

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**TIEŠSAISTES APDROŠINĀŠANAS PLATFORMAS
OPTIMIZĀCIJA**

KVALIFIKĀCIJAS DARBS

Autors: Monta Kasparsone

Studenta apliecības Nr.: mk12151

Darba vadītājs: M. dat. Valdis Vizulis

RĪGA 2018

ANOTĀCIJA

Vispārīgā datu aizsardzības regula (VDAR) ir paredzēta datu aizsardzībai un privātuma nodrošināšanai visiem pilsoņiem, kas dzīvo Eiropas Savienības valstīs. Viena no šīs regulas prasībām ir datu subjektu pieprasījums (DSP), kas paredz kādas organizācijas pakalpojumu lietotājam pieprasīt par sevi vāktos datus, kas glabājas organizācijā.

Arī tiešsaistes apdrošināšanas platformai *Bamboo*, kas tiek izstrādāta uzņēmumā, ir jānodrošina tās klientiem iegūt par sevi uzglabātos datus. Pašlaik tas tiek nodrošināts manuāli, ejot cauri visiem dokumentiem un lejupielādējot tos pa vienam. Tādēļ šī darba mērķis ir optimizēt platformu un atvieglot darbu zvanu centra darbiniekiem, kas pārvalda klientu datus *eCare* sistēmā.

Lai sasniegtu mērķi, tiek automātiski atlasīti visi klienta dati, kas tiek konvertēti uz pārskatāmāku formātu un eksportēti. Katra izdevusies un neizdevusies eksportēšana tiek reģistrēta un atbildīgā persona tiek informēta par DSP veikšanu.

Atslēgvārdi: spējā izstrāde, datu subjektu pieprasījums, vispārīgo datu aizsardzības regula; datu atlase, datu konvertēšana, C#.

ABSTRACT

Optimization of online insurance platform

The General Data Protection Regulation (GDPR) is intended for data protection and privacy of all citizens living in European Union countries. One of the requirements of this regulation is Data Subject Access Request (DSAR), that provides the users of an organization services to request the data that have been collected about them within the organization.

The online insurance platform *Bamboo*, which is developed by the company, must also provide its customers the access to the stored data. Currently it is done manually by going through all the documents and downloading them one by one. Therefore, the aim of this work is to optimize the platform and to facilitate the work of call center staff who manage customer data in the *eCare* system.

To achieve this aim, all customer data is automatically selected, then transformed to a more transparent format and exported. Each successful and unsuccessful export is registered, and the responsible person is notified about the execution of the DSAR.

Keywords: agile software development, Data Subject Access Request, General Data Protection Regulation, selection of data, data conversion, C#.

SATURS

| | |
|---|----|
| Ievads..... | 7 |
| Vārdnīca..... | 8 |
| 1. Programmatūras prasību specifikācija..... | 10 |
| 1.1. Ievads..... | 10 |
| 1.1.1. Nolūks..... | 10 |
| 1.1.2. Darbības sfēra..... | 10 |
| 1.1.3. Saistība ar citiem dokumentiem | 10 |
| 1.1.4. Pārskats..... | 10 |
| 1.2. Vispārējais apraksts | 10 |
| 1.2.1. Sistēmas lietotāji..... | 10 |
| 1.2.2. Lietotāja raksturiezīmes..... | 11 |
| 1.2.3. Produkta funkcijas | 11 |
| 1.2.4. Vispārējie ierobežojumi..... | 12 |
| 1.3. Funkcionālās prasības | 12 |
| 1.3.1. Lietotāju stāsti | 12 |
| 1.3.1.1. #82981 – <i>DSAR</i> administratora lomas izveide <i>eCare</i> lietotājiem..... | 12 |
| 1.3.1.2. #81449 – Kotējumu un polises dokumentu eksportēšana | 12 |
| 1.3.1.3. #81455 – <i>eDoc</i> e-pastu eksportēšana | 13 |
| 1.3.1.4. #85242 – <i>CheetahMail</i> (ģenerēto) e-pastu eksportēšana | 14 |
| 1.3.1.5. #81457 – Klientu un polišu komentāru eksportēšana | 14 |
| 1.3.1.6. #86764 – Telefona zvanu komentāru eksportēšana | 15 |
| 1.3.1.7. #81454 – <i>eGain</i> e-pastu eksportēšana..... | 15 |
| 1.3.1.8. #85800 – <i>eGain</i> testa datubāzes izveidošana | 15 |
| 1.3.1.9. #83007 – Automātiska <i>Redmine</i> uzdevuma izveidošana un paziņojuma nosūtīšana | 16 |
| 1.3.1.10. #86635 – Asinhrona <i>Redmine</i> uzdevuma izveide un e-pasta nosūtīšana.... | 17 |
| 1.3.1.11. #82681 – Eksportēšanas funkcijas izmantošanas reģistrēšana | 17 |

| | | |
|-----------|--|----|
| 1.3.1.12. | #82983 – Kļūdu novērošana | 17 |
| 1.4. | Nefunkcionālās prasības | 18 |
| 1.4.1. | Drošība | 18 |
| 1.4.2. | Privātums | 18 |
| 2. | Programmatūras projektējuma apraksts | 19 |
| 2.1. | Ievads | 19 |
| 2.1.1. | Nolūks..... | 19 |
| 2.1.2. | Darbības sfēra | 19 |
| 2.1.3. | Saistība ar citiem dokumentiem | 19 |
| 2.1.4. | Pārskats..... | 19 |
| 2.2. | Datu plūsmu diagramma | 20 |
| 2.2.1. | DPD 0. līmenis – <i>Bamboo</i> sistēma | 21 |
| 2.2.2. | DPD 1. līmenis – <i>Bamboo</i> sistēmas moduļi | 22 |
| 2.2.3. | DPD 2. līmenis – <i>eCare</i> modulis..... | 23 |
| 2.3. | Modeļu diagrammas | 24 |
| 2.3.1. | Aktivitāšu diagramma | 24 |
| 2.3.2. | Datu iegūšanas un apstrādes shēma..... | 26 |
| 2.4. | Datu apstrāde | 27 |
| 2.4.1. | Datu konvertēšana uz <i>HTML</i> | 27 |
| 2.4.2. | Datu eksportēšana uz <i>excel</i> | 28 |
| 2.5. | Datubāzes projektējums | 28 |
| 2.6. | Moduļu projektējums | 30 |
| 2.6.1. | <i>DSAR export</i> apdarinātājs..... | 30 |
| 2.7. | Trasējamības matrica | 30 |
| 3. | Testēšanas dokumentācija | 33 |
| 3.1. | Ievads | 33 |
| 3.1.1. | Nolūks..... | 33 |
| 3.1.2. | Saistība ar citiem dokumentiem | 33 |

| | | |
|-----------|---|----|
| 3.1.3. | Testēšanas apraksts..... | 33 |
| 3.1.4. | Testpiemēru specifikācija..... | 33 |
| 3.1.5. | Testēšanas žurnāls | 34 |
| 3.1.5.1. | Polises un kotējumu dokumentu eksportēšanas testi – #81449..... | 34 |
| 3.1.5.2. | Dokumentu konvertēšanas uz <i>HTML</i> testi – #81455, #85242 | 36 |
| 3.1.5.3. | Klientu un polišu komentāru testi – #81457 | 36 |
| 3.1.5.4. | <i>eGain</i> e-pastu test – #81454, #85800 | 37 |
| 3.1.5.5. | Telefona zvanu komentāru testi – #86764 | 38 |
| 3.1.5.6. | Eksporta izmantošanas reģistrēšanas testi – #82681 | 39 |
| 3.1.5.7. | <i>DSAR admin</i> lomas testēšana (manuālā testēšana)– #82981 | 39 |
| 3.1.5.8. | Asinhronās funkcionalitātes testēšana (manuālā testēšana) - #86635..... | 40 |
| 3.1.5.9. | <i>Redmine</i> uzdevuma un paziņojuma automātiskas izveides testēšana (manuālā testēšana)– #83007..... | 40 |
| 3.1.5.10. | <i>Redmine</i> uzdevuma izveide kļūdu gadījumā - testēšana (manuālā testēšana)– #82983 | 41 |
| 3.1.5.11. | Nefunkcionālo prasību testēšana (manuālā testēšana)..... | 41 |
| 4. | Projekta organizācija | 43 |
| 5. | Konfigurāciju pārvaldība..... | 44 |
| 6. | Kvalitātes nodrošināšana | 45 |
| 7. | Darbietilpība | 46 |
| | Secinājumi | 47 |
| | Izmantotā literatūra..... | 48 |
| | Pielikumi..... | 49 |

IEVADS

Tiešsaistes apdrošināšanas platforma *Bamboo*, kas nodrošina polišu tirdzniecību, sastāv no vairākām komponentēm, kas nodrošina tās efektīvu darbību. Viena to tām ir *eCare* – klientu apkalpošanas modulis, kas ļauj tā lietotājiem skatīt visu klientu datus, kas radušies tiem veicot kādas darbības, kas saistītas ar apdrošināšanas platformu, kā arī rediģēt to kontu iestatījumus.

Saistībā ar jauno Eiropas Savienības Vispārīgo datu aizsardzības regulu (VDAR) tās ieviešana ir nepieciešama arī *Bamboo* platformā. Tas nozīmē, ka klients kurš vēlas iegūt savus datus, kas ir laika gaitā saglabājušies tam mijiedarbojoties ar sistēmu, var veikt datu subjektu pieprasījumu (DSP), kam ir jānodrošina iespēja klientam iegūt visus par sevi glabātos datus sistēmā. Tādēļ kvalifikācijas darba ietvaros ir jāveic platformas optimizācija, lai *eCare* lietotājiem (zvanu centra darbiniekiem) nebūtu, pēc klienta pieprasījuma, jālejupielādē katrs dokuments atsevišķi (kā tas tiek realizēts pašreiz), bet tos varētu automātiski savākt un eksportēt *zip* datnē.

Darba mērķis ir DSP izstrāde, dodot iespēju *eCare* sistēmas lietotājiem automātiski eksportēt visus datus, kas saistīti ar konkrētu klientu un tā mijiedarbību ar platformas izmantošanu. Šī mērķa realizēšanai tika ieviesta šāda funkcionalitāte:

- klienta datu atlasīšana – kotējumu, polišu un personas dokumenti, e-pasti, īsziņas un komentāri;
- *XML* dokumentu konvertēšana uz *HTML*;
- iegūto dokumentu apkopošana *zip* datnē un to lejupielāde;
- dokumentu eksportēšanas notikuma reģistrēšana un paziņojuma nosūtīšana;
- kļūdainas un nepilnīgas izpildes reģistrēšana un paziņojuma nosūtīšana;

Kvalifikācijas darbs sastāv no šādām nodaļām:

- Programmatūras prasību specifikācija – izstrādātā produkta un prasību apraksts;
- programmatūras projektējuma apraksts – sistēmas attēlojums;
- testēšanas dokumentācija – veikto testu apraksti un rezultāti;
- projekta organizācija – izstrādes process un izmantotie rīki;
- konfigurāciju pārvaldība – darba izstrādē izmantotie versiju pārvaldību un konfigurāciju rīki;
- kvalitātes nodrošināšana – sistēmas kvalitātes nodrošināšanai izmantotās procedūras;
- darbietilpības novērtējums – programmatūras izstrādes reālais un paredzētais darbietilpības novērtējums.

VĀRDNĪCA

1. tabula

Vārdnīcas termini un skaidrojumi

| Termins | Skaidrojums |
|--|---|
| <i>ASP.NET</i> | Tehnoloģiju kopa .NET satvarā, kas veido tīmekļa lietotnes un XML tīmekļa pakalpojumus. |
| <i>Bamboo</i> | Tiešsaistes apdrošināšanas platforma. |
| <i>C#</i> | Daudz paradigmu programmēšanas valoda. |
| <i>CheetahMail</i> | E-pasta pakalpojuma mārketinga risinājumu sniedzējs. [2] |
| Datu subjekta pieprasījums (<i>Data Subject Access Request</i>) | Indivīda pieprasījums pēc datiem, kas tiek glabāti kādā organizācijā. [11] |
| Datu subjekts | Indivīds, par kuru ir ierakstīti personas dati. [10] |
| <i>eDoc</i> | Dokumentu apstrādāšanas un pārvaldības rīks <i>Bamboo</i> sistēmā. |
| <i>eGain</i> | Klientu un darbinieku e-pastu saziņas reģistrēšanas rīks. |
| <i>Excel</i> | Izklājlapu apstrādes programma. [7] |
| <i>HTTP</i> apdarinātājs (<i>HTTP handler</i>) | Process, kas darbojas, atbildot uz pieprasījumu, kas piegādāts ASP.NET tīmekļa lietojumprogrammai. [3] |
| <i>Jenkins</i> | Atvērtā pirmkoda automatizācijas serviss. [6] |
| Kotējums | piedāvājums, ko apdrošinātājs iesniedz klientam, norādot nosacījumos, ar kādiem tas piekrīt apdrošināšanai. Kotējums parasti ir spēkā noteiktu laiku. |
| <i>PDF</i> | Elektronisko dokumentu datņu formāts. |
| <i>Redmine</i> | Atvērtā pirmkoda tīmekļa darbu un problēmu pārvaldības rīks. [4] |
| <i>SonarQube</i> | Atvērtā pirmkoda platforma, kas izstrādāta, lai pastāvīgi pārbaudītu koda kvalitāti. [1] |
| <i>SQL</i> | Datubāzes vaicājumu un programmēšanas valoda. |
| <i>SVN</i> | Versiju kontroles rīks. |
| Vispārīgā datu aizsardzības regula (<i>General Data Protection Regulation</i>) | Regula, kas paredz personas datu aizsardzību un uzraudzību. |
| <i>XLSX</i> | Uz XML balstīta izklājlapas datne. |
| <i>XML (Extensible Markup Language)</i> | Paplašināmā iezīmēšanas valoda kas speciāli izstrādāta darbam ar tīmekļa dokumentiem. |

| | |
|-------------|--|
| <i>XSLT</i> | Valoda kas pārveido XML dokumentus – XML dokumentos vai cita tipa dokumentos (HTML, teksta, u.c.). |
| <i>Zip</i> | Arhīva datu formāts, kas atbalsta bezzudumu datu saspiešanu. |

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Programmatūras prasību specifikācijā (PPS) tiek aprakstītas funkcionālās prasības lietotāju stāstu formā un nefunkcionālās prasības. Dokuments ir paredzēts sekojošām lietotāju grupām: pasūtītāja, testētāja un izstrādātāju vajadzībām.

1.1.2. Darbības sfēra

Izstrādāto funkciju uzdevums ir optimizēt *Bamboo* platformu tā, lai *eCare* sistēmas lietotājiem (zvanu centra darbiniekiem) nav katru reizi, kad klients veic datu subjekta pieprasījumu, jālejupielādē visi tā dati manuāli (pa vienam), bet izstrādātā funkcionalitāte to dara automatizēti.

1.1.3. Saistība ar citiem dokumentiem

Dokumentā ietvertās prasības tiek projektētas programmatūras projektējuma aprakstā. Funkcionālo un nefunkcionālo prasību testēšana tiek aprakstīta testēšanas dokumentācijā.

1.1.4. Pārskats

Dokuments ietver šādas četras nodaļas:

Pirmā nodaļa – Sniedz informāciju par nolūku, darbības sfēru un saistību ar citiem dokumentiem.

Otrā nodaļa – vispārējais apraksts, kurā aprakstītas produkta perspektīvas, funkcijas, lietotāja raksturiezīmes, vispārējie ierobežojumi un pieņēmumi un atkarības.

Trešā nodaļa – funkcionālās prasības, iekļauj lietotāju stāstus un sniedz pārskatu par sistēmas funkcionalitāti.

Ceturta nodaļa – informācija par sistēmā iekļautajām nefunkcionālajām prasībām.

1.2. Vispārējais apraksts

1.2.1. Sistēmas lietotāji

Sistēmā ir pieci lietotāju veidi:

- *eCare* sistēmas zvanu centra darbinieks ar piešķirtu lomu *DSAR admin* – lietotājs, kurš ir apdrošināšanas kompānijas darbinieks, kurš sazinās ar klientiem un veic

datu subjektu pieprasījumu eksportējot visus sistēmā saglabātos klienta datus un nosūta tos viņam.

- Biznesa procesu vadītājs – konkrēts lietotājs, kurš pārbauda funkcionalitātes izmantošanu, saņemot paziņojumus katru reizi, kad tiek izmantota izstrādātā funkcionalitāte.
- Atbalsta darbinieks – lietotājs, kuram pienāk paziņojums par kļūdu nesekmīgas datu eksportēšanas laikā un attiecīgi jārīkojas, lai to novērstu.
- Tiesību administrators – lietotājs, kurš piešķir zvanu centru darbiniekiem *DSAR admin* lomu.
- Klients – apdrošināšanas platformas lietotājs, kura dati tiek glabāti sistēmā un kurš veic datu subjekta pieprasījumu, sazinoties ar *eCare* sistēmas zvanu centra darbinieku.

1.2.2. Lietotāja raksturiezīmes

Izmantot dokumentu eksportēšanas funkciju var tikai *eCare* darbinieks, kuram ir piešķirta piekļuve. Lietotājam ir jābūt pamat zināšanām darbā ar datoru un interneta pārlūku. Noder arī zināšanas par *eCare* darbības principiem.

1.2.3. Produkta funkcijas

Produkta funkcijas ir sadalītas pa lietotāju grupām.

eCare sistēmas zvanu centra darbinieka funkcijas:

- lejupielādēt *zip* dokumentu, kurš satur visus klienta datus;
- apskatīt eksportēšanas izmantošanas vēsturi.

Biznesa procesu vadītāja funkcijas:

- saņemt ziņu par funkcionalitātes izmantošanu;
- saņemt *Redmine* uzdevumu;
- skatīt eksportēšanas vēsturi.

Atbalsta darbinieka funkcijas:

- saņemt kļūdu paziņojumu par neizdevušos eksportēšanu.

Tiesību administratora funkcijas:

- piešķirt lietotājam lomu *DSAR admin*

Klienta funkcijas:

- veikt dokumentu eksportēšanas pieprasījumu.

1.2.4. Vispārējie ierobežojumi

Funkcionalitātes izmantošanai nepieciešams dators ar piekļuvi internetam un pārlūkprogramma, kas atbalsta *JavaScript* izpildi. Lai varētu izmantot eksportēšanas funkciju ir nepieciešama pieeja *eCare* sistēmai un jābūt piešķirtai *DSAR admin* lomai. Dokumenti tiek apkopoti *zip* mapē, ja ir nepieciešamība pēc datu apskatīšanas vai atarhivēšanas, tad vajag programmu - arhivatoru, kas māk eksportēt un atvērt šādu datni.

1.3. Funkcionālās prasības

1.3.1. Lietotāju stāsti

Nodaļā tiek aprakstītas programmatūras funkcionālās prasības pēc spējas izstrādes metodes lietotāju stāstu formā, kas tiek glabāti *Redmine* – projektu pārvaldības rīkā. Lietotāju stāstu identifikatori sakrīt ar *Redmine* piešķirtajiem identifikatoriem un tiek norādīti lietotāju stāstu virsrakstos.

1.3.1.1. #82981 – *DSAR* administratora lomas izveide *eCare* lietotājiem

Es kā tiesību administrators vēlos izveidot *DSAR* administratora lomu, lai to varētu piešķirt tikai tiem *eCare* darbiniekiem, kas drīkst eksportēt klientu datus.

Akceptēšanas kritēriji

- Jāizveido jauna lietotāja loma ar nosaukumu *DSAR admin*.
- Šīs lomas vienīgā pieejamā funkcija ir *DSAR* eksportēšanas iespējas.

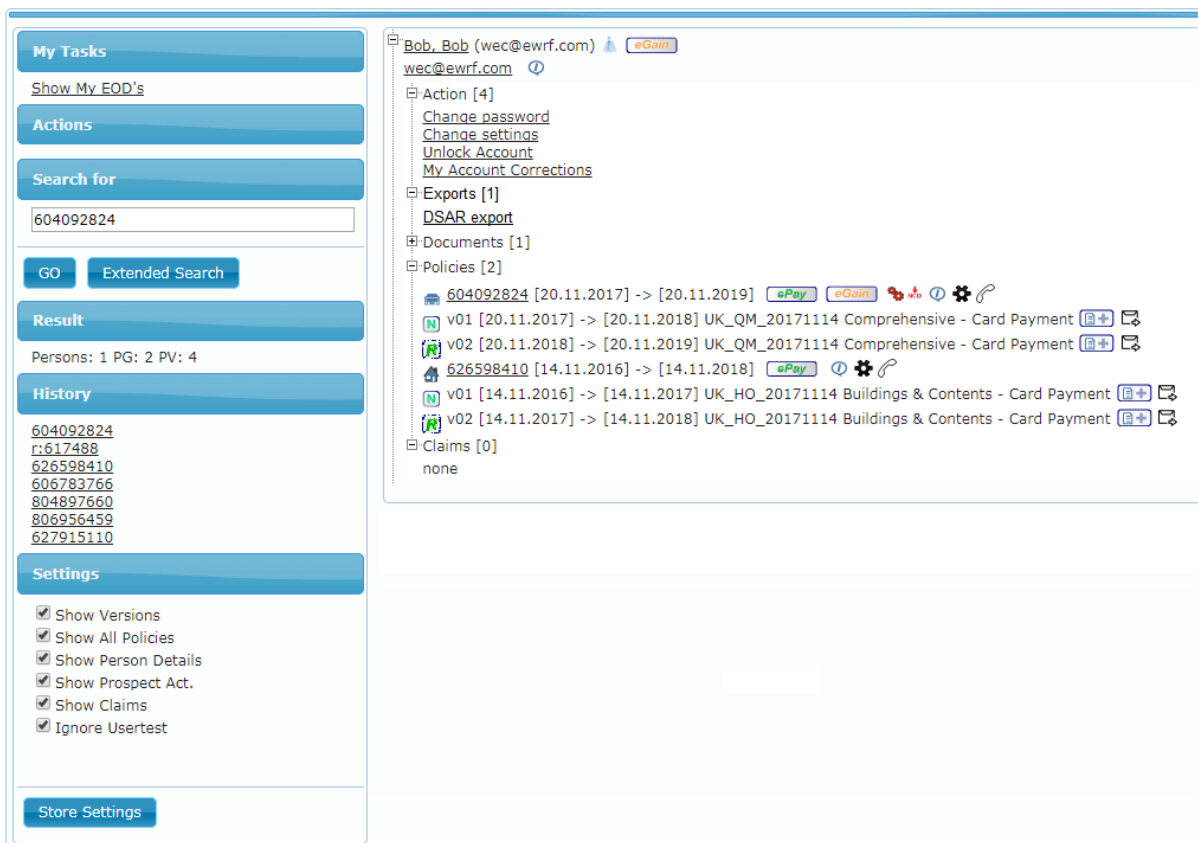
1.3.1.2. #81449 – Kotējumu un polises dokumentu eksportēšana

Es kā *eCare* lietotājs vēlos eksportēt visus klienta kotējumu un polišu dokumentus, lai nevajadzētu tos lejupielādēt manuāli.

Akceptēšanas kritēriji

- *DSAR export* ir jānodrošina automātiska funkcija, kas ļauj eksportēt klienta datus.
- Jāpievieno *eCare* sistēmā sadaļa, kas iekļauj darbību (pogu), kas lejupielādē *zip* datni, kura satur visus klienta kotējumu un polišu dokumentus visās to versijās un visās dokumentu versijās (ja dokuments ir bijis atjaunots). Attēlā 1.1. attēlota *DSAR export* darbības poga.
- Datnes nosaukumu sastāda polises vai kotējuma numurs, versija uz ko tas attiecas, dokumenta tips un tā versija.

- Funkcijai ir jābūt pieejamai tikai ar lietotāja lomu – *DSAR admin*.



1.1.att. DSAR export poga un eCare sistēmas attēlojums

1.3.1.3. #81455 – eDoc e-pastu eksportēšana

Es kā eCare lietotājs vēlos eksportēt visus eDoc e-pastus kas tiek automātiski ģenerēti klientam, tā polisēm un kotējumiem, lai nevajadzētu tos atjaunināt un lejupielādēt manuāli.

Akceptēšanas kritēriji

- Eksportētajai zip datnei ir jāsaturs klienta, kotējumu un polises e-pastu dokumentu katra versija, katrā klientu polises versijā.
- Visi e-pasti un īsziņas ir jāpārveido uz *HTML* datnes formātu.
- E-pastu datnes nosaukums sastāv no polises numura un versijas uz ko tā attiecas, dokumenta tipa, datu tipa un versijas.

Piemērs:

"604092824 v2 NCD reminder 0 email v2"

Formāts, ja dokuments ir pievienots polisei:

"(polises numurs) v(polises versija) (dokumenta nosaukums) (dokumenta tips) (dokumenta versija)"

Formāts, ja dokuments ir pievienots kotējumam:

"(kotējuma numurs) (dokumenta nosaukums) (dokumenta tips) (dokumenta versija)"

1.3.1.4. #85242 – *CheetahMail* (ģenerēto) e-pastu eksportēšana

Es kā *eCare* lietotājs vēlos eksportēt visus *CheetahMail* ģenerētos e-pastus kas tiek automātiski ģenerēti klientu polisēm, lai nevajadzētu tos atjaunināt un lejupielādēt manuāli.

Akceptēšanas kritēriji

- Eksportētajai *zip* datnei ir jāsaturs polises *CheetahMail* e-pastu katra versija, katrā klienta polises versijā.
- Šie e-pasti ir jāpārveido uz *HTML* formātu un tajos drīkst atstāt tikai klientiem nepieciešamo informāciju.
- Katras datnes nosaukums sastāv no polises numura un versijas uz ko tā attiecas, dokumenta tipa, datu tipa un versijas.

Piemērs:

"604092824 v2 NCD reminder 0 email v2"

Formāts, ja dokuments ir pievienots polisei:

"(polises numurs) v(polises versija) (dokumenta nosaukums) (dokumenta tips) (dokumenta versija)"

Formāts, ja dokuments ir pievienots kotējumam:

"(kotējuma numurs) (dokumenta nosaukums) (dokumenta tips) (dokumenta versija)"

1.3.1.5. #81457 – Klientu un polišu komentāru eksportēšana

Es kā *eCare* lietotājs vēlos eksportēt visus klientu un polišu komentārus, lai nevajadzētu tos saglabāt manuāli.

Akceptēšanas kritēriji

- Eksportētajai *zip* datnei ir jāsaturs klientu un polišu visi komentāri.
- Klientu un katras polises komentāri ir jāeksportē atsevišķā *excel* datnē, kurā tiek iekļautas kolonas – saglabāšanas datums, tips, komentārs un izpildes termiņš.
- Katras datnes nosaukums sastāv no klienta vārda un uzvārda (ja komentārs ir tam piesaistīts) vai polises numura (katrai polisei savi komentāri) un datu tipa.

Piemērs:

"John Smith Comments" vai "603968704 Comments"

Formāts, ja komentārs ir pievienots klientam:

"(klienta vārds) (klienta uzvārds) (dokumenta tips)"

Formāts, ja komentārs ir pievienots polisei:

"(polises numurs) (dokumenta tips)"

1.3.1.6. #86764 – Telefona zvanu komentāru eksportēšana

Es kā *eCare* lietotājs vēlos eksportēt visus telefona zvanu komentārus, lai nevajadzētu tos saglabāt manuāli.

Akceptēšanas kritēriji

- Eksportētajai *zip* datnei ir jāsaturs visi telefonu zvanu komentāri, kā arī tie ir jāeksportē atsevišķā *excel* datnē, kurā tiek iekļautas kolonas – saglabāšanas datums, tips, iemesls, iznākums un komentārs.
- Katras datnes nosaukums sastāv no polises numura, kam ir piesaistīti telefonu zvanu komentāri un datu tipa.

Piemērs:

“603968704 Phone comments”

Formāts:

“(polises numurs) (dokumenta tips)”

1.3.1.7. #81454 – *eGain* e-pastu eksportēšana

Es kā *eCare* lietotājs vēlos eksportēt visus klienta *eGain* e-pastus, lai nevajadzētu tos saglabāt manuāli.

Akceptēšanas kritēriji

- Eksportētajai *zip* datnei ir jāsaturs visi *eGain* e-pasti, kas tiek piesaistīti dažādiem klientu e-pastiem, kuri jāeksportē atsevišķās *excel* datnēs.
- Datu kolonas sastāv no laukiem – biļetes identifikatora, biļetes statusa, izveides datuma, statusa, priekšmeta un satura.
- Datnes nosaukums sastāv no klienta e-pasta un datu tipa.

Piemērs:

“JohnSmith@mailinator.com eGain emails”

Formāts:

“(klienta e-pasts) (dokumenta tips)”

1.3.1.8. #85800 – *eGain* testa datubāzes izveidošana

Es kā izstrādātājs vēlos izveidot *eGain* testa datubāzi, lai varētu notestēt vai lietotāju stāsts ar identifikatoru – #81454 izpilda prasības.

Akceptēšanas kritēriji

- Jāizveido *Bamboo* sistēmā jauna datubāze.
- Jāizveido datubāzē *eCare* lietotājs un jāpiešķir tiesības uz visām izstrādes vidēm.

- Jāizveido tabula – *eGainTicket*, kas aizstāj *eGain* datubāzē esošo tabulu.
- Jāizveido iekļautā procedūra, kas atlasa ierakstus no *eGainTicket* pēc klienta e-pasta adreses.
- Tabulu kolonas sastāv no laukiem (iekavās norādīts lauka nosaukums datubāzē un datu tips):
 - biļetes identifikators (TID, skaitlits),
 - biļetes statuss (St, rakstu zīme),
 - izveides datums (emailDate, datums),
 - statuss (Flw, simbolu virkne),
 - priekšmets (subject, simbolu virkne),
 - saturs (Content, simbolu virkne),
 - izveidotājs (agent, simbolu virkne),
 - e-pasta identifikators (emailId, cipars),
 - piezīme (note, simbolu virkne),
 - klienta e-pasts (USER_EMAIL, simbolu virkne).

1.3.1.9. #83007 – Automātiska *Redmine* uzdevuma izveidošana un paziņojuma nosūtīšana

Es kā biznesa procesa vadītājs vēlos, lai notiek automātiska *Redmine* uzdevuma izveidošana un paziņojumu nosūtīšana, lai varētu pārliecināties, ka klienta dati netiek nesankcionēti eksportēti.

Akceptēšanas kritēriji

- Jāizveido automātiska *Redmine* uzdevuma izveidošana, kas jāpiešķir biznesa procesa vadītājam.
- Jāizveido automātiska e-pasta nosūtīšana biznesa procesa vadītājam.
- Katrā eksportēšanas reizē ir jāizveido jauns *Redmine* uzdevums, kurā tiek aizpildīti lauki – projekta nosaukums, uzdevuma veids, priekšmets, apraksts, prioritāte, versija, vecāka elements, sākšanas datums.
- E-pasts satur informāciju par to, ka ir notikusi dokumentu eksportēšana un saiti uz *Redmine* jaunizveidoto uzdevumu.

1.3.1.10. #86635 – Asinhrona *Redmine* uzdevuma izveide un e-pasta nosūtīšana

Es kā atbalsta pārstāvis vēlos, lai *Redmine* uzdevumu izveide un e-pastu nosūtīšana notiek asinhroni, lai samazinātu *DSAR export* funkcijas darbības laiku.

Akceptēšanas kritēriji

- Jāizstrādā *Redmine* rīkā jauna uzdevuma izveide un e-pasta nosūtīšana biznesa procesa vadītājam asinhroni.

1.3.1.11. #82681 – Eksportēšanas funkcijas izmantošanas reģistrēšana

Es kā biznesa procesu vadītājs vēlos, lai tiek reģistrēta eksportēšanas funkcijas izmantošana, lai varētu auditēt un nodrošināt piemērotu izmantošanu.

Akceptēšanas kritēriji

- Katru reizi kad tiek veikta dokumentu eksportēšana, *eCare* sistēmā notiek jauna komentāra izveide.
- Komentārs tiek piesaistīts klientam un tajā tiek saglabāta informācija par funkcijas izmantošanu:
 - saglabāšanas datums,
 - *eCare* lietotājs,
 - tips,
 - komentārs,
 - izpildes termiņš,
 - statuss.

1.3.1.12. #82983 – Kļūdu novērošana

Es kā atbalsta pārstāvis vēlos veikt kļūdu novērošanu, lai neviena sistēma netiktu nelabvēlīgi ietekmēta brīdī kad *DSAR export* funkcija tiek izmantota.

Akceptēšanas kritēriji

- Neizdevušās eksportēšanas gadījumā atbalsta komandai tiek izveidots jauns *Redmine* negadījuma uzdevums, kurā ir norādīts kļūdas paziņojums un nozīmīgums.

1.4. Nefunkcionālās prasības

1.4.1. Drošība

Tikai autorizējušies lietotāji un ar *DSAR admin* lomu var redzēt un izmantot *DSAR export* funkciju.

Visas eksportēšanas kļūdas vai brīdinājumi ir jāreģistrē datu bāzē.

Funkcijas izmantošana ir jāreģistrē datu bāzē un jāizveido *Redmine* uzdevums, jānosūta e-pasts biznesa procesa vadītājam, ja eksportēšana ir bijusi sekmīga, pretējā gadījumā atbalsta darbiniekam izveido jaunu *Redmine* negadījuma uzdevumu.

1.4.2. Privātums

Klientu dati nedrīkst būt pieejami trešajām pusēm.

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

Programmatūras projektējuma aprakstā, izstrādātais produkts tiek attēlots datu plūsmas diagrammās un datu modeļos. Tiek izstrādāta apdrošināšanas kompānijas klientu datu eksportēšanas funkcija, kas ir pieejama *eCare* sistēmā.

2.1.1. Nolūks

Programmatūras projektējuma apraksts (PPA) ir izstrādāts “Tiešsaistes apdrošināšanas platformas optimizācija” projektējuma aprakstīšanai atbilstoši lietotāju stāstos izvirzītajām prasībām. Dokuments ir paredzēts sistēmas izstrādātājiem, pasūtītājam un testētājam.

2.1.2. Darbības sfēra

Izstrādātais produkts ir domāts Tiešsaistes apdrošināšanas uzņēmuma zvanu centru darbiniekiem. Programmatūras produkta uzdevums ir nodrošināt iespēju *eCare* zvanu centra darbiniekiem iegūt visus konkrēta klienta datus atbilstoši VDAR prasībām.

2.1.3. Saistība ar citiem dokumentiem

Dokuments ir lietojams kopā ar programmatūras prasību specifikāciju. Dokuments tiek izmantos testēšanas dokumentācijas sastādīšanai.

2.1.4. Pārskats

Programmatūras projektējuma apraksts sastāv no 4 nodaļām:

Pirmā nodaļa – ievads, kas sniedz informāciju par nolūku, darbības sfēru un saistību ar citiem dokumentiem.

Otrā nodaļa – datu plūsmas diagrammas, kas grafiski attēlo sistēmas darbību.

Trešā nodaļa – modeļu diagrammas, grafisks sistēmas attēlojums no dažādiem skata punktiem.

Ceturtā nodaļa – datu apstrāde, kas sniedz informāciju par dokumentu pārveidošanu.

Piektā nodaļa - datubāzes projektējums, darba ietvaros izstrādātās datubāzes projektējums un apraksts.

Sestā nodaļa – (datu plūsmu diagrammu) moduļu projektējums .

2.2. Datu plūsmu diagramma






Datu plūsmu diagrammās (DPD) tiek attēlotas tās sistēmas daļas, kuras kvalifikācijas darba ietvaros tika mainītas vai izveidotas.

DPD plūsmas ir numurētas, lai uzlabotu uzskatāmību un to paskaidrojošie teksti atrodas zem katras diagrammas attēla.

DPD apzīmējumi ir sekojoši:

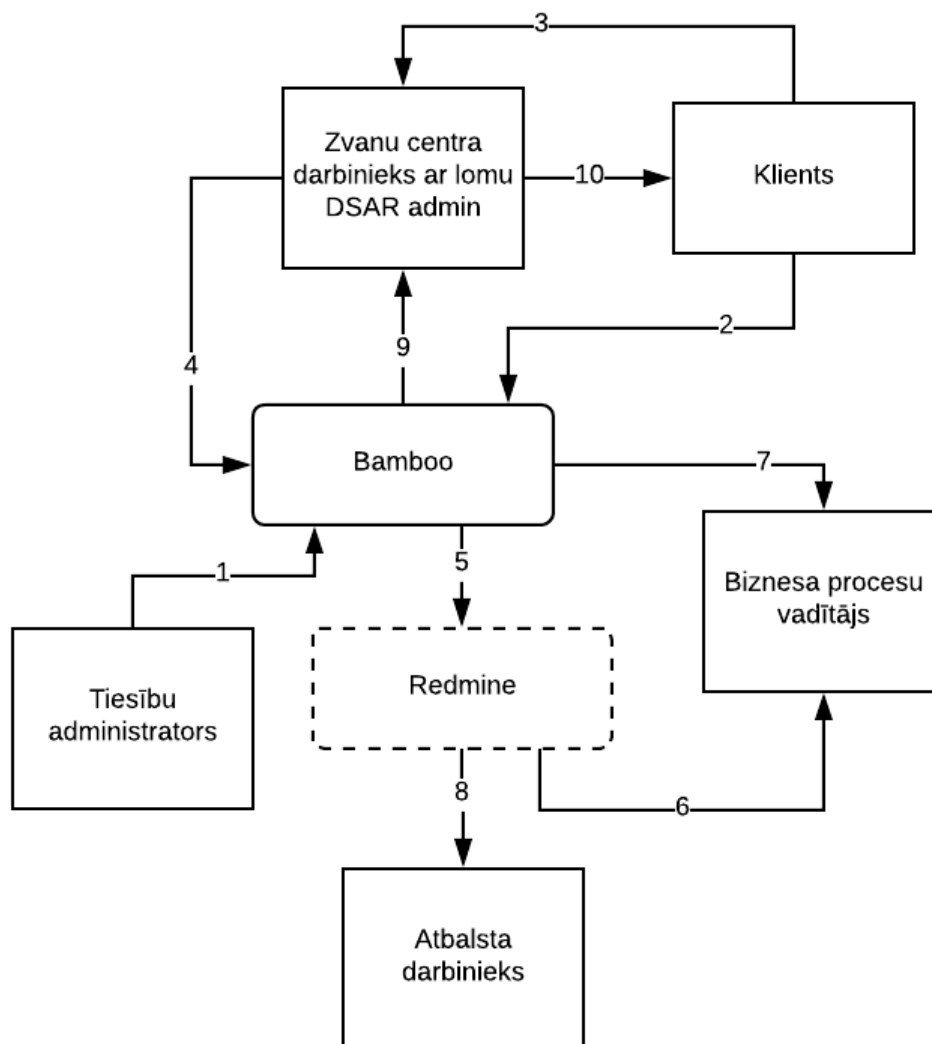
2.1. tabula

Apzīmējumi un paskaidrojumi

| Apzīmējumi | Paskaidrojumi |
|---|-----------------------|
|  | Datu plūsma |
|  | Lietotājs |
|  | Datu bāze |
|  | Prasību modulis |
|  | Ārējā lietotāju grupa |

2.2.1. DPD 0. līmenis – *Bamboo* sistēma

Attēlā 2.1. tiek attēlota sistēma un tās lietotāji.



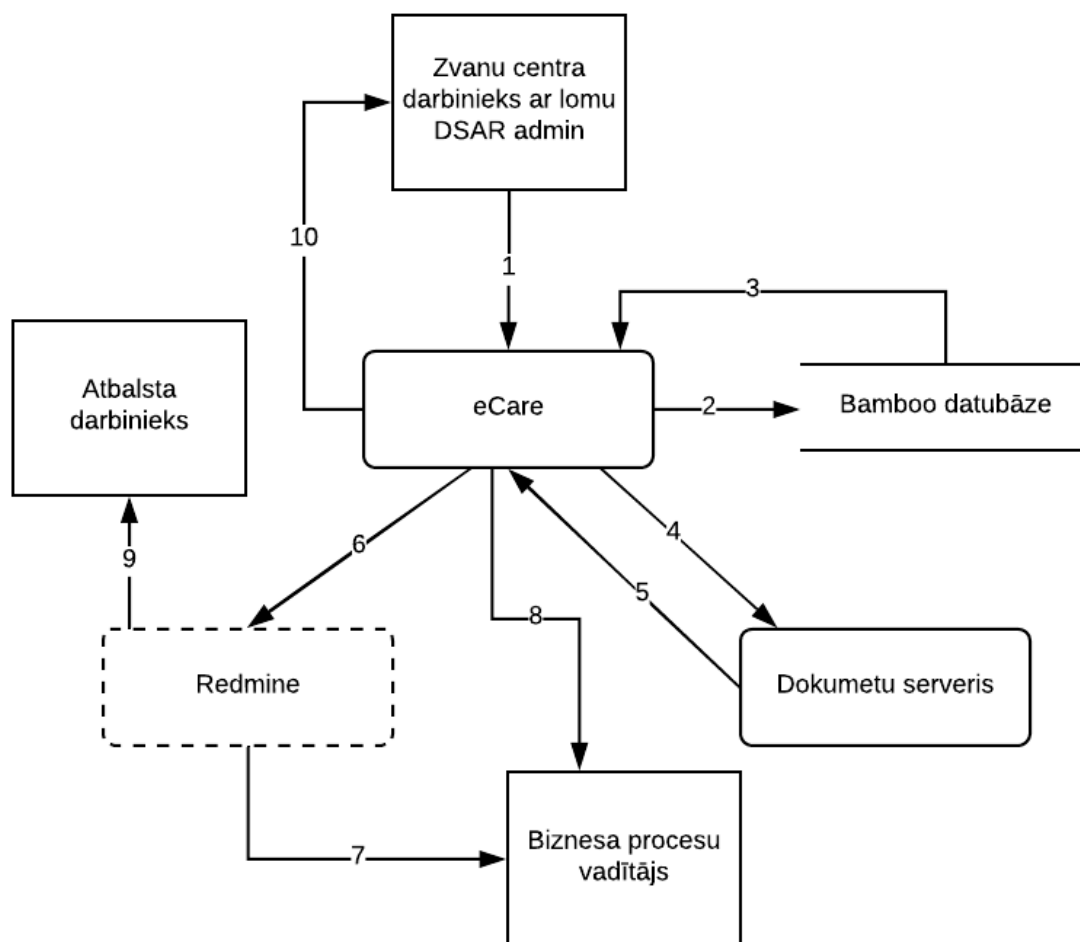
2.1.att. 0. līmeņa DPD

1. Tiesību administrators piešķir *eCare* lietotājam *DSAR admin* lomu.
2. Klients ievada datus *Bamboo* sistēmā.
3. Klients pieprasa visus par sevi sistēmā glabātos datus zvanu centra darbiniekam.
4. Zvanu centra darbinieks ar *DSAR admin* lomu, pēc klienta pasūtījuma veic datu pieprasījumu (eksportēšanu) – sistēmā ievada konkrētā klienta datus (sameklē klientu).
5. *Bamboo* sistēma izveido jaunu uzdevumu par to, ka ir veikta datu eksportēšana.
6. *Redmine* sistēma piešķir izveidoto uzdevumu biznesa procesa vadītājam.
7. *Bamboo* sistēma biznesa procesa vadītājam nosūta paziņojumu, par klienta datu eksportēšanas notikumu.

8. Kļūdainas eksportēšanas gadījumā tiek izveidots jauns negadījuma uzdevums un *Redmine* sistēma to piešķir atbalsta darbiniekam.
9. *Bamboo* sistēma atgriež visus klienta datus, zvanu centra darbiniekam.
10. Zvanu centra darbinieks datus piegādā sistēmas klientam.

2.2.2. DPD 1. līmenis – *Bamboo* sistēmas moduļi

Attēlā 2.2. tiek attēlots sistēmas sadalījums moduļos. Funkcionalitātes izstrādē tiek mainīts vienīgi *eCare* modulis, tādēļ tikai to arī attēlo. Pati sistēma sastāv no vairāk kā tikai viena moduļa.



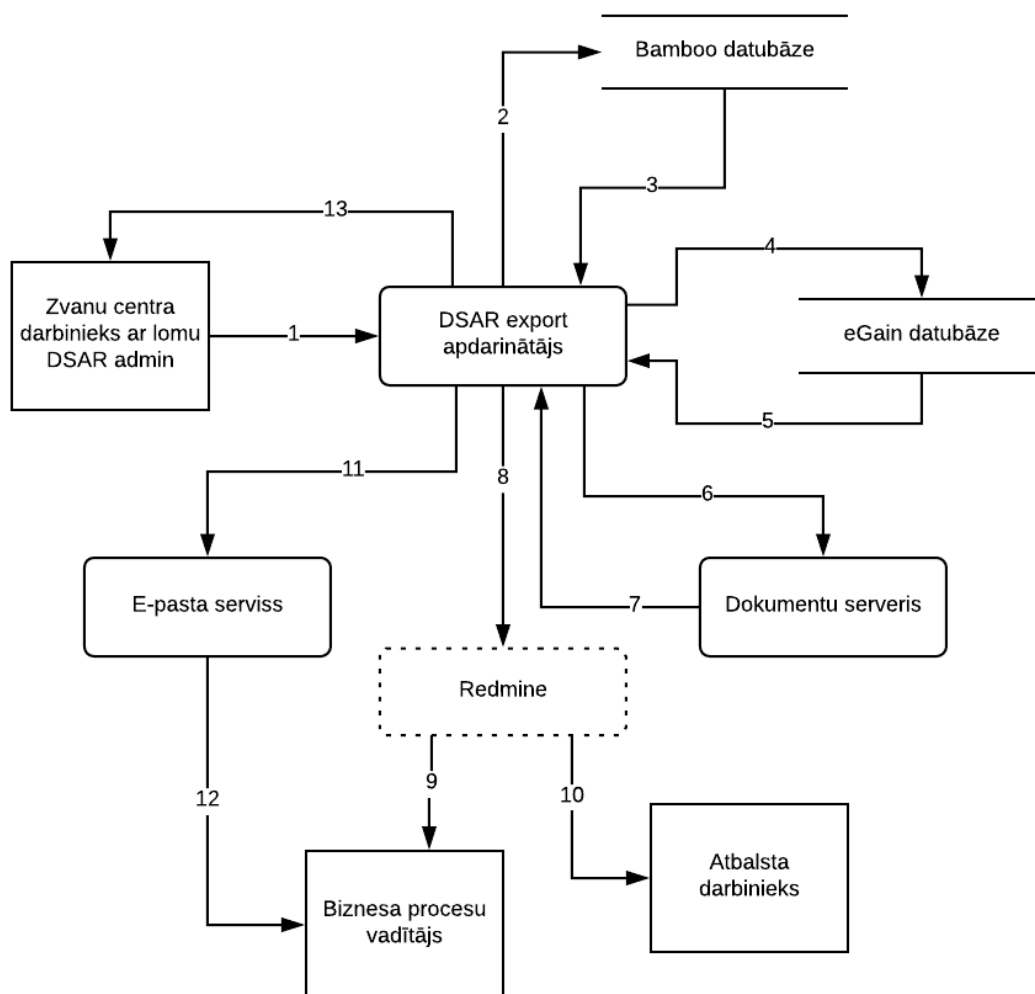
2.2.att. 1. līmeņa DPD

1. Zvanu centra darbinieks, sameklē konkrēto klientu *eCare* modulī, ievada klienta datus un veic to eksportēšanu.
2. *eCare* modulis pieprasa nepieciešamos klienta parametrus *Bamboo* datubāzei.
3. *Bamboo* datubāze atgriež visus nepieciešamos ar klientu saistītos datus.

4. *eCare* modulis dokumentu serverim pieprasa konkrētos dokumentus (no *Bamboo* datubāzes tika iegūta dokumentu atrašanās vietu serverī) – *eDoc* e-pastus, *CheetahMail* e-pastus, īsziņas, pārējos klienta dokumentus.
5. Dokumentu serveris atgriež atpakaļ pieprasītos dokumentus.
6. *eCare* modulis izveido jaunu *Redmine* uzdevumu.
7. *Redmine* piešķir jauno uzdevumu (kurā ir eksportēšanas atskaites dati) biznesa procesu vadītājam.
8. *eCare* piegādā paziņojumu par *Redmine* uzdevuma izveidi biznesa procesu vadītājam.
9. *Redmine* piešķir negadījuma uzdevumu (kurā ir dati par kļūdu) atbalsta darbiniekam.
10. *eCare* modulis sagatavo, noformē un atgriež visus klienta apkopotos datus, zvanu centra darbiniekam.

2.2.3. DPD 2. līmenis – *eCare* modulis

Attēlā 2.3. tiek attēlots *eCare* moduļa funkcija, kas ir – *DSAR export* apdarinātājs (*handler*).



2.3.att. 2. līmeņa DPD

1. Zvanu centra darbinieks ar lomu *DSAR admin* veic *DSAR export* pieprasījumu konkrētam klientam.
2. *DSAR export* apdarinātājs pieprasa klienta un saistīto dokumentu datus no *Bamboo* datubāzes, kā arī ievada datus par funkcijas izmantošanu datubāzes tabulā.
3. *Bamboo* datu bāze veic nepieciešamo datu atgriešanu.
4. *DSAR export* apdarinātājs pieprasa konkrētā klienta ierakstus no *eGain* datubāzes, meklējot pēc klienta e-pasta adreses.
5. *eGain* datubāze atgriež nepieciešamos datus.
6. *DSAR export* apdarinātājs pieprasa konkrētā klienta dokumentus (meklējot pēc dokumentu atrašanās vietas serverī) – *eDoc* e-pastus, *CheetahMail* e-pastus, īsziņas un pārējos dokumentus.
7. Serveris atgriež visus atrastos klienta dokumentus.
8. *DSAR export* apdarinātājs izveido jaunu *Redmine* uzdevumu (ja eksports ir noticis veiksmīgi vai ja neveiksmīgi).
9. *Redmine* sistēma veiksmīgas eksportēšanas gadījumā šo uzdevumu pievieno biznesa procesu vadītājam.
10. *Redmine* sistēma neveiksmīga eksporta gadījumā pievieno uzdevumu atbalsta darbiniekam.
11. *DSAR export* apdarinātājs izveido e-pastu par *Redmine* uzdevuma izveidi e-pasta servisam.
12. E-pasta serviss piegādā paziņojumu par *Redmine* uzdevuma izveidi biznesa procesu vadītājam.
13. *DSAR export* apdarinātājs sagatavo, noformē un piegādā dokumentus zvanu centra darbiniekam *zip* formātā.

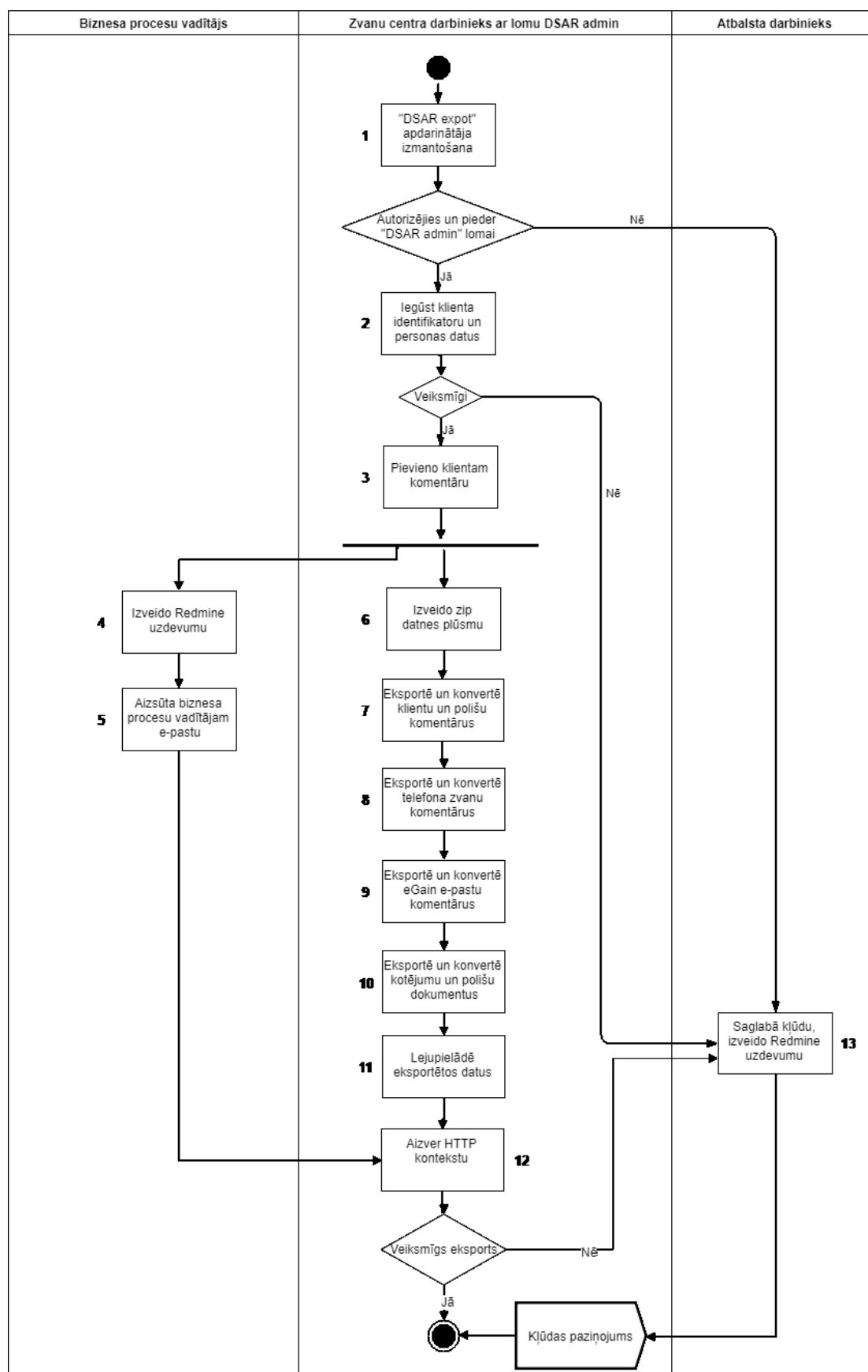
2.3. Modeļu diagrammas

Nodaļā tiek grafiski attēloti programmatūras darbības principi no dažādiem skata punktiem.

2.3.1. Aktivitāšu diagramma

Lai labāk uztvertu *DSAR export* apdarinātāja darbību, attēlā 2.4. ir attēlota *DSAR export* funkcijas aktivitāšu diagramma. Darbība sākas ar saites nospiešanu un beidzas ar kļūdu paziņojumiem vai eksportēto datu lejupielādi. Diagrammā ir parādīts, ka ja kāds no soļiem nav sanācis, ir bijušas kļūmes, tad tas tiek saglabāts kļūdu ziņojumā un lietotājam uz ekrāna parādās

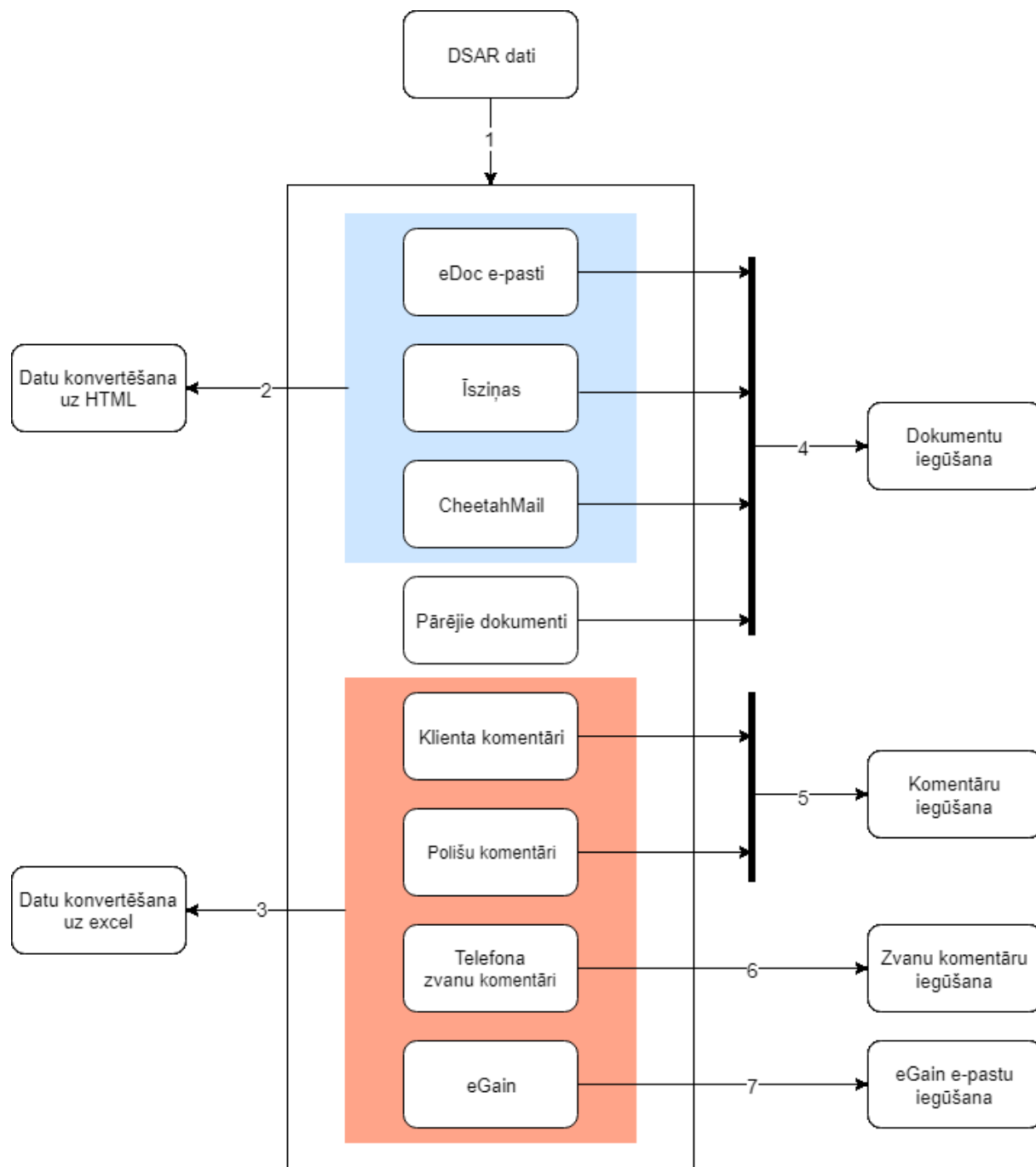
attiecīgais ziņojums. Attēlā tiek norādīti cipari pie katras darbības, kas tiek izmantoti 2.7. nodaļā prasību specifikāciju sasaistīšanai ar projektējuma aprakstu.



2.4.att. DSAR export apdarinātāja aktivitāšu diagramma

2.3.2. Datu iegūšanas un apstrādes shēma

Izstrādātās funkcionalitātes uzdevums ir konkrēta klienta visu dokumentu savākšana un pārvēršana citā datnes formātā. Attēla 2.5. ir redzama shēma, kurā ir parādīti dokumentu veidi, kas tiek savākti un to iedalījums konvertēšanā un dokumentu ieguves ceļā. Uz bultiņām norādītie cipari tiek zemāk aprakstīti – darbības un paskaidrojumi kas notiek šajā posmā.



2.5.att. Datu iegūšana un apstrāde

1. Lai klientam piegādātu visus datus, kas ir izveidojušies tam mijiedarbojoties ar *Bamboo* sistēmu, attēlā redzamie dati tiek apkopoti vienā *zip* datnē.

2. Klienta *eDoc* e-pasti, *CheetahMail* e-pasti un īsziņas tiek konvertētas no *XML* uz *HTML* datu formāta. To pārveidošanai tiek izmantots *XSLT* (paplašināmo stila lapu valodas transformācija) palīgs, kas ļauj nodefinēt, kurus datus no dokumentiem konvertēt un pielikt klāt papildus atribūtus, datus.
3. Komentāru dati un *eGain* e-pastu dati ir jāeksportē uz *excel* datni izmantojot kādu no brīvi pieejamajām bibliotēkām. Šie dati tiek glabāti dažādos formātos, tādēļ pirms eksportēšanas tie ir jā saglabā datu tabulā un jākonvertē datu tabulas ieraksti uz *excel* dokumenta tabulām.
4. Šie dokumenti glabājas uz dokumentu servera un tos atlasa izmantojot *Bamboo* datu bāzē iegūtos parametrus. Šie dati tiek piesaistīti klientam, polisei un kotējumam, un ir visi dokumenti, kas sistēmā glabājas par klientu. Lai efektīvāk iegūtu dokumentus ir jāizveido iekļautā procedūra, kas atlasa pēc noteiktiem laukiem visus datus izmantojot *SQL* vaicājumus.
5. Komentāri tiek piesaistīti klientam vai polisei, šie dati tiek glabāti *Bamboo* datubāzes tabulā. Dati tiek apstrādāti un ievadīti datu tabulās.
6. Telefona zvanu komentāri atrodas atsevišķā tabulā no klientu un polišu komentāriem. Dati tiek apstrādāti un ievadīti datu kopās.
7. *eGain* e-pasti glabājas datubāzē, kas nav pieejama izstrādātājiem un testētājiem nevienā no izstrādes vidēm. Lai iegūtu datus ir jāsimulē datubāzes tabulas visās vidēs izņemot produkcijas, kurā ir pieeja *eGain* datubāzei.

2.4. Datu apstrāde

Nodaļā tiek aprakstīta *DSAR export* funkcijas datu konvertēšana no *XML* datnes formāta uz *HTML* datnes formāta un datu eksportēšana uz *excel* datni.

2.4.1. Datu konvertēšana uz *HTML*

Apakšnodaļa attiecas uz prasību - #81455, #85242 realizācijas aprakstīšanu.

Bamboo sistēmā glabājas vairāki dokumentu veidi, *eDoc* e-pasti, īsziņas, *CheetahMail* e-pasti kuri tiek konvertēti uz *HTML* formāti.

eDoc e-pasti tiek ģenerēti pēc *ASP.NET* satvara izstrādātām veidnēm. Tā kā e-pasti ir dažādi un tajos saturiskā informācija ir katram citādāka, tad ir izstrādātas, katram e-pasta veidam sava veidne, kas satur *XML* dokumentu, ar dažādiem atribūtiem. 1. Pielikumā ir pievienots uzģenerēta e-pasta paraugs, kurā glabājas sekojoši atribūti: *document*, kas ietver sevī visu dokumentu, *recipient* – saņēmēja kontaktinformācija, *sender* – ziņas sūtītāja e-pasts, *subject* – priekšmets, *message* – galvenā dokumenta daļa, kas sevī ietver *HTML* un *CSS* kodu,

kurā glabājas attiecīgi noformēta e-pasta ziņa. Visi *eDoc* e-pasti pēc to uzģenerēšanas glabājas uz sistēmas servera *XML* datu formātā.

Īsziņas līdzīgi, kā *eDoc* e-pasti tiek ģenerēti izmantojot sistēmā izstrādātās veidnes. 2. pielikumā ir redzams uzģenerēta īsziņas dokumenta viens no paraugiem, kurš satur vairākus atribūtus: *document*, *recipient* – kas pilda tādas pašas funkcijas kā *eDoc* e-pastos, *mesage* – ziņas galvenais teksts bez nekādiem formatējumiem. *XML* kodā ir iespējams redzēt arī citus atribūtus, kas nav jāpiegādā klientam, jo nesatur ar viņu saistītu informāciju.

CheetahMail ir mārketinga risinājumu sniedzējs, kas ļauj sūtīt e-pastus, tos ģenerējot no dažādiem parametriem. Pēc to nosūtīšanas klientam, sistēmā saglabājas dati kuros ir tikai parametri vai arī kuri vēl satur *message* daļu, *XML* formātā. 3. Pielikumā ir redzams šāds e-pasts, kurš satur *message* (kas satur *HTML* un *CSS* noformētu ziņojumu) un daudzus parametrus, kuri tiek definēti kā atslēgu un vērtību pāri, kur ne visi no parametriem klientam ir vajadzīgi.

eDoc e-pastu, īsziņu un *CheetahMail* e-pastu konvertēšanai tiek izmantota *XSLT* stila lapas valoda.[5] 4. Pielikumā ir redzama stila lapa, kas konvertē šos datus *HTML* formātā. Pārveidošana notiek katram dokumenta veidam citādāk – tiek atstāti konkrētie *XML* atribūti. *eDoc* e-pastu konvertēšana notiek atstājot visus *XML* atribūtus, un pievienojot klāt papildus informāciju. *Message* atribūts tiek papildus apstrādāts. Īsziņām tiek pievienoti papildus informācija un atstāti *recipient* un *message* atribūti. *CheetahMail* tiek atstāti tikai konkrētie atslēgu un vērtību pāri, kuri ir nepieciešami klientam, kā arī *subjekt* un *mesage* dati, kas ietver *HTML* un *CSS* kodu ar visu ziņojumu.

2.4.2. Datu eksportēšana uz excel

Apakšnodaļa attiecas uz prasību - #81457, #86764, #81454 realizācijas aprakstīšanu.

Visi komentāru dati tiek glabāti *Bamboo* datu bāzē vairākās tabulās. Lai tos varētu nodot klientam tie ir jāpārveido uz pārskatāmu formātu, ko nodrošina *excel* datnes. Visus atlasītos datus eksportē uz *excel* datni tajā ievietojot tikai nepieciešamās tabulu kolonas. Pārveidošana tiek veikta uz *XLSX* datnes paplašinājumu, kas ir uz *XML* balstīts izklājlapas dokuments.

2.5. Datubāzes projektējums

Lietotāju stāstā ar identifikatoru - #81454 ir jāizstrādā neliela testa datubāze, kas satur vienu tabulu un iekļauto procedūru. Datubāzes mērķis ir nodrošināt testējamību *DSAR* eksportam, jo gan iegūtie dati, kas nāk no *eGain* datubāzes, gan pati datubāze nav pieejama *Bamboo* platformas izstrādātājiem un testētājiem, tādēļ rodas nepieciešamība izveidot jaunu testa datubāzi, kas simulē *eGain* datu bāzi un tabulas. Tabulā 2.1. ir aprakstīti *eGainTicket*

tabulas lauki, tā satur informāciju par e-pasta vēstules datiem, ko saņem klients vai kuru tas ir sūtījis *Bamboo* sistēmas darbiniekiem.

Datubāzes tabulu apraksts:

Tabulas ir aprakstītas, izmantojot šādu struktūru:

- Lauka nosaukums – tabulas kolonu nosaukumi
- Datu tips
- NULL (“+” - ja neobligāts)
- Apraksts
- Piezīmes

Saīsinājumi:

- PK - Primary Key (primārā atslēga)

2.1. tabula

Tabula *eGainTicket*

| Lauka nosaukums | Datu tips | NULL | Apraksts | Piezīmes |
|------------------------|------------------|-------------|--|-----------------|
| TID | INT | | Biļetes identifikators (TicketID) | PK |
| St | VARCHAR(1) | + | E-pasta biļetes statuss | |
| emailDate | DATETIME | + | Datums kad pievienots jauns e-pasta ieraksts | |
| Flw | VARCHAR(5) | | Nosaka e-pasta statusu – ienākošs, izejošs (In, Out) | |
| subject | VARCHAR(255) | + | E-pasta temats | |
| Content | VARCHAR(1000) | + | E-pasta ziņas saturs | |
| agent | VARCHAR(255) | + | Darbinieka vārds (kuram pienācis e-pasts, vai kurš ir to sūtījis). | |
| emailId | NUMERIC | | E-pasta identifikators | |
| note | VARCHAR(500) | + | Piezīmes | |
| USER_EMAIL | VARCHAR(50) | + | Klienta e-pasts | |

2.6. Moduļu projektējums

Zemāk aprakstīto moduli jāizstrādā izmantojot *HTTP* apdarinātāju (*handler*) [3], kas paredzēts tīmekļa lietojumprogrammu serveriem un nodrošina atbildes sniegšanu jebkurā brīdī kad tiek tam izveidots pieprasījums, kā arī *C#* programmēšanas valodu. *Bamboo* platforma izmanto *MS SQL* datubāzi, tādēļ datu iegūšanā tiek izmantota *SQL* valoda.

2.6.1. *DSAR export* apdarinātājs

Šis modulis ir paredzēts *Bamboo* projektā visu klientu datu iegūšanai, konvertēšanai un nogādāšanai klientam, kā arī atskaites veikšanai par funkcionalitātes izmantošanu. Datu iegūšanu *eCare* sistēmā lietotājs var veikt sameklējot attiecīgo klientu un nospiežot uz *DSAR export* saites. Lai varētu eksportēt datus un redzēt šo saiti, lietotājam ir jābūt autentificētam un jābūt ar *DSAR admin* lomu. Sekojoši notiek *zip* datnes lejupielāde uz lietotāja datora un asinhroniski [9] *Redmine* rīkā tiek izveidots jauns uzdevums un biznesa procesa vadītājam nosūtīts e-pasts par uzdevuma izveidi, kā arī pašā *eCare* sistēmā pie klienta tiek pievienots jauns komentārs, kurš satur informāciju par *DSAR export* izmantošanu. Savukārt, ja eksportēšana nav bijusi sekmīga, tad tiek izveidots jauns *Redmine* negadījuma uzdevums, kas tiek pievienots atbalsta darbiniekam.

2.7. Trasējamības matrica

Tabulā 2.2. tiek attēlota 2.5. attēlā norādīto *DSAR* datu un 2.4. attēlā aktivitāšu trasējamība uz prasību specifikācijas lietotāju stāstiem. Tabulas datu otrā kolona tiek sadalīta divās rindās, kur pirmajā ir aktivitāšu diagrammu aktivitātes, ar to sasaistītajiem cipariem, kas ir redzami attēlā 2.4. un otrajā datu iegūšanas un apstrādes diagrammas *DSAR* dati.

Trasējamības matrica

| Prasības | Aktivitātes un DSAR dati |
|----------|---|
| #81449 | <p>Attēls 2.4:</p> <ol style="list-style-type: none"> 1. <i>DSAR export</i> apdarinātāja izmantošana; 2. Iegūst klienta identifikatoru un datus; 6. Izveido <i>zip</i> datnes plūsmu, 10. Eksportē un konvertē kotējumu un polišu dokumentus, 11. Lejupielādē eksportētos datus, 12. Aizver <i>HTTP</i> kontekstu. <p>Attēls 2.5:</p> <p><i>eDoc</i> e-pasti, īsziņas, <i>CheetahMail</i>, pārējie dokumenti.</p> |
| #81455 | <p>Attēls 2.4:</p> <ol style="list-style-type: none"> 10. Eksportē un konvertē kotējumu un polišu dokumentus. <p>Attēls 2.5:</p> <p><i>eDoc</i> e-pasti, īsziņas</p> |
| #85242 | <p>Attēls 2.4:</p> <ol style="list-style-type: none"> 10. Eksportē un konvertē kotējumu un polišu dokumentus. <p>Attēls 2.5:</p> <p><i>CheetahMail</i>.</p> |
| #81457 | <p>Attēls 2.4:</p> <ol style="list-style-type: none"> 7. Eksportē un konvertē klientu un polišu komentārus. <p>Attēls 2.5:</p> <p>Klienta komentāri, polišu komentāri.</p> |
| #86764 | <p>Attēls 2.4:</p> <ol style="list-style-type: none"> 8. Eksportē un konvertē telefona zvanu komentārus. <p>Attēls 2.5:</p> <p>Telefona zvanu komentāri.</p> |

| | |
|--------|---|
| #81454 | <p>Attēls 2.4:</p> <p>9. Eksportē un konvertē <i>eGain</i> e-pastu komentārus.</p> <hr/> <p>Attēls 2.5:</p> <p><i>eGain</i></p> |
| #85800 | <p>Attēls 2.4:</p> <p>9. Eksportē un konvertē <i>eGain</i> e-pastu komentārus.</p> <hr/> <p>Attēls 2.5:</p> <p><i>eGain</i></p> |
| #83007 | <p>Attēls 2.4:</p> <p>4. Izveido <i>Redmine</i> uzdevumu,</p> <p>5. Aizsūta biznesa procesu vadītājam e-pastu</p> |
| #82681 | <p>Attēls 2.4:</p> <p>3. Pievieno klientam komentāru</p> <hr/> <p>Attēls 2.5:</p> <p>Klienta komentāri</p> |
| #82983 | <p>Attēls 2.4:</p> <p>13. Saglabā kļūdu, izveido <i>Redmine</i> uzdevumu</p> |

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Ievads

3.1.1. Nolūks

Šajā nodaļā tiek aprakstīta izstrādātās funkcionalitātes testēšanas gaita un rezultāti. Testēšanas procesā tiek izmantots pelēkās kastes princips, kā arī izveidoti vienību un integrācijas testi.

3.1.2. Saistība ar citiem dokumentiem

Dokumentācija izstrādāta balsoties uz programmatūras prasību specifikācijā iekļautajām funkcionālajām un nefunkcionālajām prasībām. Testēšanas dokumentācijā tiek ņemti vērā programmatūras projektējuma aprakstā iekļautie sistēmas darbības principi.

3.1.3. Testēšanas apraksts

Testēšanas žurnālā tiek apkopoti automātiski veiktie vienību testi un integrācijas testi, kā arī manuāli veiktie testi. Testēšanā tiek testēta visa galvenā sistēmas funkcionalitāte un tiek fiksēti to ievaddati, sagaidāmie rezultāti un vai tests ir veiksmīgi izpildījies.

3.1.4. Testpiemēru specifikācija

Testi tiek aprakstīti izmantojot šādu struktūru:

- Nr – testu numerācija (sanumurēti pēc kārtas).
- Tests – testa apraksts.
- Ievaddati.
- Sagaidāmais rezultāts.
- Rezultāts ('+', ja testa rezultāti sakrīt ar sagaidāmajiem rezultātiem, ja nē tad '-').

Testi tiek iedalīti pēc lietotāju stāstiem, kuru identifikatori ir pievienoti katrai testēšanas žurnāla apakšnodaļai. Integrācijas testi sistēmā tiek palaisti vienu reizi dienā un vienību testi, katru reizi kad tiek sabūvēts projekts.

3.1.5. Testēšanas žurnāls

3.1.5.1. Polises un kotējumu dokumentu eksportēšanas testi – #81449

3.1. tabula

klientu dokumentu testu tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|---|---|---|-----------|
| 1. | Ievada datus par polisi un pārbauda vai tie ir korekti apstrādāti. | Dokumenta nosaukums, dokumenta atrašanās vieta, dokumenta versijas numurs, ģenerētā dokumenta id, polises numurs, polises versijas numurs, transakcijas id, dokumenta tipa id, produkta id, datnes nosaukums. | Ievadītie dati tiek piesaistīti eksportējamu dokumentu sarakstam. Datnes nosaukums ir atbilstoši izveidots. | + |
| 2. | Ievada datus par polisi un pārbauda vai tie ir korekti apstrādāti, ja transakcijas identifikators ir tukšs. | Dokumenta nosaukums, dokumenta atrašanās vieta, dokumenta versijas numurs, ģenerētā dokumenta id, polises numurs, polises versijas numurs, transakcijas id, dokumenta tipa id, produkta id, datnes nosaukums. | Ievadītie dati tiek piesaistīti eksportējamu dokumentu sarakstam. Datnes nosaukums ir atbilstoši izveidots. | + |
| 3. | Ievada datus par kotējumu un pārbauda vai tie ir korekti apstrādāti. | Dokumenta nosaukums, dokumenta atrašanās vieta, dokumenta versijas numurs, ģenerētā dokumenta id, polises numurs, polises versijas numurs, transakcijas id, dokumenta tipa id, produkta id, datnes nosaukums. | Ievadītie dati tiek piesaistīti eksportējamu dokumentu sarakstam. Datnes nosaukums ir atbilstoši izveidots. | + |

| | | | | |
|----|--|---|---|---|
| 4. | Pārbauda vai dokumentu dati ir korekti, ja klientam nav neviens dokuments. | Tukšs saraksts. | Datnes nosaukums ir tukšs. | – (atgriež nepareizu datnes nosaukumu) |
| 5. | Nopērk klientam jaunu polisi un pārbauda vai visi dokumenti ir eksportētajā <i>zip</i> datnē. | Polises izveidošanai nepieciešamie dati, klienta dati. | Dokumentu daudzums <i>zip</i> datnē sakrīt ar klienta dokumentu daudzumu <i>eCare</i> sistēmā | – (Dokumentu skaits nesakrīt, jo vienam dokumentam ir izveidojušās divas versijas) |
| 6. | Pievieno klientam jaunu kotējumu un pārbauda vai visi dokumenti ir eksportētajā <i>zip</i> datnē | Lietotāja dati, kotējuma izveidošanai nepieciešamie dati, polises numurs. | Dokumentu daudzums <i>zip</i> datnē sakrīt ar klienta dokumentu daudzumu <i>eCare</i> sistēmā | + |

Otrā testēšanas reize tikai tiem testiem, kuri pirmajā reizē neizgāja:

3.2. tabula

Otrā testēšanas reize klientu dokumentu testu tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|--|--|---|-----------|
| 1. | Pārbauda vai dokumentu dati ir korekti, ja klientam nav neviens dokuments. | Tukšs saraksts. | Datnes nosaukums ir tukšs. | + |
| 2. | Nopērk klientam polisi un pārbauda vai visi dokumenti ir | Lietotāja dati, polises izveidošanai nepieciešamie dati, polises numurs. | Dokumentu daudzums <i>zip</i> datnē sakrīt ar klienta dokumentu daudzumu <i>eCare</i> sistēmā | + |

| | | | | |
|--|-------------------------------|--|--|--|
| | eksportētajā <i>zip</i> datnē | | | |
|--|-------------------------------|--|--|--|

3.1.5.2. Dokumentu konvertēšanas uz *HTML* testi – #81455, #85242

3.3. tabula

XML dokumentu konvertēšanas testu tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|--|---|---|-----------|
| 1. | Testē ja <i>XML</i> dokuments ir tukšs. | Dokumenta teksts. | Tukšs teksts. | + |
| 2. | Testē, ja jākonvertē īsziņas dokuments. | Īsziņas teksts ar attiecīgu <i>XML</i> noformējumu. | Pārveidots <i>XML</i> dokuments uz <i>HTML</i> dokumentu. Tiek atstāta tikai klientam nepieciešamā informācija. | + |
| 3. | Testē, ja jākonvertē <i>eDoc</i> e-pasta dokuments. | E-pasta teksts ar attiecīgu <i>XML</i> noformējumu. | Pārveidots <i>XML</i> dokuments uz <i>HTML</i> dokumentu. Tiek atstāta tikai klientam nepieciešamā informācija. | + |
| 4. | Testē ja <i>CheetahMail</i> dokumenti ir jākonvertē. | E-pasta teksts ar attiecīgu <i>XML</i> noformējumu. | Pārveidots <i>XML</i> dokuments uz <i>HTML</i> dokumentu. Tiek atstāta tikai klientam nepieciešamā informācija. | + |

3.1.5.3. Klientu un polišu komentāru testi – #81457

3.4. tabula

Klientu un polišu komentāru testu tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|--|---|---|-----------|
| 1. | Ievadītie polišu komentāra dati ir tukši. | Saglabāšanas datums, tipa apraksts, komentāra teksts, beigu datums. | <i>Excel</i> datnē ir ievadīta tukša rindiņa. | + |
| 2. | Ievadītie klientu komentāra dati ir tukši. | Saglabāšanas datums, tipa apraksts, komentāra teksts, beigu datums. | <i>Excel</i> datnē ir ievadīta tukša rindiņa. | + |
| 3. | Klientam vai polisei nav | Tukšs polišu komentāru saraksts. | <i>Excel</i> dokuments ir tukšs. | + |

| | | | | |
|----|--|--|---|---|
| | pievienoti komentāri. | | | |
| 4. | Polises komentārs tiek eksportēts. | Saglabāšanas datums, tipa apraksts, komentāra teksts, beigu datums. | Ievadītie dati tiek eksportēti un korekti ievietoti <i>excel</i> datnē. | + |
| 5. | Klienta komentārs tiek eksportēts. | Saglabāšanas datums, tipa apraksts, komentāra teksts, beigu datums. | Ievadītie dati tiek eksportēti un korekti ievietoti <i>excel</i> datnē. | + |
| 6. | Nopērk jaunu polisi un pievieno komentāru polisei un klientam un pārbauda vai <i>excel</i> datņu daudzums sakrīt ar paredzēto. | Lietotāja dati, polises izveidošanai nepieciešamie dati, polises numurs, komentāru dati. | Izveidoto <i>excel</i> dokumentu skaits (kas attiecas uz komentāriem) sakrīt ar polišu daudzumu (kurām tika pievienots komentārs) un klientu daudzumu (kas ir viens klients). | + |

3.1.5.4. *eGain* e-pastu test – #81454, #85800

3.5. tabula

eGain e-pastu testu tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|--|--|--|-----------|
| 1. | Pārbauda rezultātu, ja ievadītajā <i>eGain</i> e-pasta komentāra tekstā ir <i>HTML</i> tagi. | Biļetes id, statuss, izveidošanas laiks, stāvoklis, priekšmets, saturs, izveidotājs, e-pasta id. | Ievadītie dati ir piesaistīti komentāram, atbilstošā formātā (bez <i>HTML</i> tagiem) un <i>excel</i> datne satur nepieciešamos komentāru datus. | + |
| 2. | Pārbauda rezultātu, ja ievadītajam e-pasta komentāram nav <i>HTML</i> tagi. | Biļetes id, statuss, izveidošanas laiks, stāvoklis, priekšmets, saturs, izveidotājs, e-pasta id. | Ievadītie dati ir piesaistīti komentāram un <i>excel</i> datne satur nepieciešamos komentāru datus. | + |
| 3. | Tiek pievienots tukšs komentārs. | Biļetes id, statuss, izveidošanas laiks, stāvoklis, priekšmets, | Jāizmet izņēmums par to, ka tabulas dati nevar būt tukši. | + |

| | | | | |
|----|--|---|--|---|
| | | saturs, izveidotājs, e-pasta id. | | |
| 4. | Ja klientam nav pievienoti neviens <i>eGain</i> e-pasts. | Nav. | <i>Excel</i> datnei ir jābūt tukšai. | + |
| 5. | Pārbauda vai <i>excel</i> datnes kolonām ir pareizi nosaukumi, secība un skaits. | Jauna aizpildīta rinda <i>eGain</i> tabulā. | Kolonu nosaukumiem, to secībai un skaitam ir jāsakrīt ar prasībās minēto. | + |
| 6. | Klientam nopērk divas polises – katru uz citu e- pastu, pievieno <i>eGain</i> komentāru un pārbauda vai tie atrodas <i>zip</i> datnē. | Lietotāja dati, , polises izveidošanai nepieciešamie dati, polises numurs, <i>eGain</i> e-pasta saturošie dati. | Eksportētajā <i>zip</i> datnē ir jāatrodas attiecīgi cik e-pasti tik arī daudz <i>eGain</i> komentāru <i>excel</i> datnē. | + |

3.1.5.5. Telefona zvanu komentāru testi – #86764

3.6. tabula

Telefona zvanu komentāru testu tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|---|--|--|-----------|
| 1. | Pievieno jaunu telefona zvanu komentāru un pārbauda vai <i>zip</i> datne satur šo dokumentu. | Lietotāja dati, polises izveidošanai nepieciešamie dati, polises numurs, zvana komentāra dati. | <i>eCare</i> sistēmā ir pievienots jauns komentārs un <i>zip</i> mapē atrodas <i>excel</i> datne, kura satur pievienoto komentāru. | + |

3.1.5.6. Eksporta izmantošanas reģistrēšanas testi – #82681

3.7. tabula

DSAR export reģistrēšanas testu tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|---|--|---|-----------|
| 1. | Pievieno klientam jaunu kotējumu, <i>eCare</i> sistēmā veic <i>DSAR export</i> un pārbauda vai klientam ir pievienots komentārs par eksportu. | Lietotāja dati, kotējuma izveidošanai nepieciešamie dati, kotējuma numurs. | <i>eCare</i> sistēmā pie klienta ir pievienots jauns komentārs, kurš satur prasībās noteikto informāciju. | + |

3.1.5.7. DSAR admin lomas testēšana (manuālā testēšana)– #82981

Testēšana notiek manuāli, lai nevajadzētu automātiski visi laiku kādam lietotājam piešķirt un noņemt lomu.

3.8. tabula

DSAR admin lomas testēšanas tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|--|---------------------------------|---|-----------|
| 1. | Pievieno lomu <i>DSAR admin</i> lokālajam <i>eCare</i> lietotājam un pārbauda pogas esamību. | Lietotāja dati, polises numurs. | <i>eCare</i> sistēmā, atverot klienta informāciju ir jābūt <i>DSAR export</i> pogai. | + |
| 2. | Autorizējas <i>eCare</i> sistēmā ar lietotāju, kuram nav piešķirta <i>DSAR admin</i> loma un pārbauda pogas esamību. | Lietotāja dati, polises numurs. | <i>eCare</i> sistēmā, atverot klienta informāciju nav pievienota <i>DSAR export</i> poga. | + |

3.1.5.8. Asinhronās funkcionalitātes testēšana (manuālā testēšana) - #86635

Redmine uzdevuma un e-pasta asinhronā izpilde tiek skatīta atklūdotājā un tiek pārbaudīts vai izveidojas datu eksportēšana notiek paralēli uzdevuma un e-pasta izveidei.

3.9. tabula

***DSAR export* reģistrēšanas testu tabula**

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|----------------------------------|---------------------------------|---|-----------|
| 1. | Veic <i>DSAR export</i> funkciju | Lietotāja dati, polises numurs. | <i>Redmine</i> uzdevums un e-pasts ir asinhroni implementēti. | + |

3.1.5.9. *Redmine* uzdevuma un paziņojuma automātiskas izveides testēšana (manuālā testēšana)– #83007

Testēšana notiek manuāli, lai *Redmine* rīkā katru reizi kad tiek palaisti testi netiktu izveidoti jauni uzdevumi, kas tikai sakrātos projekta pārvaldības rīkā.

3.10. tabula

***Redmine* un e-pasta paziņojums biznesa procesu vadītājam - testēšanas tabula**

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|---|---------------------------------|---|-----------|
| 1. | <i>eCare</i> sistēmā uzspiež uz <i>DSAR export</i> un pārbauda vai <i>Redmine</i> rīkā ir izveidojies uzdevums ar parametriem kādi tiek definēti prasībās un vai e-pasts ir pienācis biznesa procesu vadītājam. | Lietotāja dati, polises numurs. | <i>Redmine</i> rīkā ir izveidojies jauns uzdevums un uz e-pastu ir atnācis paziņojums par tā izveidi. | + |

3.1.5.10. *Redmine* uzdevuma izveide kļūdu gadījumā - testēšana (manuālā testēšana)– #82983

Testēšana notiek manuāli, lai *Redmine* rīkā neizveidotos pārāk daudz uzdevumu, kā tas notiktu ja taisītu automātisko testu.

3.11. tabula

***Redmine* jauna uzdevuma izveide eksporta kļūdas gadījumā - testēšanas tabula**

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|--|---|---|-----------|
| 1. | <i>eCare</i> sistēma, veic <i>DSAR export</i> klientam, kurš neeksistē, <i>Redmine</i> rīkā pārbauda vai ir izveidojies negadījuma uzdevums. | <i>DSAR export</i> apdarinātāja vienotā resursa vietrādis, relation id. | <i>Redmine</i> rīkā ir izveidojies jauns negadījuma uzdevums. | + |

3.1.5.11. Nefunkcionālo prasību testēšana (manuālā testēšana)

Šī nodaļa atsaucas uz 1.4. nodaļu prasību specifikācijas nefunkcionālajām prasībām.

Tabulā 3.12. ir rādīti drošības prasību manuālās testēšanas rezultāti. Testēšana aptver 1.4.1. drošība – prasību specifikācijas nodaļu.

3.12. tabula

Nefunkcionālās prasības – drošība testēšanas tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|--|---|--|-----------|
| 1. | <i>eCare</i> lietotājam piešķir <i>DSAR admin</i> lomu un pēc autorizēšanās sistēmā un klienta atlases, pārbauda vai ir redzama <i>DSAR export</i> poga. | <i>DSAR admin</i> loma, lietotāja dati, polises numurs. | <i>eCare</i> sistēmā pie klienta datiem ir redzama <i>DSAR export</i> poga.. | + |

| | | | | |
|----|--|---|--|---|
| 2. | Autorizējas <i>eCare</i> sistēmā ar lietotāju, kuram nav piešķirta <i>DSAR admin</i> : loma un pārbauda vai ir pieejams <i>DSAR export</i> . | Lietotāja dati, polises numurs. | <i>eCare</i> sistēmā pie klienta datiem nav redzama <i>DSAR export</i> poga. | + |
| 3. | Veic <i>DSAR export</i> klientam kurš neeksistē. | Vienotais resursu vietrādis, relation id. | Ir jāparāda kļūdas ziņojums, kurš tiek saglabāts datubāzē un <i>Redmine</i> rīkā ir jābūt izveidotam jaunam uzdevumam, kurš satur kļūdas ziņojumu. | + |

Tabulā 3.13. ir rādīti privātuma prasību manuālās testēšanas rezultāti. Testēšana aptver 1.4.3. privātums, prasību specifikācijas nodaļu.

3.13. tabula

Nefunkcionālās prasības – privātums testēšanas tabula

| Nr. | Tests | Ievaddati | Sagaidāmais rezultāts | Rezultāts |
|-----|---|----------------|--|-----------|
| 1. | <i>eCare</i> sistēmā uzspiež uz <i>DSAR export</i> pogas. | Lietotāja dati | Pie klienta komentāriem tiek reģistrēta eksportēšana. <i>Redmine</i> sistēmā ir izveidojies jauns uzdevums, kas satur informāciju par eksportu un tiek piegādāts e-pasts, kas satur uzdevuma identifikatoru. | + |

4. PROJEKTA ORGANIZĀCIJA

Projekts tiek izstrādāts pēc spējās programmatūras izstrādes metodoloģijas, kas tika izvēlēta, jo apdrošināšanas platformu izstrādei tā ir pieņemtā prakse organizācijā un uzsākot projektu nav noteiktas visas prasības un izstrādes gaitā tās var mainīties un tikt pielabotas.

Spējās programmatūras izstrādei, tiek izmantots *SCRUM* satvars, kas ietver 2 nedēļu garus sprintus. Katrs sprints sastāv no iterācijas, kuras laikā ir jāpabeidz attiecīgie lietotāju stāsti (kuru izpildes termiņš iekļaujas laikā). Projektu izstrādāja viens cilvēks un ieteikumus programmatūras kodam un dokumentācijai piedāvāja nozares eksperti. Katru dienu tiek veidotas ik rīta sapulce, kurā apspriež ko iepriekšējā dienā izdarīja, ko darīs turpmāk un kādi ir kavēkļi. Katras iterācijas beigās tiek organizēta retrospektīvā sapulce, kurā dalās par to, kas patika sprintā, kāda bija sadarbība ar komandu, kādas grūtības bija un ko varētu uzlabot. Savukārt uzsākot iterāciju notiek sprinta plānošana, kurā izvēlas kurus lietotāju stāstus pildīs attiecīgajā sprintā.

Uzdevumi tiek iedalīti lietotāju stāstos, kurus izveido biznesa analītiķi un nepieciešamības gadījumā arī tos papildina. Izstrādes gaitā var gadīties, ka lietotāju stāsti nav pietiekami specifiski, tādēļ var *Redmine* rīkā izveidot atsauksmi tādējādi sazināties ar biznesa analītiķiem un pilnveidot tos.

Visi projekta lietotāju stāsti tiek glabāti *Redmine* pārvaldības rīkā, kurā ir iespējams pievienot tos sprintam un redzēt kura koda daļa ir saistīta ar kuru lietotāju stāstu.

5. KONFIGURĀCIJU PĀRVALDĪBA

Sistēma sastāv no diviem zariem *sprint* un *trunk* un vairākām izstrādes vidēm. Izstrāde tiek veikta *sprint* zarā. Kad tiek izveidota jauna komponente tā tiek izvietota *DEV-SPRINT* vidē. Kad lietotāju stāsts ir gatavs testēšanai tas tiek izvietots *TEST-SPRINT* vidē. Kad programmatūra ir pabeigta, tad izstrādātāji apvieno šo lietotāju stāstu no *sprint* uz *trunk* zaru. Pēc apvienošanas *trunk* zarā kods tiek izvietots *TEST-TRUNK* vidē. Pēc tā seko koda testēšana un ievietošana *UAT* vidē. Pēdējais solis ir nogādāšana *PROD* vidē, kas ir sistēmas gala produkts.

Lai pārvaldītu šīs vides tiek izmantots *SVN* rīks kas ļauj no lokālās vides (kas atrodas katram izstrādātājam uz datora) izmaiņas izvietot uz citām vidēm (uz kopējo serveri nogādāt izmaiņas) kā arī atjaunināt lokālo vidi ar jaunākajām versijām. Koda izmaiņas tiek izvietotas pēc katra lietotāju stāsta vai labojumu izpildes. Versijas tiek glabātas ar atvērtā pirmkoda *TortoiseSVN* rīku, kas ļauj salīdzināt divu versiju izmaiņas savā starpā, skatīties kurš izstrādātājs ir veicis izmaiņu kādai no versijām, atjaunot kādu no vecākajām izmaiņām un noņemt visas tagadējās izmaiņas atgriežot dokumentu uz pēdējo versiju.

TurtoiseSVN ir savienots ar projektu pārvaldības rīku *Redmine*. Katra revīzija tiek sasaistīta ar lietotāju stāstu – katrā izmaiņu izvietojumā ir norādīts stāsta identifikators. Pēc izmaiņu nosūtīšanas automatizācijas serverī *Jenkins* veic projekta pārbūvi, kas ļauj tām stāties spēkā.

6. KVALITĀTES NODROŠINĀŠANA

Izstrādātā produkta kvalitātes nodrošināšanai, tiek lietotas vairākas metodes. Tiek izstrādāta programmatūras dokumentācija – programmatūras prasību specifikācija, kas iekļauj produktā nepieciešamās prasības, programmatūras projektējuma apraksts, kas balstās uz specifikāciju un izveidoti testēšanas scenāriji un dokumentācija, kas tiek balstīti uz specifikācijas un apraksta.

Organizācijā ir izstrādāts process, kur katra lietotāju stāsta beigās to pārskata cits izstrādātājs un kļūdu gadījumā nodod atpakaļ izstrādes procesā. Vēl tiek izmantots *SonarQube* rīks – atvērtā pirmkoda platforma, kas statiski novērtē programmatūras koda kvalitāti. Pabeidzot koda daļu visi brīdinājumi tiek sasaistīti ar konkrēto izstrādātāju un tiek nosūtīti e-pastā. Papildus tam izstrādē tiek ievērota vispārpieņemtā programmēšanas labā prakse.[8]

Darba autors izstrādā vienību testus un integrācijas testus, kas pārbauda uzrakstītā koda darbību un veikspēju. Sistēma satur arī jau izveidotus vienību un integrācijas testus, kas ļauj uz dažādām vidēm novērot izstrādātās programmatūras darbību no dažādiem skata punktiem. Pirms izmaiņas nonāk produkcijas vidē ir testētāji, kas veic manuālu testēšanu testa vidēs un pārliecinās par koda kvalitāti. Vajadzības gadījumā tiek reģistrētas kļūdas, kuras labo izstrādātāji.

7. DARBIETILPĪBA

Pirms katra lietotāju stāsta veikšanas tika novērtēta, darbietilpība funkcijpunktos, kur katrs punkts apzīmē vienu cilvēkdienu (8 stundas). Novērtēšana tika veikta autoram kopā ar nozares ekspertiem. Pēc lietotāju stāsta realizēšanas, tiek salīdzināta reālā darbietilpība ar plānoto. Viss kopējais izstrādes process tiek novērtēts uz 440 stundām, savukārt reālā darbietilpība ir 480 stundas, sasniedzot 3 personmēnešu darbietilpību.

7.1. tabula

Darbietilpības novērtējuma tabula

| Lietotāju stāsta identifikators | Novērtētā darbietilpība funkcijpunktos | Reālā darbietilpība funkcijpunktos |
|---------------------------------|--|------------------------------------|
| 82981 | 1 | 1 |
| 81449 | 15 | 17 |
| 81455 | 10 | 12 |
| 85242 | 3 | 5 |
| 81457 | 4 | 5 |
| 86764 | 3 | 2 |
| 81454 | 5 | 7 |
| 85800 | 2 | 1 |
| 83007 | 5 | 2 |
| 82681 | 2 | 1 |
| 86635 | 2 | 4 |
| 82983 | 3 | 3 |
| Kopā | 55 | 60 |

SECINĀJUMI

Kvalifikācijas darba rezultātā tika optimizēta apdrošināšanas platforma, izstrādājot automatizētu *DSAR export* funkciju, kas izpilda VDAR prasības un veic DSP. Izstrādātajai programmatūrai tika izveidota atbilstoša dokumentācija – programmatūras prasību specifikācija, projektējuma apraksts un testēšanas dokumentācija.

Izstrādājot programmatūras kodu pēc spējās izstrādēs metodoloģijas, tika secināts, ka ir jāpielāgojas izmaiņām, kuras rodas prasību modificēšanā vai precizēšanā. Ne vienmēr lietotāju stāsti tiek precīzi nodefinēti, šādos gadījumos ir jāprot komunicēt ar klientu, lai izvairītos no pārpratumiem.

Rakstot programmatūras kodu, tika secināts, ka svarīgi ir izmantot labās prakses principus koda noformēšanai, tas padara to uztveramāku un atvieglo darba gaitu.

Strādājot pie kvalifikācijas darba, tika iegūta pieredze un apgūtas zināšanas šādās valodās/tehnoloģijās:

- *C#* programmēšanas valoda kopā ar *.NET* ietvaru;
- *XML*;
- *MS SQL* datu bāze;
- *XSLT* valoda;
- *SVN* versiju kontroles rīks;
- *XPath* valoda;
- *Jenkins* automatizācijas serviss;
- *SonarQube* kvalitātes nodrošināšanas platforma;
- *Regex*;
- *Redmine* projekta pārvaldības rīks.

Kvalifikācijas darba izstrāde ir devusi pieredzi programmatūras dokumentācijas rakstīšanā, testēšanā, stiprinājusi programmēšanas teorētiskās un praktiskās zināšanas, kā arī devusi zināšanas kā veikt darbietilpības novērtējumu un strādāt komandā.

IZMANTOTĀ LITERATŪRA

1. "About SonarQube." SonarQube. [tiešsaiste] – [atsauce 24.05.2018] Pieejams: <https://www.sonarqube.org/about/>.
2. "CheetahMail." Marketing Forward Blog. [tiešsaiste] – [atsauce 24.05.2018] Pieejams: <https://www.experian.com/marketing-services/cheetahmail.html/>.
3. "HTTP Handlers and HTTP Modules Overview." About Processes and Threads (Windows). [tiešsaiste] – [atsauce 24.05.2018] Pieejams: <https://msdn.microsoft.com/en-us/library/bb398986.aspx>.
4. "Redmine." Overview - Redmine. [tiešsaiste] – [atsauce 24.05.2018] Pieejams: <https://www.redmine.org/>.
5. "XML and XSLT." W3Schools Online Web Tutorials. [tiešsaiste] – [atsauce 24.05.2018] Pieejams: https://www.w3schools.com/xml/xml_xslt.asp.
6. "Jenkins User Documentation." Blue Ocean. [tiešsaiste] – [atsauce 24.05.2018] Pieejams: <https://jenkins.io/doc/>
7. Akadēmiskā terminu datu bāze – [tiešsaiste] – [atsauce 24.05.2018] Pieejams: <http://termini.lza.lv/>
8. BillWagner. "C# Coding Conventions (C# Programming Guide)." Microsoft Docs. [tiešsaiste] – [atsauce 24.05.2018] Pieejams: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>.
9. Cartermp. "Asynchronous Programming." Microsoft Docs. [tiešsaiste] – [atsauce 24.05.2018] Pieejams: <https://docs.microsoft.com/en-us/dotnet/csharp/async>.
10. LVS ISO 5127:2005, Informācija un dokumentācija – Vārdnīca, LVS/STK/22 Muzeju, arhīvu un bibliotēku darbs, 2005.
11. Stellakis, Andrew. "GDPR in Focus: What Is a Data Subject Access Request?" Q2Q - Your IT Department. [tiešsaiste] – [atsauce 24.05.2018] Pieejams: <https://www.q2q-it.com/gdpr-news/item/222-gdpr-in-focus-what-is-a-data-subject-access-request>

PIELIKUMI

1. pielikums

eGain e-pasts XML formātā

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
<recipient><![CDATA[IanBWZ.MonETQ@mailinator.net]]></recipient>
<sender><![CDATA[Quotemehappy.com <hello@quotemehappy.com>]]></sender>
<subject><![CDATA[Re: Confirmation of the set up of your Direct Debit Instruction including future
payment schedule]]></subject>
<message type="html"><![CDATA[

<body link="#009900" alink="#009900" vlink="#009900" style="font-family:arial;">
  <div>
    <p>Hello IanBWZ,</p>

    <p>This is to confirm that you have authorised us to set up a Direct Debit on your bank.</p>

    <p>Having accepted your Direct Debit details, we'd like to confirm that they are correct.</p>

    <p>Many thanks</p>
    <p>Quotemehappy.com</p>

    <br/>
    <div id="ddblock" style="background: #f2f2f2; width:600px;">
      <div id="header" style="width:580px">
        <div id="title" style="display:inline">
          <img src="" style="display:inline;padding: 10px;margin-left: 70px;">
        </div>
      </div>
    </div>

    <div id="ddcont" style="padding-bottom:20px; padding-right: 10px;">
      <ul style="list-style-type:circle;">
        <li>This Guarantee is offered by all banks and building societies that accept instructions to
pay Direct Debits </li>
      </ul>
    </div>
  </div>
</body>
]]>
</message>
</document>
```

2. pielikums Īsziņas XML formātā

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  <recipient>07436527863</recipient>
  <message>Welcome to Quotemehappy.com. We've emailed details for car policy 606457026. Download FREE
My account app http://bit.ly/QMHipphone or http://bit.ly/QMHAndroid </message>
  <originator>
    <alphatag>Happy</alphatag>
  </originator>
  <attributes>
    <costCentreId>315</costCentreId>
    <antiSocialStart>2200</antiSocialStart>
    <antiSocialEnd>0800</antiSocialEnd>
    <expireAfter>020000</expireAfter>
  </attributes>
</document>
```

CheetahMail e-pasts XML formātā

```

<?xml version="1.0" encoding="utf-8"?>

<document>

  <email>obfName.obfSurname@bankofamerica.com</email>

  <aid>2092843492</aid>

  <eid>230178</eid>

  <parameters>

    <parameter>

      <key>FIRSTNAME</key>

      <value>obfName</value>

    </parameter>

    <parameter>

      <key>POLICYNUMBER</key>

      <value>609779398</value>

    </parameter>

    <parameter>

      <key>LASTNAME</key>

      <value>obfSurname</value>

    </parameter>

    <parameter>

      <key>SURNAME</key>

      <value>obfSurname</value>

    </parameter>

  </parameters>

  <subject><![CDATA[Your renewed policy with Quotemehappy.com begins today]]></subject>

  <message type="html"><![CDATA[<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"><html><head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"><title>obfName, we've renewed
your car insurance policy</title>

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <meta name="format-detection" content="telephone=no">

  <style type="text/css">

```

```

    @media screen and (max-device-width: 640px) {

        #tx16 {font-size:16px !important;}

        .get_a_car_quote { padding: 0px !important; }

        .update_payment_details { padding: 0px !important; }


        #renewNow {padding:20px !important;}

        #amendAndRenew {padding:0px 10px !important;}

    }

</style>

</head>

<body style="background-color:#ffffff; font-family:Arial ;margin:0; padding:0; font-size:14px;">

<div style="background-color:#ffffff; margin:0 auto; padding:0; width:600px" align="center">

    <!-- TOP GREEN BAR -->

    <table style="border:none;>

        <tbody><tr>

            <td style="width:600px; height:10px; background-color:#b2dd31" bgcolor="#b2dd31"
height="10" width="600"></td>

            </tr>

            <tr>

                <td style="width:600px; height:10px; background-color:#f5fae6" bgcolor="#f5fae6"
height="10" width="600"></td>

                </tr>

            </tbody></table>

    <!-- END TOP GREEN BAR -->

</div>

</body>

</html>]]></message>

</document>

```

XSLT dokuments – *eGain* e-pastu, *CheetahMail* un īsziņu konvertēšanai

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:XSLTFeeder = ""
  exclude-result-prefixes="#default XSLTFeeder xsl" >
  <xsl:template match="/">
    <xsl:choose>
      <xsl:when test="document/aps/alert">
        <div id="xslHeader">
          Notification Message
        </div>
        <div id="xslMessage">
        </div>
      </xsl:when>
      <xsl:when test="document/sender">
        <div id="xslHeader">
          Mail sent to : <xsl:value-of select="document/recipient" /> by <xsl:value-of
select="document/sender" />
        </div>
        <div id="xslSubject">
          Subject : <xsl:value-of select="document/subject" />
        </div>
        <div id="xslMailMessage">
          <xsl:value-of select="XSLTFeeder:FormatMessagePath(document/message)" />
        </div>
      </xsl:when>
      <xsl:when test="document/eid">
        <div id="xslHeader">
          Mail sent to : <xsl:value-of select="document/email" />
        </div>
        <div id="xslSubject">
          Subject : <xsl:value-of select="document/subject" />
        </div>
        <div id="xslMailMessage">
          <xsl:value-of select="XSLTFeeder:FormatMessagePath(document/message)" />
        </div>
        <div id="xslMessageParams">
          <table>
            <tbody>
              <tr>
                <th align="left">Parameter:</th>
                <th align="left">Value:</th>
              </tr>
              <xsl:for-each select="document/parameters/parameter">
                <tr>
                  <td>

```

```

        <xsl:value-of select="key" />
      </td>
      <td>
        <xsl:value-of select="value" />
      </td>
    </tr>
  </xsl:for-each>
</tbody>
</table>
</div>
</xsl:when>
<xsl:when test="document/landlineNumberValidationRegex">
  <div id="xslHeader">
    Voice message sent to : <xsl:value-of select="document/recipient" />
  </div>
  <div id="xslMessage">
    <pre>
      <xsl:value-of select="document/message" />
    </pre>
  </div>
</xsl:when>
<xsl:otherwise>
  <div id="xslHeader">
    SMS sent to : <xsl:value-of select="document/recipient" />
  </div>
  <div id="xslMessage">
    <pre>
      <xsl:value-of select="document/message" />
    </pre>
  </div>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:stylesheet>

```

DSAR export apdarinātāja klase

```
/// <summary>
/// DSAE export handler
/// </summary>
public class DsarExportHandler : IHttpHandler
{
    protected ILookupService lookupService;

    protected ILookupService LookupService
    {
        get
        {
            if (lookupService == null)
            {
                lookupService = new LookupService();
            }

            return lookupService;
        }
    }

    protected DsarEGainExportService dsarEGainExportService;
    protected DsarEGainExportService DsarEGainExportService
    {
        get
        {
            if (dsarEGainExportService == null)
            {
                var EGainBll = new EGainBLL();
                dsarEGainExportService = new DsarEGainExportService(EGainBll);
            }

            return dsarEGainExportService;
        }
    }

    private bool IsInRole(SecurityRole Role)
    {
        string roleName = Enum.GetName(typeof(SecurityRole), Role);
        if (HttpContext.Current != null)
        {
            return HttpContext.Current.User.IsInRole(roleName);
        }
        return false;
    }

    protected IPolicyService policyService;
```

```

protected IPolicyService PolicyService
{
    get
    {
        if (policyService == null)
        {
            policyService = ServiceBroker.Resolve<IPolicyService>();
        }

        return policyService;
    }
}

protected IPolicyCommentService policyCommentService;
protected IPolicyCommentService PolicyCommentService
{
    get
    {
        if (policyCommentService == null)
        {
            policyCommentService = ServiceBroker.Resolve<IPolicyCommentService>();
        }
        return policyCommentService;
    }
}

protected DsarPolicyCommentService dsarPolicyCommentService;
protected DsarPolicyCommentService DsarPolicyCommentService
{
    get
    {
        if (dsarPolicyCommentService == null)
        {
            dsarPolicyCommentService = new DsarPolicyCommentService(PolicyCommentService,
LookupService);
        }

        return dsarPolicyCommentService;
    }
}

protected DsarPhoneCallCommentService dsarPhoneCommentService;
protected DsarPhoneCallCommentService DsarPhoneCommentService
{
    get
    {
        if (dsarPhoneCommentService == null)
        {
            dsarPhoneCommentService = new DsarPhoneCallCommentService();
        }

        return dsarPhoneCommentService;
    }
}

```



```

    }
}
/// <summary>
/// Convert emails to html format and exports °ploicy and quote documents
/// </summary>
/// <param name="dsarExportedDocumentData">DsarExportedDocumentData</param>
/// <param name="zipStream">zip stream</param>
private void ExportDocumentsToZip(int relationId, Stream zipStream)
{
    var dalDsarExportDocuments = new DalDsarExportDocuments();
    var dsarDocumentService = new DsarDocumentService(dalDsarExportDocuments);
    dsarDocumentService.ExportDocumentsToZip(relationId, zipStream);
}

/// <summary>
/// Converts policy and user comments to excel and exports them to zip
/// </summary>
/// <param name="zipStream">zip stream</param>
/// <param name="person">Person</param>
private void ExportPolicyCommentsToZip(Stream zipStream, Person person)
{
    var policyGenerals = PolicyService.GetAllPolicyGenerals(person.RelationId);
    //Exports policy comments attached to policy
    foreach (var policyGeneralId in policyGenerals)
    {
        using (MemoryStream policyCommentStream = new MemoryStream())
        {
            DsarPolicyCommentService.GetPolicyCommentsByPolicyGeneralId(policyCommentStream,
policyGeneralId.PolicyGeneralId);
            if(policyCommentStream.Length > 0 )
            {
                CompressionHelper.ZipAddEntryStream(zipStream, $"{policyGeneralId.PolicyNumber}
Comments.xlsx", policyCommentStream);
            }
        }
    }

    //Exports policy comments attached to customer
    using (MemoryStream policyCommentStream = new MemoryStream())
    {
        DsarPolicyCommentService.GetPolicyCommentsByRelationId(policyCommentStream,
person.RelationId);
        if (policyCommentStream.Length > 0)
        {
            CompressionHelper.ZipAddEntryStream(zipStream, $"{person.Firstname} {person.Surname}
Comments.xlsx", policyCommentStream);
        }
    }
}

```

```

/// <summary>
/// Converts phone call comments to excel and exports them to zip
/// </summary>
/// <param name="zipStream">zip stream</param>
/// <param name="person">Person</param>
private void ExportPhoneCommentsToZip(Stream zipStream, Person person)
{
    var policyGenerals = PolicyService.GetAllPolicyGenerals(person.RelationId);
    foreach (var policyGeneralId in policyGenerals)
    {
        //Exports Phone Comments related to policy
        using (MemoryStream phoneCommentStream = new MemoryStream())
        {
            DsarPhoneCommentService.GetCallCommentsByPolicyGeneralId(phoneCommentStream,
policyGeneralId.PolicyGeneralId);
            if (phoneCommentStream.Length > 0)
            {
                CompressionHelper.ZipAddEntryStream(zipStream, $"{policyGeneralId.PolicyNumber} Phone
comments.xlsx", phoneCommentStream);
            }
        }
    }
}

/// <summary>
/// Convets eGain emails to excel and exports them to zip
/// </summary>
/// <param name="relationId">Relation id</param>
/// <param name="zipStream">zip stream</param>
private void ExportEGainEmailsToZip(Stream zipStream, int relationId)
{
    List<string> eMailAddresses =
DsarEGainExportService.GetEmaiAddressesByRelationId(relationId);
    foreach (var mail in eMailAddresses)
    {
        string eGainFileName = $"{mail} eGain emails.xlsx";
        using (MemoryStream stream = new MemoryStream())
        {
            //Converts eGain mails to excel
            try
            {
                DsarEGainExportService.ConvertToExcel(mail, stream);
            }
            catch (Exception ex)
            {
                LogEvent.Log((LogEvent.Emitter.eCare), ex, "Conversion of eGain emails failed");
            }
            if (stream.Length > 0)
            {
                CompressionHelper.ZipAddEntryStream(zipStream, eGainFileName, stream);
            }
        }
    }
}

```

```

    }
    }
}

/// <summary>
/// Creates new comment about the use of DSAR export feature
/// </summary>
/// <param name="context">HttpContext</param>
/// <param name="relationId">Relation id</param>
private void InsertComment(HttpContext context, int relationId)
{
    string commentText = "DSAR data exported";
    DateTime? dueDate = null;
    string createdBy = context.User.Identity.Name;

    PolicyComment newComment;
    newComment = PolicyCommentService.AddPolicyCommentToPerson(relationId,
        commentText,
        createdBy,
        Ineas.Business.PolicyCommentStatus.Closed,
        Ineas.Business.PolicyCommentType.PolicyAdmin,
        dueDate,
        false);

    newComment.StorageDate = DateTime.Now;
}

private void StartContextResponse (HttpContext context, string zipName)
{
    context.Response.Clear();
    context.Response.BufferOutput = false;
    context.Response.ContentType = "application/zip";
    context.Response.AddHeader("content-disposition", $"attachment; filename={zipName}");
}
/// <summary>
/// Logs errors and shows message
/// </summary>
/// <param name="context">HttpContext</param>
/// <param name="ex">Exception</param>
/// <param name="logResponse">Response message</param>
/// <param name="logMessage">Log message</param>
private void ErrorResponse(string logMessage, HttpContext context, Exception ex, string
logResponse)
{
    string logEvent = string.Empty;
    if (!string.IsNullOrEmpty(logResponse))
    {
        logEvent = LogEvent.Log((LogEvent.Emitter.eCare), LogEvent.Severity.Error,
LogEvent.Type.SystemException, logMessage);
    }
}

```

```

        context.Response.Write(logResponse);
    }
    else
    {
        if (ex == null)
        {
            logEvent = LogEvent.Log((LogEvent.Emitter.eCare), LogEvent.Severity.Error,
LogEvent.Type.SystemException, logMessage);
            context.Response.Write($"<p>Internal Error <br> Error code: {logEvent}</p>\n");
        }
        else
        {
            logEvent = LogEvent.Log((LogEvent.Emitter.eCare), ex, "Internal Error");
            context.Response.Write($"<p>Internal Error <br> Error code: {logEvent}</p>\n");
        }
    }
    //Creates Redmine incident
    Task.Run(() => CreateDsarRedmineIncident(context, logEvent));
}

/// <summary>
/// Creates new Redmine incident when export has failed
/// </summary>
/// <param name="context">HttpContext</param>
/// <param name="logEvent">Contains log id</param>
private void CreateDsarRedmineIncident(HttpContext context, string logEvent)
{
    string apiKey = WebConfigReader.GetRedmineUserApiKey;

    var subject = Resources.Global.IncidentSubject;
    var description = $"Exportation time: {context.Timestamp}{Environment.NewLine}Log id:
{logEvent}";

    int project = WebConfigReader.GetIncidentProjectId;
    int priority = WebConfigReader.GetPriorityId;
    int assignedTo = WebConfigReader.GetIncidentAssignedToId;
    string incidentApplicationName = WebConfigReader.GetIncidentApplicationName;
    string incidentEnvironment = WebConfigReader.GetIncidentEnvironment;

    TicketManager.CreateNewIncidents(apiKey, subject, description, project, priority,
assignedTo, incidentApplicationName, incidentEnvironment);
}

/// <summary>
/// Creates Redmine ticket about the use of "Dsar export"
/// </summary>
/// <param name="userName">eCare currently logged in user name</param>
/// <param name="person">Person</param>
private Issue CreateDsarRedmineTicket (Person person, string userName)
{

```

```

    string apiKey = WebConfigReader.GetRedmineUserApiKey;

    var subject = String.Format(Resources.Global.TaskSubject, person.Firstname,
person.Surname, person.EmailAddress);
    var description = $"Customer: {person.Firstname}, {person.Surname}
({person.EmailAddress}) " +
    $"{Environment.NewLine}RelationID: {person.RelationId} " +
    $"{Environment.NewLine}User: {userName} " +
    $"{Environment.NewLine}Date: {DateTime.Now}";

    int project = WebConfigReader.GetProjectId;
    int priority = WebConfigReader.GetPriorityId;
    int assignedTo = WebConfigReader.GetAssignedToId;
    int fixedVersion = WebConfigReader.GetFixedVersion;
    int parentIssue = WebConfigReader.GetParentIssue;

    var task = TicketManager.CreateNewTask(apiKey, subject, description, project, priority,
assignedTo, fixedVersion, parentIssue);
    return task;
}

/// <summary>
/// Sends email about the use of "Dsar export"
/// </summary>
/// <param name="task">Redmine issue</param>
private void SendNotificationEmail(Issue task)
{
    var message = new IdentityMessage
    {
        Destination = WebConfigReader.GetEmailTo,
        Subject = String.Format(Resources.Global.EmailSubject, task.Id),
        Body = String.Format(Resources.Global.EmailBody, task.Id),
    };
    var emailSrevive = new EmailService();
    emailSrevive.SendMail(message, WebConfigReader.GetEmailFrom, null);
}

/// <summary>
/// Exports all files to zip and performs file conversion if necessary
/// </summary>
/// <param name="context">HttpContext</param>
/// <param name="person">Person</param>
private void ExportFilesToZip(HttpContext context, Person person)
{
    using (var zipStream =
CompressionHelper.CreateZipFileStream(context.Response.OutputStream))
    {
        ExportPolicyCommentsToZip(zipStream, person);
        ExportPhoneCommentsToZip(zipStream, person);
        ExportEGainEmailsToZip(zipStream, person.RelationId);
    }
}

```

```

        ExportDocumentsToZip(person.RelationId, zipStream);
    }
}
private void CreateRedmineTicketAndSendNotification(Person person, string userName)
{
    var issue = CreateDsarRedmineTicket(person, userName);
    SendNotificationEmail(issue);
}
/// <summary>
/// DSAR export action, downloads all user documents
/// </summary>
/// <param name="context">HttpContext</param>
public void ProcessRequest(HttpContext context)
{
    try
    {
        if (context.User.Identity.IsAuthenticated && IsInRole(SecurityRole.DsarAdmin))
        {
            //Gets relation id
            int relationId;
            bool successfullyParsed = int.TryParse(context.Request.QueryString["relationId"], out
relationId);
            if (successfullyParsed)
            {
                //Gets person
                var person = PolicyService.GetPerson(relationId);
                if (person != null)
                {
                    string zipName = $"{person.Firstname} {person.Surname} {relationId}.zip";
                    InsertComment(context, relationId);

                    Task.Run(() => CreateRedmineTicketAndSendNotification(person,
context.User.Identity.Name));

                    StartContextResponse(context, zipName);
                    ExportFilesToZip(context, person);
                }
                else
                {
                    ErrorResponse($"Unable to find person with Relation id: {relationId}", context, null,
null);
                }
            }
            else
            {
                ErrorResponse($"Relation id is in incorrect format", context, null, null);
            }
        }
    }
}

```

```

        ErrorResponse("Error Unauthorized Access", context, null, "<p>Error Unauthorized
Access</p>\n");
    }
}
catch(Exception ex)
{
    ErrorResponse("Internal Error", context, ex, null);
}
finally
{
    HttpContext.Current.Response.Flush();
    HttpContext.Current.Response.SuppressContent = true;
    HttpContext.Current.ApplicationInstance.CompleteRequest();
}
}

public bool IsReusable
{
    get
    {
        return false;
    }
}
}

```

DSAR dokumentu izgūšana

```

public virtual IList<DsarExportedDocumentData> RetrieveGeneratedDsarDocuments(int
relationId)
{
    IList<DsarExportedDocumentData> result = new List<DsarExportedDocumentData>();
    DsarExportedDocumentData item;

    SqlParameter[] parameters = new SqlParameter[]
    {
        new SqlParameter { ParameterName = "relationId", Value = relationId },
    };

    int sqlCommandTimeout = ConfigReader.GetSQLCommandTimeout();

    //Opens connection to SQL server database and fills DsarExportedDocumentData with values
    using (SqlConnection dbcResource = DataAccessLayer.createDbcResource())
    {
        using (SqlDataReader reader = SqlHelper.ExecuteReader(dbcResource, null,
CommandType.StoredProcedure, "DsarExportDocuments", sqlCommandTimeout, parameters))
        {
            while (reader.Read())
            {
                item = new DsarExportedDocumentData();
                item.DocumentPath = (string)reader["DocumentPath"];
                item.TransactionId = reader["TransactionId"] as int?;
                item.GeneratedDocumentId = (int)reader["GeneratedDocumentId"];
                item.PolicyNumber = TryCast<decimal>(reader, "policy_number");
                item.PolicyVersionNumber = TryCast<int>(reader, "policy_version_number");
                item.DocumentName = (string)reader["Description"];
                item.DocumentVersionNumber = (long)reader["DocumentVersionNumber"];
                item.DocumentTypeId = (int)reader["DocumentTypeId"];
                item.ProductId = TryCast<int>(reader, "product_id");
                result.Add(item);
            }
        }
    }
    return result;
}

```


DOKUMENTĀRĀ LAPA

Kvalifikācijas darbs „*Tiešsaistes apdrošināšanas platformas optimizācija*” 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: *Monta Kasparone* _____ .05.2018.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *M.dat. Vildis Vizulis* _____ .05.2018.

Recenzents: *M. soc.zin. Lauris Raipulis*

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): _____