47:04	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	OK
152	23.3.2012 1:46:55	23.3.2012 1:46:55	HP50_1_KILNHOOD_PRESSH_A	Press H	CFN

Slika 25: Podvojeni alarmi.

Na sliki 25 lepo vidimo podvajanje alarmov. Najprej nastopi vrstica z nepotrjenim alarmom. Prepoznamo jo po atributu `alm_status`. Nepotrjeni alarmi imajo vrednost atributa `alm_status` CFN (Change Of Normal), medtem ko imajo potrjeni alarmi vrednost OK. Nepotrjen alarm lahko prepoznamo tudi po tem, da ima atributa `start` in `finish` enaka. Sledi vrstica z odpravljenim alarmom.

Sprva se nam je rešitev zdela dokaj enostavna. Podvojene alarme bi filtrirali in nato izbrisali iz podatkovne baze. Tako bi naša podatkovna baza vsebovala samo pravilne podatke. Problem je žal bolj zahteven, saj je težko okarakterizirati podvojene alarme, ker ne obstaja oprijemljiv parameter, s katerim bi z gotovostjo pokazali na ponavljajoč alarm. S pregledom podatkov sicer hitro zaznamo podvojitev, a programu je potrebno točno določiti razliko med podvojenim in nepodvojenim alarmom. Na koncu smo le dobili vzvod po katerem bi problem rešili. Podvojitev namreč prepoznamo tudi po tem, da imata neodpravljen in odpravljen alarm enako vrednost atributa `start`. Datum in čas sta namreč enaka. S pomočjo tega pojava smo lahko v programu lovili podvojitev.

Problem smo začeli reševati na lažjem koncu in sicer pri odpravljenih alarmih. Težavo smo rešili dokaj enostavno, saj se ponovitve pojavljajo zgolj pri neodpravljeni alarmih. Nepotrjene alarme smo od drugih ločili z filtriranjem po atributu `alm_status`. Prikazali smo samo tiste, ki imajo vrednost atributa `alm_status` enak OK (slika 26).

1	start	finish	tag	opis2	alm_status
103	23.3.2012 1:28:38	23.3.2012 1:28:51	HP25_2_HH_A	HH alarm	OK
107	23.3.2012 1:34:30	23.3.2012 1:34:53	HQ11_1_H_A	H alarm	OK
118	23.3.2012 1:37:07	23.3.2012 1:39:13	HD_200M2_SPEED_MONITOR	kontrolnik vrtljajev	OK
130	23.3.2012 1:38:01	23.3.2012 1:38:55	HF63_FLOWH_A	Flow H	OK
133	23.3.2012 1:40:38	23.3.2012 1:42:49	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	OK
147	23.3.2012 1:41:07	23.3.2012 1:41:55	HS1L25_1S1_A	Minimalen nivo	OK
149	23.3.2012 1:41:51	23.3.2012 1:42:37	HD_200M2_LOCALMODE	preklopnik lokalno /avtomatsko	OK
151	23.3.2012 1:46:29	23.3.2012 1:47:04	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	OK
154	23.3.2012 1:46:55	23.3.2012 1:47:00	HP50_1_KILNHOOD_PRESSH_A	Press H	OK

Slika 26: Prikaz odpravljenih alarmov.

Ostal nam je še prikaz nepotrjenih in vseh alarmov. Ko bo ta problem rešen, bomo lažje prikazali vse alarme. V množici neodpravljenih alarmov imamo nepotrjene alarme, ki so lahko:

- pravilen podatek ali
- podvojena informacija pri potrditvi alarma.

Pri analizi podatkov smo ugotovili vzorec, po katerem se lahko orientiramo pri ločevanju pravih in lahko bi rekli »lažnivih« podatkov. V kolikor je v množici neodpravljenih alarmov število enakih imen (atribut tag) določenega alarma liho, potem velja, da je alarm s tem imenom resnično neodpravljen. Vsi ostali alarmi v tej množici so ponovitve in jih lahko zavržemo.

Sedaj lahko to ugotovitev zapišemo še kot povpraševanje. Za to smo morali uporabiti agregacijske funkcije in metodo gnezdenja stavkov SQL. Prav nam je prišlo tudi znanje iz množic. Stavek je v končni fazi izgledal nekako takole.

```
SELECT start_date, start_time, last_date, last_time, alm_status, tag,
opis1, opis2
FROM alarmi_spklv1
WHERE start in (
    SELECT start
    FROM alarmi_spklv1
    WHERE + pogoj + AND
        msg_type = 'ALARM'
    GROUP BY start, tag
    HAVING count(tag) % 2 = 1) AND
    alm_status = 'CFN'
ORDER BY start
```

V zgornji kodi imamo ugnježena select stavka. Notranji stavek poskrbi, da se upoštevajo pogoji izbrani v vmesniku in izbris duplikatov. Podatke smo morali grupirati po vrsticah start in tag. To smo dosegli z ukazom GROUP BY start, tag. Po agregaciji pa smo z ukazom HAVING count (tag) % 2 = 1 prikazali samo alarme ki se ponovijo liho krat. Zunanji stavek poskrbi za prikaz ostalih atributov alarmov, ki jih vrne notranji stavek.

Sedaj imamo pravilno prikazane nepotrjene alarme (slika 27). S tem smo rešili dobršni del problema.

Nazaj Html Pdf Sestavi poročilo

Alarmi za Področje:			OD: 26.3.2012 09:41:36	DO: 25.5.2012 10:41:36
PEC	Tip alarmov: neodpravljeni			
Začetek	Konec	Alarm	Opis napake	Opis naprave
26.3.2012	10:52	VF10001_ETOTAL	Skupna napaka	Merlec pretoka zraka - skupno
26.3.2012	10:52	VF10001_HALARM	H alarm	Merlec pretoka zraka - skupno
26.3.2012	10:52	VF10001_HHALARM	HH alarm	Merlec pretoka zraka - skupno
26.3.2012	10:52	VF10002_ETOTAL	Skupna napaka	Merlec pretoka zraka - sever
26.3.2012	10:52	VF10002_HALARM	H alarm	Merlec pretoka zraka - sever
26.3.2012	10:52	VF10002_HHALARM	HH alarm	Merlec pretoka zraka - sever
26.3.2012	11:19	E1_1_1D1_EUNITERROR	Unit error alarm	GCT 1 control unit
26.3.2012	11:19	E2_1_1D1_EUNITERROR	Unit error alarm	GCT 2 control unit
26.3.2012	16:07	HT27_1_L_A	L alarm	Kalcinator - sredina
27.3.2012	00:45	HS2L25_1S1_A	Minimalen nivo	Pogon polževk - S transport
27.3.2012	23:13	HS30_KILN60SPEED_A	Pec gre na 60% obratov zaradi blokacij	Biokacije
28.3.2012	00:59	HS30_KILN60SPEED_A	Pec gre na 60% obratov zaradi blokacij	Biokacije
28.3.2012	04:59	HS30_KILN60SPEED_A	Pec gre na 60% obratov zaradi blokacij	Biokacije
29.3.2012	18:51	HT27_1_L_A	L alarm	Kalcinator - sredina
30.3.2012	09:00	HP29_H_A	H alarm	Tesnlmi ventilator vstop v peč
30.3.2012	12:15	HS1L25_1S1_A	Minimalen nivo	Pogon polževk - Z transport
30.3.2012	12:52	VF10001_ETOTAL	Skupna napaka	Merlec pretoka zraka - skupno
30.3.2012	12:52	VF10001_HALARM	H alarm	Merlec pretoka zraka - skupno
30.3.2012	12:52	VF10001_HHALARM	HH alarm	Merlec pretoka zraka - skupno
30.3.2012	14:06	H_11_1X1_D_ANALYSERFAULT1_A	Analysers fault 1	Analiza plinov - izstop iz IT

Slika 27: Prikaz nepotrjenih alarmov.

Ker imajo nepotrjeni alarmi enako vrednost začetka in konca, izpisujemo samo čas in datum začetka. Čas in datum konca pa pustimo prazen. Tako je tudi prikaz bolj logičen, saj alarm, ki se še ni zaključil, ne more imeti določenega časa konca. S tem bo v končni fazi tudi uporabnik ločeval nepotrjene alarme od potrjenih.

Ostal nam je samo še prikaz vseh alarmov. Ker znamo prikazati nepotrjene in potrjene alarme, je prikaz vseh dokaj enostaven - alarme obeh množic združimo. Zgornjemu povpraševanju dodamo pogoj, ki zahteva prikaz potrjenih alarmov. V spodnjem povpraševanju je dodan pogoj obarvan z zeleno barvo, ostala koda je enaka povpraševanju po nepotrjenih alarmih.

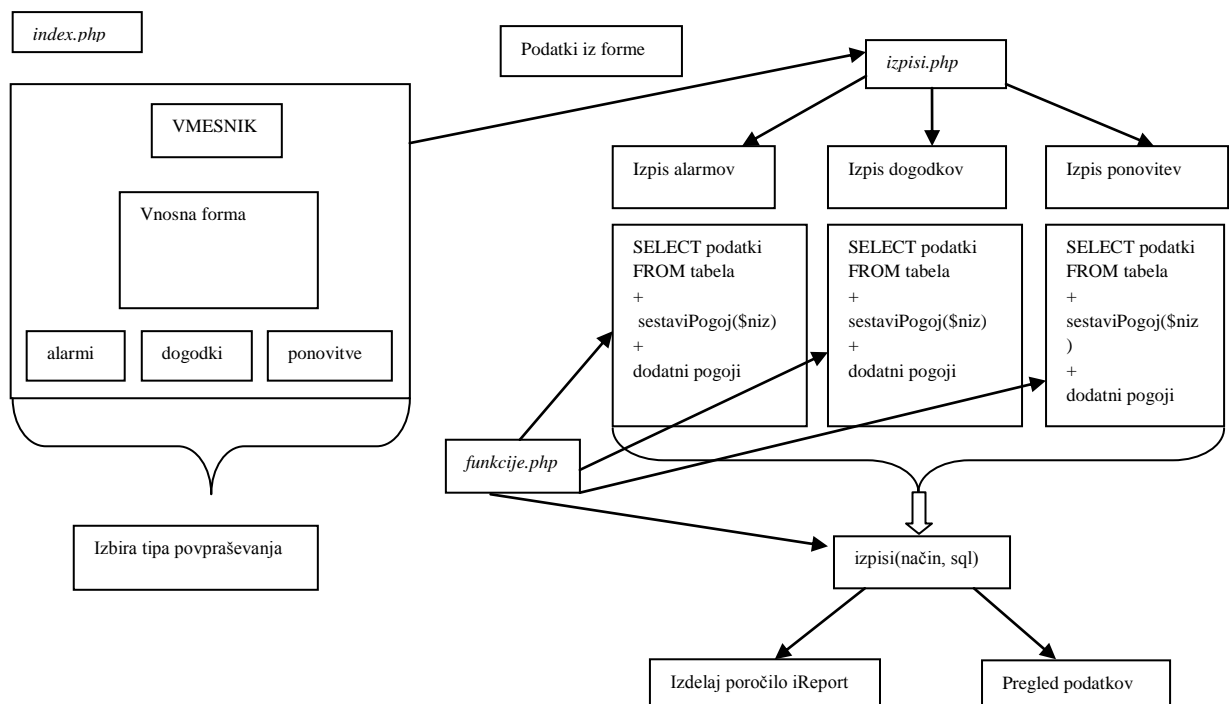
```
SELECT start_date, start_time, last_date, last_time, alm_status, tag,
opisl, opis2
FROM alarmi_spklv1
WHERE (start in (
SELECT start
FROM alarmi_spklv1
WHERE + pogoj + AND
msg_type = 'ALARM'
GROUP BY start, tag
HAVING count(tag) % 2 = 1) AND
alm_status = 'CFN') OR
(+ pogoj + AND
msg_type = 'ALARM' AND
alm_status = 'OK')
ORDER BY start
```

26.3.2012	10:19	26.3.2012	10:21	HD_455_PULL_WIRE_RIGHT	desna potezna vrstica	ROLLER CONVEYOR HD_455 PULL WIRE SWITCH RIGHT
26.3.2012	10:19	26.3.2012	10:22	HT15_H_A	H alarm	Ciklon 5 - vrh
26.3.2012	10:24	26.3.2012	10:25	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo	LACK OF TYRES SEPARACION
26.3.2012	10:24	26.3.2012	10:25	HD_0_LACK_OF_TYRES	pomanjkanje gum	LACK OF TYRES
26.3.2012	10:28	26.3.2012	10:28	HP50_1_KILNHOOD_PRESSH_A	Press H	Razlika tlakov v hlad. klinkerja
26.3.2012	10:28	26.3.2012	10:28	HP50_1_KILNHOOD_PRESSHH_A	Press HH	Razlika tlakov v hlad. klinkerja
26.3.2012	10:33	26.3.2012	10:34	HQ11_1_H_A	H alarm	Analiza plinov - izstop iz IT
26.3.2012	10:36	26.3.2012	10:36	HQ29_2_H_A	H alarm	Analizator plinov izstop vstop v peč
26.3.2012	10:39	26.3.2012	10:42	H_02_4Y1_OPEN_A	Napaka pri odpiranju ventila (se ne odpr	Segment 4 silosa LM2 (Kanal) - pozicija 1
26.3.2012	10:47	26.3.2012	10:52	HT15_H_A	H alarm	Ciklon 5 - vrh
26.3.2012	10:52	27.3.2012	07:36	VF10001_ETOTAL	Skupna napaka	Merilec pretoka zraka - skupno
26.3.2012	10:52	27.3.2012	07:36	VF10001_HALARM	H alarm	Merilec pretoka zraka - skupno
26.3.2012	10:52	27.3.2012	07:36	VF10001_HHALARM	HH alarm	Merilec pretoka zraka - skupno
26.3.2012	10:52			VF10001_ETOTAL	Skupna napaka	Merilec pretoka zraka - skupno
26.3.2012	10:52			VF10001_HALARM	H alarm	Merilec pretoka zraka - skupno
26.3.2012	10:52			VF10001_HHALARM	HH alarm	Merilec pretoka zraka - skupno
26.3.2012	10:52	27.3.2012	07:36	VF10002_HALARM	H alarm	Merilec pretoka zraka - sever
26.3.2012	10:52	27.3.2012	07:36	VF10002_HHALARM	HH alarm	Merilec pretoka zraka - sever
26.3.2012	10:52	27.3.2012	07:36	VF10002_ETOTAL	Skupna napaka	Merilec pretoka zraka - sever
26.3.2012	10:52			VF10002_ETOTAL	Skupna napaka	Merilec pretoka zraka - sever
26.3.2012	10:52			VF10002_HALARM	H alarm	Merilec pretoka zraka - sever
26.3.2012	10:52			VF10002_HHALARM	HH alarm	Merilec pretoka zraka - sever
26.3.2012	10:57	26.3.2012	10:57	HQ29_2_H_A	H alarm	Analizator plinov izstop vstop v peč
26.3.2012	10:58	26.3.2012	10:59	HDG90_6S1	Vrata nad Hd. 1300 zaprta (odpadni mat.)	VRATA NAD HD_1300 ZAPRTA (ODPADNI MATERIAL)
26.3.2012	11:14	26.3.2012	11:14	HQ29_2_H_A	H alarm	Analizator plinov izstop vstop v peč
26.3.2012	11:15	26.3.2012	11:16	E2_51_IMI_EPOWERON	Power switch off alarm	high pressure pump 2
26.3.2012	11:15	26.3.2012	11:17	HD_RAZNO_REMOVE_TYRES	odstrani gume	REMOVE TYRES DISCHARGE PLACE

Slika 28: Prikaz vseh alarmov.

Pri prikazu vseh alarmov takoj prepoznamo potrjene in nepotrjene alarme (slika 28). Nepotrjeni alarmi nimajo časa in datuma konca. Sedaj imamo pravilno povpraševanje za vse tipe alarmov.

Tukaj se naše izdelovanje aplikacije konča. Aplikacija je pripravljena na testiranje. Za lažji pregled delovanja aplikacije smo predstavili diagram, ki predstavlja delovanje komponent aplikacije in pošiljanje podatkov med njimi (slika 29).



Slika 29: Diagram delovanje aplikacije.

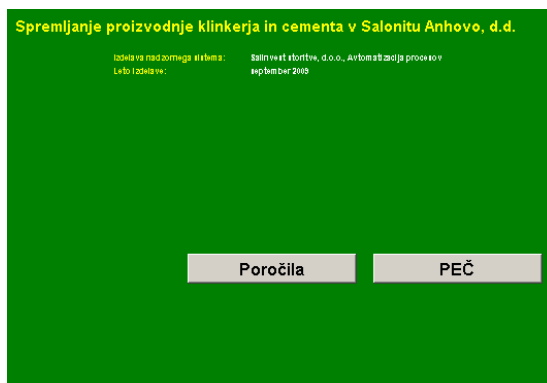
V nadaljevanju poglavja bomo prikazali, kako smo aplikacijo za izdelavo poročil integrirali v nadzorni sistem SCADA in kako smo do nje dostopali.

4.6 Integracija aplikacije v SCADA nadzorni sistem

Operater, ki nadzira stanje proizvodnega procesa preko nadzornega sistema in v njem izvršuje ukaze, nima dostopa do drugih programov operacijskega sistema. Zato je bilo omogočanje uporabe aplikacije v nadzornem sistemu visoka prioriteta pri izdelavi aplikacije.

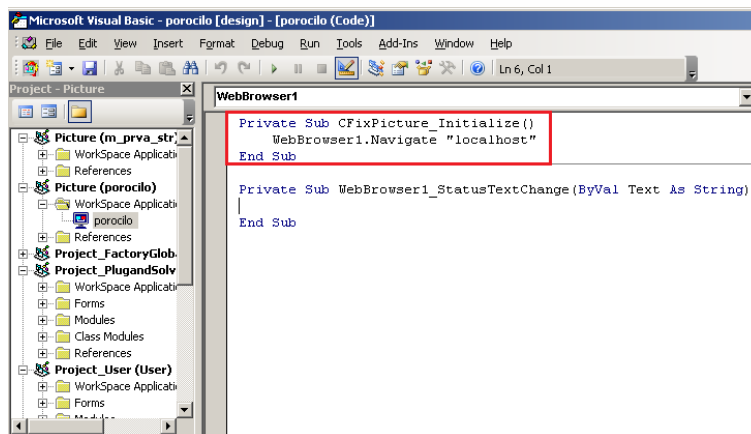
Nadzorni sistem omogoča obilo dodatkov. Eden izmed njih je tudi dodajanje OCX [22] oziroma ActiveX dodatkov. OCX izhaja iz Microsoftove tehnologije OLE (ang. Object Linking and Embedding), ki omogoča vgrajevanje in povezovanje dokumentov in ostalih objektov. Med drugimi dodatki smo imeli na voljo tudi OCX komponento programa Microsoft Explorer, ki omogoča vgradnjo brskalnikovega okna v sliko nadzornega sistema.

Pri zagonu SCADA odjemalca in programa *Workspace* se nam odpre uvodno okno nadzornega sistema (slika 30). Na uvodno stran smo dodali gumb *Poročila*, ki odpre novo sliko z vgrajenim oknom brskalnika.



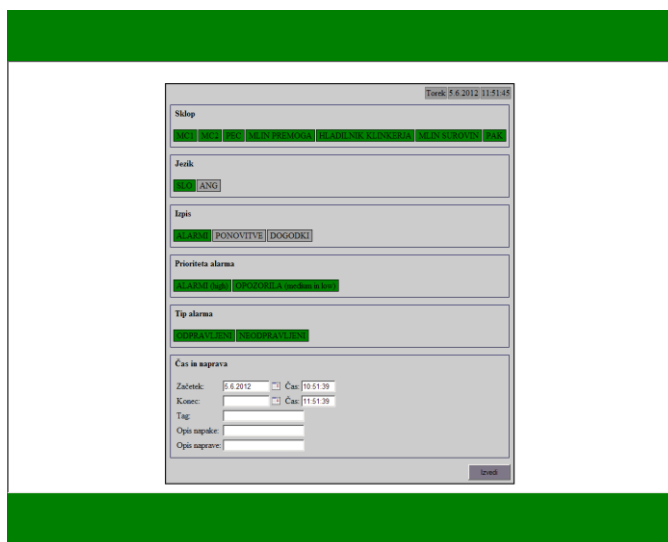
Slika 30: Uvodna stran nadzornega sistema SCADA.

Kreirali smo novo sliko in nanjo dodali OCX komponento programa Internet Explorer. Da bi se v oknu brskalnika naložila aplikacija za izdelavo poročil smo morali navesti naslov Apache strežnika. To smo storili s pomočjo VBA skripte (slika 31).



Slika 31: VBA urejevalnik.

S pomočjo ukaza `Navigate` v funkciji `CFixPicture_Initialize()`, ki se zažene ob inicializaciji okna, smo brskalniku določili katero stran naj naloži ob zagonu. Ker je datoteka `index.php` našega projekta postavljena v korenski direktorij strežnika, je potrebno določiti samo ime računalnika na katerem je strežnik z aplikacijo. V našem primeru je to lokalni računalnik zato smo kot argument ukaza podali niz `localhost`. V kolikor bi do aplikacije dostopali iz drugega računalnika bi morali kot argument ukaza nastaviti ime računalnika na katerem je aplikacija, ki je v našem primeru `PECV8`. Po ponovnem zagonu slike nadzornega sistema, se je v oknu brskalnika pojavila naša aplikacija (slika 32).



Slika 32: Prikaz delovanje aplikacije v nadzornem sistemu.

Sedaj je aplikacija pripravljena tudi za uporabo v nadzornem sistemu. Pri vgraditvi smo želeli prikazati enega izmed možnih načinov vpeljave spletne aplikacije v nadzorni sistem. Naš namen je bil testiranje delovanja aplikacije znotraj nadzornega sistema SCADA. Za dokončno umestitev aplikacije v SCADA sistem, bi se morali posvetovati z nadrejenimi, ki skrbijo za pravilno delovanje nadzornega sistema.

5 Pregled in uporaba aplikacije

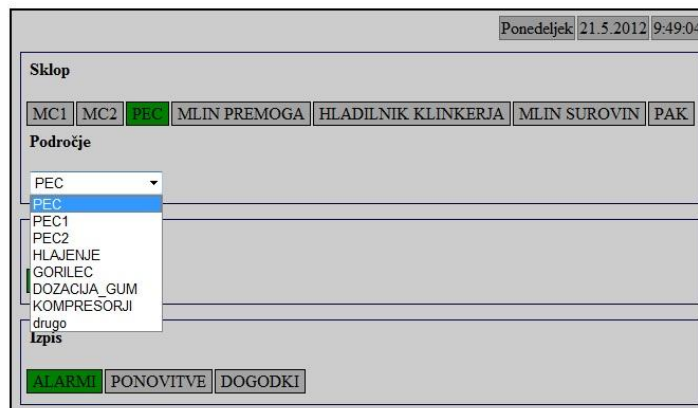
Naša aplikacija živi na strežniku. Dostopna je vsem računalnikom v omrežju. Aplikacija bi se lahko uporabljala tudi na mobilnih napravah, a smo uporabo in testiranje omejili na računalnik. Do aplikacije dostopamo preko IP naslova ali domene strežnika, na katerem aplikacija živi. V našem primeru je to ime računalnika PECV8. Za domene skrbi DNS strežnik podjetja. Do aplikacije lahko dostopoma na dva načina. Prvi je preko nadzornega sistema, drugi pa preko brskalnika. Ob zagonu se nam najprej prikaže uporabniški vmesnik (slika 33).

Slika 33: Končna verzija uporabniškega vmesnika.

V končni verziji uporabniškega vmesnika smo upoštevali vse zahteve uporabnikov. Glede na starejšo verzijo vmesnika, je nov vmesnik bolj pregleden in uporabniku bolj prijazen. Privzete vrednosti vmesnika omogočajo uporabniku hitro izvrševanje povpraševanj, saj so nastavljene na vrednosti po katerih se najbolj pogosto povprašuje.

V zgornjem desnem kotu smo postavili informacije o datumu in uri, ki uporabniku pomagajo pri pravilnem vnosu časa in datuma v vnosno masko z datumom in časom. V nadaljevanju imamo šest ločenih področij. Vsako nosi informacijo o določenem delu, ki na koncu sestavlja celoto povpraševanja. Izbrane vrednosti so obarvane z zeleno barvo, neizbrane pa z sivo. Tako lahko uporabnik hitro opazi, kateri elementi so izbrani in kateri niso. Ob zagonu

aplikacije ima vmesnik izbrane privzete vrednosti, tako, da v večini primerov ima uporabnik malo dela z izbiro. Nov vmesnik omogoča tudi izbiro več elementov v oknih Sklop, Prioriteta alarma in Tip alarma. Starejša verzija aplikacije je omogočala izbiro le enega elementa. Ta funkcionalnost ima največji poudarek pri izbiri sklopov, saj ima sedaj uporabnik izbiro več kot enega sklopa, kar prej ni bilo mogoče. Pri izbiri enega sklopa, pa se uporabniku prikaže izbirni meni za izbiro področja. Tako lahko še bolj podrobno določi področje, iz katerega si bo ogledal izbrane informacije (slika 34).



Slika 34: Izbira področja v vmesniku.

Ko uporabnik nastavi izbrane parametre, klikne gumb Izvedi in s tem izvede povpraševanje. Glede na izbran tip izpisa se v html dokumentu izpišejo podatki iz podatkovne baze. Zaradi boljše preglednosti smo vsako drugo vrstico obarvali. V zgornjem delu zaslona imamo opravilno vrstico, ki uporabniku omogoča vrnitev nazaj na vmesnik in s tem ponovno izbiro parametrov ali možnost izdelave poročil (slika 35 zgoraj desno). Pred izpisom vsebine uporabniku izpišemo glavo poročila, v kateri so najbolj pomembne informacije o vsebini.

Nazaj		Hudi Pdf Sestavi poročilo	
Alarmi za Področje:		OD: 29.3.2012 08:29:48	DO: 30.3.2012 09:29:48
Ta alarmov: vsi			
Začetek	Konec	Alarm	Opis napake
29.3.2012 08:36	29.3.2012 08:37	HT79_BEFHETEMPH_A	Skupen H alarm dveh meritev
29.3.2012 08:36	29.3.2012 08:37	HT79_1_CDBEFHE_TEMP_A	temperature control duct before heat exc
29.3.2012 08:40	29.3.2012 08:42	HT15_H_A	H alarm
29.3.2012 08:48	29.3.2012 14:26	H_29_IM_D_LOW&MPC&GASFLOW_A	Nizki pretoki zajema zraka < 10 t/h
29.3.2012 08:48	29.3.2012 08:48	HD_HAZOO_REMOVE_TYRES	odstrani gume
29.3.2012 08:54	29.3.2012 08:59	HD_435_VISOKA_GUMA	Previsoka guma na HD 435
29.3.2012 08:55	29.3.2012 08:56	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo
29.3.2012 08:57	29.3.2012 08:59	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo
29.3.2012 09:01	29.3.2012 09:01	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo
29.3.2012 09:02	29.3.2012 09:03	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo
29.3.2012 09:02	29.3.2012 09:03	HD_0_LACK_OF_TYRES	pomanjkanje gum
29.3.2012 09:12	29.3.2012 09:12	HQ11_1_H_A	H alarm
29.3.2012 09:21	29.3.2012 09:21	HD_1060_LOCALMODE	preklonnik lokalno avtomatsko
29.3.2012 09:21	29.3.2012 09:21	HD_1020_LOCALMODE	preklonnik lokalno avtomatsko
29.3.2012 09:24	29.3.2012 09:24	HQ11_3_L_A	L alarm
29.3.2012 09:32	29.3.2012 09:34	HT79_BEFHETEMPH_A	Skupen H alarm dveh meritev
29.3.2012 09:32	29.3.2012 09:34	HT79_1_CDBEFHE_TEMP_A	temperature control duct before heat exc
29.3.2012 09:37	29.3.2012 09:40	HPQ_BLASTER_L_A	L alarm
29.3.2012 09:38	29.3.2012 09:39	HPQ_BLASTER_LL_A	Ll alarm
29.3.2012 09:43	29.3.2012 09:43	HQ11_3_L_A	L alarm
29.3.2012 09:44	29.3.2012 09:45	HQ11_3_L_A	L alarm
29.3.2012 09:45	29.3.2012 09:45	HP50_1_KILNHOOD_PRESSH_A	Press H
29.3.2012 09:45	29.3.2012 09:45	HP50_1_KILNHOOD_PRESSHH_A	Press HH
29.3.2012 09:46	29.3.2012 09:47	HQ11_3_L_A	L alarm
29.3.2012 09:59	29.3.2012 10:00	HT79_BEFHETEMPH_A	Skupen H alarm dveh meritev
29.3.2012 09:59	29.3.2012 10:00	HT79_1_CDBEFHE_TEMP_A	temperature control duct before heat exc
29.3.2012 10:00	29.3.2012 10:00	HQ11_1_L_A	L alarm
29.3.2012 10:04	29.3.2012 10:04	HQ11_3_L_A	L alarm
29.3.2012 10:05	29.3.2012 10:07	HT15_H_A	H alarm
29.3.2012 10:07	29.3.2012 10:26	HD_GROUP_OPERATION_STOP	zaustavitev operacij
29.3.2012 10:07	29.3.2012 10:26	HD_425_PULL_WIRE_LEFT	leva potemna vrstica
29.3.2012 10:08	29.3.2012 10:31	HD_0_LACK_OF_TYRES_SEPARACION	pomanjkanje gum za separacijo
29.3.2012 10:10	29.3.2012 10:10	HD_425_PULL_WIRE_RIGHT	desna potemna vrstica
29.3.2012 10:10	29.3.2012 10:10	HD_420_LOCALMODE	preklonnik lokalno avtomatsko
29.3.2012 10:12	29.3.2012 10:12	HD_425_PULL_WIRE_RIGHT	desna potemna vrstica
29.3.2012 10:13	29.3.2012 10:27	HD_435_VISOKA_GUMA	Previsoka guma na HD 435
29.3.2012 10:14	29.3.2012 10:14	HD_430_LOCALMODE	preklonnik lokalno avtomatsko

Slika 35: Izpis alarmov.

V kolikor se uporabnik odloči izpis natisniti, ima vedno možnost ustvariti primerno poročilo za tiskanje. Uporabniku smo dali na izbiro dve možnosti izvoza poročila (izvoz v pdf obliki in html dokumentu). Primarni je izvoz v pdf dokument, kateri je najbolj priporočljiv, saj zagotavlja pravilno oblikovanje dokumenta. Za prikazovanje vsebine pdf smo uporabili Adobov Pdf Reader. Program omogoča prikazovanje vsebin pdf kar v samem brskalniku. Pri izvozu poročila v pdf dokument velja opozoriti na težave pri odpiranju pdf vsebine v 64-bitnih brskalnikih. Težavo smo opazili tako pri Microsoftovem Explorerju, kot tudi pri Mozillini 64-bitni različici Firefoxa, ki jo imenujejo Waterfox. Zato priporočamo uporabo zgolj 32-bitne brskalnike. V podjetju tega problema nismo zaznali, saj uporabljajo zgolj 32-bitne operacijske sisteme in s tem 32-bitne brskalnike.

Problem smo zaznali tudi pri integraciji z nadzornim sistemom. Bralnik pdf dokumentov namreč zavzame celoten zaslon brskalnika in tako prekrije ostalo vsebino strani. To nam je onemogočalo prikaz gumba za vrnitev na začetno stran. Pri brskalnikih bi ta problem rešili z uporabo ukazov brskalnika, pri vtičniku v nadzornem sistemu pa te opcije nimamo. Rešitev problema smo izvedli z uporabo novega okna, namenjenega prikazovanju zgolj poročila. Pri izvedbi smo uporabili html gradnik `iframe`, ki omogoča vgrajevanje več html dokumentov v trenutni html dokument. Tako smo uspeli na isti strani prikazati opravilno vrstico z gumbom za vrnitev na začetno stran in vsebino poročila. Uporabnik lahko nato vsebino shrani ali natisne s pomočjo funkcij bralnika pdf vsebin. Na sliki 36 smo prikazali primer izpisa poročila alarmov v pdf dokumentu.

Domov

Poročilo alarmov

Alarmi za področje: PEC od : 29.3.2012 10:38:56 do : 30.3.2012 11:38:56

Začetek	Konec	Alarm	Opis napake	Opis naprave
29.3.2012 10:41:48	29.3.2012 10:44:24	HT9_BEPHETEMPH_A	Skupen H alarm dveh mestev	Temperatura pred hladilnim stolpom
29.3.2012 10:41:48	29.3.2012 10:44:24	HT9_1_CDBEPHE_TEMPH_A	temperature control duct before heat etc.	Temperatura pred hladilnim stolpom
29.3.2012 10:48:26	29.3.2012 10:49:48	HT15_H_A	H alarm	Čalson 5 - vzh
29.3.2012 10:58:19	29.3.2012 11:00:55	HT15_H_A	H alarm	Čalson 5 - vzh
29.3.2012 11:01:53	29.3.2012 11:16:25	HT90_KELNWOOD_TEMPH_A	Temp L	Temperatura izstop iz peči
29.3.2012 11:06:33	29.3.2012 11:06:40	HQ11_3_1_A	L alarm	0
29.3.2012 11:08:05	29.3.2012 11:08:16	HQ11_3_1_A	L alarm	0
29.3.2012 11:08:49	29.3.2012 11:09:03	HQ11_3_1_A	L alarm	0
29.3.2012 11:08:49	29.3.2012 11:09:34	HQ11_3_1_A	L alarm	0
29.3.2012 11:10:41	29.3.2012 11:10:55	HQ11_3_1_A	L alarm	0
29.3.2012 11:11:38	29.3.2012 11:12:25	HQ11_3_1_A	L alarm	0
29.3.2012 11:12:23	29.3.2012 11:16:00	HS1P96_H_A	H Alarm	Hidravlično olje - Z
29.3.2012 11:12:33	29.3.2012 11:15:34	HQ11_3_1_A	L alarm	0
29.3.2012 11:12:48	29.3.2012 17:12:49	HS2L15_151_A	Minimalen nivo	Pogon polbenk - S/transport
29.3.2012 11:13:45	29.3.2012 11:15:31	HPQ_BLAISTER_1_A	L alarm	Blaster
29.3.2012 11:15:10	29.3.2012 11:15:21	HPQ_BLAISTER_LL_A	LL alarm	Blaster
29.3.2012 11:15:10	29.3.2012 11:20:28	HT15_H_A	H alarm	Čalson 5 - vzh
29.3.2012 11:16:36	29.3.2012 11:17:34	HSP02_1_A	L Alarm	Vpik v gorilce petnik
29.3.2012 11:17:07	29.3.2012 11:17:29	HQ11_1_H_A	H alarm	Analiza plinov - izstop iz IT
29.3.2012 11:18:29	29.3.2012 11:19:06	HS1P96_H_A	H Alarm	Hidravlično olje - S
29.3.2012 11:19:05	29.3.2012 11:19:50	HQ11_3_1_A	L alarm	0
29.3.2012 11:27:56	29.3.2012 11:27:59	HSP02_1_A	L Alarm	Vpik v gorilce petnik
29.3.2012 11:29:26	29.3.2012 11:29:47	HD690_651	Vrata nad Hd_1300 zapeta (odporna mast)	VRATA NAD HD_1300 ZAPRTA (ODPADNI
29.3.2012 11:33:45	29.3.2012 11:33:57	HD_700_ACCUMULATION	akumulacija	CONVEYOR WITH CORRUGATED SIDE WALLS
29.3.2012 11:34:15	29.3.2012 11:34:53	HQ11_3_1_A	L alarm	0
29.3.2012 11:38:06	29.3.2012 11:38:25	HQ11_3_1_A	L alarm	0
29.3.2012 11:39:54	29.3.2012 11:41:44	HD_460_LOCALMODE	preklopnik lokalno /avtomatsko	ROLLER CONVEYOR_HD_460 LOCAL MODE

Stran 1 od 28

Slika 36: Prikaz poročila alarmov v pdf dokumentu.

Kot drugo možnost smo uporabniku ponudili izvoz poročila v html dokument. To možnost bo uporabnik uporabil, v primeru, da v računalniku nima nameščenega bralnika pdf vsebin. Temu načinu se je bolje izogniti, saj v nekaterih primerih ne zagotavlja oblikovno pravilnega izpisa vsebin. V primeru, da se uporabnik vselej odloči za ta način, mora sam nastaviti parametre tiskanja in nastavitve postavitve strani. Na sliki 37 bomo prikazali izgled poročila v html dokumentu.

Domov

Poročilo alarmov

Začetek	Konec	Alarm	Opis napake	Opis naprave
28.3.2012 16:16:54	28.3.2012 16:16:56	HNPO0_HALARM	H ALARM	ALARM H
28.3.2012 16:17:56	28.3.2012 18:28:15	VP10101_ETOTAL	Slupna napaka	Merilec šifra v stanesi posodi
28.3.2012 16:17:56	28.3.2012 18:28:15	VP10101_LALARM	L alarm	Merilec šifra v stanesi posodi
28.3.2012 16:18:05	28.3.2012 16:18:09	HNPO0_HALARM	H ALARM	ALARM H
28.3.2012 16:18:05	28.3.2012 16:18:09	HNPO0_HALARM	H ALARM	ALARM H
28.3.2012 16:19:02	28.3.2012 16:21:08	HP09_483_PRESMIND_A	HP09-483: Tisk zrak na dovodu k tehtajo	Talstina posoda
28.3.2012 16:20:18	28.3.2012 16:27:37	HNPO0_HALARM	H ALARM	ALARM H
28.3.2012 16:28:36	28.3.2012 16:28:40	HP00_1_KILNHOOD_PRESH_A	Press H	Razlika tlakov v hlad. Kisljerja
28.3.2012 16:28:36	28.3.2012 16:28:40	HP00_1_KILNHOOD_PRESHH_A	Press HH	Razlika tlakov v hlad. Kisljerja
28.3.2012 16:30:25	28.3.2012 16:31:02	EIP01_HALARM	H Alarm	H Alarm
28.3.2012 16:33:38	28.3.2012 16:33:40	HNPO0_HALARM	H ALARM	ALARM H
28.3.2012 16:34:35	28.3.2012 16:34:36	HNPO0_HALARM	H ALARM	ALARM H
28.3.2012 16:35:11	28.3.2012 16:35:26	HQ11_1_H_A	H alarm	Aerolisa plinov - izstop iz IT
28.3.2012 16:35:20	28.3.2012 16:35:21	HS1P11_381_PMAN_A	deltaP H alarm	DeltaP filter silosa 5
28.3.2012 16:45:54	28.3.2012 16:46:07	HD_RAZNO_REMOVE_TYRES	odstrani gume	REMOVE TYRES DISCHARGE PLACE
28.3.2012 16:46:18	28.3.2012 16:46:20	HNPO0_HALARM	H ALARM	ALARM H
28.3.2012 16:54:45	28.3.2012 17:01:05	HD_RAZNO_REMOVE_TYRES	odstrani gume	REMOVE TYRES DISCHARGE PLACE
28.3.2012 16:54:48	28.3.2012 16:54:50	HNPO0_HALARM	H ALARM	ALARM H
28.3.2012 16:54:48	28.3.2012 16:54:50	HNPO0_HALARM	H ALARM	ALARM H
28.3.2012 16:56:19	28.3.2012 16:56:22	HP00_1_KILNHOOD_PRESH_A	Press H	Razlika tlakov v hlad. Kisljerja
28.3.2012 16:56:19	28.3.2012 16:56:22	HP00_1_KILNHOOD_PRESHH_A	Press HH	Razlika tlakov v hlad. Kisljerja
28.3.2012 17:00:08	28.3.2012 17:00:28	HQ11_3_L_A	L alarm	0
28.3.2012 17:00:12	28.3.2012 17:50:37	HT11_H_A	H alarm	Čiščenje 5 - vrh
28.3.2012 17:01:14	28.3.2012 17:03:07	HD_RAZNO_REMOVE_TYRES	odstrani gume	REMOVE TYRES DISCHARGE PLACE
28.3.2012 17:02:44	28.3.2012 17:03:00	HP29_H_A	H alarm	Tesnilni ventilator vstop v pee
28.3.2012 17:02:44	28.3.2012 17:02:51	HP00_1_KILNHOOD_PRESH_A	Press H	Razlika tlakov v hlad. Kisljerja
28.3.2012 17:02:44	28.3.2012 17:02:51	HP00_1_KILNHOOD_PRESHH_A	Press HH	Razlika tlakov v hlad. Kisljerja

Stran 1 od 53

Slika 37: Prikaz poročila alarmov v html dokumentu.

6 Sklepne ugotovitve

V diplomskem delu smo uspešno rešili problem poročanja v nadzornem sistemu SCADA. Aplikacija sedaj omogoča uporabniku veliko možnosti pri povpraševanju, hiter pregled podatkov in izpis podatkov v obliki poročila. Navdušeni smo bili nad hitrostjo izvajanja operacij in izgradnji poročil. Zadovoljni smo bili z odzivom uporabnikov v podjetju, ki so pozitivno ocenili izgled vmesnika, izpis podatkov ter boljši dostop aplikacije. Predvsem so bili zadovoljni z grafično podobo poročil. Sicer aplikacija še ni pripravljena za vsakodnevno uporabo v podjetju, saj bi morali v ta namen zajeti tudi podatke ostalih sklopov proizvodnega procesa. Smo pa podjetju podali dobro smernico možne rešitve prikazovanja podatkov o alarmih in dogodkih.

Vedno bolj strmimo k kvalitetnejšim informacijam. Tudi v naši aplikaciji smo opazili, da bi nekatere informacije lahko bolje predstavili. Predvsem pri dogodkih bi želeli podatke še bolj razčleniti in uporabniku tako omogočiti še bolj precizno povpraševanje. Da bi to omogočili, bi morali narediti veliko sprememb na področju načina shranjevanja podatkov oziroma konfiguraciji podatkovnih bazah. Poseči bi morali tudi izven področja naše diplomske naloge, saj je vir vseh informacij nadzorni sistem. Pri spreminjanju procesne podatkovne baze nadzornega sistema bi morali spremembe dobro načrtovati, saj bi morali podatke predstaviti tako, da bi bili dobri tako za našo aplikacijo, kot za slike nadzornega sistema.

Poleg sprememb procesne podatkovne baze bi morali izboljšati tudi našo relacijsko bazo. Pri tem ciljamo predvsem na področje varnosti podatkov. V kolikor bi podatkovno bazo selili na strežnik, bi morali vzpostaviti varnostno kopiranje in podvojevanje podatkov.

Tudi samo aplikacijo bi lahko nadgradili, da bi prikazovala boljše in bolj ažurne informacije. Poleg alarmov in dogodkov bi v prikazovanje lahko vključili tudi meritve. Te se hranijo na strežniku iHistorian. Preučiti bi morali možnost pridobivanja podatkov iz strežnika in format podatkov. Orodje iReport ima pester izbor grafov, tako da bi dosedanja poročila lahko obogatili tudi z grafi meritev. Tudi v aplikacijo bi lahko vpeljali varnostne mehanizme, ki bi omogočali prijavo v sistem. S tem bi izboljšali varnost aplikacije in boljši nadzor nad uporabniki.

Med izdelavo aplikacije smo naleteli na nemalo ovir. Spoznati smo se morali z novimi tehnologijami in osvojiti znanje programskih jezikov. Največjo težavo smo imeli pri izbiri in vpeljavi orodja za izdelavo poročil v aplikacijo, kar nam je vzelo veliko časa, predvsem pa truda. Cilj diplomske naloge je bil dosežen. S tem smo pridobili nova programerska znanja, predvsem pa novo izkušnjo v življenju. Četudi se nekateri problemi na začetku zdijo nerešljivi, jih lahko z voljo in trudom uspešno rešimo.

Literatura in viri

- [1] Salonit Anhovo, d.d, *Predstavitev ob 80-letnici Salonita Anhovo*, Anhovo: Salonit Anhovo, d.d., 2001.
- [2] Smo delniška družba, brošura o podjetju Salonit Anhovo, 1999.
- [3] Jože Volfand, "Monitoring izvajajo zunanje inštitucije", EOL, št. 60, strani 44 in 45, 2011.
- [4] Ge Fanuc, *Proficy™ HMI/SCADA iFIX Fundamentals*, Ge Fanuc Automation North America, Inc., 2004, poglavji 1 in 2.
- [5] Robert Schneider, *Microsoft SQL Server 2005 Express Edition For Dummies*, Indiana: Wiley Publishing, 2006, poglavja 3, 8 in 9.
- [6] Raghu Ramakrishnan, Johannes Gehrke, *Database Management Systems, Third Edition*, New York: McGraw-Hill, 2003, poglavje 5.
- [7] Rasmus Lerford, Kevin Tatroe, *Programing PHP*, Sebastpool: O'Reilly, 2002.
- [8] Giulio Toffoli, *The Definitive Guide to iReport*, Barkley: Apress, 2007.
- [9] (2012) Uradna spletna stran podjetja Salonit Anhovo, d.d.. Dostopno na: <http://www.salonit.si/>.
- [10] (2012) Peter King, "SCADA Systems – Looking Ahead". Dostopno na: http://www.controlmicrosystems.com/media/page-body-files/white-papers/WP_10.pdf.
- [11] (2012) Ge Intelligent Platforms, "HMI/SCADA - iFix". Dostopno na: http://www.ge-ip.com/products/family/proficy_hmiscada_ifix.
- [12] (2012) Wikipedia, "Microsoft Excel". Dostopno na: http://en.wikipedia.org/wiki/Microsoft_Excel.
- [13] (2012) Uradna spletna stran programa Notepad++. Dostopno na: <http://notepad-plus-plus.org>.
- [14] (2012) Adobe, "Dreamweaver". Dostopno na: <http://www.adobe.com/products/dreamweaver.html>.
- [15] (2012) Hkvstore, "PHP Report Maker". Dostopno na: <http://www.hkvstore.com/phpreportmaker/default.asp>.

- [16] (2012) Jaspersoft, "iReport". Dostopno na:
<http://jasperforge.org/projects/ireport>.
- [17] (2012) Sourceforge, "PHP/Java Bridge". Dostopno na:
<http://php-java-bridge.sourceforge.net/pjb/>.
- [18] (2012) Wampserver, "WAMP". Dostopno na:
<http://www.wampserver.com/en>.
- [19] (2012) Wikipedia, "ODBC". Dostopno na:
<http://en.wikipedia.org/wiki/ODBC>.
- [20] (2012) w3schools, "Css". Dostopno na:
<http://www.w3schools.com/css>.
- [21] (2012) w3schools, "JavaScript". Dostopno na:
<http://www.w3schools.com/js>.
- [22] (2012) Wikipedia, "OCX". Dostopno na:
http://en.wikipedia.org/wiki/Object_Linking_and_Embedding.
- [23] (2012) Wikipedia, "PDF". Dostopno na:
http://en.wikipedia.org/wiki/Portable_Document_Format.