

LATVIJAS UNIVERSITĀTES  
DATORIKAS FAKULTĀTE

**Uzņēmuma reģistra datu integrācija ar lietvedības  
sistēmu LIETVARIS**

KVALIFIKĀCIJAS DARBS

Autors: **Laura Tīrumniece**

Studenta apliecības Nr.: lt16028

Darba vadītājs: Ilmārs Urbāns

RĪGA 2018

## ANOTĀCIJA

Darbā aprakstīta lietvedības sistēmas LIETVARIS risinājumu izstrāde, kas paredz jaunas funkcionalitātes izstrādi un esošu problēmu labošanu.

Risinājumi padara sistēmu ērtāku lietošanai, nodrošina juridisku personu meklēšanu un juridiskas personas datu izgūšanu pēc reģistrācijas numura, kas būtiski uzlabo sistēmas LIETVARIS lietotāja darba efektivitāti.

**Atslēgvārdi:** lietvedība, LIETVARIS, uzņēmuma reģistrs, uzlabojumi, spējā programmatūras izstrāde.

## **ABSTRACT**

### **REGISTER OF ENTERPRISES DATA INTEGRATION IN RECORD-KEEPING INFORMATION SYSTEM LIETVARIS**

The paper describes the development of additions to LIETVARIS record-keeping system which ensure problem solving and the development of a new functionality.

The founded additions ensure the system to be easier for general usage providing legal person and their personal data retrieval based on their registration numbers which increasingly enhance the work efficiency of LIETVARIS users.

**Keywords:** record-keeping, LIETVARIS, register of enterprises data, improvements, agile software development.

# SATURS

VĀRDNĪCA.....	7
1. IEVADS .....	9
2. IZMANTOTO TEHNOLOĢIJU ANALĪZE.....	11
3. LIETOTĀJSTĀSTI .....	13
3.1. Piekļuves tiesības un sistēmas paziņojumi .....	13
3.1.1. Tiesības kontaktpersonu modulī.....	13
3.1.2. Sistēmas paziņojumi .....	14
3.2. Lietotāju stāsti .....	14
3.2.1. Jaunas kontaktpersonas pievienošanas formas aizvēršana.....	14
3.2.2. Kontaktpersonas kartītes aizvēršana jaunā cilnē.....	15
3.2.3. Jaunas kontaktpersonas pievienošanas formas izmēra pielāgošana loga izmēra izmaiņām.....	15
3.2.4. Kontaktpersonas kartītes saglabāšanas pogas izmērs .....	16
3.2.5. Kontaktpersonu kartītes šķirklju informācija.....	16
3.2.6. Kontaktpersonas kartītes datu pārbaude.....	16
3.2.7. Jaunas kontaktpersonas pievienošanas formas ievades datu pārbaude .....	17
3.2.8. Juridiskas personas datu meklēšana pēc reģistrācijas numura un formas aizpildīšana .....	18
3.2.9. Juridiskas personas datu brīvā meklēšana un formas aizpildīšana.....	19
4. TRASĒJAMĪBAS TABULA .....	21
5. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS .....	23
5.1. Ievads.....	23
5.1.1. Nolūks.....	23
5.1.2. Darbības sfēra .....	23
5.1.3. Definīcijas.....	23
5.1.4. Saistība ar citiem dokumentiem .....	23
5.1.5. Pārskats.....	23
5.2. Darba uzdevuma DU3282 projektējums .....	24
5.2.1. Datu detalizētais projektējums.....	25

5.2.1.1.	Tabula IN_NMAKS_JP .....	25
5.2.1.2.	Tabula IN_SAITADRPER .....	26
5.2.1.3.	Tabula ADRESES .....	26
5.2.1.4.	Tabula IN_KL_ADRVEIDS .....	27
5.2.1.5.	Tabula CL_COUNTRIES .....	27
5.2.2.	Funkcionālais projektējums .....	27
5.2.2.1.	Pakotne I_UR_INTEGRATION_PCK .....	27
5.2.2.2.	Pakotne APP_URINTEGRATION_PCK .....	29
5.2.2.3.	Funkcija <i>findCompany</i> .....	29
5.2.3.	LIETVARIS projekta gatavie risinājumi.....	31
5.2.4.	Darba uzdevuma DU3282 uzlabojumu saskarnes.....	31
5.2.4.1.	Juridisku personu meklēšana pēc reģistrācijas numura.....	31
5.3.	Darba uzdevums DU3284 projektējums.....	36
5.3.1.	Datu detalizēts projektējums .....	38
5.3.1.1.	Tabula ES_SYNC .....	38
5.3.1.2.	Tabula IN_NMAKS_JP .....	38
5.3.1.3.	Tabula IN_SAITADRPER .....	39
5.3.1.4.	Tabula ADRESES .....	39
5.3.1.5.	Tabula IN_KL_ADRVEIDS .....	40
5.3.1.6.	Tabula CL_COUNTRIES .....	40
5.3.2.	Funkcionālais projektējums.....	40
5.3.2.1.	Pakotne APP_URINTEGRATION_PCK .....	41
5.3.2.2.	Pakotne COMPANY_ES_IMPORT_PCK .....	41
5.3.2.3.	Funkcija <i>urSearch</i> .....	43
5.3.2.4.	Funkcija <i>onSelect</i> .....	43
5.3.3.	Darba uzdevuma DU3284 Elasticsearch projektējums .....	44
5.3.4.	LIETVARIS projekta gatavie risinājumi.....	45
5.3.5.	Darba uzdevuma DU3284 labojumu saskarnes.....	46
5.3.5.1.	Juridisku personu brīvā meklēšana .....	46
5.4.	Darba uzdevuma “Labojumi kontaktpersonu modulī” projektējums .....	50
5.4.1.	Labotā moduļa vispārīgs apraksts .....	51
5.4.2.	Datu detalizētais projektējums.....	52
5.4.2.1.	Tabula DOCUMENT_PERSONS.....	52

5.4.2.2. Tabula CLIENTS .....	52
5.4.3.    Detalizēts moduļa projektējums .....	53
5.4.4.    Darba uzdevuma “Labojumi kontaktpersonu modulī” labojumu saskarnes .....	56
5.4.4.1. Pogas “Atgriezties” funkcionalitātes labojums .....	56
5.4.4.2. Jaunas kontaktpersonas pievienošanas formas loga izmērs .....	56
5.4.4.3. Pogas “Saglabāt” vizuālais uzlabojums .....	57
5.4.4.4. Pogas “Atgriezties” funkcionalitātes labojums .....	58
5.4.4.5. Šķirkļu informatīvā teksta labojums .....	58
5.4.4.6. Kļūdu paziņojuma labojums.....	59
5.4.4.7. Ārvalstnieka reģ.nr./pers.koda pārbaude .....	60
6.    PROJEKTA ORGANIZĀCIJA .....	61
7.    KVALITĀTES NODROŠINĀŠANA.....	65
8.    KONFIGURĀCIJU PĀRVALDĪBA .....	67
8.1.    Dokumentācijas konfigurācijas pārvaldība .....	67
8.2.    Programmkoda konfigurācijas pārvaldība.....	67
9.    TESTĒŠANAS DOKUMENTĀCIJA .....	71
9.1.    Testēšanas žurnāls .....	71
9.1.1.    Testēšanas plāns .....	71
9.1.2.    Testu rezultātu apzīmējumi .....	71
9.1.3.    Funkcionālā testēšana .....	71
9.1.3.1. Lietotājstāstu funkcionālā testēšana .....	73
9.1.3.2. Kļūdu apstrāde.....	85
9.1.4.    Automatizētā vienībtestēšana .....	86
9.2.    Testēšanas secinājumi.....	87
10.   DARBIETILPĪBAS NOVĒRTĒJUMS.....	88
11.   REZULTĀTI .....	90
12.   SECINĀJUMI .....	91
IZMANTOTIE LITERATŪRAS AVOTI .....	92
PIELIKUMS .....	93

## VĀRDNĪCA

Vārds	Skaidrojums
DVS	Dokumentu vadības sistēma.
Lietvedība	Dokumentēšanas un ar dokumentiem saistītas darba organizācijas sistēma.
LIETVARIS	Dokumentu vadības sistēma.
DB	Datu bāze.
JSON	Datu apmaiņas formāts, kas paredzēts strukturētai datu pārraidei.
DU	Darba uzdevums.
Pakotne	Funkciju, procedūru, konstanšu kopums <i>Oracle PL/SQL</i> programmēšanas valodā.
PPA	Programmatūras projektējuma apraksts.
Git	Versiju kontroles sistēma.
Repozitorijs	Centralizēta datu bāze, kas nodrošina kopējās informācijas uzturēšanu un pieeju tai atbilstoši lietotājam definētām tiesībām.
Lietotājstāsti	Īsi stāsti no lietotāja perspektīvas, kas apraksta kaut kādas prasības.
Redmine	Atvērtā pirmkoda projektu un resursu plānošanas un vadības rīks.
Jenkins	Nepārtrauktās integrācijas sistēma, kas nodrošina programmatūras koda kompilēšanu, būvēšanu un testēšanu.
Spējā programmatūras izstrāde	( <i>angļu val. "Agile software development"</i> ) Informācijas sistēmu ātra izstrāde un ieviešana, ar pakāpeniskiem tuvinājumiem uzlabojot šīs sistēmas un iekļaujot tajās jaunākās lietotāju prasības.
Forma	Iespiests vai citādi izgatavots dokuments ar iezīmētām vietām noteiktas informācijas

	ierakstīšanai. Piemēram, jaunas kontaktpersonas pievienošanas forma.
Lauks	Ieraksta daļa ar patstāvīgu funkcionālu nozīmi, ko datora programma apstrādā kā atsevišķu datu elementu (piemēram, komandas adreses lauks, operanda lauks, ziņojuma datu lauks.
Izmaiņu pieprasījums	(no <i>angļu val.</i> “ <i>Pull request</i> ”) Pieprasījums, kuru veido izstrādātājs, lai sapludinātu savas pirmkoda izmaiņas ar kādu no repozitorija zariem <i>Git</i> versiju kontroles sistēmā.
Ietvars	Programmatūra, kas atvieglo programmaproduktu komponentu izstrādi un apvienošanu, nodrošinot pamata funkcijas un bibliotēkas.
Sprints	Spējās programmatūras izstrādes attīstības posms, kurā tiek veikta prasību detalizācija, kļūdu labošana un programmatūras tālāka attīstība.
Stāstu punkti	(no <i>angļu val.</i> “ <i>Story points</i> ”) Mērvienība, ar kuru mēra uzdevuma vai problēmas sarežģītību un plānoto patērējamo laiku tā izstrādei.
Juridiska persona	Uzņēmums, iestāde vai organizācija, kas, pildot kādas funkcijas, rīkojas savā vārdā kā patstāvīgs civiltiesību un civilpienākumu subjekts.
UR	Uzņēmumu reģistrs.
Atslēgvārds	Vārds vai to kopa, kas tiek izmantots kāda dokumenta vai teksta satura noteikšanai un meklēšanai.
NM	Nodokļu maksātājs.
VPS	Vienotā pašvaldību sistēma.

# 1. IEVADS

## Par lietvedības sistēmu LIETVARIS

LIETVARIS ir uzņēmuma SIA "ZZ Dats" izstrādāta lietvedības sistēma, kas nodrošina tādas svarīgas lietvedības sistēmas funkcijas kā dokumentu reģistrāciju, datu un datņu uzglabāšanu, dokumentu saņemšanu un nosūtīšanu adresātam, pārskatu veidošanu, meklēšanas funkcionalitāti un uzdevumu kontroles izpildi.

LIETVARIS sistēma ir daļa no Vienotās pašvaldību sistēmas (turpmāk tekstā - VPS). Sistēma ir integrēta ar citām VPS lietojumprogrammām, to skaitā, nekustamā īpašuma nodokļu administrēšanas sistēmu "NINO", sociālas palīdzības administrēšanas sistēmu "SOPA", pašvaldību darbinieku uzskaites sistēmu "KADRI", resursu vadības un uzskaites sistēmu "G-VEDIS".

2018.gada maijā LIETVARIS sistēmu sekmīgi izmanto jau 27 Latvijas novadu pašvaldībās un tiek izstrādāta sistēmas versija, kas tiks integrēta A/S "Augstspriegumu tīkls" darbā.

## Kvalifikācijas darba mērķis

Kvalifikācijas darba mērķis ir izstrādāt risinājumu, kas atvieglotu lietvedības sistēmas LIETVARIS lietotāju darbu kontaktpersonu modulī, nodrošinātu juridisku personu meklēšanu, kā arī uzlabotu sistēmas ērtu lietošanu.

Lai nodrošinātu veiksmīgu darba izstrādi, tika izvirzīti sekojoši uzdevumi:

- izstrādāt lietotājstāstus 9 darba uzdevumiem;
- izstrādāt programmatūras projektējuma aprakstu;
- izstrādāt programmkodu 9 darba uzdevumiem kontaktpersonu modulī;
- veikt funkcionālo testēšanu un vienībtestēšanu;
- izstrādāt testēšanas dokumentāciju;
- izstrādāt projekta organizācijas, kvalitātes nodrošināšanas un konfigurāciju pārvaldības aprakstu un veikt darbietilpības novērtējumu.

## Darba struktūra

Kvalifikācijas darba dokumentācija sastāv no sekojošām daļām:

- 1.nodaļa apraksta lietvedības sistēmu LIETVARIS, kvalifikācijas darba mērķus un uzdevumus, izvirzīto mērķu sasniegšanai;
- 2.nodaļa apraksta darba izstrādē izmantotās tehnoloģijas;
- 3.nodaļa apraksta lietotājstāstus, kur katram darba uzdevumam ir savs lietotājstāsta apraksts;

- 4.nodaļa satur trasējamības tabulu, kas apraksta saistību starp lietotājistāstiem, darba uzdevumiem, saskarņu labojumiem un DU projektējumiem;
- 5.nodaļa apraksta programmatūras projektējuma aprakstu, kurš ir sadalīts 3 darba uzdevumu projektējumos: darba uzdevuma DU3282 projektējumā, darba uzdevuma DU3284 projektējumā un darba uzdevuma “Labojumi kontaktpersonu modulī” projektējumā, kas apvieno darba uzdevumus DU4783, DU2592, DU1115, DU2964, DU4406, DU4863;
- 6.nodaļā tiek aprakstīta LIETVARIS projekta organizācija;
- 7.nodaļa izklāsta darba izstrādes laikā ievērotos kvalitātes standartus;
- 8.nodaļā tiek aprakstīti kvalifikācijas darbā izmantotie pārvaldības rīki un principi;
- 9.nodaļa satur testēšanas dokumentāciju, kas apraksta veiktos funkcionālos testus un vienībtestus;
- 10.nodaļa satur darbietilpības novērtējumu, kurā aprakstīts kvalifikācijas darba izstrādāšanai patērētais laiks;
- 11.nodaļa apkopo sasniegtos rezultātus;
- 12.nodaļa satur darba autora secinājumus.

## 2. IZMANTOTO TEHNOLOĢIJU ANALĪZE

Lietvedības sistēmas LIETVARIS izstrādē pašlaik tiek izmantots plašs klāsts ar tehnoloģijām, taču šī kvalifikācijas darba autors izstrādes laikā saskārās ar sekojošām tehnoloģijām:

- *Microsoft .NET Core*;
- *AngularJS*;
- *Oracle Database*;
- *Elasticsearch*.

### ***Microsoft .NET Core***

*.NET Core* ir bezmaksas atvērta pirmkoda izstrādes platforma dažādu veidu lietojumprogrammu veidošanai. Šo ietvaru var izmantot gan uz *Windows*, gan uz *Linux*, gan uz *macOS* operētājsistēmām, kas arī ir viena no lielākajām šī produkta priekšrocībām. *.NET* atbalsta vairākas valodas : *C#, F#, Visual Basic* [12]. Darba izstrādes laikā risinājumi tika implementēti, izmantojot *C#* valodu.

### ***AngularJS***

*AngularJS* ir atvērta pirmkoda tīmekļa lietotņu ietvars. Ietvars atbalsta tikai *JavaScript* valodu. Pateicoties *AngularJS* priekšrocībām:

- plaši pieejama dokumentācija;
- piedāvā komponentes un kontrolierus;
- divvirzienu datu sasaiste, kas nodrošina dinamisku satura attēlošanu;
- lietotnes klienta daļas un servera daļas atdalīšana, kas nodrošina paralēlu izstrādes procesu;
- testējamība;
- paplašina *HTML*,

LIETVARIS projekts izmanto šo tehnoloģiju [13].

### ***Oracle Database***

*Oracle* datu bāze ir kompānijas *ORACLE* izstrādāta datu bāzes pārvaldības sistēma. Tā nodrošina datu glabāšanu, atbalsta objektorientētas tabulas, kas ļauj saglabāt procedūras un funkcijas tabulās [2]. *Oracle* ir pierādījusi sevi kā stabilu datu bāzes pārvaldības sistēmu, kas ir viens no faktoriem tās implementācijai LIETVARIS projektā.

### *Elasticsearch*

*Elasticsearch* ir dokumentu glabātuve, kas nodrošina apjomīgu dokumentu glabāšanu un dokumentu meklēšanu gandrīz reālajā laikā, kas ir viens no galvenajiem iemesliem tehnoloģijas implementācijai LIETVARIS projektā [9].

Dokumenti tiek glabāti indeksos. LIETVARIS projektā datu atlasīšanai tiek izmantota gan *Oracle* datu bāze, gan *Elasticsearch*. Lai aizpildītu *Elasticsearch* indeksu ar vajadzīgajiem datiem, *Oracle* datu bāzē tiek veidotas procedūras, kas sinhronizē tabulu datus ar indeksu, nodrošinot datu vienlīdzību abās datu uzglabāšanas vietās.

### 3. LIETOTĀJSTĀSTI

#### 3.1. Piekļuves tiesības un sistēmas paziņojumi

Lietotājstāsti nav precīzi veidoti pēc spējas programmatūras izstrādes principiem, tie ir lietotājstāsti, kas ir papildināti ar detalizētu prasību specifikāciju, jo tajos tiek detalizētas tādas lietas kā lietotāju tiesības un vēlamie atgriežamie rezultāti.

Lai lietotājstāstu apskate būtu vieglāk uztverama, visi iespējamie sistēmas paziņojumi aprakstīti atsevišķā tabulā (skatīt 3.3.tabulu) un turpmāk dokumentā tiek izmantotas atsauces uz tiem, lietojot tiem piešķirtos identifikatorus. Tabulās 3.1 un 3.2 ir aprakstītas lietotāju piekļuves tiesības, kuras turpmāk dokumentā tiek izmantotas.

##### 3.1.1. Tiesības kontaktpersonu modulī

Lietotāja piekļuves tiesības kontaktpersonu modulī ir sekojošas. Pavisam kopā ir 3 tiesību līmeņi – “Skatīt”, “Labot” un “Dzēst”. Kartītes pieejamību apraksta 3.1. tabula.

*3.1.Tabula*

**Pieejas tiesības kontaktpersonu modulī kontaktpersonas kartītei un jaunas kontaktpersonas pievienošanas formai**

<b>Tiesības:</b>	<b>Pieejamība:</b>	<b>Kartītes pieejamība</b>	<b>Veidot jaunu, labot</b>
	Skatīt	X	
	Labot	X	X
	Dzēst	X	X

Kartītē pieejamo pogu pieejamību apraksta 3.2. tabula.

*3.2.Tabula*

**Pieejas tiesības kontaktpersonu modulī pogām “Atgriezties” un “Saglabāt”**

<b>Tiesības uz kartīti:</b>	<b>Pieejamība:</b>	<b>Atgriezties</b>	<b>Saglabāt</b>
	Skatīt	X	
	Labot	X	X
	Dzēst	X	X

### 3.1.2. Sistēmas paziņojumi

Tabulā 3.3. aprakstīti sistēmas paziņojumi, ar kuriem kvalifikācijas darba autors darba uzdevumu izstrādes laikā saskārās un izmantoja.

3.3.Tabula

#### Sistēmas paziņojumu apraksts

Identifikators	Apraksts
SP01	Norādītais kods jau eksistē!
SP02	Neprecīzi norādīts personas kods!
SP03	Neprecīzi norādīts reģistrācijas numurs!
SP04	Ievadīts nekorekts reģistrācijas numurs!
SP05	Pēc ievadītā reģistrācijas numura kontaktpersona netika atrasta!
SP06	Lai meklētu ievadiet vismaz 3 simbolus
SP07	Nav atrasts neviens ieraksts
SP08	Atrasti X ieraksti
SP09	Nav ievadīts reģistrācijas numurs!

\*Simbola X vietā ir atlasīto ierakstu skaits.

### 3.2. Lietotāju stāsti

Lietotājstāsti izstrādāti, ņemot vērā *Redmine* projektu un resursu plānošanas un vadības rīkā definētos darba uzdevumus. Lietotājstāstu un *Redmine* darba uzdevumu saistību skatīt 1.pielikumā.

#### 3.2.1. Jaunas kontaktpersonas pievienošanas formas aizvēršana

Lietotājstāsts
Kā lietvedības sistēmas LIETVARIS lietotājs ar atbilstošām piekļuves tiesībām, es varu atvērt jaunas kontaktpersonas pievienošanas formu un arī aizvērt to.
Akceptējošie kritēriji
<ul style="list-style-type: none"><li>Lietotājiem ar tiesību "Labot" vai "Dzēst" jābūt iespējai atvērt jaunas kontaktpersonas pievienošanas formu un aizvērt to ar pogas "Atgriezties" palīdzību.</li><li>Lietotāji ar tiesību "Skatīt" nevar atvērt jaunas kontaktpersonas pievienošanas formu.</li></ul>
Identifikators

DU4783
--------

### 3.2.2. Kontaktpersonas kartītes aizvēršana jaunā cilnē

Lietotājstāsts
Kā lietvedības sistēmas LIETVARIS lietotājs ar tiesībām piekļūt attiecīgajai kartītei, es varu atvērt kontaktpersonas kartīti jaunā cilnē vai atvērt to no e-pasta jaunā cilnē, un arī aizvērt to.
Akceptējošie kritēriji
<ul style="list-style-type: none"><li>Lietotāji ar tiesību “Labot” vai “Dzēst” var atvērt jaunas kontaktpersonas pievienošanas formu jaunā cilnē vai ar e-pastā pievienotu hipersaiti un aizvērt to ar pogas “Aizvērt logu” palīdzību.</li><li>Atverot kartīti jaunā cilnē, lietotājs nevar vairāk atgriezties uz kontaktpersonu sarakstu, tas ir, poga “Atgriezties” ir noņemta.</li></ul>
Identifikators
DU2592

### 3.2.3. Jaunas kontaktpersonas pievienošanas formas izmēra

#### pielāgošana loga izmēra izmaiņām

Lietotājstāsts
Es kā lietvedības sistēmas LIETVARIS lietotājs ar atbilstošām piekļuves tiesībām varu atvērt jaunas kontaktpersonas pievienošanas formu un mainīt loga izmērus, un redzēt visus formas lauku nosaukumus.
Akceptējošie kritēriji
<ul style="list-style-type: none"><li>Lietotāji ar tiesību “Labot” vai “Dzēst” var atvērt jaunas kontaktpersonas pievienošanas formu un brīvi mainīt loga izmēru, nepazaudējot formas lauku nosaukumus.</li><li>Lietotāji ar tiesību “Skatīt” nevar atvērt jaunas kontaktpersonas pievienošanas formu.</li></ul>
Identifikators
DU1115

### 3.2.4. Kontaktpersonas kartītes saglabāšanas pogas izmērs

Lietotājstāsts
Es kā lietvedības sistēmas LIETVARIS lietotājs ar tiesībām atvērt attiecīgo kartīti varu atvērt kontaktpersonas kartīti un redzēt pogas “Saglabāt” un “Atgriezties” vienādos izmēros.
Akceptējošie kritēriji
<ul style="list-style-type: none"><li>Lietotāji ar piekļuves tiesībām “Labot” vai “Dzēst”, atverot kontaktpersonas kartīti, pogas “Saglabāt” un “Atgriezties” redz vienāda izmēra.</li><li>Lietotāji ar tiesību “Skatīt” nevar atvērt kontaktpersonas kartīti.</li></ul>
Identifikators
DU2964

### 3.2.5. Kontaktpersonu kartītes šķirķļu informācija

Lietotājstāsts
Es kā lietvedības sistēmas LIETVARIS lietotājs ar atbilstošām tiesībām varu atvērt kontaktpersonas kartītē šķirķļus “Sarakste” un “Notikumi” un redzēt visu vajadzīgo informāciju, kas nedublējas.
Akceptējošie kritēriji
<ul style="list-style-type: none"><li>Lietotāji ar piekļuves tiesībām “Labot” vai “Dzēst”, atverot kontaktpersonas kartīti, pie šķirķļu nosaukumiem “Sarakste” un “Notikumi” neredz lieku informāciju, kas dublē šķirķļu nosaukumus.</li><li>Lietotāji ar tiesību “Skatīt” nevar atvērt kontaktpersonas kartīti.</li></ul>
Identifikators
DU1123

### 3.2.6. Kontaktpersonas kartītes datu pārbaude

Lietotājstāsts
Kā lietvedības sistēmas LIETVARIS lietotājs ar atbilstošām piekļuves tiesībām, es varu atvērt kontaktpersonas kartīti un redzēt kļūdas paziņojumu, ievadot jau eksistējošu personas kodu.
Akceptējošie kritēriji
<ul style="list-style-type: none"><li>Lietotāji ar piekļuves tiesībām “Labot” vai “Dzēst” var atvērt kontaktpersonas kartiņu un labot kontaktpersonas personas kodu/reģistrācijas numuru.</li></ul>

<ul style="list-style-type: none"> <li>○ Lietotājs, izvēloties kontaktpersonas veidu “Fiziska persona”, nevar ievadīt vairāk par 11 simboliem. Ja lietotājs ievada nekorektu personas kodu, tad tiek atgriezts SP02 kļūdas paziņojums.</li> <li>○ Lietotājs, izvēloties kontaktpersonas veidu “Juridiska persona”, nevar ievadīt vairāk par 11 simboliem. Ja lietotājs ievada nekorektu reģistrācijas numuru, tad tiek atgriezts SP03 kļūdas paziņojums.</li> <li>○ Lietotājs, izvēloties kontaktpersonas veida “Ārvalstnieks”, nevar ievadīt vairāk par 14 simboliem.</li> <li>○ Ja tiek mēģināts saglabāt kontaktpersonas ar identiskiem personas kodiem/reģ.nr., tad tiek atgriezts kļūdas paziņojums SP01.</li> <li>○ Ja tiek ievadīts unikāls personas kods/reģ.nr., tad izmaiņas tiek saglabātas.</li> <li>● Lietotāji ar tiesību “Skatīt” nevar atvērt kontaktpersonas kartīti.</li> </ul>
Identifikators
DU4406

### **3.2.7. Jaunas kontaktpersonas pievienošanas formas ievades datu pārbaude**

Lietotājs
Es kā lietvedības sistēmas LIETVARIS lietotājs ar atbilstošām piekļuves tiesībām varu atvērt jaunas kontaktpersonas pievienošanas formu un pievienot korektu “Ārvalstnieka” reģistrācijas numuru vai personas kodu.
Akceptējošie kritēriji
<ul style="list-style-type: none"> <li>● Lietotāji ar tiesību “Labot” vai “Dzēst” var atvērt jaunas kontaktpersonas pievienošanas formu.</li> <li>● Lietotājam izvēloties kontaktpersonas veidu “Ārvalstnieks”, formas lauku nosaukumi mainās. Lauka nosaukums “Personas kods” mainās uz “Reģ.nr./pers.kods”.</li> <li>● Lietotājs laukā “Reģ.nr./pers.kods” var ievadīt līdz 14 simboliem (burti, cipari) garu reģistrācijas numuru vai personas kodu.</li> <li>● Lietotāji ar tiesību “Skatīt” nevar atvērt jaunas kontaktpersonas pievienošanas formu.</li> </ul>
Identifikators
DU4863

### 3.2.8. Juridiskas personas datu meklēšana pēc reģistrācijas numura un formas aizpildīšana

Lietotārstāsts
Es kā lietvedības sistēmas LIETVARIS lietotājs ar atbilstošām piekļuves tiesībām varu atvērt jaunas kontaktpersonas pievienošanas formu un pēc reģistrācijas numura ievadīšanas, nospiežot pogu, aizpildīt formas laukus ar juridiskās personas datiem.
Akceptējošie kritēriji
<ul style="list-style-type: none"><li>• Lietotāji ar piekļuves tiesībām “Labot” vai “Dzēst” var atvērt jaunas kontaktpersonas pievienošanas formu.</li><li>• Jaunas kontaktpersonas pievienošanas formas skatā, ja ir izvēlēts kontaktpersonas veids “Juridiska persona”, ir jābūt pogai “Meklēt UR”, pēc kuras nospiešanas tiks izgūti kontaktpersonas dati pēc ievadītā reģistrācijas numura.</li><li>• Lietotājs formas laukā “Reģistrācijas numurs” nevar ievadīt vairāk par 11 cipariem.<ul style="list-style-type: none"><li>○ Ja lietotājs ievada mazāk par 11 cipariem, tad lietotājam tiek atgriezts kļūdas paziņojums SP04.</li><li>○ Ja lietotājs ievada nekorektu reģistrācijas numuru, tas ir, izmantojot rakstu zīmes, lietotājam tiek atgriezts kļūdas paziņojums SP04.</li></ul></li><li>• Ja lietotājs ievada reģistrācijas numuru, kas nav uzņēmumu reģistrā, pēc pogas “Meklēt UR” nospiešanas, lietotājam tiek atgriezts kļūdas paziņojums SP05.</li><li>• Ja lietotājs ievada korektu reģistrācijas numuru, kas ir uzņēmuma reģistrā, pēc pogas “Meklēt UR” nospiešanas, formas lauki “Nosaukums” un “Adrese” tiek aizpildīti ar attiecīgās juridiskās personas datiem. Ja atlasītā uzņēmuma adrese datu bāzē nav pieejama strukturētā veidā, tad adreses lauks netiek aizpildīts (šī uzdevuma izstrāde turpināsies arī pēc kvalifikācijas darba izstrādes, tiks labota šī nepilnība).</li><li>• Lietotājam nospiežot pogu “Meklēt UR” , ja nav ievadīts reģistrācijas numurs, jāatgriež kļūdas paziņojums SP09.</li><li>• Lietotāji ar tiesību “Skatīt” nevar atvērt jaunas kontaktpersonas pievienošanas formu.</li></ul>
Identifikators
DU3282

### 3.2.9. Juridiskas personas datu brīvā meklēšana un formas aizpildīšana

Lietotājstāsts
Kā lietvedības sistēmas LIETVARIS lietotājam, man ir iespēja veikt uzņēmumu meklēšanu pēc uzņēmuma nosaukuma, adreses vai reģistrācijas numura. Pēc uzņēmuma atrašanas, es varu aizpildīt jaunas kontaktpersonas pievienošanas formu ar atlasītajiem uzņēmuma datiem.
Akceptējošie kritēriji
<ul style="list-style-type: none"><li>• Lietotāji ar piekļuves tiesībām “Labot” vai “Dzēst” var atvērt jaunas kontaktpersonas pievienošanas formu.</li><li>• Jaunas kontaktpersonas pievienošanas formas skatā, ja ir izvēlēts kontaktpersonas veids “Juridiska persona”, ir jābūt pogai “Uzņēmuma meklēšana”, pēc kuras nospiešanas tiek atgriezts dialoga logs “Uzņēmuma meklēšana”.<ul style="list-style-type: none"><li>○ Dialoga logam ir jāparādās uznirstoša dialoga loga veidā.</li><li>○ Dialoga logam ir jāsaturs ievades lauks ar nosaukumu “Brīvā meklēšana”.</li><li>○ Dialoga logā jābūt pogai “Meklēt”, pēc kuras nospiešanas notiks uzņēmuma meklēšana.</li><li>○ Dialoga logā jābūt informatīvam tekstam par meklēšanas rezultātiem.</li><li>○ Dialoga logam jāsaturs tabula ar sekojošām kolonnām:<ul style="list-style-type: none"><li>- kolonna bez nosaukuma, kas satur pogas “Pievienot”, pēc kuru nospiešanas, jaunas kontaktpersonas pievienošanas forma tiek aizpildīta ar izvēlētajiem uzņēmuma datiem, tas ir, adresi, reģistrācijas numuru un nosaukumu;</li><li>- “Nosaukums” kolonna, kas satur uzņēmuma nosaukumu;</li><li>- “Reģ.nr.” kolonna, kas satur uzņēmuma reģistrācijas numuru;</li><li>- “Adrese” kolonna, kas satur uzņēmuma adresi.</li></ul></li><li>○ Dialoga logā ir jābūt pogai “Labi”, pēc kuras nospiešanas dialoga logs aizveras.</li><li>○ Dialoga logam ir jāsaturs poga, kuras nosaukums ir simbols “X” , pēc kura nospiešanas dialoga logs tiek aizvērts.</li></ul></li><li>• Ja lietotājs ievada mazāk par 3 simboliem ievadlaukā “Brīvā meklēšana”, dialoga logā tiek atgriezts informatīvais teksts SP06.</li><li>• Ja lietotājs ievadlaukā “Brīvā meklēšana” ievada simbolus, pēc kuriem nevar atrast uzņēmumus, lietotājam dialoga logā tiek atgriezts informatīvais teksts SP07.</li></ul>

- Ja lietotājs ievadlaukā “Brīvā meklēšana” ievada vismaz 3 simbolus, pēc kuriem tiek atlasīti uzņēmumi, dialoga logā tiek atgriezts informatīvais teksts SP08.
- Dialoga logā zem tabulas ir jābūt norādītam lapu skaitam, lapas numuram un ir jābūt iespējai navigēt no vienas lapas uz citu.
- Lietotāji ar tiesību “Skatīt” nevar atvērt jaunas kontaktpersonas pievienošanas formu.
- Dialoga loga “Uzņēmuma meklēšana” saskarnes uzmetums.

Uzņēmuma meklēšana

Brīvā meklēšana  Meklēt

Lai meklētu ievadiet vismaz 3 simbolus

	Nosaukums	Reģistrācijas numurs	Adrese
Pievienot	.....	.....	.....

<< < Lapa 0 no 0 > >>

Labi

2.1.att. Dialoga loga “Uzņēmuma meklēšana” saskarnes uzmetums

Identifikators

DU3284

## 4. TRASĒJAMĪBAS TABULA

Trasējāmības tabula apraksta kvalifikācijas darbā saistību starp darba uzdevumiem, to lietotārstāstiem, saskarņu labojumiem un darba uzdevumu projektējumiem.

4.1.Tabula

**Trasējāmības tabula**

Identifikators	DU apraksts	Lietotārstāsta nodaļa	Saskarnes labojuma nodaļa	Projektējuma nodaļa
<b>1. Jaunas kontaktpersonas pievienošanas forma</b>				
DU4783	Labot pogas “Atgriezties” funkcionalitāti.	3.2.1.	5.4.4.1.	5.4.
DU1115	Labot formas izmērus, ja loga izmērs ir 1024x768 un mazāks tā, lai visus lauka nosaukumus būtu iespējams redzēt.	3.2.3.	5.4.4.2.	5.4.
DU3282	Juridisku personu meklēšana pēc reģistrācijas numura.	3.2.8.	5.2.4.1.	5.2.
DU3284	Juridisku personu brīvā meklēšana .	3.2.9.	5.3.5.1.	5.3.
<b>2. Kontaktpersonas kartīte</b>				
DU2592	Labot pogas “Atgriezties” funkcionalitāti, atverot to jaunā cilnē.	3.2.2.	5.4.4.4.	5.4.
DU2964	Labot pogas “Saglabāt” izmērus.	3.2.4.	5.4.4.3.	5.4.
DU1123	Labot šķirkļu sniegto informāciju.	3.2.5.	5.4.4.5.	5.4.
DU4406	Labot kļūdu paziņojumu pie vienādiem personas kodiem.	3.2.6.	5.4.4.6.	5.4.
DU4863	Veikt izmaiņas jaunas kontaktpersonas pievienošanas formā,	3.2.7.	5.4.4.7.	5.4.

	pievienojot jaunu ārvalstnieku. Ārvalstnieku reģistrācijas numura/personas koda ievadišana.			
--	---	--	--	--

## **5. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS**

### **5.1. Ievads**

#### **5.1.1. Nolūks**

Šis dokuments ir lietvedības sistēmas LIETVARIS funkcionālo labojumu un papildinājumu realizācijas apraksts. Programmatūras projektējuma apraksta mērķis ir nodrošināt sistēmas labojumu un papildinājumu veiksmīgu izstrādi atbilstoši izvirzītajām prasībām. Šis dokuments ir paredzēts sistēmas labojumu un papildinājumu izstrādātājam un testētājiem, kā palīgmateriāls precīzai prasību realizācijai sistēmā.

#### **5.1.2. Darbības sfēra**

Lietvedības sistēmas LIETVARIS veiktie uzlabojumi paredz paplašināt kontaktpersonu moduļa funkcionalitāti un lietotāja saskarni.

#### **5.1.3. Definīcijas**

Definīcijas un saīsinājumus skatīt nodaļā “Vārdnīca”

#### **5.1.4. Saistība ar citiem dokumentiem**

Programmatūras projektējuma apraksts balstās uz kvalifikācijas darba “Uzņēmuma datu integrācija ar lietvedības sistēmu LIETVARIS” 3.nodaļas definētiem lietotājistāstiem un apraksta to realizāciju.

Dokuments saistīts un lietojams kopā ar dokumentu “Testēšanas dokumentācija”.

#### **5.1.5. Pārskats**

Programmatūras projektējuma apraksts sastāv no 4 daļām:

- Ievada, kas ietver dokumenta nolūku, darbības sfēru, atsauci uz dokumentā izmantotajām definīcijām, kā arī saistību ar citiem dokumentiem.
- Darba uzdevuma DU3282 projektējuma, kas ietver sevī, gan datu detalizēto projektējumu, gan funkcionālo projektējumu, gan laboto saskarņu apraksts.
- Darba uzdevuma DU3284 projektējuma, kas satur gan datu detalizēto projektējumu, gan funkcionālo projektējumu, gan laboto saskarņu apraksts.

- Darba uzdevuma “Labojumi kontaktpersonu modulī” projektējums, kas apraksta veiktos uzlabojumus kontaktpersonu modulī, datu detalizēto projektējumu un satur laboto saskarņu aprakstu.

## 5.2. Darba uzdevuma DU3282 projektējums

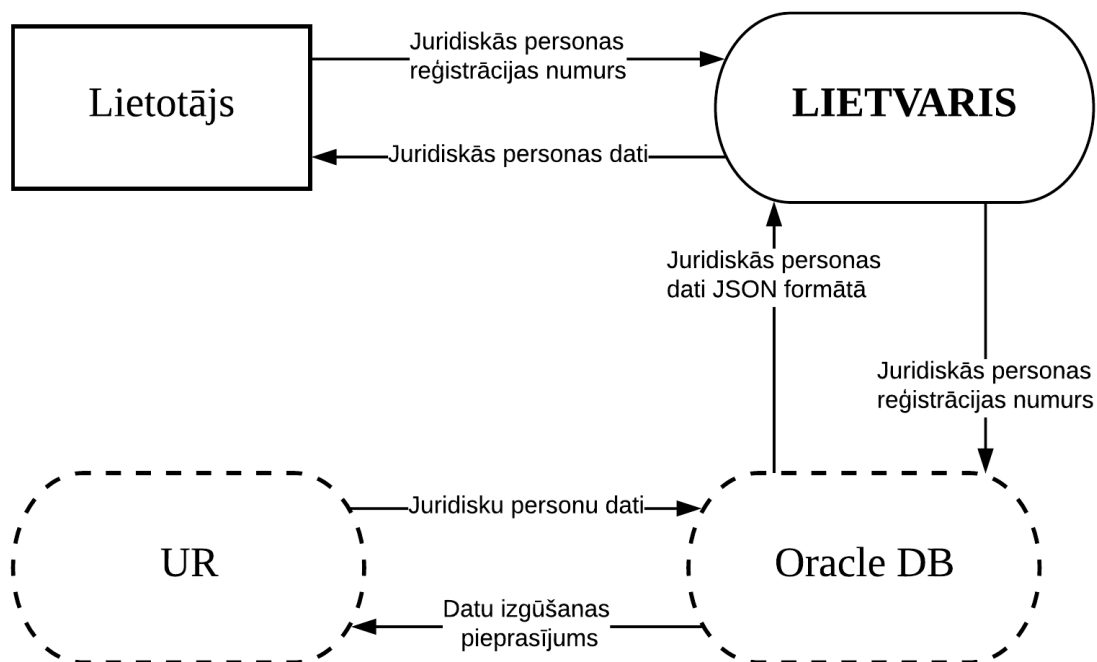
Identifikators: DU3282

Darba uzdevuma mērķis ir papildināt jaunas kontaktpersonas pievienošanas formu pie kontaktpersonas veida “Juridiska persona”, pievienojot jaunu pogu “Meklēt UR”, pēc kuras nospiešanas, ja ir ievadīts korekts reģistrācijas numurs, tiek aizpildīti formas lauki ar attiecīgās juridiskās personas datiem.

Ņemot vērā lietvedības sistēmas LIETVARIS izmantotās tehnoloģijas un to implementāciju, tika nolemts, ka darba uzdevuma izstrādē tiks izmantota *Oracle* datu bāze, *Microsoft .NET Core* ietvars un *JavaScript* valodas ietvars *AngularJS*.

Shēma 5.1. attēlo sadarbību starp lietvedības sistēmu LIETVARIS, sistēmas lietotāju un ārējām sistēmām, kas tiek izmantotas darba uzdevuma DU3282 implementācijā.

No *Uzņēmumu reģistra* (turpmāk tekstā - UR) atvērtajiem datiem tiek izgūti dati par uzņēmumiem un tiek aizpildītas lietvedības sistēmas LIETVARIS datu bāzes tabulas. Lietotājs LIETVARIS sistēmā ievada juridiskās personas reģistrācijas numuru. LIETVARIS uz *Oracle* datu bāzi sūta datu izgūšanas pieprasījumu pēc lietotāja ievadītā reģistrācijas numura. *Oracle* datu bāze saņem pieprasījumu un atgriež atlasītos uzņēmuma datus JSON formātā. LIETVARIS sistēma apstrādā saņemtos juridiskās personas datus. Lietotājam juridiskās personas dati attēlojas jaunas kontaktpersonas pievienošanas formā.



5.1.att. Darba uzdevuma DU3282 shēma

## 5.2.1. Datu detalizētais projektējums

Šajā nodaļā ir aprakstītas tabulas, kuras darba autoram nācās izmantot, izstrādājot darba uzdevumu DU3282.

### 5.2.1.1. Tabula IN\_NMAKS\_JP

Tabula IN\_NMAKS\_JP paredzēta, lai glabātu tajā nodokļu maksātāju (turpmāk dokumentā - NM) pamatdatus. Programmā šī tabula tiek izmantota, lai veidotu vaicājumus un atlasītu vajadzīgo informāciju.

5.1. Tabula

Tabula IN\_NMAKS\_JP, kurā tiek glabāti NM pamatdati

Kolonnas nosaukums	Datu tips	Apraksts
ID_NM	NUMBER(10)	Unikāls ierakstu identifikators
UR_NOS	NUMBER(10)	NM juridiskais nosaukums
NMR_KODS	VARCHAR2(11)	Vienotais NM reģistrācijas numurs
AKTIVS	NUMBER(1)	Ieraksta aktivitātes pazīme (-1 aktīvs, 0 neaktīvs, 1 dzēsts)

\*Norādīti tikai izmantotie tabulas IN\_NMAKS\_JP lauki.

### 5.2.1.2.Tabula IN\_SAITEADRPERS

Tabula paredzēta, lai tajā uzturētu NM nestrukturētās adreses. IN\_SAITEADRPERS tabula programmkodā tiek izmantota kopā ar tabulām ADRESES, IN\_KL\_ADRVEIDS un IN\_NMAKS\_JP, lai veidotu vaicājumus.

5.2.Tabula

**Tabula IN\_SAITEADRPERS, kurā tiek glabātas NM nestrukturētās adreses**

Kolonnas nosaukums	Datu tips	Apraksts
SAPI_JURPERSID	NUMBER(10)	Personas identifikators, saite ar IN_NMAKS_JP.ID_NM
SAPI_AID	NUMBER(10)	Strukturētās adreses identifikators, saite ar ADRESES.AID
SAPI_ADRVEIDS	NUMBER(10)	Adreses veids, saite ar IN_KL_ADRVEIDS.AV_ID
SAPI_AKTIVS	NUMBER(1)	Ieraksta aktivitātes pazīme

\*Norādīti tikai izmantotie tabulas IN\_SAITEADRPERS lauki.

### 5.2.1.3.Tabula ADRESES

Tabula paredzēta, lai glabātu personu strukturētās adreses. Tabula ADRESES programmkodā tiek izmantota kopā ar tabulām IN\_SAITEADRPERS un CL\_COUNTRIES, lai vaicājumā atlasītu personu strukturētās adreses.

5.3.Tabula

**Tabula ADRESES, kurā tiek glabātas personu strukturētās adreses**

Kolonnas nosaukums	Datu tips	Apraksts
AID	NUMBER(10)	Unikāls adreses identifikators
A_VAID_IR	NUMBER(10)	Unikāls valsts identifikators
ADRPASTAINDEX	VARCHAR2(10)	Pasta indekss
APILS	VARCHAR2(160)	Rajona, pagasta, pilsētas nosaukums
AADR	VARCHAR2(80)	Ielas vai mājas nosaukums

\*Norādīti tikai izmantotie tabulas ADRESES lauki.

#### 5.2.1.4.Tabula IN\_KL\_ADRVEIDS

Tabula IN\_KL\_ADRVEIDS paredzēta adrešu veidu klasifikatoru uzturēšanai.

5.4.Tabula

Tabula IN\_KL\_ADRVEIDS, kurā tiek glabāti adrešu veidu klasifikatori

Kolonnas nosaukums	Datu tips	Apraksts
AV_ID	NUMBER(10)	Unikāls ierakstu identifikators
AV_KODS	VARCHAR2(10)	Ieraksta kods
AV_AKTIVS	NUMBER(1)	Ieraksta aktivitātes pazīme

\*Norādīti tikai izmantotie tabulas IN\_KL\_ADRVEIDS lauki.

#### 5.2.1.5.Tabula CL\_COUNTRIES

Tabula paredzēta, lai glabātu tajā valsts kodus ar to nosaukumiem latviešu valodā un angļu valodā.

5.5.Tabula

Tabula CL\_COUNTRIES, kurā tiek glabāti valsts kodi ar to nosaukumiem

Kolonnas nosaukums	Datu tips	Apraksts
VAKD	NUMBER(10)	Unikāls valsts identifikators
VANOSLV	VARCHAR2(39)	Valsts nosaukums latviešu valodā

\*Norādīti tikai izmantotie tabulas CL\_COUNTRIES lauki.

### 5.2.2. Funkcionālais projektējums

Šajā nodaļā ir aprakstītas pakotnes un funkcijas, kuras darba autoram nācās izmantot vai izveidot, izstrādājot darba uzdevumu DU3282.

#### 5.2.2.1.Pakotne I\_UR\_INTEGRATION\_PCK

Pakotne I\_UR\_INTEGRATION\_PCK ir paredzēta juridisku personu datu izgūšanai pēc lietotāja ievadītiem parametriem, tas ir, reģistrācijas numura.

Pakotne ietver sevī procedūru *GetDataUsingRegCode*, kas nodrošina juridisku personu datu izgūšana pēc reģistrācijas numura.

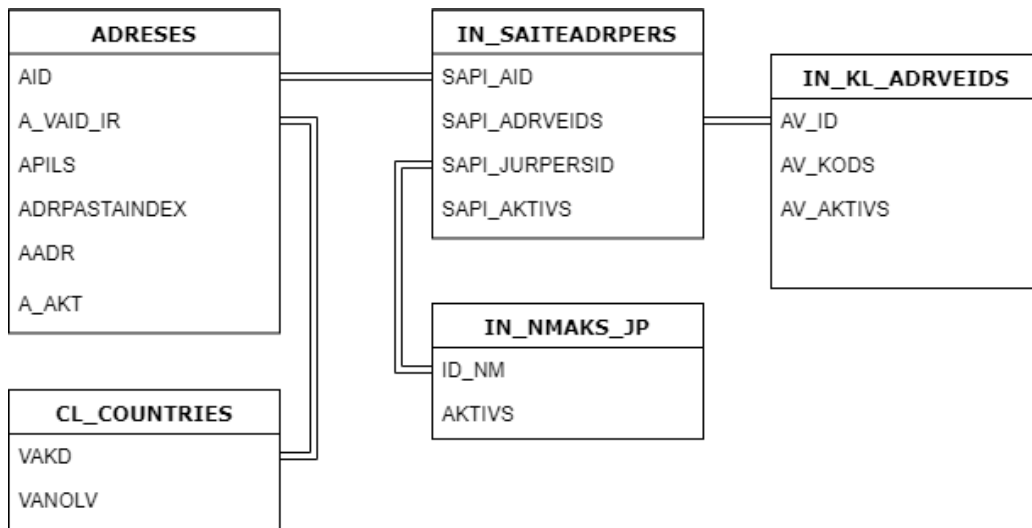
Pakotnes I\_UR\_INTEGRATION\_PCK procedūrā *GetDataUsingRegCode* tiek veikti vaicājumi, kas visus atlasītos datus ielasa objektos, kas pēc tam tiek izmantoti, lai noformētu JSON objektu, kuru arī beigās atgriež. Procedūras aprakstu skatīt 5.6. tabulā.

5.6.Tabula

#### Procedūras *GetDataUsingRegCode* apraksts

Mērķis
Izgūt uzņēmuma datus pēc reģistrācijas numura.
Apraksts
<p>1. Pārbaudes:</p> <ul style="list-style-type: none"> <li>• vai reģistrācijas numurs ir ievadīts;</li> <li>• vai reģistrācijas numurs validējas, tas ir, ir korekts;</li> <li>• vai ievadītais reģistrācijas numurs ir atrodams datu bāzē.</li> </ul> <p>2. Tiek atlasīti vajadzīgie juridiskās personas dati:</p> <ul style="list-style-type: none"> <li>• juridiskās personas nosaukums;</li> <li>• juridiskās personas reģistrācijas numurs;</li> <li>• juridiskās personas strukturētā adrese: <ul style="list-style-type: none"> <li>○ adreses identifikators;</li> <li>○ ielas nosaukums;</li> <li>○ pilsētas nosaukums;</li> <li>○ pasta indekss;</li> <li>○ valsts identifikators;</li> <li>○ valsts nosaukums.</li> </ul> </li> </ul> <p>Strukturētās adreses vaicājuma shematisko attēlojumu skatīt 5.2.attēlā.</p> <p>3. Atlasītie juridiskās personas dati tiek ievietoti JSON objektā.</p>
Izvaddati
JSON objekts, kurā ir saglabāti dati par juridisko personu.

Funkcijas *GetDataUsingRegCode* programmkodu skatīt 7.pielikumā. Atgriezto JSON objekta struktūru skatīt 10.pielikumā.



5.2.att. Tabulu sasaistes shēma procedūras *GetDataUsingRegCode* strukturētās adreses vaicājumam

### 5.2.2.2.Pakotne APP\_URINTEGRATION\_PCK

Pakotne APP\_URINTEGRATION\_PCK ir paredzēta juridisku personu datu izgūšanai pēc lietotāja ievadītiem parametriem, tas ir, reģistrācijas numuriem un juridiskās personas datu nodošanai uz *Elasticsearch* indeksu.

Pakotnes APP\_URINTEGRATION\_PCK procedūra *GetDataUsingRegCode* izsauc pakotnes I\_UR\_INTEGRATION\_PCK procedūru *GetDataUsingRegCode*. Šāda izsaukšana ir vajadzīga, jo pakotne I\_UR\_INTEGRATION\_PCK nav pieejama klienta puses tehnoloģijām, atšķirībā no pakotnes APP\_URINTEGRATION\_PCK.

Procedūras *GetDataUsingRegCode* funkcionālo aprakstu skatīt 5.6.tabulā.

### 5.2.2.3.Funkcija *findCompany*

Funkcija, pēc formā ievadīta reģistrācijas numura, no datu bāzes izgūst un atgriež formas lauku 'Nosaukums' un 'Adrese' datus. Šī funkcija ir rakstīta *JavaScript* valodā.

Lai nodrošinātu šīs funkcijas izpildi, tika izveidots savienojums starp .NET un Oracle datu bāzes pakas APP\_URINTEGRATION\_PCK procedūru *GetDataUsingRegCode* (skatīt 2.pielikumu).

**Funkcijas *findCompany* apraksts**

Mērķis
Atgriezt juridiskās personas datus jaunas kontaktpersonas pievienošanas formā.
Apraksts
<p>1. Pārbaude:</p> <ul style="list-style-type: none"> <li>• vai ir atstāts tukšs “Reģistrācijas numurs” ievadlauks;</li> <li>• vai ievadlauka “Reģistrācijas numurs” saturs ir nenoteikts (<i>no angļu valodas – “undefined”</i>);</li> <li>• ja viena no augstāk minētajām pārbaudēm ir patiesa, tad tiek atgriezts kļūdas paziņojums SP09, citādāk tiek izpildīts 2.punkts.</li> </ul> <p>2. No datu bāzes procedūras <i>GetDataUsingRegCode</i> (skatīt 5.6.tabulu) tiek izgūts JSON objekts pēc formā ievadītā reģistrācijas numura.</p> <p>3. Ja atgrieztā objekta statuss ir :</p> <ul style="list-style-type: none"> <li>• ‘ok’ jeb pozitīvs, tad no datu bāzes izgūtā JSON objekta tiek izgūti vajadzīgie dati (juridiskās personas nosaukums un adrese) un aizpildīti jaunas kontaktpersonas pievienošanas formas lauki “Nosaukums” un “Adrese”.</li> <li>• ‘error’ jeb negatīvs, tad jaunas kontaktpersonas pievienošanas lauku “Nosaukums” un “Adrese” saturs tiek noņemts un tiek atgriezts kļūdas paziņojums SP04 vai SP05.</li> </ul>
Izvaddati
Jaunas kontaktpersonas pievienošanas formas lauku “Nosaukums” un “Adrese” dati.

### 5.2.3. LIETVARIS projekta gatavie risinājumi

Lai nodrošinātu koda kvalitāti un ievērotu projekta LIETVARIS programmēšanas principus, darba uzdevuma DU3282 izstrādē tika izmantoti jau gatavi risinājumi.

5.8.Tabula

#### LIETVARIS projekta gatavie risinājumi

Nr.	Pakotne	Funkcija, procedūra	Apraksts
1.	U_JSON_PCK	<i>SetStringProperty</i>	Ieraksta JSON objektā tekstu
		<i>SetIntProperty</i>	Ieraksta JSON objektā veselu skaitļu vērtības
		<i>writeToClob</i>	Veic JSON objekta konvertāciju uz NCLOB datu tipu
2.	I_UR_INTEGRATION_PCK	<i>InternalFormFullAddress</i>	Formatē strukturēto adresi
3.	U_ERROR_PCK	<i>AddErrorToList</i>	Kļūdu ziņojumu sarakstam pievieno kļūdu

### 5.2.4. Darba uzdevuma DU3282 uzlabojumu saskarnes

Nodaļā ir aprakstīti visi veiktie LIETVARIS sistēmas saskarnes labojumi, darba uzdevuma DU3282 implementācijai.

#### 5.2.4.1. Juridisku personu meklēšana pēc reģistrācijas numura

Apraksts: Papildināt jaunas kontaktpersonas pievienošanas formu ar iespēju meklēt juridiskas personas datus pēc reģistrācijas numura.

Identifikators: DU3282

5.3.attēlā redzama jaunas kontaktpersonas pievienošanas forma.

5.9.Tabula

**DU3282 saskarnes labojumu apraksts, 1.daļa**

Nr.	Labotā vai izveidotā datne	Veiktās izmaiņas
1.	<i>Edit.cshtml.</i>	Datnei pievienota papildus programmkoda rindiņa, kas definē pogu “Meklēt UR” un funkcijas izsaukumu, kas tiks izpildīta, nospiežot to. Poga noformēta vadoties pēc LIETVARIS projekta vadlīnijām.

- Ar 1 apzīmēta pievienotā poga, pēc kuras nospiešanas notiks juridisku personu meklēšana.

Atgriezties

Jauns notikums

Kontaktpersonas veids: Juridiska persona

Nosaukums:\*

Reģistrācijas numurs: Meklēt UR 1

Adrese:

Tālrunis:

Fakss:

E-pasts:

Amats:

Piezīmes:

5.3.att. Jaunas kontaktpersonas pievienošanas forma

5.4.attēlā redzama jaunas kontaktpersonas forma un kļūdas paziņojums.

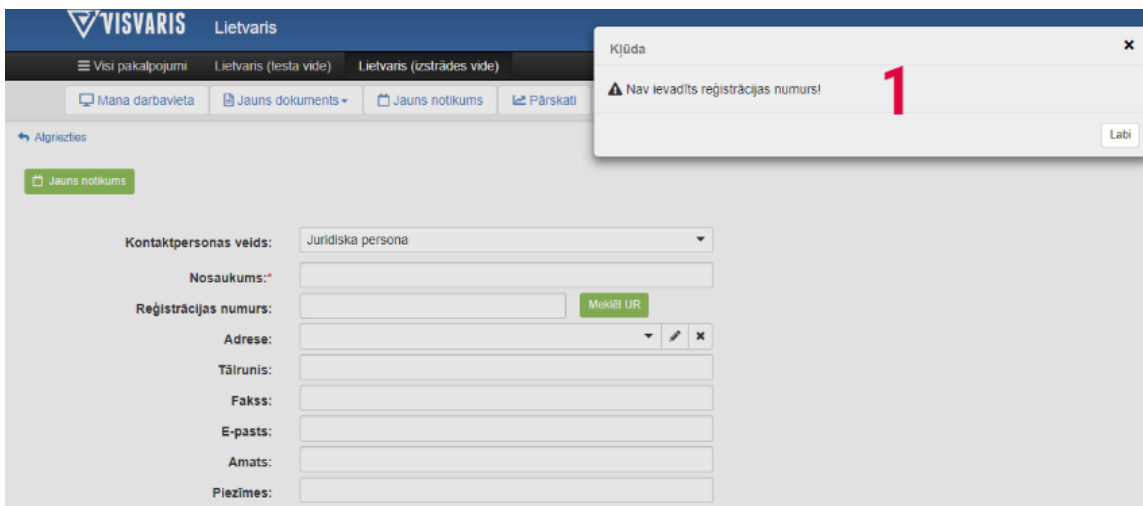
5.10.Tabula

**DU3282 saskarnes labojumu apraksts, 2.daļa**

Nr.	Labotā vai izveidotā datne	Veiktās izmaiņas
1.	<i>Contact.person.edit.controller.js</i>	Datnei funkcijas <i>findCompany</i> programmkodā pievienota papildus pārbaude, kas pārbauda, vai ievadlauks

		“Reģistrācijas numurs” ir tukšs. Ja pārbaude izpildās, tiek atgriezts kļūdas paziņojums.
--	--	--

- Ar 1 apzīmēts kļūdas paziņojums, kas attēlojās, kad nav ievadīts reģistrācijas numurs un tiek nospiesta poga “Meklēt UR”.



5.4.att. Jaunas kontaktpersonas pievienošanas forma

5.5.attēlā redzama jaunas kontaktpersonas pievienošanas forma un kļūdas paziņojums.

5.11.Tabula

#### DU3282 saskarnes labojumu apraksts, 3.daļa

Nr.	Labotā vai izveidotā datne	Veiktās izmaiņas
1.	<i>Pakotne I_UR_INTEGRATION_PCK</i>	Pakotnes procedūrai <i>GetDataUsingRegCode</i> ir izstrādāta datu ievades pārbaude, kas pārbauda ievadītā reģistrācijas numura korektumu (11 simboli, tikai cipari). Ja pārbaude neizpildās, tiek atgriezts kļūdas paziņojums.
2.	<i>Pakotne I_SERVER_SETUP</i>	Pakotnes procedūrai <i>populateErrors</i> pievienota papildus programmkoda rindiņa, kas definē jaunu kļūdas

		paziņojumu, kas tiks atgriezts nekorekta reģistrācijas numura ievades gadījumā.
3.	<i>Edit.cshtml.</i>	Ievadlauka “Reģistrācijas numurs ” definējumā tiek pielikts papildus nosacījums – maksimālais simbolu skaits ir 11.

- Ar 1 atzīmēts nekorekti ievadīts reģistrācijas numurs.
- Ar 2 atzīmēts kļūdas paziņojums, kas vēsta par nekorekti ievadītu reģistrācijas numuru.

The screenshot shows the VISVARIS web application interface. At the top, there is a navigation bar with 'Lietvaris' and several menu items. Below the navigation bar, there are tabs for 'Mana darbavieta', 'Jauns dokuments', 'Jauns notikums', and 'Pārskati'. The main content area displays a form for adding a new contact person. The form includes fields for 'Kontaktpersonas veids' (set to 'Juridiska persona'), 'Nosaukums', 'Reģistrācijas numurs' (containing '4000327846' and marked with a red '1'), 'Adrese', 'Tālrunis', 'Fakss', 'E-pasts', 'Amats', and 'Piezīmes'. An error message box is overlaid on the form, stating 'Kļūda' and 'Ievadīts nekorekts reģistrācijas numurs!' with a red '2' next to it. A 'Labi' button is visible in the bottom right corner of the error message box.

5.5.att. Jaunas kontaktpersonas pievienošanas forma

5.6.attēlā redzama jaunas kontaktpersonas pievienošanas forma un kļūdas paziņojums.

5.12.Tabula

#### DU3282 saskarnes labojumu apraksts, 4.daļa

Nr.		Labotā vai izveidotā datne	Veiktās izmaiņas
1.		<i>Pakotne</i> I_UR_INTEGRATION_PCK	Pakotnes procedūrai <i>GetDataUsingRegCode</i> ir izstrādāta datu ievades pārbaude, kas pārbauda ievadītā reģistrācijas numura eksistenci uzņēmuma reģistra datu bāzē. Ja pārbaude

			neizpildās, tiek atgriezts kļūdas paziņojums.
2.		<i>Pakotne I_SERVER_SETUP</i>	Pakotnes procedūrai <i>populateErrors</i> pievienota papildus programmkoda rindiņa, kas definē jaunu kļūdas paziņojumu, kas tiks atgriezts, ja ievadītais reģistrācijas numurs nav atrodams uzņēmuma reģistra datu bāzē.

- Ar 1 atzīmēts reģistrācijas numurs, kas nav uzņēmumu reģistrā
- Ar 2 atzīmēts kļūdu paziņojums, kas ziņo, ka pēc ievadītā reģistrācijas numura kontaktpersona nav atrasta.

### 5.6.att. Jaunas kontaktpersonas pievienošanas forma

5.7.attēlā redzama jaunas kontaktpersonas pievienošanas forma.

## DU3282 saskarnes labojumu apraksts, 5.daļa

Nr.	Labotā vai izveidotā datne	Veiktās izmaiņas
1.	<i>Contact.person.edit.controller.js</i>	Izveidota funkcija <i>findCompany</i> , kas pēc atlasītajiem juridiskās personas datiem, aizpilda jaunas kontaktpersonas pievienošanas formu. Funkcijas aprakstu skatīt 4.7.tabulā.

- Ar 1 atzīmēta forma, kura tika aizpildīta pēc korekta reģistrācijas numura ievades un pogas “Meklēt UR” nospiešanas.

< [Atgriezties](#)  
 Jauns notikums

Kontaktpersonas veids: Juridiska persona  
 Nosaukums\*: Sabiedrība ar ierobežotu atbildību "ZZ Dats"  
 Reģistrācijas numurs: 40003278467 Meklēt UR  
 Adrese: Elizabetes iela 41/43, Rīga, LV-1010  
 Tālrunis:  
 Fakss:  
 E-pasts:  
 Amats:  
 Piezīmes:

5.7.att. Aizpildīta jaunas kontaktpersonas pievienošanas forma

### 5.3. Darba uzdevums DU3284 projektējums

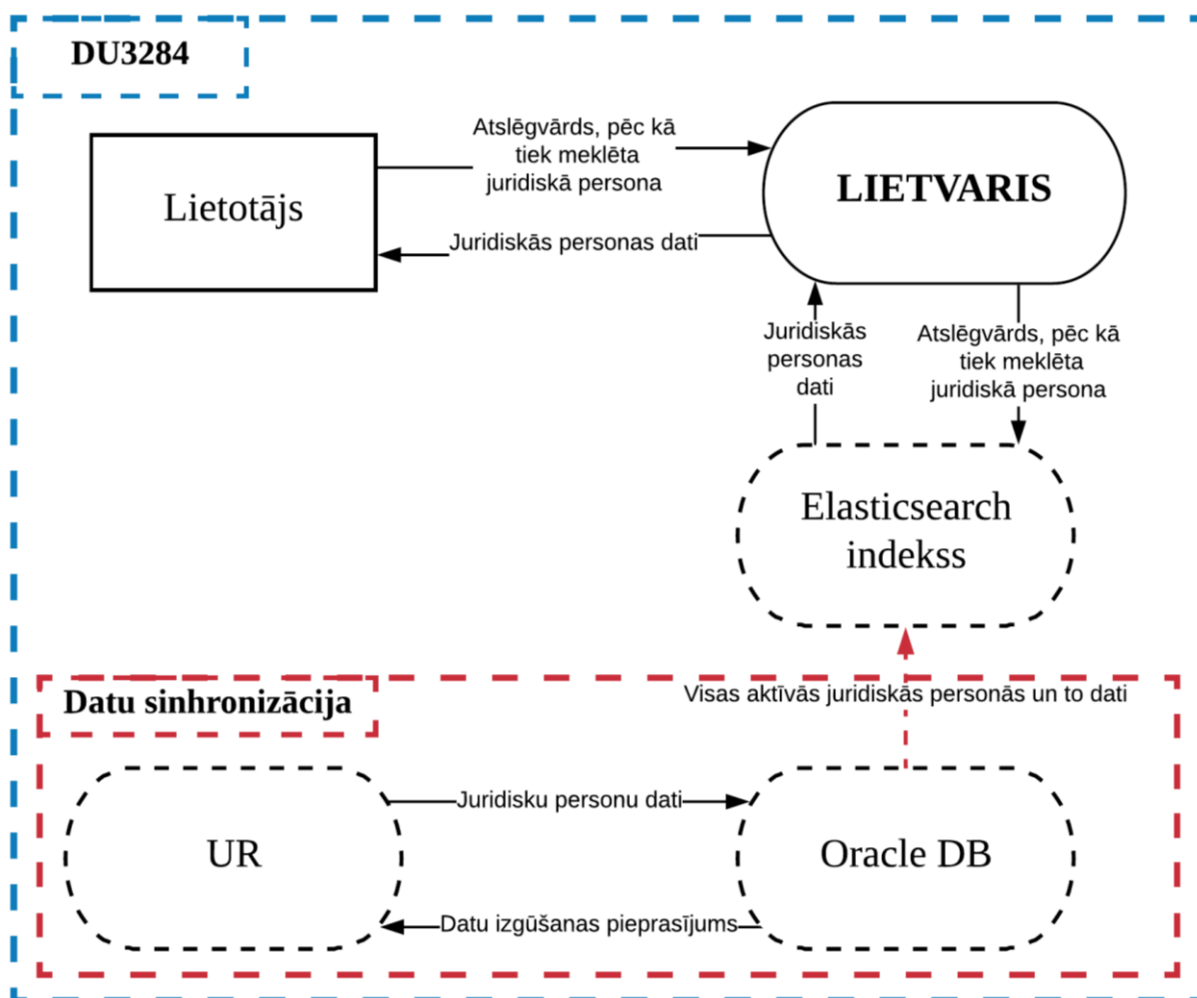
Identifikators: DU3284

Uzdevuma mērķis ir papildināt jaunas kontaktpersonas pievienošanas formu pie kontaktpersonas veida “Juridiska persona” ar pogu “Uzņēmuma reģistra dati”, pēc kuras nospiešanas tiek atgriezts dialoga logs, kurā, ievadot atslēgvārdus, var izvēlēties vajadzīgo juridisko personu un aizpildīt formu ar izvēlētās juridiskās personas datiem.

Darba uzdevums implementēts, izmantojot *Oracle* datu bāzi, *Elasticsearch* dokumentu glabātuvī, *Microsoft .NET Core* ietvaru un *JavaScript* ietvaru *AngularJS*. Lai nodrošinātu datu meklēšanu, izmantots *Elasticsearch* risinājums, jo datu atlase pēc brīvi ievadīta teksta notiek būtiski ātrāk, nekā datu bāzē.

Shēma 5.8. attēlo sadarbību starp LIETVARIS sistēmu, sistēmas lietotāju un ārējām sistēmām – *Oracle* datu bāzi, *UR* atvērtajiem datiem un *Elasticsearch* dokumentu glabātuvi.

No *UR* atvērtajiem datiem tiek izgūti dati par uzņēmumiem un tiek aizpildītas lietvedības sistēmas LIETVARIS datu bāzes tabulas. Datu bāze sinhronizējas ar *Elasticsearch* indeksu un nosūta visus datus uz to. Lietotājs LIETVARIS sistēmā brīvās meklēšanas ievadlaukā ievada atslēgvārdus pēc kā meklēs juridisko personu. LIETVARIS sūta datu izgūšanas pieprasījumu uz *Elasticsearch* indeksu pēc lietotāja ievadītā atslēgvārda. *Elasticsearch* indekss saņem pieprasījumu, atlasa vajadzīgos datus un atgriež atlasīto juridisku personu datus (JSON formātā). LIETVARIS apstrādā saņemtus datus un atgriež tos lietotājam dialoga loga tabulā.



5.8.att. Darba uzdevumam DU3284 shēma

### 5.3.1. Datu detalizēts projektējums

Šajā nodaļā ir aprakstītas tabulas, kuras darba autoram nācās izmantot, izstrādājot darba uzdevumu DU3284.

#### 5.3.1.1. Tabula ES\_SYNC

Tabula ES\_SYNC paredzēta, lai apkopotu *Elasticsearch* sinhronizācijas darbu definīcijas. Darba izstrādātājs izmantoja ES\_SYNC tabulu, lai saglabātu tajā procedūru, kuras programmkodā tiek atjaunoti *Elasticsearch* indeksa dati. Ieliekot procedūru ES\_SYNC tabulā, izstrādātājs panāca, ka procedūra tiek kompilēta, tas ir, notiek datu pārsinhronizācija vienu reizi diennaktī (biežums atkarīgs no uzstādītās SYNC\_INTERVAL vērtības).

5.14. Tabula

Tabula ES\_SYNC, kurā tiek glabātas procedūras priekš *Elasticsearch* indeksu sinhronizācijas

Kolonnas nosaukums	Datu tips	Apraksts
PROCEDURE_NAME	NVARCHAR2(100)	Kāda procedūra tika izpildīta, lai sinhronizētu datus uz <i>Elasticsearch</i>
SYNC_INTERVAL	INTERVAL DAY(2) TO SECOND(6)	Cik bieži jāveic sinhronizācija
LAST_DATE	DATE	Pēdējā sinhronizācija
LAST_STATUSS	NVARCHAR2(200)	Pēdējās izpildes statuss
LAST_ERROR	NVARCHAR2(2000)	Pēdējās sinhronizācijas kļūda

#### 5.3.1.2. Tabula IN\_NMAKS\_JP

Tabula IN\_NMAKS\_JP paredzēta, lai apkopotu tajā NM pamatdatus. Programmkodā tabula tiek izmantota, lai izgūtu no tās informāciju par juridisko personu un veidotu relāciju ar tabulām, no kurām nepieciešams izgūt atlikušos datus par juridisko personu.

5.15. Tabula

Tabula IN\_NMAKS\_JP, kurā tiek glabāti NM pamatdati

Kolonnas nosaukums	Datu tips	Apraksts
NMR_KODS	VARCHAR2(11)	Vienotais NM reģistrācijas numurs
NM_PEDDAT	DATE	Pēdējo izmaiņu datums

AKTIVS	NUMBER(1)	Ieraksta aktivitātes pazīme (- 1 aktīvs, 0 neaktīvs, 1 dzēsts)
ID_NM	NUMBER(10)	Unikāls ierakstu identifikators
JUR_NOS	NUMBER(10)	NM juridiskais nosaukums

\*Norādīti tikai izmantotie tabulas IN\_NMAKS\_JP lauki.

### 5.3.1.3.Tabula IN\_SAITEADRPERS

Tabula paredzēta, lai tajā glabātu NM nestrukturētās adreses. IN\_SAITEADRPERS tabula programmkodā tiek izmantota kopā ar tabulām ADRESES, IN\_KL\_ADRVEIDS un IN\_NMAKS\_JP, lai veidotu vaicājumus.

5.16.Tabula

**Tabula IN\_SAITEADRPERS, kurā tiek glabātas NM nestrukturētās adreses**

Kolonnas nosaukums	Datu tips	Apraksts
SAPI_JURPERSID	NUMBER(10)	Personas identifikators, saite ar IN_NMAKS_JP.ID_NM
SAPI_AID	NUMBER(10)	Strukturētās adreses identifikators, saite ar ADRESES.AID
SAPI_ADRVEIDS	NUMBER(10)	Adreses veids, saite ar IN_KL_ADRVEIDS.AV_ID
SAPI_AKTIVS	NUMBER(1)	Ieraksta aktivitātes pazīme

\*Norādīti tikai izmantotie tabulas IN\_SAITEADRPERS lauki.

### 5.3.1.4.Tabula ADRESES

Tabula paredzēta, lai glabātu personas strukturēto adresi. ADRESES tabula darbā tiek izmantota kopā ar tabulu IN\_SAITEADRPERS, lai izgūtu juridiskās personas adresi.

5.17.Tabula

**Tabula ADRESES, kurā tiek glabātas personu strukturētās adreses**

Kolonnas nosaukums	Datu tips	Apraksts
AID	NUMBER(10)	Unikāls adreses identifikators
A_VOID_IR	NUMBER(10)	Unikāls valsts identifikators

ADRPASTAINDEX	VARCHAR2(10)	Pasta indekss
APILS	VARCHAR2(160)	Rajona, pagasta, pilsētas nosaukums
AADR	VARCHAR2(80)	Ielas vai mājas nosaukums

\*Norādīti tikai izmantotie tabulas ADRESES lauki.

### 5.3.1.5.Tabula IN\_KL\_ADRVEIDS

Tabula IN\_KL\_ADRVEIDS paredzēta adrešu veidu klasifikatoru uzglabāšanai. Darbā tiek izmantota kopā ar tabulu IN\_SAITEADRPERS.

5.18.Tabula

**Tabula IN\_KL\_ADRVEIDS, kurā tiek glabāti adrešu veidu klasifikatori**

Kolonnas nosaukums	Datu tips	Apraksts
AV_ID	NUMBER(10)	Unikāls ierakstu identifikators
AV_KODS	VARCHAR2(10)	Ieraksta kods
AV_AKTIVS	NUMBER(1)	Ieraksta aktivitātes pazīme

\*Norādīti tikai izmantotie tabulas IN\_KL\_ADRVEIDS lauki.

### 5.3.1.6.Tabula CL\_COUNTRIES

Tabula paredzēta, lai glabātu tajā valsts kodus ar to nosaukumiem latviešu valodā un angļu valodā. Programmkode tabula tiek izmantota, lai izgūtu valsts identifikatoru un valsts nosaukumu latviešu valodā.

5.19.Tabula

**Tabula CL\_COUNTRIES, kurā tiek glabāti valsts kodi ar to nosaukumiem**

Kolonnas nosaukums	Datu tips	Apraksts
VAKD	NUMBER(10)	Unikāls valsts identifikators
VANOSLV	VARCHAR2(39)	Valsts nosaukums latviešu valodā

\*Norādīti tikai izmantotie tabulas CL\_COUNTRIES lauki.

## 5.3.2. Funkcionālais projektējums

Šajā nodaļā ir aprakstītas pakotnes un funkcijas, kuras darba autoram nācās izmantot vai izveidot, izstrādājot darba uzdevumu DU3284.

### 5.3.2.1.Pakotne APP\_URINTEGRATION\_PCK

Pakotne APP\_URINTEGRATION\_PCK ir paredzēta juridisku personu datu izgūšanai pēc lietotāja ievadītā reģistrācijas numura un uzņēmuma datu nodošanai uz *Elasticsearch* indeksu.

Pakotne APP\_URINTEGRATION\_PCK satur procedūru *CompaniesSyncWithIndex*, kas nodrošina darba uzdevuma DU3284 juridisko personu datu nodošanu uz *Elasticsearch* indeksu.

5.20.Tabula

#### Procedūras *CompaniesSyncWithIndex* apraksts

Mērķis
Nodot juridisku personu datus uz <i>Elasticsearch</i> indeksu.
Apraksts
<ol style="list-style-type: none"><li>1. Tiek izgūti juridisko personu dati, kurus sūtīs uz <i>Elasticsearch</i> indeksu (visi aktīvie juridisko personu dati – nosaukums, adrese, reģistrācijas numurs). Juridisko personu dati tiek izgūti, izmantojot pakotnes APP_URINTEGRATION_PCK procedūru <i>GetDataUsingRegCode</i>, kuras aprakstu skatīt 5.6.tabulā.</li><li>2. Pagaidu NCLOB mainīgais tiek aizpildīts ar juridisko personu datiem.</li><li>3. Atlasa 50 juridisko personu datus un sūta tos uz <i>Elasticsearch</i> indeksu (dati tiek sūtīti pa 50 juridisko personu datiem vienā reizē).</li><li>4. Procedūras beigās tiek veikta pārbaude, vai ir vēl palikuši kādi juridisko personu dati pagaidu NCLOB mainīgajā, kas nav aizsūtīti:<ul style="list-style-type: none"><li>• ja ir palikuši vēl neaizsūtītie dati, tad tos nosūta;</li><li>• ja visi dati ir nosūtīti, tad nedara neko.</li></ul></li></ol>
Izvaddati
Atlasīto juridisku personu dati tiek nosūtīti uz <i>Elasticsearch</i> indeksu.

### 5.3.2.2.Pakotne COMPANY\_ES\_IMPORT\_PCK

Pakotne COMPANY\_ES\_IMPORT\_PCK ir paredzēta juridisko personu datu atjaunošanai un sinhronizācijai ar *Elasticsearch* indeksu.

Pakotne COMPANY\_ES\_IMPORT\_PCK satur procedūru *UpdateAll*. Procedūras aprakstu skatīt 5.21.tabulā.

### Procedūras *UpdateAll* funkcionālais apraksts

Mērķis
Atjaunot juridisku personu datus <i>Elasticsearch</i> indeksā.
Apraksts
<ol style="list-style-type: none"> <li>1. Tiek veidots pagaidu NCLOB, kas tiek aizpildīts ar juridisko personu datiem (nosaukums, reģistrācijas numurs, adrese), kuru aktivitātes statuss kopš pēdējās datu atjaunošanas datuma (salīdzina ar tabulas IN_NMAKS_JP lauku NM_PEDDAT) ir mainījies uz 1 (likvidēts uzņēmums). Juridisko personu dati tiek izgūti, izmantojot pakotnes APP_URINTEGRATION_PCK procedūru <i>GetDataUsingRegCode</i>, kuras aprakstu skatīt 5.6.tabulā.</li> <li>2. Sūta dzēsto juridisko personu datus uz <i>Elasticsearch</i> indeksu, lai indeksā atjaunotu datus un izdzēstu neaktīvās juridiskās personas. Vienā datu atjaunošanas reizē tiek dzēsti 100 juridisko personu dati (sūta pa 100 juridisko personu datiem). Pēc katras sūtīšanas reizes NCLOB mainīgais tiek iztīrīts, lai nerastos atmiņas problēmas.</li> <li>3. Pārbauda, vai ir palicis vēl kāds neaktīvais ieraksts, kas nav nosūtīts uz <i>Elasticsearch</i> indeksu: <ul style="list-style-type: none"> <li>• ja ir, tad nosūta to uz <i>Elasticsearch</i> indeksu;</li> <li>• ja visi neaktīvie dati ir nosūtīti, tad nedara neko.</li> </ul> </li> <li>4. Pagaidu izveidotais NCLOB mainīgais tiek aizpildīts ar juridisko personu datiem(nosaukums, reģistrācijas numurs, adrese), kuru dati ir mainījušies pēc pēdējās datu atjaunošanas reizes (salīdzina ar tabulas IN_NMAKS_JP lauku NM_PEDDAT).</li> <li>5. Uz <i>Elasticsearch</i> tiek sūtīti atlasītie juridisko personu dati, kuru datus <i>Elasticsearch</i> indeksā ir nepieciešams atjaunot. Vienā datu atjaunošanas reizē tiek atjaunoti 100 juridisko personu dati. Pēc katras sūtīšanas reizes NCLOB mainīgais tiek iztīrīts, lai nerastos atmiņas problēmas.</li> <li>6. Pārbauda, vai ir palikuši vēl dati, kas nav nosūtīti uz <i>Elasticsearch</i> indeksu: <ul style="list-style-type: none"> <li>• ja ir, tad nosūta tos uz <i>Elasticsearch</i> indeksu;</li> <li>• ja visi dati ir nosūtīti, tad nedara neko.</li> </ul> </li> </ol>
Izvaddati
Uz <i>Elasticsearch</i> tiek nosūtīti juridisku personu datu atjauninājumi, piemēram, uzņēmuma adreses maiņa vai uzņēmuma likvidācija.

Procedūras *UpdateAll* programmkodu skatīt 9.pielikumā.

### 5.3.2.3.Funkcija *urSearch*

Funkcija izstrādāta *JavaScript* valodā, izmantojot *AngularJS* ietvaru. Funkcijas aprakstu skatīt 5.22.tabulā.

5.22.Tabula

#### Funkcijas *urSearch* apraksts

Mērķis
Atgriezt dialoga logu “Uzņēmuma meklēšana”.
Apraksts
<ol style="list-style-type: none"><li>1. Tiek definēti dialoga loga iestatījumi. Definē tādas lietas kā dialoga loga nosaukumu, dialoga loga veidni, dialoga loga izmēru.</li><li>2. Funkcijas dialoga loga iestatījumos tiek definēta arī funkcija, kas tiks izpildīta pēc pogas “Pievienot” nospiešanas.</li></ol>
Izvaddati
Dialoga logs “Uzņēmuma meklēšana”.

Funkcijas *urSearch* programmkodu skatīt 3.pielikumā.

### 5.3.2.4.Funkcija *onSelect*

Funkcijas aprakstu skatīt 5.23.tabulā.

5.23.Tabula

#### Funkcijas *onSelect* apraksts

Mērķis
Aizpildīt jaunas kontaktpersonas pievienošanas formas lauku “Nosaukums”, “Reģistrācijas numurs” un “Adrese” ar atlasītajiem juridiskās personas datiem.
Apraksts
<ol style="list-style-type: none"><li>1. Funkcijai kā ieejas parametri tiek padoti atlasītie juridisko personu dati (juridiskās personas nosaukums, reģistrācijas numurs un adrese).</li><li>2. Jaunas kontaktpersonas pievienošanas formas lauki “Nosaukums”, “Reģistrācijas numurs” un “Adrese” tiek aizpildīti ar atlasītās juridiskās personas datiem.</li></ol>
Izvaddati
Atlasītie juridiskās personas dati. Jaunas kontaktpersonas pievienošanas formā lauki “Nosaukums”, “Reģistrācijas numurs” un “Adrese” tiek aizpildīti ar atlasītajiem datiem.

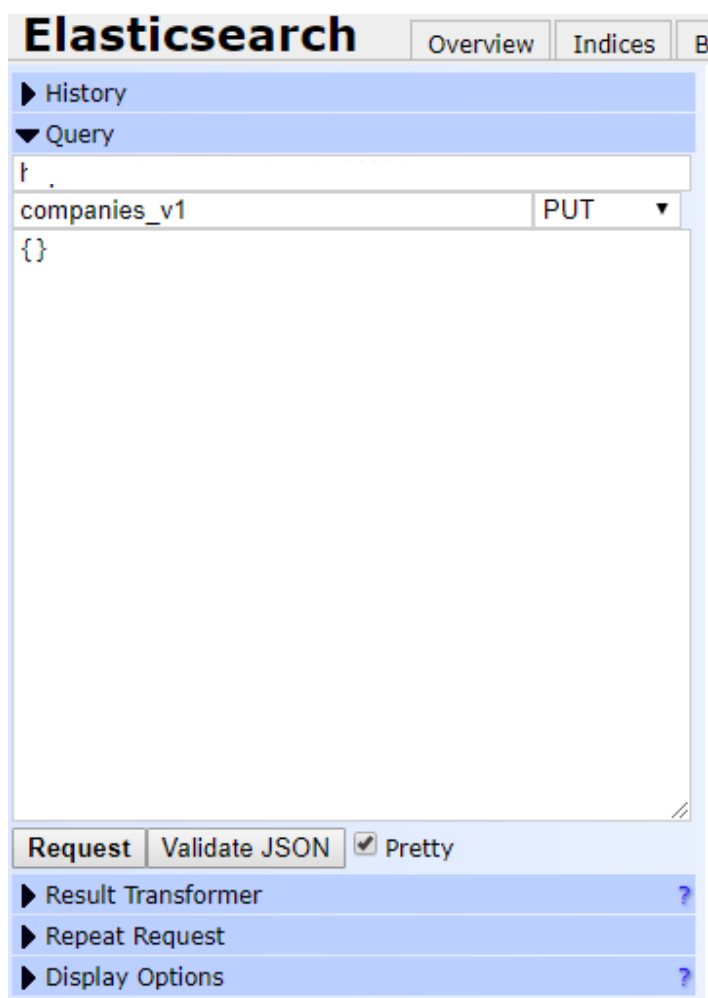
Funkcijas *onSelect* programmkodu skatīt 4.pielikumā.

### 5.3.3. Darba uzdevuma DU3284 *Elasticsearch* projektējums

Lai implementētu darba uzdevuma DU3284 galveno funkcionalitāti – juridisku personu meklēšanu -, izveidots *Elasticsearch* indekss “companies\_v1”, kurā tiek uzturēti juridisku personu dati. *Elasticsearch* indeksam piešķirta saīsnē (no angļu valodas – “alias”) “companies”, lai programmkoda izstrāde būtu ērtāka.

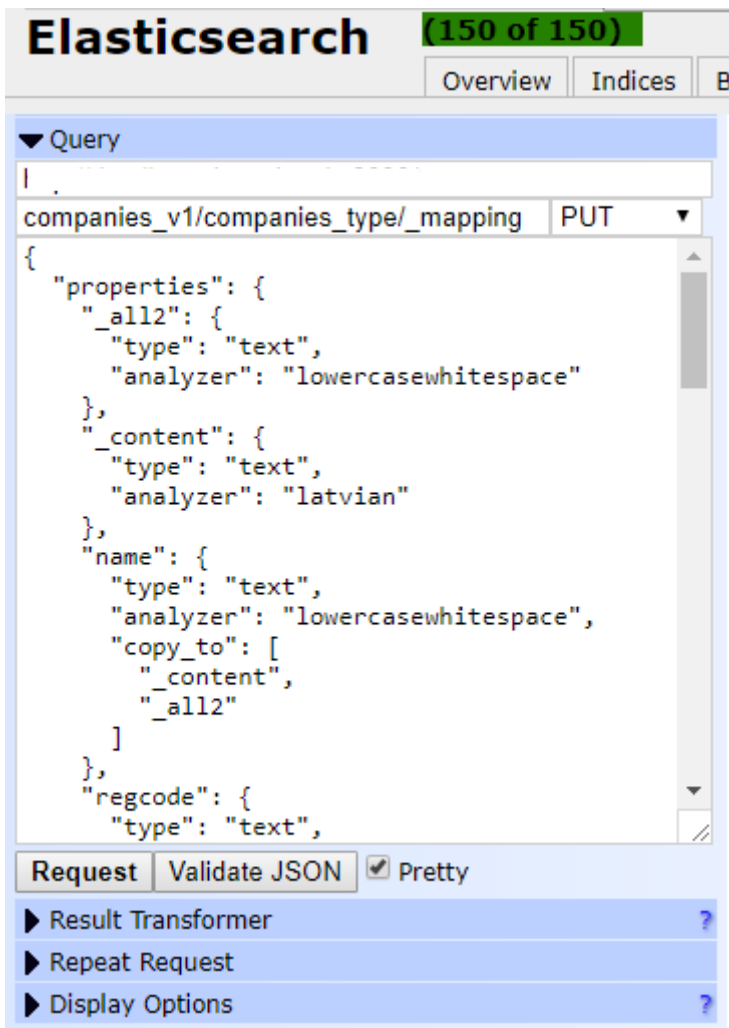
Indeksa izveide notiek sekojoši:

1. Tiek izveidots pats indekss (skatīt 5.9.attēlu). *Elasticsearch* pārvaldības saskarnē, izvēloties šķirklī “Any request”, ievada jaunā indeksa nosaukumu un izvēlās “PUT” metodi. Pieprasījuma laukā raksta figūriekavas. Spiež pogu “Request”.



5.9.att. *Elasticsearch* indeksa “companies\_v1” izveide

2. Tiek definēta izveidotā indeksa uzbūve JSON formātā, tas ir, kādus datu laukus saturēs indekss. Darba uzdevuma DU3284 specifikas dēļ tika izveidoti tie lauki, kurus būs nepieciešams izmantot, lai aizpildītu jaunas kontaktpersonas pievienošanas formas laukus (skatīt 5.10. attēlu). Indeksa uzbūves fragmentu skatīt 5.pielikumā.



5.10.att. Elasticsearch indeksa “companies\_v1” uzbūves (datu lauku) definēšana

Indekss “companies\_v1” tiek aizpildīts ar vajadzīgo informāciju, izmantojot procedūru *CompaniesSyncWithIndex*. Procedūras aprakstu skatīt 5.20.tabulā.

### 5.3.4. LIETVARIS projekta gatavie risinājumi

Lai nodrošinātu koda kvalitāti un ievērotu projekta LIETVARIS programmēšanas principus, darba uzdevuma DU3284 izstrādē izmantoti jau gatavi risinājumi.

5.24.Tabula

#### LIETVARIS projekta gatavie risinājumi

Pakotne	Funkcija, procedūra	Apraksts
U_JSON_PCK	<i>SetStringProperty</i>	Ieraksta JSON objektā tekstu
	<i>SetIntProperty</i>	Ieraksta JSON objektā veselu skaitļu vērtības

	<i>writeToClob</i>	Veic JSON objekta konvertāciju uz NCLOB datu tipu
I_UR_INTEGRATION_PCK	<i>InternalFormFullAddress</i>	Formatē strukturēto adresi
U_LOB_PCK	<i>WriteAppendText</i>	Pievieno tekstu NCLOB mainīgajam
	<i>WriteAppendNClob</i>	Pievieno NCLOB mainīgā saturu citam NCLOB mainīgajam
ELASTIC_PCK	<i>Send</i>	Datu sūtīšana uz <i>ElasticSearch</i> un pieprasījuma rezultāta saņemšana
U_ERROR_PCK	<i>AddErrorToList</i>	Kļūdu ziņojumu sarakstam pievieno kļūdu

### 5.3.5. Darba uzdevuma DU3284 labojumu saskarnes

Nodaļā ir aprakstīti visi veiktie LIETVARIS sistēmas saskarnes labojumi, darba uzdevuma DU3284 implementācijai.

#### 5.3.5.1. Juridisku personu brīvā meklēšana

Apraksts: Papildināt jaunas kontaktpersonas pievienošanas formu ar iespēju veikt juridisku personu datu brīvo meklēšanu pēc atslēgvārda un aizpildīt formu ar izvēlētās juridiskās personas datiem.

Identifikators: DU3284

5.11.attēlā redzama jaunas kontaktpersonas pievienošanas forma.

5.25.Tabula

#### DU3284 saskarnes labojumu apraksts, 1.daļa

Nr.	Labotā vai izveidotā datne	Veiktās izmaiņas
1.	<i>Edit.cshtml.</i>	Pievienota papildus programmkoda rindiņa, kas definē pogu "Uzņēmuma reģistra dati" un funkcijas izsaukums, kas tiks izpildīta, nospiežot to. Poga

		noformēta vadoties pēc LIETVARIS projekta vadlīnijām.
2.	<i>Contact.person.edit.controller.js</i>	Izveidota funkcija <i>urSearch</i> , kas atgriež dialoga logu, pēc pogas “Uzņēmuma reģistra dati” nospiešanas. Funkcijas aprakstu skatīt 5.17.tabulā.

- Ar 1 atzīmēta poga, pēc kuras nospiešanas tiek atgriezts dialoga logs “Uzņēmuma meklēšana”.

5.11.att. Jaunas kontaktpersonas pievienošanas forma

5.12. attēlā redzams brīvās meklēšanas dialoga logs.

5.26.Tabula

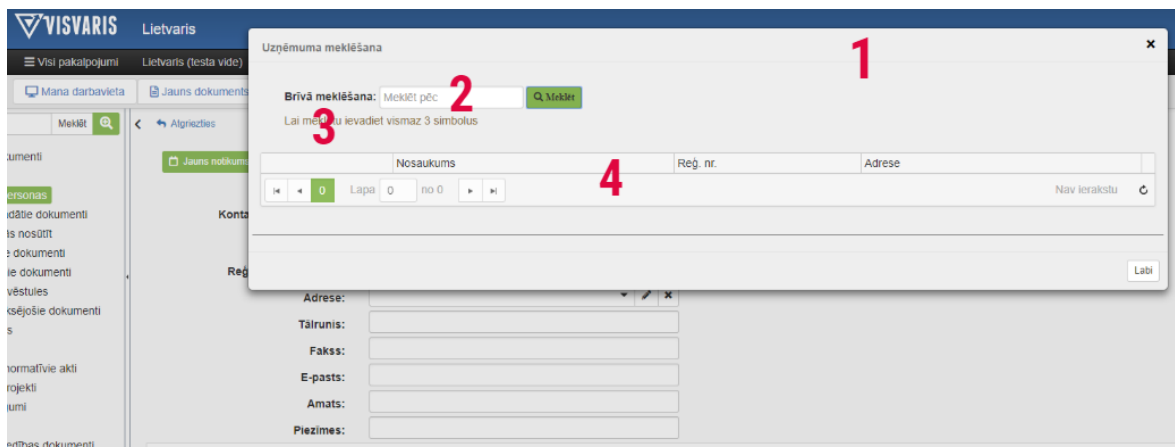
**DU3284 saskarnes labojumu apraksts, 2.daļa**

Nr.	Labotā vai izveidotā datne	Veiktās izmaiņas
1.	<i>companySearch.cshhtml</i>	Izveidota dialoga loga veidne. Tajā tiek definēts: loga ievades lauks “Brīvā meklēšana”, poga “Meklēt”, informatīvais teksts, loga tabulas izsaukums.

2.	<i>Search.select.company.search.controller.js</i>	Izveidots kontrolieris (izmantojot <i>AngularJS</i> ), kurā ir nedefinētas tabulas kolonnas, tabulas iestatījumi, nedefinēta meklēšanas loģika un atlasīto datu nodošana uz citu kontrolieri.
----	---	---

Datnes *search.select.company.search.controller.js* programmkodu skatīt 11.pielikumā.

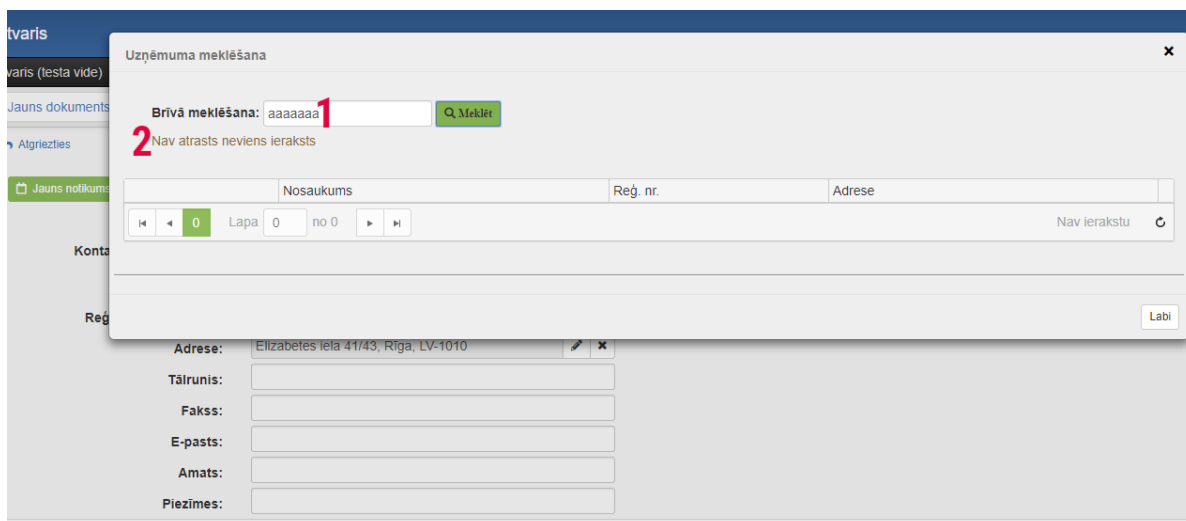
- Ar 1 atzīmēts dialoga logs “Uzņēmuma meklēšana”.
- Ar 2 atzīmēts ievadlauks “Brīvā meklēšana”, kur jāievada atslēgvārdi, pēc kā notiks juridiskās personas meklēšana.
- Ar 3 atzīmēts informatīvais teksts, kad nav ievadīti vismaz 3 simboli.
- Ar 4 atzīmēta tabula, kurā tiks atlasītas juridiskās personas.



#### 5.12.att. Jaunas kontaktpersonas pievienošanas forma un dialoga logs “Uzņēmuma meklēšana”

5.13.attēlā redzams brīvās meklēšanas dialoga logs.

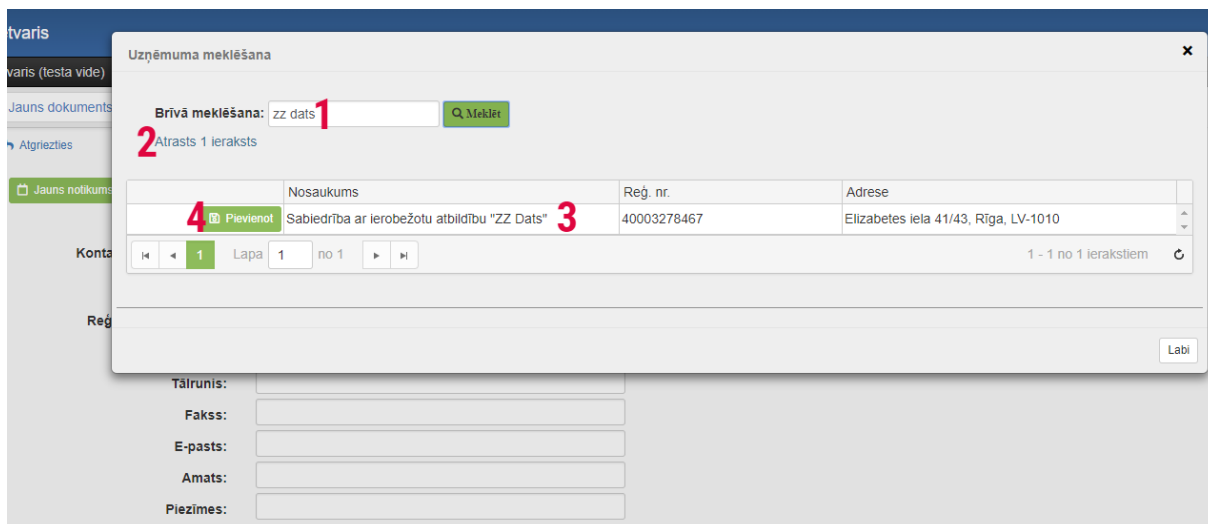
- Ar 1 atzīmēts atslēgvārds, pēc kura notiek meklēšana.
- Ar 2 atzīmēts informatīvais teksts, par nesekmīgiem meklēšanas rezultātiem



5.13.att. Jaunas kontaktpersonas pievienošanas forma un dialoga logs “Uzņēmuma meklēšana”

5.14.attēlā redzams brīvās meklēšanas dialoga logs.

- Ar 1 atzīmēts atslēgvārds, kurš tika ievadīts ievadlaukā.
- Ar 2 atzīmēts informatīvais teksts, par sekmīgiem meklēšanas rezultātiem.
- Ar 3 atzīmētā atlasītā juridiskā persona.
- Ar 4 atzīmēta poga “Pievienot”, pēc kuras nospiešanas, forma tiks aizpildīta ar atlasītās juridiskās personas datiem.



5.14.att. Jaunas kontaktpersonas pievienošanas forma un dialoga logs “Uzņēmuma meklēšana”

5.15.attēlā redzama jauns kontaktpersonas pievienošanas forma.

## DU3284 saskarnes labojumu apraksts, 3.daļa

Nr.	Labotā vai izveidotā datne	Veiktās izmaiņas
1.	<i>Contact.person.edit.controller.js</i>	Izveidota funkcija <i>onSelect</i> , kas pēc pogas “Pievienot” nospiešanas, aizpildīs jaunas kontaktpersonas pievienošanas formu ar juridiskās personas datiem. Funkcijas aprakstu skatīt 5.18.tabulā.

- Ar 1 atzīmēta forma, kura tika aizpildīta, pēc veiksmīgas juridiskas personas brīvās meklēšanas un pogas “Pievienot” nospiešanas.

Atgriezties

Jauns notikums Uzņēmuma reģistra dati

Kontaktpersonas veids: Juridiska persona

Nosaukums:\* Sabiedrība ar ierobežotu atbildību "ZZ Dats" 1

Reģistrācijas numurs: 40003278467

Adrese: Elizabetes iela 41/43, Rīga, LV-1010

Tālrunis:

Fakss:

E-pasts:

Amats:

Piezīmes:

5.15.att. Aizpildīta jaunas kontaktpersonas pievienošanas forma

## 5.4. Darba uzdevuma “Labojumi kontaktpersonu modulī” projektējums

Darba uzdevuma mērķis ir labot lietvedības sistēmas LIETVARIS nepilnības kontaktpersonu modulī.

## 5.4.1. Labotā moduļa vispārīgs apraksts

5.28.Tabula

### Kontaktpersonu modulī mainītās datnes

Nolūks
Labot kontaktpersonas kartītes un jaunas kontaktpersonas pievienošanas formas kļūdas – gan vizuālās, gan funkcionālās.
Objekta nosaukums
<ol style="list-style-type: none"><li>1. contact.person.edit.controller.js</li><li>2. edit.cshtml</li><li>3. router.data.service.js</li><li>4. header.component.js</li><li>5. save.changes.popup.component.js</li><li>6. edit.head.template.html</li><li>7. save.changes.popup.component.close.template.html</li><li>8. I_SERVER_SETUP_PCK @LIETVARIS'</li><li>9. APP_CLIENT_PCK@LIETVARIS</li><li>10. I_CLIENT_PCK@LIETVARIS</li></ol>
Funkcionalitāte
<p>Lietotājs var brīvi samazināt loga izmēru, nepazaudējot formas lauku nosaukumus (1., 2.)</p> <p>Pogas “Atgriezties” funkcionalitātes labojums (1.)</p> <p>Pogas “Atgriezties” funkcionālais labojums, kad kartīte atvērta jaunā cilnē (3., 4., 6.)</p> <p>Nomainīt pogas “Atgriezties” nosaukumu uz “Aizvērt logu”, kad kartīte atvērta jaunā cilnē (5.,7.)</p> <p>Pogas “Saglabāt” izmēra pielāgošana citām skata pogām (2.)</p> <p>Kontaktpersonas šķirklis “Sarakste” un “Notikumi” noņemta informācija, kas dublējas(2.)</p> <p>Kļūdu paziņojuma labojums pie identiskiem personas kodiem (8.)</p> <p>Pievienojot jaunu ārvalstnieku kā kontaktpersonu, laukā “Reģ.nr./pers.kods” atļaut ievadīt līdz 14 simboliem (8., 9., 10.)</p> <p>Pievienojot jaunu ārvalstnieku kā kontaktpersonu, formas laukam ir jāmaina nosaukums uz “Reģ.nr./pers.kods”, kā arī laukā nevar ievadīt vairāk par 14 simboliem (2.)</p>

## 5.4.2. Datu detalizētais projektējums

Šajā nodaļā ir aprakstītas tabulas, kuras darba autoram nācās izmantot un labot, izstrādājot darba uzdevumu “Labojumi kontaktpersonu modulī”, kas ietver 7 mazākus darba uzdevumus.

### 5.4.2.1. Tabula DOCUMENT\_PERSONS

Tabula DOCUMENT\_PERSONS paredzēta, lai glabātu tajā dokumentam piesaistītā korespondenta/adresāta datus konkrētajā dokumentā. Darba uzdevuma izstrādes laikā tabulas lauks CODE tika palielināts, lai varētu implementēt darba uzdevumu DU4863.

5.29. Tabula

#### Tabula DOCUMENT\_PERSONS

Kolonnas nosaukums	Datu tips	Apraksts
CODE	VARCHAR2(14)	Personas kods vai vienotais NM reģistrācijas numurs

\*Norādīti tikai izmantotie tabulas DOCUMENT\_PERSONS lauki.

### 5.4.2.2. Tabula CLIENTS

Tabula paredzēta kontaktpersonu datu glabāšanai. Darba uzdevuma izstrādes laikā tabulas lauks CODE tika palielināts, lai varētu implementēt darba uzdevumu DU4863.

5.30. Tabula

#### Tabula CLIENTS

Kolonnas nosaukums	Datu tips	Apraksts
CODE	VARCHAR2(14)	Personas kods vai vienotais NM reģistrācijas numurs

\*Norādīti tikai izmantotie tabulas CLIENTS lauki.

### 5.4.3. Detalizēts moduļa projektējums

Nodaļas ietvaros ir aprakstīti funkciju un procedūru veiktie labojumi.

5.31.Tabula

#### Kontaktpersonu modulī veiktie labojumi

Nr.	Nosaukums	Veiktās izmaiņas	Funkcionalitāte	Lietotājstāsts
1.	Contact.person.edit.controller.js	<ul style="list-style-type: none"> <li>Pievienota objekta <i>contactPersonSavedCopy</i> datu papildus inicializācija.</li> <li>Laboti formas izmēra parametri.</li> </ul>	<ul style="list-style-type: none"> <li>Atkārtoti ielādē datus, lai poga "Aizvērt" strādātu.</li> <li>Jaunas kontaktpersonas pievienošanas formas izmērus var mainīt un nepazaudēt lauka nosaukumus.</li> </ul>	3.2.1., 3.2.3.
2.	Edit.cshtml	<ul style="list-style-type: none"> <li>Laboti formas izmēra parametri</li> <li>Pievienota papildus stils pogai "Saglabāt" - lg-btn.</li> <li>Noņemta lieko šķirkļu informācija.</li> <li>Ārvalstnieku reģ.nr/pers.kodam pievienota papildus pārbaude.</li> </ul>	<ul style="list-style-type: none"> <li>Var mainīt loga izmērus, formas lauku nosaukumi nepārklāsies un nepazudīs.</li> <li>Formas visas pogas ir vienāda izmēra.</li> <li>Atverot šķirkļus, nav redzama lieka informācija.</li> </ul>	3.2.3., 3.2.4., 3.2.5., 3.2.7.

			<ul style="list-style-type: none"> <li>Ievadlaukā nevar ierakstīt vairāk par 14 simboliem.</li> </ul>	
3.	routerDataService	<ul style="list-style-type: none"> <li>Izveidota funkcija <i>_hasHistory</i></li> <li>Pieliktas papildus pārbaudes <i>_navigateBack</i> funkcijai</li> </ul>	<ul style="list-style-type: none"> <li>Funkcija nosaka, vai atvērtais logs ir ar pārlūkošanas vēsturi vai bez tās.</li> <li>Pārbaudes nosaka, kāda veida “atpakaļ” darbība būs jāveic – cilnes aizvēršana vai atgriešanās iepriekšējā skatā.</li> </ul>	3.2.2.
4.	<i>\$onInit</i>	<ul style="list-style-type: none"> <li>Pievienota pogas izsaukšana, ja pārbaude izpildās.</li> </ul>	<ul style="list-style-type: none"> <li>Poga tiek izsaukta tikai tad, kad pēc funkcijas <i>_hasHistory</i> tiek noteikts, vai logam ir pārlūkošanas vēsture vai tas ir tikko atvērts jaunā cilnē.</li> </ul>	3.2.2.
5.	<i>\$onInit</i>	<ul style="list-style-type: none"> <li>Pievienota funkcija, kas maina pogas nosaukumu, atkarībā no pārlūkošanas vēstures.</li> </ul>	<ul style="list-style-type: none"> <li>Pogas nosaukums tiek mainīts uz “Aizvērt logu”, ja pēc funkcijas <i>_hasHistory</i> tiek noteikts, ka tas ir tikko atvērts skats jaunā cilnē.</li> </ul>	3.2.2
6.	<i>Edit.head.template.html</i>	<ul style="list-style-type: none"> <li>Pievienota papildus pogas pārbaude.</li> </ul>	<ul style="list-style-type: none"> <li>Poga “Atgriezties” tiek noņemta, ja logs ir atvērts jaunā cilnē.</li> </ul>	3.2.2.

7.	<i>save.changes.popup.component.close.template.html</i>	<ul style="list-style-type: none"> <li>• Pievienots mainīgās vērtības izsaukums.</li> </ul>	<ul style="list-style-type: none"> <li>• Atkarībā no tā, vai logs ir ar pārlūkošanas vēsturi vai tas ir atvērts jaunā cilnē, pogas nosaukums tiek mainīts.</li> </ul>	3.2.2.
8.	<i>populateErrors</i>	<ul style="list-style-type: none"> <li>• Labots kļūdu paziņojums.</li> <li>• Izveidots kļūdu paziņojums.</li> </ul>	<ul style="list-style-type: none"> <li>• Lietotājam tiek izvadīti korekti kļūdas paziņojumi.</li> </ul>	3.2.6., 3.2.7.
9.	<i>Save</i>	<ul style="list-style-type: none"> <li>• Mainīts mainīgā datu tipa lielums.</li> </ul>	<ul style="list-style-type: none"> <li>• Ārvalstnieki var ievadīt līdz 14 simboliem garu reģ.nr./pers.kodu.</li> </ul>	3.2.7.
10.	<i>CheckPersOrRegCode</i>	<ul style="list-style-type: none"> <li>• Pievienota papildus pārbaude ārvalstnieku reģ.nr./pers.kodiem.</li> </ul>	<ul style="list-style-type: none"> <li>• Ārvalstnieks var ievadīt ne tikai ciparus, bet arī burtus kā reģ.nr./pers.kodu.</li> </ul>	3.2.7.

## 5.4.4. Darba uzdevuma “Labojumi kontaktpersonu modulī” labojumu saskarnes

Nodaļā ir aprakstīti visi veiktie LIETVARIS sistēmas saskarnes labojumi, darba uzdevuma “Labojumi kontaktpersonu modulī” implementācijai.

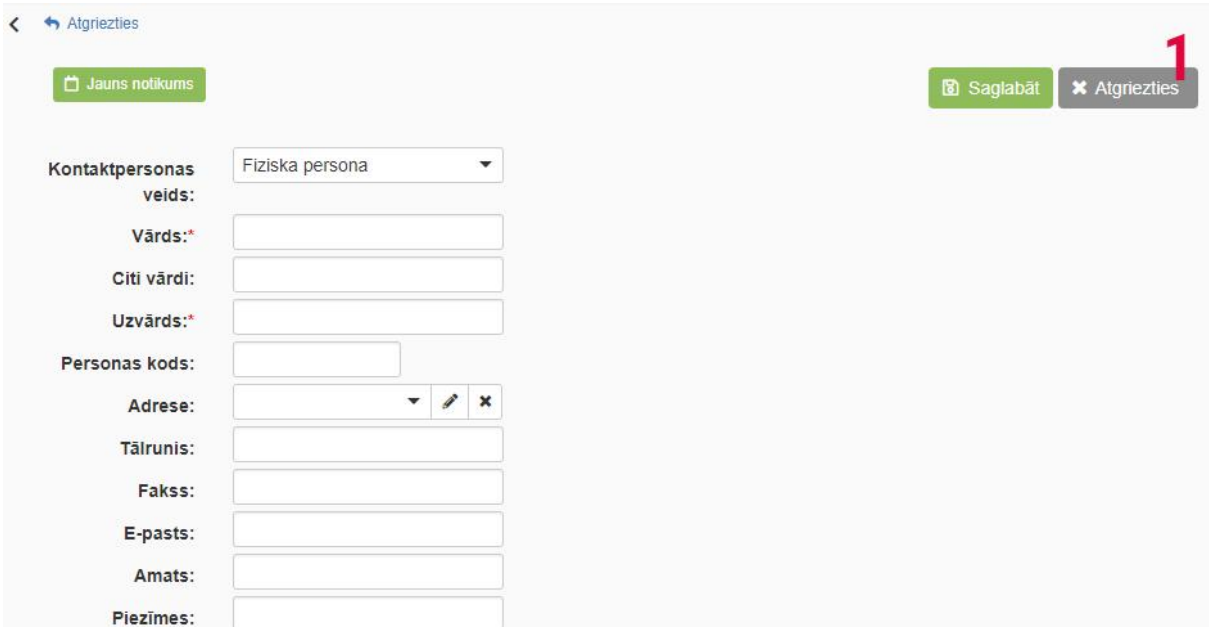
### 5.4.4.1.Pogas “Atgriezties” funkcionalitātes labojums

Apraksts: Labot jaunas kontaktpersonas formas pogas “Atgriezties” funkcionalitāti.

Identifikators: DU4783

5.16.attēlā redzama jaunas kontaktpersonas pievienošanas forma.

- Ar 1 atzīmēta poga, kuras funkcionalitāte tika labota.



The screenshot shows a web interface for adding a contact. At the top left is a back arrow and the text 'Atgriezties'. Below it is a green button with a calendar icon and the text 'Jauns notikums'. At the top right are two buttons: a green 'Saglabāt' button and a grey 'Atgriezties' button with a red '1' above it. The form fields are: 'Kontaktpersonas veids:' with a dropdown menu showing 'Fiziska persona'; 'Vārds:\*' with a text input; 'Citi vārdi:' with a text input; 'Uzvārds:\*' with a text input; 'Personas kods:' with a text input; 'Adrese:' with a text input, a dropdown arrow, an edit icon, and a delete icon; 'Tālrunis:' with a text input; 'Fakss:' with a text input; 'E-pasts:' with a text input; 'Amats:' with a text input; and 'Piezīmes:' with a text input.

5.16.att.Jaunas kontaktpersonas pievienošanas forma

### 5.4.4.2.Jaunas kontaktpersonas pievienošanas formas loga izmērs

Apraksts: Labot jaunas kontaktpersonas pievienošanas formas izmērus, ja loga izmērs ir 1024x768 un mazāks tā, lai visus lauka nosaukumus būtu iespējams redzēt.

Identifikators: DU1115

5.17.attēlā redzama jaunas kontaktpersonas pievienošanas forma, kuras loga izmērs ir mazāks par 1024x768.

- Ar 1 atzīmēts atvērtā loga izmērs.
- Ar 2 atzīmēts lauka nosaukums “Kontaktpersonas veids”.

VISVARIS Lietvaris Lietotājs: Babīte Babīte | Beigt darbu 990px x 644px  
 Mana darbavieta Jauns dokuments Jauns notikums Pārskati Kontrole  
 Meklēt  
 Mani dokumenti  
 Notikumi  
**Kontaktpersonas**  
 Neapstrādātie dokumenti  
 Neizdevās nosūtīt  
 Saņemtie dokumenti  
 Nosūtāmie dokumenti  
 Iekšējās vēstules  
 Faktus fiksējošie dokumenti  
 Personāls  
 Rīkojumi  
 Iekšējie normatīvie akti  
 Lēmumprojekti  
 Darba līgumi  
 Līgumi  
 Grāmatvedības dokumenti  
 Uzdevumi  
 Sagataves  
 Valdes sēdes

Atgriezies  
 Jauns notikums  
 Saglabāt  
 Atgriezies

**2** Kontaktpersonas veids: Fiziska persona  
 Vārds:\*  
 Citi vārdi:  
 Uzvārds:\*  
 Personas kods:  
 Adrese:  
 Tālrunis:  
 Fakss:  
 E-pasts:  
 Amats:

5.17.att. Jaunas kontaktpersonas pievienošanas forma

#### 5.4.4.3. Pogas “Saglabāt” vizuālais uzlabojums

Apraksts: Labot kontaktpersonas kartītes pogas “Saglabāt” izmērus.

Identifikators: DU2964

5.18.attēlā redzama kontaktpersonas kartīte.

- Ar 1 atzīmēta poga, kuras izmērs tika mainīts.

Atgriezies  
 Jauns notikums  
 Saglabāt  
 Atgriezies

Kontaktpersonas veids: Fiziska persona  
 Vārds:\* Vārds  
 Citi vārdi:  
 Uzvārds:\* Uzvārds  
 Personas kods:  
 Adrese: c iela, AFGANISTĀNA  
 Tālrunis:  
 Fakss:  
 E-pasts:  
 Amats:  
 Piezīmes:

Pārstāvis  
 Nosaukums PIlnvarojums līdz E-pasts  
 + Pievienot pārstāvi

5.18.att. Kontaktpersonas kartīte

#### 5.4.4.4. Pogas "Atgriezties" funkcionalitātes labojums

Apraksts: Labot kontaktpersonas kartītes pogas "Atgriezties" funkcionalitāti, atverot to no e-pasta jaunā cilnē. Izmainīt esošo pogas izskatu.

Identifikators: DU2592

5.19.attēlā redzama kontaktpersonas kartīte.

- Ar 1 ir atzīmēta poga, kuras izskats un funkcionalitāte jaunā cilnē tika labota tā, lai, nospiežot to, cilne aizvērtos.
- Ar 2 atzīmēta vieta, kur tika atspējota aktīvā saite.

Jauns notikums

Saglabāt Aizvērt logu

Kontaktpersonas veids: Fiziska persona

Vārds:\* Vārds

Citi vārdi:

Uzvārds:\* Uzvārds

Personas kods: 130994-10492

Adrese: c iela, AFGANISTĀNA

Tālrunis: 29466284

Fakss:

E-pasts: test@test.lv

Amats: Svarīgs amats

Piezīmes: Svarīga piezīme

Pārstāvis

Darbības	Dokumenta veids	Reģistrācijas datums	Reģistrācijas numurs	Saturs	Korespondenta nu...	Korespondenta da...
Atvērt	Nosūtāmais dokuments	04.04.2018.	BN-2-5/18/134-2.5	test		

50 Ieraksti lapā 1 - 1 no 1 ierakstiem

5.19.att. Kontaktpersonas kartīte

#### 5.4.4.5.Šķirkļu informatīvā teksta labojums

Apraksts: Labot kontaktpersonas kartītes šķirkļu sniegto informāciju.

Identifikators: DU1123

5.20.attēlā redzama kontaktpersonas kartīte.

- Ar 1 atzīmēts labotā šķirkļa informācija.

Atgriezties

Jauns notikums Saglabāt Atgriezties

**Kontaktpersonas veids:** Fiziska persona

**Vārds:** Vārds

**Citi vārdi:**

**Uzvārds:** Uzvārds

**Personas kods:**

**Adrese:** c iela, AFGANISTĀNA

**Tālrunis:**

**Fakss:**

**E-pasts:**

**Amats:**

**Piezīmes:**

Sarakste **Notikumi** Kontaktinformācija

1

Darbības	Dokumenta veids	Reģistrācijas datums	Reģistrācijas numurs	Saturs	Korespondenta nu...	Korespondenta da...
Atvērt	Nosūtāmais dokuments	04.04.2018.	EMC5000105	test		

1 - 1 no 1 ierakstiem

5.20.att. Kontaktpersonas kartīte

#### 5.4.4.6. Kļūdu paziņojuma labojums

Apraksts: Labot kļūdu paziņojumu pie vienādiem personas kodiem.

Identifikators: DU4406

5.21.attēlā redzama kontaktpersonas kartīte.

- Ar 1 atzīmēts labotais kļūdu paziņojums.

Lietvaris

Lietvaris (testa vide) Lietvaris (izstrādes vide)

Jauns dokuments Jauns notikums Pārskati

Atgriezties

Jauns notikums

**Kontaktpersonas veids:** Fiziska persona

**Vārds:** TestVārds

**Citi vārdi:**

**Uzvārds:** TestUzvārds

**Personas kods:** 12345678910

**Adrese:** Testētāju iela, Aizkraukle

**Tālrunis:**

**Fakss:**

**E-pasts:** test@test.lv

**Amats:** Amats

**Piezīmes:**

**Pārstāvis**

Nosaukums

+ Pievienot pārstāvi

Kļūda

⚠ Norādītais kods jau eksistē! 1

Labi

5.21.att. Kontaktpersonas kartīte

#### 5.4.4.7. Ārvalstnieka reģ.nr./pers.koda pārbaude

Apraksts: Veikt izmaiņas jaunas kontaktpersonas pievienošanas formā, pievienojot jaunu ārvalstnieku.

Identifikators: DU4863

5.22.attēlā redzama jaunas kontaktpersonas pievienošanas forma.

- Ar 1 atzīmēts ievades lauka “Reģ.nr./pers.kods” nosaukums.
- Ar 2 atzīmēts ievades lauks “Reģ.nr./pers.kods”, kurā var ievadīt līdz 14 zīmēm.

Atgriezies

Jauns notikums

Kontaktpersonas veids: Ārvalstnieks

Nosaukums:\*

1 Reģ.nr./pers.kods: 12345678911234 2

Adrese:

Tālrunis:

Fakss:

E-pasts:

Amats:

Piezīmes:

5.22.att. Jaunas kontaktpersonas pievienošanas forma

## 6. PROJEKTA ORGANIZĀCIJA

Kontaktpersonu moduļa papildinājumi un uzlabojumi izstrādāti LIETVARIS projekta ietvaros. Darba izstrādē izmantoti LIETVARIS projektā izmantotie darba organizācijas rīki un metodes, kā arī pieņemtie standarti.

2018.gada maijā LIETVARIS komandā strādā 15 cilvēku komanda, kas sastāv no programmētājiem, sistēmas analītiķiem, testētājiem, klientu konsultantes un projekta vadītājas. Sakarā ar LIETVARIS sistēmas strauju attīstību un pieprasījumu Latvijas pašvaldībās, LIETVARIS komanda strauji paplašinās, lai varētu nodrošināt kvalitatīvu sistēmas piegādi klientiem.

LIETVARIS projekts seko spējās programmatūras izstrādes principiem, kas paredz izstrādes procesa iterāciju ik pēc pāris nedēļām, LIETVARIS projekta gadījumā aptuveni 2 nedēļām. Šo vienu izstrādes laika posmu sauc par sprintu. Sprinta laikā komanda strādā pie klienta jauno prasību īstenošanas un sistēmas uzlabošanas. Pirms katra sprinta sākuma tiek sasaukta sprinta sapulce, kurā tiek apspriesti komandas dalībnieku paveiktais iepriekšējā sprintā un tiek definēti nākošie uzdevumi nākošajam sprintam. Sprinta ietvaros katram izstrādātājam ir uzdoti uzdevumi, kuru daudzums ir atkarīgs no to sarežģītības. Ja izstrādātājs izpilda visus sprinta uzdevumus pirms tā beigām, tad viņš var turpināt pildīt nākošā sprinta uzdevumus, ja tādi ir jau definēti.

Projekta uzdevumu pārvaldība notiek, izmantojot atvērtā pirmkoda projektu un resursu plānošanas un vadības rīku *Redmine*. *Redmine* rīkā katram projekta dalībniekam ir redzami savi uzdevumi, kā arī nepieciešamības gadījumā dalībnieks var apskatīt citu projektā iesaistīto darbinieku uzdevumus. Katram uzdevumam tiek piešķirts *Redmine* uzdevuma identifikators, pēc kura var identificēt to. Atverot uzdevuma definējuma skatu, darbinieks var redzēt uzdevuma autoru, darbinieku, kam šis konkrētais uzdevums ir uzdots, sistēmas versijas numuru, uzdevuma kategoriju, uzdevuma izpildes laiku, uzdevuma aprakstu, līdz kuram uzdevums jāizpilda un citus uzdevuma nosacījumus. Uzdevuma definējuma skatu *Redmine* rīkā skatīt 6.1.attēlā.

The screenshot shows the Redmine interface for task #3284. The task title is "2. Pieslēgt ElasticSearch un brīvo meklēšanu". The status is "Testēšana". The assignee is Laura Tirumniece. The due date is 25.04.2018. The start date is 26.02.2018. The spent time is 0.00h. The created date is 26.02.2018 10:35. The category is Integrācija. The milestone is v1.35. The story points are 40. The task description includes instructions for testing and technical requirements for ElasticSearch integration.

### 6.1.att. Darba uzdevums DU3282 projektu un resursu plānošanas un pārvaldības rīkā Redmine

Redmine rīka “Agile” sadaļā ir redzami visi projekta darbiniekam piešķirtie uzdevumi, kas ir sadalīti pa uzdevuma dzīves cikla posmiem – uzdevums, kas ir izstrādes procesā, uzdevums, kas ir testēšanas procesā vai uzdevums ir jau atrisināts. Uzdevumiem šajā skatā var uzlikt filtrus, piemēram, tiks rādīti izvēlētās sistēmas versijas uzdevumi.

The screenshot shows the Redmine Agile board. The board is divided into columns representing different stages of the task lifecycle: "Izstrāde" (Development) with sub-columns "Jauns" (New) and "Procesā" (In Progress); "Testēšana" (Testing) with sub-columns "Jauns" (New) and "Procesā" (In Progress); "Atrisināts" (Resolved); and "Atzūkmes" (Backlog). The board contains several task cards, each with a title, description, assignee, and status. The cards are color-coded by stage: yellow for "Jauns" and light blue for "Procesā".

### 6.2.att. “Agile” sadaļa projektu un resursu plānošanas un pārvaldības rīkā Redmine

Ja projekta izstrādātājs atrod kļūdu, kas nav pierēģistrēta *Redmine* pārvaldības rīkā, izstrādātājs pats var pierēģistrēt kļūdu, protams, konsultējoties ar projekta sistēmas analītiķiem.

6.3.att. Jauna uzdevuma pievienošanas forma projektu un resursu plānošanas un pārvaldības rīkā *Redmine*

Pierēģistrētos uzdevumus var klasificēt sekojoši:

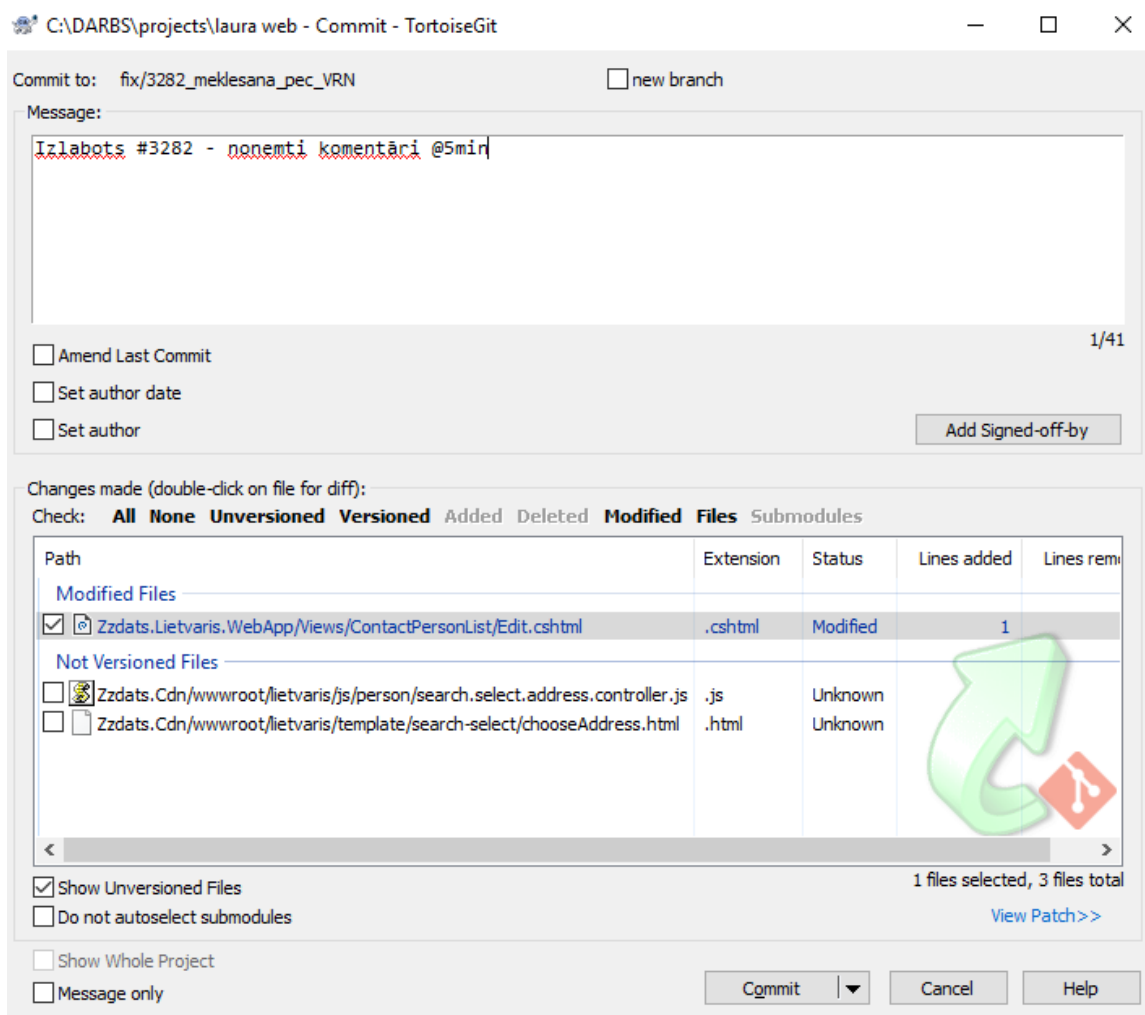
- Kļūdas, kas ir klientu pieteiktas kļūdas vai testēšanas laikā atklātas kļūdas;
- Lietotāju stāsti, kas ir jaunas funkcionalitātes apraksti, kas tiek sadalīti pa mazākiem uzdevumiem;
- Uzdevumi – lietotāju stāstu apakšuzdevumi, kā arī citi darbi, kas nav aprakstīti ar lietotāju stāstu palīdzību.

Uzdevumiem tiek piešķirtas to prioritātes. Ir 5 prioritāšu līmeņi: zems, normāls, augsts, steidzams, tūlītējs. Pirmie jāpilda uzdevumi ar augstākajām prioritātēm.

Uzdevumu sarežģītība un iespējamais patērējamais laiks vērtēts stāstu punktos (*no angļu valodas – “story points”*).

LIETVARIS projektā pārvaldības rīks *Redmine* ir integrēts ar versionēšanas sistēmu *Git* un automatizācijas platformu *Jenkins*, kas nodrošina programmatūras koda kompilēšanu, būvēšanu un testēšanu. Šāda integrācija nodrošina uzdevumu automatizētu dzīves cikla posmu pārlīkšanu uz nākošo posmu. Ja izstrādātājs, iesūtot programmkoda izmaiņas *Git* versionēšanas sistēmā, komentārā ieraksta *Redmine* reģistrētā uzdevuma numuru un atslēgvārdus “izpildīts”,

“izlabots” vai “fixed”, tad *Redmine* šis uzdevums automātiski tiks atzīmēts kā izpildīts un *Redmine* rīka “Agile” sadaļā uzdevums tiek pārcelts no viena dzīves cikla posma uz nākošo, tas ir, no izstrādes un testēšanu. Uzdevuma dzīves cikla posmus skatīt 6.2. attēlā. Izmaiņu iesūtīšanu *Git* versionēšanas sistēmā skatīt 6.4.attēlā.



#### 6.4.att. Izmaiņu saglabāšana versiju kontroles sistēmā *Git*, izmantojot *TortoiseGit* rīku

Saziņa starp LIETVARIS projektā iesaistītajiem darbiniekiem notiek klātienē vai arī izmantojot tērzēšanas rīku *Mattermost*. Katru darba dienas rītu visi komandas LIETVARIS darbinieki sanāk kopā uz rīta sapulci (no *angļu valodas* – “*stand-up meeting*”), lai apspriestu iepriekšējā dienā izdarīto, turpmākos plānus, grūtības, ar kurām saskārās darbinieks un iespējamās problēmu risinājumus.

## 7. KVALITĀTES NODROŠINĀŠANA

Izstrādāto darba uzdevumu kvalitātes nodrošināšanai, darba autors veidoja dokumentāciju, kurā tika aprakstīti lietotārstāsti ar nepieciešamajām prasībām, tika veikts programmatūras projektējuma apraksts, kā arī aprakstīta testēšanas dokumentācija.

Programmatūras prasību definēšana netika veikta vadoties pēc valsts standarta, jo projekts LIETVARIS vadās pēc spējās programmatūras izstrādes principiem, kas nodrošina tuvu sadarbību starp klientu un darba vadītāju.

Ņemot vērā, ka uzņēmumam SIA “ZZ Dats” ir iegūts kvalitātes vadības standarts ISO 9001:2015 un informācijas pārvaldības drošības standarts ISO/IEC 27001:2013, darba izstrādē tika sekots šo standartu vadlīnijām, kā arī papildus tiem tika ievēroti uzņēmuma un LIETVARIS projekta labās programmēšanas principi.

Lai nodrošinātu programmkoda pārskatamību, tika ievērots LIETVARIS projekta programmēšanas vienotais stils, kas ietver mainīgo nosaukumu veidošanu pēc “camelCase” (pirmais vārds ar mazajiem burtiem, katra nākoša vārda pirmais burts ar lielo burtu) stilu. Mainīgo nosaukumiem jābūt labi aprakstošiem (piemēram, tā vietā, lai rakstītu *text1*, tam ir jābūt, piem., *buttonText* . Mainīgā nosaukumam jābūt pašsaprotamam, lai būtu skaidrs, ko tas apzīmē un dara programmkodā), programmkodam ir jābūt komentētam, ir jābūt datu bāzes procedūru un funkciju aprakstiem (skatīt 6.pielikumu), jāievēro atstarpes u.c.

Kvalifikācijas darba dokumentācijas uzturēšanai tika izmantots *OneDrive* pakalpojums, kas piedāvā glabāt un sinhronizēt dokumentus, datnes u.c. mākonī, pieslēdzoties tam ar *Microsoft Office 365* kontu. Kā viena no šī pakalpojuma priekšrocībām jāmin automātiskā saglabāšana.

Lai programmkods nedublētos, tika izmantoti jau gatavi risinājumi, piemēram, JSON formatēšanas funkcijas un procedūras (skatīt 5.8. tabulu un 5.24.tabulu).

Pirms programmkoda izmaiņu iesūtīšanas *Git* versionēšanas sistēmā, visām jaunizveidotajām funkcionalitātēm vai to labojumiem tika veikta testēšana, lai pārliecinātos, ka veiktās izmaiņas ir bijušas veiksmīgas.

Pēc testu veikšanas, visas izmaiņas - jaunas funkcionalitātes ieviešana, uzlabojumi – tika saglabātas versiju kontroles rīkā *Git*, rakstot komentārus par veiktajām izmaiņām un *Redmine* uzdevuma numuru, un atslēgvārdiem (skatīt 6.4.attēlu).

Pirms programmkoda uzlabojumu sapludināšanas LIETVARIS projektā, tiek veidots izmaiņu pieprasījums, kuru izskata cits LIETVARIS izstrādātājs. Izmaiņu pieprasījums tiek veidots, lai nodrošinātu programmkoda kvalitāti un LIETVARIS projekta vadlīniju ievērošanu.

Ja izstrādātājs, kurš recenzē izmaiņu pieprasījumu, konstatē programmkoda nepilnības, viņš var komentēt attiecīgo izmaiņu pieprasījumu un sniegt ieteikumus, lai labotu atrastās programmkoda nepilnības.

Ar uzņēmuma un LIETVARIS projekta programmēšanas standartiem un vadlīnijām darba autors iepazinās uzņēmuma izstrādātajā vietnē “ZZ Dats vikipēdija”.

Dokumentā izmantota trasējamības tabula, kas apraksta saistību starp darba uzdevuma identifikatoriem, lietotājstāstiem, darba uzdevumu projektējumiem un laboto saskarņu aprakstiem.

## 8. KONFIGURĀCIJU PĀRVALDĪBA

### 8.1. Dokumentācijas konfigurācijas pārvaldība

Dokumentācijas versionēšanā tika izmantots *OneDrive* pakalpojums, kas nodrošina dokumentu uzglabāšanu un datu sinhronizāciju. Izmantot šo pakalpojumu var, autorizējoties ar *Microsoft Office 365* kontu.

### 8.2. Programmkode konfigurācijas pārvaldība

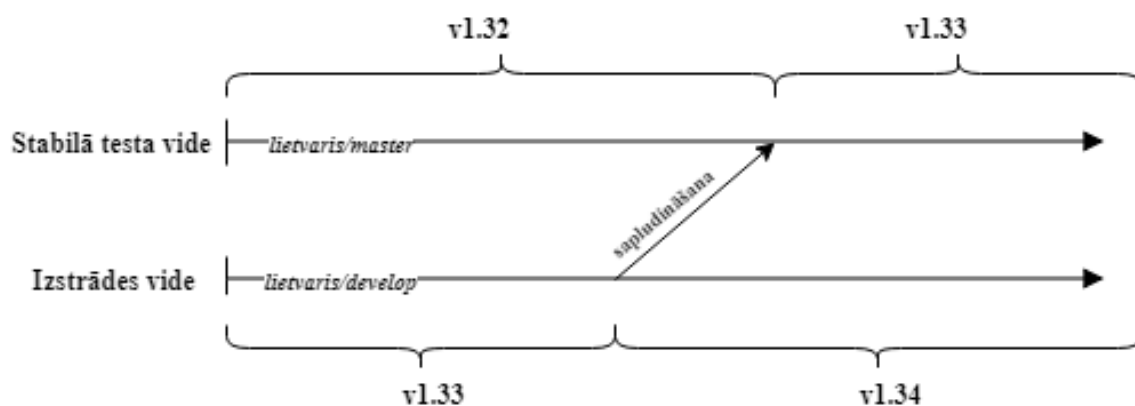
Programmkode versionēšana tika nodrošināta, vadoties pēc SIA “ZZ Dats” un projekta LIETVARIS vadlīnijām. Uzņēmumam ir uzstādīts savs *Git* serveris, kurā tiek uzturēti vairāki SIA “ZZ Dats” projektu repositoņi.

LIETVARIS projektam ir izdalīti divi repositoņi: “*lietvaris/web*” un “*lietvaris/database*”. Lai nodrošinātu paralēlo programmkoda izstrādi, izstrādātājam bija nepieciešams izveidot savus “*lietvaris/web*” un “*lietvaris/database*” repositoņus, klonējot tos uz savas lokālās darbstacijas (*no angļu valodas – “fork”*), lai varētu veikt programmkoda izmaiņas, netraucējot citu izstrādātāju darbu.

Repositoņi “*lietvaris/web*” un “*lietvaris/database*” zaru struktūra ir līdzīga. Lai labāk organizētu izmaiņas, katrā repositoņā ir 2 galvenie zari – “*master*” un “*develop*”- un citi projektam nozīmīgu izmaiņu zari. Zaru “*master*” var saukt par stabilās testa vides zaru, jo tajā ir saglabātas izmaiņas, kas ir paredzētas produkcijai. Zaru “*develop*” var saukt par izstrādes vides zaru, jo tajā tiek sapludināti sprinta laikā izpildīto darba uzdevumu programmkodi. Zaru “*develop*” izmaiņas tiek sapludinātas ar zaru “*master*”, kad izmaiņas “*develop*” zarā pats izstrādātājs ir notestējis un sistēmas izmaiņas var sākt testēt projektā iesaistītie testētāji. Produkcijā izmaiņas tiek ieviestas, kad izmaiņas “*master*” zarā ir notestētas un pilda savu funkcionalitāti. Lai ieviestu “*master*” zaru izmaiņas produkcijā, tiek veidots atsevišķs zars produkcijai, kur zaru nosaukumu veido pēc principiem:

- zaru nosaukuma sākumā jābūt atslēgvārdam “*release*”, kas norāda, ka tas ir produkcijas zars;
- zaru nosaukums jāturpina ar slīpsvītru un sistēmas versijas numuru (“*release/v1.34*”).

Kad produkcijas zaru izmaiņas ir veiksmīgi uzliktas klientu produkcijas vidēs, tiek izveidota atzīme (*no angļu valodas – “tag”*), ar versijas numuru nosaukumā (piem., “*v1.34*”), kas norāda, ka visas izmaiņas līdz šai atzīmei ir uzliktas produkcijā ar attiecīgo versijas numuru.



8.1.att. LIETVARIS projekta repozitoriju zaru struktūras shēma

Katram jaunam darba uzdevumam tika veidots savs atzars, balstoties uz “lietvaris/develop” zaru. Zara nosaukums tika veidots pēc principiem:

- zaru nosaukuma sākumā jāmin, vai šajā zarā tiks saglabātas jaunas funkcionalitātes programmkods (“*feat/...*”) vai kādas eksistējošas funkcionalitātes uzlabojums (“*fix/...*”);
- zara nosaukumā ir jābūt *Redmine* uzdevuma identifikatoram (“*feat/3282\_...*”);
- zara nosaukumā ir jābūt atslēgvārdiem, kas apraksta, kādas izmaiņas tiks veiktas vai arī kurā sistēmas daļā tiks veiktas izmaiņas (“*feat/3282\_uznemumu\_meklesana\_pec\_VRN*”);
- zara nosaukumā netiek izmantotas diakritiskās zīmes (garumzīmes un mīkstinājuma zīmes).

Pēc katra darba uzdevuma izpildes (funkcionalitātes izveidošanas vai uzlabošanas), izmaiņas tika saglabātas attiecīgajā repozitorijā, ja izmaiņas tika veiktas datu bāzes pusē vai tika veidots *Elasticsearch* projektējums, tad izmaiņas tika saglabātas “lietvaris/database” atdalītajā repozitorijā “Laura/database” un ja izmaiņas tika veiktas .NET projektā, tad izmaiņas tika saglabātas “lietvaris/web” atdalītajā repozitorijā “Laura/web”.

Laura / web  
atdalīts no lietvaris/web

Nevērot 1 | Pievienot zvaigznīti 0 | Atdalīts 0

Kods | Revīzijas 2584 | Laidieni 12 | Iestatījumi

Nav apraksta

Atzars: master | web | HTTPS | SSH | https://git.zzdats.lv/Laura/web

Laura Tirumniece	2be1d4aaaf9	Izlabots #1115 Kontaktpersonas: uz 1024x768 neredz visa lauka nosaukumu	2 mēneši atpakaļ
.vscode	365bffe442	Izlaboti vscode faili	1 gadu atpakaļ
Lib	cd3f048dd7		1 gadu atpakaļ
LietvarisTestFramework	86414dd126	Izlaboti visi warningi	3 mēneši atpakaļ
LietvarisTests	f7927a01bd	Pielikti 2 triviāli Selenium testi	6 mēneši atpakaļ
Zzdats.Cdn	2be1d4aaaf9	Izlabots #1115 Kontaktpersonas: uz 1024x768 neredz visa lauka nosauku...	2 mēneši atpakaļ
Zzdats.FileUpload	3db21299bb	Sākotnējais git	1 gadu atpakaļ

## 8.2.att. LIETVARIS projekta repozitorija “lietvaris/web” atdalītais repozitorijs “Laura/web”

Pēc programmkoda izmaiņu saglabāšanas atdalītajos repozitorijos, lai uzlabojumus ieviestu LIETVARIS projektā, bija jāveic izmaiņu pieprasījums uz “lietvaris/web” vai “lietvaris/database” repozitorijiem. Izmaiņu pieprasījumu izskata citi projekta izstrādātāji. Ja viss ir korekti, tad izmaiņu pieprasījums tiek sapludināts ar “lietvaris/database” repozitorija zaru “develop” vai “lietvaris/web” repozitorija “develop” zaru.

lietvaris / web

Vērot 6 Pievienot zvaigznīti 2 Atdalīts 7

Kods Izmaiņu pieprasījumi 9 Revīzijas 3472 Laidieni 12 Vikivietne

Etiketes Atskaites punkti Jauns izmaiņu pieprasījums

#534 Labots #3282 - uzņēmumu meklēšana pēc reģistrācijas numura [noņemts komentārs] Labot

Atvērta Laura vēlas sapludināt 1 revīzijas no Laura/fix/3282\_meklesana\_pec\_VRN uz lietvaris/develop

Saruna 0 Revīzijas 1 Izmainītie faili 1

Laura komentēja 1 minūti atpakaļ  
#3282

✓ Šo izmaiņu pieprasījumu var automātiski sapludināt.

Rakstīt Priekšskatījums

Ievietiet failus šeit vai noklikšķiniet, lai augšupielādētu.

Etiketes  
Nav etiķešu

Atskaites punkts  
Nav atskaites punktu

Atbildīgais  
Nav atbildīgā

1 dalībnieki

### 7.3.att. Izmaiņu pieprasījums “lietvaris/web” repositoijā

Lai nodrošinātu ātru un efektīvu programmkoda izmaiņu saglabāšanu, kā arī atjaunotu uz lokālās darbstacijas esošo projektu ar citu izstrādātāju izmaiņām, tika izmantots *Git* klienta rīks *TortoiseGit*. *TortoiseGit* rīka izmaiņu saglabāšanas piemēru skatīt 6.4. attēlā.

## 9. TESTĒŠANAS DOKUMENTĀCIJA

Šajā dokumentā tiek aprakstīti testi, kas tika izmantoti, lai testētu pievienotās un labotās sistēmas LIETVARIS funkcionalitātes. Darba uzdevumu izstrādes laikā, pēc katra labojuma vai funkcionalitātes papildinājuma tika veikti testi, lai pārbaudītu izstrādātā programmkoda darbību, lai novērstu kļūdas uzreiz, ja tādas bija, kā arī pārlicinātos, vai izstrādātā funkcionalitāte atbilst lietotājistāstiem.

Pēc tam, kad izstrādātājs ir pats notestējis savus veiktos labojumus, tālāk izstrādātos labojumus sistēmā testē LIETVARIS projekta testētāji. Testētāji, ja atrod kādu kļūdu vai nepilnību, atgriež darba uzdevumu atpakaļ ar kļūdu aprakstiem izstrādātājiem.

Dokuments saistīts un lietojams kopā ar dokumentiem “Lietotājistāsti” un “Programmatūras projektējuma apraksts”.

### 9.1. Testēšanas žurnāls

#### 9.1.1. Testēšanas plāns

Testēšanas laikā jāveic sekojošas pārbaudes:

- jāpārbauda ievaddatu validācija un korektums;
- jāpārbauda, vai izvaddati sakrīt ar sagaidāmo rezultātu;
- jāpārbauda kļūdas paziņojumu atgriešana;
- jāpārbauda saskarņu atbilstība lietotājistāstiem.

Testēšana tika veikta divos veidos: tika izstrādāti vienībtesti un tika veikta funkcionālā testēšana.

#### 9.1.2. Testu rezultātu apzīmējumi

*9.1.Tabula*

##### Testa rezultātu apzīmējumi

Apzīmējums	Apraksts
OK	Tests izpildījās veiksmīgi.
ER	Tests neizpildījās, ir problēmas vai kļūdas.

#### 9.1.3. Funkcionālā testēšana

Kontaktpersonu modulī veiktās izmaiņas un uzlabojumi tika testēti pēc melnās kastes principa. Ievērojot definētos lietotājistāstus, tika izveidoti akceptējošie kritēriji jeb scenāriji, pēc

kuriem notika manuālā testēšana. Testēšana notika katru reizi pēc funkcionalitātes uzlabojuma vai jaunas funkcionalitātes izstrādes beigām.

“Dzēst” piekļuves tiesības ietver visas “Labot” piekļuves tiesības, tāpēc funkcionālā testēšana tika veikta ar lietotāju, kuram ir “Labot” tiesības. Pieejas tiesības skatīt 3.1.tabulā.

### 9.1.3.1.Lietotājstāstu funkcionālā testēšana

Testēšanas scenāriji tika veidoti pēc definētajiem lietotājstāstiem (skatīt 3.nodaļu).

9.2.Tabula

#### Funkcionālās testēšanas žurnāls

Testa nr.	DU identifikators	Akceptējošie kritēriji	Sagaidāmais rezultāts	Statuss
1.	DU4783	<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>Jaunas kontaktpersonas pievienošanas formā tiek nospiesta poga “Atgriezties”.</li> </ul>	<ul style="list-style-type: none"> <li>Jaunā kontaktpersonas pievienošanas forma atveras.</li> <li>Jaunas kontaktpersonas pievienošanas forma aizveras.</li> </ul>	OK
2.		<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Skatīt” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>Jaunas kontaktpersonas pievienošanas formā tiek nospiesta poga “Atgriezties”.</li> </ul>	<ul style="list-style-type: none"> <li>Lietotājam nav iespējas atvērt jaunas kontaktpersonas pievienošanas formu, līdz ar to pogu “Atgriezties” nevar nospiest.</li> </ul>	OK
1.	DU2592	<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Labot” atver kontaktpersonas kartīti jaunā cilnē.</li> <li>Lietotājs nospiež pogu “Aizvērt logu”.</li> </ul>	<ul style="list-style-type: none"> <li>Kontaktpersonas kartīte atveras jaunā cilnē.</li> <li>Lietotājs neredz pogu “Atgriezties”.</li> <li>Lietotājs redz pogu “Aizvērt logu”.</li> </ul>	OK

			<ul style="list-style-type: none"> <li>• Cilne, kurā ir atvērta dokumentu kartīte, aizveras.</li> </ul>	
2.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Skatīt” atver kontaktpersonas kartīti jaunā cilnē.</li> <li>• Lietotājs nospiež pogu “Aizvērt logu”.</li> </ul>	<ul style="list-style-type: none"> <li>• Lietotājam nav piekļuves kontaktpersonu kartītei, tā netiek atvērta.</li> <li>• Pogu “Aizvērt logu” nevar nospiež.</li> </ul>	OK
1.	DU1115	<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>• Lietotājs maina loga izmērus.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>• Formas ievadlauku nosaukumi nepazūd, visi ir redzami.</li> </ul>	OK
2.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Skatīt” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>• Lietotājs maina loga izmērus.</li> </ul>	<ul style="list-style-type: none"> <li>• Neatveras jaunas kontaktpersonas pievienošanas forma, loga izmērus nevar mainīt.</li> </ul>	OK
1.	DU2964	<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver kontaktpersonas kartīti.</li> </ul>	<ul style="list-style-type: none"> <li>• Kontaktpersonas kartīte atveras. Poga “Saglabāt” ir vienāda stila un izmēra kā citas pogas.</li> </ul>	OK
2.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Skatīt” atver kontaktpersonas kartīti.</li> </ul>	<ul style="list-style-type: none"> <li>• Kontaktpersonas kartīte neatveras, pogu izmērus nevar apskatīt.</li> </ul>	OK

1.	DU1123	<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Labot” atver kontaktpersonas kartīti.</li> <li>Lietotājs atver “Sarakste” šķirkli.</li> <li>Lietotājs atver “Notikumi” šķirkli.</li> </ul>	<ul style="list-style-type: none"> <li>Atveras kontaktpersonas kartīte.</li> <li>Atveras “Sarakste” šķirklis. Lietotājs neredz lieku tekstu, kas dublē šķirkļa nosaukumu.</li> <li>Atveras “Nosaukums” šķirklis. Lietotājs neredz lieku tekstu, kas dublē šķirkļa nosaukumu.</li> </ul>	OK
2.		<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Skatīt” atver kontaktpersonas kartīti.</li> <li>Lietotājs atver “Sarakste” šķirkli.</li> <li>Lietotājs atver “Notikumi” šķirkli.</li> </ul>	<ul style="list-style-type: none"> <li>Kontaktpersonu kartīte neatveras.</li> <li>Šķirklis “Sarakste” neatveras.</li> <li>Šķirklis “Nosaukums” neatveras.</li> </ul>	OK
1.	DU4406	<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Labot” atver kontaktpersonas kartīti.</li> <li>Lietotājs ievadlaukā “Personas kods” ievada jau sistēmā eksistējošas kontaktpersonas personas kodu.</li> <li>Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>Atveras kontaktpersonas kartīte.</li> <li>Ievadlaukā ievadīts personas kods.</li> <li>Tiek atgriezts kļūdas paziņojums SP01.</li> </ul>	OK
2.		<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Skatīt” atver kontaktpersonas kartīti.</li> </ul>	<ul style="list-style-type: none"> <li>Neatveras kontaktpersonas kartīte.</li> <li>Nevar ievadīt personas kodu.</li> </ul>	OK

		<ul style="list-style-type: none"> <li>• Lietotājs ievadlaukā “Personas kods” ievada jau sistēmā eksistējošas kontaktpersonas personas kodu.</li> <li>• Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>• Nevar nospiegt pogu “Saglabāt”.</li> </ul>	
3.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver kontaktpersonas kartīti.</li> <li>• Kontaktpersonas veids ir “Fiziska persona”</li> <li>• Lietotājs ievadlaukā “Personas kods” ievada unikālu, korektu personas kodu.</li> <li>• Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras kontaktpersonas kartīte.</li> <li>• Ir atvērta kontaktpersona, kura ir fiziska persona.</li> <li>• Ievadlaukā ir ievadīts korekts personas kods.</li> <li>• Izmaiņas tiek saglabātas.</li> </ul>	OK
4.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver kontaktpersonas kartīti.</li> <li>• Kontaktpersonas veids ir “Juridiska persona”</li> <li>• Lietotājs ievadlaukā “Reģistrācijas numurs” ievada unikālu, korektu reģistrācijas numuru.</li> <li>• Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras kontaktpersonas kartīte.</li> <li>• Ir atvērta kontaktpersona, kura ir juridiska persona.</li> <li>• Ievadlaukā ir ievadīts korekts reģistrācijas numurs.</li> <li>• Izmaiņas tiek saglabātas.</li> </ul>	OK

5.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver kontaktpersonas kartīti.</li> <li>• Kontaktpersonas veids ir “Ārvalstnieks”</li> <li>• Lietotājs ievadlaukā “Reģ.nr./pers.kods” ievada unikālu, korektu reģ.nr./pers.kodu.</li> <li>• Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras kontaktpersonas kartīte.</li> <li>• Ir atvērta kontaktpersona, kura ir ārvalstnieks.</li> <li>• Ievadlaukā ir ievadīts korekts reģistrācijas numurs/personas kods.</li> <li>• Izmaiņas tiek saglabātas.</li> </ul>	OK
6.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver kontaktpersonas kartīti.</li> <li>• Kontaktpersonas veids ir “Fiziska persona”</li> <li>• Lietotājs ievadlaukā “Personas kods” ievada 10 simbolu garu personas kodu.</li> <li>• Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras kontaktpersonas kartīte.</li> <li>• Ir atvērta kontaktpersona, kura ir fiziska persona.</li> <li>• Ievadlaukā ir ievadīts personas kods</li> <li>• Tiek atgriezts kļūdas paziņojums SP02.</li> </ul>	OK
7.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver kontaktpersonas kartīti.</li> <li>• Kontaktpersonas veids ir “Juridiska persona”</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras kontaktpersonas kartīte.</li> <li>• Ir atvērta kontaktpersona, kura ir juridiska persona.</li> </ul>	OK

		<ul style="list-style-type: none"> <li>Lietotājs ievadlaukā “Reģistrācijas numurs” ievada reģistrācijas numuru, kas ir īsāks par 11 simboliem.</li> <li>Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>Ievadlaukā ir ievadīts nekorekts reģistrācijas numurs.</li> <li>Tiek atgriezts kļūdas paziņojums SP03.</li> </ul>	
1.	DU4863	<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>Lietotājs izvēlās kontaktpersonas veidu “Ārvalstnieks”.</li> <li>Formas ievadlaukā “Reģ.nr./pers.kods” lietotājs ievada 14 simbolu garu reģistrācijas numuru.</li> <li>Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>Kontaktpersonas veids ir “Ārvalstnieks”.</li> <li>Ievadlaukā ievadīts korekts reģistrācijas numurs.</li> <li>Lietotājam nospiežot pogu “Saglabāt”, tiek atgriezts kļūdas paziņojums par obligāto lauku neaizpildīšanu.</li> </ul>	OK
2.		<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>Lietotājs izvēlās kontaktpersonas veidu “Ārvalstnieks”.</li> </ul>	<ul style="list-style-type: none"> <li>Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>Kontaktpersonas veids ir “Ārvalstnieks”.</li> <li>Ievadlaukā ievadīts korekts reģistrācijas numurs.</li> <li>Formas lauks “Nosaukums” satur ārvalstnieka vārdu.</li> </ul>	ER

		<ul style="list-style-type: none"> <li>Formas ievadlaukā “Reģ.nr./pers.kods” lietotājs ievada 13 simbolu garu reģistrācijas numuru.</li> <li>Formas obligātais lauks “Nosaukums” tiek aizpildīts ar ārvalstnieka vārdu.</li> <li>Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>Tiek atgriezts paziņojums par nekorekta reģistrācijas numura/personas koda ievadi.</li> </ul>	
3.		<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Skatīt” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>Lietotājs izvēlās kontaktpersonas veidu “Ārvalstnieks”.</li> <li>Formas ievadlaukā “Reģ.nr./pers.kods” lietotājs ievada 14 simbolu garu reģistrācijas numuru.</li> <li>Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>Neatver jaunas kontaktpersonas pievienošanas formu.</li> <li>Nevar izvēlēties kontaktpersonas veidu.</li> <li>Nevar aizpildīt formas laukus.</li> <li>Nevar nospiež pogu “Saglabāt”.</li> </ul>	OK
4.		<ul style="list-style-type: none"> <li>Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>Lietotājs izvēlās kontaktpersonas veidu “Ārvalstnieks”.</li> </ul>	<ul style="list-style-type: none"> <li>Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>Kontaktpersonas veids ir “Ārvalstnieks”.</li> <li>Ievadlaukā ievadīts korekts reģistrācijas numurs.</li> </ul>	OK

		<ul style="list-style-type: none"> <li>• Formas ievadlaukā “Reģ.nr./pers.kods” lietotājs ievada 14 simbolu garu reģistrācijas numuru.</li> <li>• Formas obligātais lauks “Nosaukums” tiek aizpildīts ar ārvalstnieka vārdu. Lietotājs nospiež pogu “Saglabāt”.</li> </ul>	<ul style="list-style-type: none"> <li>• Formas lauks “Nosaukums” satur ārvalstnieka vārdu.</li> <li>• Kontaktpersona tiek saglabāta.</li> </ul>	
1.	DU3282	<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersona pievienošanas formu.</li> <li>• Lietotājs izvēlās kontaktpersonas veidu “Juridiska persona”.</li> <li>• Formas ievadlaukā “Reģistrācijas numurs” lietotājs ievada 11 simbolu garu reģistrācijas numuru.</li> <li>• Lietotājs nospiež pogu “Meklēt UR”.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>• Ir izvēlēts kontaktpersonas veids “Juridiska persona”.</li> <li>• Ievadlaukā ir ievadīts korekts reģistrācijas numurs.</li> <li>• Formas ievadlauki “Nosaukums” un “Adrese” tiek aizpildīti ar attiecīgajiem juridiskās personas datiem (pēc ievadītā reģistrācijas numura).</li> </ul>	OK
2.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersona pievienošanas formu.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>• Ir izvēlēts kontaktpersonas veids “Juridiska persona”.</li> </ul>	OK

		<ul style="list-style-type: none"> <li>• Lietotājs izvēlās kontaktpersonas veidu “Juridiska persona”.</li> <li>• Formas ievadlaukā “Reģistrācijas numurs” lietotājs ievada 11 simbolu garu uzņēmuma reģistrācijas numuru, kas ir likvidēts, tas ir, nav reģistrēts.</li> <li>• Lietotājs nospiež pogu “Meklēt UR”.</li> </ul>	<ul style="list-style-type: none"> <li>• Ievadlaukā ir ievadīts reģistrācijas numurs.</li> <li>• Tiek atgriezts kļūdas paziņojums SP05.</li> </ul>	
3.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersona pievienošanas formu.</li> <li>• Lietotājs izvēlās kontaktpersonas veidu “Juridiska persona”.</li> <li>• Formas ievadlaukā “Reģistrācijas numurs” lietotājs ievada nekorektu reģistrācijas numuru, tas ir, reģistrācijas numurs ir mazāks par 11 simboliem vai tas satur burtus.</li> <li>• Lietotājs nospiež pogu “Meklēt UR”.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>• Ir izvēlēts kontaktpersonas veids “Juridiska persona”.</li> <li>• Ievadlaukā ir ievadīts reģistrācijas numurs.</li> <li>• Tiek atgriezts kļūdas paziņojums SP04.</li> </ul>	OK

4.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Skatīt” atver jaunas kontaktpersona pievienošanas formu.</li> <li>• Lietotājs izvēlās kontaktpersonas veidu “Juridiska persona”.</li> <li>• Formas ievadlaukā “Reģistrācijas numurs” lietotājs ievada 11 simbolu garu reģistrācijas numuru.</li> <li>• Lietotājs nospiež pogu “Meklēt UR”.</li> </ul>	<ul style="list-style-type: none"> <li>• Nevar atvērt jaunas kontaktpersonas pievienošanas formu.</li> <li>• Nevar izvēlēties kontaktpersonas veidu.</li> <li>• Nevar ievadīt reģistrācijas numuru.</li> <li>• Nevar nospiegt pogu “Meklēt UR”.</li> </ul>	OK
5.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersona pievienošanas formu.</li> <li>• Lietotājs izvēlās kontaktpersonas veidu “Juridiska persona”.</li> <li>• Formas ievadlauku “Reģistrācijas numurs” lietotājs atstāj tukšu.</li> <li>• Lietotājs nospiež pogu “Meklēt UR”.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>• Ir izvēlēts kontaktpersonas veids “Juridiska persona”.</li> <li>• Ievadlauks ir tukšs.</li> <li>• Tiek atgriezts sistēmas kļūdas paziņojums.</li> </ul>	ER
1.	DU3284	<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersonas pievienošanas formu.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras jaunas kontaktpersonas pievienošanas forma.</li> </ul>	OK

		<ul style="list-style-type: none"> <li>• Lietotājs izvēlās kontaktpersonas veidu “Juridiska persona”.</li> <li>• Lietotājs nospiež pogu “Uzņēmuma reģistra dati”.</li> <li>• Lietotājs ievadlaukā “Brīvā meklēšana” ievada atslēgvārdus <i>ZZ Dats</i>.</li> <li>• Lietotājs nospiež pogu “Meklēt”.</li> <li>• Lietotājs nospiež pogu “Pievienot”.</li> </ul>	<ul style="list-style-type: none"> <li>• Ir izvēlēts kontaktpersona veids “Juridiska persona”.</li> <li>• Atveras dialoga logs “Uzņēmuma meklēšana”.</li> <li>• Ievadlaukā ievadīts atslēgvārds.</li> <li>• Tiek atlasīts uzņēmums. Tiek atgriezts informatīvais paziņojums SP08.</li> <li>• Jaunas kontaktpersonas pievienošanas formas lauki “Nosaukums”, “Reģistrācijas numurs” un “Adrese” tiek aizpildīti ar atlasītās juridiskās personas datiem.</li> </ul>	
2.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>• Lietotājs izvēlās kontaktpersonas veidu “Juridiska persona”.</li> <li>• Lietotājs nospiež pogu “Uzņēmuma reģistra dati”.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>• Ir izvēlēts kontaktpersona veids “Juridiska persona”.</li> <li>• Atveras dialoga logs “Uzņēmuma meklēšana”.</li> <li>• Ievadlaukā ievadīts atslēgvārds.</li> </ul>	OK

		<ul style="list-style-type: none"> <li>• Lietotājs ievadlaukā “Brīvā meklēšana” ievada atslēgvārdus, pēc kā nevar atlasīt juridisku personu.</li> <li>• Lietotājs nospiež pogu “Meklēt”.</li> </ul>	<ul style="list-style-type: none"> <li>• Tiek atgriezts informatīvs paziņojums SP07.</li> </ul>	
3.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Labot” atver jaunas kontaktpersonas pievienošanas formu.</li> <li>• Lietotājs izvēlās kontaktpersonas veidu “Juridiska persona”.</li> <li>• Lietotājs nospiež pogu “Uzņēmuma reģistra dati”.</li> <li>• Lietotājs ievadlaukā “Brīvā meklēšana” ievada atslēgvārdus ZZ (atslēgvārdu simbolu skaits mazāks par 3).</li> <li>• Lietotājs nospiež pogu “Meklēt”.</li> </ul>	<ul style="list-style-type: none"> <li>• Atveras jaunas kontaktpersonas pievienošanas forma.</li> <li>• Ir izvēlēts kontaktpersona veids “Juridiska persona”.</li> <li>• Atveras dialoga logs “Uzņēmuma meklēšana”.</li> <li>• Ievadlaukā ievadīts atslēgvārds.</li> <li>• Tiek atgriezts informatīvais paziņojums SP06.</li> </ul>	OK
4.		<ul style="list-style-type: none"> <li>• Lietotājs ar tiesībām “Skatīt” atver jaunas kontaktpersonas pievienošanas formu.</li> </ul>	<ul style="list-style-type: none"> <li>• Nevar atvērt jauns kontaktpersonas pievienošanas formu.</li> </ul>	OK

### 9.1.3.2.Kļūdu apstrāde

Testēšanas laikā atrastās kļūdas tiek reģistrētas testēšanas kļūdu apstrādes žurnālā. Pēc kļūdas labojuma, tiek veikta atkārtota testēšana.

9.3.Tabula

#### Kļūdu apstrādes žurnāls

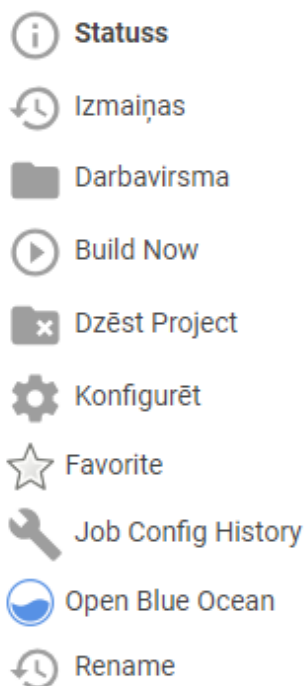
Testa nr.	DU identifikators	Kļūda	Labojums	Statuss
2.	DU4863	Lietotājs jaunas kontaktpersonas pievienošanas formas laukā "Reģ.nr./pers.kods" nevar ievadīt mazāk par 14 simboliem.	Tā kā ārvalstnieku reģistrācijas numuriem vai personas kodiem var būt dažāds simbolu skaits, tika veiktas izmaiņas datu bāzes procedūras pārbaudē. Tika nomainīts parametrs, ka reģistrācijas numurs vai personas kods var būt no 1 līdz 14 simboliem garš.	OK
5.	DU3282	Lietotājam tiek atgriezts nekorekts kļūdas paziņojums par to, ka ievadlauks ir tukšs ( <i>no angļu valodas – "undefined"</i> ) un notiek JavaScript kļūda.	Funkcija <i>findCompany</i> tika papildināta ar ievadlauka pārbaudēm. Tika pievienota pārbaude, ja ievadlauks ir tukšs vai nenoteikts, tad tiek atgriezts kļūdas paziņojums SP09.	OK

#### 9.1.4. Automatizētā vienībtestēšana

Vienībtesti tiek rakstīti projekta lietotāja “dev\_lietvaris\_ut” datu bāzē. Tika izveidota jauna pakotne UT\_GETCOMPANY, kurā tika izstrādāti vienībtesti šī kvalifikācijas darba testēšanai.

Vienībtestā tiek izveidota procedūra, kas saņemot ieejas datus – darba uzdevuma DU3282 gadījumā reģistrācijas kodu – izpilda procedūru, kura tiek testēta. Tad tiek salīdzinātas iegūtās vērtības ar procedūrā manuāli nodefinētām vēlamajām vērtībām. Ja vērtības sakrīt, tad tests ir nostrādājis veiksmīgi. Vienībtesta programmkoda fragmentu skatīt 8.pielikumā.

Vienībtesti tiek automātiski reizi diennaktī darbināti, izmantojot automatizācijas platformu *Jenkins*, taču izstrādātājs var pats manuāli palaist to darbību nospiežot pogu “Build Now” (skatīt 9.1.attēlu).



9.1.att. *Jenkins* automatizētās platformas rīkjosla automatizācijas platformā *Jenkins*

Kļūdu gadījumos tiek saņemts paziņojums uz e-pastu (tiem izstrādātājiem, kuriem konfigurācijā ir norādīts e-pasts) vai uz tērzēšanas rīku *Mattermost*.


Testu rezultātus ir iespējams apskatīt (skatīt 9.2.attēlu). Šajā skatā var redzēt, cik testi ir nostrādājuši un cik ir izgāzušies.

## Test Result : UT\_GETCOMPANY

0 neveiksmīgi testi (±0)

5 testi (±0)

Ilga 0 ms

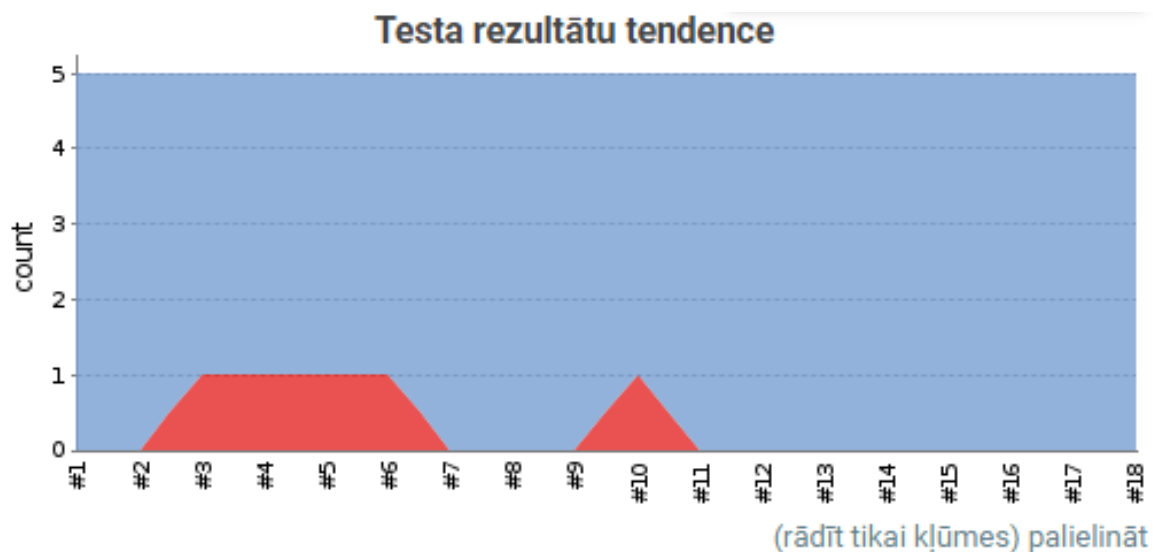
 Pievienot aprakstu

### Visi testi

Testa nosaukums	Ilgums	Stāvoklis
1UT_GETCOMPANYDATAERROR	0 ms	Passed
2UT_GETCOMPANYDATAERROR	0 ms	Passed
3UT_GETCOMPANYDATAERROR	0 ms	Passed
4UT_GETCOMPANYDATAERROR	0 ms	Passed
5UT_GETCOMPANYDATAOK	0 ms	Passed

9.2..att. Automātiskās vienībtestēšanas rezultātu apskatīšana automatizācijas platformā *Jenkins*

Vienībtestu rezultātus ir iespējams apskatīt arī grafiskā veidā (skatīt 8.3.attēlu).



9.3.att. Automatizētās vienībtestēšanas rezultātu grafisks attēlojums automatizācijas platformā *Jenkins*

## 9.2. Testēšanas secinājumi

Testēšanas laikā atklātās kļūdas ir veiksmīgi novērstas. Tika izpildīti testēšanas plānā visi nostādītie punkti (skatīt 9.1.1. nodaļu). Sistēmas testētās funkcionalitātes darbojas korekti.

## 10.DARBIETILPĪBAS NOVĒRTĒJUMS

Darbietilpības uzskaitē veikta katru reizi pēc darba uzdevuma izpildes, norādot patērēto laiku *Redmine* projektu un resursu plānošanas un vadības rīkā. Ņemot vērā, ka LIETVARIS projekts vadās pēc spējās programmatūras izstrādes principiem, darbietilpības novērtēšanai izmantoti stāstu punkti (*no angļu valodas – “story points”*) kā plānotā laika mērvienība. Faktiski patērētais laiks novērtēts stundās. Veicot darbietilpības novērtēšanu, ņemts vērā, ka darba autoram nav bijusi pieredze darbā ar izstrādes rīkiem, tehnoloģijām un programmēšanas valodām, kuras izmanto LIETVARIS projekta izstrādē.

10.1.Tabula

### Darbietilpības novērtējums

Nosaukums		Story points	Faktiski patērētais laiks (h)
Lietotārstāsti		30	24
PPA		60	80
DU4783		3	1
DU2592		3	6
DU1115		5	10
DU2964		1	0.25
DU1123		1	0.25
DU4406		2	2
DU4863		5	5
DU3282	Pakotnes I_UR_INTEGRATION_PCK izstrāde	15	35
	Pakotnes APP_URINTEGRATION_PCK izstrāde	5	3
	Funkcijas <i>findCompany</i> izstrāde	10	20
	Datu bāzes un .NET savienošana	10	20
	Saskarnes labojumi	1	2
DU3284	Pakotnes APP_URINTEGRATION_PCK izstrāde	8	40
	Pakotnes COMPANY_ES_IMPORT_PCK izstrāde	9	30
	Funkcijas <i>urSearch</i> izstrāde	3	16

	Funkcijas <i>onSelect</i> izstrāde	1	3
	Citu svarīgu komponentu izstrāde .NET (piem., kontrolieris u.c.)	5	25
	Dialoga loga veidne, pogu izveidošana	3	20
	<i>Elasticsearch</i> indeksa projektējums un savienošana ar .NET	15	16
	Testēšana	40	30
	Iepazīšanās ar izstrādes rīkiem, tehnoloģijām un vidēm	60	60
	Kvalifikācijas darba pārējo nodaļu izstrāde	40	20
	Testēšanas dokumentācijas izstrāde	20	17
	Dokumenta noformēšana	8	11
	<b>KOPĀ</b>	<b>361</b>	<b>496,5</b>

Plānotais patērētais laiks ir pielīdzināms 361 stāstu punktiem. Viens sprints pielīdzināms 10 cilvēkdienām un 10 cilvēkdienas ir pielīdzināmas 80 cilvēkstundām, ja vienā cilvēkdienā tiek strādāts 8 cilvēkstundas. 3 cilvēkmēnešos ir vidēji 6 sprinti, kas nozīmē, ka 3 cilvēkmēneši ir pielīdzināmi 480 cilvēkstundu darbam.

Ņemot vērā, ka vienā cilvēkdienā ir 8 darba cilvēkstundas, 1 cilvēkmēnesis ir vidēji 20 cilvēkdienas, tad vienā cilvēkmēnesī ir 160 stundas. Var secināt, ka faktiskais patērētais laiks atbilsts 3,1 cilvēkmēnešiem.

Pēc aprēķinu veikšanas, var spriest, ka faktiski patērētais laiks (496,5 cilvēkstundas) pārsniedz plānoto patērēto laiku (480 cilvēkstundas) par 16,5 cilvēkstundām un var secināt, ka faktiski patērētais laiks pārsniedz 3 mēnešu darbietilpību.

## 11.REZULTĀTI

Kvalifikācijas darbā izvirzītais mērķis ir sasniegts. Izstrādāts risinājums, kas atvieglotu lietvedības sistēmas LIETVARIS lietotāju darbu kontaktpersonu modulī, nodrošinātu juridisko personu meklēšanu un uzlabotu sistēmas ērtu lietošanu. Izvirzītie uzdevumi mērķa sasniegšanai izpildīti.

Izstrādātais darba uzdevums “Labojumi kontaktpersonu modulī” ir ieviests LIETVARIS sistēmas produkcijā. Darba uzdevums DU3282 un darba uzdevums DU3284 ir gatavi ieviešanai LIETVARIS sistēmas produkcijā.

Pēc darba uzdevumu ieviešanas sistēmas produkcijā, ir plānots, ka darba uzdevums DU3282 tiks papildināts ar iespēju izvēlēties juridiskās personas adresi, ja tai būs vairākas adreses.

## 12.SECINĀJUMI

Izstrādājot kvalifikācijas darbu, iegūta neatsverama pieredze darbā ar *Oracle PL/SQL Developer*, *Microsoft Visual Studio .NET* izstrādes vidēm un apgūtas programmēšanas valodas *Oracle PL/SQL*, *C#*. Apgūts darbs ar *Elasticsearch* tehnoloģiju, *Git* versionēšanas sistēmu.

Teorētiski apgūtās zināšanas tika pielietotas praktiski, kas deva dziļāku izpratni par programmēšanas valodām - *JavaScript*, *HTML*, *CSS* - , kā arī par IT nozares specifiku un darba vidi.

Kvalifikācijas darba izstrādes laikā iegūtās zināšanas un atziņas palīdzēs tālākai sevis pilnveidošanai IT nozarē.

## IZMANTOTIE LITERATŪRAS AVOTI

1. ZZ Dats vikipēdija [tiešsaiste] – [atsauce 20.05.2018.] Pieejams, autentificējoties kā ZZ Dats darbiniekam: <http://wiki.zzdats.lv/>
2. Oracle Help Center [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <https://docs.oracle.com/en/>
3. Oracle Community [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <https://community.oracle.com/>
4. Pluralsight [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <https://app.pluralsight.com/library/>
5. Akadēmiskā terminu vārdnīca [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <http://termini.lza.lv/term.php?>
6. Balsamiq Cloud [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <https://balsamiq.cloud/>
7. Draw.io [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <https://www.draw.io/>
8. Lursoft [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <https://www.lursoft.lv/lv/uznemumu-datu-bazes>
9. Uzņēmumu reģistrs [tiešsaiste] – [atsauce 26.05.2018.] Pieejams: <https://www.ur.gov.lv/lv/>
10. Elastic [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <https://www.elastic.co/guide/index.html>
11. Javascript Tutorial [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <https://www.w3schools.com/js/default.asp>
12. Tavirekini.lv [tiešsaiste] – [atsauce 20.05.2018.] Pieejams: <https://www.tavirekini.lv/noderigi/2018-gada-darba-dienu-kalendars>
13. .NET [tiešsaiste] – [atsauce 24.05.2018.] Pieejams: <https://www.microsoft.com/net/learn/what-is-dotnet>
14. AngularJS [tiešsaiste] – [atsauce 24.05.2018.] Pieejams: <https://angularjs.org/>

## PIELIKUMS

1.pielikums

Trasējāmības tabula starp lietotājtāstiem un *Redmine* DU

<b>DU identifikators</b>	<b><i>Redmine</i> DU numurs</b>	<b>Lietotājtāsta nodaļa</b>
DU4783	#4783	3.2.1.
DU2592	#2592	3.2.2.
DU1115	#1115	3.2.3.
DU2964	#2964	3.2.4.
DU1123	#1123	3.2.5.
DU4406	#4406	3.2.6.
DU4863	#4863	3.2.7.
DU3282	#3282	3.2.8.
DU3284	#3284	3.2.9.

```
[HttpPost]
    public IActionResult GetDataUsingRegCode([FromBody] string
request)
    {
        return
Content(_oracle.Execute("URIntegration.GetDataUsingRegCode",
request), "application/json");
    }
}
```

```
$scope.urSearch = function () {
    /// <summary>
    /// Funkcija, kas atgriež dialoga logu
    /// </summary>
    var dlgCompanySearch = { // definē dialoga loga opcijas
        angularAppName: 'Zzdats.Lietvaris',
        title: 'Uzņēmuma meklēšana', // dialoga loga nosaukums
        templateUrl: 'lietvaris/template/search-select/companySearch.html', //
dialoga loga veidne
        dialogClassName: 'modal60', // izmērs
        data: {
            onSelect: $scope.onSelect // funkcija, kura tiks izpildīta, kad
nospiedīs pogu 'Pievienot'
        }
    };
    dialogService.showDialog2(dlgCompanySearch); // atgriež dialoga logu
};
```

**Funkcijas *onSelect* programmkods**

```
$scope.onSelect = function (data) {  
    /// <summary>  
    /// Funkcija, kas aizpilda jaunas kontaktpersonas peivienošanas formas  
    laukus  
    /// 'Nosaukums', 'Reģistrācijas numurs', 'Adrese' ar izvēlētās juridiskās  
    personas datiem (nospiežot pogu 'Pievienot' dialoga loga 'Uzņēmuma meklēšana')  
    /// </summary>  
    /// <param name="contact" type="type"></param>  
    $scope.$apply(function () {  
  
        $scope.contactPerson.lastName = data.name; // aizpilda nosaukumu  
        $scope.contactPerson.code = data.regcode; // aizpilda reģistrācijas  
        numuru  
        $scope.primaryAddress = data.address[0]; // aizpilda adresi  
  
    });  
};
```

```
{
  "properties" : {
    "_all2" : {
      "type" : "text",
      "analyzer" : "lowercasewhitespace"
    },
    "_content" : {
      "type" : "text",
      "analyzer" : "latvian"
    },
    "name" : { // juridiskās personas nosaukums
      "type" : "text",
      "analyzer" : "lowercasewhitespace",
      "copy_to" : ["_content", "_all2"]
    },
    "regcode" : { // juridiskās personas reģistrācijas numurs
      "type" : "text",
      "copy_to" : ["_content", "_all2"],
      "fields" : {
        "keyword" : {
          "type" : "keyword"
        }
      }
    },
    "address" : { // juridiskās personas strukturētā adrese
      "type": "nested", // objekta tips
      "properties" : {
        "id" : {
          "type" : "long"
        },
        "text" : { // ielas nosaukums
          "type" : "text",
          "analyzer" : "lowercasewhitespace",
          "copy_to" : ["_content", "_all2"]
        },
        "placeText" : { // pilsētas nsoaukums
          "type" : "text",
          "analyzer" : "lowercasewhitespace",
          "copy_to" : ["_content", "_all2"]
        },
        "postalCode" : { // pasta indekss
          "type" : "text",
          "analyzer" : "lowercasewhitespace",
          "copy_to" : ["_content", "_all2"]
        },
        "fullText" : { //pilnā adrese
          "type" : "text",
          "analyzer" : "lowercasewhitespace",
          "copy_to" : ["_content", "_all2"]
        },
        "country" : { // valsts
          "type" : "object",
          "properties" :{
            "id" : { // valsts identifikators
              "type" : "long"
            }
          }
        },
      }
    }
  }
}
```

```
    "name" : { //valts nosaukums latviešu valodā
    "type" : "text",
    "copy_to" : ["_content", "_all2"]
  }
}
}
}
```

**I\_UR\_INTEGRATION\_PCK procedūras *GetDataUsingRegCode* apraksts**

```

--<GetDataUsingRegCode description="Uzņēmuma datu izgūšana pēc
reģistrācijas numura" >
  -- <pi_regCode      type="string"    description="Vienotais NM
reģistrācijas numurs">
  -- <po_data        type="object"    description="Uzņēmuma dati (adrese,
nosaukums, reģistrācijas numurs)">
  --   <CompanyInfo  description="Uzņēmuma dati">
  --     <Name        type="string"    description="Uzņēmuma juridiskais
nosaukums">
  --     <RegCode     type="string"    description="Vienotais NM
reģistrācijas numurs">
  --     <Address     description="Uzņēmuma adrese">
  --       <Id        type="long"      description="Adrešes
identifikators">
  --         <Text     type="string"    description="Adrešes ielas,
mājavārda vai mājas un dzīvokļa numura tekstuālā daļa">
  --         <PlaceText type="string"  description="Adrešes pilsētas,
ciema un novada tekstuālā daļa">
  --         <PostalCode type="string"  description="Pasta indekss">
  --         <Country  description="Valsts">
  --         <Id       type="int"      description="Valsts
identifikators">
  --         <Name     type="string"    description="Valsts nosaukums">
  --         </Country>
  --         </Address>
  --       </CompanyInfo>
  --   </po_data>
  --   <po_errors     description="Kļūdu saraksts" />
--</GetDataUsingRegCode>

```

I\_UR\_INTEGRATION\_PCK procedūras *GetDataUsingRegCode* programmkods

```

procedure GetDataUsingRegCode (pi_regCode in in_nmaks_jp.nmr_kods%type,
                               po_data    out nclob,
                               po_errors  out errorlist) is

    v_count    in_nmaks_jp.nmr_kods%type;
    v_json     json := json();
    v_jsub     json := json();
    v_jsub2    json := json();
    v_jlist    json_list;

    --mainīgie datu atlasīšanai
    compName   in_nmaks_jp.jur_nos%type;
    regCode    in_nmaks_jp.nmr_kods%type;

begin
    -- pārbauda, vai ir ievadīts reģistrācijas numurs
    if pi_regCode is null then
        return;
    end if;
    -- pārbauda, vai ievadītais reģistrācijas numurs ir atrodams
    select count(*)
    into v_count
    from in_nmaks_jp c
    where nmr_kods = pi_regCode
    and c.aktivs = -1;

    if (v_count = 0) then
        -- ievadīto datu pārbaude
        if (REGEXP_LIKE(pi_regCode, '^([0-9]){11}$') = false) then
            u_error_pck.AddErrorToList(po_errors, null,
                'err:validation:regCode');
        else
            u_error_pck.AddErrorToList(po_errors, null, 'err:404:regCode');
        end if;
    else
        --atlasa vajadzīgos datus
        select jur_nos, nmr_kods
        into compName, regCode
        from in_nmaks_jp ca
        where ca.nmr_kods = pi_regCode
        and ca.aktivs = -1;

        -- veido json
        v_json := json();

        u_json_pck.SetStringProperty(v_json, 'name', compName);
        u_json_pck.SetStringProperty(v_json, 'regcode', regCode);

        -- veido masīvu
        v_jlist := json_list();
        -- atlasa strukturēto adresi
        for v_address in (select co.aid, co.a_vaid_ir, co.adrpastaindex,
            co.apils, co.aadr, cca.vanoslv
            from in_saiteadrpers ca
            inner join address co

```

```

                                on (ca.sapi_aid = co.aid and co.a_akt = -1
and ca.sapi_aktivs = -1)
                                inner join in_kl_adrveids cc
                                on (ca.sapi_adrveids = cc.av_id and
cc.av_kods=11 and cc.av_aktivs = -1)
                                inner join in_nmaks_jp ccc
                                on (ca.sapi_jurpersid = ccc.id_nm and
ccc.aktivs = -1)
                                inner join cl_countries cca
                                on (co.a_void_ir = cca.vakd /*and
cca.vaaktivs = -1*/)
                                where ccc.nmr_kods = pi_regCode) loop

    v_jsub := json();

    --adrese
    u_json_pck.SetIntProperty(v_jsub, 'id', v_address.aid);
    u_json_pck.SetStringProperty(v_jsub, 'text', v_address.aadr); -
--street
    u_json_pck.SetStringProperty(v_jsub, 'placeText',
v_address.apils); --city
    u_json_pck.SetStringProperty(v_jsub, 'postalCode',
v_address.adrpastaindex);
    u_json_pck.SetStringProperty(v_jsub, 'fullText',
InternalFormFullAddress(v_address.aadr, v_address.apils,
v_address.adrpastaindex)); --fullAddress

    --valsts
    v_jsub2 := json();
    u_json_pck.SetIntProperty(v_jsub2, 'id', v_address.a_void_ir);
--countryId
    u_json_pck.SetStringProperty(v_jsub2,
'name',v_address.vanoslv); --country

    --ieliek valsts datus adreses json'ā
    v_jsub.put('country', v_jsub2);
    --ieliek adresi masīvā
    v_jlist.append(v_jsub.to_json_value);
end loop;

-- ievieto visus datus json'ā
v_json.put('address', v_jlist.to_json_value());

--atgriež datus
dbms_lob.createtemporary(po_data, true);
u_json_pck.writeToClob(pi_json => v_json, po_data => po_data);
end if;
end GetDataUsingRegCode;

```

**UT\_GETCOMPANY pakotnes procedūras *ut\_GetCompanyDataError* programmkods  
(vienībtesti)**

```

procedure ut_GetCompanyDataError is
    v_out          nclob;
    v_json         json;

    procedure DoTest(in_description varchar2, pi_regCode varchar2) is
        v_out nclob;
        v_json json;

    begin
        app_pck.Execute('1', 'URINTEGRATION.GetDataUsingRegCode', pi_regCode,
v_out);
        dbms_output.put_line(to_char(v_out));
        v_json := json(v_out);

        --Salīdzina iegūto vērtību ar vēlamo vērtību
        ut_assert.eq(in_description,
                    v_json.get('status').get_string(),
                    'error');

    exception
        when others then
            raise;
    end DoTest;

    begin

        --Nekorekts reģistrācijas numurs - gan cipari, gan burti, gan
pārsniedz simbolu skaita limitu
        dbms_output.put_line('ut_GetCompanyData - Test1');
        DoTest('Test1', 'ZzDatsZzDatsZzDatsZzDats');

        --Pēc reģistrācijas numura kontaktpersonu nevar atrast
        dbms_output.put_line('ut_GetCompanyData - Test2');
        DoTest('Test2', '40003278466');

        -- Nekorekts reģistrācijas numurs - 10 simboli
        dbms_output.put_line('ut_GetCompanyData - Test3');
        DoTest('Test3', '4000327846');

        -- Nekorekts reģistrācijas numurs - pārsniegts simbolu skaita limits
(11)
        dbms_output.put_line('ut_GetCompanyData - Test4');
        DoTest('Test4', '4000327846777');

    end ut_GetCompanyDataError;

```

COMPANY\_ES\_IMPORT\_PCK procedūras *UpdateAll* programmkods

```

procedure UpdateAll (pi_date in date default null) is

    v_update_count integer; --atjaunoto ierakstu skaitītājs
    v_flush_count  integer; --sūtāmo ierakstu skaitītājs (sūtīs pa 100)
    v_delete_count integer; --dzēsto ierakstu skaitītājs

    v_request      nclob;
    v_company_json nclob;

    v_errors       errorlist := errorlist();
    v_error        number;
    v_response     nclob;
    v_message      nclob;

begin
    v_update_count := 0;
    v_flush_count  := 0;
    v_delete_count := 0;

    dbms_lob.createtemporary(v_request, true);
    dbms_lob.createtemporary(v_company_json, true);

    -- dzēš neaktīvo uzņēmumu datus
    for c in (select ca.nmr_kods
              from in_nmaks_jp ca
              where ((pi_date is null) or (ca.nm_peddat > pi_date))
                    and ca.aktivs = 1) loop

        u_lob_pck.WriteAppendText(v_request,
            '{"delete":{"_index":"' || c_storage ||
                '","_type":"' || c_storage_type ||
                '","_id":"' || g_golbal_prefix || to_char(c.nmr_kods
            ) || '"}' || chr(10));
        --skaitītāji
        v_delete_count := v_delete_count + 1;
        v_flush_count  := v_flush_count + 1;

        if v_flush_count >= c_flush_count then
            v_flush_count := 0;
            begin
                elastic_pck.Send(pi_request => v_request,
                                po_response => v_response,
                                po_error    => v_error,
                                po_message  => v_message);

                --atbrīvo atmiņu
                dbms_lob.freetemporary(v_request);
                dbms_lob.createtemporary(v_request, true);
            exception
            when others then
                u_error_pck.AddErrorToList(v_errors, null,
                    'err:int:company:sync');
                --atbrīvo atmiņu
                dbms_lob.freetemporary(v_request);
                dbms_lob.createtemporary(v_request, true);
            end;
        end if;
    end loop;
end;

```

```

end loop;

-- pārbauda, vai nav palicis vēl kāds ieraksts
if dbms_lob.getlength(v_request) > 0 then
    elastic_pck.Send(pi_request => v_request,
                    po_response => v_response,
                    po_error    => v_error,
                    po_message  => v_message);
    --atbrīvo atmiņu
    dbms_lob.freetemporary(v_request);
    dbms_lob.createtemporary(v_request, true);

    if v_error != 0 then
        u_error_pck.AddErrorToList(v_errors,
                                   null,
                                   'err:int:company:sync');
    end if;
end if;

-- uzņēmumu datu papildināšana
for c in (select ca.nmr_kods
          from in_nmaks_jp ca
          where ((pi_date is null) or (ca.nm_peddat > pi_date))
                and ca.aktivs = -1 )loop

    i_ur_integration_pck.GetDataUsingRegCode(c.nmr_kods,
                                             v_company_json, v_errors);

    u_lob_pck.WriteAppendText(v_request,
                              '{"index":{"_index":"" || c_storage
                                         || '"',"_type":"" || c_storage_type
                                         || '"',"_id":"" || g_golbal_prefix ||
to_char(c.nmr_kods) || '"}}' || chr(10));

    u_lob_pck.WriteAppendNClob(v_request, v_company_json);
    --atbrīvo atmiņu
    dbms_lob.freetemporary(v_company_json);

    u_lob_pck.WriteAppendText(v_request, chr(10));

    --skaitītāji
    v_update_count := v_update_count + 1;
    v_flush_count  := v_flush_count + 1;
    --sūta atjaunos datus uz ElasticSearch
    if v_flush_count >= c_flush_count then
        v_flush_count := 0;
        begin
            elastic_pck.Send(pi_request => v_request,
                            po_response => v_response,
                            po_error    => v_error,
                            po_message  => v_message);
            --atbrīvo atmiņu
            dbms_lob.freetemporary(v_request);
            dbms_lob.createtemporary(v_request, true);
        exception
            when others then
                u_error_pck.AddErrorToList(v_errors, null,
                                           'err:int:company:sync');
            --atbrīvo atmiņu
            dbms_lob.freetemporary(v_request);

```

```

        dbms_lob.createtemporary(v_request, true);
    end;
end if;
end loop;

dbms_output.put_line(to_char(v_update_count)||' uzņēmumu datu
atjauninājumi nosūtīti uz Elasticsearch.');
```

dbms\_output.put\_line(to\_char(v\_delete\_count)||' uzņēmumu dzēstie
dati nosūtīti uz Elasticsearch.');

```

-- pārbauda, vai nav palicis vēl kāds ieraksts
if dbms_lob.getlength(v_request) > 0 then
    elastic_pck.Send(pi_request => v_request,
                    po_response => v_response,
                    po_error    => v_error,
                    po_message => v_message);

    --atbrīvo atmiņu
    dbms_lob.freetemporary(v_request);

    if v_error != 0 then
        u_error_pck.AddErrorToList(v_errors,
                                    null,
                                    'err:int:company:sync');

    end if;
end if;
end UpdateAll;
```

**I\_UR\_INTEGRATION\_PCK** pakotnes *GetDataUsingRegCode* procedūras JSON objekta struktūra

```
{
  "name": "Sabiedr\u012Bba ar ierobe\u017Eotu atbild\u012Bbu \"ZZ Dats\"",
  "regcode": 40003278467,
  "address": {
    "id": 1300313,
    "text": "Elizabetes iela 41/43",
    "placeText": "R\u012Bga",
    "postalCode": "LV-1010",
    "fullText": "Elizabetes iela 41/43, R\u012Bga, LV-1010",
    "country": {
      "id": 428,
      "name": "LATVIJA"
    }
  }
}
```

*AngularJS kontrolieris search.select.company.search.controller.js*

```

(function () {
  'use strict';

  function searchSelectCompanySearchController(
    $scope,
    $rootScope,
    $timeout,
    urlService,
    uiService,
    $log,
    notificationService,
    dialogService,
    personService,
    scopeService,
    employeeService,
    contactPersonService
  ) {

    $scope.showMainTemplate = true;
    $scope.minSearchLength = 3; //minimālais simbolu skaits - pēc kā meklēs
    $scope.searchValue = '';
    $scope.searchValueToService = '';
    $scope.searched = false; //vai vismaz 1x ir lietotājs izsaucis meklēšanu

    //datasource ir serviss
    $scope.dataSource = new kendo.data.DataSource({
      transport: {
        read: {
          url: urlService.baseUrl + "/Person/GetCompanies",
          type: "POST",
          dataType: "json",
          data: function () {
            return {
              quickSearch: $scope.searchValueToService,
              minSearchLength: $scope.minSearchLength
            };
          }
        }
      },
      schema: {
        data: 'data',
        total: 'totalItems'
      },
      pageSize: 5,
      serverPaging: true,
      serverFiltering: true,
      serverSorting: true
    });
    $scope.companyGridOptions = { //tabulas iestatījumi
      dataSource: $scope.dataSource,
      sortable: true,
      groupable: false,
      reorderable: false,
      columnMenu: false,
      pageable: {
        refresh: true,
        input: true,

```

```

        buttonCount: 5
    },
    columns: [//kolonnu definēšana
        { //pievienot poga
            template: "<button ng-click=\"select(dataItem)\" class=\"btn btn-
primary action-button-right\"><span class=\"far fa-save\"></span>&nbsp;
Pievienot</button>",
            width: "70px",
            sortable: false
        },
        { //nosaukums
            internalName: "Name",
            field: "name",
            sortable: false,
            title: "Nosaukums",
            width: "150px"
        },
        { //reģistrācijas numrus
            internalName: "RegCode",
            field: "regcode",
            title: "Reģ. nr.",
            sortable: false,
            width: "100px"
        },
        { //adrese
            internalName: "Address",
            //field: "address",
            title: "Adrese",
            template: "{{getCompanyAddress(dataItem)}}",
            sortable: false,
            width: "150px"
        }
    ]
};

$scope.getCompanyAddress = function (dataItem) {
    /// <summary>
    ///Funkcija, kas izgūst adresi no dataItem
    /// </summary>
    /// <param name="dataItem" type="type"></param>

    var address = dataItem.address;
    if (address != null && address.length > 0) {
        return address[0].fullText; //ņem pirmo adresi
    }
    return '';
};

$scope.select = function (dataItem) {
    /// <summary>
    /// Funkcija tiek izsaukta no grid ieraksta pogas "Pievienot"
    /// </summary>
    /// <param name="dataItem" type="type"></param>

    if (dataItem != null) {

        $scope.options.onSelect(dataItem); //padod uzņēmuma datus

        $scope.closeDialog(); //aizver dialoga logu
    }
};

```

```

$scope.filterKeypress = function (event) {
    /// <summary>
    /// definējam funkciju kas tiks izsaukta rediģējot brīvā tekstu filtru
    /// </summary>
    if (event.which === 13) {
        $scope.search();
    }
};

$scope.checkSearch = function () {
    /// <summary>
    /// funkcija
    /// </summary>
    if ($scope.searchValue.length < $scope.minSearchLength) {
        return false;
    }
    return true;
};

$scope.search = function () {
    /// <summary>
    /// funkcija, kas nodrošina meklēšanu
    /// </summary>
    if ($scope.checkSearch()) {
        $scope.searchValueToService = $scope.searchValue;

        $scope.dataSource.read();
        $scope.dataSource.page(1); //uzliek, ka aktīva ir pirmā lapa, ja
iepriekš ir navigēts
        $scope.searched = true;
        return;
    }
};
}

angular.module('Zzdats.Lietvaris').controller('searchSelectCompanySearchController',
searchSelectCompanySearchController);
})();

```

Kvalifikācijas darbs „*Uzņēmuma reģistra datu integrācija ar lietvedības sistēmu LIETVARIS*” 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: *Laura Tīrumniece* \_\_\_\_\_ .05.2018.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *Ilmārs Urbāns* \_\_\_\_\_ .05.2018.

Recenzents: *M.dat. Inguna Pede*

Darbs iesniegts 28.05.2018.

Kvalifikācijas darbu pārbaudījumu komisijas sekretāre: *Darja Solodovņikova* \_\_\_\_\_

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

\_\_\_\_.06.2018. prot. Nr. \_\_\_\_\_

Komisijas sekretārs(-e): \_\_\_\_\_