

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**PUBLISKO DATU SISTĒMAS ADMINISTRĒŠANAS MODUĻA
UZLABOŠANA**

KVALIFIKĀCIJAS DARBS

Autors: **Andris Jakubovskis**

Stud. apl. nr. aj14031

Darba vadītājs: B.dat. Jevgēnijs Trofimovs

RĪGA 2016

ANOTĀCIJA

Darbā ir aprakstīts Valsts ieņēmumu dienesta publiskās datu bāzes administrēšanas moduļa uzlabošana, izmantojot Modelis-Skats-Kontrolleris (MVC) šablonu. Programmējot izmantota objektorientētā pieeja. Sistēmas pamatbūtība ir ļaut administratoram veikt pārskatus par ierakstu skaitiem gan oriģinālajā datu bāzē, gan publiski pieejamajā datu bāzē.

Atslēgvārdi: publiskā datu bāze, administrēšanas modulis, pārskats, MVC.

ABSTRACT

The paper described the State Revenue Service public database administration module enhancement through a Model-View-Controller (MVC) pattern. Object-oriented approach is used for programming. The core of the system is to allow the administrator to carry out the number of entries in both the original database and publicly available database.

Keywords: public database, administration module, overview, MVC.

SATURA RĀDĪTĀJS

Anotācija	2
Abstract	3
Satura rādītājs	4
Vārdnīca	7
Ievads	8
1. Programmatūras prasību specifikācija.....	9
1.1. Ievads	9
1.1.1. Nolūks	9
1.1.2. Darbības sfēra.....	9
1.1.3. Defīnīcijas	9
1.1.4. Saistība ar citiem dokumentiem	9
1.1.5. Dokumenta pārskats	9
1.2. Vispārējs apraksts	10
1.2.1. Sistēmas lietotāji	10
1.2.2. Lietotāja raksturzīmes	10
1.2.3. Produkta funkcijas.....	10
1.2.4. Vispārējie ierobežojumi	10
1.2.4.1. Atbilstība standartiem	10
1.2.4.2. Aparatūras ierobežojumi	10
1.2.5. Pieņēmumi un atkarības	11
1.3. Funkcionālās prasības	11
1.3.1. Pārskata izveidošana	11
1.3.2 Informācijas par ierakstiem abās datubāzēs atgriešana JSON formā.....	14
1.4. Nefunkcionālās prasības	15
2. Programmatūras projektējuma apraksts	16
2.1. Ievads	16
2.1.1. Nolūks	16
2.1.2. Darbības sfēra.....	16
2.1.3. Defīnīcijas	16
2.1.4. Saistība ar citiem dokumentiem	16

2.1.5. Dokumenta pārskats	16
2.2. Dekompozīcijas apraksts	17
2.2.1. “Home” kontrolieris	17
2.2.2. “Select” klase	17
2.2.3. “Index” skats	17
2.4. Detalizēts projektējuma apraksts	17
2.4.1. Lietotāja saskarnes apraksts	17
2.4.2. Datu iegūšana JSON formātā	18
2.4.3. Pārskata aizpildīšana ar datiem pēc datu izsaukšanas	19
2.4.4. Datu transformāciju informācijas vienumi	19
2.5. Nefunkcionālo prasību realizācija	32
2.6. Prasību trasējamības matrica	33
3. Testēšanas dokumentācija.....	34
3.1. Ievads	34
3.1.1. Nolūks	34
3.1.2. Testēšanas metodika.....	34
3.1.3. Defīnīcijas	34
3.1.4. Saistība ar citiem dokumentiem	34
3.2. Testēšanas žurnāls.....	35
3.2.1. Vienībtesti	35
3.2.2. Akcepttesti.....	36
3.2.3. Nefunkcionālo prasību testēšana.....	39
4. Projekta organizācija.....	40
5. Kvalitātes nodrošināšana	41
6. Konfigurāciju pārvaldība un versijkontrolē	42
7. Darbietilpības novērtējums	43
8. Secinājumi	44
9. Izmantoto avotu un literatūras saraksts.....	45
10. Pielikumi.....	46
10.1. Koda atbilstība standartiem	46
10.1.1. HTML koda atbilstība HTML5 standartam	46
10.1.2. CSS koda atbilstība CSS3 standartam	46
10.2. Ekrānformas.....	47
10.2.1. Tikko atvērts pārskats bez veikta pieprasījuma pēc datiem	47

10.2.2. Pārskats pēc datu pieprasīšanas.....	48
10.3. Programmatūras pirmkoda piemēri	48
10.3.1. “Home” kontrolieris	48
10.3.2. “aUnits” klase.....	49
10.3.2.1. “aUnits” objekta piemērs	50
10.3.3. “Index” skats	50
10.3.4. “fillData” funkcija.....	51
10.3.5. “db_exec” funkcija.....	52
10.3.6. “getCount” funkcija	53

VĀRDNĪCA

Saīsinājums	Skaidrojums
CDB	NIS un VADIS centrālās datu bāzes, kuras tiek izmantotas informācijas atjaunošanai PDB
CMIS	Valsts ieņēmumu dienesta Centrālā muitas informācijas sistēma
CSS	<i>Cascading Style Sheets</i> - kaskadētas stila lapas
DAK	Darījumu apliecinošās kvītis
DB	Datu bāze
HTML	<i>HyperText Markup Language</i> – hiperteksta iezīmēšanas valoda
JSON	<i>JavaScript Object Notation</i> – JavaScript Objektu Notācija
KA	Kases aparāts vai elektroniskā kases sistēma
MUN	Mikrouzņēmumu nodokļa maksātāji
NIS	Nodokļu informācijas sistēma
NM	Nodokļu maksātājs
PDB	VID Publiskojamo datu bāze
PVN	Pievienotās vērtības nodoklis
SDV	Saimnieciskās darbības veicējs
SLO	Sabiedriskā labuma organizācija
SNK	Nodokļu maksātāju (komersantu) taksācijas gadā samaksāto VID administrēto nodokļu kopsummas
URL	<i>Uniform Resource Locator</i> – Vienotais resursu vietrādis
VADIS	Valsts amatpersonu deklarāciju informācijas sistēma
VAP	Valsts amatpersonu pārkāpumi
VID	Valsts ieņēmumu dienests
WWW	<i>World Wide Web</i> - globālais tīmeklis

IEVADS

Kvalifikācijas darba tēma ir “Publisko datu sistēmas administrēšanas moduļa uzlabošana”. Tēma ir aktuāla, jo VID sistēmas administratoram ir nepieciešams salīdzināt ierakstu skaitu starp VID CDB un VID PDB, lai būtu iespējams sagatavot salīdzinošos pārskatus par abām datu bāzēm un veiktajām datu transformācijām. Tēma tika izvēlēta pēc darba vadītāja ieteikuma.

Darba mērķis ir uzlabot VID administrēšanas moduļa daļu, kurā iespējams veikt pārskatus par veiktajām datu transformācijām, padarot kodu atbilstošu mūsdienu standartiem un izmantojot objektorietētu pieeju. Piekļuve šiem pārskatiem ir dota tikai sistēmas administratoriem.

Darbs sastāv no programmatūras prasību specifikācijas, programmatūras projektējuma apraksta un testēšanas dokumentācijas.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Programmatūras prasību specifikācijas nolūks ir aprakstīt VID Publiskojamo datu bāzes administrēšanas moduļa lietotāju prasības. Dokumentā ir aprakstītas sistēmas nefunkcionālas, funkcionālas. Dokumenta mērķis ir precīza lietotāju prasību formulēšana, lai izstrādātājs varētu viennozīmīgi realizēt pasūtītāja prasības.

1.1.2. Darbības sfēra

Sistēma VID PDB administrēšanas modulis ir paredzēts darbiniekiem, kas ir šīs sistēmas administratori un kuriem nepieciešams veikt sistēmas pārskatus, lai varētu iegūt informāciju VID CDB un VID PDB darbību.

1.1.3. Definīcijas

Definīciju un apzīmējumu sarakstu iespējams nodaļā “Vārdnīca”.

1.1.4. Saistība ar citiem dokumentiem

Dokuments saistīts ar programmatūras projektējuma aprakstu un testēšanas dokumentāciju.

Dokuments noformēts pēc LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis” standarta prasībām.

1.1.5. Dokumenta pārskats

Dokuments sastāv no sekojošām nodaļām:

- Ievads
- Vispārējs apraksts
- Funkcionālas prasības
- Nefunkcionālas prasības

1.2. Vispārējs apraksts

1.2.1. Sistēmas lietotāji

Sistēmas administratīvajam daļai var piekļūt tikai viena lietotāju grupa - sistēmas administrators.

1.2.2. Lietotāja raksturzīmes

Veiksmīgai strādāšanai ar sistēmu nepieciešama ierīce ar interneta pieslēgumu un interneta pārlūkprogrammu. Lietotājs var strādāt ar sistēmas administratīvo moduli, tad, ja tas ir sistēmas administrators.

1.2.3. Produkta funkcijas

Administratīvajā modulī sistēmas administrators var veikt pārskatus par veiktajām datu transformācijām no VID oriģinālajām datu bāzēm uz PDB. Administratora darba vieta ir noformēta kā WWW lapa uz VID iekšējā lietojuma servera. Pārskati administratoram pieejami kā HTML lapas. VID CDB serveris atrodas VID iekšējā uguns mūra aizsardzībā (atšķirībā no PDB un tās WWW servera, kas atrodas tikai ārējā uguns mūra aizsardzībā), un tam piekļuve gan VID oriģinālajai datu bāzei, gan PDB. Piekļuve abām datu bāzēm vienlaicīgi ir nepieciešama, lai būtu iespējams sagatavot salīdzinošos pārskatus par abām datu bāzēm un veiktajām datu transformācijām.

1.2.4. Vispārējie ierobežojumi

1.2.4.1. Atbilstība standartiem

- HTML kodam jāatbilst W3C HTML5 standartam.
- CSS kodam jāatbilst W3C CSS3 standartam.

1.2.4.2. Aparatūras ierobežojumi

Sistēmas atdarbība ir atkarīga no servera, uz kā tā darbojas, veiktspējas, no tā cik sistēma dotajā brīdī ir noslogota, no sistēmas lietotāja datora veiktspējas un interneta pieslēguma ātruma.

Lai veiksmīgi strādātu ar sistēmu nepieciešama ierīce ar interneta pieslēgumu un pārlūkprogrammu Google Chrome ar versiju, kas nav vecāka par 35. versiju, jo vecākas neatbalsta dinamisko skriptu (JavaScript) izpildi.

1.2.5. Pieņēmumi un atkarības

Serveris pieejams tikai pašam sistēmas administratoram un tīmekļa lapa pieejama tikai lokāli.

Prasību apkopošanas laikā netika identificēti faktori, kuru izmaiņas var atstāt iespaidu uz prasību realizāciju.

1.3. Funkcionālās prasības

1.1. tabula

1.3.1. Pārskata izveidošana
Identifikators: create_database_data_comparison
Ievads
Sistēmas administrators var veikt pārskatu par ierakstu skaitu VID CDB un VID PDB. Ja abi skaitļi sakrīt, tas nozīmē, ka VID oriģinālajā datu bāzē nav nākuši klāt jauni ieraksti. Raksturojošie skaitļi ir svarīgi datu transformāciju kontrolei – vai transformāciju laikā visi nepieciešamie ieraksti no VID CDB ir transformēti PDB.
Ievade
Datu transformāciju informācijas vienums: <ul style="list-style-type: none">• NM dati• PVN dati• KA dati• Licenču dati (LIC)• VADIS dati• SLO dati• SDV dati• SNK dati

- DAK dati
- MUN dati
- VAP dati

Apstrāde

Atkarībā no informācijas vienuma iegūst datus no VID CDB un VID PDB.

1.2. tabula

Vienums	Iegūstamā informācija
NM dati	Ierakstu skaits tabulā "Nodokļu maksātājs" (<i>nodokl_maks</i>)
PVN dati	Ierakstu skaits tabulā "Pievienotās vērtības nodokļa maksātāji" (<i>pvn</i>)
KA dati	Ierakstu skaits tabulā "KA markas, modeļi un modifikācijas" (<i>ka_modelis</i>)
	Ierakstu skaits tabulā "Kases aparātu tirgotāji un servisa dienesti" (<i>ka_tirgserv</i>)
	Ierakstu skaits tabulā "Kases aparāti" (<i>ka_reg</i>)
	Ierakstu skaits tabulā "Atļautās KA markas, modeļi un modifikācijas" (<i>ka_atlmodeli</i>)
Licenču dati (LIC)	Ierakstu skaits tabulā "Licences" (<i>apmd_lic</i>)
	Ierakstu skaits tabulā "Darbības vietas tips" (<i>apmd_licviettips</i>)
	Ierakstu skaits tabulā "Darbības vieta" (<i>apmd_licvieta</i>)
	Ierakstu skaits tabulā "Akcīzes preču veids" (<i>ap_prece</i>)
	Ierakstu skaits tabulā "Darbības veids ar akcīzes precēm" (<i>ap_darbveids</i>)
	Ierakstu skaits tabulā "Speciālajā atļaujā norādītās preces" (<i>ap_nolprece</i>)
	Ierakstu skaits tabulā "Speciālajā atļaujā norādītie darbības veidi" (<i>ap_noldveidi</i>)
	Ierakstu skaits tabulā "Licenču īpašās atzīmes par darbībām" (<i>ap_ipatzpardarb</i>)
VADIS dati	Ierakstu skaits tabulā "Personas" (<i>personas</i>)

	Ierakstu skaits tabulā "Valsts amatpersonu deklarācijas" (<i>deklaracija</i>)
	Ierakstu skaits tabulā "Nekustamie īpašumi" (<i>nekust_ip</i>)
	Ierakstu skaits tabulā "Transporta līdzekļi" (<i>transp_lidz</i>)
	Ierakstu skaits tabulā "Iznomātie/nomātie īpašumi" (<i>iznom_ip</i>)
	Ierakstu skaits tabulā "Ieņemamie amati" (<i>amati</i>)
	Ierakstu skaits tabulā "Ienākumi" (<i>ienakumi</i>)
	Ierakstu skaits tabulā "Kapitāla daļas un vērtspapīri" (<i>kap_dalas</i>)
	Ierakstu skaits tabulā
	"Skaidrās naudas uzkrājumi" (<i>nauda</i>)
	Ierakstu skaits tabulā "Radinieki" (<i>radinieki</i>)
	Ierakstu skaits tabulā "Darījumi" (<i>darijumi</i>)
	Ierakstu skaits tabulā "Nekustamo īpašumu līdzīpašnieki" (<i>nekustip_sat</i>)
	Ierakstu skaits tabulā "Informācija" (<i>informacija</i>)
	Ierakstu skaits tabulā "Parādsaistības" (<i>publ_paradi</i>)
	Ierakstu skaits tabulā "Aizdevumi" (<i>publ_aizd</i>)
	Ierakstu skaits tabulā "Patiesā labuma guvēji" (<i>pat_labguv</i>)
	Ierakstu skaits tabulā "Deklarāciju veidi" (<i>dekl_veids</i>)
SLO dati	Ierakstu skaits tabulā „Sabiedriskā labuma organizācijas” (<i>slo_registr</i>)
	Ierakstu skaits tabulā „Lēmumi par sabiedriskā labuma organizācijas statusa piešķiršanu/atņemšanu” (<i>slo_iesnlem</i>)
	Ierakstu skaits tabulā „Sabiedriskā labuma organizācijas darbības jomas” (<i>slo_darbjoma</i>)
	Ierakstu skaits tabulā „SLO darbības pārskati un turpmākās darbības plāni” (<i>slo_darbparsk</i>)
	Ierakstu skaits tabulā „Sabiedriskā labuma darbības jomas” (<i>sablalab_darbjoma</i>)
SDV dati	Ierakstu skaits tabulā „Saimnieciskās darbības veicēji” (<i>sdv_fp</i>)

	Ierakstu skaits tabulā „Saimnieciskās darbības apturēšana un atjaunošana” (<i>sd_aptatj</i>)
SNK dati	Ierakstu skaits tabulā "Nodokļu maksātāju (komersantu) taksācijas gadā samaksāto VID administrēto nodokļu kopsummas" (<i>nmk_snk</i> (NIS))
	Ierakstu skaits tabulā "Nodokļu maksātāju (komersantu) taksācijas gadā samaksāto VID administrēto nodokļu kopsummas" (<i>nmk_snk</i> (CMIS))
DAK dati	Ierakstu skaits tabulā "Darījumu apliecināšanās kvītis" (<i>dak_reg</i>)
	Ierakstu skaits tabulā "Darījumu reģistrētās kvītis" (<i>dak_regr</i>)
MUN dati	Ierakstu skaits tabulā "Mikrouzņēmumu nodokļa maksātāji" (<i>mun_reg</i>)
VAP dati	Ierakstu skaits tabulā "Valsts amatpersonu pārkāpumi" (<i>vap_reg</i>)
Izvade	
Parāda ierakstu skaistu katrā no datubāzēm.	

1.3. tabula

1.3.2 Informācijas par ierakstiem abās datubāzēs atgriešana JSON formā
Identifikators: return_JSON_report
Ievads
Pēc konkrēta URL ar padotu konkrētu parametru var iegūt JSON objektu ar informāciju par datiem VID PDB un VID CDB.
Ievade
URL vietne uz lapu ar padotu parametru par pieprasāmajiem datiem. Padodamie parametri: Skatīt 1.3.1. “Pārskata izveidošana” sadaļu ievade
Apstrāde
Atkarībā no informācijas vienuma iegūst datus no VID CDB un VID PDB.

Skatīt 1.2. tabulu, lai redzētu iegūstamos datus.
Izvade
JSON objekts ar iegūstamo informāciju par ierakstu skaitiem abās datubāzēs

1.4. Nefunkcionālās prasības

1.3. tabula

Prasības numurs	Prasības apraksts
NF.1.	Lietotāja saskarne – tiek īstenota caur tīmekļa pārlūkprogrammu. Lietotāja saskarnē tiek izmantota grafiskā saskarne.
NF.2.	Valoda – Sistēmas saturam jābūt latviešu valodā.
NF.3.	Pārskatāmība – Sistēmas saturam jābūt intuitīvam, labi pārskatāmam, ērtam un nevajadzētu raksties grūtībām to lietot.
NF.4.	Sistēmas vizuālās pamatiezīmes, kas attiecas uz visiem skatiem – Lapas augšā atrodas galvene. Galvenes kreisajā malā (lapas kreisajā augšējā stūrī) ir VID logo. Galvenes apakšējā daļā atrodas izvēlne, kur iespējams pārslēgties starp vietnēm uz citiem skatiem. Lapas apakšējā daļā ir autortiesību paziņojums un informācija par sistēmas versiju.
NF.5.	HTML koda atbilstība HTML5 standartam
NF.6.	CSS koda atbilstība CSS3 standartam

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Nolūks

Programmatūras projektējuma apraksta nolūks ir aprakstīt VID Publiskojamo datu bāzes administrēšanas moduļa programmatūras prasību specifikācijā minēto funkcionālo un nefunkcionālo prasību realizācijas projektējumu.

2.1.2. Darbības sfēra

Sistēma VID PDB administrēšanas modulis ir paredzēts darbiniekiem, kas ir šīs sistēmas administratori un kuriem nepieciešams veikt sistēmas pārskatus, lai varētu iegūt informāciju VID CDB un VID PDB darbību.

2.1.3. Definīcijas

Definīciju un apzīmējumu sarakstu iespējams nodaļā “Vārdnīca”.

2.1.4. Saistība ar citiem dokumentiem

Dokuments veidots pēc un lietojams kopā ar dokumentu “Programmatūras prasību specifikācija” un pēc šī dokumenta ir veidots dokuments “Testēšanas dokumentācija”.

Dokuments noformēts pēc „LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai” standarta prasībām.

2.1.5. Dokumenta pārskats

Dokuments sastāv no sekojošām nodaļām:

- Dekompozīcijas apraksts
- Atkarību apraksts
- Detalizēts projektējuma apraksts
- Nefunkcionālo prasību realizācija

- Prasību trasējamības matrica

2.2. Dekompozīcijas apraksts

Sistēma tiek izstrādāta pēc MVC principa.

Sistēma satur sekojošas daļas:

2.2.1. “Home” kontrolieris

Kontrolieris, kas caur kuru var piekļūt gan “Index” skatam, gan iegūt JSON rezultātu ar datu atlasi no CDB un PDB.

2.2.2. “Select” klase

Klase, kur tiek realizēta programmas darījumu loģika.

2.2.3. “Index” skats

Skats, kurā redzams pārskats par datubāžu ierakstiem.

2.4. Detalizēts projektējuma apraksts

2.4.1 Lietotāja saskarnes apraksts

Sistēmas administratoram tiek nodrošināti pārskati par veiktajām datu transformācijām no VID oriģinālajām datu bāzēm uz PDB. Pārskats sastāv no blokiem, kas dalās pēc informācijas vienumiem. Bloka virsraksts satur informācijas vienuma nosaukumu.

Pēc bloka virsraksta seko informācijas vienumu raksturojošas informācijas (parametru) detalizācijas bloks, kas ietver datus no VID oriģinālajām DB un PDB. Datu sagatavošana aprakstīta nodaļā 2.4.3. “Datu transformāciju informācijas vienumi”.

Tabulas kreisajā daļā izvietota spiedpoga “Atjaunot”, kuru nospiežot, administrators var atjaunot informāciju par atbilstošo detalizācijas bloku. Spiedpoga ieviesta tādēļ, lai pārskata atvēršanās brīdī nenotiktu automātiska visu detalizācijas bloku informācijas atjaunošana – tas strādā lēni lielā informācijas apjoma dēļ un tādejādi palielina pārskata sagatavošanas laiku.

Tiem informācijas vienumiem, par kuriem detalizācijas bloka dati nav atjaunoti, informācijas ailēs ievietots teksts “Nav datu”. Lai atjaunotu detalizācijas datus, jānospiež detalizācijas blokam

atbilstošā spiedpoga “Atjaunot”. Skatīt nodaļu 10.2.1. “Tikko atvērts pārskats bez veikta pieprasījuma pēc datiem”, lai redzētu ekrānformu.

Katras norādītās datu atlasēs rezultātā tiek sagatavots atbilstošā informācijas vienuma ierakstu skaits VID oriģinālajā datu bāzē un ierakstu skaits PDB. Ja abi skaitļi sakrīt, tas nozīmē, ka VID oriģinālajā datu bāzē nav nākuši klāt jauni ieraksti. Skatīt nodaļu 10.2.2. “Pārskats pēc datu pieprasīšanas”, lai redzētu ekrānformu.

Raksturojošie skaitļi ir svarīgi datu transformācijas kontrolei – vai transformāciju laikā visi nepieciešamie ieraksti no VID CDB ir transformēti PDB.

Ja pēc detalizācijas datu atjaunošanas informācijas ailēs parādās teksts “Kļūda”, tas nozīmē, ka datu atjaunošanas laikā datu bāzē notikusi kļūda un detalizācijas dati nav atjaunoti.

2.4.2. Datu iegūšana JSON formātā

JSON rezultātu iespējams iegūt apmeklējot konkrētu URL *Home/getInfo*, kam tiek padots parametrs par pieprasāmo informācijas vienumu. Parametrs tiek padots konkr URL beigās rakstot “?unit=*parametrs*”, kur *parametrs* ir kāds no PPS minētajiem informācijas vienumiem. Tātad lai iegūtu JSON no vietnes sākumlapas būtu jānodod uz “/Home/getInfo“?unit=*parametrs*”, kur parametra vērtība var būt kāds no nodaļā 1.3.1. pie ievades minētajiem apzīmējumiem (kas tur rakstīti ar lielajiem burtiem).

Atkarībā no padotā parametra tiek veikts konkrēts skaits ar noteiktām datu atlasēm. Lai redzētu datu atlasēs pēc padotā informācijas vienuma, skatīt nodaļu 2.4.3. “Datu transformāciju informācijas vienumi”. Datu atlasēs rezultāti tiek saglabāti kā viens klases objekts, jeb viena atgriezmā rezultātu rinda simbolizē vienu klases objektu. Tiek izveidots saraksts ar sādien objektiem un tālāk padots kontrolierim, kas tos saņem JSON formā.

Rezultāts tiek atgriezts kā JSON masīvs. Atgrieztais JSON masīvs ir trīsdimensionāls. Pirmajā dimensijā tas sastāv no 3 vērtībām. Pirmajā vērtībā glabājas informācija par to kādi dati ir atlasīti, savukārt otrā un trešā vērtība attiecīgi glabā datu atlasēs rezultātus no VID CDB un VID PDB. Otrajā dimensijā masīva vērtību skaits ir atkarīgs no pieprasītā vienuma, jo datu atlasu skaits dažādiem vienumiem var būt dažāds. Trešajā dimensijā tam ir viena vērtība “result”, kas glabā sevī atgrieztos rezultātus.

2.4.3. Pārskata aizpildīšana ar datiem pēc datu izsaukšanas

Katra poga “Atjaunot” darbojas uz savu tabulu, kurā iespējams izveidot pārskatu un katrai pogai ir jau klāt parametrs, ko tā pados JavaScript funkcijai. Izsauktā JavaScript funkcija veic AJAX pierasījumu pēc JSON datiem padodot saņemto parametru no nospiešanās pogas un padodot to tālāk vietnei, kas aprakstīta nodaļā 2.4.2. “Datu iegūšana JSON formātā”. Veiksmīgas datu saņemšanas gadījumā funkcija nomaina atskaites tabulas lauku saturu ar datiem no pieprasījuma, savukārt neveiksmīgas datu atslases gadījumā, tā lauku saturu aizpilda ar paziņojumu “Kļūda”.

2.4.4. Datu transformāciju informācijas vienumi

Nākošajos punktos aprakstīti vaicājumi VID oriģinālajām datu bāzēm un PDB, kuru rezultāti tiek izmantoti pārskata par veiktajām datu transformācijām sagatavošanā.

2.4.4.1. NM dati

2.1. tabula

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā “Nodokļu maksātājs” (<i>nodokl_maks</i>)	<i>SELECT count(*) FROM nodokl_maks</i>	<i>SELECT count(*) FROM nodokl_maks</i>

2.4.4.2. PVN dati

2.2. tabula

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā “Pievienotās vērtības nodokļa maksātāji” (<i>pvn</i>)	<i>EXECUTE PROCEDURE pdb_get_count (‘pvn’);</i>	<i>SELECT count(*) FROM pvn</i>

2.4.4.3. KA dati

2.3. tabula

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā “KA markas, modeļi un modifikācijas” (<i>ka_modelis</i>)	<i>SELECT count(*) FROM ka_modelis mo, ka_marka ma</i>	<i>SELECT count(*) FROM ka_modelis</i>

Parametrs	Datu avots	PDB
	<i>WHERE mo.id_kamarka=ma.id_kamarka</i>	
Ierakstu skaits tabulā “Kases aparātu tirgotāji un servisa dienesti” (<i>ka_tirgserv</i>)	<i>SELECT FIRST 1 (</i> <i>(SELECT count(*) FROM ka_tirgserv</i> <i>WHERE ier_stat='A' AND tirgotajs='0') +</i> <i>(SELECT count(p.id_katirgservp) FROM</i> <i>ka_tirgservp p, ka_tirgserv t WHERE</i> <i>t.ier_stat='A' AND p.ier_stat='A' AND</i> <i>p.id_katirgservp<>-1 AND</i> <i>p.id_katirgserv=t.id_katirgserv AND</i> <i>p.strv_veids='I' AND t.tirgotajs='0')</i> <i>FROM ka_tirgserv</i>	<i>SELECT count(*)</i> <i>FROM ka_tirgserv</i>
Ierakstu skaits tabulā “Kases aparāti” (<i>ka_reg</i>)	<i>SELECT count(*)</i> <i>FROM ka_lietot l, ka_uzstvietas u,</i> <i>ka_reg r, ka_veids v, ka_statizm s</i> <i>WHERE r.id_kareg=u.id_kareg</i> <i>AND r.id_kareg=l.id_kareg</i> <i>AND r.id_kareg=s.id_kareg</i> <i>AND r.id_kaveids=v.id_kaveids</i> <i>AND r.ier_stat='A'</i> <i>AND l.ier_stat='A'</i> <i>AND u.ier_stat='A'</i> <i>AND s.ier_stat='A'</i> <i>AND r.id_kareg<>-1</i> <i>AND s.id_kastat IN (990000001,</i> <i>990000004)</i>	<i>SELECT count(*)</i> <i>FROM ka_reg</i>
Ierakstu skaits tabulā “Atļautās KA markas, modeļi un modifikācijas” (<i>ka_atlmodeli</i>)	<i>SELECT FIRST 1 (</i> <i>(SELECT count(*) FROM ka_atlmodeli</i> <i>WHERE id_kaatlmodeli<>-1 AND</i> <i>ier_stat='A'</i> <i>AND id_katirgserv IN (SELECT</i> <i>id_katirgserv FROM ka_tirgserv WHERE</i> <i>ier_stat='A' AND id_katirgserv<>-1 AND</i> <i>id_katirgservp=-1 AND tirgotajs='0'))+</i> <i>(SELECT count(*) FROM ka_atlmodeli</i> <i>WHERE id_katirgservp<>-1 AND</i> <i>ier_stat='A'</i> <i>AND id_katirgservp IN (SELECT</i> <i>id_katirgservp FROM ka_tirgservp WHERE</i> <i>ier_stat='A' AND id_katirgservp<>-1 AND</i> <i>strv_veids='I'))</i> <i>) FROM ka_atlmodeli</i>	<i>SELECT count(*)</i> <i>FROM ka_atlmodeli</i>

2.4.4.4. Licenču dati

2.4. tabula

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Licences" (apmd_lic)	<pre>SELECT FIRST 1 (SELECT count(*) FROM apmd_lic l, precu_grupa p, apmd_nmpaz a WHERE l.id_apmdnmpaz=a.id_apmdnmpaz AND p.id_precugrupa=l.id_precugrupa AND l.statuss<>'P' AND l.id_apmdlic<>-1 AND serija in ('AM', 'AV', 'BJ', 'BM', 'BV', 'MT', 'TM', 'TV', 'VT') AND id_dokanuliem<>990000022)+ (SELECT count(*) FROM an_nolik WHERE statuss<>'X' AND statuss<>'P' AND id_doktips in(990000241, 990000246, 990000247, 990000242, 990000243)) FROM apmd_lic;</pre>	<pre>SELECT count(*) FROM apmd_lic</pre>
Ierakstu skaits tabulā "Darbības vietas tips" (apmd_licviettips)	<pre>SELECT count(*) FROM apmd_licviettips;</pre>	<pre>SELECT count(*) FROM apmd_licviettips;</pre>
Ierakstu skaits tabulā "Darbības vieta" (apmd_licvieta)	<pre>SELECT FIRST 1 (SELECT count(*) FROM apmd_licvieta WHERE id_adrese<>-1 AND id_apmdlic in (select id_apmdlic from apmd_lic where statuss<>'P' AND id_apmdlic<>-1 AND serija in('AM', 'BM', 'BV', 'MT', 'TM', 'TV', 'VT')) AND id_apmdlic IN (SELECT id_apmdlic FROM apmd_lic l, precu_grupa p, apmd_nmpaz a WHERE l.id_apmdnmpaz=a.id_apmdnmpaz AND p.id_precugrupa=l.id_precugrupa</pre>	<pre>SELECT count(*) FROM apmd_licvieta</pre>

Parametrs	Datu avots	PDB
	<pre> AND l.statuss<>'P' AND l.id_apmdlic<>-1 AND serija in('AM','BJ', 'BM', 'BV','MT', 'TM', 'TV', 'VT') AND id_dokanuliem<>990000022)))+(SELECT count(*) FROM an_nolik WHERE statuss<>'X' AND statuss<>'P' AND id_doktips in(990000241, 990000246, 990000247, 990000242, 990000243) AND id_annolik IN (SELECT id_annolik FROM an_nolik WHERE statuss<>'X' AND statuss<>'P' AND id_doktips in(990000241, 990000246, 990000247, 990000242, 990000243))) FROM apmd_licvieta ; </pre>	
Ierakstu skaits tabulā "Akcīzes preču veids" (ap_prece)	<pre> SELECT count(*) FROM ap_prece; </pre>	<pre> SELECT count(*) FROM ap_prece; </pre>
Ierakstu skaits tabulā "Darbības veids ar akcīzes precēm" (ap_darbveids)	<pre> SELECT count(*) FROM ap_darbveids; </pre>	<pre> SELECT count(*) FROM ap_darbveids; </pre>
Ierakstu skaits tabulā "Speciālajā atļaujā norādītās preces" (ap_nolprece)	<pre> execute procedure dn_apnol('ap_nolprece'); SELECT count(*) FROM tt_apnolprece WHERE id_apmdlic in (SELECT id_annolik+1000000000 FROM an_nolik WHERE statuss<>'X' AND statuss<>'P' AND id_doktips in (990000241, 990000246, 990000247, 990000242, 990000243)); </pre>	<pre> SELECT count(*) FROM ap_nolprece </pre>
Ierakstu skaits tabulā "Speciālajā atļaujā norādītie darbības veidi" (ap_noldveidi)	<pre> execute procedure dn_apnol('ap_noldveidi'); SELECT count(*) FROM tt_apnoldveidi WHERE id_apmdlic in (SELECT id_annolik+1000000000 FROM an_nolik WHERE statuss<>'X' AND statuss<>'P' AND id_doktips in (990000241, 990000246, 990000247, 990000242, 990000243)); </pre>	<pre> SELECT count(*) FROM ap_noldveidi </pre>

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Licenču īpašās atzīmes par darbībām" (ap_ipatzpardarb)	<i>execute procedure dn_apnol('ap_ipatzpardarb'); SELECT count(*) FROM tt_apipatzpardarb WHERE id_apmdlic in (SELECT id_annolik+1000000000 FROM an_nolik WHERE statuss<>'X' AND statuss<>'P' AND id_doktips in (990000241, 990000246, 990000247, 990000242, 990000243));</i>	<i>SELECT count(*) FROM ap_noldveidi</i>
Ierakstu skaits tabulā "Speciālo atļauju apturēšana" (apmd_licapt)	<i>SELECT FIRST 1 ((SELECT count(*) FROM apmd_licapt WHERE id_apmdlic IN (SELECT id_apmdlic FROM apmd_lic l, precu_grupa p, apmd_nmpaz a WHERE l.id_apmdnmpaz=a.id_apmdnmpaz AND p.id_precugrupa=l.id_precugrupa AND l.statuss<>'P' AND l.id_apmdlic<>-1 AND serija in('AM','BJ', 'BM', 'BV','MT', 'TM', 'TV', 'VT') AND id_dokanuliem<>990000022)))+(SELECT count(*) FROM an_nolikapt WHERE id_annolik IN (SELECT id_annolik FROM an_nolik WHERE statuss<>'X' AND statuss<>'P' AND id_doktips in(990000241, 990000246, 990000247, 990000242, 990000243)))) FROM an_nolikapt;</i>	<i>SELECT count(*) FROM apmd_licapt</i>

2.4.4.5. VADIS dati

2.5. tabula

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Personas" (personas)	<i>SELECT count(*) FROM personas</i>	<i>SELECT count(*) FROM personas</i>
Ierakstu skaits tabulā "Valsts amatpersonu	<i>SELECT count(*)</i>	<i>SELECT count(*)</i>

Parametrs	Datu avots	PDB
deklarācijas” (deklaracija)	<pre>FROM deklaracija d, dekl_veidi v, organiza o, amati a WHERE d.id_deklveids=v.id_deklveids AND o.id_organiz=d.id_organiz AND a.id_amats=d.id_amats AND d.id_aktuala is NULL AND year(d.datums_iesn) >= "2014"</pre>	<pre>FROM deklaracija</pre>
Ierakstu skaits tabulā “Nekustamie īpašumi” (nekust_ip)	<pre>SELECT count(*) FROM nekust_ip1 WHERE id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN (“342, 345, 347”) AND year(d.datums_iesn) >= “2014”)</pre>	<pre>SELECT count(*) FROM nekust_ip</pre>
Ierakstu skaits tabulā “Transporta līdzekļi” (transp_lidz)	<pre>SELECT count(*) FROM transp_lidz2 WHERE id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN (“342, 345, 347”) AND year(d.datums_iesn) >= “2014”)</pre>	<pre>SELECT count(*) FROM transp_lidz</pre>
Ierakstu skaits tabulā “Iznomātie/nomātie īpašumi” (iznom_ip)	<pre>SELECT count(*) FROM iznom_ip3 WHERE id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat</pre>	<pre>SELECT count(*) FROM iznom_ip</pre>

Parametrs	Datu avots	PDB
	<pre> AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN ("342, 345, 347") AND year(d.datums_iesn) >= "2014") </pre>	
Ierakstu skaits tabulā "Ieņemamie amati" (amati)	<pre> SELECT count(*) FROM ienem_amati4 a, organiza o WHERE a.id_organiz=o.id_organiz AND a.id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN ("342, 345, 347") AND year(d.datums_iesn) >= "2014") </pre>	<pre> SELECT count(*) FROM amati </pre>
Ierakstu skaits tabulā "Ienākumi" (ienakumi)	<pre> SELECT count(*) FROM ienakumi5 i, organiza o, valuta v WHERE i.id_organiz=o.id_organiz AND v.id_valuta=i.id_valuta AND i.id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN ("342, 345, 347") AND year(d.datums_iesn) >= "2014") </pre>	<pre> SELECT count(*) FROM ienakumi </pre>
Ierakstu skaits tabulā "Kapitāla daļas un vērtspapīri" (kap_dalas)	<pre> SELECT count(*) FROM kap_dalas7 k, organiza o WHERE k.id_organiz=o.id_organiz AND k.id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, </pre>	<pre> SELECT count(*) FROM kap_dalas </pre>

Parametrs	Datu avots	PDB
	<pre> deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN ("342, 345, 347") AND year(d.datums_iesn) >= "2014") </pre>	
Ierakstu skaits tabulā "Skaidrās naudas uzkrājumi" (nauda)	<pre> SELECT count(*) FROM nauda8 n, valuta v WHERE ((v.id_valuta=n.id_valsk AND n.id_valsk<>-1) OR (n.id_valbezsk=v.id_valuta AND n.id_valbezsk<>-1) OR (v.id_valuta=n.id_valbezsk AND n.id_valsk=-1 AND n.id_valbezsk=-1)) AND n.id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN ("342, 345, 347") AND year(d.datums_iesn) >= "2014") </pre>	<pre> SELECT count(*) FROM nauda </pre>
Ierakstu skaits tabulā "Radinieki" (radinieki)	<pre> SELECT count(*) FROM radinieki9 r, radniecibas_veids v, deklaracija d WHERE r.id_radnveids=v.id_radnveids AND d.id_dekl=r.id_dekl and (d.pirms_51='1' or (d.pirms_51='0' and r.nepilng='0')) AND r.id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL </pre>	<pre> SELECT count(*) FROM radinieki </pre>

Parametrs	Datu avots	PDB
	<pre>AND dk.numurs NOT IN ("342, 345, 347") AND year(d.datums_iesn) >= "2014")</pre>	
Ierakstu skaits tabulā "Darījumi" (darijumi)	<pre>SELECT count(*) FROM publ_darijumi pd, valuta v WHERE pd.id_valuta = v.id_valuta AND pd.id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN ("342, 345, 347") AND year(d.datums_iesn) >= "2014");</pre>	<pre>SELECT count(*) FROM darijumi</pre>
Ierakstu skaits tabulā "Nekustamo īpašumu līdzīpašnieki" (nekustip_sat)	<pre>SELECT count(*) FROM nekustip_sat n, organiza o WHERE n.id_organiz=o.id_organiz AND id_nekustip IN (SELECT id_nekustip FROM nekust_ip1 WHERE id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN ("342, 345, 347") AND year(d.datums_iesn) >= "2014"))</pre>	<pre>SELECT count(*) FROM nekustip_sat</pre>
Ierakstu skaits tabulā "Informācija" (informacija)	<pre>SELECT count(*) FROM informacija WHERE id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL</pre>	<pre>SELECT count(*) FROM informacija</pre>

Parametrs	Datu avots	PDB
	<i>AND dk.numurs NOT IN (“342, 345, 347”) AND year(d.datums_iesn) >= “2014”);</i>	
Ierakstu skaits tabulā "Parādsaistības" (publ_paradi)	<i>SELECT count(*) FROM publ_paradi pp, valuta v WHERE pp.id_valuta = v.id_valuta AND pp.id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN (“342, 345, 347”) AND year(d.datums_iesn) >= “2014”);</i>	<i>SELECT count(*) FROM publ_paradi</i>
Ierakstu skaits tabulā "Aizdevumi" (publ_aizd)	<i>SELECT count(*) FROM publ_aizd pa, valuta v WHERE pa.id_valuta = v.id_valuta AND pa.id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN (“342, 345, 347”) AND year(d.datums_iesn) >= “2014”);</i>	<i>SELECT count(*) FROM publ_aizd</i>
Ierakstu skaits tabulā "Patiesā labuma guvēji" (pat_labguv)	<i>SELECT count(*) FROM pat_labguv WHERE id_dekl IN (SELECT distinct d.id_dekl FROM monitors m, dekl_kateg dk, deklaracija d WHERE dk.id_deklkat=m.id_deklkat AND d.id_dekl=m.id_dekl AND d.dal_ievadita<>'I' AND d.id_aktuala is NULL AND dk.numurs NOT IN (“342, 345, 347”) AND year(d.datums_iesn) >= “2014”)</i>	<i>SELECT count(*) FROM pat_labguv</i>

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Deklarāciju veidi" (<i>dekl_veids</i>)	<i>SELECT count(*) FROM dekl_veidi</i>	<i>SELECT count(*) FROM dekl_veids</i>

2.4.4.6. SLO dati

2.6. tabula

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā „Sabiedriskā labuma organizācijas” (<i>slo_registr</i>)	<i>SELECT count(*) FROM slo_registr r, slo_iesnlem i WHERE r.id_sloregistr=i.id_sloregistr AND i.ier_stat='A' AND i.id_sloregistr>-1 AND i.id_nakapstlem=-1;</i>	<i>SELECT count(*) FROM slo_registr;</i>
Ierakstu skaits tabulā „Lēmumi par sabiedriskā labuma organizācijas statusa piešķiršanu/atņemšanu” (<i>slo_iesnlem</i>)	<i>SELECT count(*) FROM slo_registr r, slo_iesnlem i, slo_lemveids l WHERE r.id_sloregistr=i.id_sloregistr AND i.id_slolemveids=l.id_slolemveids AND l.kods IN (select l.kods from slo_lemveids l where l.kods in ('3', '6', '7', '9')) AND i.ier_stat='A' AND i.id_sloregistr>-1;</i>	<i>SELECT count(*) FROM slo_iesnlem;</i>
Ierakstu skaits tabulā „Sabiedriskā labuma organizācijas darbības jomas” (<i>slo_darbjoma</i>)	<i>SELECT FIRST 1 ((Select count(*) from slo_darbjoma d, sablabs_darbjoma s, slo_iesnlem i_p, slo_iesnlem i where i.ier_stat='A' AND i.id_sloregistr>-1 AND i.id_nakapstlem=-1 AND d.id_sloiesnlem=i.id_sloiesnlem AND d.darbiba in ('0','1') AND d.id_sablabs_darbjoma=s.id_sablabs_darbjoma AND d.id_pievlem=i_p.id_sloiesnlem) + (SELECT count(*) from slo_darbjoma d, slo_iesnlem i_pienemt, slo_iesnlem i_likv WHERE i_likv.ier_stat='A' AND i_likv.id_sloregistr>-1 AND i_likv.id_nakapstlem=-1 AND i_likv.id_sloiesnlem=i_pienemt.id_nakapstlem AND i_pienemt.id_sloiesnlem=d.id_sloiesnlem AND d.darbiba in ('0','1') and i_likv.dok_tips IN ('2', '4'))</i>	<i>SELECT count(*) FROM slo_darbjoma;</i>

Parametrs	Datu avots	PDB
) <i>FROM slo_darbjoma;</i>	
Ierakstu skaits tabulā „SLO darbības pārskati un turpmākās darbības plāni” (<i>slo_darbparsk</i>)	<i>SELECT count(*) FROM slo_darbparsk WHERE ier_stat='A' and apst_stat='1';</i>	<i>SELECT count(*) FROM slo_darbparsk;</i>
Ierakstu skaits tabulā „Sabiedriskā labuma darbības jomas” (<i>sablab_darbjoma</i>)	<i>SELECT count(*) FROM sablab_darbjoma;</i>	<i>SELECT count(*) FROM sablab_darbjoma;</i>

2.4.4.7. SDV dati

2.7. tabula

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā „Saimnieciskās darbības veicēji” (<i>sdv_fp</i>)	<i>EXECUTE PROCEDURE pdb_get_count ('sdv_fp');</i>	<i>SELECT count(*) FROM sdv_fp;</i>
Ierakstu skaits tabulā „Saimnieciskās darbības apturēšana un atjaunošana” (<i>sd_aptatj</i>)	<i>EXECUTE PROCEDURE pdb_get_count ('sd_aptatj');</i>	<i>SELECT count(*) FROM sd_aptatj;</i>

2.4.4.8. SNK dati

2.8. tabula

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Nodokļu maksātāju (komersantu) taksācijas gadā samaksāto VID administrēto nodokļu kopsummas" (<i>nmk_snk</i> (NIS))	<i>SELECT COUNT(*) FROM pnm_pdbdati pnm, gads g WHERE pnm.id_gads=g.id_gads AND g.gada_num>=year(current - 3 units year);</i>	<i>SELECT COUNT(*) FROM nmk_snk WHERE avots = 'NIS'</i>
Ierakstu skaits tabulā "Nodokļu maksātāju (komersantu) taksācijas gadā samaksāto VID administrēto nodokļu	<i>SELECT COUNT(*) FROM nm_iamaksas_pdb nmi, iamaksas_pdb i WHERE nmi.id_iamaksaspdb=i.id_iamaksaspdb AND i.gads>=year(current - 3 units year);</i>	<i>SELECT COUNT(*) FROM nmk_snk WHERE avots = 'CMIS'</i>

Parametrs	Datu avots	PDB
kopsummas" (<i>nmk_snk</i> (CMIS))		

2.4.4.9. DAK dati

2.9. tabula

Parametrs	Datu avots (NIS DB)	PDB
Ierakstu skaits tabulā "Darījumu apliecinošās kvītis" (<i>dak_reg</i>)	<i>SELECT COUNT(*)</i> <i>FROM dak_registr</i> <i>WHERE dok_stat = '.'</i>	<i>SELECT COUNT(*)</i> <i>FROM dak_reg</i>
Ierakstu skaits tabulā "Darījumu reģistrētās kvītis" (<i>dak_regr</i>)	<i>SELECT COUNT(*)</i> <i>FROM dak_regrinda</i>	<i>SELECT COUNT(*)</i> <i>FROM dak_regr</i>

2.4.4.10. MUN dati

2.10. tabula

Parametrs	Datu avots (NIS DB)	PDB
Ierakstu skaits tabulā "Mikrouzņēmumu nodokļa maksātāji" (<i>mun_reg</i>)	<i>SELECT COUNT(*)</i> <i>FROM nodokl_maks nm,</i> <i>nodoklis n,</i> <i>nodokl_veids v</i> <i>WHERE nm.id_nodoklmaks =</i> <i>n.id_nodoklmaks</i> <i>and n.id_nodoklveids =</i> <i>v.id_nodoklveids</i> <i>and v.kods = 7200</i> <i>and nm.ier_stat = "A"</i> <i>and (n.ier_stat = "A"</i> <i>OR (n.ier_stat='S' AND n.svitrots='0'</i> <i>AND n.izsl_datums IS NOT NULL))</i>	<i>SELECT COUNT(*)</i> <i>FROM mun_reg</i>

2.4.4.11. VAP dati

2.11. tabula

Parametrs	Datu avots (VADIS DB)	PDB
Ierakstu skaits tabulā "Valsts amatpersonu pārkāpumi" (<i>vap_reg</i>)	<pre>SELECT COUNT(*) FROM parkapumi WHERE drikst_publ = '1' AND (izp_datums >= TODAY - 365 OR izp_datums is NULL)</pre>	<pre>SELECT COUNT(*) FROM vap_reg</pre>

2.5. Nefunkcionālo prasību realizācija

2.12. tabula

Prasības numurs	Prasības realizēšana
NF.1.	Vietnes saturs grafiski attēlojas caur interneta pārlūku.
NF.2.	Tīmekļa lapas saturs ir attēlots latviešu valodā.
NF.3.	Vietnes tiek veiktota tā, lai lietotājam tā būtu viegli saprotama un pārskatāma.
NF.4.	Lapai ir universāls noformējums, kas attainosies citos skatos.
NF.5.	HTML kods ir rakstīts atbilstoši W3C HTML5 standartam
NF.6.	CSS kods ir rakstīts atbilstoši W3C CSS3 standartam

2.6. Prasību trasējamības matrica

2.13. tabula

Prasība	Realizācija (entitija)
create_database_data_comparison	Home, Select, Index
return_JSON_report	Home, Select
NF.1.	Skat. 2.5. Nefunkcionālo prasību realizācija
NF.2.	
NF.3.	
NF.4.	
NF.5.	
NF.6.	

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Ievads

3.1.1. Nolūks

Testēšanas dokumentācijas nolūks ir aprakstīt VID Publiskojamo datu bāzes administrēšanas moduļa vienībtestēšanas rezultātus. Testēšanas mērķis ir nodrošināt PPS minēto sistēmas funkciju izpildi.

3.1.2. Testēšanas metodika

Testēšanas žurnālā apkopoti automātiski (ar pirmkoda palīdzību) veiktie vienībtesti, to skaidrojums, padotais parametrs un iznākums, manuāli veiktie akcepttesti ar skaidrojumu par to, kad tests ir veismīgs, padodamo parametru un iznāku. Testēšanas žurnālā apkopoti testu iznākumi, kas veiksti pēc programmatūras izstrādes. Veiksmīga testa izpildes gadījumā iznākums ir "+", savukārt neveiksmīga testa izpildes vai nepareiza rezultāta gadījumā iznākums ir "-".

Sistēmas testēšana ir veikta pēc "pelēkas kastes" principa. Sistēmas testēšanā tika pārbaudīta sistēmas funkcionalitāte, izmantojot funkcionālas prasības un projektējumu. Sistēmas funkcijas tika testētas izmantojot korektus un nekorektus ievaddatus.

Izstrādātājs arī pārbauda HTML un CSS failu atbilstību PPS uzstādītajiem HTML5 un CSS3 standartiem pret attiecīgi HTML un CSS kodu.

3.1.3. Definīcijas

Definīciju un apzīmējumu sarakstu iespējams aplūkot nodaļā "Vārdnīca".

3.1.4. Saistība ar citiem dokumentiem

Testēšanas dokumentācija izstrādāta balstoties uz programmatūras prasību specifikācijā minētajām funkcionālām prasībām, nefunkcionālajām un programmatūras projektējuma aprakstu.

Dokumenta uzstādē un noformēšanā izmantotas "LVS 70:1996 Programmatūras testēšanas dokumentācija" standarta prasības.

3.2. Testēšanas žurnāls

3.2.1. Vienībtesti

3.1. tabula

Testa nosaukums (testējamās prasības identifikators)	Skaidrojums	Parametrs	Iznākums
select_getCount_success (create_database_data_comparison)	Tests ir veiksmīgs, ja pēc argumenta padošanas (padodamie argumenti ir noteikti PPS) tiek atgriezts rezultāts ar veiktām datu atlasēm no abām datubāzēm.	“nm”	+
		“pvn”	+
		“ka”	+
		“lic”	+
		“vadis”	+
		“slo”	+
		“sdv”	+
		“snk”	+
		“dak”	+
		“mun”	+
“vap”	+		
select_getCount_invalidargument_emptylist (create_database_data_comparison)	Tests ir veiksmīgs, ja padodot kādu nepareizu argumentu (kāds, kas nav noteikts PPS) tiek atgriezts tukšs saraksts.	“invalid”	+

3.2.2. Akceptttesti

3.2. tabula

Testa nosaukums (testējamās prasības identifikators)	Skaidrojums	Parametrs	Iznākums
db_exec_success (return_JSON_report)	Tests ir veiksmīgs, ja veicot konkrētu pieprasījumu no konkrētās datu bāzes iegūt kodus skaitlisku rezultātu	“vid_pdb”, { "SELECT COUNT(*) FROM nodokl_maks"} }	+
getCount_success (return_JSON_report)	Tests ir veiksmīgs, ja pēc argumenta padošanas (padodamie argumenti ir noteikti nodaļā 1.3.1 “Pārskata izveidošana”) tiek atgriezts saraksts, kas satur datus ar veiktajā, datu atlasēm.	“nm”	+
		“pvn”	+
		“ka”	+
		“lic”	+
		“vadis”	+
		“slo”	+
		“sdv”	+
		“snk”	+
		“dak”	+
		“mun”	+
“vap”	+		

<p>getCount_invalidargument_emptylist (return_JSON_report)</p>	<p>Tests ir veiksmīgs, ja padodot kādu nepareizu argumentu (kāds, kas nav noteikts nodaļā 1.3.1 “Pārskata izveidošana”) tiek atgriezts tukšs saraksts.</p>	<p>“invalid”</p>	<p>+</p>
<p>getInfo_success (return_JSON_report)</p>	<p>Tests ir veiksmīgs, ja pēc argumenta padošanas (padodamie argumenti ir noteikti nodaļā 1.3.1 “Pārskata izveidošana”) tiek atgriezts rezultāts ar vektām datu atlasēm no abām datubāzēm JSON formā.</p>	<p>“nm”</p>	<p>+</p>
		<p>“pvn”</p>	<p>+</p>
		<p>“ka”</p>	<p>+</p>
		<p>“lic”</p>	<p>+</p>
		<p>“vadis”</p>	<p>+</p>
		<p>“slo”</p>	<p>+</p>
		<p>“sdv”</p>	<p>+</p>
		<p>“snk”</p>	<p>+</p>
		<p>“dak”</p>	<p>+</p>
<p>“mun”</p>	<p>+</p>		
<p>“vap”</p>	<p>+</p>		
<p>getInfo_invalidargument_emptylist (return_JSON_report)</p>	<p>Tests ir veiksmīgs, ja padodot kādu nepareizu</p>	<p>“invalid”</p>	<p>+</p>

	argumentu (kāds, kas nav noteikts PPS) tiek atgriezts tukšs JSON ar tukšu vērtību pie result.		
fillData_success (create_database_data_comparison)	Tests ir veiksmīgs, ja pēc argumenta padošanas (padodamie argumenti ir noteikti nodaļā 1.3.1 “Pārskata izveidošana”) “Index” skatā tabulas vērtības nomainas ar datu alases rezultātiem.	“nm”	+
		“pvn”	+
		“ka”	+
		“lic”	+
		“vadis”	+
		“slo”	+
		“sdv”	+
		“snk”	+
		“dak”	+
		“mun”	+
“vap”	+		

3.2.3. Nefunkcionālo prasību testēšana

3.3. tabula

Prasība	Darbība	Sagaidāmais rezultāts	Iznākums
NF.1.	Ieraksta interneta pārlūkā URL uz sistēmas administratīvo moduli.	Atveras grafiski realizēts sistēmas administratīvais modulis.	+
NF.2.	Atver administratīvo moduli.	Vietnes saturs ir latviešu valodā.	+
NF.3.	Atver administratīvo moduli.	Lapas saturs ir uzskatāms, intuitīvs un ērti lietojams.	+
NF.4.	Atver administratīvo moduli.	Lapas augšā ir galvene, kur redzams VID logo. Lapas apakšējā daļā redzams autotiesību paziņojums un informācija par sistēmas versiju.	+
NF.5.	HTML koda atbilstība HTML5 standartam tiek pārbaudīta vietnē: <ul style="list-style-type: none"> • https://validator.w3.org/ 	HTML kods atbilst HTML5 standartam	+ (Ekrānuņēmums nodaļā 10.1.1.)
NF.6.	CSS koda atbilstība CSS3 standartam tiek pārbaudīta vietnē: <ul style="list-style-type: none"> • https://jigsaw.w3.org/css-validator/ 	CSS kods atbilst CSS3 standartam	+ (Ekrānuņēmums nodaļā 10.1.2.)

4. PROJEKTA ORGANIZĀCIJA

VID publiskās datu sistēmas administrēšanas modulis tika izstrādāts pēc “ūdenskrituma” modeļa, jo darba sākumā tika uzstādītas nepieciešamās programmatūras funkcijas. Balstoties un PPS tika izstrādāts PPA uz kura savukārt balstoties tika izstrādāts programmas kods.

Izstrādātās sistēmas vienībtestēšana notika pēc sistēmas izstrādes, tāpat kā sistēmas dokumentēšana. Autors veica moduļu izstrādi patstāvīgi un arī ar darba vadītāja palīdzību. Produkta projektējumu, testēšanu, dokumentēšanu un pašu programmēšanu veica darba autors.

Projekts izstrādāts Microsoft Visual Studio 2013 izstādes vide. Kvalifikācijas darbs noformēts izmantojot Microsoft Word 2016. Pirmkoda pārvaldībai izmantots Microsoft Team Foundation Server.

5. KVALITĀTES NODROŠINĀŠANA

Lai izstrādātajā projektā būtu nodrošināta kvalitāte darba autors ievēro vairākus standartus, kas attiecas gan uz programmatūras kodu, gan tās dokumentēšanu. PPS, PPA un testēšanas dokumentācija tika izstrādāti pēc programminženierijas nozarē jau ilgi esošajiem valsts standartiem savukārt programmas kods izstādāts balstoties uz starptautiskiem koda standartiem.

Programmatūras pirmkods izstrādāts balstoties uz MVC pieeju, kas ir vienots programmēšanas stils. Programmējot izmantota objektorientētā pieeja.

Sistēma veidošanā izmantotais HTML un CSS kods atbilst attiecīgi HTML5 un CSS standartiem. Tas tika pārbaudīts interneta vietnēs, ko izstrādājusi W3C organizācija, kas arī ir šo standartu autors. Koda atbilstība standartiem veikta šajās vietnēs:

- <http://validator.w3.org>
- <http://jigsaw.w3.org/css-validator>

Sistēma tika testēta lokāli pēc svarīgu funkciju pabeigšanas, taču kad programma tika pilnībā pabeigta, tad tika testētas visas funkcijas ar dažādiem padotajiem parametriem (skat. "Testēšanas dokumentācija").

PPS, PPA un testēšanas dokumentācija tika izstādāta attiecīgi balstoties uz sekojošiem valsts standartiem:

- LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis
- LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai
- LVS 70:1996 Programmatūras testēšanas dokumentācija

6. KONFIGURĀCIJU PĀRVALDĪBA UN VERSIJKONTROLE

Par vispiemērotāko rīku sistēmas versiju parvaldībai tika izvēlēts Microsoft Team Foundation Server, kas ir iebūvēts iekš Microsoft Visual Studio – izstrādes vides, kur tika izstrādāta programmatūra. Team Foundation Server nodrošina sistēmas pirmkoda un vienībtestu glabāšanu Microsoft serverī.

Programmatūras kods tika sinhronizēts ar Microsoft serveri pēc lielu funkciju vai klašu pabeigšanas. Katrā sinhronizēšanas reizē tika pievienots komentārs ar izmaiņām.

7. DARBIETILPĪBAS NOVĒRTĒJUMS

Sistēmas izstrādes darbietilpības noteikšanai tika izmantota eksperta metode. Pa cik darba autoram nebija iepriekšējas pieredzes programmēšanā ar C# programmēšanas valodu un maza pieredze ar MVC un JavaScript un nebija pieredzes darbā ar JSON, AJAX un jQuery bija grūti plānot cik daudz laika nepieciešams, lai izstrādātu sistēmu. Attiecībā uz PPA norādītajām funkcijām tika aprēķināts, ka tās būtu iespējams izstrādāt trīs personmēnešu laikā.

Šādi bija plānots būt laika sadalījumam pa darbības jomām:

- Prasību specificēšana - 0,4 personmēneši
- Projektēšana – 0,6 personmēneši
- Kodēšana – 1 personmēneši
- Testēšana – 0,5 personmēneši
- Dokumentēšana - 0,5 personmēneši

Pēc sistēmas izveidošanas tika konstatēts, ka noteiktā darbietilpība ir tikusi pareizi novērtēta, taču laika sadalījums pa darbības jomām ir bijis cits:

- Prasību specificēšana - 0,3 personmēneši,
- Projektēšana – 0,6 personmēneši
- Kodēšana – 1,3 personmēneši,
- Testēšana – 0,4 personmēneši,
- Dokumentēšana - 0,4 personmēneši,

Darba kodēšana ir prasījusi vairāk laika nekā pirmstam paredzēts lielā programmēšanas valodu apjoma dēļ, savukārt pārējās darbības jomas ir tik cik plānots vai nedaudz mazāk nekā plānots.

8. SECINĀJUMI

Kvalifikācijas darba gaitā tika izveidots VID PDB administrēšanas modulis, kurā iespējams veikt pārskatus par ierakstu skaitu VID CDB un VID PDB, tādējādi iegūstot informāciju par to vai dati no CDB ir nonākuši PDB. Darba rezultāts ir funkcionējoša programma, kas atbilst sākotnēji izvirzītajai PPS un kas ir notestēta.

Strādājot pie kvalifikācijas darba tika saprāts, ka programmēšanas labās prakses ievērošana atvieglo koda lasīšanu, taču labi pārlasāma un kvalitatīva koda izstrāde prasa daudz laika, jo tiek patērēts ievērojams laiks domājot vai nav iespējams programmu optimizēt un padarīt vēl vienkāršāku.

Kvalifikācijas darba izstrādes gaitā tika iegūts daudz jaunu zināšanu gan par programmatūras izstrādes procesu kā tādu, gan arī apgūts daudz jauna saistībā ar programmēšanu. Kā piemēram JavaScript bibliotēka jQuery, AJAX pieprasījuma veidošana, JSON objekta iegūšana un programmēšana C# valodā.

9. IZMANTOTO AVOTU UN LITERATŪRAS SARAKSTS

1. Asp.net ajax JSON Example, C# WebMethod with sql database – Codepedia
<http://codepedia.info/jquery-ajax-json-example-asp-net-sql-database/> [Skatīts: 14.04.2016]
2. ASP.NET MVC Introduction
http://www.w3schools.com/aspnet/mvc_intro.asp [Skatīts: 20.02.2016]
3. ASP.NET Razor C# Syntax
http://www.w3schools.com/aspnet/razor_syntax.asp [Skatīts: 02.05.2016]
4. C# - Basic Syntax
http://www.tutorialspoint.com/csharp/csharp_basic_syntax.htm [Skatīts: 02.03.2016]
5. C# ADO.NET ODBC Tutorial
<http://www.easysoft.com/developer/languages/csharp/ado-net-odbc.html> [Skatīts: 06.04.2016]
6. Create and run unit tests with Visual Studio | Visual Studio Team Services
<https://www.visualstudio.com/en-us/docs/tfvc/create-and-run-unit-tests-vs> [Skatīts: 19.05.2016]
7. HTML5 Migration
http://www.w3schools.com/html/html5_migration.asp [Skatīts: 17.05.2016]
8. Ieteicamā prakste programmatūras projektējuma aprakstīšanai, LVS 72:1996, Latvijas Valsts standarts
9. jQuery API Documentation
<http://api.jquery.com/> [Skatīts: 24.05.2016]
10. Programmatūras prasību specifikācijas ceļvedis, LVS 68:1996, Latvijas Valsts standarts
11. Programmatūras testēšanas dokumentācija, LVS 70:1996, Latvijas Valsts standarts

10. PIELIKUMI

10.1. Koda atbilstība standartiem

10.1.1. HTML koda atbilstība HTML5 standartam

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contents of text-input area

Checker Input

Show source outline image report

Check by: text input

```
<!DOCTYPE html>
<html>
<head>
  <title>Informācijas viedums</title>
  <meta charset="utf-8">
  <link href="/Content/style.css" rel="stylesheet" type="text/css" />
  <script src="Scripts/jquery-1.8.2.min.js" type="text/javascript"></script>
  <script src="Scripts/scripts.js" type="text/javascript"></script>
</head>
<body>
  <div id="container">
    <!-- Header -->
    <header>
      <div class="print">
```

Document checking completed. No errors or warnings to show.

10.1.2. CSS koda atbilstība CSS3 standartam

W3C The W3C CSS Validation Service
W3C CSS Validator results for TextArea (CSS level 3)

Jump to: [Validated CSS](#)


W3C CSS Validator results for TextArea (CSS level 3)

Congratulations! No Error Found.

This document validates as [CSS level 3](#)

10.2. Ekrānformas

10.2.1. Tikko atvērts pārskats bez veikta pieprasījuma pēc datiem



Jūs – godīgi maksājat nodokļus,
mēs – profesionāli apkalpojam!

Par VID Aktualitātes Nodokļi Muita Kontakti

Informācijas vienumi

NM dati

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Nodokļu maksātājs" (nodokl_maks)	Nav datu	Nav datu

Atjaunot

PVN dati

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Pievienotās vērtības nodokļa maksātāji" (pvn)	Nav datu	Nav datu

Atjaunot

KA dati

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "KA markas, modeļi un modifikācijas" (ka_modelis)	Nav datu	Nav datu
Ierakstu skaits tabulā "Kases aparātus apkalpojošie dienesti" (ka_tirgserv)	Nav datu	Nav datu
Ierakstu skaits tabulā "Atļautās KA markas, modeļi un modifikācijas" (ka_atmodeli)	Nav datu	Nav datu
Ierakstu skaits tabulā "Kases aparāti" (ka_reg)	Nav datu	Nav datu

Atjaunot

Licenču dati

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Licences" (apmd_lic)	Nav datu	Nav datu
Ierakstu skaits tabulā "Darbības vietas tips" (apmd_licviettips)	Nav datu	Nav datu
Ierakstu skaits tabulā "Darbības vieta" (apmd_licvieta)	Nav datu	Nav datu
Ierakstu skaits tabulā "Akcīzes preču veids" (ap_prece)	Nav datu	Nav datu
Ierakstu skaits tabulā "Darbības veids ar akcīzes precēm" (ap_darbveids)	Nav datu	Nav datu
Ierakstu skaits tabulā "Speciālajā atļaujā norādītās preces" (ap_nolprece)	Nav datu	Nav datu
Ierakstu skaits tabulā "Speciālajā atļaujā norādītie darbības veidi" (ap_noldveidi)	Nav datu	Nav datu
Ierakstu skaits tabulā "Licenču īpašās atzīmes par darbībām" (ap_ipatzpardarb)	Nav datu	Nav datu
Ierakstu skaits tabulā "Speciālo atļauju apturēšana" (apmd_licapt)	Nav datu	Nav datu

Atjaunot

10.2.2. Pārskats pēc datu pieprasīšanas

Informācijas vienumi

NM dati

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Nodokļu maksātājs" (nodokl_maks)	2181742	2181814

Atjaunot

PVN dati

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Pievienotās vērtības nodokļa maksātāji" (pvn)	90523	90521

Atjaunot

KA dati

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "KA markas, modeļi un modifikācijas" (ka_modelis)	572	569
Ierakstu skaits tabulā "Kases aparātus apkalpojošie dienesti" (ka_tirgserv)	212	213
Ierakstu skaits tabulā "Atļautās KA markas, modeļi un modifikācijas" (ka_atimodeli)	2355	2352
Ierakstu skaits tabulā "Kases aparāti" (ka_reg)	10361	10329

Atjaunot

Licenču dati

Parametrs	Datu avots	PDB
Ierakstu skaits tabulā "Licences" (apmd_lic)	44662	44657
Ierakstu skaits tabulā "Darbības vietas tips" (apmd_licviettips)	277	277
Ierakstu skaits tabulā "Darbības vieta" (apmd_licvieta)	11928	11924
Ierakstu skaits tabulā "Akcīzes preču veids" (ap_prece)	29	29
Ierakstu skaits tabulā "Darbības veids ar akcīzes precēm" (ap_darbveids)	47	47
Ierakstu skaits tabulā "Speciālajā atļaujā norādītās preces" (ap_nolprece)	1083	1074
Ierakstu skaits tabulā "Speciālajā atļaujā norādītie darbības veidi" (ap_noldveidi)	1271	1269
Ierakstu skaits tabulā "Licenču īpašās atzīmes par darbībām" (ap_ipatzpardon)	104	1269
Ierakstu skaits tabulā "Speciālo atļauju apturēšana" (apmd_licapt)	119	119

Atjaunot

10.3. Programmatūras pirmkoda piemēri

10.3.1. "Home" kontrolieris

```
namespace VID.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index() // Pārskata aplūkošanai
        {
            return View();
        }
    }
}
```

```

    }

    public JsonResult getInfo(string unit) // JSON datu saņemšanai
    {
        return Json(new { result = select.getCount(unit) },
            JsonRequestBehavior.AllowGet);
    }
}

```

10.3.2. “aUnits” klase

```

namespace VID.Managers
{
    // realizet ar get set properties
    public enum _rqstNames { nm, pvn, ka, lic, vadis, slo, sdv, snk, dak,
mun, vap }; // ir konkrēti definēti pieprasījumi, kas glabājas enum sarakstā
    public class aUnits
    {
        // Klases iekšējie mainīgie
        private string _header;
        private string _query2;
        private string _dbName;

        public _rqstNames rqstName; // Pieprasījuma nosaukums
        public string header // Datu atlasē nosaukums
        {
            get { return _header; }
            set { _header = value; }
        }
        public string[] queries1; // Izpildāmie SQL pieprasījumi priekš datu
atlasē no CDB
        public string query2 // Izpildāmie SQL pieprasījumi priekš datu
atlasē no PDB
        {
            get { return _query2; }
            set { _query2 = value; }
        }
        public string dbName // Atsevišķos objektos glabājas datu bāzes
nosaukums
        {
            get { return _dbName; }
            set { _dbName = value; }
        }
    }
}

```

10.3.2.1. “aUnits” objekta piemērs

```
        aUnitsArray[40] = new aUnits()
        {
rqstName = _rqstNames.snk,

header = "Ierakstu skaits tabulā \"Nodokļu maksātāju (komersantu) taksācijas
gadā samaksāto VID administrēto nodokļu kopsummas\" (nmk_snk (CMIS))",

queries1 = new string[] { "SELECT COUNT(*) FROM nm_iemaksas_pdb nmi,
iemaksas_pdb i WHERE nmi.id_iemaksaspdb=i.id_iemaksaspdb AND
i.gads>=year(current - 3 units year);" },

query2 = "SELECT COUNT(*) FROM nmk_snk WHERE avots = 'CMIS'",
        dbName = "cmis"

    }
}
```

10.3.3. “Index” skats

```
@{
    ViewBag.Title = "Informācijas vienumi"; // Lapas nosaukums
    var aUnitsArray = VID.Managers.select.getaUnitsArray(); // Iegūst masīvu
ar aUnits klases objektiem.
    var aGroupsArray = VID.Managers.select.getaGroupsArray(); // Iegūst
masīvu, kas satur informācijas vienuma nosaukumu un pieprasījumu tam.
}
@for (int i = 0; i < aGroupsArray.GetLength(0); i++) // Lai izveidotu tabulas
ar datu atlasē veidiem no CDB un PDB
{
    <h2>@aGroupsArray[i,1]</h2>
    <table id="@aGroupsArray[i,0]">
        <thead>
            <tr>
                <th>Parametrs</th>
                <th>Datu avots</th>
                <th>PDB</th>
            </tr>
        </thead>
        <tbody>
            @for (int a = 0; a < aUnitsArray.Length; a++)
            {
```

```

        if (aUnitsArray[a].rqstName.ToString() == aGroupsArray[i, 0])
// Atkarībā no pieprasījuma izveido attiecīgās datu tabulas rindas ar datu
// atlasēm.
        {
            <tr>
                <td>@aUnitsArray[a].header</td>
                <td class="resultData">Nav datu</td>
                <td class="resultData">Nav datu</td>
            </tr>
        }
    }
</tbody>
</table>

    <p><input type="button" value="Atjaunot"
onclick="fillData('@aGroupsArray[i,0]')" /></p> // Katrai pogai ir savs
parametrs ko tā padod to nospiežot

}

```

10.3.4. "fillData" funkcija

```

function fillData(unit) { // Saņem parametru no nospiešanās pogas
    $.ajax('home/getInfo?unit=' + unit, { // Pieprasa JSON datus no "Home"
// kontrolierā
        success: function (data) { // Veiksmīgi saņemot JSON objektu
            for (var i = 0; i < data.result.length; i++) {
                var row = $('#' + unit + ' tr:has(td:contains("' +
data.result[i].header + '"'))' // Atlasa rindu, kas satur konkrēto datu
// pieprasījumu
                row.find('td:eq(1)').html(data.result[i].DatuAvots) //
// Izmaina programmas HTML uz iegūto rezultātu no CDB
                row.find('td:eq(2)').html(data.result[i].PDB) // Izmaina
// HTML uz iegūto rezultātu no PDB
            }
        },
        // Rodoties kļūdai JSON objekta saņemšanā, lauki tiek aizpildīti par
// paziņojumu "Kļūda"
        error: function () {
            var rows = $('#' + unit + ' tr');
            for (var j = 1; j < rows.length; j++) {
                rows[j].find('td:eq(1)').html("Kļūda")
                rows[j].find('td:eq(2)').html("Kļūda")
            }
        }
    })
}

```

10.3.5. "db_exec" funkcija

```
public static int db_exec(string strDataSource, string[] strSQL) // Veic SQL
pieprasījumu noteiktai datubāzei
{
    string DB_ROLE;
    int result = 0;
    using (OdbcConnection DbConnection = new OdbcConnection("DSN=" +
strDataSource)) // Izveido savienojumu un darbību beigās to aizver
    {
        DbConnection.Open();
        OdbcCommand DbCommand = DbConnection.CreateCommand();
        // Nosaka, ar kādu DB lietotāja lomu tiks izpildīti
pieprasījumi datu b_zei.
        switch (strDataSource)
        {
            case "vadis":
                DB_ROLE = "dn";
                break;
            case "vid":
                DB_ROLE = "allsel";
                break;
            case "vid_pdb":
                DB_ROLE = "pdbuser";
                break;
            case "cmis":
                DB_ROLE = "dn";
                break;
            default:
                DB_ROLE = "pdbuser";
                break;
        }

        DbCommand.CommandText = "SET ROLE " + DB_ROLE;
        DbCommand.ExecuteNonQuery();

        for (int i = 0; i < strSQL.Length; i++) // Izpilda visus
strSQL masīvā esošos pieprasījumus
        {
            if (i == strSQL.Length - 1)
            {
                DbCommand.CommandText = strSQL[i];
                DbCommand.CommandTimeout = 1000;
                OdbcDataReader DbReader = DbCommand.ExecuteReader();
                DbReader.Read();
                result = DbReader.GetInt32(0);
            }
            else
            {
                DbCommand.CommandText = strSQL[i];
            }
        }
    }
}
```

```

        DbCommand.ExecuteNonQuery();
    }
}
return result;
}

```

10.3.6. “getCount” funkcija

```

public static List<result> getCount(string strUnit) // Izveido sarakstu ar
objektiem, kuros glabājas datu atlasēs rezultāti
{
    // Mainīgo deklarācija, kuros glabāsiet datu atlasēs rezultāti
    int DArS;
    int PDBrS;

    List<result> resultList = new List<result>(); // Saraksta
deklarēšana

    aUnits[] aUnitsArray = getaUnitsArray(); //Iegūst masīvu ar SQL
pieprasījumiem

    if ((strUnit != ""))
    {
        for (int i = 0; i < aUnitsArray.Length; i++) //Iet cauri
aUnits masīvam meklējot īstos SQL pierprasījumus
        {
            if ((aUnitsArray[i].rqstName.ToString() == strUnit))
            {
                if (((strUnit == "vadis") || (strUnit == "vap")))
                {
                    DArS = db_exec("vadis", aUnitsArray[i].queries1);
// Kad īstais pierprasījums atrasts, izpilda pierprasījumu no CDB
                }
                else if ((strUnit == "seed"))
                {
                    DArS = db_exec("es_dss",
aUnitsArray[i].queries1);
                }
                else if ((strUnit == "snk"))
                {
                    DArS = db_exec(aUnitsArray[i].dbName,
aUnitsArray[i].queries1);
                }
                else
                {
                    DArS = db_exec("vid", aUnitsArray[i].queries1);
                }
            }
        }
    }
}

```

```

        PDBrs = db_exec("vid_pdb", new string[] {
aUnitsArray[i].query2 }); // Izpilda pieprasījumu no PDB

        resultList.Add(new result() //Iegūtos skaitļus
pievieno sarakstam
        {
            header = aUnitsArray[i].header,
            DatuAvots = DArS,
            PDB = PDBrs,
        });
    }
}
return resultList;
}

```

Kvalifikācijas darbs „*Publisko datu sistēmas administrēšanas moduļa uzlabošana*” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Andris Jakobovskis* _____ **30.05.2016.**

Rekomendēju darbu aizstāvēšanai

Darba vadītājs **B.Dat Jevgēnijs Trofimovs** _____ **30.05.2016.**

Recenzents: *dipl.inž. Miks Rozenbergs*

Darbs iesniegts 30.05.2016.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2016. prot. Nr. _____

Komisijas sekretārs(-e): _____