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 zasnuje 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 podpisi 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 vođenje 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 proizvodnega procesa ................................................................. 7  
   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
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 predlog alarmov v orodju iReport Designer. ................................. 22
Slika 15: Prikaz delovanja povezave med programskim jezikoma Java in PHP. ............... 24
Slika 16: Rezultat programa zbirka.php ............................................................................. 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 delovanja aplikacije. ............................................................................. 43
Slika 30: Uvodna stran nadzornega sistema SCADA. ....................................................... 44
Slika 31: VBA urejevalnik ..................................................................................................... 45
Slika 32: Prikaz delovanja 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 programska 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.


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


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


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 karieran. Sonaravnni 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 Salonit 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 Salonit 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 Salonit 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 halji, 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 toplobe, kjer dimni plini material predhodno segrejejo. Material nato potuje v peč, kjer se specišče 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 kamionskim oziroma vagonskimi cisternami. Poleg kamionskega in železniškega prevoza (predvsem za namene izvoza na bližji vzhod) uporablja tudi ladijski transport.

Z realizacijo zadnjega investicijskega ciklusa je postala cementarna ena najmodernejših 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 proizvaja 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 energetsko 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 PLK nameščeni v proizvodnji.

![Diagram SCADA](image)

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 proizvajanje 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.


2.2.2 Procesna podatkovna baza


Slika 4: Slika področja Gorilec iz nadzornega sistema SCADA.
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.

Slika 5: Prikaz podatkovnih blokov iz procesne podatkovne baze.

2.3 Idejna rešitev problema


Poleg samih alarmov imamo v sistemu tudi dogodke, ki nosijo informacijo o akcijah, ki jo sproži operater na komandnem pultu. Taki dogodki so recimo prički 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 oblacijskih storitev, ki omogočajo dostopnost vsebin ne glede na uporabnikovo lokacijo. Tudi naša ideja je narediti aplikacijo v obliki spletna storitev. 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 ustreznati 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 vrednostmi,
- izbiro med pripadajočimi področji, v kolikor uporabnik izbere en sklop,
- upoštevanje vseh pripadajočih področij, v kolikor je izbran več kot en sklop,
- postavitev izbirnih območij naj bo v vrstnem redu kot ga prikazuje slika 6.
1. **Sklop**

<table>
<thead>
<tr>
<th>FILTER</th>
<th>MS</th>
<th>PEČ</th>
<th>MPK</th>
<th>MC</th>
<th>PAKIRKA</th>
<th>ENERGETIKA</th>
</tr>
</thead>
</table>

2. **Področje**

<table>
<thead>
<tr>
<th>FILTER1</th>
<th>MS1</th>
<th>PEČ1</th>
<th>MC1</th>
<th>ELEKTRIKA</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILTER2</td>
<td>MS2</td>
<td>PEČ2</td>
<td>MC2</td>
<td>KOMP. ZRAK</td>
</tr>
<tr>
<td>ODPRAŠKI</td>
<td>TR. MS</td>
<td></td>
<td>TR. MC.</td>
<td></td>
</tr>
</tbody>
</table>

3. **Izbera jezika**

<table>
<thead>
<tr>
<th>SLO</th>
<th>ANG</th>
</tr>
</thead>
</table>

4. **Izbera: alarmi, ponovitve alarmov ali dogodki**

<table>
<thead>
<tr>
<th>ALARMI</th>
<th>PONOVTVEVE</th>
<th>DOGODKI</th>
</tr>
</thead>
</table>

5. **Izbera prioritete alarmov**

<table>
<thead>
<tr>
<th>ALARMI (high)</th>
<th>OPOZORILA (medium in low)</th>
</tr>
</thead>
</table>

6. **Tip alarma**

<table>
<thead>
<tr>
<th>ODPRAVLJENI</th>
<th>NEODPRAVLJENI</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>Začetek</th>
<th>Konec</th>
<th>Alarm (tag)</th>
<th>Opis napake (opis 2)</th>
<th>Opis naprave (opis 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>29.3.2012 15:51</td>
<td>29.3.2012 15:53</td>
<td>HT15_L_A</td>
<td>L alarm</td>
<td>Ciklon 5 - vrh</td>
</tr>
<tr>
<td>29.3.2012 15:51</td>
<td></td>
<td>VF10081_ETOTAL</td>
<td>Skupna napaka</td>
<td>Merilec pretoka zraka - skupno</td>
</tr>
</tbody>
</table>

2. Ponovitve alarmov

<table>
<thead>
<tr>
<th>Ponovitev</th>
<th>Alarm</th>
<th>Opis</th>
</tr>
</thead>
<tbody>
<tr>
<td>164</td>
<td>HP58_GRTDRV2_OILPH_A</td>
<td>Rešetke hladilnika S OilPress H</td>
</tr>
<tr>
<td>70</td>
<td>HD_0_LACK_OF_TYRES_SEPARACION</td>
<td>LACK OF TYRES SEPARATION pomanjkanje gum za separacijo</td>
</tr>
</tbody>
</table>

3. Dogodki

<table>
<thead>
<tr>
<th>Pojavitev</th>
<th>Dogodek</th>
</tr>
</thead>
</table>

Slika 7: Smernice pri izpisu različnih tipov podatkov.

3.2 Orodja za interakcijo s podatkovno bazo


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


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

![Slika 10: Uporaba programa Notepad++](image)

**3.3.2 Adobe Dreamweaver**


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 uredja in omeje 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.
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 namestitev aplikacije in orodja za branje poročil na vsak računalnik.
Slika 12: Izpis alarmov s pomočjo obstoječe aplikacije.

3.4.2 Tiskanje podatkov iz html dokumenta


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 nog 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


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.


![Slika 13: Izdelovanje poročila s pomočjo programa PHP Report Maker.](image)

3.4.4 Izdelava poročil s pomočjo orodja iReport


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 tabelaričnega zapisa v html dokumentu. Ne bi potrebovali nastavljati 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 izbiral 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.


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 prepozna začetek in konec strani, ter omogočil izpis glave in noge. Pri tem poskusom 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.


Njegova največja prednost pred ostalimi orodji je omogočanje klicev funkcij za izgradnjo poročil 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
24


4.2 Vzpostavitev povezave PHP/Java Bridge


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


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

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

// razredi Java
```
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).
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.


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

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.

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

4.4.2 Programiranje logike aplikacije


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
```


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 MCI_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 smotreno, 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 najpogostejše ponovljeni alarmi na začetku. Število ponovljenih alarmov izračunamo s pomočjo stavka SQL.

```sql
SELECT COUNT(tag) as skupaj, tag, opis1, opis2
FROM alarni_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
```

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),
- `sestaviPodročje()` – stavku SQL doda pogoj za izbiro področja,
- `sestaviPrioriteta()` – omogoča sestavo pravilnega povpraševanja, glede na izbrano prioritetno.

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
<?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.


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.

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


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

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

    if(isset($_POST["področje"])){
        $področje = "" . $_POST["področje"]; }
$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='glavaD'>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.


Sintaksa jezika CSS je enostavna. Za vsak element navedemo oblikovanje v bloku, omejenem z zavitimi 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 unikaten element dokumenta. Id naslavljanje uporablja atribut id v html dokumentu in je določen s simbolom »#«.
Zgornja koda prikazuje oblikovanje glave izpisanih podatkov. Rezultat oblikovanja je viden na sliki 22.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PEC</td>
<td>Tip alarmov: vsi</td>
<td></td>
</tr>
<tr>
<td>Začetek</td>
<td>Konček</td>
<td>Alarm</td>
</tr>
<tr>
<td>Opis napake</td>
<td>Opis naprave</td>
<td></td>
</tr>
</tbody>
</table>

Slika 22: Rezultat zgornjega oblikovanja glave izpisa alarmov.

b) Izdelava predlog poročil z orodjem iReport Designer

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 paleto 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


```php
function izberiPredlogo($tip){
    $CompileManager = new JasperCompileManager();
    $predloga;
```
Pri izpisovanju poročila smo uporabljali naslednje razrede:

- JapserCompileManager() – 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 ponavljačih alarmov

V sistemu imamo odpravljene in nove ali neodpravljene alarme. Odpravljeni 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.

Slika 25: Podvojeni alarmi.


Problem smo začeli reševati na lažjem koncu in sicer pri odpravljenih alarmih. Težavo smo rešili dokaj enostavno, saj se ponovitve pojavljajo zgolj pri neodpravljenih alarmih. Nepotrgene alarme smo od drugih ločili z filtriranjem po atributu alm_status. Prikazali smo samo tiste, ki imajo vrednost atributa alm_status skoraj OK (slika 26).
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 pravilnih in lahko bi rekli »lažnivih« podatkov. V kolikor je v množici neodpravljenih alarmov število enakih imen (attribut 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 zavrzemo.

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 množice. Stavek je v končni fazi izgledal nekak

```
SELECT start_date, start_time, last_date, last_time, alm_status, tag, opis1, opis2
FROM alarmi_spk1v1
WHERE start in (SELECT start
FROM alarmi_spk1v1
WHERE + pogoj + AND
msg_type = 'ALARM'
GROUP BY start, tag
HAVING count(tag) % 2 = 1) AND
alm_status = 'CFN'
ORDER BY start
```

V zgornji kodni imamo ugnezdena select stavka. Notranji stavek poskrbi, da se upoštevajo pogojii 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.

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.


```sql
SELECT start_date, start_time, last_date, last_time, alm_status, tag, opis1, opis2
FROM alarmi_spk1v1
WHERE (start in (SELECT start
FROM alarmi_spk1v1
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
```
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).
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 sistem in v njem izvrlšuje ukaze, nima dostopa do drugih programov operacijskega sistema. Zato je bilo omogočanje uporabe aplikacije v nadzornem sistemu visoka prioriteva pri izdelavi aplikacije.


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).

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


![Slika 33: Končna verzija uporabniškega vmesnika.](image)

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

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.
Slika 35: Izpis alarmov.


Slika 36: Prikaz poročila alarmov v pdf dokumentu.

### Slika 37: Prikaz poročila alarmov v html dokumentu.
6 Sklepne ugotovitve


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.


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. Četudi se nekateri problemi na začetku zdijo nerešljivi, jih lahko z voljo in trudom uspešno rešimo.
Literatura in viri


