

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**DATU IZMAIŅU PĀRLŪKOŠANAS MODUĻA IZSTRĀDE
NEKUSTAMĀ ĪPAŠUMA NODOKĻA ADMINISTRĒŠANAS
SISTĒMAI NINO**

KVALIFIKĀCIJAS DARBS

Autors: **Lauma Svilpe**

Studenta apliecības Nr.: ls16043

Darba vadītājs: M.inž. Viktors Salijs

RĪGA 2018

ANOTĀCIJA

Darbā aprakstīta izstrādes gaita nekustamā īpašuma nodokļa administrēšanas sistēmas NINO papildinājuma – datu izmaiņu pārlūkošanas moduļa - risinājumam. Tas paredzēts NINO lietotājiem ērtākai un savām vajadzībām pielāgojamai datu izmaiņu vēstures pārlūkošanai.

Risinājums atvieglo gan Latvijas pašvaldību nodokļu inspektoru darbu, gan šo darbinieku kontroli ar datu izmaiņām saistītos jautājumos, kā arī samazina nepieciešamību griezties pēc palīdzības pie NINO klientu konsultantiem. Datu izmaiņu vēsturi iespējams attēlot uz ekrāna vai izklājlapu lietotnē Microsoft Excel, saņemot informāciju, kāds lietotājs, kad un kā izmainījis konkrēta NINO objekta datus.

Programmprodukta izstrāde veikta, ievērojot ūdenskrituma dzīves cikla metodoloģijas pamatprincipus. Programmkoda izstrāde balstās uz *Oracle Database* un *Microsoft Visual Basic* tehnoloģijām.

Atslēgvārdi: NINO, datu izmaiņu vēstures pārlūkošana, ūdenskrituma metodoloģija, Oracle PL/SQL, Microsoft Visual Basic.

ABSTRACT

DEVELOPMENT OF A DATA CHANGES BROWSING MODULE FOR REAL ESTATE TAX ADMINISTRATION SYSTEM NINO

The paper describes the development of a solution – data changes browsing module, which is intended for more convenient and customizable browsing for the history of data changes in the real estate tax administration system NINO for its users.

The solution facilitates tax inspector work in local governments of Latvia as well as employee control over data modifications, and reduces the need to turn to NINO client consultants for help. The history of data changes shows when and how data has been changed and which user has changed it. The history can be displayed in NINO or in Microsoft Excel spreadsheet.

The solution has been developed using waterfall methodology. The development is based on *Oracle Database* and *Microsoft Visual Basic* technologies.

Keywords: NINO, browsing for data changes history, waterfall methodology, *Oracle PL/SQL*, *Microsoft Visual Basic*.

SATURS

IEVADS	6
VĀRDNĪCA	8
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	9
1.1. Ievads	9
1.1.1. Nolūks.....	9
1.1.2. Darbības sfēra	9
1.1.3. Definīcijas.....	9
1.1.4. Saistība ar citiem dokumentiem	9
1.2. Vispārējs apraksts	9
1.2.1. Produkta perspektīva	9
1.2.2. Produkta funkcijas	10
1.2.3. Lietotāja raksturozīmes.....	10
1.2.4. Vispārējie ierobežojumi.....	10
1.2.5. Pieņēmumi un atkarības	11
1.3. Funkcionālās prasības	11
1.3.1. Ārējās saskarnes prasības	11
1.3.1.1. Lietotāja saskarne	11
1.3.1.2. Programmatūras saskarne	13
1.3.2. Sistēmas paziņojumi	13
1.3.3. Formas izsaukšanas modulis (FIM).....	14
1.3.4. Meklēšanas parametru modulis (MPM)	15
1.3.5. Datu atlases modulis (DAM).....	18
1.4. Nefunkcionālās prasības	19
1.4. Izmaiņas NINO vienumos	19
2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....	20
2.1. Ievads	20
2.1.1. Nolūks.....	20
2.1.2. Darbības sfēra	20
2.1.3. Definīcijas.....	20
2.2. Saistība ar citiem dokumentiem.....	20
2.3. Dekompozīcijas apraksts	20
2.3.1. Moduļu dekompozīcija	22
2.3.2. Procesu dekompozīcija	22
2.3.2.1. Datu izmaiņu pārlūkošanas process.....	22
2.3.2.2. Konfigurāciju ģenerēšanas process	23
2.3.3. Datu dekompozīcija	23

2.3.3.1.	Tabulas konfigurācija	23
2.3.3.2.	Meklēšanas parametri	23
2.3.3.3.	Tabula DATA_CHANGES_BROWSER.....	24
2.3.3.4.	Tabula DCB_USER_SETTINGS.....	24
2.4.	Atkarības apraksts.....	25
2.4.1.	Starpmoduļu atkarības.....	25
2.4.2.	Datu atkarības	26
2.5.	Saskarnes apraksts.....	28
2.6.	Detalizētais projektējums.....	29
2.6.1.	Datu detalizētais projektējums.....	29
2.6.1.1.	Tabula DATA_CHANGES_BROWSER.....	29
2.6.1.2.	Tabula DCB_USER_SETTINGS.....	30
2.6.1.3.	Tabulu konfigurācijas	31
2.6.2.	Funkcionālais projektējums.....	34
2.6.2.1.	Pakotne DATA_CHANGES_BROWSER_PCK	36
2.6.2.2.	Pakotne DCB_USER_SETTINGS_PCK	39
2.6.2.3.	Pakotne DATA_CHANGES_BROWSER_CONF_PCK	41
2.6.2.4.	Pakotne DATA_CHANGES_BROWSER_CONST_PCK.....	44
2.6.2.5.	Forma frmDataChangesBrowser	44
2.6.2.6.	Izmaiņas NINO vienumos	49
2.6.2.7.	Pastāvošo NINO risinājumu izmantošana	51
2.7.	Trasējamības tabula	52
3.	TESTĒŠANAS DOKUMENTĀCIJA	53
3.1.	Ievads	53
3.1.1.	Nolūks.....	53
3.1.2.	Darbības sfēra.....	53
3.1.3.	Definīcijas.....	53
3.2.	Saistība ar citiem dokumentiem.....	53
3.3.	Testēšanas plāns.....	53
3.3.1.	Automatizēta vienībtestēšana	54
3.4.	Testēšanas žurnāls.....	57
3.4.1.	PPS funkcionālā testēšana	57
3.4.2.	Augsta līmeņa funkcionālā testēšana.....	63
3.5.	Testēšanas kļūdu apstrāde.....	67
3.6.	Testēšanas kopsavilkuma pārskats.....	68
4.	PROJEKTA ORGANIZĀCIJA.....	69
5.	KVALITĀTES NODROŠINĀŠANA	70
6.	KONFIGURĀCIJU PĀRVALDĪBA	72

6.1. Dokumentācijas konfigurāciju pārvaldība	72
6.2. Programmkoda konfigurācijas pārvaldība	72
7. DARBIETILPĪBAS NOVĒRTĒJUMS	74
REZULTĀTI	77
SECINĀJUMI	78
IZMANTOTIE LITERATŪRAS AVOTI.....	79
PIELIKUMS	80

ATTĒLU SARAKSTS

1.1. att. 0.līmeņa DPD	10
1.2.att. Datu izmaiņu pārlūkošanas moduļa ekrānforma	11
1.3. att. Datu izmaiņu pārlūkošanas moduļa ekrānforma ar sistēmas paziņojumu.....	12
1.4. att. Datu izmaiņu pārlūkošanas moduļa ekrānforma ar grupēšanas tabulu un nolaižamo sarakstu saistītajiem sarakstiem.....	13
2.1. att. Dekompozīcijas entītiju savstarpējā pakļautība	21
2.2. att. 1.līmeņa DPD	25
2.3. att. Loģiskais ER modelis.....	26
2.4. att. Konceptuālais ER modelis.....	27
2.5. att. Datu izmaiņu pārlūkošanas moduļa ekrānformas projektējums.....	28
2.6. att. Datu izmaiņu pārlūkošanas moduļa ekrānformas ar grupēšanas tabulu un nolaižamo sarakstu saistītajiem sarakstiem projektējums	29
2.7. att. 2.līmeņa DPD	35
2.8. att. Procedūras get_data algoritma projektējuma blokshēma bu_trigger izmaiņu reģistrēšanas mehānisma gadījumā	38
2.9. att. Funkcijas generate projektējuma algoritma blokshēma	43
2.10. att. Procedūras LoadForm projektējuma algoritma blokshēma.....	45
2.11. att. Procedūras LoadDataChanges projektējuma algoritma blokshēma	50
3.1. att. Automatizētās vienībtestēšanas žurnāls automatizācijas platformā Jenkins	54
3.2. att. Automatizētās vienībtestēšanas žurnāla grafisks attēlojums automatizācijas platformā Jenkins	55
3.3. att. Automatizētās vienībtestēšanas kopsavilkums automatizācijas platformā Jenkins	55
3.4. att. Neizdevies vienībtests automatizācijas platformā Jenkins	56
6.1. att. Dokumenta versionēšana failu uzglabāšanas un apmaiņas vietnē OneDrive.....	72
6.2. att. Datu bāzes pakotnes saglabāšana versiju kontroles sistēmā Git	73
6.3. att. Datu bāzes pakotnes saglabāšana versiju kontroles sistēmā Git, izmantojot TortoiseGit rīku	73

IEVADS

Par sistēmu NINO

NINO ir uzņēmuma SIA “ZZ Dats” izstrādāta nekustamā īpašuma nodokļa administrēšanas sistēma, kas tiek izmantota visās Latvijas pašvaldībās. Sistēmas galvenās funkcijas ir:

- Informācijas uzskaitē un aktualizācija par nekustamā īpašuma nodokļa objektiem, fiziskajām un juridiskajām personām – nekustamā īpašuma nodokļa maksātājiem, kā arī citām ar nekustamo īpašumu saistītām personām;
- Nekustamā īpašuma nodokļa administrēšana – nodokļa aprēķins, maksāšanas paziņojumu sagatavošana, maksājumu identificēšana un reģistrēšana, darbības ar nekustamā īpašuma nodokļa parādiem u.c. darbību nodrošināšana;
- Dokumentu uzskaitē – maksāšanas paziņojumi, atgādinājumi, brīdinājumi u.c. dokumenti.

Lietojumprogramma tiešsaistē vai ar citu programmu starpniecību izmanto šādu ārēju valsts reģistru datus:

- Valsts adrešu reģistrs;
- Nekustamā īpašuma valsts kadastra informācijas sistēma;
- Iedzīvotāju reģistrs;
- Valsts vienotā datorizētā zemesgrāmata;
- Nodokļu informācijas sistēma;
- Uzņēmumu reģistra informācijas sistēma.

Kvalifikācijas darba mērķis

Kvalifikācijas darba mērķis ir izstrādāt risinājumu, kas atvieglotu NINO konsultantu darbu un nodrošinātu iespēju NINO lietotājiem pārlūkot sistēmā reģistrēto datu izmaiņu vēsturi, lai noskaidrotu kā, kad un kurš lietotājs ir mainījis datus. Lai sasniegtu mērķi, tiek izvirzīti sekojoši darba uzdevumi:

1. Izstrādāt programmatūras prasību specifikāciju;
2. Izstrādāt programmatūras projektējuma aprakstu;
3. Izstrādāt moduļa programmkodu;
4. Veikt vienībtestēšanu un funkcionālo testēšanu, aprakstot iegūtos rezultātus testēšanas dokumentācijā;
5. Aprakstīt projekta organizāciju, konfigurāciju pārvaldību, izmantotos kvalitātes nodrošināšanas principus;

6. Veikt programmatūras izstrādes darbietilpības novērtējumu.

Darba aktualitāte

Ne visām informācijas vienībām NINO ir nodrošināta datu izmaiņu vēstures pārlūkošana un pašreizējais risinājums ir izstrādāts sen, kā rezultātā tas vairs neatbilst prasībām. NINO lietotājiem – gan pašvaldību darbiniekiem, gan SIA “ZZ Dats” klientu konsultantiem un izstrādes komandai – datu izmaiņu pārlūkošanas modulis sniegtu iespēju ērtā un savām vajadzībām pielāgojamā veidā pārlūkot datu izmaiņu vēsturi.

Darba struktūra

Izstrādātais kvalifikācijas darbs sastāv no programmatūras prasību specifikācijas (turpmāk tekstā - PPS), programmatūras projektējuma apraksta (turpmāk tekstā - PPA), testēšanas dokumentācijas, pielikumiem un gatava programmaprodukta.

VĀRDNĪCA

Vārds	Skaidrojums
Attēlošana	Parādīšana uz ekrāna.
DPD	Datu plūsmas diagramma.
Datu izmaiņu vēstures tabula	Tabula datu izmaiņu pārlūkošanas formā, kurā tiek attēlota datu izmaiņu vēsture konkrētam objektam.
Grupēšanas tabula	Papildus tabula datu izmaiņu pārlūkošanas formā, pēc kuras ierakstiem tiek veikta datu izmaiņu vēstures grupēšana.
Ievaddati	Ieejas dati, kas nepieciešami funkcijas/procedūras darbībai.
Izmaiņu tabula	Tabula, kurā glabājas pamattabulas ierakstu vēsturiskie dati.
Izvaddati	Izejas dati, kas rodas pēc funkcijas/procedūras izpildītajām operācijām.
JSON	Datu apmaiņas formāts, kas paredzēts strukturētai datu pārraidei.
Konfigurācija	JSON formāta konfigurācijas fails.
Meklēšanas parametri	Datu lauku kopums, kurā ietilpst lietotāja atzīmētie datu lauki, kurus attēlot un datu lauki, kuru izmaiņas pārlūkot.
NINO	Nekustamā īpašuma nodokļa administrēšanas sistēma.
Objektu identificējoši dati	Dati, kas viennozīmīgi un lietotājam saprotamā tekstā identificē objektu.
Objekta tips	Tabulas nosaukums, kuram pieder objekts (piemēram, zemes vienības objekta tips ir “parceles”, jo zemes vienības objekti glabājas tabulā ar nosaukumu “parceles”).
Pakotne	Funkciju, procedūru, konstanšu kopums Oracle PL/SQL programmēšanas valodā.
Pamattabula	Tabula, kurā glabājas uz šo brīdi aktuālie objekta dati.
PPA	Programmatūras projektējuma apraksts.
PPS	Programmatūras prasību specifikācija.
Programmprodukts	Strādājošs programmkods kopā ar dokumentāciju.
Saistītais saraksts	Tabula, kas ir saistīta ar pamattabulu un kurai arī ir iespējams pārlūkot datu izmaiņu vēsturi.
SQL	(angļu val. “ <i>Structured Query Language</i> ”) Strukturēta vaicājumvaloda, ko izmanto datu bāzes pārvaldības sistēmās.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

PPS nolūks ir aprakstīt NINO datu izmaiņu pārlūkošanas moduļa funkcionālās un nefunkcionālās prasības, kā arī esošu sistēmas vienumu, kuros nepieciešams veikt izmaiņas, funkcionālās prasības. PPS paredzēta pasūtītājam moduļa prasību definēšanai un izstrādātājam precīzai prasību realizācijai sistēmā.

1.1.2. Darbības sfēra

Datu izmaiņu pārlūkošanas moduļa mērķis ir nodrošināt iespēju uzskatāmā, ērtā un individuāli pielāgojamā veidā pārlūkot dažādu NINO objektu datu lauku izmaiņu vēsturi. Modulis paredzēts visiem NINO lietotājiem, kuriem ir atbilstošās piekļuves tiesības. Izstrādājot šādu moduli, NINO objektu izmaiņas kļūst trasējamas, līdz ar to tiek paaugstināta nodokļu inspektoru darba efektivitāte, atvieglota darbinieku kontrole ar datu izmaiņu veikšanu saistītos jautājumos, kā arī samazināta nepieciešamība griezties pēc palīdzības pie NINO klientu konsultantiem ar datu izmaiņām saistītos jautājumos.

1.1.3. Definīcijas

Definīcijas un saīsinājumus skatīt nodaļā “Vārdnīca”.

1.1.4. Saistība ar citiem dokumentiem

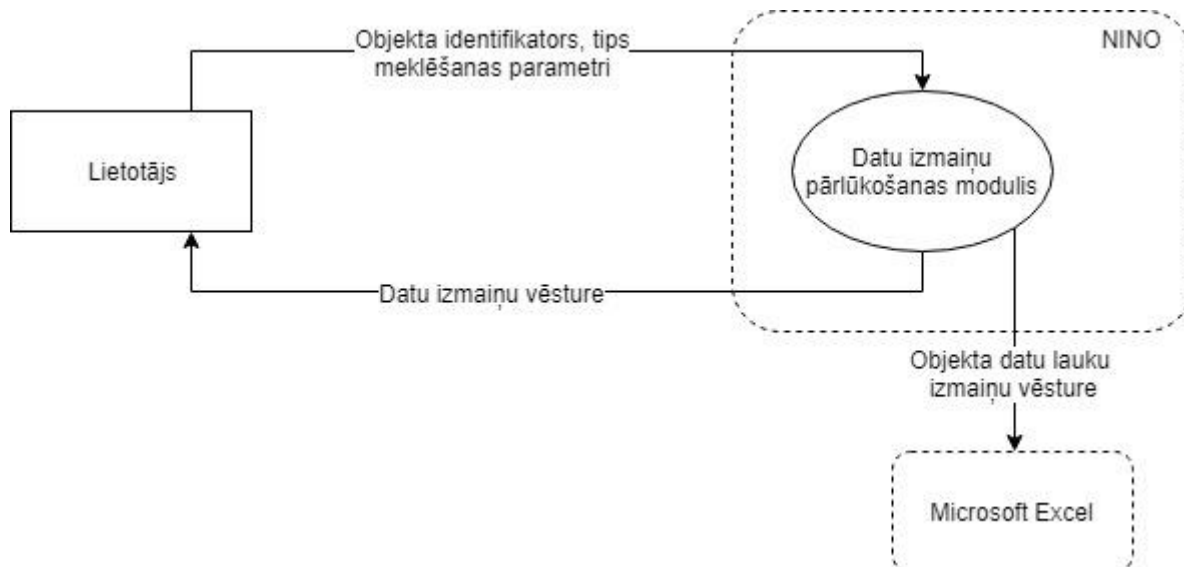
PPS izstrādāta, balstoties uz standarta LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis” [8] prasībām. PPS tiks izmantots dokumentu “Programmatūras projektējuma apraksts” (skat. nodaļu 2.) un “Testēšanas dokumentācija” (skat. nodaļu 3.) izstrādē.

1.2. Vispārējs apraksts

1.2.1. Produkta perspektīva

Datu izmaiņu pārlūkošanas modulis ir NINO sastāvdaļa un pilnas tā funkcionalitātes nodrošināšanai nepieciešama piekļuve lietotnei Microsoft Excel. Modulis paredzēts NINO

lietotājiem, lai pielāgojamā veidā apskatītu dažādu objektu datu lauku izmaiņu vēsturi (0.līmeņa DPD skat. att. 1.1). No iepriekšminētā izriet, ka sekmīgai datu izmaiņu pārlūkošanas moduļa darbībai lietotāja datorā nepieciešams uzstādīt NINO ar atbilstošām piekļuves tiesībām tajā, kā arī, lai nodrošinātu pilnu funkcionalitāti, uz datora jābūt uzstādītai izklājlapu lietotnei Microsoft Excel.



1.1. att. 0.līmeņa DPD

1.2.2. Produkta funkcijas

Lietotājs, kurš ir autentificējies sistēmā un kuram ir sniegtas nepieciešamās piekļuves tiesības, var veikt dažādu objektu (zemes, ēku, personu, darba procesu un uzdevumu, atvieglojumu) datu lauku izmaiņu attēlošanu pēc viņa izvēlētajiem meklēšanas parametriem.

1.2.3. Lietotāja raksturiezīmes

Lai darbotos ar datu izmaiņu pārlūkošanas moduli, lietotājam jāpiemīt prasmēm darbā ar NINO.

1.2.4. Vispārējie ierobežojumi

Datu izmaiņu pārlūkošanas modulim jāiekļaujas kopējā sistēmas arhitektūrā, un moduļa izstrādē jāņem vērā sistēmas kopējās saskarnes dizaina vadlīnijas, izmantojamās tehnoloģijas un kopējās nefunkcionālās prasības.

1.2.5. Pieņēmumi un atkarības

Pieņemts, ka uz lietotāja datora ir uzstādīta sistēma NINO un izklājlapu lietotne Microsoft Excel, kā arī sniegtas nepieciešamās piekļuves tiesības.

1.3. Funkcionālās prasības

1.3.1. Ārējās saskarnes prasības

1.3.1.1. Lietotāja saskarne

Lietotāja saskarnes nodrošināšanai izmantojami Microsoft Visual Basic 6 rīki. Datu izmaiņu pārlūkošanas modulim nepieciešams izveidot vienu jaunu formu (skat. att. 1.2.).

The application window displays the following components:

- Attēlojamās kolonnas (Columns to display):** A list of checkboxes for selecting columns. Red number 1 is next to 'Izmaiņu veicējs' (Change maker).
- Kādas izmaiņas attēlot (Which changes to display):** A list of checkboxes for selecting types of changes. Red number 2 is next to 'Personas kods/reģistrācijas numurs' (Person's code/registration number).
- Data Table:** A table with 4 columns: 'Adrese' (Address), 'Spēkā no' (Valid from), 'Spēkā līdz' (Valid until), and 'Izmaiņu veicējs' (Change maker). Red number 3 is next to the 'Spēkā līdz' column header. The table contains three rows of data.
- Buttons:** 'Atzīmēt visus' (Mark all), 'Noņemt visus' (Remove all), 'Attēlot' (Display), 'Drukāt' (Print), and 'Beigt' (End).

Adrese	Spēkā no	Spēkā līdz	Izmaiņu veicējs
Rīgas iela 1	01.01.2018 00:00:00		Anna Kalniņa
Lauku iela 3	01.12.2017 11:11:11	01.01.2018 00:00:00	Apmaiņa ar kadastru
Zemes iela 5	01.01.1990 22:22:22	01.12.2017 11:11:11	NINO

1.2.att. Datu izmaiņu pārlūkošanas moduļa ekrānforma

Turpmāk attēlā (skat. att. 1.2) redzamais saraksts Nr.1 dokumentā tiks saukts kā “Attēlojamo kolonnu saraksts”. Šajā sarakstā iespējams atzīmēt, kādus datu laukus jeb tabulas kolonnas attēlot datu izmaiņu vēstures tabulā uz ekrāna. Sarakstā Nr. 2 jeb “Izmaiņu kolonnu sarakstā” iespējams atzīmēt, kuru datu lauku izmaiņas pārlūkot. Abos sarakstos izvēles rūtiņās atzīmētie datu lauki – “Meklēšanas parametri”. Tabulā Nr.3 jeb “Datu izmaiņu vēstures tabulā” attēlotas objekta datu izmaiņas.

Attēlā (skat. att. 1.2) redzama datu izmaiņu pārlūkošanas forma, kurā NINO lietotājs izvēlēties pārlūkot kādas personas datu izmaiņu vēsturi. Meklēšanas parametri ir sekojoši: lietotājs tabulā vēlas attēlot adreses, izmaiņu veicēja un datumu kolonnas un pārlūkot, kā, kad un kas ir izmainījis personas adresi (attēlojamo un izmaiņu kolonnu sarakstos atzīmēti atbilstošie datu lauki). Pēc pogas “Attēlot” nospiešanas datu tabulā tiek parādīta datu izmaiņu vēsture, ņemot vērā ievadītos meklēšanas parametrus. Datu izmaiņu vēstures tabulas kolonnu “Spēkā no” un “Spēkā līdz” informācija attiecināma uz datu laukiem, kas atzīmēti izmaiņu kolonnu sarakstā.

1.3. att. Datu izmaiņu pārlūkošanas moduļa ekrānforma ar sistēmas paziņojumu

Ja NINO lietotājs ir nospiedis pogu “Attēlot”, taču nav atzīmējis nevienu datu lauku izmaiņu kolonnu sarakstā, tiek parādīts kļūdas paziņojums (att. 1.3).

Datu izmaiņu pārlūks - atvieglojumu piešķiršana objektiem - Jānis Bērziņš (123456-54321) - Personas ar 3 un vairāk bērniem-90%

Attēlojamās kolonnas

☒ Izmaiņu veicējs

☒ Piešķirt

☒ Spēkā no

Atzīmēt visus Noņemt visus

Kādas izmaiņas attēlot

☒ Piešķirt

Atzīmēt visus Noņemt visus

Saistītie saraksti

1. Attēlot

Veids	Piešķirt	Nosaukums	Kad.	Konts	Plat.kopā	Plat.atviegl.
Z	Jā	Zemes iela	1234-5678-	123456	1000	1000
Ē	Nē	Zemes iela	1234-5678-	123456		

2.

Piešķirt	Spēkā no	Spēkā līdz	Izmaiņu veicējs
Nē	01.11.2018 00:00:00		Anna Kalniņa
Jā	01.02.2018 11:11:11	01.11.2018 00:00:00	Apmaiņa ar kadastru
Nē	01.01.2018 22:22:22	01.02.2018 11:11:11	NINO

Attēlot Drukāt Beigt

1.4. att. Datu izmaiņu pārlūkošanas moduļa ekrānforma ar grupēšanas tabulu un nolaižamo sarakstu saistītajiem sarakstiem

Ja tabulai eksistē saistītie saraksti, tad datu izmaiņu pārlūkošanas forma tiek papildināta ar nolaižamo sarakstu (skat. att. 1.4 - lauks Nr. 1.), kā arī, ja datu izmaiņu attēlošanai nepieciešama grupēšanas tabula, tā tiek attēlota virs datu izmaiņu vēstures tabulas (skat. att. 1.4 - lauks Nr. 2.) un ieraksti datu izmaiņu vēstures tabulā veidojas, balstoties uz grupēšanas tabulā atlasīto ierakstu.

1.3.1.2. Programmatūras saskarne

Moduļa izstrādei nepieciešams izmantot Oracle Database 12c datu bāzi, kā arī Oracle PL/SQL Developer 12 un Microsoft Visual Basic 6 izstrādes vides.

1.3.2. Sistēmas paziņojumi

Datu izmaiņu pārlūkošanas moduļa sistēmas paziņojumus skatīt tabulā 1.1.

1.1. tabula

Datu izmaiņu pārlūkošanas moduļa sistēmas paziņojumi


Identifikators	Paziņojuma teksts
P1	Atzīmējiet izmaiņu laukus!

1.3.3. Formas izsaušanas modulis (FIM)

Formas izsaušanas moduļa funkcijas aprakstītas tabulās 1.2. un 1.3.

1.2. tabula

Formas atvēršana (FIM01)

Mērķis	
Atvērt datu izmaiņu pārlūkošanas formu ar visām nepieciešamajām formā ietilpstošajām komponentēm.	
Funkcijas izsaukums	
<u>Veids</u>	<u>Nosaukums</u>
Poga	Datu izmaiņu pārlūkošanas moduļa izsaušanas poga () saraksta formā
Poga	“Attēlot” (pie saistīto sarakstu nolaižamā saraksta)
Apstrāde	
<p>Notiek pārbaude, vai forma jau ir atvērta konkrētajam objektam:</p> <ul style="list-style-type: none">• Jā - forma tiek novietota priekšpusē virs visām pārējām dotajā brīdī atvērtajām formām;• Nē - forma tiek atvērta. <p>Jau esošajam formas nosaukumam “Datu izmaiņu pārlūks” tiek pievienoti lietotājam atpazīstami objektu identificējoši dati (katram objekta tipam atbilstošos identificējošos datus skat. 2.pielikumā).</p> <p>Ja nepieciešams, formā tiek attēlots nolaižamais saraksts ar objekta tipam atbilstošajiem saistītajiem sarakstiem un/vai grupēšanas tabula (abu komponentu nepieciešamību un detaļas skat. 2.pielikumā).</p> <p>Izsauc funkcijas MPM01 un MPM07, lai attēlotu datu laukus un pēdējos meklēšanas parametrus.</p> <p>Izsauc funkciju DAM01, lai attēlotu datu izmaiņu vēsturi.</p>	

1.3. tabula

Formas aizvēršana (FIM02)

Mērķis	
Aizvērt datu izmaiņu pārlūkošanas formu.	
Funkcijas izsaukums	
<u>Veids</u>	<u>Nosaukums</u>

Poga	“Beigt”
Apstrāde	
Tiek aizvērtā datu izmaiņu pārlūkošanas forma.	

1.3.4. Meklēšanas parametru modulis (MPM)

Meklēšanas parametru moduļa funkcijas aprakstītas tabulās 1.4., 1.5., 1.6., 1.7., 1.8., 1.9., 1.10.

1.4. tabula

Datu lauku attēlošana (MPM01)

Mērķis	
Attēlot lietotājam divus sarakstus - attēlojamo un izmaiņu kolonnu sarakstu. Katram objekta tipam atbilstošos attēlojamās datu laukus un iespēju pārlūkot to izmaiņas skatīt 1. pielikumā.	
Funkcijas izsaukums	
<u>Veids</u>	<u>Nosaukums</u>
Funkcija	FIM01
Ievaddati	
<u>Lauka nosaukums</u>	<u>Obligāts</u>
Objekta tips	Jā
Konkrēts objekta parametrs	Jā
Apstrāde	
No datu bāzes tiek iegūti nepieciešamie datu lauki un parādīti attēlojamo un izmaiņu kolonnu sarakstos.	
Izvaddati	
<ul style="list-style-type: none"> Objekta tipam atbilstoši attēlojamie datu lauki; Datu lauki, kuru izmaiņas konkrētajam objekta tipam iespējams pārlūkot. 	

1.5. tabula

Datu lauku atzīmēšana (MPM02)

Mērķis
Ļaut lietotājam ar izvēles rūtiņas palīdzību atzīmēt datu laukus no attēlojamo un/vai izmaiņu kolonnu sarakstiem.
Funkcijas izsaukums

<u>Veids</u>	<u>Nosaukums</u>
Izvēles rūtiņa	Datu lauka nosaukums
Ievaddati	
<u>Lauka nosaukums</u>	<u>Obligāts</u>
Datu lauks	Jā
Apstrāde	
Izvēles rūtiņa tiek atzīmēta.	

1.6. tabula

Visu datu lauku atzīmēšana attēlojamo kolonnu sarakstā (MPM03)

Mērķis	
Ļaut lietotājam vienlaicīgi atzīmēt visas izvēles rūtiņas attēlojamo kolonnu sarakstā.	
Funkcijas izsaukums	
<u>Veids</u>	<u>Nosaukums</u>
Poga	“Atzīmēt visus”
Apstrāde	
Tiek atzīmētas visas izvēles rūtiņas attēlojamo kolonnu sarakstā.	

1.7. tabula

Visu atzīmju noņemšana no datu laukiem attēlojamo kolonnu sarakstā (MPM04)

Mērķis	
Ļaut lietotājam vienlaicīgi noņemt visas atzīmes no izvēles rūtiņām attēlojamo kolonnu sarakstā, saglabājot atzīmes pie obligātajiem datu laukiem.	
Funkcijas izsaukums	
<u>Veids</u>	<u>Nosaukums</u>
Poga	“Noņemt visus”
Apstrāde	
Tiek noņemtas atzīmes no visiem datu laukiem attēlojamo kolonnu sarakstā, izņemot obligātajiem datu laukiem (objekta tipam atbilstošos obligātos datu laukus skatīt 1.pielikumā) un tiem, kuri ir atzīmēti izmaiņu kolonnu sarakstā.	

1.8. tabula

Visu datu lauku atzīmēšana izmaiņu kolonnu sarakstā (MPM05)

Mērķis

Ļaut lietotājam vienlaicīgi atzīmēt visas izvēles rūtiņas izmaiņu kolonnu sarakstā.
Funkcijas izsaukums: MPM03
Ievaddati: MPM03
Apstrāde
Tiek atzīmētas visas izvēles rūtiņas izmaiņu kolonnu sarakstā.

1.9. tabula

Visu atzīmju noņemšana no datu laukiem izmaiņu kolonnu sarakstā (MPM06)

Mērķis	
Ļaut lietotājam vienlaicīgi noņemt visas atzīmes no izvēles rūtiņām izmaiņu kolonnu sarakstā.	
Funkcijas izsaukums	
<u>Veids</u>	<u>Nosaukums</u>
Poga	“Noņemt visus”
Apstrāde	
Tiek noņemtas atzīmes no visiem datu laukiem izmaiņu kolonnu sarakstā.	

1.10. tabula

Pēdējo lietotāja meklēšanas parametru attēlošana (MPM07)

Mērķis	
Izvēles rūtiņās atzīmēt tos pašus meklēšanas parametrus, kas bijuši atzīmēti pēdējā šī tipa objekta datu izmaiņu vēstures attēlošanas reizē.	
Funkcijas izsaukums	
<u>Veids</u>	<u>Nosaukums</u>
Funkcija	FIM01
Apstrāde	
No datu bāzes tiek iegūti pēdējie konkrētā lietotāja meklēšanas parametri un atzīmētas attiecīgās izvēles rūtiņas blakus parametru nosaukumiem.	

1.3.5. Datu atlasē modulis (DAM)

Datu atlasē moduļa funkcijas aprakstītas tabulās 1.11., 1.12.

1.11. tabula

Datu izmaiņu vēstures attēlošana uz ekrāna (DAM01)

Mērķis	
Tabulā attēlot datu izmaiņu vēsturi par objektu, ņemot vērā lietotāja ievadītos meklēšanas parametrus.	
Funkcijas izsaukums	
<u>Veids</u>	<u>Nosaukums</u>
Poga	“Attēlot” (pie datu izmaiņu vēstures tabulas)
Funkcija	FIM01
Tabulas ieraksta atlasīšana	Grupēšanas tabulas ieraksta atlasīšana
Ievaddati	
<u>Lauka nosaukums</u>	<u>Obligāts</u>
Meklēšanas parametri	Jā
Apstrāde	
Pārbauda, vai izmaiņu kolonnu sarakstā ir atzīmēts vismaz viens datu lauks: <ul style="list-style-type: none">• Nē – tiek parādīts P1;• Jā – ņemot vērā meklēšanas parametrus, atlasa nepieciešamos datus no datu bāzes, saglabā lietotāja meklēšanas parametrus datu bāzē.	
Izvaddati	
Datu izmaiņu vēsture	

1.12. tabula

Datu izmaiņu vēstures attēlošana lietotnē Microsoft Excel (DAM02)

Mērķis	
Attēlot datu izmaiņu vēsturi par konkrēto objektu lietotnē Microsoft Excel, pievienojot informāciju par meklēšanas parametriem un objekta identificēšanai nepieciešamo informāciju.	
Funkcijas izsaukums	
<u>Veids</u>	<u>Nosaukums</u>
Poga	“Drukāt”
Ievaddati	

<u>Lauka nosaukums</u>	<u>Obligāts</u>
Datu izmaiņu vēsture	Jā
Apstrāde	
Tiek atvērta lietotne Microsoft Excel un uz to no datu izmaiņu vēstures tabulas tiek eksportēti dati, kā arī pievienoti meklēšanas parametri un objektu identificējoši dati (katram objekta tipam atbilstošos identificējošos datu laukus skatīt 2. pielikumā).	
Izvaddati	
Lietotnē Microsoft Excel attēlota datu izmaiņu vēsture.	

1.4. Nefunkcionālās prasības

Tā kā datu izmaiņu pārlūkošanas modulis ir NINO sastāvdaļa, tā izstrādes laikā jāņem vērā kopējās NINO nefunkcionālās prasības [4], no kurām uz datu izmaiņu pārlūkošanas moduli tieši attiecas zemāk uzskaitītās.

Veiktspējas prasības:


- Lietotāju skaits ir neierobežots;
- Pieslēgumu skaits un ātrdarbība ir atkarīga no tīkla un servera veiktspējas;
- Tabulu ierakstu skaits datu bāzē var pārsniegt vairākus miljonus ierakstu.

Drošības prasības:

- Izstrādātie papildinājumi ir jāuzstāda uz RDVIS infrastruktūras serveriem. Izstrādāto papildinājumu uzstādīšana un darbināšana nedrīkst radīt papildu riskus RDVIS IT infrastruktūras fiziskajai un loģiskajai aizsardzībai;
- Tiek saglabāta datu integritāte;
- Neviena no risinājuma moduļiem un datu pārsūtīšanas posmiem nedrīkst vājināt sistēmas drošību.

1.5. Izmaiņas NINO vienumos

NINO esošajā vienumā – saraksta formā – nepieciešams mainīt esošās pogas

() funkcijas izsaukumu uz datu izmaiņu pārlūkošanas moduļa formas atvēršanas funkcijas FIM01 izsaukumu. Darba procesu/uzdevumu un atvieglojumu reģistra sarakstu formā šī poga ir atspējota, tāpēc to nepieciešams iespējot.

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Nolūks

PPA izveidots, lai aprakstītu NINO datu izmaiņu pārlūkošanas moduļa projektējumu, balstoties uz PPS aprakstītajām prasībām. PPA paredzēts moduļa izstrādātājiem.

2.1.2. Darbības sfēra

Datu izmaiņu pārlūkošanas moduļa mērķis ir nodrošināt iespēju uzskatāmā, ērtā un individuāli pielāgojamā veidā pārlūkot dažādu NINO objektu datu lauku izmaiņu vēsturi. Modulis paredzēts visiem NINO lietotājiem, kuriem ir atbilstošās piekļuves tiesības. Izstrādājot šādu moduli, NINO objektu izmaiņas kļūst trasējamas, līdz ar to tiek paaugstināta nodokļu inspektoru darba efektivitāte, atvieglota darbinieku kontrole ar datu izmaiņu veikšanu saistītos jautājumos, kā arī samazināta nepieciešamība griezties pēc palīdzības pie NINO klientu konsultantiem ar datu izmaiņām saistītos jautājumos.

2.1.3. Definīcijas

Definīcijas un saīsinājumus skatīt nodaļā “Vārdnīca”.

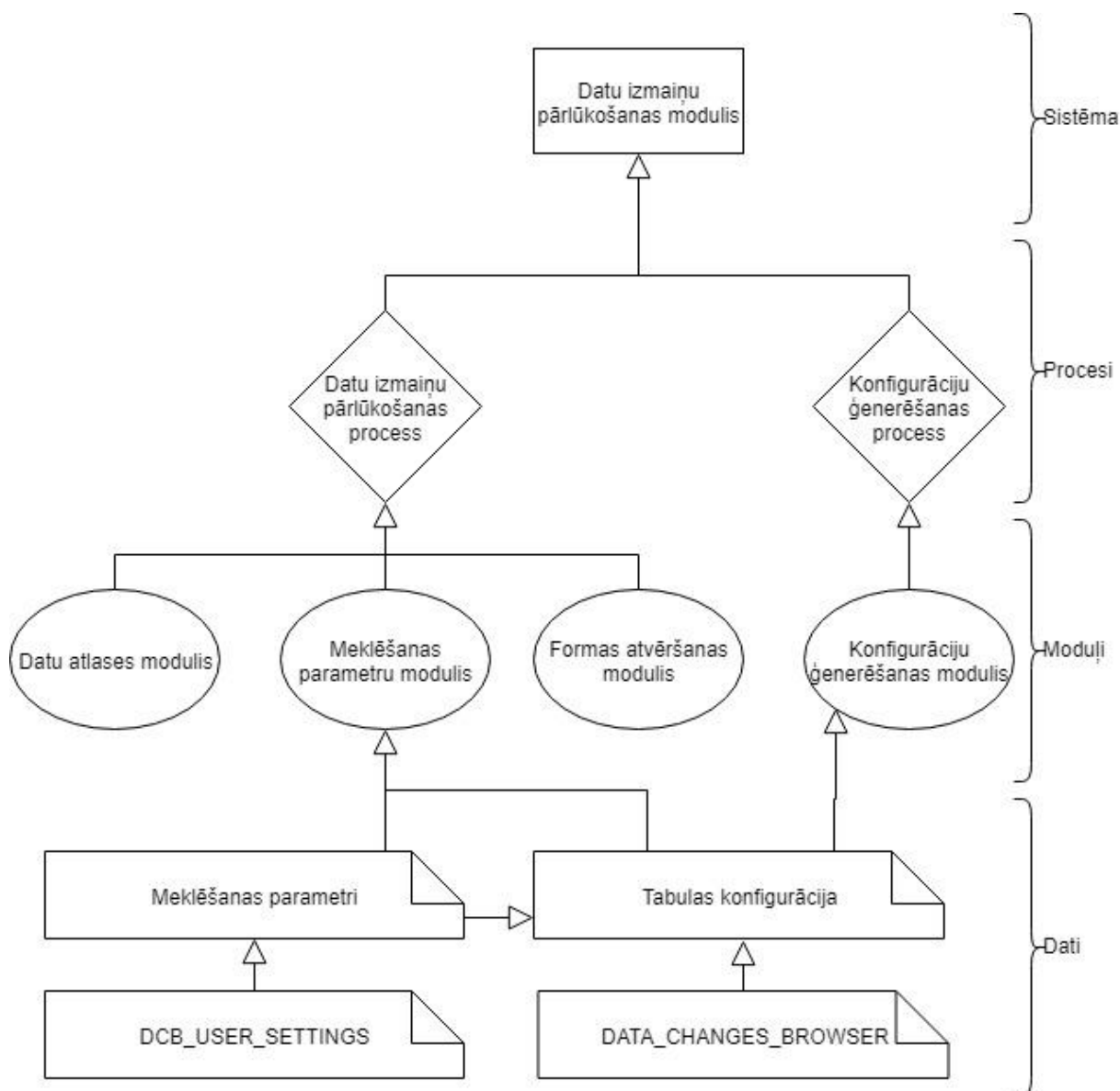
2.2. Saistība ar citiem dokumentiem.

PPA tika izstrādāts, balstoties uz dokumenta “Programmatūras prasību specifikācija” (skat. nodaļu 1.) prasībām un ievērojot standartu LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai” [9]. PPA tiks izmantots dokumenta “Testēšanas dokumentācija” (skat. nodaļu 3.) izstrādē.

2.3. Dekompozīcijas apraksts

Datu izmaiņu pārlūkošanas moduļa darbība balstīta uz procesiem, funkciju moduļiem un datu struktūrām, kur visi šie vienumi savā starpā ir saistīti. Attēlā (skat. att. 2.1.) redzama dekompozīcijas entītiņu savstarpējā pakļautība (entītiņu aprakstus skat. nodaļā 2.3.1.). Datu izmaiņu pārlūkošanas modulis sastāv no diviem galvenajiem procesiem, kas var tikt izmantoti

vienlaicīgi, šie procesi sastāv no vairākiem moduļiem. Konfigurāciju ģenerēšanas modulī tiek ģenerētas tabulu konfigurācijas, kas derīgas saglabāšanai tabulā DATA_CHANGES_BROWSER. Savukārt meklēšanas parametri tiek veidoti, balstoties uz lietotāja ievaddatiem un atbilst tabulu konfigurāciju struktūrai. Tie tiek saglabāti tabulā DCB_USER_SETTINGS. Meklēšanas parametrus un tabulu konfigurācijas izmanto meklēšanas parametru modulis. Datu atlasē moduļa funkcijas iespējams izsaukt pēc meklēšanas parametru moduļa funkciju izpildes, savukārt meklēšanas parametru moduļa darbībai nepieciešams izsaukt formas atvēršanas moduļa funkcijas.



2.1. att. Dekompozīcijas entitiju savstarpējā pakļautība

2.3.1. Moduļu dekompozīcija

Moduļu dekompozīcijas aprakstu skat. tabulā 2.1.

2.1. tabula

Moduļu dekompozīcijas apraksts

Modulis	Nolūks	Funkcijas
Formas atvēršanas modulis	Atvērt un aizvērt datu izmaiņu pārlūkošanas formu.	Formas atvēršana konkrētam objektam, formas aizvēršana, pēc nepieciešamības grupēšanas tabulas vai nolaižamā saraksta saistītajiem sarakstiem parādīšana.
Meklēšanas parametru modulis	Ļaut lietotājam atzīmēt tikai tos datu laukus un tikai tās datu lauku izmaiņas, kas viņam dotajā brīdī nepieciešamas.	Attēlojamo un izmaiņu kolonnu sarakstos attēlot objekta tipam atbilstošos datu laukus (skat. 1.pielikumā), ļaut lietotājam atzīmēt viņam nepieciešamos datu laukus, atzīmēt vai noņemt visas atzīmes vienlaicīgi, saglabāt atzīmētos meklēšanas parametrus un parādīt pēdējos saglabātos parametrus pie formas atvēršanas.
Datu atlasē modulis	Parādīt datu izmaiņu vēsturi tabulā uz ekrāna vai lietotnē Microsoft Excel.	Datu izmaiņu vēstures attēlošana tabulā uz ekrāna, datu izmaiņu vēstures attēlošana lietotnē Microsoft Excel.
Konfigurāciju ģenerēšanas modulis	Ģenerēt tabulas konfigurāciju datu izmaiņu pārlūkošanas moduļim.	Sasaistīt pamattabulas un izmaiņu tabulas kolonnu nosaukumus, pēc iespējas identificēt izmaiņu datumu un lietotāja kolonnas, pēc noteiktas struktūras veidot tabulas konfigurāciju JSON formātā (skat. nodaļā 2.6.1.3).

2.3.2. Procesu dekompozīcija

Procesi datu izmaiņu pārlūkošanas modulī var tikt izmantoti un ir spējīgi darboties vienlaicīgi. Datu izmaiņu vēstures atlasīšanas procesu paredzēts izmantot visiem NINO lietotājiem un izstrādātājiem, bet konfigurāciju ģenerēšanas procesu, kas pieejams no Oracle PL/SQL Developer vides, - NINO izstrādātājiem.

2.3.2.1. Datu izmaiņu pārlūkošanas process

Tips: process

Nolūks: nodrošināt datu izmaiņu pārlūkošanu

Funkcijas: datu izmaiņu pārlūkošanas formas atvēršana, meklēšanas parametru iestatīšana, datu izmaiņu vēstures atlasīšana.

2.3.2.2. Konfigurāciju ģenerēšanas process

Tips: process

Nolūks: ģenerēt tabulas konfigurāciju

Funkcijas: skat. tabulā 2.1. tabula

2.3.3. Datu dekompozīcija

Datu izmaiņu pārlūkošanas moduļa funkcionalitātes nodrošināšanai ne tikai jāveido jaunas datu struktūras (skat. nodaļas 2.3.3.1., 2.3.3.2., 2.3.3.3., 2.3.3.4.), bet arī jāizmanto daļa jau esošo NINO datu bāzes tabulu – pamattabulas USER_TASKS, PERSONAS_NOD, PARCELES, ATV_REG, ATV_REG_PIEM, ATV_REG_OBJ, izmaiņu tabulas BU_USER_TASKS, BU_PERSONAS_NOD, BU_PARCELES, BU_ATV_REG, BU_ATV_REG_OBJ un klasifikatoru tabulas LIETOTAJI, USER_TASK_STATUS_CL, PERSSTATKL, ATVIEGLOJUMI, ATV_REG_PIEM_KL. Pamattabulas ir objektu tabulas ar nosaukumu bez prefiksa, bet izmaiņu tabulas – ar prefiksu “BU_”. Katru reizi, kad pamattabulā tiek izmainīts ieraksts, nostrādā triggeris (angļu val. “*trigger*”), kas izveido jaunu ierakstu izmaiņu tabulā un saglabā tur ieraksta datus tādus, kādi tie bijuši pirms izmaiņu veikšanas.

2.3.3.1. Tabulas konfigurācija

Tips: JSON konfigurācijas fails

Nolūks: pēc noteiktas struktūras (skat. nodaļu 2.6.1.3.) glabāt informāciju par pamattabulas un tai atbilstošās izmaiņu tabulas kolonnām.

2.3.3.2. Meklēšanas parametri

Tips: JSON konfigurācijas fails

Nolūks: pēc noteiktas struktūras glabāt informāciju par pamattabulas un tai atbilstošās izmaiņu tabulas datu laukiem, kurus lietotājs atzīmējis attēlojamo un izmaiņu kolonnu sarakstos.

2.3.3.3. Tabula DATA_CHANGES_BROWSER

Tips: tabula

Nolūks: glabāt pamattabulas un tai atbilstošās izmaiņu tabulas noteiktas struktūras konfigurāciju.

2.3.3.4. Tabula DCB_USER_SETTINGS

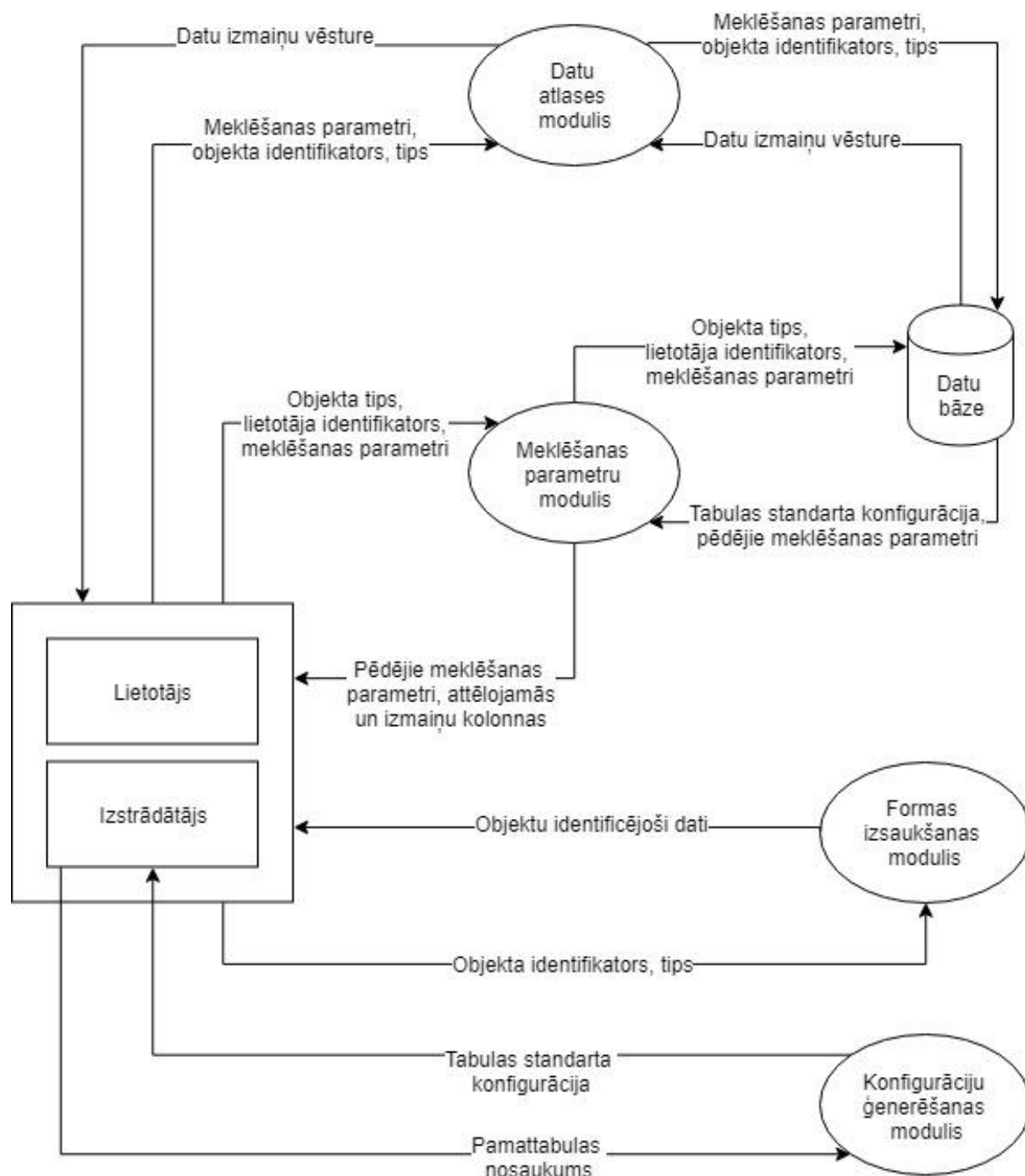
Tips: tabula

Nolūks: pēc noteiktas struktūras glabāt lietotāja pēdējos saglabātos meklēšanas parametrus.

2.4. Atkarības apraksts

2.4.1. Starpmoduļu atkarības.

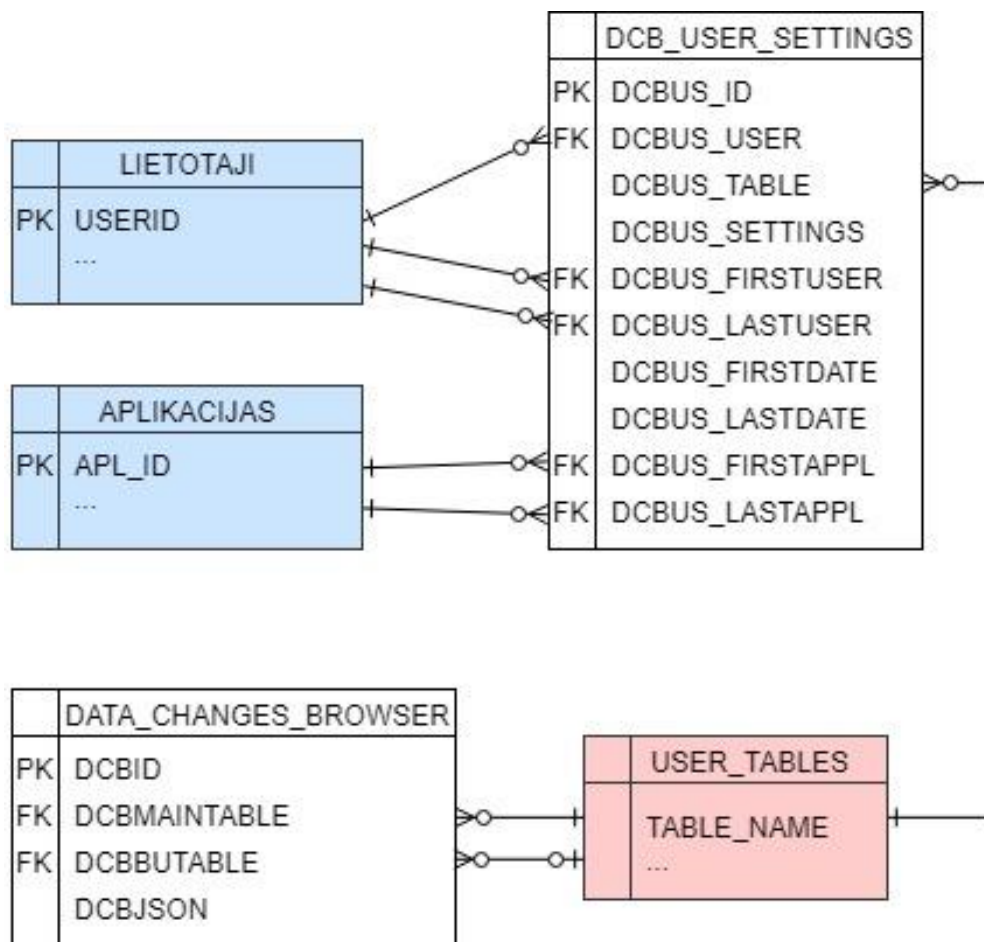
Moduļus savā starpā saista datu plūsmas, kas sākotnēji tiek veidotas lietotāja darbības rezultātā, bet pēc tam pārveidotas un izsūtītas gan no moduļa uz moduli, gan no moduļa uz datu bāzi un otrādi, kā arī atpakaļ uz lietotāju. Starpmoduļu atkarības attēlotas 1. līmeņa DPD (skat att. 2.2.).



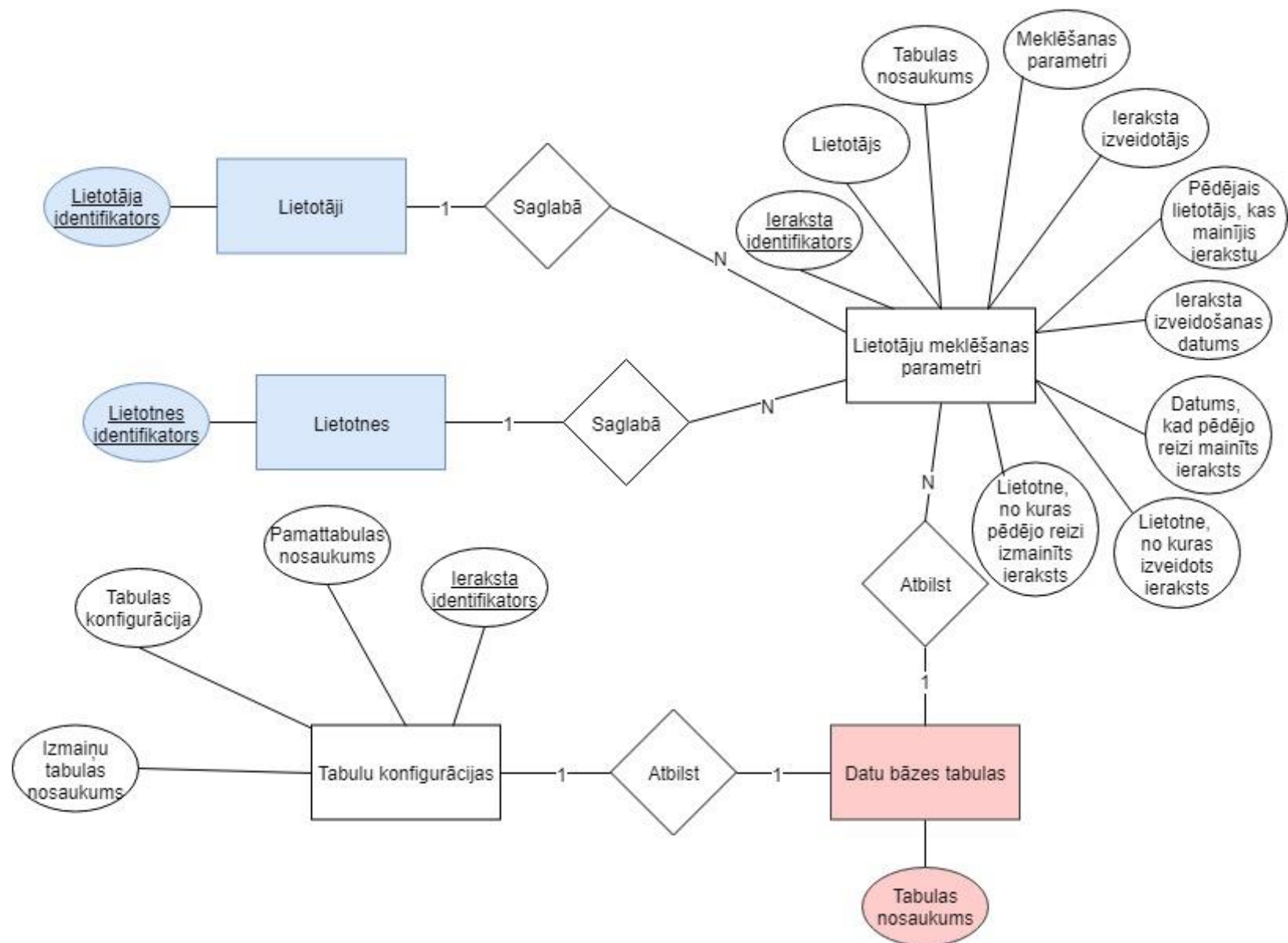
2.2. att. 1.līmeņa DPD

2.4.2. Datu atkarības

Datu atkarības skatīt loģiskajā ER modelī (skat. att. 2.3.) un konceptuālajā ER modelī (skat. att. 2.4.). ER modeļos attēlotas tabulas, kas jāveido no jauna (attēlā baltā krāsā), kā arī tabulas (attēlā zilā krāsā) un skati (attēlā rozā krāsā), kas jau pastāv NINO sistēmā un ir saistīti ar izveidojamajām tabulām.



2.3. att. Loģiskais ER modelis



2.4. att. Konceptuālais ER modelis

2.5. Saskarnes apraksts.

Attēlā (skat att. 2.5.) redzams datu izmaiņu pārlūkošanas moduļa lietotāja saskarnes projektējums. Laukos Nr. 1. (turpmāk tekstā- attēlojamo kolonnu sarakstā) un 2. (turpmāk tekstā - izmaiņu kolonnu sarakstā) redzami datu lauki no tabulas konfigurācijas (datu laukus, kā arī atzīmes tajos atgriež meklēšanas parametru modulis). Tabula Nr. 3 jeb datu izmaiņu vēstures tabula iegūta no datu atlases moduļa, nospiežot pogu Nr. 4 vai izsaucot formas atvēršanas moduļa funkcijas (jeb atverot datu izmaiņu pārlūkošanas formu). Pogas Nr. 5., 6., 7., 8. izsauc meklēšanas parametru moduļa funkcijas, bet poga Nr. 9. izsauc datu atlases moduļa funkciju. Formas izsaukšanas modulis nodrošina formas atvēršanu, pogas Nr. 10. funkcionalitāti un formas nosaukuma maiņu.

Datu izmaiņu pārlūks - personas/uzņēmuma vēsture - Jānis Bērziņš (123456-54321)

Attēlojamās kolonnas

- ☒ Adrese
- ☐ Citi vārdi
- ☒ Izmaiņu veicējs
- ☐ Miršanas datums
- ☐ Personas kods/reģistrācijas numurs
- ☐ Pilsēta, pasta indekss
- ☒ Spēkā no
- ☒ Spēkā līdz

5. Atzīmēt visus 6. Noņemt visus

Kādas izmaiņas attēlot

- ☒ Adrese
- ☐ Citi vārdi
- ☐ Miršanas datums
- ☐ Personas kods/reģistrācijas numurs
- ☐ Piezīmes
- ☐ Pilsēta, pasta indekss
- ☐ Tālrunis
- ☐ Uzvārds/Nosaukums

7. Atzīmēt visus 8. Noņemt visus

Adrese	Spēkā no	Spēkā līdz	Izmaiņu veicējs
Rīgas iela 1	01.01.2018 00:00:00		Anna Kalniņa
Lauku iela 3	01.12.2017 11:11:11	01.01.2018 00:00:00	Apmaiņa ar kadastru
Zemes iela 5	01.01.1990 22:22:22	01.12.2017 11:11:11	NINO

4. Attēlot 9. Drukāt 10. Beigt

2.5. att. Datu izmaiņu pārlūkošanas moduļa ekrānformas projektējums

Attēlā (skat. att. 2.6.) redzma datu izmaiņu pārlūkošanas forma, kurai pievienotas papildus komponentes – grupēšanas tabula (Nr. 1), kuras aizpildīšana tiek veikta datu atlases modulī un nolaižamais saraksts saistītajiem sarakstiem (Nr. 2), kura parādīšana tiek noteikta formas izsaukšanas modulī.

Datu izmaiņu pārļūks - atvieglojumu piešķiršana objektiem - Jānis Bērziņš (123456-54321) - Personas ar 3 un vairāk bērniem-90%

Attēlojamās kolonnas

☒ Izmaiņu veicējs

☒ Piešķirt

☒ Spēkā no

Atzīmēt visus Noņemt visus

Kādas izmaiņas attēlot

☒ Piešķirt

Atzīmēt visus Noņemt visus

Saistītie saraksti

2. Attēlot

Veids	Piešķirt	Nosaukums	Kad.	Konts	Plat.kopā	Plat.atviegl.
Z	Jā	Zemes iela	1234-5678-	123456	1000	1000
Ē	Nē	Zemes iela	1234-5678-	123456		

1.

Piešķirt	Spēkā no	Spēkā līdz	Izmaiņu veicējs
Nē	01.11.2018 00:00:00		Anna Kalniņa
Jā	01.02.2018 11:11:11	01.11.2018 00:00:00	Apmaiņa ar kadastru
Nē	01.01.2018 22:22:22	01.02.2018 11:11:11	NINO

Attēlot Drukāt Beigt

2.6. att. Datu izmaiņu pārļūkošanas moduļa ekrānformas ar grupēšanas tabulu un nolaižamo sarakstu saistītajiem sarakstiem projektējums

2.6. Detalizētais projektējums

Detalizētajā projektējumā aprakstīts datu izmaiņu pārļūkošanas moduļa datu struktūru projektējums, kā arī detalizēti funkciju apraksti, zema līmeņa algoritmu projektējumi un shēmas.

2.6.1. Datu detalizētais projektējums

2.6.1.1. Tabula DATA_CHANGES_BROWSER

Tabula DATA_CHANGES_BROWSER paredzēta, lai glabātu pamattabulas un tai atbilstošās izmaiņu tabulas nosaukumu un noteiktas struktūras tabulas konfigurāciju JSON formātā (tabulas aprakstu skat. tabulā 2.2.). Identifikatora un ārējo atslēgu lauki tiek indeksēti ātrākai datu atlases vaicājumu izpildīšanai, kā arī DCBJSON laukam piemērots ierobežojums, kas nodrošina derīga JSON tipa objekta ievietošanu tabulā. Tabulas identifikatora kolonnā DCBID tiek ievietotas secības (angļu val. “sequence”) ar nosaukumu DATA_CHANGES_BROWSER_SQ vērtības.

Kolonnas nosaukums	Datu tips	Apraksts
DCBID	NUMBER	PK, ieraksta identifikators.
DCBMAINTABLE	VARCHAR2(30)	Pamattabulas nosaukums.
DCBBUTABLE	VARCHAR2(30)	Pamattabulai atbilstošās izmaiņu tabulas nosaukums.
DCBJSON	CLOB	Tabulas konfigurācijas fails JSON formātā (konfigurācijas struktūru skatīt nodaļā 2.6.1.3.).

2.6.1.2. Tabula DCB_USER_SETTINGS

Tabula paredzēta, lai saglabātu tajā pēdējos lietotāja saglabātos meklēšanas parametrus jeb konfigurāciju, kas tiek glabāta JSON formātā ar tādu pašu struktūru kā tabulas DATA_CHANGES_BROWSER laukā DCBJSON (tabulas aprakstu skat. tabulā 2.3.). Identifikatora un ārējo atslēgu lauki tiek indeksēti ātrākai datu atlases vaicājumu izpildīšanai, kā arī DCBUS_SETTINGS laukam piemērots ierobežojums, kas nodrošina derīga JSON tipa objekta ievietošanu tabulā. Tabulas identifikatora kolonnā DCBUS_ID tiek ievietotas secības (angļu val. “*sequence*”) ar nosaukumu DCB_USER_SETTINGS_SEQ vērtības.

Kolonnas nosaukums	Datu tips	Apraksts
DCBUS_ID	NUMBER(10)	PK, ieraksta identifikators.
DCBUS_USER	NUMBER(10)	FK, lietotājs, uz kuru attiecas meklēšanas parametri.
DCBUS_TABLE	VARCHAR2(30)	FK, tabula jeb objekta tips, uz kuru attiecas meklēšanas parametri.
DCBUS_SETTINGS	CLOB	Meklēšanas parametri.
DCBUS_FIRSTUSER	NUMBER(10)	FK, ieraksta izveidotāja identifikators.
DCBUS_LASTUSER	NUMBER(10)	FK, lietotāja identifikators, kas pēdējais izmainījis ierakstu.
DCBUS_FIRSTDATE	DATE	Ieraksta izveidošanas datums.
DCBUS_LASTDATE	DATE	Pēdējo izmaiņu datums.
DCBUS_FIRSTAPPL	NUMBER(10)	FK, lietotnes identifikators, no kuras izveidots ieraksts.

Kolonnas nosaukums	Datu tips	Apraksts
DCBUS_LASTAPPL	NUMBER(10)	FK, lietotnes identifikators, no kuras pēdējo reizi izmainīts ieraksts.

2.6.1.3. Tabulu konfigurācijas

Balstoties uz tabulu konfigurācijām JSON formātā, dinamiski tiek veidots datu atlases vaicājums, kā arī nodrošināta datu izmaiņu pārlūkošanas formā attēlojamā saturiskā informācija. Ar konfigurāciju palīdzību tiek glabāti arī lietotāja pēdējie meklēšanas parametri. Konfigurācijas paraugu skatīt 3.pielikumā.

Katra konfigurācija sastāv no galvenā objekta, kas satur vispārējus objektu raksturojošus parametrus, kas nepieciešami gan datu atlases vaicājuma veidošanai, gan datu izmaiņu pārlūkošanas formas attēlošanai (konfigurācijas struktūru skat. tabulā 2.4.).

2.4. tabula

Objekta sastāvdaļas	Apraksts
"Mechanism": ""	Izmaiņu reģistrēšanas mehānisms ("table" – datu lauku izmaiņas tiek reģistrētas pamattabulā vai "bu_trigger" – izmaiņas tiek reģistrētas izmaiņu tabulā), konkrētam objekta tipam atbilstošo izmaiņu reģistrēšanas mehānismu skat. 2.pielikumā.
"Name": ""	Objekta tipu raksturojošs nosaukums, kas izmantots datu izmaiņu pārlūkošanas formas nosaukumā.
"Columns": ""	Masīvs, kas satur objektus ar informāciju par tabulas kolonnām (struktūru skat. tabulā 2.5.).
"GridCaption": ""	Datu izmaiņu vēstures tabulas nosaukums, kas tiks izmantots datu izmaiņu pārlūkošanas formā.
"RelatedObjects": ""	Objekta saistītie saraksti (struktūru skat. tabulā 2.6.).
"RelatedObjectsCombo": ""	Norāde ("true" vai "false") uz to, vai formā tiek attēlots nolaižamais saraksts ar saistītajiem sarakstiem.
"AdditionalGrid": ""	Masīvs ar parametriem, kas nepieciešami grupēšanas tabulas attēlošanai datu izmaiņu pārlūkošanas formā (struktūru skat. tabulā 2.8.).

Ja konfigurācija tiek padota kā meklēšanas parametri, tad tajā atrodas arī lauks "Mervieniba", kas norāda uz mērvienību (m² vai ha), kāda uzstādīta platību attēlošanai NINO.

Konfigurācijas masīva “*Columns*” objekti satur informāciju par tabulas kolonnām jeb datu laukiem. Šo objektu struktūru skat. tabulā 2.5. Ja ar kādu no parametriem datu lauku raksturot nav iespējams, tas objektā netiek iekļauts vai tā vērtība tiek uzstādīta uz *null*.

2.5. tabula

Objekta sastāvdaļas	Apraksts
"MainColumn": ""	Pamattabulas kolonnas nosaukums.
"BuColumn": ""	Pamattabulas kolonnai atbilstošās izmaiņu tabulas kolonnas nosaukums.
"Datatype": ""	Lauka datu tips.
"Key": ""	Atslēgas vārds, kas raksturo šo kolonnu un tiks izmantots datu bāzes vaicājumā.
"Selectable": ""	Norāde (“true” vai “false”) uz to, vai šī kolonna tiks attēlota formas attēlojamo kolonnu sarakstā.
"Comparable": ""	Norāde (“true” vai “false”) uz to, vai šī kolonna tiks attēlota formas izmaiņu kolonnu sarakstā.
"VisibleName": ""	Lietotājam attēlojamais kolonnas nosaukums.
"Obligatory": ""	Norāde (“true” vai “false”) uz to, ka šis objekts ir obligāts.
"Optional": ""	Norāde (“true” vai “false”) uz to, ka šis objekts ir neobligāts.
"Identifier": ""	Norāde (“true” vai “false”) uz to, ka šī tabulas kolonna satur klasifikatoru.
"DataTable": ""	Ja lauks “Identifier” uzstādīts uz “true”, šeit tiek norādīta klasifikatora tabula, no kuras jāatlasa dati.
"TextColumn": ""	Ja lauks “Identifier” uzstādīts uz “true”, šeit tiek norādīta nepieciešamā teksta kolonna no tabulas “DataTable”.
"IdColumn": ""	Ja lauks “Identifier” uzstādīts uz “true”, šeit tiek norādīta nepieciešamā identifikatora kolonna no tabulas “DataTable”.
"IsUser": ""	Norāde uz to, ka šajā kolonnā glabājas NINO lietotāja identifikators, līdz ar to tiek atrasts lietotāja vārds un uzvārds, kas iegūts no datu bāzes tabulas LIETOTAJI.
"Area": ""	Ja šajā kolonnā norādītā vērtība ir platība, tad šis atzīmēts kā “true”, lai nodrošinātu pareizu platību attēlošanu formā.
"Currency": ""	Ja šajā kolonnā norādītā vērtība lietotājam attēlojama kā valūta (LVL vai EUR), tad šis uzstādīts uz “true”.

Konfigurācijas masīvs “*RelatedObjects*” satur informāciju par objekta saistītajiem sarakstiem. Masīva objekta struktūru skatīt tabulā 2.6.

2.6. tabula

Objekta sastāvdaļas	Apraksts
"TableName": ""	Saistītā saraksta objekta tips jeb datu tabulas nosaukums.
"VisibleName": ""	Lietotājam attēlojamais saistītā saraksta nosaukums.

“*AdditionalGrid*” ir konfigurācijas objekts, kas satur informāciju par grupēšanas tabulu. Objekta struktūru skatīt tabulā 2.7.

2.7. tabula

Objekta sastāvdaļas	Apraksts
"Visible": ""	Norāde (“true” vai “false”) uz to, vai formā būs iespējota grupēšanas tabula.
"GridCaption": ""	Formā attēlojamais grupēšanas tabulas nosaukums.
"Columns": ""	Masīvs ar informāciju par grupēšanas tabulas kolonnām.

Informācija par grupēšanas tabulas kolonnām tiek glabāta konfigurācijas objekta “*AdditionalGrid*” masīvā “*Columns*”. Masīva objektu struktūru skat. tabulā 2.8.

2.8. tabula

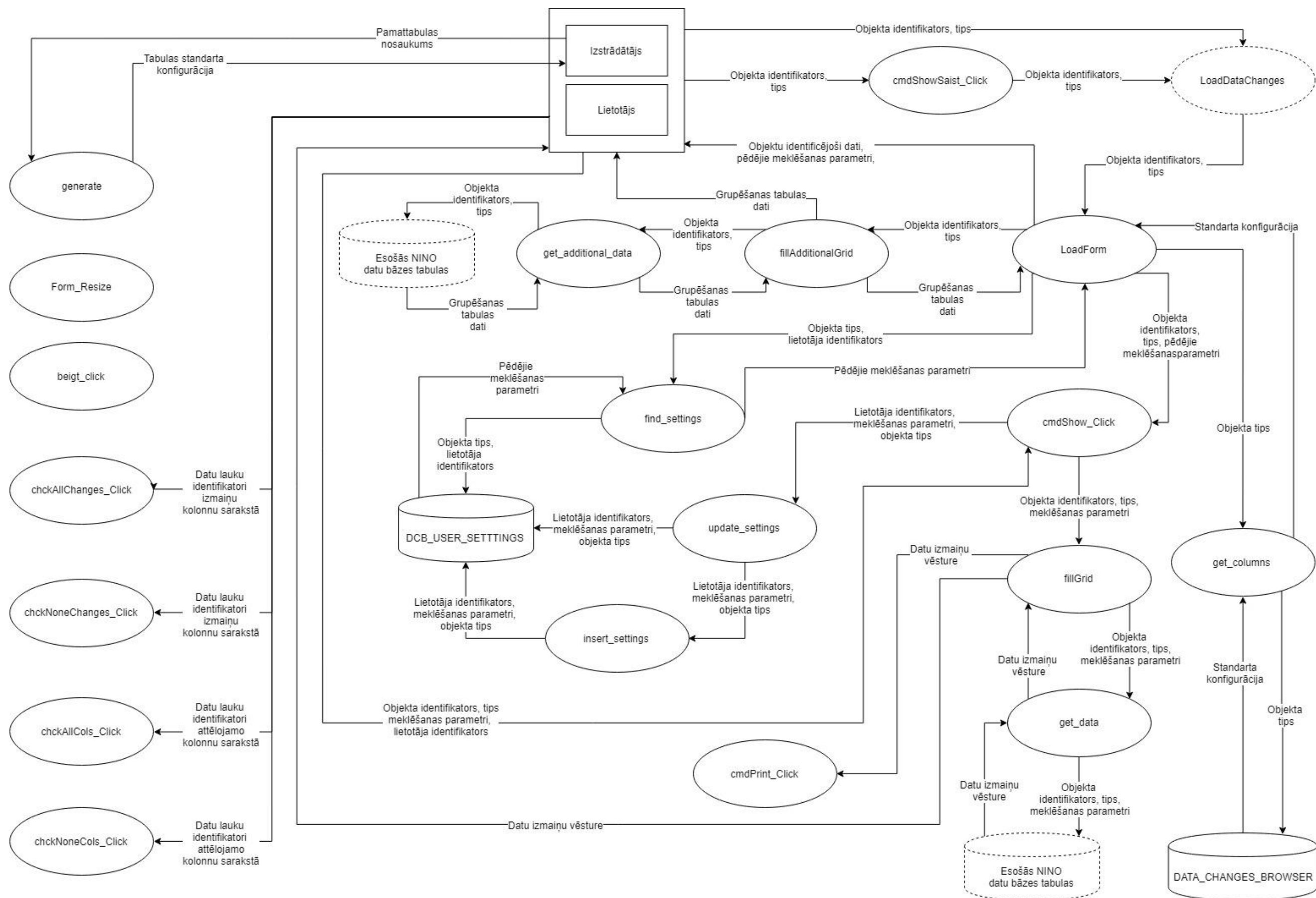
Objekta sastāvdaļas	Apraksts
"Key": ""	Atslēgas vārds, kas raksturo kolonnu un tiks izmantots datu atlases vaicājumā.
"VisibleName": ""	Lietotājam attēlojamais kolonnas nosaukums.
"Datatype": ""	Kolonnas datu tips.
"AdditionalIDCol": ""	Objekta identifikatora kolonnas nosaukums.
"Priority": ""	Objekta identifikatora kolonnas prioritāte (visaugstākā prioritāte ir 1, tātad pirmkārt atlasa pēc datiem no šīs kolonnas šūnas, bet, ja tā ir tukša, tad meklē ar nākamo prioritāti).
"Visible": ""	Norāde (“true” vai “false”) uz to, vai grupēšanas tabulā šī kolonna tiek attēlota.
"Excel": ""	Norāde (“true” vai “false”) uz to, vai šīs kolonnas dati tiek attēloti kā datu atlases kritēriji, drukājot datu izmaiņu vēsturi uz lietotni MS Excel.

2.6.2. Funkcionālais projektējums

Datu izmaiņu pārlūkošanas moduļa funkcijas un procedūras sadalītas pakotnēs un formās, kas ietilpst nodaļā 2.3.1. aprakstītajos moduļos. Šo moduļu un procedūru, funkciju savstarpējo atbilstību skatīt tabulā 2.9. Otrā līmeņa DPD skat. att. 2.7.

2.9. tabula

Moduļa nosaukums	Procedūru, funkciju nosaukumi, nodaļas	Nodaļa
Formas izsaukšanas modulis	LoadForm, beigt_Click, Form_Resize, showShowSaist	2.6.2.5.
Meklēšanas parametru modulis	get_columns, insert_settings, find_settings, update_settings	2.6.2.2.
	chckAllCols_Click, chckAllChanges_Click, chckNoneCols_Click, chckNoneChanges_Click	2.6.2.5.
Datu atlases modulis	get_data, get_additional_data	2.6.2.1.
	fillGrid, cmdShow_Click, cmdPrint_Click, fillAdditionalGrid	2.6.2.5.
	konstantēm	2.6.2.4.
Konfigurāciju ģenerēšanas modulis	generate	2.6.2.3.



2.7. att. 2.līmeņa DPD

2.6.2.1. Pakotne DATA_CHANGES_BROWSER_PCK

Pakotne paredzēta dinamiskā datu bāzes vaicājuma būvēšanai un izsaukšanai, balstoties uz saņemtajiem lietotāja ievadītajiem meklēšanas parametriem vai tabulu konfigurāciju failiem (skat. tabulu 2.10.), kā arī grupēšanas tabulai nepieciešamo datu atlasei (skat. tabulu 2.11.).

Pakotnes procedūrā *get_data* tiek ģenerēts un veidots dinamisks vaicājums, kura saturs balstīts uz konfigurācijas failu. Veidu, kādā dati tiks atlasīti, nosaka tabulai atbilstošais izmaiņu reģistrēšanas mehānisms. Datu izmaiņu pārlūkošanas modulis darbojas sekojošiem mehānismiem:

- *bu_trigger* mehānisms – datu izmaiņas tiek reģistrētas izmaiņu tabulā;
- *table* mehānisms – datu izmaiņas tiek reģistrētas pamattabulā un izmaiņu tabulā nepastāv.

Vaicājums sastāv no 5 galvenajām daļām:

1. nepieciešamā ieraksta atrašana pamattabulā;
2. ar nepieciešamo ierakstu saistīto izmaiņu ierakstu atrašana izmaiņu tabulā (tikai *bu_trigger* mehānismam);
3. izmainīto un aktuālo vērtību noteikšana (tikai *bu_trigger* mehānismam);
4. lietotājam zīmīgu datu atlasīšana;
5. nepieciešamo datu izmaiņu atlasīšanas nosacījums (tikai *bu_trigger* mehānismam).

No augstākminētā uzskaitījuma 1. un 2. daļas tiek implementētas kā vienkārši atlases vaicājumi, ņemot vērā konkrētu ieraksta identifikatoru. Abi vaicājumi tiek savienoti ar operatora *union all* palīdzību, lai iegūtu gan izmaiņu vēstures, gan aktuālos datus. Vaicājumos tiek atlasīti objekta tipam atbilstošie obligātie datu lauki (skat. 1.pielikumā) un pārējie konfigurācijā atrodamie datu lauki.

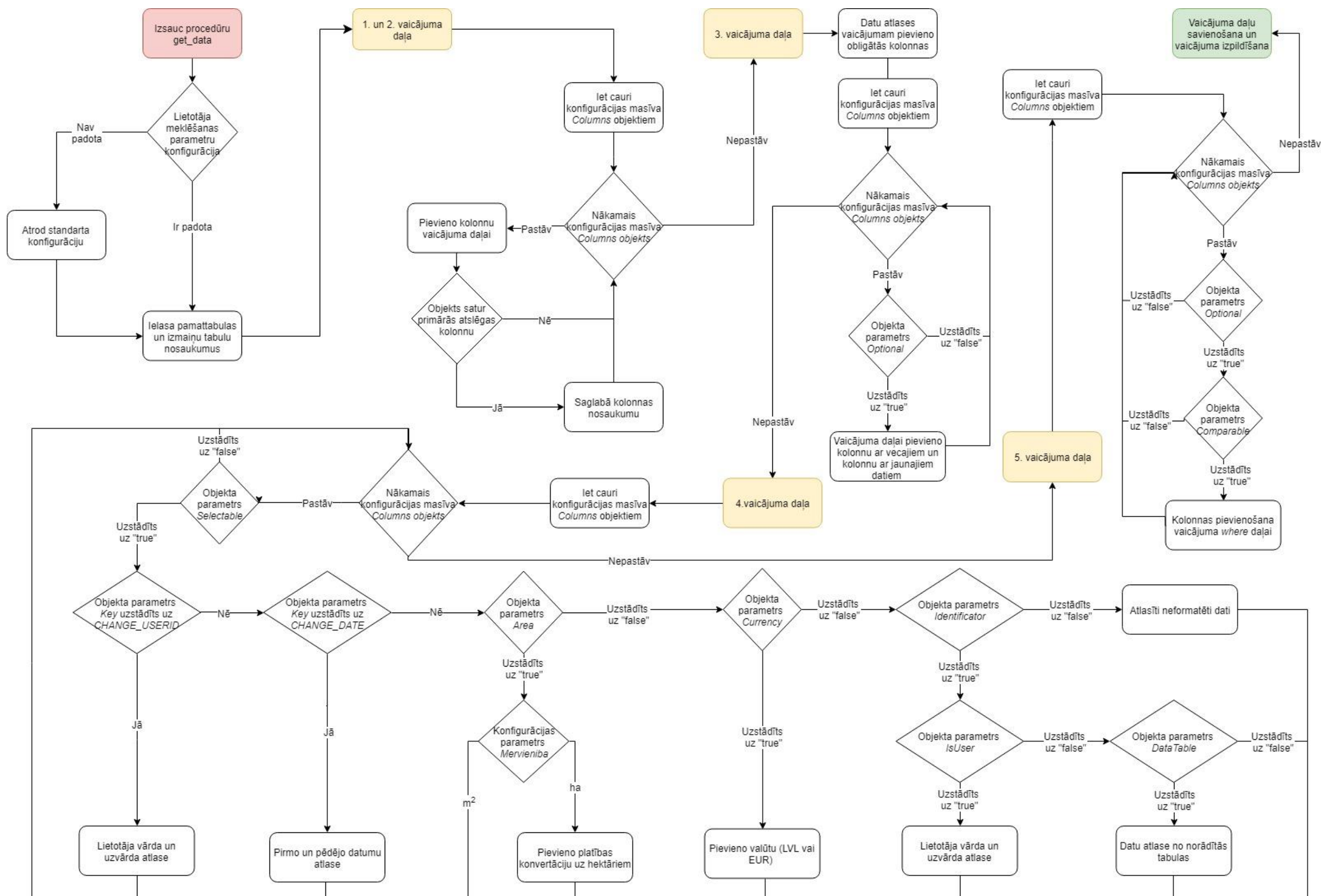
3. daļas pamatā tiek izmantotas analītiskās funkcijas, lai noteiktu uz izmaiņu veikšanas brīdi aktuālās un neaktuālās datu lauku vērtības, pēc kā ar 4. daļas palīdzību no kolonnām, kuras glabā tikai klasifikatorus uz citu tabulu ierakstiem, tiek iegūti lietotājam zīmīgi dati.

Vaicājuma 5. daļa ir SQL vaicājuma nosacījuma daļa (angļu val. – “*where statement*”), kas nosaka, kurām kolonnām piemērot datu atlases kritērijus un noteikt datu izmaiņas. Šajā vaicājuma daļā tiek izmantotas globālās konstantes no datu bāzes pakotnes DATA_CHANGES_BROWSER_CONST_PCK.

Procedūras *get_data* sastādītā vaicājuma paraugu skatīt 4. pielikumā.

Dinamiska vaicājuma ģenerēšana un izpildīšana (procedūra *get_data*)

Mērķis			
Ģenerēt dinamisko vaicājumu, balstoties uz konfigurāciju, un izpildīt to.			
Ievade			
<u>Lauka nosaukums</u>	<u>Datu tips</u>	<u>Obligāts</u>	<u>Apraksts</u>
pi_id	number(10)	+	Objekta identifikators
pi_table	varchar2	+	Pamattabulas nosaukums
pio_columns	clob	-	Lietotāja meklēšanas parametru konfigurācija
Apstrāde			
<ol style="list-style-type: none"> 1. Pārbauda, vai <i>pio_columns</i> ir aizpildīts ar lietotāja meklēšanas parametru konfigurāciju. Ja nav, tad lietotāja meklēšanas parametru vietā ielasa standarta konfigurāciju tabulai ar nosaukumu <i>pi_table</i> no DATA_CHANGES_BROWSER tabulas. 2. Veic konfigurācijas konvertāciju no <i>clob</i> uz <i>json</i> tipu, izmantojot pakotni <i>ZZJSON.JSON_EXT</i> (skat. nodaļu 2.6.2.7). 3. Nosaka izmaiņu reģistrēšanas mehānismu. 4. Apstrādā katru konfigurācijas masīva “<i>Columns</i>” objektu, sastādot dinamisko vaicājumu ar nepieciešamajiem nosaukumiem, SQL sintakses elementiem, atslēgvārdiem un operatoriem, kā arī, ievērojot izmaiņu reģistrēšanas mehānismu: <ul style="list-style-type: none"> • <i>bu_trigger</i> – tiek sastādītas visas vaicājuma daļas (algoritma projektējuma shēmu skat. att. 2.8. - sarkanā krāsā atzīmēts procedūras izsaukums, dzeltenā krāsā vaicājuma daļu sastādīšanas sākums, bet zaļā – procedūras beigas.). • <i>table</i> – tiek sastādītas tikai 1. un 4. vaicājuma daļa. 5. Izpilda sastādīto dinamisko vaicājumu. 			
Izvade			
No tabulām atlasītie dati mainīgajā <i>po_data</i> ar tipu <i>sys_refcursor</i> .			



2.8. att. Procedūras *get_data* algoritma projektējuma blokhēma *bu_trigger* izmaiņu reģistrēšanas mehānisma gadījumā

Grupēšanas tabulas informācijas iegūšana (procedūra *get_additional_data*)

Mērķis			
Atlasīt grupēšanas tabulā attēlojamus datus.			
Ievade			
<u>Lauka nosaukums</u>	<u>Datu tips</u>	<u>Obligāts</u>	<u>Apraksts</u>
pi_id	number(10)	+	Objekta identifikators
pi_table	varchar2	+	Pamattabulas nosaukums
pi_user	varchar2	+	Lietotāja identifikators
pi_apl	varchar2	+	Lietotnes identifikators
Apstrāde			
Tiek noteikts, kādai tabulai grupēšanas tabulas informācijas iegūšana izsaukta un veikta datu atlase.			
Izvade			
No tabulām atlasītie dati mainīgajā <i>po_data</i> ar tipu <i>sys_refcursor</i> .			

2.6.2.2. Pakotne DCB_USER_SETTINGS_PCK

Pakotne DCB_USER_SETTINGS_PCK nodrošina darbības ar lietotāja ievadītajiem meklēšanas parametriem – saglabā jaunus un veic esošo parametru labojumus, kā arī atrod lietotāja pēdējos saglabātos meklēšanas parametrus. Pakotnes funkciju un procedūru aprakstus skatīt tabulās 2.12., 2.13., 2.14., 2.15.

Meklēšanas parametru pievienošana (procedūra *insert_settings*)

Mērķis			
Ja lietotājs konkrētam objekta tipam datu izmaiņu pārlūkošanas formu atvēris pirmo reizi un ievadījis meklēšanas parametrus, tie tiek saglabāti datu bāzē			
Ievade			
<u>Lauka nosaukums</u>	<u>Datu tips</u>	<u>Obligāts</u>	<u>Apraksts</u>
pi_data	clob	+	Meklēšanas parametri
pi_user	number	+	Lietotāja identifikators
pi_apl	number	+	Lietotnes identifikators
pi_table	varchar2	+	Tabulas nosaukums

Apstrāde
Pievieno jaunu ierakstu tabulā DCB_USER_SETTINGS.

2.13. tabula

Meklēšanas parametru atjaunošana (procedūra *update_settings*)

Mērķis			
Atjaunot meklēšanas parametrus, saglabājot pēdējos lietotāja ievadītos meklēšanas parametrus datu bāzē.			
Ievade			
<u>Lauka nosaukums</u>	<u>Datu tips</u>	<u>Obligāts</u>	<u>Apraksts</u>
pi_data	clob	+	Meklēšanas parametri
pi_user	number	+	Lietotāja identifikators
pi_apl	number	+	Lietotnes identifikators
pi_table	varchar2	+	Tabulas nosaukums
Apstrāde			
No <i>pi_data</i> iegūst tabulas nosaukumu un pārbauda, vai lietotājam ar identifikatoru <i>pi_user</i> eksistē ieraksts ar tabulas nosaukumu <i>pi_table</i> :			
<ul style="list-style-type: none"> Jā - atjauno meklēšanas parametrus, ja tie nesakrīt ar tabulā jau esošajiem; Nē - pievieno jaunus parametrus (skat. tabulā 2.12.). 			
Izvade			
Lietotāja meklēšanas parametri konkrētajai tabulai mainīgajā <i>po_data</i> ar tipu <i>clob</i> .			

2.14. tabula

Pēdējo meklēšanas parametru atrašana (procedūra *find_settings*)

Mērķis			
Atrast lietotāja pēdējos saglabātos meklēšanas parametrus konkrētam objekta tipam.			
Ievade			
<u>Lauka nosaukums</u>	<u>Datu tips</u>	<u>Obligāts</u>	<u>Apraksts</u>
pi_data	clob	+	Meklēšanas parametri
pi_user	number	+	Lietotāja identifikators
pi_apl	number	+	Lietotnes identifikators
pi_raj	number	+	Pilsētas vai novada identifikators
Apstrāde			

No <i>pi_data</i> iegūst tabulas nosaukumu un tabulā DCB_USER_SETTINGS atrod, kuri meklēšanas parametri atbilst konkrētajam lietotājam un tabulai. Ja nekas netiek atrasts, tas nozīmē, ka lietotājs šādam objekta tipam formu ir atvēris pirmo reizi, līdz ar to tiek atrasta un atgriezta standarta konfigurācija šai tabulai.
Izvade
Lietotāja pēdējie saglabātie meklēšanas parametri vai standarta konfigurācija ar datu tipu <i>clob</i> .

2.15. tabula

Tabulai atbilstošās konfigurācijas atrašana (procedūra *get_columns*)

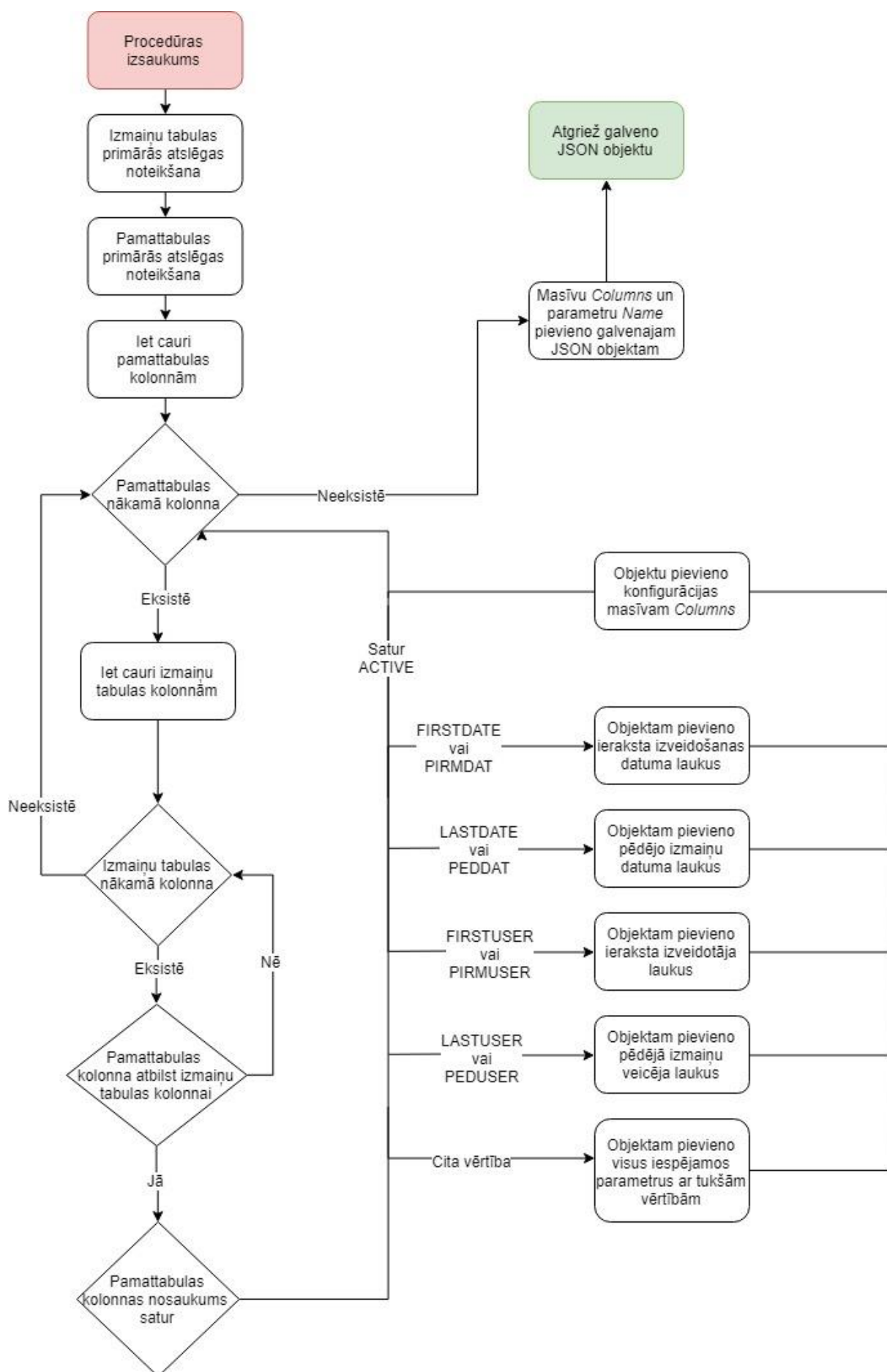
Mērķis			
Atrast tabulai atbilstošo standarta konfigurāciju no tabulas DATA_CHANGES_BROWSER datu lauku attēlošanai datu izmaiņu pārlūkošanas formā.			
Ievade			
<u>Lauka nosaukums</u>	<u>Datu tips</u>	<u>Obligāts</u>	<u>Apraksts</u>
pi_data	clob	+	Lietotāja meklēšanas parametri
pi_user	number	+	Lietotāja identifikators
pi_apl	number	+	Lietotnes identifikators
pi_raj	number	+	Pilsētas vai novada identifikators
Apstrāde			
No saņemtajiem meklēšanas parametriem iegūst tabulas nosaukumu un atgriež atbilstošās tabulas standarta konfigurāciju.			
Izvade			
Konkrētās tabulas standarta konfigurācija mainīgajā <i>po_data</i> ar tipu <i>clob</i> .			

2.6.2.3. Pakotne DATA_CHANGES_BROWSER_CONF_PCK

DATA_CHANGES_BROWSER_CONF_PCK pakotne ir paredzēta daļēji automātiskai tabulu konfigurāciju ģenerēšanai. Pakotnē ir funkcijas un procedūras, kas sasaista pamattabulu un izmaiņu tabulu kolonnas, tādā veidā ģenerējot konfigurācijas masīvu “Columns” objektus. Obligātie konfigurācijas objekti tiek aizpildīti automātiski, bet neobligāto objektu ģenerēšana ir daļēji automātiska. Tāpēc pirms ievietošanas DATA_CHANGES_BROWSER tabulā konfigurācija ir jāpārskata un jāpievieno vai jāredīģē vajadzīgie lauki. Pakotnes funkcijas *generate* aprakstu skatīt tabulā 2.16.

Standarta konfigurācijas ģenerēšana (funkcija *generate*)

Mērķis			
Ģenerēt tabulas standarta konfigurāciju.			
Ievade			
<u>Lauka nosaukums</u>	<u>Datu tips</u>	<u>Obligāts</u>	<u>Apraksts</u>
pi_table	varchar2	+	Tabulas nosaukums
Apstrāde			
<p>Vispirms tiek atrastas pamattabulas un izmaiņu tabulas primārās atslēgas. Pēc tam katrai pamattabulas kolonnai tiek piemeklēta atbilstošā izmaiņu tabulas kolonna, kā arī atpazīti un pievienoti jau aizpildīti obligātie lauki. Neobligātajiem laukiem aizpildītas tikai “MainColumn”, “BuColumn” un “DataType” vērtības, pārējās pievienotas, bet atstātas tukšas. Algoritma projektējuma shēmu skat. att. 2.9. (ar sarkanu krāsu atzīmēts procedūras izsaukums jeb algoritma sākums, bet ar zaļu krāsu – procedūras iznākums).</p>			
Izvade			
JSON objekts ar tabulas <i>pi_table</i> konfigurāciju mainīgajā result ar tipu <i>clob</i> .			



2.9. att. Funkcijas *generate* projektējuma algoritma blokhēma

2.6.2.4. Pakotne DATA_CHANGES_BROWSER_CONST_PCK

Pakotne DATA_CHANGES_BROWSER_CONST_PCK paredzēta konstanšu glabāšanai. Tajā atrodas vairākiem datu tiem (*varchar2*, *number*, *date*, *timestamp*) atbilstošas konstantes, kas nepieciešamas dinamiskā vaicājuma veidošanas procesā. Šīs konstantes tiek iznestas ārpus pakotnes, kurā tās tiks izmantotas, lai pakotne nekļūtu sesijas jūtīga.

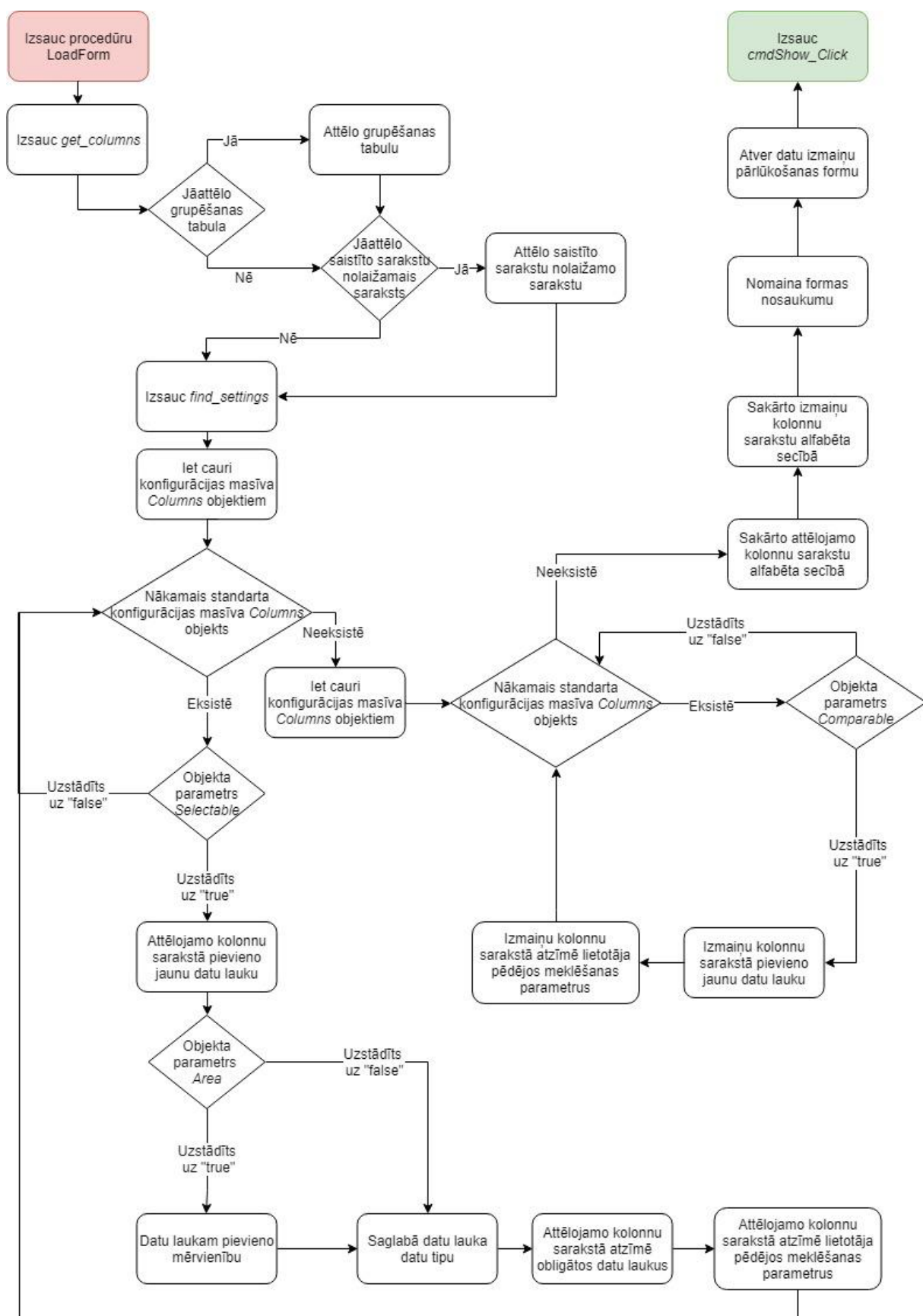
2.6.2.5. Forma frmDataChangesBrowser

Forma frmDataChangesBrowser ir Microsoft Visual Basic 6 izstrādes vides forma, kas paredzēta datu bāzes procedūru izsaukšanai, saņemto datu apstrādei, kārtošanai un attēlošanai. Formas funkcijas un procedūras skatīt tabulās 2.17., 2.18., 2.19., 2.20., 2.21., 2.22., 2.23., 2.24., 2.25., 2.26., 2.27., 2.28.

2.17. tabula

Datu izmaiņu pārlūkošanas formas atvēršana (procedūra *LoadForm*)

Mērķis			
Atvērt formu un parādīt datu laukus, kurus iespējams attēlot un kuru izmaiņas pārlūkot.			
Ievade			
<u>Lauka nosaukums</u>	<u>Datu tips</u>	<u>Obligāts</u>	<u>Apraksts</u>
p_id	string	+	Objekta identifikators
tabl	string	+	Objekta tabulas nosaukums jeb tips
show	boolean	+	Vai rādīt datus uzreiz pēc formas atvēršanas (“true” – jā, “false” – nē)
caption	string	+	Objektu identificējoši dati
Apstrāde			
Izsauc funkcijas <i>get_columns</i> (skat. tabulu 2.15.) un <i>find_settings</i> (skat. tabulu 2.14.). Pievieno formas nosaukumam objektu identificējošos datus (<i>caption</i>), aizpilda attēlojamo un izmaiņu kolonnu sarakstus, attēlojot tajos lietotāja pēdējos meklēšanas parametrus un, ja parametrs <i>show</i> ir uzstādīts uz “true”, izsauc procedūru <i>cmdShowClick</i> (skat. tabulu 2.19.), lai attēlotu datu izmaiņu vēsturi. Algoritma projektējuma shēmu skatīt att 2.10. (ar sarkanu krāsu atzīmēts procedūras izsaukums, ar zaļu krāsu – procedūras beigas).			



2.10. att. Procedūras LoadForm projektējuma algoritma blokshēma

Datu izmaiņu vēstures tabulas aizpildīšana (procedūra *FillGrid*)

Mērķis			
Aizpildīt datu izmaiņu vēstures tabulu ar paredzētajiem datiem.			
Ievaddati			
<u>Lauka nosaukums</u>	<u>Datu tips</u>	<u>Obligāts</u>	<u>Apraksts</u>
g_mekl_param	Object	+	Lietotāja meklēšanas parametri
selectedCols	Object	+	Standarta konfigurācija
Apstrāde			
Izsauc datu bāzes procedūru <i>get_data</i> (skat. nodaļu 2.6.2.1) un aizpilda datu izmaiņu vēstures tabulu <i>tdbResult(0)</i> (ar datu tipu <i>TDBGrid</i>) ar saņemtajiem datiem.			

Kolonnu parādīšana tabulā (procedūra *cmdShow_Click*)

Mērķis		
Parādīt lietotāja meklēšanas parametrus atzīmētās attēlojamās kolonnas tabulā un saglabāt meklēšanas parametrus.		
Ievaddati		
<u>Lauka nosaukums</u>	<u>Obligāts</u>	<u>Apraksts</u>
g_mekl_param	+	Meklēšanas parametri
Apstrāde		
<p>Pārbauda, vai attēlojamo kolonnu sarakstā atzīmēts vismaz viens datu lauks:</p> <ul style="list-style-type: none"> Jā - balstoties uz meklēšanas parametriem, datu izmaiņu vēstures tabulai <i>tdbResult(0)</i> pievieno jaunas kolonnas (tādā secībā, kādā tās atrodas konfigurācijā) un, izmantojot moduli <i>modJSON</i> (sakt. 2.6.2.7), ģenerē lietotāja meklēšanas parametru konfigurāciju. Pēc tam izsauc meklēšanas parametru saglabāšanas procedūru <i>update_settings</i> (skat. tabulu 2.13. tabula). Pārbauda, vai formā paredzēts attēlot grupēšanas tabulu: <ul style="list-style-type: none"> Jā – meklēšanas parametru konfigurācijai pievieno grupēšanas tabulas informāciju, pēc tam izsauc procedūru <i>FillGrid</i> (skat. tabulu 2.18.); Nē - izsauc procedūru <i>FillGrid</i> (skat. 2.18.); Nē - P1 (sistēmas paziņojumus skatīt PPS nodaļā 1.3.2.). 		

Formas izmēra maiņa (procedūra *Form_Resize*)

Mērķis
Atverot datu izmaiņu pārlūkošanas formu vai mainot tās izmērus, atbilstoši pielāgot arī formā esošo pogu, sarakstu, tabulu novietojumu un izmērus, pēc nepieciešamības nodrošināt formas papildus sastāvdaļu (grupēšanas tabula, saistīto sarakstu nolaižamais saraksts) parādīšanos formā.
Apstrāde
Attēlojamo un izmaiņu kolonnu sarakstu augstumu maina līdz ar formas augstumu, platumu atstājot nemainīgu. Datu izmaiņu vēstures un grupēšanas tabulu augstumu un platumu maina līdz ar formas augstumu un platumu. Pogas maina savu novietojumu relatīvi pret lauku izmēru. Atbilstoši tabulas standarta konfigurācijai, tiek parādītas vai noņemtas papildus formas sastāvdaļas (grupēšanas tabula, saistīto sarakstu nolaižamais saraksts).

Visu parametru atzīmēšana attēlojamo kolonnu sarakstā (procedūra *chkAllCols_Click*)

Mērķis
Atzīmēt visus parametrus attēlojamo kolonnu sarakstā.
Apstrāde
Tiek atzīmētas visas izvēles rūtiņas attēlojamo kolonnu sarakstā.

Visu parametru atzīmēšana izmaiņu kolonnu sarakstā (procedūra *chkAllChanges_Click*)

Mērķis
Atzīmēt visus parametrus izmaiņu kolonnu sarakstā.
Apstrāde
Tiek atzīmētas visas izvēles rūtiņas izmaiņu kolonnu sarakstā, kā arī tiek pārbaudīts, vai datu lauks ar šādu vērtību ir atzīmēts attēlojamo kolonnu sarakstā: <ul style="list-style-type: none"> Jā - atzīme pie izvēles rūtiņas tiek saglabāta; Nē - izvēles rūtiņa tiek atzīmēta.

Visu parametru atzīmju noņemšana attēlojamo kolonnu sarakstā (procedūra *chckNoneCols_Click*)

Mērķis
Noņemt visas atzīmes no parametriem attēlojamo kolonnu sarakstā.
Apstrāde
Tiek pārbaudīts, vai datu lauks ar šādu vērtību jau ir ticis atzīmēts izmaiņu kolonnu sarakstā: <ul style="list-style-type: none"> Jā - atzīme pie izvēles rūtiņas tiek saglabāta; Nē - atzīme no izvēles rūtiņas tiek noņemta. Atzīmes izvēles rūtiņās tiek atstātas arī obligātajiem datu laukiem (obligātos datu laukus skat. 1.pielikumā).

Visu parametru atzīmju noņemšana izmaiņu kolonnu sarakstā (procedūra *chckNoneCols_Click*)

Mērķis
Noņemt visas atzīmes no parametriem izmaiņu kolonnu sarakstā.
Apstrāde
Tiek noņemtas visas atzīmes no izvēles rūtiņām izmaiņu kolonnu sarakstā.

Datu izmaiņu vēstures parādīšana lietotnē Microsoft Excel (procedūra *CmdPrint_Click*)

Mērķis
Parādīt datu izmaiņu vēsturi lietotnē Microsoft Excel.
Apstrāde
Tiek izsaukta NINO projekta moduļa <i>NeNi_I_INFOABI</i> funkcija <i>ExportToExcel</i> , kam kā ieejas parametri tiek padoti: datu izmaiņu vēstures tabula <i>tdbResult(0)</i> (skat. tabulu 2.18.), pati forma <i>frmDataChangesBrowser</i> , kā arī masīvs, kas glabā datu izmaiņu vēstures tabulas <i>tdbResult(0)</i> kolonnu datu tipu identifikatorus, lai lietotnē Microsoft Excel šūnās tiktu noteikts pareizais datu tips.

Formas aizvēršana (procedūra *beigt_Click*)

Mērķis
Aizvērt datu izmaiņu pārlūkošanas formu.

Apstrāde
Datu izmaiņu pārlūkošanas forma tiek aizvērta.

2.27. tabula

Grupēšanas tabulas aizpildīšana (procedūra *fillAdditionalGrid*)

Mērķis		
Aizpildīt grupēšanas tabulu ar datiem.		
Ievaddati		
<u>Lauka nosaukums</u>	<u>Obligāts</u>	<u>Apraksts</u>
Table	+	Objekta tips
Apstrāde		
Izsauc datu bāzes procedūru <i>get_additional_data</i> (skat. tabulu 2.11.) un aizpilda tabulas <i>tdbResult(1)</i> kolonnas ar atlasītajiem datiem.		

2.28. tabula

Datu izmaiņu pārlūkošanas formas atvēršana saistītajam sarakstam (procedūra *cmdShowSaist_Click*)

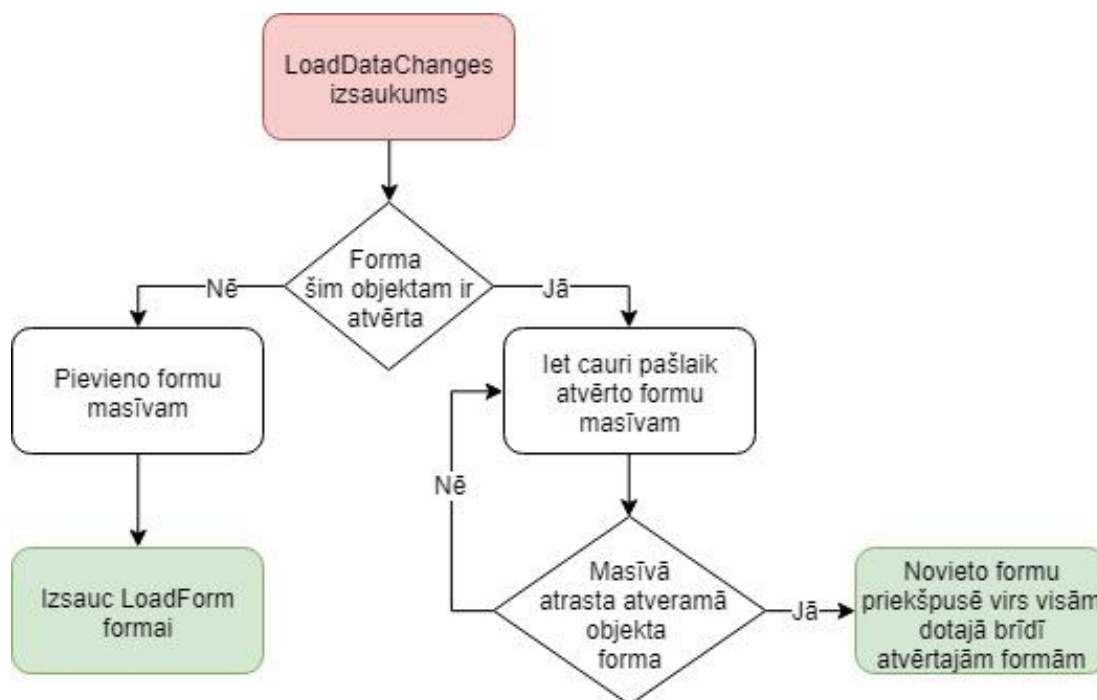
Mērķis
Atvērt datu izmaiņu pārlūkošanas formu saistītā saraksta objektam.
Apstrāde
Tiek izsaukta moduļa <i>NeNi_A_modLabaPuse</i> procedūra <i>LoadDataChanges</i> (skat. tabulu 2.29.), esošā objekta tipa vietā padodot saistītā saraksta objekta tipu.

2.6.2.6. Izmaiņas NINO vienumos

Lai nodrošinātu datu izmaiņu vēstures formas atvēršanu no saraksta formas, jāveic izmaiņas *frmSaraksti* formā. Formas procedūrai *tlbDok_ButtonClick* pie Datu izmaiņu pārlūkošanas moduļa atvēršanas pogas (līdz šim - poga, kas nodrošināja iepriekšējā datu izmaiņu pārlūkošanas risinājuma atvēršanu) jāpievieno papildus nosacījumi – ja sarakstā esošā objekta tips ir *parceles*, *tasks*, *atv_reg* vai *personas_nod*, tiek izsaukta moduļa *NeNi_A_modLabaPuse* procedūra *LoadDataChanges* (skat. tabulu 2.29) ar visiem funkcijai *LoadForm* (skat. tabulu 2.17.) nepieciešamajiem ieejas parametriem. Darba procesu/uzdevumu un atveiglojumu reģistra sarakstos datu izmaiņu pārlūkošanas moduļa formas atvēršanas poga jāiespējo.

Procedūra LoadDataChanges

Mērķis			
Nodrošināt iespēju atvērt vienu vai vairākas datu izmaiņu pārlūkošanas formas reizē, nodrošinot to nedublēšanos.			
Ievade			
Lauka nosaukums	Datu tips	Obligāts	Apraksts
in_ID	String	+	Objekta identifikators
in_table	String	+	Objekta tabulas nosaukums jeb tips
show	Boolean	+	Vai rādīt datus uzreiz pēc formas atvēršanas (“True” – jā, “False” – nē)
Caption	String	+	Objektu identificējoši dati
Apstrāde			
<p>Pārbauda, vai forma konkrētajam objektam jau ir atvērta:</p> <ul style="list-style-type: none"> • Jā – atrod to starp atvērtajām formām un novieto priekšpusē virs visām pārējām dotajā brīdī atvērtajām formām; • Nē – izveido jaunu formas objektu un izsauc tam procedūru <i>LoadForm</i> (skat. att. 2.11.). 			

2.11. att. Procedūras *LoadDataChanges* projektējuma algoritma blokshēma

2.6.2.7. Pastāvošo NINO risinājumu izmantošana

Lai nodrošinātu datu izmaiņu pārlūkošanas moduļa funkcionalitāti, nepieciešams izmantot jau pastāvošus NINO risinājumus. Izmantojamus vienumus un to izmantošanas mērķi skatīt tabulā 2.30.

2.30. tabula

Izmantojamie pastāvošie NINO risinājumi

Pakotne/modulis	Procedūra/funkcija	Ieejas parametri	Izmantošanas mērķis
ZZJSON.JSON_EXT	get_string	<ul style="list-style-type: none"> •JSON objekts •JSON ceļš 	Teksta virknes iegūšana no JSON objekta.
	get_json_list	<ul style="list-style-type: none"> •JSON objekts •JSON ceļš 	Masīva iegūšana no JSON objekta.
	put	<ul style="list-style-type: none"> •JSON objekts •JSON ceļš •JSON elements 	Objekta pievienošanai JSON failam.
NeNi_I_INFOABI	ExportToExcel	<ul style="list-style-type: none"> •Formas datu izmaiņu pārlūkošanas tabula (tdbResult(0)) •Masīvs, kas atsaucas uz tabulas kolonnu datu tipiem. 	Datu tabulas eksportēšana lietotnē MS Excel.
NeNi_A_modKopejais	ir_forma	<ul style="list-style-type: none"> •Formas atzīme (angļu val. "Tag") 	Pārbaude, vai šī forma dotajā brīdī ir atvērta.
NINO_ATV	get_pers_obj	<ul style="list-style-type: none"> •Konta numurs •Atvieglojuma identifikators •Atvieglojuma tips •Lietotāja identifikators •Lietotnes identifikators •Datums 	Ar personu saistīto objektu saraksta iegūšana (paredzēts grupēšanas tabulai datu izmaiņu pārlūkošanas formā).
modJSON	CreateJSON	Nav	Izveidot JSON objektu
	CreateJSONList	Nav	Izveidot JSON masīva objektu
	ExecuteJSONHelper	<ul style="list-style-type: none"> •Datu bāzes pakotnes un procedūras nosaukums •Datu bāzes procedūrai nepieciešamie ieejas parametri 	Datu bāzes procedūras izsaukšana.
	ToString	<ul style="list-style-type: none"> •JSON objekts 	JSON formāta mainīgā datu konvertācija uz datu tipu String.
NeNi_I_ModKopejais_DAO	OraProc2	<ul style="list-style-type: none"> •Datu bāzes pakotnes un procedūras nosaukums 	Datu bāzes procedūras izsaukšana.

Pakotne/modulis	Procedūra/ funkcija	Ieejas parametri	Izmantošanas mērķis
		<ul style="list-style-type: none"> Datu bāzes procedūrai nepieciešamie ieejas parametri 	
	kluda	<ul style="list-style-type: none"> Kļūda Formas un funkcijas/ procedūras nosaukums 	Kļūdu reģistrēšanai datu bāzes tabulā

2.7. Trasējamības tabula

Trasējamības tabulu (no PPS funkcijām uz PPA nodaļām) skatīt tabulā 2.31.

2.31. tabula

PPS	PPA nodaļas
Formas atvēršana (FIM01)	Forma frmDataChangesBrowser (2.6.2.5.), izmaiņas NINO vienumos (2.6.2.6.), pastāvošo NINO risinājumu izmantošana (2.6.2.7.)
Datu lauku attēlošana (MPM01)	Pakotne DCB_USER_SETTINGS PCK (2.6.2.2.), forma frmDataChangesBrowser (2.6.2.5.), pastāvošo NINO risinājumu izmantošana (2.6.2.7.)
Datu lauku atzīmēšana (MPM02)	Forma frmDataChangesBrowser (2.6.2.5)
Visu datu lauku atzīmēšana attēlojamo kolonnu sarakstā (MPM03)	Forma frmDataChangesBrowser (2.6.2.5)
Visu atzīmju noņemšana no datu laukiem attēlojamo kolonnu sarakstā (MPM04)	Forma frmDataChangesBrowser (2.6.2.5)
Visu datu lauku atzīmēšana izmaiņu kolonnu sarakstā (MPM05)	Forma frmDataChangesBrowser (2.6.2.5)
Visu atzīmju noņemšana no datu laukiem izmaiņu kolonnu sarakstā (MPM06)	Forma frmDataChangesBrowser (2.6.2.5)
Pēdējo lietotāja meklēšanas parametru attēlošana (MPM07)	Pakotne DCB_USER_SETTINGS PCK (2.6.2.2.), forma frmDataChangesBrowser (2.6.2.5.), pastāvošo NINO risinājumu izmantošana (2.6.2.7.)
Datu izmaiņu vēstures attēlošana uz ekrāna (DAM01)	Pakotne DATA_CHANGES_BROWSER_PCK (2.6.2.1.), Pakotne DATA_CHANGES_BROWSER_CONST_PCK (2.6.2.4.), forma frmDataChangesBrowser (2.6.2.5.), pastāvošo NINO risinājumu izmantošana (2.6.2.7.)
Datu izmaiņu vēstures attēlošana lietotnē Microsoft Excel (DAM02)	Forma frmDataChangesBrowser (2.6.2.5.), izmaiņas NINO vienumos (2.6.2.6.), pastāvošo NINO risinājumu izmantošana (2.6.2.7.)

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Ievads

3.1.1. Nolūks

Testēšanas dokumentācijas nolūks ir aprakstīt datu izmaiņu pārlūkošanas moduļa testēšanas gaitu, balstoties uz PPS uzstādītajām prasībām un PPA aprakstīto projektējumu, kā arī fiksēt veiktās testēšanas rezultātus. Dokuments paredzēts sistēmas izstrādātājiem.

3.1.2. Darbības sfēra

Datu izmaiņu pārlūkošanas moduļa mērķis ir nodrošināt iespēju uzskatāmā, ērtā un individuāli pielāgojamā veidā pārlūkot dažādu NINO objektu datu lauku izmaiņu vēsturi. Modulis paredzēts visiem NINO lietotājiem, kuriem ir atbilstošās piekļuves tiesības. Izstrādājot šādu moduli, NINO objektu izmaiņas kļūst trasējamas, līdz ar to tiek paaugstināta nodokļu inspektoru darba efektivitāte, atvieglota darbinieku kontrole ar datu izmaiņu veikšanu saistītos jautājumos, kā arī samazināta nepieciešamība griezties pēc palīdzības pie NINO klientu konsultantiem ar datu izmaiņām saistītos jautājumos.

3.1.3. Definīcijas

Definīcijas un saīsinājumus skatīt nodaļā “Vārdnīca”.

3.2. Saistība ar citiem dokumentiem

Testēšanas dokumentācija tika izstrādāta, balstoties uz PPS un PPA, kā arī ievērojot standartu LVS 70:1996 “Programmatūras testēšanas dokumentācija” [10].

3.3. Testēšanas plāns

Testēšanu paredzēts veikt divos līmeņos. Pirmais testēšanas līmenis ir automatizēta vienībtestēšana (skat. nodaļu 3.3.1). Izstrādātājs programmēšanas gaitā līdz ar katras jaunas funkcijas/funkcionalitātes ieviešanu izstrādā šai funkcijai atbilstošus vienībtestus, izmantojot *utPLSQL* automatizētās vienībtestēšanas ietvaru (testpiemēra programmkoda fragmentu skat. 8.pielikumā). Šie testi automātiski tiek darbināti automatizācijas platformā *Jenkins* reizi

diennaktī, bet nepieciešamības gadījumā iespējams testēšanu uzsākt arī manuāli. Lai izstrādātu vienībtestus, datu bāzē nepieciešams izveidot jaunu lietotāju, kur glabāsies vienībtestu pakotnes (paši testi jānoformē atbilstoši *utPLSQL* dokumentācijai [5]). Automatizācijas platformā *Jenkins* jāizveido jauns testēšanas viens. Testus nepieciešams darbināt pēc katras funkcionalitātes ieviešanas. Kļūdu gadījumos tiek saņemts paziņojums uz izstrādātāja e-pastu ar saiti uz *Jenkins*, kur iespējams apskatīt radušās kļūdas aprakstu.

Otrais testēšanas līmenis ir PPS funkcionālā testēšana un augsta līmeņa funkcionālā testēšana. Šajos līmeņos datu izmaiņu pārlūkošanas modulis tiek testēts manuāli, ievērojot testpiemēra scenārijus. Testēšanas laikā tiek pārbaudītas visas PPS minētās funkcionālās prasības. Šo testēšanu paredzēts veikt pēc datu izmaiņu pārlūkošanas moduļa programmkoda izstrādes. Testēšanas scenāriji, sagaidāmie un faktiskie rezultāti tiek pierakstīti testēšanas žurnālā. Ja radušās kļūdas, tās tiek izlabotas, labojumi reģistrēti testēšanas kļūdu apstrādes žurnālā un notiek šīs funkcijas un ar to saistīto funkciju atkārtota testēšana.

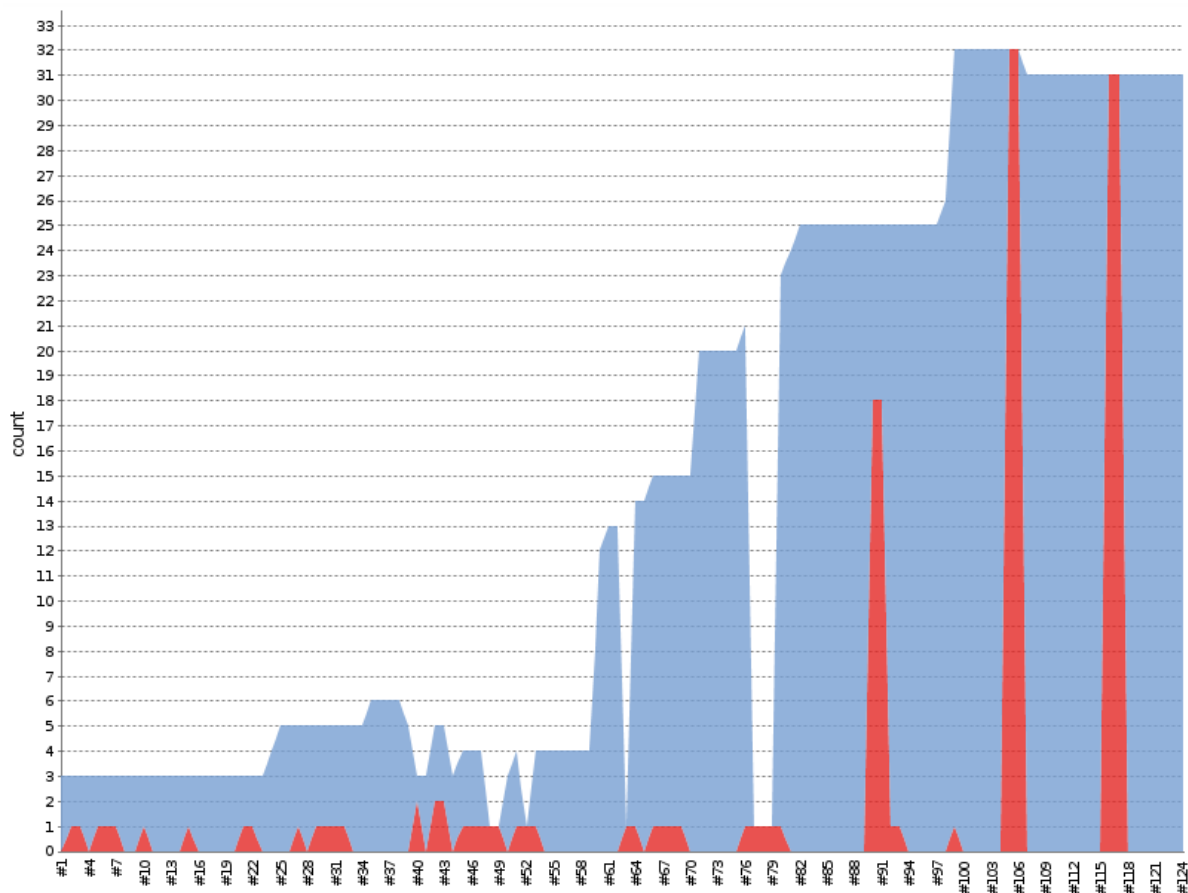
3.3.1. Automatizēta vienībtestēšana

Automatizētās vienībtestēšanas žurnāls tiek nodrošināts ar automatizācijas platformas *Jenkins* palīdzību. Tajā (skat. att. 3.1.) apskatāma veiktās vienībtestēšanas vēsture un iegūtie testēšanas rezultāti. *Jenkins* izsūta informējošu e-pastu, ja tests ir neveiksmīgs vai tas pēc neveiksmes pirmo reizi ir bijis veiksmīgs.

Builds for Lauma Svilpe			
	Build	Time Since ↑	Status
✓	Zzdats.CUKA.ZR.DataChanges.Database #118	6 days 2 hr	back to normal
!	Zzdats.CUKA.ZR.DataChanges.Database #117	6 days 2 hr	broken since build #116
✓	Zzdats.CUKA.ZR.DataChanges.Database #110	11 days	stable
✓	Zzdats.CUKA.ZR.DataChanges.Database #109	11 days	stable
✓	Zzdats.CUKA.ZR.DataChanges.Database #108	12 days	stable
✓	Zzdats.CUKA.ZR.DataChanges.Database #107	12 days	back to normal
!	Zzdats.CUKA.ZR.DataChanges.Database #106	12 days	broken since build #105
✓	Zzdats.CUKA.ZR.DataChanges.Database #101	15 days	stable
✓	Zzdats.CUKA.ZR.DataChanges.Database #100	15 days	back to normal

3.1. att. Automatizētās vienībtestēšanas žurnāls automatizācijas platformā Jenkins

Veiktās vienībtestēšanas rezultātus iespējams apskatīt arī grafiski (att. 3.2.), kas nodrošina iespēju novērtēt vispārējo testēšanas kopskatu.



3.2. att. Automatizētās vienībtestēšanas žurnāla grafisks attēlojums automatizācijas platformā Jenkins

Tāpat *Jenkins* pieejams kopsavilkums par vispārējo testēšanas stāvokli (skat. att. 3.3.) .


W	Description	%
🔍	Test Result: 0 tests failing out of a total of 31 tests.	100
🔍	Build stability: No recent builds failed.	100

3.3. att. Automatizētās vienībtestēšanas kopsavilkums automatizācijas platformā Jenkins

Nepieciešamības gadījumā iespējams apskatīt konkrētu vienībtestēšanas būvējumu, kā arī apskatīt katru testu atsevišķi. Ja tests iziets neveiksmīgi, tiek parādīta informācija, kas ar to saistīta, kā arī iespējams uzstādīt rādāmo ziņojumu (skat. att. 3.4.).

Failed

datachanges.dcb_ut_pck.CheckDirectPERSONASQuery (from datachanges.dcb_ut_pck.dcb_ut_pck)

Failing for the past 2 builds (Since  #21)

[Took 4 days 3 hr.](#)

 [add description](#)

Stacktrace

```
"Incorrect selected row count directly from PERSONAS_NOD and BU_PERSONAS_NOD tables"  
Actual: 0 (number) was expected to equal: 5 (number)  
at "ZR_UT.DCB_UT_PCK.CHECK_DIRECT_PERSONAS_QUERY", line 154 ut.expect(a_actual => v_counter,  
a_message => 'Incorrect selected row count directly from PERSONAS_NOD and BU_PERSONAS_NOD  
tables').to_equal(5);
```

```
"Invalid direct query from PERSONAS_NOD and BU_PERSONAS_NOD tables"  
Actual: FALSE (boolean) was expected to equal: TRUE (boolean)  
at "ZR_UT.DCB_UT_PCK.CHECK_DIRECT_PERSONAS_QUERY", line 155 ut.expect(a_actual => v_valid,  
a_message => 'Invalid direct query from PERSONAS_NOD and BU_PERSONAS_NOD  
tables').to_equal(true);
```

3.4. att. Neizdevies vienībtests automatizācijas platformā Jenkins

Vienībtestēšanai izvēlēts manuāli pievienot ierakstus pamattabulai, rediģēt tos (kā rezultātā rodas ieraksti izmaiņu tabulā) un izsaukt datu lauku izmaiņu vēstures atlases procedūru (*get_data*). Procedūras atgrieztie dati tiek salīdzināti ar pievienotajiem datiem. Rezultātā iespējams noskaidrot, vai procedūras algoritms strādā korekti – pareizi sasaista izmainītos datu laukus ar laiku un lietotāju, kas tos izmainījis, kā arī attēlo datu izmaiņu vēsturi pareizā secībā. Vienībtestēšanā tiek pārbaudīta arī atsevišķu funkciju, kas nav saistīta ar manuāli pievienotajiem tabulu ierakstiem, darbība.

Automatizēto vienībtestēšanu paredzēts veikt Oracle datu bāzes funkcijām un procedūrām, jo uz tām balstās datu izmaiņu pārlūkošanas moduļa galvenā funkcionalitāte.

3.4. Testēšanas žurnāls

3.4.1. PPS funkcionālā testēšana

PPS funkcionālajā testēšanā par pamatu ņemta katra PPS funkcija un no dažādiem aspektiem pārbaudīta tās izpilde programmā (skat. tabulā 3.1). Datu izmaiņu pārlūkošanas forma turpmāk apzīmēta kā “DIPF”.

3.1. tabula

PPS funkcionālās testēšanas žurnāls

Nr.	PPS funkcija	Testpiemēra scenārijs	Sagaidāmais rezultāts	Rezultāts	
1.	FIM01	Tiek nospiesta DIPF izsaukšanas poga darba procesu/uzdevumu sarakstā. DIPF šobrīd nav atvērta.	Tiek atvērta DIPF darba procesam/uzdevumam.	OK, 02.05.2018	OK, 10.05.2018
2.		Tiek nospiesta DIPF izsaukšanas poga zemes vienību sarakstā. DIPF šobrīd nav atvērta.	Tiek atvērta DIPF zemes vienībai.	OK, 02.05.2018	OK, 10.05.2018
3.		Tiek nospiesta DIPF izsaukšanas poga personu sarakstā. DIPF šobrīd nav atvērta.	Tiek atvērta DIPF personai.	OK, 02.05.2018	OK, 10.05.2018
4.		Tiek nospiesta DIPF izsaukšanas poga ēku sarakstā.	DIPF vietā tiek atvērts līdz šim pastāvošais risinājums - forma “Izmaiņas ēkai”, jo šim objekta tipam datu izmaiņu pārlūkošana nav iespējota.	OK, 02.05.2018	OK, 10.05.2018

Nr.	PPS funkcija	Testpiemēra scenārijs	Sagaidāmais rezultāts	Rezultāts	
5.		Tiek nospiesta DIPF izsaukšanas poga zemes vienību sarakstā. DIPF zemes vienībai jau ir atvērta un atrodas zem vairākām citām dotajā brīdī atvērtām formām.	DIPF zemes vienībai tiek novietota priekšpusē virs visām pārējām dotajā brīdī atvērtajām formām.	KĻŪDA (02.05.2018) – tiek atvērta vēl viena tās pašas zemes vienības forma.	OK, 10.05.2018
6.		Tiek nospiesta DIPF izsaukšanas poga zemes vienību sarakstā. Dotajā brīdī ir atvērta DIPF personai.	Tiek atvērta papildus vēl viena DIPF zemes vienībai.	OK, 02.05.2018	OK, 10.05.2018
7.	FIM02	DIPF tiek nospiesta poga “Beigt”.	Tiek aizvērta DIPF.	OK, 02.05.2018	OK, 10.05.2018
8.	FIM03	Tiek atvērta DIPF zemes vienībai.	DIPF nosaukumā tiek parādīta zemes vienības adrese, kadastra numurs.	OK, 02.05.2018	OK, 10.05.2018
9.		Tiek atvērta DIPF darba procesam/uzdevumam.	DIPF nosaukumā tiek parādīts darba procesa vai uzdevuma numurs.	OK, 02.05.2018	OK, 10.05.2018
10.		Tiek atvērta DIPF personai.	DIPF nosaukumā tiek parādīts personas vārds un uzvārds vai uzņēmuma nosaukums, personas kods vai uzņēmuma reģistrācijas numurs.	OK, 02.05.2018	OK, 10.05.2018

Nr.	PPS funkcija	Testpiemēra scenārijs	Sagaidāmais rezultāts	Rezultāts	
11.		Tiek atvērta DIPF atviegljumam	DIPF nosaukumā tiek parādīts personas vārds, uzvārds, personas kods un atviegljuma veids.	OK, 02.05.2018	OK, 10.05.2018
12.	FIM04	Tiek atvērta DIPF atviegljumam	Formas kreisajā apakšējā stūrī attēlots nolaižamais saraksts ar atviegljuma saistītajiem sarakstiem.	OK, 02.05.2018	OK, 10.05.2018
13.	FIM05	Tiek atvērta saistītā saraksta “Lēmums NINO” DIPF forma	Tiek atvērta DIPF par atviegljumu piešķiršanu.	OK, 02.05.2018	OK, 10.05.2018
14.		Tiek atvērta saistītā saraksta “Objekti” DIPF forma	Tieka atvērta DIPF par atviegljumu piešķiršanu objektiem.	OK, 02.05.2018	OK, 10.05.2018
15.	FIM06	Tiek atvērta saistītā saraksta “Objekti” DIPF forma	Tiek atvērta DIPF, kurā virs datu izmaiņu vēstures tabulas attēlota grupēšanas tabula.	OK, 02.05.2018	OK, 10.05.2018
16.	MPM01	Tiek atvērta DIPF zemes vienībai.	Attēlojamo un izmaiņu kolonnu sarakstos tiek attēloti 1.pielikumā zemes vienībai noteiktie datu lauki.	OK, 02.05.2018	OK, 10.05.2018
17.		Tiek atvērta DIPF darba procesam/uzdevumam.	Attēlojamo un izmaiņu kolonnu sarakstos tiek attēloti 1.pielikumā darba procesam/uzdevumam noteiktie datu lauki.	OK, 02.05.2018	OK, 10.05.2018

Nr.	PPS funkcija	Testpiemēra scenārijs	Sagaidāmais rezultāts	Rezultāts	
18.		Tiek atvērta DIPF personai.	Attēlojamo un izmaiņu kolonnu sarakstos tiek attēloti 1.pielikumā noteiktie personai noteiktie datu lauki.	OK, 02.05.2018	OK, 10.05.2018
19.	MPM02	Nospiesta jebkura tukša izvēles rūtiņa attēlojamo kolonnu sarakstā.	Nospiestā izvēles rūtiņa tiek atzīmēta.	OK, 02.05.2018	OK, 10.05.2018
20.		Nospiesta izvēles rūtiņa "Spēkā no" attēlojamo kolonnu sarakstā.	Nospiestā rūtiņa paliek atzīmēta.	OK, 02.05.2018	OK, 10.05.2018
21.	MPM02	Nospiesta izvēles rūtiņa "Spēkā līdz" attēlojamo kolonnu sarakstā.	Nospiestā rūtiņa paliek atzīmēta.	OK, 02.05.2018	OK, 10.05.2018
22.		Nospiesta izvēles rūtiņa "Izmaiņu veicējs" attēlojamo kolonnu sarakstā.	Nospiestā rūtiņa paliek atzīmēta.	OK, 02.05.2018	OK, 10.05.2018
23.		Nospiesta jebkura tukša izvēles rūtiņa izmaiņu kolonnu sarakstā.	Tiek atzīmēta nospiestā izvēles rūtiņa, kā arī izvēles rūtiņa ar tādu pašu nosaukumu attēlojamo kolonnu sarakstā.	OK, 02.05.2018	OK, 10.05.2018
24.		Nospiesta jebkura atzīmēta izvēles rūtiņa izmaiņu kolonnu sarakstā.	Atzīme tiek noņemta no nospiestās izvēles rūtiņas, kā arī no izvēles rūtiņas ar tādu pašu nosaukumu attēlojamo kolonnu sarakstā.	OK, 02.05.2018	OK, 10.05.2018
25.	MPM03	Nospiesta poga "Atzīmēt visus" pie attēlojamo kolonnu saraksta.	Tiek atzīmētas visas izvēles rūtiņas attēlojamo kolonnu sarakstā.	OK, 02.05.2018	OK, 10.05.2018

Nr.	PPS funkcija	Testpiemēra scenārijs	Sagaidāmais rezultāts	Rezultāts	
26.	MPM04	Nospiesta poga “Noņemt visus” pie attēlojamo kolonnu saraksta. Izmaiņu kolonnu sarakstā nav atzīmēta neviena izvēles rūtiņa.	Tiek noņemtas visas atzīmes no izvēles rūtiņām attēlojamo kolonnu sarakstā, izņemot “Spēkā no”, “Spēkā līdz”, “Izmaiņu veicējs”.	OK, 02.05.2018	OK, 10.05.2018
27.		Atvērta DIPF zemes vienībai. Nospiesta poga “Noņemt visus” pie attēlojamo kolonnu saraksta. Izmaiņu kolonnu sarakstā atzīmēta rūtiņa “Platība”.	Tiek noņemtas visas atzīmes no izvēles rūtiņām attēlojamo kolonnu sarakstā, izņemot “Spēkā no”, “Spēkā līdz”, “Izmaiņu veicējs”, “Platība”.	OK, 02.05.2018	OK, 10.05.2018
28.	MPM05	Nospiesta poga “Atzīmēt visus” pie izmaiņu kolonnu saraksta.	Tiek atzīmētas visas izvēles rūtiņas izmaiņu kolonnu sarakstā.	OK, 02.05.2018	OK, 10.05.2018
29.	MPM06	Nospiesta poga “Noņemt visus” pie izmaiņu kolonnu saraksta.	Tiek noņemta visas atzīmes no izvēles rūtiņām izmaiņu kolonnu saraksta.	OK, 02.05.2018	OK, 10.05.2018
30.	MPM07	Atvērta DIPF zemes vienībai. Attēlojamo kolonnu sarakstā atzīmēti lauki “Spēkā no”, “Spēkā līdz”, “Izmaiņu veicējs”, “Platība”. Izmaiņu kolonnu sarakstā atzīmēts lauks “Platība”. Nospiesta poga “Attēlot”, pēc tam poga “Beigt”. Atkal tiek atvērta DIPF zemes vienībai.	Attēlojamo kolonnu sarakstā atzīmēti lauki “Spēkā no”, “Spēkā līdz”, “Izmaiņu veicējs”, “Platība”. Izmaiņu kolonnu sarakstā atzīmēts lauks “Platība”.	OK, 02.05.2018	OK, 10.05.2018

Nr.	PPS funkcija	Testpiemēra scenārijs	Sagaidāmais rezultāts	Rezultāts	
31.	DAM01	Atvērta DIPF zemes vienībai. attēlojamo kolonnu sarakstā atzīmēti lauki “Spēkā no”, “Spēkā līdz”, “Izmaiņu veicējs”, “Platība”. Izmaiņu kolonnu sarakstā netiek atzīmēts neviens datu lauks. Nospiesta poga “Attēlot”.	Tiek parādīts P1.	OK, 02.05.2018	OK, 10.05.2018
32.		Atvērta DIPF zemes vienībai. attēlojamo kolonnu sarakstā atzīmēti lauki “Spēkā no”, “Spēkā līdz”, “Izmaiņu veicējs”, “Platība”. Izmaiņu kolonnu sarakstā atzīmēts lauks “Platība”. Nospiesta poga “Attēlot”.	Datu tabulā tiek attēlotas kolonnas “Spēkā no”, “Spēkā līdz”, “Izmaiņu veicējs”, “Platība” un izmaiņu vēsture datu laukam “Platība”.	OK, 02.05.2018	OK, 10.05.2018
33.		Atvērta DIPF atvieglojumu reģistra saistītajam sarakstam “Objekti”. Grupēšanas tabulā veikti peles klikšķi uz ierakstiem.	Datu izmaiņu tabulā pēc katra peles klikšķa parādās ieraksti par iezīmēto grupēšanas tabulas ierakstu.	KĻŪDA (02.05.2018) – korekti dati attēloti tikai pēc diviem izdarītiem klikšķiem	OK, 10.05.2018
34.	DAM01	27. testpiemēra scenārijs ar dažādām atzīmēto izvēles rūtiņu kombinācijām	Datu tabulā tiek attēlotas kolonnas ar tādiem nosaukumiem, kādi atzīmēti	OK, 02.05.2018	OK, 10.05.2018

Nr.	PPS funkcija	Testpiemēra scenārijs	Sagaidāmais rezultāts	Rezultāts	
		attēlojamo un izmaiņu kolonnu sarakstos. Izmaiņu kolonnu sarakstā vienmēr jābūt atzīmētai vismaz vienai rūtiņai.	attēlojamo kolonnu sarakstā, un izmaiņu vēsture datu laukiem, kuri atzīmēti izmaiņu kolonnu sarakstā.		
35.	DAM02	27.testiemēra scenārijs, pēc tam nospiesta poga "Drukāt".	Tiek atvērta Microsoft Excel darbgrāmata, kurā parādīti objektu identificējoši dati, ievadītie meklēšanas parametri un datu tabula, kurā attēlotas kolonnas ar tādiem nosaukumiem, kādi atzīmēti attēlojamo kolonnu sarakstā, un izmaiņu vēsture datu laukiem, kuri atzīmēti izmaiņu kolonnu sarakstā.	OK, 02.05.2018	OK, 10.05.2018

3.4.2. Augsta līmeņa funkcionālā testēšana

Augsta līmeņa funkcionālajā testēšanā par pamatu ņemti testpiemēru scenāriji, kas līdzinās reālām lietotāja darbībām datu izmaiņu pārlūkošanas moduļa formā (skat. tabulā 3.2.).

3.2. tabula

Augsta līmeņa funkcionālās testēšanas žurnāls

Nr.	Testpiemēra apraksts	Testpiemēru scenāriji	Sagaidāmais rezultāts	Rezultāts	
36.	Pirmo reizi atverot objekta tipam atbilstošo DIPF, visi	Tiek atvērta DIPF no lietotāja, kurš vēl ne reizi nav pārlūkojis	Tiek atvērta DIPF, kur attēlojamo un izmaiņu kolonnu	OK, 02.05.2018	OK, 10.05.2018

Nr.	Testpiemēra apraksts	Testpiemēru scenāriji	Sagaidāmais rezultāts	Rezultāts	
	datu lauki attēlojamo un izmaiņu kolonnu sarakstos ir atzīmēti un tiek attēlota visu pieejamo (skat. 1.pielikumu) datu lauku izmaiņu vēsture.	datu lauku izmaiņu vēsturi zemes vienībai.	sarakstos attēloti nepieciešamie lauki (skat. 1.pielikumā) un tie visi ir atzīmēti. Datu tabulā tiek attēlotas visas iespējamās kolonnas un visu iespējamo datu lauku izmaiņu vēsture.		
37.	Lietotāja meklēšanas parametros netiek saglabāti parametri, ja izmaiņu kolonnu sarakstā nav atzīmēts neviens datu lauks.	DIPF attēlojamo kolonnu sarakstā netiek atzīmēts neviens lauks. Nospiesta poga “Attēlot”, aizvērts P1 un DIPF. DIPF šim pašam objektam atvērta no jauna.	Izmaiņu kolonnu sarakstā atzīmēti datu lauki, kuri bijuši atzīmēti pēdējā veiksmīgajā datu lauku izmaiņu vēstures pārlūkošanas reizē.	OK, 02.05.2018	OK, 10.05.2018
38.	Nav iespējams noņemt atzīmi attēlojamo kolonnu sarakstā, ja tas atzīmēts izmaiņu kolonnu sarakstā.	Nospiesta poga “Atzīmēt visus” izmaiņu kolonnu sarakstā, pēc tam nospiesta poga “Noņemt visus” attēlojamo kolonnu sarakstā.	Attēlojamo kolonnu sarakstā paliek atzīmēti visi lauki no izmaiņu kolonnu saraksta un lauki “Spēkā no”, “Spēkā līdz”, “Izmaiņu veicējs”	KĻŪDA (02.05.2018) – tiek noņemtas visas atzīmes attēlojamo kolonnu sarakstā.	OK, 10.05.2018
39.	Meklēšanas parametri tiek saglabāti tikai līdz ar pogas “Attēlot” nospiešanu.	Tiek atzīmēti jebkādi meklēšanas parametri, kas atšķirīgi no esošajiem un	Tiek atzīmētas no testpiemēra scenārija izpildes laikā	OK, 02.05.2018	OK, 10.05.2018

Nr.	Testpiemēra apraksts	Testpiemēru scenāriji	Sagaidāmais rezultāts	Rezultāts	
		neizraisa P1. Nospiesta poga “Beigt”. Pēc tam DIPF atvērta no jauna tam pašam objekta tipam.	atzīmētajiem datu laukiem atšķirīgas izvēles rūtiņas.		
40.	Uz lietotni Microsoft Excel tiek eksportēti dati, kas redzami datu tabulā uz ekrāna, nevis dati pēc meklēšanas parametru atzīmēšanas.	Tiek atvērta DIPF un nospiesta poga “Attēlot”. Pēc tam atzīmēti no esošajiem atšķirīgi meklēšanas parametri un nospiesta poga “Drukāt”.	Lietotnes Microsoft Excel darbgrāmatā tiek attēloti dati no datu tabulas uz ekrāna, kas iegūta pēc pogas “Attēlot” nospiešanas.	KĻŪDA (02.05.2018) – Microsoft Excel tiek attēloti pareizie dati bet nepareizi meklēšanas parametri.	OK, 10.05.2018
41.	Uz lietotni Microsoft Excel iespējams eksportēt tikai aizpildītu datu izmaiņu vēstures tabulu.	DIPF attēlojamo kolonnu sarakstā netiek atzīmēts neviens lauks. Nospiesta poga “Attēlot”, aizvērts P1. Nospiesta poga drukāt.	Nekas nenotiek - poga ir atspējota, jo nav datu, ko parādīt.	KĻŪDA (02.05.2018) – tiek parādīts kļūdas paziņojums un atvērts Microsoft Excel fails.	OK, 10.05.2018
42.	Ja ir atvērta DIPF forma atvieglājumu reģistra	Tiek atvērta DIPF atvieglājumu reģistra ierakstam	Netiek atvērta neviena jauna forma, bet notiek navigēšana	OK, 02.05.2018	OK, 10.05.2018

Nr.	Testpiemēra apraksts	Testpiemēru scenāriji	Sagaidāmais rezultāts	Rezultāts	
	ierakstam, kā arī DIPF formas tā saistītajiem sarakstiem, iespējams navigēt starp tām, izmantojot nolaižamo sarakstu un neviena no formām netiek atvērta no jauna.	un tā saistītajiem sarakstiem (“Objekti” un “Lēmums NINO”). Katrā no formām ar nolaižamā saraksta palīdzību tiek atvērtas formas vēlreiz.	starp 3 dotajā brīdī atvērtajām DIPF formām.		

3.5. Testēšanas kļūdu apstrāde

Testēšanas kļūdu apstrādē (skat. tabulā 3.3.) aprakstīts testpiemēra izgāšanās iemesls, kā šī kļūda izlabota, kā arī veikta atkārtota testēšana un atzīmēts rezultāts.

3.3. tabula

Testēšanas kļūdu apstrāde

Testpiemēra Nr.	Kļūdas cēlonis	Labojums	Testpiemēra izpildes rezultāts
5.	Formas atzīme (angļu val. <i>tag</i>) tiek izveidots kļūdaini, līdz ar to procedūra <i>ir_forma</i> nenostādā pareizi.	Izvēlēts vienots formas atzīmes veidošanas stils arī saistīto sarakstu objektiem.	OK, 10.05.2018
33.	Tiek izmantota procedūra <i>tdbResult_Click</i> , kas nostrādā uz klikšķa brīdi, tātad pirms jaunā ieraksta atlasīšanas, kā rezultātā tiek atlasīta datu izmaiņu vēsture par iepriekš iezīmēto ierakstu.	Procedūras <i>tdbResult_Click</i> vietā izmantota procedūra <i>tdbResult_RowColChange</i> , kas nostrādā uz iezīmētā ieraksta maiņas brīdi, līdz ar to atlasa datus par tikko iezīmēto ierakstu.	OK, 10.05.2018
38.	Pārbaude uz attēlojamo datu lauku nosaukumiem ieviesta tikai pie konkrēta datu lauka atzīmēšanas.	Pie pogas piespiešanas ieviesta pārbaude uz attēlojamo datu lauku nosaukumiem.	OK, 10.05.2018
40.	Microsoft Excel attēlotie meklēšanas parametri iegūti no izmaiņu kolonnu saraksta, kurš starp pogas "Attēlot" un "Drukāt" nospiešanu var mainīties, līdz ar to tiek attēloti	Microsoft Excel attēloti meklēšanas parametri no lietotāja meklēšanas parametru konfigurācijas, kuriem parametrs " <i>Comparable</i> " uzstādīts uz " <i>true</i> ".	OK, 10.05.2018

	nepareizi meklēšanas parametri		
41.	Līdz ar kļūdas paziņojuma atvēršanu, tiek nodzēsti dati no datu lauku izmaiņu vēstures tabulas, bet, tā kā Microsoft Excel datus ņem no tās, tiek parādīts kļūdas paziņojums.	Pēc kļūdas paziņojuma parādīšanas atspējota poga “Drukāt”, kas vēlāk tiek iespējota tikai pēc veiksmīgas datu atlases.	OK, 10.05.2018

3.6. Testēšanas kopsavilkuma pārskats

Kopumā testēšana noritējusi veiksmīgi. Visas testēšanas laikā atrastās kļūdas izlabotas un pēc atkārtotas testēšanas vairs netiek novērotas. Apliecinājums par prasību testu noklājumu - tabulas 3.1 kolonnā “PPS funkcijas identifikators” – testēšanas žurnālā iekļautas visas PPS minētās funkcijas.

Vienībtestēšanas laikā novērotās svārstības pārsvarā pamatojamas ar izmaiņām galvenajā datu atlases vaicājumā (piemēram, datu atlases vaicājumam pievienojot vai no tā noņemot atlasāmos datu laukus). Konkrētajā gadījumā vienībtestēšana palīdzējusi izvairīties no loģikas kļūdām, kas ietekmē ierakstu secību un atlasāmo datu lauku skaitu, kā arī ievērojami uzlabojusi programmkoda kvalitāti. Ja testēšanas būvējumā visi (vai lielākā daļa) testpiemēri izrādījās neizdevušies, problēmas cēlonis bija saistīts ar kopējās sistēmas darbību.

Funkcionālā testēšana nesusi būtiskus rezultātus, izpildot augsta līmeņa testpiemērus, jo pieredzes trūkuma darbā ar NINO dēļ darba autoram bijis sarežģīti paredzēt, kā pēc vairāku soļu izpildes darbojas izstrādes vidē iebūvētās vai NINO jau realizētās procedūras.

4. PROJEKTA ORGANIZĀCIJA

Datu izmaiņu pārlūkošanas moduļa programmaprodukta izstrāde ietilpst NINO attīstības projektā, kura pasūtītāji ir Rīgas domes Finanšu departaments, bet lietotāji ir Rīgas domes Pašvaldības ieņēmumu pārvaldes darbinieki. Par programmkoda un dokumentācijas izstrādi atbildīgs darba autors, taču nepārtraukti notikušas konsultācijas arī ar pārējiem NINO izstrādes komandas dalībniekiem. Galvenokārt konsultācijas notikušas sarunu ceļā, bet dažkārt arī ar e-pastu starpniecību vai izmantojot ZZ Dats iekšējās saziņas rīku Mattermost. Failu apmaiņai izmantots ZZ Dats esošais failu apmaiņas serveris.

Datu izmaiņu pārlūkošanas moduļa izstrādes gaita vadīta pēc ūdenskrituma dzīves cikla modeļa, kas nosaka darba dalīšanu pa fāzēm ar atgriezenisko saiti uz iepriekšējo fāzi. Vispirms no NINO komandas puses tika veikta sistēmanalīze un formulēta datu izmaiņu pārlūkošanas moduļa galvenā funkcionalitāte. Pēc tam no darba autora puses veikta prasību apkopošana (PPS) un projektējuma izstrāde (PPA). Daļu projektējuma (augsta līmeņa datu projektējums) darba autors izstrādājis prasību apkopošanas laikā, lai iegūtu skaidrību par sistēmā izmantojamajiem datu objektiem un ar tiem saistītās funkcionalitātes ieviešanu. Pēc PPS izstrādes veikta programmatūras projektēšana, kuras laikā ar NINO komandas palīdzību noteikti galvenie principi, uz kuriem tiks balstīta datu izmaiņu pārlūkošanas moduļa funkcionalitāte. Projektējums apkopots PPA dokumentā. Pēc tam autors izstrādājis programmkodu. Darba autors programmkoda izstrādes laikā veicis vienībtestēšanu un pēc izstrādes beigām – PPS funkciju un augsta līmeņa funkcionālo testēšanu. Testēšanas plāns, rezultāti un kļūdu apraksti ar labojumiem apkopoti testēšanas dokumentācijā. Programmkoda izstrādes laikā veiktās izmaiņas regulāri saglabātas versiju kontroles sistēmā Git.

Izstrādāto datu izmaiņu pārlūkošanas moduli paredzēts ieviest produkcijā kopā ar pārējiem 2018.gada NINO attīstības projekta papildinājumiem, kam nodošanas termiņš ir 2018.gada 30.novembris.

5. KVALITĀTES NODROŠINĀŠANA

Izstrādātā programmaprodukta kvalitātes nodrošināšanai ievēroti vairāki standarti, veikta labās programmēšanas prakses izpēte un ievērošana, kā arī saņemti ieteikumi no NINO komandas darbiniekiem.

Programmaprodukta dokumentācijas izstrādē ievēroti sekojoši valsts standarti programminženierijā:

- LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis” [8];
- LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai” [9];
- LVS 70:1996 “Programmatūras testēšanas dokumentācija” [10].

Izstrādātā dokumentācija un ar to saistītie faili glabāti, izmantojot servisu OneDrive, kas nodrošina piekļuvi dokumentam no jebkuras ierīces, izmantojot Microsoft Office 365 kontu, kā arī automātiski veic dokumenta versionēšanu, sniedzot iespēju piekļūt un atjaunot vecākas dokumentācijas versijas. Jaunākā dokumentācijas versija regulāri saglabāta arī lokāli datorā, kā arī ZZ Dats esošajā failu apmaiņas serverī, lai nepieciešamības gadījumā dokumentu varētu apskatīt pārējie ZZ Dats darbinieki.

Kvalitatīva programmkoda izstrādei galvenokārt tika izmantotas konsultācijas ar NINO programmētājiem, lai nodrošinātu ar pastāvošo sistēmu vienotu programmēšanas stilu un sistēmas kopējās saskarnes dizaina vadlīniju ievērošanu. Lai novērstu koda dublēšanos, galvenās procedūras pēc iespējas tikai veidotas universālas, kā arī funkcionalitātes nodrošināšanai pēc nepieciešamības izmantoti jau pastāvošie komandas izstrādātie risinājumi (piemēram, JSON formāta failu veidošana un nolasīšana Oracle PL/SQL Developer un Microsoft Visual Basic izstrādes vidēs) (skat. nodaļu 2.6.2.7). Programmkods tika regulāri komentēts latviešu valodā, lai padarītu to vieglāk lasāmu, saprotamu un modificējamu. Programmkoda noformējums Microsoft Visual Basic 6 izstrādes vidē nodrošināts, izmantojot iebūvēto IntelliSense rīku, bet Oracle PL/SQL Developer - izmantojot PL/SQL Beautifier rīku. Šie rīki palīdz automātiski sakārtot programmkodu pēc labā programmēšanas stila (ievietojot nepieciešamās atstarpes, pāriešanu jaunā rindā, Microsoft Visual Basic 6 vidē uzreiz tiek signalizēts arī par daļu sintakses kļūdām), kas palīdz uzlabot koda lasāmību un novērst pārskatīšanās kļūdas. Pēc lielāku programmkoda daļu izstrādes veikta koda reorganizācija (angļu val. “*code refactoring*”), lai uzlabotu koda lasāmību un padarītu kodu vienkāršāku, strukturētāku. Lai izslēgtu SQL injekciju iespējamību, izmantoti saistītie mainīgie (angļu val. “*bind variables*”). Veikta nepieciešamā kļūdu apstrāde pēc projektā pastāvošajām kļūdu apstrādes vadlīnijām.

Programmkoda kvalitātes nodrošināšanai veikta arī vienībtestēšana, izmantojot automatizēto vienībtestēšanas ietvaru *utPLSQL* un *Jenkins* automatizācijas platformu. Par neizpildītiem testiem saņemti paziņojumi e-pastā, kas ļauj nekavējoties apzināt kļūdu rašanās iemeslus un izlabot tās. Pēc programmkoda izstrādes veikta funkcionālā testēšana. Testēšanas plāns, testēšanas žurnāls un rezultāti piefiksēti testēšanas dokumentācijā.

Informācijas izpēte par ZZ Dats programmkoda kvalitātes nodrošināšanas priekšrakstiem tika veikta arī uzņēmuma izstrādātajā vietnē “ZZ Dats vikipēdija” [2] un koplietošanas failos.

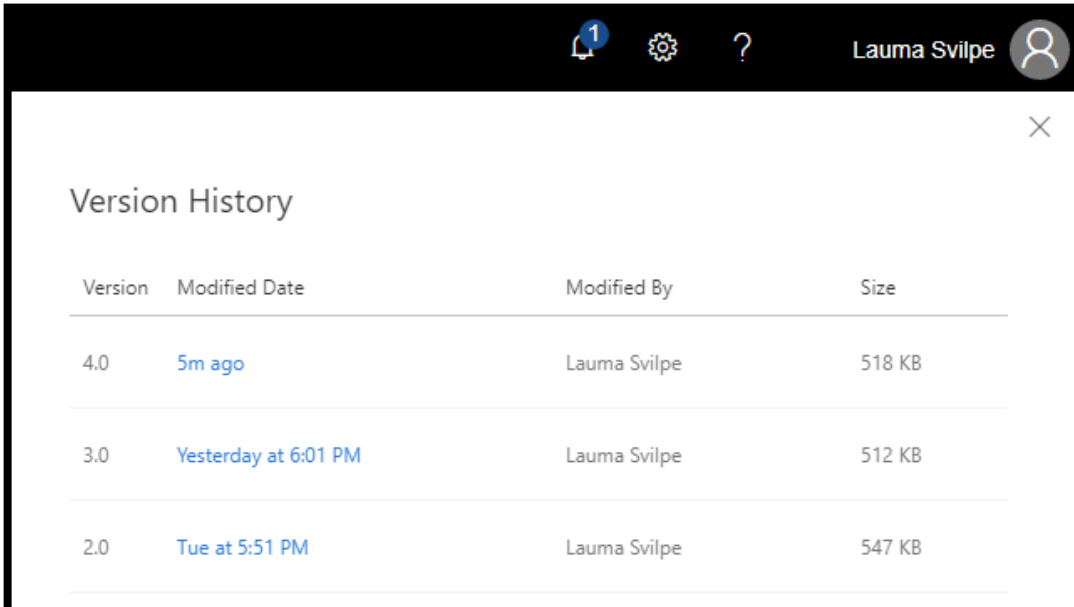
Oracle datu bāzes pakotnes pēc jaunas un strādājošas funkcionalitātes ieviešanas vai uzlabojumu veikšanas regulāri tika saglabātas versiju kontroles rīkā Git ar paskaidrojošiem komentāriem par to, kas esošajā versijā izmainīts.

Dokumentā izmantotas trasējamības tabulas, kas ļauj izsekot, kādas PPS minētās prasības ietvaros izstrādāta konkrētā programmkoda daļa.

6. KONFIGURĀCIJU PĀRVALDĪBA

6.1. Dokumentācijas konfigurāciju pārvaldība

Dokumentācijas versiju kontrole tika veikta, izmantojot failu uzglabāšanas un apmaiņas servisu OneDrive, kurā regulāri saglabāts dokumentācijas teksta fails un tā uzlabojumi (jaunākā dokumentācijas versija katras dienas beigās vai pēc nepieciešamības tika augšupielādēta OneDrive servisā, taču dienas laikā veiktās izmaiņas tiek glabātas lokāli uz datora). OneDrive serviss nodrošina automātisku versionēšanu, līdz ar to iespējams apskatīt jebkuru no iepriekšējām dokumenta versijām (skat. att. 6.1.), kā arī, izmantojot Microsoft Office 365 kontu, iespējams piekļūt dokumentiem no jebkuras ierīces.



The screenshot shows the 'Version History' window in OneDrive. At the top, there is a header bar with a notification icon (1), a settings gear, a help question mark, and the user's name 'Lauma Svilpe' next to a profile picture. The main content area has a title 'Version History' and a close button (X). Below the title is a table with four columns: 'Version', 'Modified Date', 'Modified By', and 'Size'. The table lists three versions of a document.

Version	Modified Date	Modified By	Size
4.0	5m ago	Lauma Svilpe	518 KB
3.0	Yesterday at 6:01 PM	Lauma Svilpe	512 KB
2.0	Tue at 5:51 PM	Lauma Svilpe	547 KB


6.1. att. Dokumenta versionēšana failu uzglabāšanas un apmaiņas vietnē OneDrive

6.2. Programmkoda konfigurācijas pārvaldība

Programmkoda versionēšana tika realizēta, balstoties uz uzņēmumā un NINO projektā pastāvošajām vadlīnijām. Datu bāzes pakotnes regulāri pēc katras funkcionalitātes vai uzlabojumu veikšanas tika saglabātas versiju kontroles sistēmā *Git* projekta repozitorijā “*database/nino*”, komentējot kā un kādas pakotnes ir izmainītas (skat. att. 6.2.).

DATA_CHANGES_BROWSER_CONF_PCK sagatavo daļēji aizpildītu json konfigurāciju.

DATA_CHANGES_BROWSER_PCK ja nav ievadīts pirmais izmaiņu veicējs, tad raksta kā 'NINO'.

 **LaumaS** <lauma.svilpe@zzdats.lv> 2 weeks ago parent [f6658861de](#) commit [ac475351c1](#) [Browse Source](#)

2 changed files with 206 additions and 122 deletions [Split View](#) [Show Diff Stats](#)

6.2. att. Datu bāzes pakotnes saglabāšana versiju kontroles sistēmā Git

Ērtākai izmaiņu saglabāšanai un citu izstrādātāju veikto izmaiņu iegūšanai tiek izmantots rīks *TortoiseGit*. Pirms katru izmaiņu iesūtīšanas versiju kontroles sistēmā ar komandas *Pull* palīdzību tiek iegūta jaunākā programmkoda versija, pēc kā seko savu izmaiņu iesūtīšana (skat. att. 6.3.).

C:\DARBS\git.zzdats.lv\database\nino\code\zr\CUKA - Commit - TortoiseGit

Commit to: master ☐ new branch

Message:

DATA_CHANGES_BROWSER_PCK - sakārtots kods, procedūra get_columns pārvietota uz DCB_USER_SETTINGS_PCK
DATA_CHANGES_BROWSER_CONF_PCK - pievienota kļūdu apstrāde

☐ Amend Last Commit ☐ Show diff to last commit 3/1

☐ Set author date

☐ Set author [Add Signed-off-by](#)

Changes made (double-click on file for diff):

Check: **All** None Unversioned **Versioned** Added Deleted **Modified** Files Submodules

Path	Extension	Status	Lines added	Lines removed
<input checked="" type="checkbox"/> code/zr/CUKA/DATA_CHANGES_BROWSER_CONF_PCK.pck	.pck	Modified	8	4
<input checked="" type="checkbox"/> code/zr/CUKA/DATA_CHANGES_BROWSER_PCK.pck	.pck	Modified	210	167
<input checked="" type="checkbox"/> code/zr/CUKA/DCB_USER_SETTINGS_PCK.pck	.pck	Modified	27	1

☒ Show Unversioned Files 3 files selected, 3 files total

☐ Do not autoselect submodules [View Patch>>](#)

☐ Show Whole Project

☐ Message only

[Commit & Push](#) [Cancel](#) [Help](#)

6.3. att. Datu bāzes pakotnes saglabāšana versiju kontroles sistēmā Git, izmantojot TortoiseGit rīku

7. DARBIETILPĪBAS NOVĒRTĒJUMS

Kvalifikācijas darba ietvaros izstrādātas aptuveni 1800 programmkoda rindiņas Oracle PL/SQL Developer izstrādes vidē un aptuveni 1000 - Microsoft Visual Basic 6.0 izstrādes vidē, kas kopumā viedo 2800 programmkoda rindiņas (no tām apmēram trešdaļa – automatizētās vienībtestēšanas programmkods). Tabulā DATA_CHANGES_BROWSER izveidoti 6 ieraksti, kur JSON formāta tabulu konfigurāciju failos kopumā saglabātas aptuveni 850 konfigurācijas rindiņas.

Darbietilpības novērtējumā izmantota ekspertu metode, kurā ņemts vērā kvalifikācijas darba autora viedoklis, kā arī speciālista viedoklis no NINO izstrādātāju komandas par katras kvalifikācijas darba daļas izpildes laiku. Veicot darbietilpības novērtēšanu, ņemts vērā, ka darba autoram nav bijusi pieredze darbā ar rīkiem, tehnoloģijām un programmēšanas valodām, kas tiek izmantotas NINO programmkoda izstrādē. Pēc kvalifikācijas darba izstrādes veikts aptuvenš novērtējums par reālo darba izpildes laiku. Iegūtos datus skat. tabulā 7.1. Darbietilpības novērtējums veikts stundās.

7.1. tabula

Darbietilpības novērtējums

Daļas nosaukums	Darba autors (h)	Speciālists (h)	Vidējais novērtējums (h)	Patērētais laiks (h)
Iepazīšanās ar izstrādes vidēm, rīkiem, tehnoloģijām	80	70	75	78
PPS	90	95	92,5	80
PPA	100	115	107,5	104
Datu bāzes tabulu izveidošana un JSON konfigurāciju sastādīšana	8	6	7	10
FIM01	5	4	4,5	6
FIM02	1	1	1	0,25
FIM03	3	3	3	3,5
FIM04	2	2	2	2
FIM05	4	3	3,5	8
FIM06	2	2	2	3
MPM01	7	8	7,5	10

MPM02	1	1	1	1
MPM03	1	1	1	0,25
MPM04	2	3	2,5	2
MPM05	1	1	1	0,25
MPM06	1	1	1	0,25
MPM07	10	12	11	8
DAM01	70	74	72	104
DAM02	20	18	19	32
Testēšana	40	45	42,5	40
Lietotāja saskarnes izstrāde	20	15	17,5	16
Kvalifikācijas darba pārējās nodaļas	20	25	22,5	20
Kopā	488	505	496,5	528,5

Speciālista plānotais darbietilpības novērtējums (505 stundas) izrādījās tuvāks reāli patērētajam laikam, kas skaidrojams ar speciālista pieredzi darbā ar NINO izstrādi. Vairākas kvalifikācijas darba daļas tika izstrādātas ātrāk, nekā tas bija plānots, piemēram, dokumentācijas izstrāde, dažu pogu funkcionalitāte, tomēr darba galvenajai daļai – datu atlasē vaicājumam - tika patērēts krietni vairāk laika. Tas skaidrojams ar līdzšinējās pieredzes trūkumu JSON konfigurāciju izmantošanā un PL/SQL dinamisko vaicājumu būvēšanā.

Kvalifikācijas darba darbietilpības vidējais plānotais novērtējums ir 496,5 stundas, bet reāli patērētais laiks – aptuveni 528,5 stundas. Nesakritība saistāma ar to, ka nav iespējams precīzi novērtēt pieredzes un zināšanu līmeni, kas būs nepieciešamas, izstrādājot darbu.

Ņemot vērā, ka viena cilvēkdiena ir vienāda ar 8 nostrādātām cilvēkstundām, cilvēkmēnešu skaitu iespējams iegūt divos veidos:

- 1 cilvēkmēnesis = 20 cilvēkdienas = 160 cilvēkstundas, ņemot vērā vidējo darba dienu skaitu kalendārajā mēnesī;
- 1 cilvēkmēnesis = 17,5 cilvēkdienas = 140 cilvēkstundas, ņemot vērā cilvēka produktivitāti, iespējamās svētku dienas, pusdienu pārtraukums, nelielus atslodzes pārtraukumus u.c. attaisnojošus iemeslus [3].

Veicot aprēķinus, iegūti sekojoši rezultāti - pirmajā gadījumā darbietilpības novērtējums vienāds ar 3,3 cilvēkmēnešiem, bet otrajā – 3,78 cilvēkmēnešiem, abos gadījumos pārsniedzot 3 cilvēkmēnešu darbietilpību. Vērtējot pēc reāli patērēto mēnešu skaita, lielāka sakritība vērojama, pieņemot, ka mēnesī ietilpināmas 17,5 cilvēkdienas, jo kvalifikācijas darba izstrāde

notikusi studiju kursā paredzētās prakses ietvaros (prakses sākuma datums - 2018.gada 29.janvāris), ievērojot noteikto darba ilgumu (40 stundas nedēļā), kā arī Latvijā noteiktās svētku dienas - brīvdienas.

REZULTĀTI

Kvalifikācijas darbā izvirzītais mērķis ir sasniegts - izstrādāts risinājums, kas atvieglotu NINO konsultantu darbu un nodrošinātu iespēju NINO lietotājiem pārlūkot sistēmā reģistrēto datu izmaiņu vēsturi, lai noskaidrotu kā, kad un kurš lietotājs ir mainījis datus. Ir izpildīti mērķa sasniegšanai pakārtotie uzdevumi – izstrādāta gan programmaprodukta dokumentācija, gan strādājošs programmkods (programmkoda fragmentus skat. 5. un 6. pielikumā). Dokumentācijā iekļauti PPS, PPA un testēšanas dokumentācija, kā arī aprakstīta projekta organizācija, kvalitātes nodrošināšana, konfigurāciju pārvaldība un veikts darbietilpības novērtējums. Izstrādātais risinājums veidots pēc iespējas universāls, lai jaunu objekta tipu pievienošana izmaiņu pārlūkošanas modulim aizņemtu pēc iespējas mazāk laika un resursu.

Vadoties pēc PPS prasībām un PPA projektējuma, izstrādāta atbilstoša lietotāja saskarne (ekrānattēlus skat. 7. pielikumā).

SECINĀJUMI

Izstrādājot kvalifikācijas darbu, tika iegūta pieredze darbā ar Oracle PL/SQL Developer un Microsoft Visual Basic izstrādes vidēm, kā arī PL/SQL un Visual Basic programmēšanas valodām. Tika iegūtas zināšanas par JSON konfigurāciju failu veidošanu un to plašajiem pielietojumu veidiem.

Kvalifikācijas darba izstrādes laikā gūtas vairākas atziņas par programmprodukta izstrādes procesu:

- Programmēšanas sākumā pats galvenais ir pilnībā izprast gan prasības, gan esošā risinājuma darbību, lai nerastos situācijas, kad kāda no uzprogrammētajām daļām ir lieka vai pat negatīvi ietekmē sistēmas darbību kopumā;
- Lai arī programmēšana pārsvarā notiek individuāli katram pie savas darbstacijas, ļoti svarīga ir komunikācija starp projekta komandas dalībniekiem, lai ieklausītos citu programmētāju pieredzē, tādējādi atrodot vislabāko problēmas risinājumu;
- Programmējot vienmēr jācenšas meklēt universāli problēmas risinājumi;
- Programmprodukta dokumentācijas izstrādes laikā pieturēties pie konsekventiem darba sākumā definētiem izmantojamajiem terminiem, lai prasības vai projektējums netiktu pārprasts;
- Komentāri programmkodā ievērojami palīdz ne tikai izprast programmas darbību uzturētājam, kurš pats to nav programmējis, bet arī pašam tās izstrādātājam, pēc laika veicot kādus uzlabojumus;
- Svarīgi (it īpaši programmētājam ar nelielu darba pieredzi) ir regulāri veikt programmkoda reorganizāciju (angļu val. “*refactoring*”), lai padarītu to vieglāk lasāmu, uzlabotu ātrdarbību un pārbaudītu algoritmu efektivitāti;
- Regulāra vienībtestu izstrāde un darbināšana programmēšanas laikā palīdz izvairīties no neparedzētu kļūdu rašanās pirms tās ir būtiski ietekmējušas sistēmas darbību kopumā.

Kopumā pēc kvalifikācijas darba izstrādes secinu, ka, no vienas puses, izstrādāt programmproduktu jau pastāvoša projekta ietvaros ir daudz sarežģītāk, nekā veidojot projektu individuāli no jauna, jo pastāvošā un stabilā projektā tiek izmantoti dažādi gatavi risinājumi, kā arī jāievēro noteiktie standarti, tāpēc, ņemot vērā, ka iepriekšējā pieredze darbā ar šo projektu un tehnoloģijām nav bijusi, ļoti daudz laika tiek pavadīts, analizējot pastāvošos risinājumus. Taču no otras puses, ir iespējams iegūt vērtīgas pieredzējušu programmētāju konsultācijas par efektīvāko risinājumu izvēli, kas nebūtu iespējams individuālā projektā.

IZMANTOTIE LITERATŪRAS AVOTI

1. Akadēmiskā terminu datubāze [tiešsaiste] – [atsauce 09.05.2018.] Pieejams internetā: <http://termini.lza.lv/term.php?>
2. ZZ Dats vikipēdija [tiešsaiste] – [atsauce 09.05.2018.] Pieejams internetā ar autentificēšanos ZZ Dats darbiniekiem: <http://wiki.zzdats.lv/>
3. Darbietilpības novērtējums [tiešsaiste] – [atsauce 09.05.2018.] Pieejams internetā: <http://www.finance-helpdesk.org/front/ShowArticle.aspx?ItemID=662>
4. NINO Programmatūras prasību specifikācija – pieejams tikai ZZ Dats darbiniekiem
5. Vienībtestēšanas ietvara utPLSQL dokumentācija [tiešsaiste] – [atsauce 16.05.2018.] Pieejams internetā: <http://utplsql.org/documentation/>
6. Oracle Database 12c dokumentācija [tiešsaiste] – [atsauce 16.05.2018.] Pieejams internetā: <https://docs.oracle.com/database/121/index.html>
7. Oracle PL/SQL tiešsaistes kursi [tiešsaiste] – [atsauce 16.05.2018.] Pieejams internetā: <https://www.pluralsight.com/>
8. Informācijas tehnoloģijas Standartizācijas tehniskā komiteja 1996. LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis”.
9. Informācijas tehnoloģijas Standartizācijas tehniskā komiteja 1996. LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai”;
10. Informācijas tehnoloģijas Standartizācijas tehniskā komiteja 1996. LVS 70:1996 “Programmatūras testēšanas dokumentācija” .

PIELIKUMS

1.pielikums

Informācija par datu izmaiņu pārlūkošanas moduli iekļaujamajiem datu laukiem

Datu lauka nosaukums	Vai iespējams pārlūkot datu lauka izmaiņas	Obligāts
TASKS		
Apraksts	Jā	Nē
Atbildīgais lietotājs	Jā	Nē
Izmaiņu veicējs	Nē	Jā
Izpildes statuss	Jā	Nē
Spēkā no	Nē	Jā
Spēkā līdz	Nē	Jā
PARCELES		
Adrese	Jā	Nē
Izmaiņu veicējs	Nē	Jā
Kopējā kadastrālā vērtība	Jā	Nē
Kopējās kadastrālās vērtības valūta	Nē	Nē
Nodokļu kadastrālā vērtība	Jā	Nē
Nodokļu kadastrālās vērtības valūta	Nē	Nē
Piezīmes	Jā	Nē
Platība	Jā	Nē
Spēkā no	Nē	Jā
Spēkā līdz	Nē	Jā
PERSONAS_NOD		
Adrese	Jā	Nē
Citi vārdi	Jā	Nē
E-pasts	Jā	Nē
Izmaiņu veicējs	Nē	Jā
Miršanas datums	Jā	Nē
Nosaukums	Jā	Nē
Personas kods	Jā	Nē
Piezīmes	Jā	Nē
Pilsēta, pasta indekss	Jā	Nē
Reģistrācijas numurs	Jā	Nē
Statuss	Jā	Nē
Spēkā no	Nē	Jā
Spēkā līdz	Nē	Jā
Tālrunis	Jā	Nē
Uzvārds	Jā	Nē
Vārds	Jā	Nē
ATV_REG		
Atvieglojuma perioda beigu datums	Jā	Nē
Atvieglojuma perioda sākuma datums	Jā	Nē
Atvieglojuma veids	Jā	Nē
Datums, kurā ieraksts reģistrēts sistēmā	Nē	Nē

Izmaiņu veicējs	Nē	Jā
Lēmuma datums	Nē	Nē
Lēmums	Jā	Nē
NINO Atvieglojuma sākuma datums	Jā	Nē
NINO Atvieglojuma beigu datums	Jā	Nē
Piezīmes/Atteikuma iemesls	Jā	Nē
Spēkā līdz	Nē	Jā
Spēkā no	Nē	Jā
ATV_REG_PIEM		
Statuss	Nē	Nē
Īpašums	Nē	Nē
Konts	Nē	Jā
Kopējā piešķirto atvieglojumu summa	Nē	Nē
Taksācijas gads	Nē	Nē
Aprēķina veikšanas datums	Nē	Jā
Lietotājs, kas veicis aprēķinu	Nē	Jā
ATV_REG_OBJ		
Piešķirt	Jā	Nē
Spēkā no	Nē	Jā
Spēkā līdz	Nē	Jā
Izmaiņu veicējs	Nē	Jā

Informācija par datu izmaiņu pārlūkošanas moduli iekļaujamajām tabulām

Objekta tips jeb tabulas nosaukums	Objektu identificējoša informācija	Saistīto sarakstu tabulas	Pastāv grupēšanas tabula	Izmaiņu reģistrēšanas mehānisms
Tasks	Darba procesa vai uzdevuma numurs	Nav	Nē	bu_trigger
Parceles	Adrese, kadastra numurs	Nav	Nē	bu_trigger
Personas_nod	Vārds vai nosaukums Uzvārds personas kods vai reģistrācijas numurs	Nav	Nē	bu_trigger
Atv_reg	Vārds vai nosaukums Uzvārds personas kods vai reģistrācijas numurs Atvieglojuma veids	Atv_reg_piem, Atv_reg_obj	Nē	bu_trigger
Atv_reg_piem	Vārds vai nosaukums Uzvārds personas kods vai reģistrācijas numurs Atvieglojuma veids	Atv_reg, Atv_reg_obj	Nē	table
Atv_reg_obj	Vārds vai nosaukums Uzvārds personas kods vai reģistrācijas numurs Atvieglojuma veids	Atv_reg, Atv_reg_piem	Jā	bu_trigger

Tabulas standarta konfigurācijas paraugs (tabulai ATV_REG_OBJ)

```

{
  "Mechanism": "bu_trigger",
  "Name": "atvieglojumu piešķiršana objektiem",
  "GridCaption": "Objekta datu izmaiņas",
  "Columns": [
    {
      "MainColumn": "null",
      "BuColumn": "BUI_ID",
      "Key": "BUI_ID",
      "Obligatory": "true"
    },
    {
      "MainColumn": "aro_atvrid",
      "BuColumn": "bu_aro_atvrid",
      "Key": "PRIMARY_KEY_ID",
      "Obligatory": "true"
    },
    {
      "MainColumn": "AROGRANT",
      "BuColumn": "BU_AROGRANT",
      "Datatype": "NUMBER",
      "Key": "AROGRANT",
      "Selectable": "true",
      "Comparable": "true",
      "VisibleName": "Piešķirt",
      "obj_grant": "true",
      "Optional": "true"
    },
    {
      "MainColumn": "AROPIRMDAT",
      "BuColumn": "BU_AROPIRMDAT",
      "Datatype": "DATE",
      "Key": "FIRST_DATE",
      "Selectable": "false",
      "Obligatory": "true"
    },
    {
      "MainColumn": "AROPEDDAT",
      "BuColumn": "BU_AROPEDDAT",
      "Datatype": "DATE",
      "Key": "CHANGE_DATE",
      "Selectable": "true",
      "VisibleName": "Spēkā no",
      "Obligatory": "true"
    },
    {
      "Datatype": "DATE",
      "Key": "VALID_TO",
      "Selectable": "true",
      "VisibleName": "Spēkā līdz",
      "Obligatory": "true"
    }
  ]
}

```

```

        "MainColumn": "AROPIRMLIET",
        "BuColumn": "BU_AROPIRMLIET",
        "Datatype": "NUMBER",
        "Key": "FIRST_USERID",
        "Selectable": "false",
        "Obligatory": "true"
    },
    {
        "MainColumn": "AROPEDLIET",
        "BuColumn": "BU_AROPEDLIET",
        "Datatype": "VARCHAR2",
        "Key": "CHANGE_USERID",
        "Selectable": "true",
        "VisibleName": "Izmaiņu veicējs",
        "IsUser": "true",
        "Obligatory": "true"
    }
],
"RelatedObjects": [
    {
        "TableName": "atv_reg",
        "VisibleName": "Atvieglojumu reģistrs"
    },
    {
        "TableName": "atv_reg_piem",
        "VisibleName": "Lēmums NINO"
    }
],
"RelatedObjectsCombo": "true",
"AdditionalGrid": {
    "Visible" : "true",
    "GridCaption": "Objekti",
    "Columns": [
        {
            "Key": "OBJTYPE",
            "VisibleName": "Veids",
            "Datatype": "VARCHAR2",
            "Visible": "true",
            "Excel": "true"
        },
        {
            "Key": "PIESKIRT",
            "VisibleName": "Piešķirt",
            "Datatype": "VARCHAR2",
            "Visible": "true"
        },
        {
            "Key": "ADRESE",
            "VisibleName": "Nosaukums/Adrese",
            "Datatype": "VARCHAR2",
            "Visible": "true",
            "Excel": "true"
        },
        {
            "Key": "KADAPZ",
            "VisibleName": "Kad.Num/Kad.Apz.",
            "Datatype": "VARCHAR2",

```

```

        "Visible": "true",
        "Excel": "true"
    },
    {
        "Key": "KONTS",
        "VisibleName": "Konts",
        "Datatype": "VARCHAR2",
        "AdditionalIDCol": "ARO_IPKD",
        "Visible": "true",
        "Excel": "true",
        "Priority": 2
    },
    {
        "Key": "KOPAPLAT",
        "VisibleName": "Plat.kopā",
        "Datatype": "NUMBER",
        "Visible": "true"
    },
    {
        "Key": "PLAT",
        "VisibleName": "Plat.atviegl.",
        "Datatype": "NUMBER",
        "Visible": "true"
    },
    {
        "Key": "PARCID",
        "Visible": "false",
        "AdditionalIDCol": "ARO_PARCID",
        "Priority": 1
    }
]
}

```

**Uz tabulas standarta konfigurācijas balstīta dinamiski ģenerēta datu atlasēšanas vaicājuma paraugs
(tabulai TASKS)**

```

select null,
       case
         when data.UTRESPONSIBLEUSER = 1 then
           'NINO'
         else
           (select l.useruzv || ' ' || l.userwards
            from lietotaji l
            where l.userid = data.UTRESPONSIBLEUSER)
       end as UTRESPONSIBLEUSER,
       (select UTSCNAME
        from USER_TASK_STATUS_CL
        where utscid = data.UTSOLVESTATUS) as UTSOLVESTATUS,
       data.UTDESC,
       data.CHANGE_DATE,
       lead(data.change_date, 1) over(order by data.change_date, bui_id
nulls last) as valid_to,
       case
         when data.change_userid = 1 or data.change_userid = 0 or
              data.change_userid is null then
           'NINO'
         else
           (select l.useruzv || ' ' || l.userwards
            from lietotaji l
            where l.userid = case
                          when data.change_date is null then
                            data.first_userid
                          else
                            data.change_userid
                        end)
       end as change_userid
from (select bu.bui_id,
            bu.primary_key_id,
            nvl(nvl(lag(bu.change_date, 1)
                       over(order by bu.change_date, bui_id nulls last),
                          bu.first_date),
                change_date) as change_date,
            nvl(nvl(lag(bu.change_userid, 1)
                       over(order by bu.change_date, bui_id nulls last),
                          bu.first_userid),
                change_userid) as change_userid,
            bu.first_userid,
            bu.first_date,
            bu.UTRESPONSIBLEUSER,
            lag(bu.UTRESPONSIBLEUSER, 1) over(order by bu.change_date,
bui_id nulls last) as old_UTRESPONSIBLEUSER,
            bu.UTSOLVESTATUS,
            lag(bu.UTSOLVESTATUS, 1) over(order by bu.change_date,
bui_id nulls last) as old_UTSOLVESTATUS,
            bu.UTDESC,
            lag(bu.UTDESC, 1) over(order by bu.change_date, bui_id nulls
last) as old_UTDESC
      from (select null
            as BUI_ID,
            UTID
            as PRIMARY_KEY_ID,
            UTRESPONSIBLEUSER
            as UTRESPONSIBLEUSER,
            UTSOLVESTATUS
            as UTSOLVESTATUS,
            UTDESC
            as UTDESC,
            UTFIRSTDATE
            as FIRST_DATE,
            UTLASTDATE
            as CHANGE_DATE,

```

```

        UTFIRSTUSER      as FIRST_USERID,
        UTLASTUSER       as CHANGE_USERID
    from tasks
    where UTID = :1
union all
select BUI_ID            as BUI_ID,
       BU_UTID           as PRIMARY_KEY_ID,
       BU_UTRESPONSIBLEUSER as UTRESPONSIBLEUSER,
       BU_UTSOLVESTATUS  as UTSOLVESTATUS,
       BU_UTDESC         as UTDESC,
       BU_UTFIRSTDATE    as FIRST_DATE,
       BU_UTLASTDATE     as CHANGE_DATE,
       BU_UTFIRSTUSER    as FIRST_USERID,
       BU_UTLASTUSER     as CHANGE_USERID
    from bu_tasks
    where bu_UTID = :2) bu) data
where (nvl(data.UTRESPONSIBLEUSER, -9999999999) <>
       nvl(data.old_UTRESPONSIBLEUSER, -9999999999) or
       nvl(data.UTSOLVESTATUS, -9999999999) <>
       nvl(data.old_UTSOLVESTATUS, -9999999999) or
       nvl(data.UTDESC, '#####') <>
       nvl(data.old_UTDESC, '#####'))
order by change_date desc nulls last, bui_id desc nulls first

```


Programmkoda fragments no Oracle PL/SQL Developer 12 izstrādes vides

```

--galvenā procedūra (pēc noklusējuma datu atlase notiek pēc standarta konfigurācijas)
procedure get_data(pi_id      in number,
                  pi_table   in varchar2,
                  pio_columns in out clob,
                  po_data    out sys_refcursor
                  ) as

v_main_table varchar2(30);      --pamattabulas nosaukums
v_bu_table   varchar2(30);      --izmaiņu tabulas nosaukums
v_json       zzjson.json;       --pio_columns konvertēts uz json
v_query      varchar2(32000);   --vaicājuma teksts
begin
    --ja netiek padoti lietotāja meklēšanas parametri, izmanto standarta konfigurāciju
    if pio_columns is null then
        select dcbjson
        into pio_columns
        from data_changes_browser
        where dcbmaintable = lower(pi_table);
    end if;

    --ielasa pamattabulas un izmaiņu tabulas nosaukumu, izmaiņu reģistrēšanas mehānismu
    select dcbmaintable, dcbbutable, zzjson.json_ext.get_string(zzjson.json(dcbjson), 'Mechanism')
    into v_main_table, v_bu_table, g_mechanism
    from data_changes_browser
    where dcbmaintable = lower(pi_table);

    v_json := zzjson.json(pio_columns);

    --izmaiņu reģistrēšanas mehānisms nosaka tālāko algoritmu
    case g_mechanism
        when 'bu_trigger' then --bu trigerā mehānisms

            if v_json.exist('AdditionalID') then --nepieciešams ņemt vērā papildus ID, lai atlasītu datus
                v_query := bu_trigger_mechanism(v_main_table, v_bu_table, v_json);
                --dbms_output.put_line(v_query); --datu atlases vaicājuma drukāšana
            end if;
        end case;
end;

```

```

        open po_data for v_query using in pi_id, zzjson.json_ext.get_string(v_json, 'AdditionalID'), pi_id,
zzjson.json_ext.get_string(v_json, 'AdditionalID');
    else
        v_query := bu_trigger_mechanism(v_main_table, v_bu_table, v_json);
        --dbms_output.put_line(v_query); --datu atlasē vaicājuma drukāšana
        open po_data for v_query using in pi_id, pi_id;
    end if;

    when 'table' then --mehānisms, kur izmaiņas attēlotas pašā tabulā
        v_query := table_mechanism(v_main_table, v_json);

        --dbms_output.put_line(v_query); --datu atlasē vaicājuma drukāšana
        open po_data for v_query using in pi_id;

    end case;
    pio_columns := '';

end get_data;

--funkcija atlasa datus no pamattabulas un izmaiņu tabulas
function direct_table_query(pi_v_columns_list in  zzjson.json_list,
                        pi_main_table      in  varchar2,
                        pi_bu_table        in  varchar2,
                        pi_selected        in  zzjson.json)

return varchar2
is

    v_main_table varchar2(32000); --pamattabulas vaicājums
    v_bu_table   varchar2(32000); --izmaiņu tabulas vaicājums
    v_object     zzjson.json;     --masīva 'Columns' objekts
    v_id_col     varchar2(30);     --primārās atslēgas kolonnas nosaukums
    v_key        varchar2(30);     --masīva 'Columns' objekta atslēga
    v_comma      varchar2(10) default ',';
    v_result_query varchar2(32000);
begin
    for i in 1 .. pi_v_columns_list.count loop--iet cauri json masīva 'Columns' objektiem

        v_object := zzjson.json(pi_v_columns_list.get(i));

```

```

if v_object.exist('MainColumn') then

    v_key := zzjson.json_ext.get_string(v_object, 'Key');
    --komats jāliek pirms visiem elementiem, izņemot pirmo

    if i <> 1 then
        v_main_table := v_main_table || v_comma;
        v_bu_table    := v_bu_table || v_comma;
    end if;

    --kolonnas ar atbilstošajiem nosaukumiem tiek pievienotas vaicājuma tekstam
    v_main_table := v_main_table || zzjson.json_ext.get_string(v_object, 'MainColumn') || ' as ' || v_key;
    v_bu_table    := v_bu_table || zzjson.json_ext.get_string(v_object, 'BuColumn') || ' as ' || v_key;

    --saglabā identifikatora kolonnas nosaukumu
    if zzjson.json_ext.get_string(v_object, 'Key') = 'PRIMARY_KEY_ID' then
        v_id_col := zzjson.json_ext.get_string(v_object, 'MainColumn');
    end if;
end if;
end loop;

if pi_selected.exist('AdditionalID') then
    v_result_query := 'select ' || v_main_table || ' from ' || pi_main_table || ' where ' || v_id_col || '= :1 and ' ||
zzjson.json_ext.get_string(pi_selected, 'AdditionalIDCol') || ' = :3'
        || ' union all ' ||
        'select ' || v_bu_table || ' from ' || pi_bu_table || ' where bu_' || v_id_col || ' = :2 and bu_' ||
zzjson.json_ext.get_string(pi_selected, 'AdditionalIDCol') || ' = :4';
else
    v_result_query := 'select ' || v_main_table || ' from ' || pi_main_table || ' where ' || v_id_col || '= :1'
        || ' union all ' ||
        'select ' || v_bu_table || ' from ' || pi_bu_table || ' where bu_' || v_id_col || ' = :2';
end if;
--atgriež vaicājuma tekstu
return v_result_query;

end direct_table_query;

--vaicājuma sastādīšana, balstoties uz bu trigger mehānismu
function bu_trigger_mechanism (pi_main_table in varchar2,--pamattabulas nosaukums

```

```

        pi_bu_table    in varchar2,--izmaiņu tabulas nosaukums
        pi_selected    in zzjson.json)
return varchar2--lietotāja atzīmētie meklēšanas parametri
is

v_columns_list zzjson.json_list;
v_direct_table_query varchar2(32000);
v_changes_query varchar2(32000);
v_selected_columns varchar2(32000);
v_where_statement varchar2(32000);
v_order varchar2(100) default 'order by change_date desc nulls last, bui_id desc nulls first'; --datus kārtu pēc
izmaiņu veikšanas dilstošā secībā, pēc tam pēc izmaiņu tabulas identifikatora dilstošā secībā

begin
    if pi_selected.exist('Columns') then

        v_columns_list := zzjson.json_ext.get_json_list(pi_selected, 'Columns');

        --katra funkcija atgriež kādu no dinamiskā vaicājuma daļām
        v_direct_table_query := direct_table_query (v_columns_list, pi_main_table, pi_bu_table, pi_selected);
        v_changes_query      := changes_query      (v_columns_list);
        v_selected_columns    := selected_columns  (v_columns_list, pi_selected);
        v_where_statement     := where_statement   (v_columns_list);

    end if;

    return v_selected_columns || ' from ' || '(' || v_changes_query || ' from ' || '(' || v_direct_table_query || ') bu )
data' || v_where_statement || v_order;

end bu_trigger_mechanism;

```

Programmkoda fragments no Microsoft Visual Basic 6 izstrādes vides

```

'-----
' Procedure : fillGrid
' Author    : LaumaS
' Date      : 09.03.2018
' Purpose    : Aizpilda tdbResult(0) grid ar datu lauku izmaiņu vēsturi
'-----
'
Private Sub FillGrid(selectedCols As Object)
    Dim Data          As Object          'atlasīto izmaiņu datu objekts
    Dim i              As Integer         'skaitītājs
    Dim GridArray      As New XArrayDB    'grida masīvs
    On Error GoTo ErrH

    'galvenā datu atlases vaicājuma izpilde datu bāzē
    If Not OraProc2("data_changes_browser_pck.get_data", _
        "pi_id", ID, 1, _
        "pi_table", g_table, 1, _
        "pio_columns", modJSON.ToString(g_mekl_param), 1, _
        "po_data", Data, 2) Then

        Exit Sub
    End If

    'atlasīto ierakstu skaits
    g_result_count = 0
    While Not Data.EOF
        g_result_count = g_result_count + 1
        Data.MoveNext
    Wend

    Data.MoveFirst
    GridArray.ReDim 1, g_result_count, 1, tdbResult(0).Columns.Count
    'Atlasīto vērtību ielasīšana masīvā GridArray

```

```

        For i = 1 To g_result_count
            For j = 1 To tdbResult(0).Columns.Count
                GridArray(i, j) = Data.Fields(selectedCols(j).Item("Key")).Value
            Next
            Data.MoveNext
        Next
        'grida tdbResult(0) pielāgošana vērtību attēlošanai tajā
        tdbResult(0).Array = GridArray
        tdbResult(0).Rebind
        tdbResult(0).Refresh
        tdbResult(0).Bookmark = Null
        tdbResult(0).MarqueeStyle = dbgHighlightRow
    Exit Sub
ErrH:
    'Pazin ""
    kluda Err, "frmDataChangesBrowser.fillGrid"
    Exit Sub
    Resume
End Sub

'-----
' Procedure : LoadForm
' Author    : LaumaS
' Date      : 12.03.2018
' Purpose   : Ielādē formu un sarakstus ar kolonnu, kuras iespējams attēlot un kuru izmaiņas atlasīt,
nosaukumiem.
'-----
'
Public Sub LoadForm(pi_id As String, pi_table As String, pi_show As Boolean, pi_caption As String)
    'Dim strTag As String          'formas tags
    Dim selectedCols As Object    'lietotāja pēdējie iestatījumi
    Dim LI As ListItem            'listview rinda
    Dim j As Integer              'skaitītājs
    On Error GoTo ErrH
    '-----globālo mainīgo aizpildīšana-----

```

```

g_form_caption = pi_caption
g_table = pi_table
ID = pi_id
'-----

Set g_mekl_param = modJSON.CreateJson()
g_mekl_param.Add "Table", g_table

'standarta json konfigurācijas saņemšana
If modJSON.ExecuteJsonHelper("dcb_user_settings_pck.get_columns", g_mekl_param, g_conf) = False Then
    Exit Sub
End If

Me.ZOrder
Me.show
Me.WindowState = vbNormal
Me.Caption = "Datu izmaiņu pārlūks - " & g_conf("Name") & " - " & g_form_caption
Me.Tag = Me.Name & pi_table & pi_id

Call ClearGrid(0)
Call ClearGrid(1)
selectColumns.ListItems.Clear
selectChanges.ListItems.Clear

'ja jāattēlo papildus grids
If g_conf("AdditionalGrid").Item("Visible") = True Then
    Call fillAdditionalGrid
End If

'aizpilda combobox ar saistīto sarakstu nosaukumiem
For Each jsonItem In g_conf("RelatedObjects")
    comboSaist.AddItem jsonItem.Item("VisibleName")
Next

saist.visible = g_conf("RelatedObjectsCombo")

```

```

comboSaist.visible = g_conf("RelatedObjectsCombo")
cmdShowSaist.visible = g_conf("RelatedObjectsCombo")

'lietotāja pēdējo meklēšanas parametru saņemšana
If modJSON.ExecuteJsonHelper("dcb_user_settings_pck.find_settings", g_mekl_param, selectedCols) = False
Then
    Exit Sub
End If

For Each jsonItem In g_conf("Columns")

    'aizpilda listview ar atlasāmo kolonnu nosaukumiem (tagi norāda uz datu tipu priekš teksta sakārtojuma
    gridā)
    If jsonItem.Item("Selectable") = True Then
        Set LI = selectColumns.ListItems.Add(, jsonItem.Item("Key"), jsonItem.Item("VisibleName"))

        If jsonItem.Item("Area") = "true" Then
            LI.Text = LI.Text & " (" & sar_mervieniba & ")"
        End If
        'listview elementa tags ir datu tips
        'ja datu tips ir number, bet šis ir identifikators, tad tabulā būs jāattēlo kā varchar2
        If jsonItem.Item("Datatype") = "NUMBER" And jsonItem.Item("Identificator") = "true" Then
            LI.Tag = "varchar2"
        ElseIf jsonItem.Item("Datatype") = "NUMBER" Then
            LI.Tag = "number"
        ElseIf jsonItem.Item("Datatype") = "DATE" Then
            LI.Tag = "date"
        ElseIf jsonItem.Item("Datatype") = "VARCHAR2" Then
            LI.Tag = "varchar2"
        End If

        'obligātie lauki automātiski tiek atzīmēti
        If jsonItem.Item("Obligatory") = True Then
            LI.Checked = True
        End If
    End If
End For

```



```

        LI.ForeColor = &H808080
    End If

    'attēlo pēdējos lietotāja meklēšanas parametrus attēlojamo kolonnu sarakstā
    For j = 1 To selectedCols("Columns").Count

        If selectedCols("Columns").Item(j).Item("Key") = jsonItem.Item("Key") And
selectedCols("Columns").Item(j).Item("Selectable") = "true" Then
            LI.Checked = True
            Exit For
        End If
    Next
End If

' aizpilda izmaiņu kolonnu sarakstu
If jsonItem.Item("Comparable") = True Then

    Set LI = selectChanges.ListItems.Add(, jsonItem.Item("Key"), jsonItem.Item("VisibleName"))

    If jsonItem.Item("Area") = "true" Then
        LI.Text = LI.Text & " (" & sar_mervieniba & ")"
    End If

    'attēlo pēdējos lietotāja meklēšanas parametrus izmaiņu kolonnu sarakstā
    For j = 1 To selectedCols("Columns").Count

        If selectedCols("Columns").Item(j).Item("Key") = jsonItem.Item("Key") And
selectedCols("Columns").Item(j).Item("Comparable") = "true" Then
            LI.Checked = True

            For Each columnItem In selectColumns.ListItems
                'atzīmē nepieciešamo arī attēlojamo kolonnu sarakstā
                If columnItem.KEY = LI.KEY Then
                    columnItem.Checked = True
                    columnItem.ForeColor = &H808080
                End If
            Next
        End If
    Next
End If

```

```

                End If
            Next
        End If
    Next
End If
Next

selectColumns.selectedItem.selected = False
'sarakstos datu lauki sakārtoti alfabētiskā secībā
selectColumns.sorted = True
selectChanges.sorted = True

If pi_show = True Then
    'izsauc datu izmaiņu vēstures tabulas aizpildīšanu
    Call cmdShow_Click
End If

'formas gridu nosaukumu maiņa
tdbResult(0).Caption = g_conf("GridCaption")
If g_conf("AdditionalGrid").Item("Visible") = True Then
    tdbResult(1).Caption = g_conf("AdditionalGrid").Item("GridCaption")
End If

Exit Sub
ErrH:
    kluda Err, "frmDataChangesBrowser.LoadForm"
Exit Sub
Resume
End Sub

```

Datu izmaiņu pārlūkošanas moduļa ekrānuzņēmumi

Datu izmaiņu pārlūkošana - zemes vienības vēsture - Raiņa bulvāris 19 (0100 005 0045)

Attēlojamās kolonnas

☐ Adrese
☒ Izmaiņu veicējs
☒ Kopējā kadastrālā vērtība
☒ Kopējās kadastrālās vērtības valūta
☐ Nodokļu kadastrālā vērtība
☐ Nodokļu kadastrālās vērtības valūta
☐ Piezīmes
☐ Platība (m²)
☒ Spēkā līdz
☒ Spēkā no

Atzīmēt visus **Noņemt visus**

Kādas izmaiņas attēlot

☐ Adrese
☒ Kopējā kadastrālā vērtība
☐ Nodokļu kadastrālā vērtība
☐ Piezīmes
☐ Platība (m²)

Atzīmēt visus **Noņemt visus**

	Kopējā kadastrālā vērtība	Kopējās kadastrālās vērtības valūta	Spēkā no	Spēkā līdz	Izmaiņu veicējs
▶	2066883,00	EUR	17.01.2017 19:23:22		Apmaina ar Kadastru
	1687251,00	EUR	05.01.2014 16:53:56	17.01.2017 19:23:22	Apmaina ar Kadastru
	948640,00	LVL	04.01.2012 02:12:10	05.01.2014 16:53:56	Apmaina ar Kadastru
	1043504,00	LVL	18.01.2010 22:59:04	04.01.2012 02:12:10	Apmaina ar Kadastru
	1667235,00	LVL	17.01.2008 19:20:38	18.01.2010 22:59:04	Apmaina ar Kadastru
	450604,00	LVL	09.01.2003 03:03:48	17.01.2008 19:20:38	Apmaina ar Kadastru
	462035,00	LVL	24.09.2002 09:17:01	09.01.2003 03:03:48	Apmaina ar Kadastru
	579856,00	LVL	13.01.2001 14:06:27	24.09.2002 09:17:01	Apmaina ar Kadastru

Attēlot **Drukāt**

Beigt

Datu izmaiņu pārlūkošana - zemes vienības vēsture - Raiņa bulvāris 19 (0100 005 0045)

Attēlojamās kolonnas

- ☐ Adrese
- ☒ Izmaiņu veicējs
- ☒ Kopējā kadastrālā vērtība
- ☒ Kopējās kadastrālās vērtības valūta
- ☐ Nodokļu kadastrālā vērtība
- ☐ Nodokļu kadastrālās vērtības valūta
- ☐ Piezīmes
- ☐ Platība (m²)
- ☒ Spēkā līdz
- ☒ Spēkā no

Atzīmēt visus Noņemt visus

Kādas izmaiņas attēlot

- ☐ Adrese
- ☐ Kopējā kadastrālā vērtība
- ☐ Nodokļu kadastrālā vērtība
- ☐ Piezīmes
- ☐ Platība (m²)

Atzīmēt visus Noņemt visus

Atzīmējiet izmaiņu laukus!

Labi

Attēlot Drukāt Beigt

Datu izmaiņu pārļūks - atvieglojumu piešķiršana objektiem - 4.8.1(no 2016.g.)-Personas ar 3 un vairāk bērniem(19.gadi) - 90%

Attēlojamās kolonnas

☒ Izmaiņu veicējs

☒ Piešķirt

☒ Spēkā līdz

☒ Spēkā no

Atzīmēt visus Noņemt visus

Kādas izmaiņas attēlot

☒ Piešķirt

Atzīmēt visus Noņemt visus

Saistītie saraksti

▼ Attēlot

Objekti							
	Veids	Piešķirt	Nosaukums/Adrese	Kad.Num/Kad.Apz.	Konts	Plat.kopā	Plat.atviegl.
	Z	Jā	Starta iela 10A	0100-087-2086	791563	1059	
	E	Objektiem	Starta iela 10A	0100-087-2086	791565		
▶	E	Jā	Starta iela 10A- 1	0100-925-3020	6732260		
	Z	Jā	Starta iela 10A- 1	0100-925-3020	6732259		
	E	Nē	Starta iela 10A- 2	0100-925-3122	6733451		
	Z	Jā	Starta iela 10A- 2	0100-925-3122	6733450		

Objekta datu izmaiņas				
	Piešķirt	Spēkā no	Spēkā līdz	Izmaiņu veicējs
▶	Jā	24.04.2018 16:04:28		Svilpe Lauma
	Nē	24.04.2018 14:28:31	24.04.2018 16:04:28	Svilpe Lauma
	Nav zināms	24.04.2018 14:25:01	24.04.2018 14:28:31	Svilpe Lauma

Attēlot Drukāt Beigt

Programmkoda fragments no automatizētās vienībtestēšanas pakotnes, izmantojot *utPLSQL***vienībtestēšanas ietvaru**

```

procedure load_pers_data
is
  person personas_nod%rowtype;

begin
  g_pkd := zr.personas_nod_pkd_sq.nextval;

  person.pkd := g_pkd;
  person.pkods := '111111-11111';
  person.puzv := 'TestsUzvārds';
  person.pv := 'TestsVārds';
  person.padr := 'Testa iela 1';
  person.pstatus := 2;
  person.ppirmdat := sysdate;
  person.ppeddat := sysdate;
  person.ppirmuser := 1;
  person.ppeduser := 1;
  person.pappl := 1;
  person.papplped := 1;

  insert into personas_nod values person;

  update personas_nod --8.rinda
  set ptel = '12345678',
     ppeddat = sysdate
  where pkd = g_pkd;

  update personas_nod --7.rinda
  set ppiez = 'Testa piezīme',
     ppeddat = sysdate
  where pkd = g_pkd;

  update personas_nod --6.rinda
  set ppils = 'Rīga',
     ppeddat = sysdate
  where pkd = g_pkd;

  update personas_nod --5.rinda
  set puzv = 'TestsUzvārds2',
     ppeddat = sysdate
  where pkd = g_pkd;

  update personas_nod --4.rinda
  set pcitivardi = 'TestsCitsVārds',
     ppeddat = sysdate
  where pkd = g_pkd;

  update personas_nod --3.rinda
  set pemails = 'tests@tests.lv',
     ppeddat = sysdate
  where pkd = g_pkd;

  update personas_nod --2.rinda
  set pv = 'TestsVārds2',
     ppeddat = sysdate
  where pkd = g_pkd;

  update personas_nod --1.rinda

```

```

set pkods = '222222-22222',
    ppeddat = sysdate,
    pmirsanasdat = sysdate
where pkd = g_pkd;

update personas_nod
set puzv_garais = '1234',
    ppeddat = sysdate
where pkd = g_pkd;

end load_pers_data;

--izpilda get_data procedūru
procedure load_data_changes(v_changes in out sys_refcursor, v_table in
varchar2)
is
    v_columns clob default null;
begin
    ZR.DATA_CHANGES_BROWSER_PCK.GET_DATA(pi_id => g_pkd,
                                           pi_table => v_table,
                                           pio_columns => v_columns,
                                           po_data => v_changes);

end load_data_changes;

--pārbauda personas e-pasta maiņu
procedure check_person_email
is

    v_changes sys_refcursor;
    i number(10) default 0;
    v_emails varchar2(100);

begin
    load_data_changes(v_changes, 'personas_nod');

    loop
        fetch v_changes into bui_id, pkods, puzv, pv, pcitivardi, padr,
ppils, ptel, pemails, pstatus, pmirsanasdat, ppiez, change_date, valid_to,
change_userid;
        exit when v_changes%notfound;
        i := i + 1;
        case i
            when 3 then v_emails := pemails;
            else null;
        end case;

    end loop;
    close v_changes;
    ut.expect(a_actual => v_emails, a_message => 'Incorrect email change
shown').to_equal('tests@tests.lv');
end check_person_email;

```

Kvalifikācijas darbs „*Datu izmaiņu pārlūkošanas moduļa izstrāde nekustamā īpašuma nodokļa administrēšanas sistēmai NINO*” 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: *Lauma Svilpe* _____ .05.2018.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *M.inž. Viktors Salijs* _____ .05.2018.

Recenzents: *M.dat. Egons Šolmanis*

Darbs iesniegts 28.05.2018.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2018. prot. Nr. _____

Komisijas sekretārs(-e): _____