

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**RISINĀJUMI APDROŠINĀŠANAS SISTĒMAS
TIA DARBĪBAS UZLABOŠANAI**

BAKALaura DARBS

Autors: **Oskars Kalniņš**

Studenta apliecības Nr.: ok13015

Darba vadītājs: docents Dr. dat. Aivars Niedrītis

RĪGA 2017

ANOTĀCIJA

Apdrošināšanas sistēma TIA ir sistēma, kas realizē apdrošināšanas biznesa pārvaldībai nepieciešamos pamata procesus. Bakalaura darba "Risinājumi apdrošināšanas sistēmas TIA darbības uzlabošanai" mērķis ir rast ieskatu, kā izstrādātāji var pielāgot sistēmas funkcionalitāti, lai tā atbilstu apdrošināšanas kompānijas specifiskajām prasībām. Darbā ir atrodams vispārīgs apdrošināšanas sistēmas TIA apraksts, kas parāda sistēmas uzbūvi un veidus, kā izstrādātāji ar sistēmu var strādāt. Darbā ir aprakstīti divi klientam piegādāti sistēmas uzlabojumi, kas demonstrē konkrētus piemērus tam, kā izstrādātājam ir iespējams pielāgot sistēmas darbību.

Atslēgas vārdi: apdrošināšanas sistēma TIA, biznesa lietotnes, prasību analīze, lietotņu pielāgošana, Oracle.

ANOTATION

Title: "Solutions for improving operation of insurance application TIA"

Insurance system TIA is a system that implements basic processes that are necessary for management of insurance business. Bachelor's thesis "Solutions for improving operation of insurance application TIA" goal is to provide insight of how developers can adjust system's functionality to conform to specific requirements of an insurance company. In this thesis, you will find and overall description of insurance system TIA, which shows structure of the system and ways in which developers, can work with the system. Thesis describes two improvements, which have been delivered to client, which provides specific examples of how developers can adjust system's workflow.

Key words: insurance application TIA, business applications, requirement analysis, application customization, Oracle.

SATURS

Apzīmējumu saraksts	1
Ievads	2
1. Apdrošināšanas sistēma TIA.....	3
1.1. Vispārējs sistēmas apraksts	3
1.2. Sistēmas tehniskais apraksts	4
1.3. Darbs ar apdrošināšanas sistēmu TIA	5
2. Mēneša beigu grāmatojumu optimizācija	9
2.1. Eksistējošais TIA kodola risinājums.....	9
2.2. Biznesa problēma	11
2.3. Problēmas analīze.....	12
2.4. Izvēlētais risinājums.....	15
2.5. Secinājumi.....	20
3. Risinājums automašīnu parka apdrošināšanas produkta maksājumu apstrādei	21
3.1. TIA kodola risinājums	21
3.2. Biznesa prasības	24
3.3. Risinājums transakciju pārvešanai uz plānotajiem rēķiniem	25
3.4. Risinājums gada pēdējā rēķina sagatavošanai.....	33
3.5. Risinājums visu transakciju redzamībai rēķinā.....	38
Rezultāti	42
Secinājumi.....	43
Izmantotāliteratūra	44
Pielikumi	45
1. pielikums. Mēneša beigu grāmatojumu skata izpildes plāns pirms optimizācijas.....	45
2. pielikums. Mēneša beigu grāmatojumu skata izpildes plāns pēc optimizācijas	46

APZĪMĒJUMU SARAKSTS

Apzīmējums	Skaidrojums
Atsauces tabula	Tabula, kas glabā identifikatoru aprakstu.
Dalītais lietojums	Lietojumprocesu kopums, kas saistīts ar kādu datoru sistēmu un sastāv no procedūrām, kuras var tikt izpildītas dažādos datoros.
Entītija	Kādam konkrētam lietojumam (sistēmai) nozīmīgs reāls vai abstrakts objekts vai šo objektu kopums.
Informācijas sistēma	Iekārtu, procedūru un personāla kopums, kas ir izveidots, strādā un tiek uzturēts, lai vāktu, uzkrātu, apstrādātu, uzglabātu un izmantotu informāciju.
Klients	Iekārta vai lietojumprocess, kam parasti ir tikai viens lietotājs un kas izmanto servera pakalpojumus.
Mērogojamība	Sistēmas programmatūras vai aparatūras spēja normāli funkcionēt gadījumā, ja tās funkcionēšanas apjoms vai konfigurācija tiek mainīta.
Pakešuzdevums	Pakešapstrādes režīmā izpildāms darbs.
Saskarne	Funkciju kopa, kas nodrošina datu apmaiņu starp divām ierīcēm, divām lietojumprogrammām vai starp lietotāju un lietojumprogrammu.
SOA	Uz pakalpojumiem orientēta arhitektūra (no angļu <i>service – oriented architecture</i>) ir programmatūras izstrādes pieeja, kurā pakalpojumi dažādām sistēmas komponentēm, tiek piegādāti izmantojot tīmekli.

IEVADS

Mūsdienās darbs ar gatavām informācijas sistēmām, kuras ir nepieciešams pielāgot konkrēta klienta biznesa specifikai, kļūst arvien biežāks. Tirgū ir pieejamas daudz dažādas informācijas sistēmas, kuras var pildīt noteiktam biznesam specifiskus uzdevumus. Diemžēl šīs sistēmas lieliski strādā vispārīgos biznesa gadījumos, bet tās nepiedāvā pasūtītājam specifiskus biznesa procesus, tāpēc kādam šie procesi ir jārealizē.

Bakalaura darba mērķis ir radīt ieskatu izstrādes darbā ar apdrošināšanas sistēmu TIA. Tā ir apdrošināšanas sistēma, kura realizē visus apdrošināšanas biznesam nepieciešamos procesus. Šī sistēma ir izstrādāta 3 slāņos, kas ir būvēti viens uz otra: kodola, valstij specifisko un klientam specifisko risinājumu slāņos. Kodola un valsts slāņos ir realizēti apdrošināšanas biznesa vadīšanai nepieciešamie pamata procesi, bet, ieviešot sistēmu, izstrādātājiem ir jāpielāgo sistēma darbība klienta vajadzībām, izmantojot klientam specifisko risinājumu slāni.

Bakalaura darbā ir sniegts apdrošināšanas sistēmas TIA apraksts, kā arī aprakstītas tehnoloģijas, kas ir izmatotas, lai realizētu šo sistēmu. Darbā ir augstā līmenī aprakstīti veidi, kā izstrādātāji drīkst strādāt ar sistēmu TIA, neiejaucoties sistēmas kodola slānī, kuru atbalsta sistēmas izstrādātājs "TIA Technology".

Darba turpinājumā ir detalizēti aprakstīti divu reālu biznesa problēmu risinājumi, lai rastu uzskatāmu ieskatu darbā ar sistēmu. Viens no risinājumiem ir virsgrāmatas perioda beigu grāmatojumu optimizācija, kas parāda, ko ir iespējams darīt šāda veida sistēmās gadījumos, kad kodola risinājumi sāk kļūt pārāk lēni, lai tos efektīvi lietotu. Otrs aprakstītais risinājums ir sistēmas maksājumu procesa pielāgošana, lai tas atbilstu klienta prasībām automašīnu parka apdrošināšanas produktam.

Abi šajā darbā aprakstītie risinājumi ir pilnībā vai daļēji autora izstrādāti. Bakalaura darbam nav pievienots risinājumu pirmkods, jo tas ir konfidenciāls un ir klienta īpašums. Darbā aprakstītie risinājumi ir piegādāti klientam un tiek lietoti apdrošināšanas biznesa pārvaldībai.

1. APDROŠINĀŠANAS SISTĒMA TIA

Šī nodaļa satur īsu sistēmas TIA aprakstu, lai veidotu kontekstu lasītājam par to, kas vispār ir šī sistēma un kā ar to strādāt. Nodaļa satur vispārēju sistēmas funkcionālo aprakstu, sistēmas tehniskā risinājuma un tajā izmantot tehnoloģiju aprakstu un nelielu ieskatu par to, kā izstrādātāji var strādāt ar šo sistēmu.

1.1. Vispārējs sistēmas apraksts

Apdrošināšanas sistēma TIA ir kompānijas “TIA Technology” izstrādāts apdrošināšanas sistēmas kodols. Nosaukums TIA ir abreviatūra no “*The Insurance Application*” jeb apdrošināšanas lietotne latviešu valodā. Šo sistēmu izmanto vairākas apdrošināšanas kompānijas Skandināvijā, Baltijas valstīs un Āfrikā. Šobrīd jaunākā sistēmas versija ir 7.5, bet šajā darbā aprakstītie risinājumi ir izstrādāti sistēmas versijai 6.4.

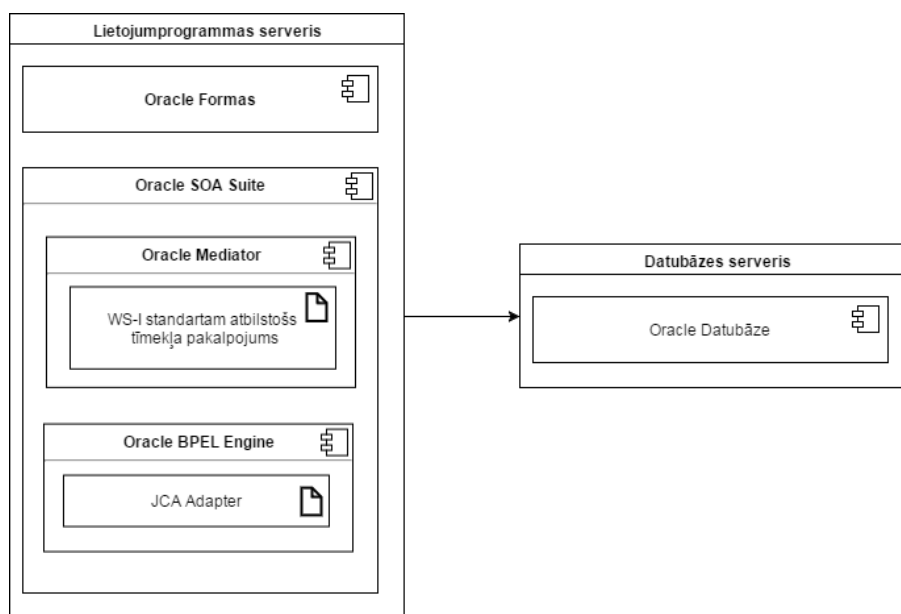
Sistēma nodrošina visu apdrošināšanas biznesa veiksmīgai pārvaldībai nepieciešamo funkcionalitāti. Sistēmas kodolā ir realizēti apdrošināšanas biznesa standartiem atbilstoši procesi, kas ir pielāgojami katra klienta individuālajām vajadzībām. Sistēma ir izdalāma vairākos moduļos, katrs no kuriem pilda noteiktu apdrošināšanas biznesa aspektu. Sistēmas moduļi ir:

- Polišu pārvaldības modulis - nodrošina visu apdrošināšanas polišu transakciju apstrādi un ar šīm transakcijām saistīto dokumentu izveidi;
- Atlīdzības pieprasījumu pārvaldības modulis – ļauj izveidot un pārvaldīt apdrošināšanas kompānijas klientu atlīdzības pieprasījumus;
- Produkta definīcijas modulis – satur nepieciešamo funkcionalitāti, lai sistēmā definētu jaunus apdrošināšanas produktus;
- Klientu modulis – ļauj sistēmā izveidot jaunus un pārvaldīt jau esošos kompānijas klientus, kā arī pārvaldīt mārketinga kampaņas;
- Grāmatvedības modulis – nodrošina izejošo un ienākošo maksājumu apstrādi, kā arī nodrošina virsgrāmatas datu sagatavošanu apstrādei ārējā virsgrāmatas sistēmā;
- Tehnoloģijas platforma – realizē pakešuzdevumu izpildi, sistēmas lokalizāciju, izdruku sagatavošanu un ļauj implementēt saskarnes ar ārējām sistēmām.

Sistēmas TIA kodolu uztur kompānija “TIA Technology” un tas nodrošina augstāk minēto funkciju darbību, bet šie moduļi ir augstā mērā pielāgojami, izmantojot lietotāja funkcijas, kas ļauj izstrādātājiem modificēt sistēmas standarta uzvedību.

1.2. Sistēmas tehniskais apraksts

Apdrošināšanas sistēma TIA ir izstrādāta pieturoties pie SOA programmatūras arhitektūras. Šāda arhitektūra ir izvēlēta, jo sistēmai ir jānodrošina iespēja apmainīties ar datiem, gan ar dažādām klienta iekšējām sistēmām, piemēram, ar grāmatvedības sistēmu, gan ar dažādām ārējām sistēmām, piemēram, iedzīvotāju vai automašīnu reģistriem. Papildu tam šāda arhitektūra ļauj izstrādāt dalītu lietojumu, kas nodrošina sistēmas mērogojamību un pieejamību no dažādiem klientu tipiem. Šāda arhitektūra ļauj arī sistēmu izdalīt slāņos, tādējādi ļaujot to pielāgot vairākos slāņos neietekmējot zemāka slāņa funkcionalitāti.



1.1. att. Sistēmas TIA arhitektūras shēma

Apdrošināšanas sistēma TIA ir realizēta Oracle ekosistēmā. Attēlā 1.1. ir redzamas sistēmas arhitektūras shēma. Sistēmas pamatā ir Oracle datubāzes serveris, kuram ir pieslēgts Oracle lietojumprogrammu serveris. Datubāzes serverī strādā Oracle datubāze, kas veic datu un datubāzes loģikas glabāšanu un izpildi. Datubāzē ir realizēta gandrīz visa TIA loģika, kas ir saistīta ar biznesa loģiku un datu apstrādi.

Lietojumprogrammas serverī, kas ir pieslēgts datubāzes serverim, strādā Oracle formas un Oracle SOA Suite. Oracle formas tiek izmantotas kā lietotāja saskarne apdrošināšanas kompānijas

darbiniekiem. Šajā modulī ir realizēta loģika datu attēlošanai un dati tiek sagatavoti apstrādei datu bāzē. Sākot no TIA versijas 7 Oracle formas ir aizstātas ar Oracle ADF, kas veic tādas pašas funkcijas kā Oracle Forms. Oracle SOA Suite tiek izmantots, lai realizētu apdrošināšanas kompānijas klientiem pieejamos kompānijas tīmekļa pakalpojumus vai realizētu saskarni ar citām sistēmām. Izmantojot Oracle Mediator TIA ir iespējams pieslēgt WS-I standartam atbilstošu tīmekļa pakalpojumu, no kura ir iespējams iegādāties apdrošināšanas polises vai arī izmantot kādu citu kompānijas piedāvāto pakalpojumu. Ar Oracle BPEL Engine palīdzību ir realizēts JCA Adapter, kas ļauj integrēt sistēmu TIA ar citām sistēmām un realizēt datu apmaiņu ar šīm sistēmām.

Šajā darbā aprakstītie uzlabojumi ir izstrādāti TIA versijā 6.4, kas uzstādīta uz Oracle datubāzes versijas 11g.

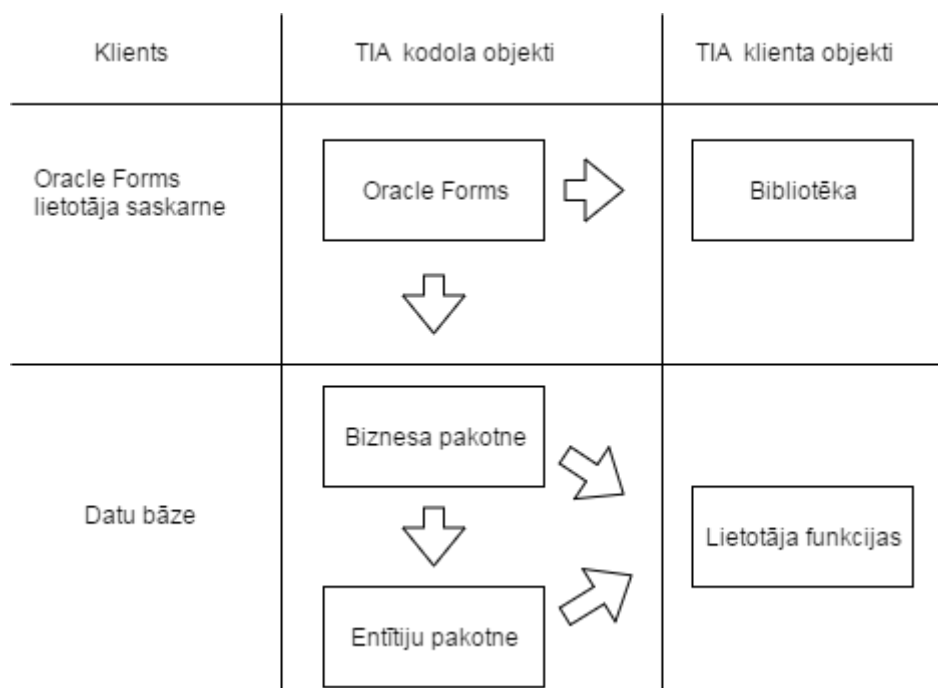
1.3. Darbs ar apdrošināšanas sistēmu TIA

Tā kā sistēma TIA ir bāzēta Oracle ekosistēmā, tad darbs ar to notiek izmantojot Oracle piedāvātās tehnoloģijas. Datu bāze ir izstrādāta izmantojot programmēšanas valodu PL/SQL. Apdrošināšanas sistēmas TIA lietotāja saskarne ir izstrādāta izmantojot Oracle Forms, kas arī izmanto programmēšanas valodu PL/SQL, lai realizētu programmas vienības un formu triggerus. Izdruku izstrādei tiek izmantots Oracle Reports.

Apdrošināšanas sistēmu TIA var izdalīt 3 slāņos: TIA kodols, valsts funkcionalitātes slānis un biznesa pielāgojumu slānis. Pamata slānis ir TIA kodols, kas vispārīgi realizē apdrošināšanas biznesa veikšanai nepieciešamos procesus. Uz TIA kodola tiek būvēts valsts funkcionalitātes slānis, kas pielāgo sistēmas funkcionalitāti atbilstoši tās valsts specifikai, kurā kompānija strādā. Augstākais slānis ir biznesa pielāgojumu slānis, kurā sistēma tiek pielāgota konkrētā klienta vajadzībām.

TIA kodolā realizētā funkcionalitāte ir vienāda pilnīgi visām TIA implementācijām. TIA kodols satur pilnīgi visas bibliotēkās, formas, izdrukas, tabulas, skatus un datubāzē iekļautās procedūras, kas tiek piegādātas kopā ar sistēmu TIA, izņemot datu bāzes objektus, kas ietilpst biznesa pielāgojumu slānī. Šo slāni uztur kompānija “TIA Technology” un tajā nav atļauts veikt izmaiņas. Ja kādā no TIA kodola objektiem tomēr tiek veiktas izmaiņas, tad “TIA Technology” šo objektu vairs neatbalsta. Šajā slānī ir realizēta funkcionalitāte, kas atbilst apdrošināšanas biznesa standartiem. Teorētiski TIA kodols ļauj veiksmīgi vadīt apdrošināšanas biznesu, bet tajā realizētā biznesa loģika ir ļoti vienkārša un var neatbilst ne valsts specifikai, ne kompānijas prasībām.

Apdrošināšanas sistēmas TIA kodolā ir iestrādātas lietotāja izejas jeb lietotājas funkcijas, kas ļauj izstrādātājiem pielāgot sistēmas funkcionalitāti.



1.2. att. Apdrošināšanas sistēmas TIA objektu veidu shēma

Attēlā 1.2. ir redzams kāda veida sistēmas TIA kodola objektiem ir pieejamas kāda veida lietotāja izejas. Ar kodolu piegādātajām sistēmas formām ir pieejamas lietotāja bibliotēkas, kuras ļauj pielāgot formu darbību pie dažādiem formu notikumiem. Sistēmas kodola biznesa pakotnes realizē apdrošināšanas procesu biznesa loģiku. Šajās pakotnēs ir iestrādātas lietotājas funkcijas, kas ļauj pielāgot noteiktus sistēmas procesus noteiktos brīžos. Entītiņu pakotnes ir veids, kādā sistēma ļauj izveidot vai atjaunot datus datubāzē. Šajās pakotnēs ir iestrādātas lietotāja izejas pirms un pēc dati datubāzē tiek izveidoti vai atjaunoti.

Valsts funkcionalitātes slānis pielāgo sistēmu TIA tās valsts vai reģiona specifiskām prasībām, kurā apdrošināšanas kompānija nodarbojas ar biznesu. Valsts funkcionalitātes slānis parasti satur:

- Valsts likumā prasīto funkcionalitāti;
- Datu pielāgojumus vietējiem standartiem, piemēram, adrešu vai pasta indeksu formātus;
- Standartus informācijas nodošanai vietējām varas iestādēm;
- Valstij specifiskus standarta modulūsus;
- Standarta lietotāja saskarnes tulkojumus.

Valsts funkcionalitātes slānis nav uzskatāms par TIA kodolu, bet valsts funkcionalitātes slāņa objekti ir uzskatāmi par TIA kodola objektiem, līdz ar to uz tiem attiecas tādi paši noteikumi par

izmaiņu veikšanu kā uz TIA kodolu. Arī ar valsts funkcionalitātes slāni tiek piegādāti objekti, kas ietilpst biznesa pielāgojumu slānī, jo arī šo funkcionalitāti var būt nepieciešams pielāgot konkrētā pasūtītāja vajadzībām. Ar valsts funkcionalitātes slāni var tikt ieviestas tādas funkcijas kā nodokļu regulas, banku saskarnes, saskarnes ar automašīnu reģistriem, saskarnes ar citām ārējām sistēmām un datu avotiem u.c.

Augstākais slānis ir biznesa pielāgojumu slānis. Šajā slānī ietilpst funkcijas, procedūras, bibliotēkas un skati, kas ir ieviesti ar mērķi radīt piekļuves punktus sistēmas ieviesējiem. Šādi ieejas punkti parasti ir izveidoti sistēmas procesu atslēgas brīžos, kad izstrādātājiem varētu būt nepieciešams iejaukties sistēmas darbībā. Piemēram, maksājuma izveides procesā ieejas punkti ir izveidoti brīdī, kad transakcija tiek izvēlēta iekļaušanai maksājumā un pēc tam, kad maksājums ir izveidots. Atsevišķiem procesiem, piemēram, finanšu transakcijas izvēlei iekļaušanai maksājumā, ir izveidots arī ieejas punkts procesa izpildes laikā, kas dod papildu iespēju izstrādātājiem pielāgot šī procesa norisi. Šos ieejas punktus sistēmā var atpazīt pēc prefiksa “UFxx_”, kur xx ir sistēmas moduļa un apakš moduļa numurs. Šajos ieejas punktos notiek lielākā daļa darbu sistēmas pielāgošanai, un tie ļauj pielāgot sistēmas darbību neizmainot sistēmas kodolu.

Apdrošināšanas sistēma TIA piedāvā lietotāja izejas punktu visām formām, kuras ietilpst TIA kodola risinājumā. Šīs lietotāja izejas tiek izsauktas no formas triggeriem, kas nostrādā pirms formas brīdī, kad formas fokuss pārslēdzas uz jaunu lauku, un pēc formas. Šos triggerus var izmantot, lai pielāgotu katras formas uzvedību un funkcionalitāti klienta specifiskajām vajadzībām. Šīs izejas ir realizētas Oracle Forms bibliotēkā un tās ietilpst sistēmas biznesa pielāgojumu slānī.

Papildu darbībām ar sistēmas biznesa pielāgojumu slāni, sistēmas uzvedību ir iespējams pielāgot izmantojot lietojuma iestatījumus. Ar lietojuma iestatījumiem ir iespējams norādīt sistēmai noklusētās vērtības, kas ietekmē sistēmas darbību. Ar lietojuma iestatījumiem var definēt tādas vienkāršas lietas kā datuma formāts un to, cik ilgi klients ir jāglabā nesen izmantoto klientu sarakstā, kā arī sarežģītākas lietas, piemēram, kā izvēlēties apdrošināšanas produkta versiju vai kā izvēlēties polises gada beigu datumu.

Vēl viena iespēja kā pielāgot apdrošināšanas sistēmu TIA ir definēt jaunas formas. Šādā veidā ir iespējams izveidot pilnībā pielāgotas formas, kuras var izmantot pilnīgi jebkādu lietotāja definētu loģiku. Tas ļauj izveidot funkcionalitāti, kas ir pilnībā neatkarīga no TIA kodola. To var darīt gadījumos, kad pasūtītājam ir nepieciešama funkcionalitāte, kas vispār neeksistē TIA kodolā, bet nav ieteicams šādi rīkoties gadījumos, kad sistēmā TIA jau eksistē līdzīga funkcionalitāte, kuru ir iespējams pielāgot, jo šādā veidā palielinās risks, kuru uzņemas izstrādātājs saistībā ar

potenciālajām kļūdām, kuras var tikt ieviestas sistēmā. Parasti jaunas formas tiek definētas tad, kad klients vēlas sistēmā redzēt datus, kurus TIA kodols sistēmā neparāda, piemēram, atsevišķa forma, kurā ir redzamas tikai atlīdzības izmaksas, kuras nepieciešams manuāli autorizēt.

Apdrošināšanas sistēma TIA ir augstā mērā pielāgojama katra klienta specifiskajām vajadzībām. Pirmais un vienkāršākais veids kā pielāgot apdrošināšanas sistēmu TIA ir izmantot lietojuma iestatījumus, kas ļauj neiejaucoties sistēmas pirmkodā pielāgot sistēmas darbību klienta vajadzībām. Sistēmā ir pieejamas arī lietotāja izejas, kas ļauj izstrādātājiem pielāgot sistēmu specifiskām klienta vajadzībām. Ja ar šīm divām lietām nav iespējams sistēmu pielāgot klienta vajadzībām, tad ir iespējams definēt pilnīgi jaunu funkcionalitāti, kas var būt neatkarīga no sistēmas TIA kodola.

2. MĒNEŠA BEIGU GRĀMATOJUMU OPTIMIZĀCIJA

Šajā nodaļā ir aprakstīts uzlabojums apdrošināšanas sistēmā TIA, kas optimizē kompānijas rezervju aprēķinu, kas notiek mēneša jeb grāmatvedības perioda beigās. Nodaļā ir aprakstīts sistēmas TIA standarta risinājums - ko tas dara, kāpēc šis risinājums sagādāja problēmas klientam, risinājuma analīze un pats risinājums.

2.1. Eksistējošais TIA kodola risinājums

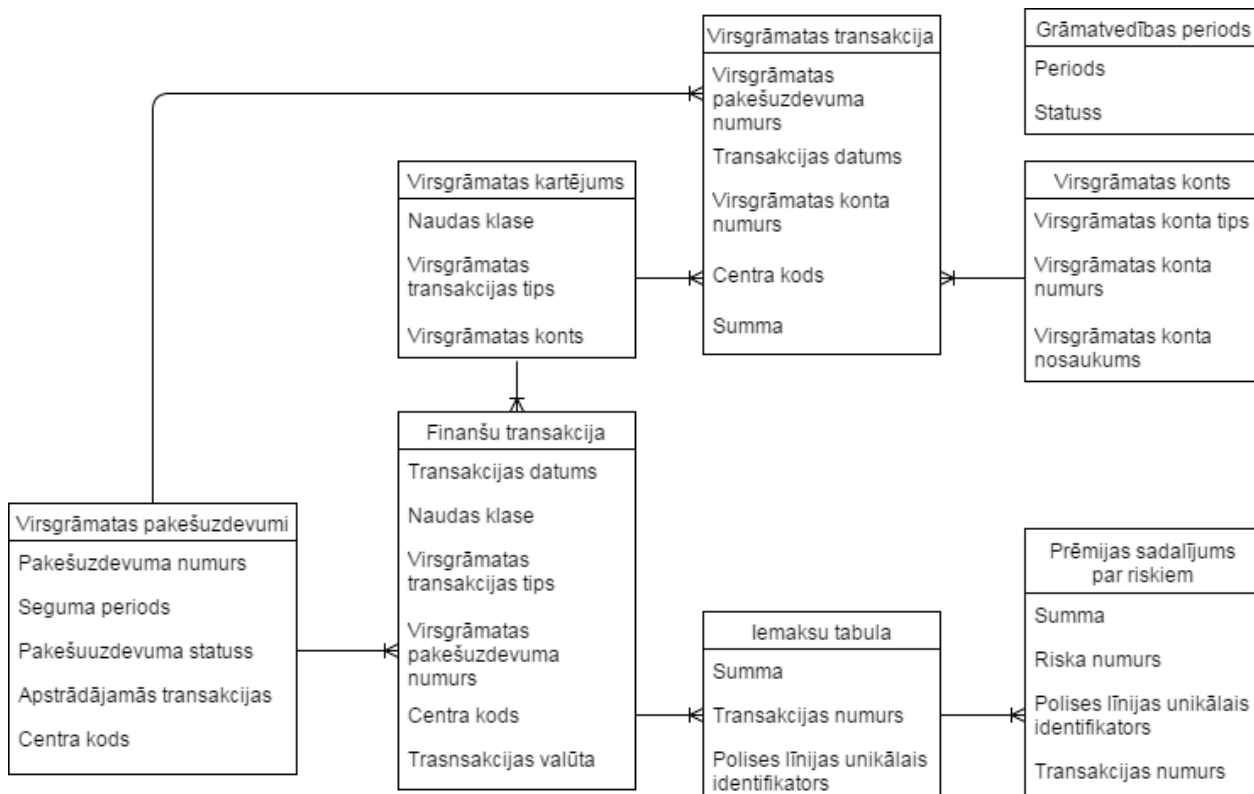
Apdrošināšanas sistēma TIA nepilda grāmatvedības sistēmas lomu, bet TIA kodolā ir realizēta funkcionalitāte virsgrāmatas datu sagatavošanai un eksportēšanai uz arēju grāmatvedības sistēmu tālākai apstrādei. Virsgrāmatas dati satur informāciju par pilnīgi visām finanšu transakcijām, kas ir notikušas sistēmā TIA, neatkarīgi no tā, kurā modulī šīs transakcijas ir radītas. Sistēmā TIA ir izdalītas divu veidu virsgrāmatas transakcijas:

- Ikdienas grāmatojumi;
- Grāmatvedības perioda beigu grāmatojumi.

Ikdienas grāmatojumi ir transakcijas, kas ir saistītas ar konkrētām darbībām sistēmā TIA, piemēram, jaunu apdrošināšanas polišu pārdošana un atlīdzību pieteikumi jeb transakcijas, kurās tiek iekasēta prēmija no klienta vai arī tiek izveidotas atlīdzības klientiem, vai šo atlīdzību aplēses. Šādas transakcijas tiek grāmatotas katru dienu. Grāmatvedības perioda beigu grāmatojumi ir transakcijas, kuras tiek grāmatotas periodiski, sekojot lietotāja definētiem grāmatojuma periodiem, kas parasti sakrīt ar kalendārajiem mēnešiem. Šo transakciju piemēri ir nopelnītās prēmijas aprēķins un prēmijas rezervju aprēķins. Šādas transakcijas tiek aprēķinātas pirms grāmatvedības perioda slēgšanas.

Sistēmas TIA kodolā ir definēti vairāki grāmatojumu tipi, kurus klients var izmantot vai neizmantojot atkarībā no individuālajām klienta vajadzībām. To, kādas ir virsgrāmatas transakcijas un kādā veidā tās grāmatot, var ietekmēt konkrētās kompānijas grāmatvedības noteikumi un juridiskās prasības. Šī iemesla dēļ sistēmā TIA ir iespējams ne tikai definēt, kuras no noklusējuma virsgrāmatas transakcijām grāmatot un kuras nē, bet arī definēt jaunas transakcijas un transakciju tipus. Lai varētu apstrādāt šīs jaunās virsgrāmatas transakcijas un transakcijas tipus, sistēmā ir iestrādāta lietotāja izeja, kurā var izveidot jaunus SQL vaicājumus, ar kuriem apstrādāt specifiskam

klientam nepieciešamās transakcijas. Šādā veidā var pielāgot arī to, kā sistēma apstrādā noklusētos virsgrāmatas transakcijas tipus.



2.1. att. Sistēmas TIA virsgrāmatas apakšmoduļa entītiju relācijas diagramma

Attēlā 2.1. ir redzama apdrošināšanas sistēmas TIA grāmatvedības moduļa virsgrāmatas apakšmoduļa entītiju relācijas diagramma. Šī diagramma ir vienkāršota un satur tikai galvenos virsgrāmatas apakšmoduļa tabulu laukus. Šī apakšmoduļa mērķis ir izveidot ierakstus virsgrāmatas transakciju tabulā. Dati šajā tabulā tiek izmantoti par pamatu virsgrāmatas eksportam uz ārēju grāmatvedības sistēmu, kura veic šo datu tālāku apstrādi.

Virsgrāmatas transakciju tabulā tiek grāmatotas finanšu transakcijas, bet atkarībā no sistēmas iestatījumiem vienai finanšu transakcijai var būt vairāk par vienu grāmatojumu. Grāmatojumus ir iespējams radīt par katru risku, ko viena finanšu transakcija sedz. Lai izveidotu grāmatojumus par katru risku, ko sedz viena finanšu transakcija, tiek izmantota tabula, kas glabā prēmijas sadalījumu par riskiem. Lai iegūtu korektu informāciju no šīs tabulas, tiek izmantota iemaksu tabula, kas glabā informāciju par finanšu transakcijas prēmijas sadalījumu par polises līnijām vienā polisē.

Informācija par katru izpildīto virsgrāmatas pakešuzdevumu tiek glabāta atsevišķā tabulā. Šeit tiek glabāta informācija gan par ikdienas virsgrāmatas grāmatojumu pakešuzdevumiem, gan periodiskajiem virsgrāmatas pakešuzdevumiem. Tabulā ir informācija par katra pakešuzdevuma

seguma periodu, pakešuzdevuma izpildes statusu, apstrādājamajiem virsgrāmatas transakciju tiptiem un kādam centra kodam pakešuzdevums ir izpildīts.

Virsrāmatas transakcijas tiek grāmatotas dažādos virsgrāmatas kontos. Šie konti tiek noteikti izmantojot parametrus, kas ir atrodami uz katras transakcijas ieraksta datu bāzē. Galvenie parametri ir naudas klase un virsgrāmatas transakcijas tips. Šie parametri uz transakcijas tiek uzstādīti tās izveidošanas brīdī un to vērtībām ir jāeksistē attiecīgās atsauču tabulās. Papildu šiem diviem parametriem ir iespējams definēt vēl desmit parametrus, pēc kuriem izdalīt virsgrāmatu transakcijas. Ja tiek definēti papildu parametri, tad katram virsgrāmatas kontam ir jānorāda, vai šis parametrs ir obligāts, neobligāts vai tiek ignorēts. To, kādā kontā dotajam grāmatojumam ir jānonāk, sistēma nosaka pēc kartējuma tabulas, kurā kā parametri tiek izmantoti augstāk minētā naudas klase un virsgrāmatas transakcijas tips. Papildu kartēšanas tabulas izmantošanai, ja nepieciešams, virsgrāmatas konta izvēli ir iespējams ietekmēt izmantojot sistēmā iestrādātu lietotāja funkciju.

Gan ikdienas grāmatojumi, gan grāmatvedības perioda beigu grāmatojumi tiek iegūti balstoties uz virsgrāmatas skatiem. Kurus no noklusētajiem grāmatojumu tiptiem izmantot un kurus neizmantot, ir iespējams norādīt izmantojot lietotāja funkciju, kurā tiek atrasta katras virsgrāmatas transakcijas pretējā transakcija. Ja funkcija neatrod transakcijas tipu transakcijai, kuras vaicājums tiek apstrādāts skatā, tad netiek atgriezts neviens ieraksts, kuram ir norādīts dotais transakcijas tips. Lai varētu apstrādāt nestandarta virsgrāmatas transakciju tipus, katram no šiem skatiem ir lietotāja izeja, kurā ir iespējams definēt jaunus vaicājumus, ar kuriem apstrādāt šos jaunus transakciju tipus. Visām transakcijām, kas tiek iegrāmatotas sistēmas TIA virsgrāmatā, tiek veikts pretējs grāmatojums. Tas tiek darīts, lai sistēmas grāmatojumi atbilstu grāmatvedības prasībām un tajos būtu redzams katrai transakcijai atbilstošais kredīta un debeta grāmatojums. Ja grāmatošanas procesā izveidotais debets un kredīts nesakrīt un pārsniedz pieļaujamo apaļošanas kļūdu, tad sistēma šo darbu atceļ un tiek atgriezts kļūdas paziņojums, lai šie kļūdainie grāmatojumi netiek nosūtīti tālākai apstrādei. Apaļošanas kļūdu katrs klients var norādīt pēc saviem ieskatiem.

2.2. Biznesa problēma

Sākotnēji klientam tika piegādāts risinājums, kas izmanto klientam pielāgotus TIA kodola SQL vaicājumus virsgrāmatas datu atlasei, bet aprēķinu veikšanai tika izmantotas TIA kodola funkcijas. Šāds risinājums grāmatvedības perioda beigu grāmatojumiem tiek izmantots, lai

aprēķinātu prēmijas rezervju izmaiņas pēdējā grāmatvedības periodā kompānijas un kompānijas filiāļu piešķirto atlaižu izmaiņu pēdējā grāmatvedības periodā.

Sākotnēji šis risinājums atbilda klienta prasībām un strādāja veiksmīgi, bet ar laiku, pieaugot datu apjomam sistēmā, šis risinājums sāka strādāt pārāk lēni. Piegadātais risinājums šos aprēķinus sāka veikt ilgāk kā 16 stundu laikā, līdz ar to sagādājot problēmas klientam. Problēmu iemesls ir tas, ka pēc pakešuzdevuma, kas veic grāmatvedības perioda beigu grāmatojumus, sāk strādāt pakešuzdevums, kas sagatavo maksājumu datnes bankām. Šīs datnes bankām ir jāgatavo un jānosūta līdz darba dienas beigām, lai bankas tās varētu veiksmīgi apstrādāt, kas nozīmē to, ka tās ir jāgatavo līdz pulksten 16:00. Grāmatvedības perioda beigu process ir pirmais process, kas tiek palaists pēc tam, kad grāmatvedības periods ir beidzies. Šo procesu nevar sākt pirms perioda beigām, jo tas noslēdz grāmatvedības periodu, kas nozīmē to, ka pēc procesa palaišanas, šajā periodā vairs nav iespējams izveidot jaunas finanšu transakcijas. Šī iemesla dēļ 16 stundas grāmatojumu veikšanai ir pārāk liels laika periods un šādā gadījumā maksājumu failus, kas sagatavoti par dienu, kad notiek grāmatojumi, nav iespējams nosūtīt bankai. Tā kā ir nepieciešams, lai transakcijas datumi šajos failos sakristu ar dienu, kad tie ir apstrādāti bankā, tad šie sagatavotie faili ir nederīgi un tie ir vai nu manuāli jālabo, vai arī jāgatavo no jauna nākamajā dienā. Šī iemesla dēļ tika nolemts, ka ir jāoptimizē grāmatvedības perioda beigu grāmatojumu process.

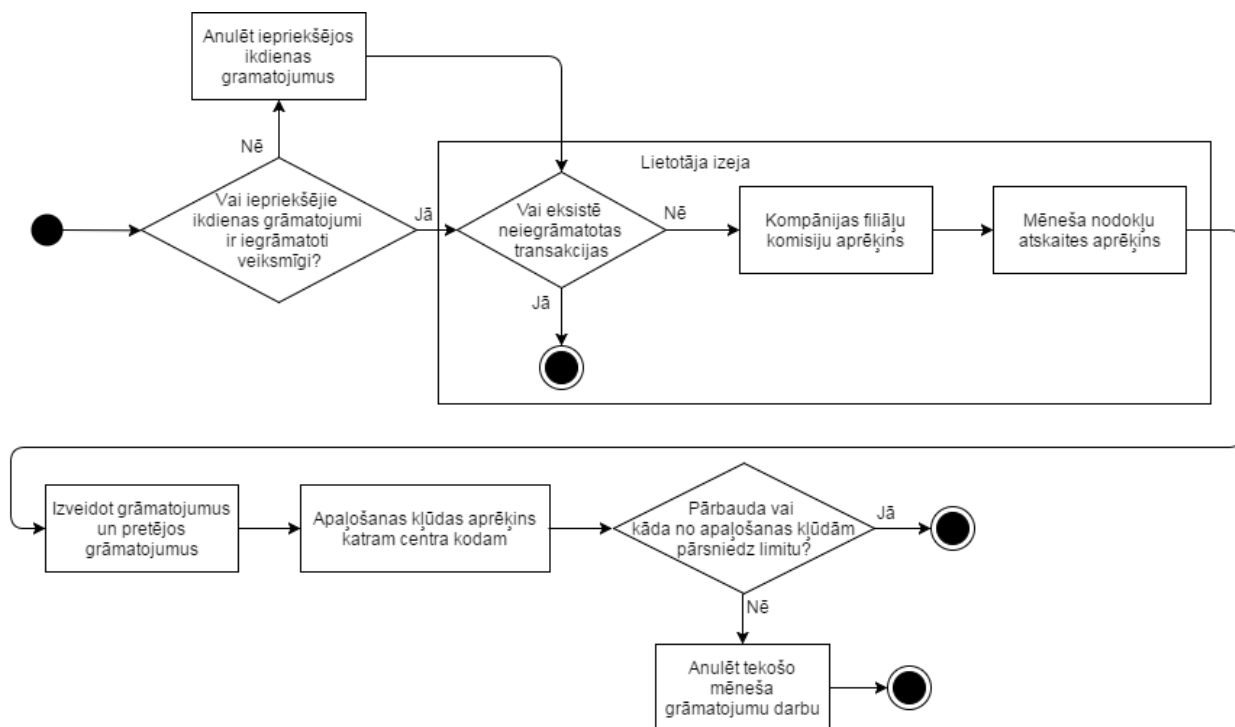
2.3. Problēmas analīze

Kā jau iepriekš minēts, tad sistēma TIA ir realizēta Oracle vidē, kas nozīmē to, ka apskatāmais risinājums ir izstrādāts Oracle PL/SQL programmēšanas valodā. Oracle ieteikumi par to, kas var palēnināt PL/SQL programmu darbību ir sekojoši:

- “Veci koda fragmenti, kas neizmanto jaunākas PL/SQL valodas iespējas;
- Veci dinamiski SQL vaicājumi, kas ir rakstīti izmantojot DBMS_SQL pakotni;
- Kods, kas daudz laika patērē apstrādājot SQL vaicājumus;
- Funkcijas, kuras tiek izsauktas SQL vaicājumos un var tikt izpildītas miljoniem reižu;
- Kods, kas patērē daudz laika apstrādājot SQL vaicājumu rezultātus;
- Kods, kas veica daudz matemātisku darbību;
- Kods, kas patērē daudz laika apstrādājot PL/SQL komandas [1].”

Paturot prātā šos faktorus, tiks veikta koda statistiska analīze, lai redzētu, vai problēma ilgai procedūras izpildei ir PL/SQL kodā. Lai varētu veiksmīgi aprakstīt risinājuma analīzi, tika

uzzīmēta grāmatvedības perioda beigu grāmatojumu procesa aktivitāšu diagramma. Tas ir nepieciešams, lai šajā darbā būtu punkti, uz kuriem atsaukties, veicot koda statisko analīzi. Ar aktivitāšu diagrammas palīdzību grāmatvedības perioda beigu grāmatojumu procesu var sadalīt mazākās daļās, kuru katru ir iespējams analizēt, šādā veidā atvieglojot analīzes procesu. Aktivitāšu diagramma parāda ko un kādā secībā dara grāmatvedības perioda beigu grāmatojumu process. Grāmatvedības perioda beigu grāmatojumu procesa aktivitāšu diagramma ir redzama attēlā 2.2.



2.2. att. Grāmatvedības perioda beigu grāmatojumu procesa aktivitāšu diagramma

Pirmais solis, ko grāmatvedības perioda beigu grāmatojumu process veic, ir pārbaudīt, vai pēdējais izpildītais ikdienas grāmatojumu darbs ir izpildīts veiksmīgi. Tā ir standarta darbība, kas tiek veikta, lai dzēstu grāmatojumus, ko ir izveidojis iepriekšējais grāmatojumu pakešuzdevums gadījumos, kad tas ir beidzies ar kļūdu, kuras dēļ tas nav spējis pats anulēt savus rezultātus. Tas tiek darīts tāpēc, lai pārlicinātos, ka sistēmā nav kļūdaini iegrāmatotu transakciju. Tā kā šī ir vienkārša operācija un tiek veikta TIA kodolā, tad šeit optimizācija nav nepieciešama.

Pēc tam, kad ir veikta iepriekšējā pakešuzdevuma pārbaude, tiek izsaukta lietotāja funkcija, kas ļauj izstrādātājiem veikt klientam specifiskas darbības, kuras nepieciešams veikt pirms tiek veikti grāmatvedības perioda beigu grāmatojumi. Tiek pārbaudīts, vai norādītajā grāmatvedības periodā neeksistē neiegrāmatotas transakcijas. Ja šādas transakcijas eksistē, tad grāmatvedības perioda beigu grāmatošanas process tiek pārtraukts. Lai veiktu šo pārbaudi, tiek izsaukts SQL vaicājums, kurš meklē, vai starp grāmatvedības perioda beigām un sākumu eksistē kāda

transakcija, kura nav iegrāmatota. Kad transakcija ir iegrāmatota, tad uz tās tiek saglabāts grāmatojuma darba numurs, kurš šo transakciju ir iegrāmatojis. Noklusēti šis numurs ir 0, tāpēc to, vai transakcija ir iegrāmatota, var noteikt pēc šī lauka vērtības. Tā kā šis lauks ir indeksēts, tad šis vaicājums ir efektīvs un tiek izpildīts salīdzinoši īsā laika periodā. Arī šeit optimizācija nav nepieciešama.

Tagad lietotāja izejā tiek izpildīta kompānijas filiāļu komisiju aprēķina funkcionalitāte. Šī funkcionalitāte ir klientam specifisks risinājums, kurš aprēķina komisijas apmēru, kas apdrošināšanas kompānijai ir jāizmaksā savām filiālēm par pārdotajiem kompānijas apdrošinātajiem riskiem. Tās pamatā ir komplicēts SQL vaicājums, kas pēc klienta noteiktas loģikas aprēķina šo komisiju apmērus katrai transakcijai un iegūtos rezultātus saglabā tabulā tālākai apstrādei. Tā kā šis vaicājums apstrādā tikai nelielu daļu no datu bāzē saglabātajām transakcijām, tad šis process tiek izpildīts aptuveni 10 minūtēs. Lai arī 10 minūtes ir salīdzinoši ilgs laiks, tas tomēr ir tikai neliela daļa no 16 stundām, ko aizņem šis process. Tas nozīmē to, ka var rasties nepieciešamība šo procesu optimizēt nākotnē, bet šīs problēmas kontekstā iespējamais ieguvums ir tik neliels, ka nav jēgas mēģināt šo procesu paātrināt.

Pēdējais process, kas tiek veikts lietotāja funkcijā ir mēneša nodokļu atskaites aprēķins. Arī šis ir klientam specifisks risinājums, kas aprēķina nodokļu apjomu, ko klients ir parādā valstij par pagājušo mēnesi. Šis risinājums balstās uz vairākiem SQL vaicājumiem, katrs no kuriem veic viena nodokļa aprēķinu, piemēram, pievienotās vērtības nodokļa un satiksmes nodokļa aprēķinus. Šie SQL vaicājumi ir salīdzinoši vienkārši un tie lieto datu apkopošanas funkcijas, nodokļu apjoma aprēķināšanai. Lai arī datu apkopošanas funkcijas var prasīt daudz laika to izpildei, šie vaicājumi savu darbu pabeidz aptuveni 5 minūšu laikā, tāpēc, līdzīgi kā filiāļu komisiju aprēķinu procesam, šī procesa optimizācija kopumā ir maznozīmīga.

Pēc tam, kad programma iziet no iepriekš izsauktās lietotāja funkcijas, tiek sākts rezervju grāmatošanas process. Šī ir galvenā darbība, kas šajā procesā tiek veikta, un tieši tā arī aizņem visvairāk laika un resursu. Šo procesu veic izmantojot SQL skatu, kurš sastāv no vairākiem sarežģītiem SQL vaicājumiem, katrs no kuriem apstrādā vienu virsgrāmatas transakcijas tipu. Tā kā vienā mēnesī tiek izveidotas ļoti daudz transakcijas katram virsgrāmatas transakciju tipam, tad katrs vaicājums skatā apstrādā daudz rezultātu. Lai arī tiek izpildīti tikai 4 vaicājumi un to izpilde notiek 4 paralēlos darbos, izmantojot Oracle AQ, kas nozīmē, ka katru transakcijas tipu apstrādā viens process, šis darbs vienlīdz ilgst aptuveni 14 stundas. Tas nozīmē to, ka tieši šis process

visvairāk ietekmē darba izpildes laiku un tam ir jāpievērš lielākā uzmanība optimizējot grāmatvedības perioda beigu grāmatojumu procesu.

Pēc tam, kad grāmatojumi ir izveidoti, tiek veiktas pēdējās pārbaudes pirms šis process tiek noslēgts. Šajā brīdī tiek aprēķināta apaļošanas kļūda katram centra kodam un tiek pārbaudīts, vai šī kļūda pārsniedz apaļošanas kļūdas ierobežojumu. Ja kļūda nepārsniedz šo ierobežojumu, tiek izveidots jauns grāmatojums kļūdas izlīdzināšanai, pretējā gadījumā darbs tiek anulēts un pakešuzdevums beidzas ar kļūdu. Apaļošanas kļūdas aprēķins notiek izmantojot vienkāršu SQL vaicājumu, kurš vēršas pie grāmatojumu tabulas izmantojot virsgrāmatas pakešuzdevuma numuru, kurš ir indeksēts, un summē kredīta un debeta apmērus grupējot tos pēc centra koda. Centra kods ir lauks, kurš reprezentē kompānijas filiāles, tāpēc šajā brīdī tiek pārbaudīts, vai rezerves sakrīt katrai kompānijas filiālei.

Pēc šīs risinājuma analīzes var secināt, ka, optimizējot šo procesu, galvenā uzmanība ir jāpievērš tieši galvenajai grāmatošanas darbībai, kas aizņem lielāko šī procesa darbības laiku. Pēc šī procesa optimizācijas ir jāizvērtē optimizācijas rezultāti un jālemj par tālāko darbību.

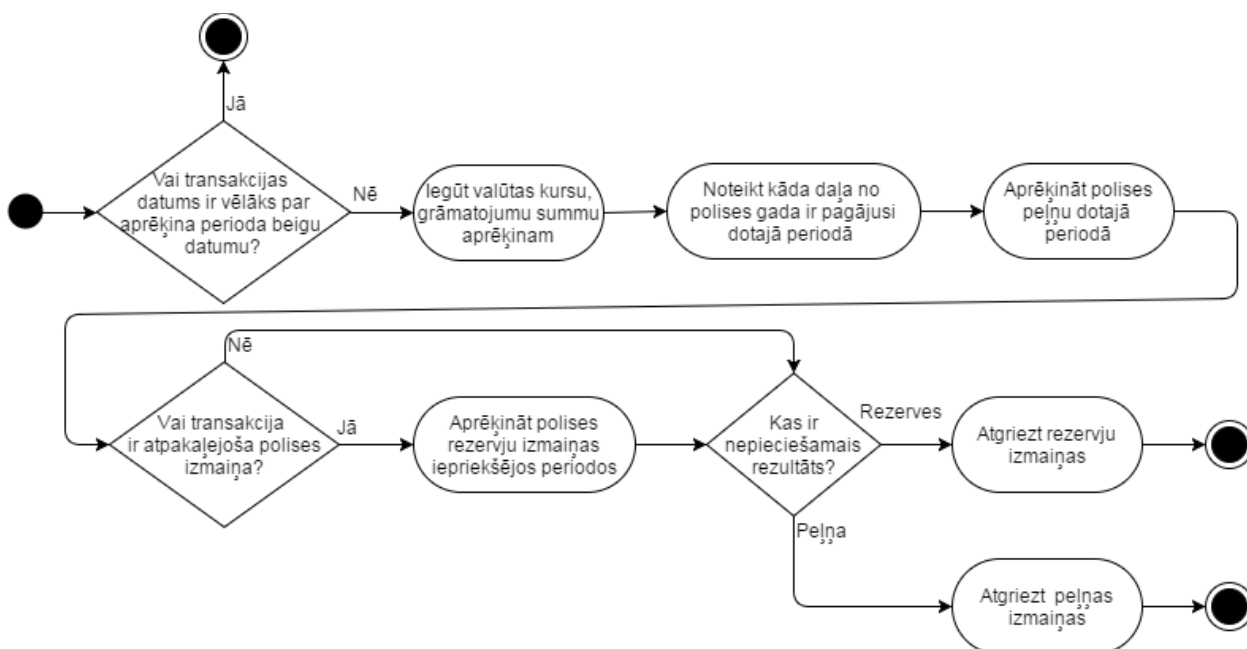
2.4. Izvēlētais risinājums

Optimizācija tiks veikta galvenajam grāmatvedības perioda beigu grāmatojumu skatam. Tas nozīmē to, ka no Oracle ieteikumiem PL/SQL koda optimizācijai var atņemt vecus koda fragmentus, vecus dinamiskus SQL vaicājumus un kodu, kas ilgi apstrādā PL/SQL darbības, jo šajā gadījumā darbs notiek ar SQL skatu. Šajā gadījumā ir skaidrs, ka ilgās apstrādes iemesls ir kods, kas ilgi apstrādā SQL vaicājumu vai arī kods ilgi apstrādā SQL vaicājuma rezultātus. Tas nozīmē, ka uzmanība ir jāpievērš tam, kā optimizēt SQL vaicājumu.

No sākuma aplūkosim, ko sistēma dara ar vaicājuma atgrieztajiem rezultātiem. Kā jau iepriekš minēts, tad tieši pats SQL skats veic datu apstrādi. Šajā gadījumā skata atgrieztie dati tiek ievietoti tabulā, no kuras šie dati tiek izmantoti tālākai apstrādei. Šī darbība ir realizēta ar vienkāršu INSERT INTO komandu, kurai kā datu avots tiek piegādāts SQL vaicājums, kurš atgriež transakcijas no rezervju jeb grāmatvedības perioda beigu grāmatojumu skata un savieno šo skatu ar tabulu, kurā glabājas virsgrāmatas konti. Dati šajā SQL vaicājumā tiek atlasīti pēc grāmatojumu pakešuzdevuma numura, kurš skatā tiek izmantots, lai noskaidrotu, kādas transakcijas ir jāatgriež, un centra koda, ja tas ir norādīts. Šajā gadījumā klientam nav vajadzības norādīt centra kodu, tāpēc tas netiek norādīts. Skats tiek savienots ar virsgrāmatas kontu tabulu, lai nodrošinātu to, ka

transakcijas netiek iegrāmatotas neeksistējošos virsgrāmatas kontos. Tas nozīmē, ka šajā gadījumā ilga datu apstrāde nav iemesls procedūras ilgajai izpildei.

SQL vaicājumi, kuri tiek izpildīti rezervju skatā, izsauc trīs funkcijas. Viena funkcija veic vienas transakcijas radītās rezervju un peļņas izmaiņas aprēķinu. Otra funkcija atgriež pamata valūtu kompānijas filiālei pēc tai padotā filiāles centra koda. Trešā funkcija tiek izmantota, lai noskaidrotu virsgrāmatas kontu, kurā dotā transakcija ir jāiegrāmato. Šīs trīs funkcijas ir jāanalizē un jānoskaidro, vai tās varētu palēnināt SQL vaicājuma darbību. Tā kā šīs funkcijas tiek izsauktas katrai apstrādājamajai transakcijai, tad nelieli uzlabojumi šo funkciju darbībā var radīt lielas izmaiņas SQL vaicājuma kopējā apstrādes laikā.



2.3. att. Funkcijas transakcijas rezervju un peļņas izmaiņu aprēķināšanai aktivitāšu diagramma

Transakcijas rezervju un peļņas izmaiņu aprēķināšanas funkcijas aktivitāšu diagramma ir redzama attēlā 2.3. Kā redzams šī funkcija apstrādā tikai transakcijas, kuras ir radītas pirms grāmatvedības perioda beigām, jo, ja šī transakcija ir radīta vēlāk, tad šī transakcija nav radījusi peļņas vai rezervju izmaiņu dotajā periodā. Lai iegūtu valūtas kursu grāmatojumu, aprēķiniem tiek izmantoti divi SQL vaicājumi, kuri iegūst nepieciešamo valūtas kursa tipu un dotās filiāles pamata kursu. Ja transakcijas valūta nesakrīt ar grāmatojumiem nepieciešamo valūtu, tad valūtu pārveidošanas aprēķinu veikšanai tiek iegūts aktuālais valūtas kurss.. Prēmijas rezervju un peļņas izmaiņu aprēķiniem ir nepieciešams iegūt polises seguma perioda daļu, kas ir pagājusi dotajā grāmatvedības periodā. Lai to aprēķinātu, tiek izsaukta funkcija, kura, izmantojot grāmatvedības perioda sākuma datumu un beigu datumu, kā arī transakcijas seguma perioda sākuma un beigu

datumus, iegūst procentuālu daļu no polises seguma perioda, kas ir pagājis iepriekšējā grāmatvedības periodā. Lai aprēķinātu peļņu, kas ir nopelnīta iepriekšējā grāmatvedības periodā, valūtas kurss un seguma perioda daļas procents tiek sareizināts ar transakcijas summu, kas ir šīs funkcijas parametrs. Ja transakcija ir izveidota ar seguma sākuma periodu, kas ir pirms noslēdzamā grāmatvedības perioda sākuma perioda, tad peļņas izmaiņa tiek aprēķināta arī viesiem iepriekšējiem periodiem, kurus transakcija sedz. Pēdējā darbība, kas funkcijā tiek veikta, ir rezervju izmaiņas aprēķins, ja tas ir nepieciešams. Tā nepieciešamību nosaka pēc parametra, kas funkcijai tiek padots no SQL vaicājuma, kas to izsauc. Gadījumā, ja transakcijas izveides datums ietilpst dotajā grāmatvedības periodā, tad rezerves tiek aprēķinātas no jauna, no kopējā prēmijas apjoma atņemot jau nopelnīto prēmiju. Visos citos gadījumos prēmijas rezervju izmaiņa ir tāda pati kā nopelnītā prēmija, bet ar pretēju zīmi. Funkcija atgriež nepieciešamo vērtību atkarībā no šī paša parametra.

Virsrāmātas konta pamata valūtas iegūšanas funkcija ir vienkārša funkcija, kas, izmantojot filiāles centra kodu, iegūst valūtu, kādā tiks veikts grāmatojums. Šī funkcionalitāte ir nepieciešama gadījumos, kad klienta kompānija ievieš vienu sistēmu vairākās valstīs ar dažādām valūtām vai arī dažādas valūtas sistēmā tiek izmantotas citu iemeslu dēļ.

Funkcija virsrāmātas grāmatojumu konta noteikšanai ir lietotāja funkcija, kas ļauj izstrādātājiem ieviest lietotājam specifisku loģiku, kuru izmantojot izvēlēties transakcijas virsrāmātas kontu. Funkcija kā parametrus saņem transakcijas naudas klasi, virsrāmātas transakcijas tipu, transakcijas datums, apdrošināšanas polises līnijas unikālo identifikatoru, kurai transakcija tika radīta, karogu, kurš nosaka, vai šai transakcijai ir jāeksistē virsrāmātas kontam, filiāles centra kodu un transakcijas numuru. Ja nepieciešams, izmantojot šo funkcionalitāti, ir iespējams iegūt papildu informāciju par transakciju, kas var būt nepieciešama, lai noteiktu, kurā virsrāmātas kontā grāmatot transakciju. Konkrētā klienta gadījumā funkcijā tiek apstrādātas kompānijas filiāļu piešķirtās atlaides apdrošināšanas polisēm, kuras sedz šīs filiāles, tāpēc tās tiek apstrādātas atsevišķi. Papildu tam visām transakcijām ar vienkāršām loģiskām darbībām tiek noteikts nepieciešamais virsrāmātas transakcijas tips un izmantojot kartējumu tabulu tiek atrasts attiecīgais virsrāmātas konts. Šajā funkcijā optimizācija nav nepieciešama, jo vairumā gadījumu tiek izpildītas tikai loģiskās darbības un tiek izsaukts viens vienkāršs SQL vaicājums, lai atrastu nepieciešamo grāmatojumu kontu.

Pēc SQL vaicājumu izpētes var secināt, ka funkcijas, kuras tiek izsauktas no šiem vaicājumiem, ir iespējams optimizēt un ka ar šo optimizāciju var pietikt, lai grāmatvedības perioda

beigu grāmatojumi tiktu izpildīti klientam pieņemamā laikā. Papildu tam var veikt pašu SQL vaicājumu optimizāciju, lai šie vaicājumi pavadītu īsāku laiku atlasot datus. Optimizācija tiks sākota ar funkciju optimizāciju un vēlāk uzmanība tiks pievērsta potenciālajiem SQL vaicājumu uzlabojumiem.

Pirmā funkcija, kas tiks optimizēta, ir funkcija prēmijas rezervju un peļņas aprēķiniem. Tā kā šī ir TIA kodola funkcija un to tiešā veidā modificēt nevar, tad, lai to pielāgotu, tika izveidota jauna identiska funkcija, kura tiks pielāgota un izsaukta no SQL vaicājumiem. Šīs funkcijas optimizācija ir ļoti vienkārša. Tā kā polises seguma perioda daļas aprēķins un peļņas aprēķins ir pietiekoši vienkāršas darbības un neizdara vaicājumus datu bāzē, tad var uzskatīt, ka šīs darbība optimizēt nav nepieciešams, jo parasti visvairāk laika PL/SQL procedūrās aizņem tieši datu iegūšana no datubāzes. Šī iemesla dēļ, funkcijā tiks cieti iekodēta valūta kādu izmantot grāmatojumu veikšanai. To var darīt tāpēc, ka klients sistēmā TIA izmanto tikai vienu valūtu. Šī optimizācija ir svarīga tāpēc, ka valūtas noteikšanai tiek izmantoti divi SQL vaicājumi – viens, lai noteiktu valūtas kursa tipu, un otrs, lai iegūtu valūtas kursu. Tā vietā tiek cieti iekodēts valūtas kursa tips un valūta, kuri tālāk tiek izmantoti, lai veiktu rezervju un peļņas aprēķinu. Lai arī šī optimizācija šķiet minimāla, tomēr, tā kā šī funkcija tiek izsaukta divas reizes katram rezultātam un vaicājums katru reizi apstrādā vairākus tūkstošus rezultātu, tad arī šādas nelielas optimizācijas var radīt jūtamu uzlabojumu procesa darbības laikā.

Kā jau iepriekš minēts, klients, kuram šī optimizācija tiek veikta, sistēmā izmanto tikai vienu valūtu. Šī iemesla dēļ grāmatvedības perioda beigu grāmatojumu skatā jau uzreiz cieti nokodē valūtu, kādā grāmatojumi tiek iegūti. Šādā veidā var izlaist vēl vienu lieku SQL vaicājumu, kas šī procesa laikā tiek izpildīts katram atgrieztajam rezultātam, jo parasti šī informācija tiek iegūta no sistēmas atsauču tabulām. Arī šāds neliels uzlabojums var radīt jūtamu uzlabojumu procesa darbības laikā, jo tas izslēdz tūkstošiem SQL vaicājumu, kuri tiek izpildīti procesā.

Pēdējā funkcija, kas tiek izsaukta no grāmatvedības perioda beigu grāmatojumu skata, ir lietotāja funkcija, kura nosaka grāmatvedības kontu, kurā grāmatojums ir jāieraksta. Lai arī šajā funkcijā koda optimizācija nav nepieciešama, tās darbības ātrumu var uzlabot pārveidojot to par deterministisku funkciju. To var izdarīt funkcijas definīcijai pievienojot atslēgas vārdu *deterministic*. Tas nozīmē to, ka funkcijas atgrieztie rezultāti tiek saglabāti atmiņā un katru reizi, kad funkcija tiek izsaukta, tiek pārbaudīts, vai funkcija ar šādu parametru kopu jau ir izsaukta, un, ja tā ir, tad tiek atgriezts tas pats rezultāts, kas iepriekšējo reizi, kad funkcija tika izsaukta ar šiem parametriem. Šī iemesla dēļ šo atslēgas vārdu var izmantot tikai tad, ja funkcija ar vieniem un tiem

pašiem parametriem atgriež vienu un to pašu vērtību. Šī funkcija tāda ir. Papildu tam, lai funkcija katru reizi netiktu izsaukta ar unikālu parametru kopu, funkcijai netiek padots polises līnijas unikālais identifikators un transakcijas numurs, jo šie parametri funkcijā netiek izmantoti. Tā kā šis darbs tiek izpildīts paralēlās sesijā un katra no šīm sesijām apstrādā tikai vienu naudas klasi, tad palielinās iespēja, ka šī funkcija netiks izpildīta, bet uzreiz tiks atgriezts vajadzīgais konts. Šādā veidā tiek izlaisti papildu SQL vaicājumi virsgrāmatas konta noteikšanai, kas arī var nedaudz uzlabot SQL vaicājuma darbības laiku.

Tagad tiks optimizēts SQL vaicājums, kurš iegūst datus, kas tiek grāmatoti grāmatvedības perioda beigu grāmatojumu procesā. Šī SQL vaicājuma izpildes plānu var redzēt 1. pielikumā. Šajā plānā ir iekļauti tikai SQL vaicājumi, kuri patiešām tiek izpildīti. Izpildes plāna kolonna “Apraksts” satur informāciju par to, kā tiek iegūti dati un savienotas tabulas. “Kolonna “Objekta vārds” satur informāciju par to, kāds datu bāzes objekts tiek izmantots konkrētās darbības veikšanai. Kolonna “Procesora Izmaksas” satur informāciju par procesora paredzamo noslodzi, kas rodas izpildot šo informāciju un ir proporcionāla mašīnas cikliem, kas nepieciešami, lai izpildītu operāciju. Kolonna “Ievades/Izvides Izmaksas” satur informāciju par paredzamo ievades/izvades noslodzi, kas nepieciešama operācijas izpildei un ir proporcionāla datu bloku skaitam, kas ir jānolasa, lai izpildītu operāciju. Kolonna “Kardinalitāte” satur informāciju par paredzamo rindiņu skaitu, kurām tiks piekļūts, lai izpildītu operāciju. Kolonna “Baiti” ir paredzamais baitu skaits, kuriem tiks piekļūts, lai izpildītu operāciju [2].”

SQL vaicājuma izpildes plānā ir redzams, ka liela daļa no procesora noslodzes tiek radīta, lai veiktu iemaksu tabulas un prēmijas sadalījuma par riskiem tabulas pilnu pārlassi. Tas notiek tāpēc, ka SQL vaicājums, kurš tika piegādāts klientam, izmantoja pieeju, ko izmanto TIA kodols, tas ir, lai iegūtu datus par transakcijām, tika izmantots skats, kurš glabā šo informāciju. Diemžēl, šis skats veic tabulu pilnu pārlassi, tāpēc šāda pieeja nav labākais risinājums. Lai risinātu šo problēmu, no SQL vaicājumiem tika izņemts šis skats un tas tika aizvietots ar tiešu piekļuvi šīm abām tabulām. Šādā veidā pilnībā tiek pārlassīta tikai iemaksu tabula, un no tās uzreiz tiek atsijāti nevajadzīgie dati. Šī tabula tiek savienota ar prēmijas sadalījuma par riskiem tabulu izmantojot polises līnijas unikālo identifikatoru, kas ir indeksēts lauks, un tādējādi šis vaicājums tiek padarīts efektīvāks.

Pēc visu optimizācijas darbu veikšanas, var redzēt, ka SQL vaicājums tika padarīts daudz efektīvāks. SQL vaicājuma izpildes plānu pēc optimizācijas var redzēt 2. pielikumā. Salīdzinot izpildes plānus pirms un pēc optimizācijas var redzēt, ka SQL vaicājuma paredzamā efektivitāte ir uzlabota. Paredzamā procesora noslodze ir samazināta aptuveni 20 reīžu, paredzamā

ievades/izvades noslodze ir samazināta aptuveni 30 reižu, kā arī paredzamais nolasāmo datu apjoms ir samazinājies aptuveni 50000 reižu. Biznesam piegādātais jaunais risinājums mēneša beigu grāmatojums veic aptuveni 6 stundu laikā, kas ir klientam pieņemams laika periods un tas nozīmē to, ka grāmatojumi ir veikti līdz nākamās darba dienas sākumam.

2.5. Secinājumi

Pēc šo optimizācijas darbu veikšanas var secināt to, ka SQL vaicājumi, kas strādā ar lielu datu apjomu, ļoti reti strādās veiksmīgi, ja risinājums ir vispārīgs, lai derētu visiem iespējamajiem gadījumiem. Šajā gadījumā “TIA Technology” piegādātie risinājumi bija tik nepiemēroti klienta vajadzībām, ka pat to izmantošana par pamatu klientam nepieciešamās biznesa loģikas izveidei noveda pie risinājuma, kas strādāja pārāk ilgi, lai varētu tikt uzskatīts par lietojamu.

SQL vaicājumu izstrādē, izstrādātājam ir jābūt piesardzīgam attiecībā uz funkcijām, kādas tiek izmantotas vaicājumā. Šajā nodaļā aprakstītā situācija ir piemērs situācijai, kad lieki un nevajadzīgi funkciju izsaukumi, lai nodrošinātu sistēmas funkcionalitāti, kas klientam nav nepieciešama, padarīja sistēmas procesus klientam nederīgus. Šādā situācijā, ja tas ir iespējams, labāk ir izmantot cieti iekodētas vērtības nekā tūkstošiem reižu vērsties pie datu bāzes, lai iegūtu vienu un to pašu vērtību.

Strādājot ar lielām un sarežģītām sistēmām ir jāpārlicinās, vai gadījumā sistēmā nav iestrādāta funkcionalitāte, kas vispārīgajā gadījumā palīdz sistēmai funkcionēt, bet konkrētajā sistēmas implementācijā, šī funkcionalitāte var būt par traucēkli sistēmas veiksmīgai darbībai.

3. RISINĀJUMS AUTOMAŠĪNU PARKA APDROŠINĀŠANAS PRODUKTA MAKSĀJUMU APSTRĀDEI

Šajā nodaļā ir aprakstīts apdrošināšanas sistēmas TIA uzlabojums, kas pielāgo sistēmas maksājumu funkcionalitāti, lai tas atbilstu pasūtītāja prasībām automašīnu parka apdrošināšanas produkta maksājumu apstrādei. Šajā nodaļā ir atrodams TIA kodola risinājums maksājumu izveidei, biznesa prasības maksājumu apstrādei automašīnu parka apdrošināšanas produktam un šī risinājuma apraksts.

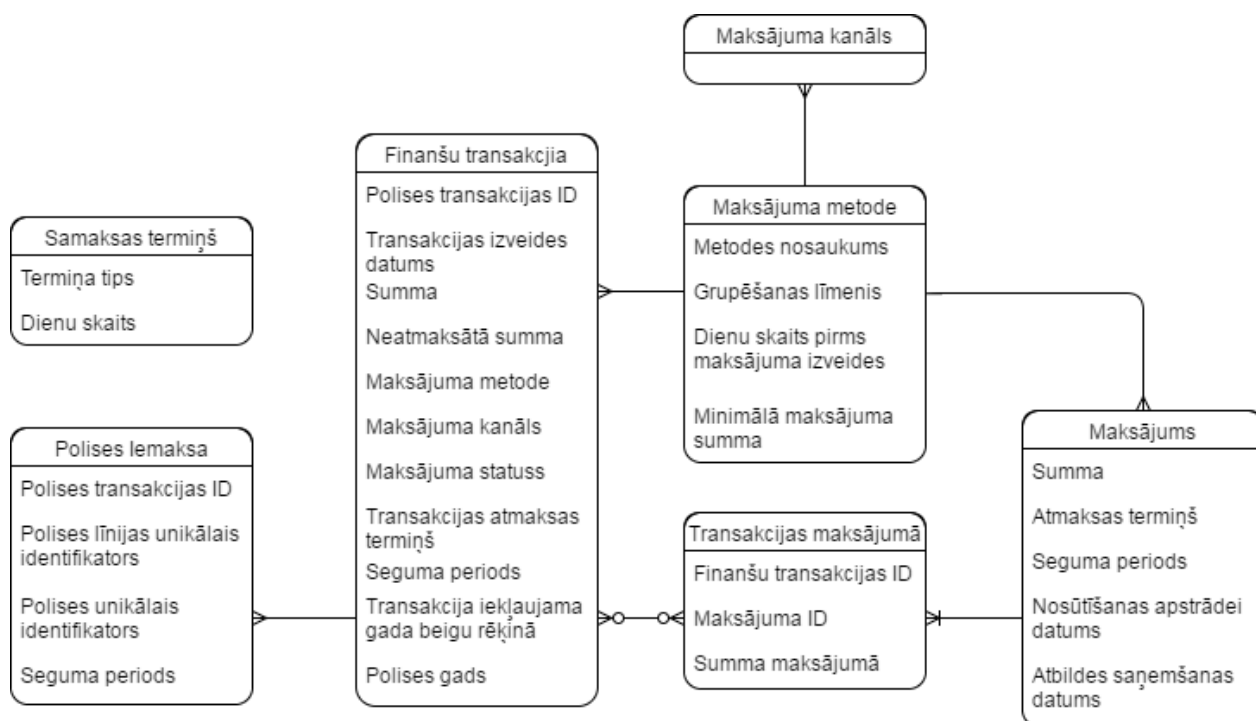
3.1. TIA kodola risinājums

Apdrošināšanas sistēmā TIA transakcijas var tikt radītas no dažādiem sistēmas moduļiem, piemēram, polises moduļa, kad tiek izveidota jauna vai labota eksistējoša polise, vai atlīdzību pieprasījumu pārvaldības moduļa, kad klientam tiek izmaksāta nauda par atlīdzības pieprasījumu, vai arī tiek atmaksāti citi saistīti pakalpojumi. Sistēmas TIA grāmatvedības modulis kalpo, kā no sistēmas izejošo un sistēma ienākošo maksājumu apstrādes modulis visiem sistēmas moduļiem, kuros tiek radītas finansiālas transakcijas, kurām ir jānokļūst pie ar kompāniju saistītajām pusēm. Grāmatvedības modulis veic visus ar šīm transakcijām saistītos uzdevumus, kā prēmiju ievākšanu, atlīdzību atmaksu un komisiju izmaksu, lai veiksmīgi īstenotu klientu apkalpošanu. Bet, kā jau iepriekš minēts, šis modulis nestrādā kā virsgrāmatas sistēma, tas tikai sagatavo datus eksportēšanai uz ārēju virsgrāmatas sistēmu.

Lai grāmatvedības modulis varētu veiksmīgi funkcionēt, sistēmā TIA ir jāveic moduļa sagatavošanas darbi. Starp šiem darbiem ir samaksu termiņu nosacījumu uzstādīšana. Apdrošināšanas sistēmā TIA ir iespējams definēt dažādus samaksu termiņus, piemēram, dienu skaits pēc transakcijas izveides, dienu skaits pēc nākamās nedēļas vai mēneša sākuma, vai noteikta diena nedēļā, mēnesī u.c. Ja aprēķinātais maksājuma termiņa beigu datums nav darba diena, tad maksājuma termiņš tiek noteikts nākamajā darba dienā pēc aprēķinātā datuma. Sistēmas lietojuma iestatījumos ir jānorāda noklusētos samaksas termiņu nosacījumus gan ienākošajiem maksājumiem, gan izejošajiem maksājumiem.

Sistēmā ir jādefinē arī maksājumu metodes, kas tiks izmantotas transakciju izmaksai. Maksājumu metode nosaka to, kā grupēt transakcijas ienākošajos un izejošajos maksājumos, piemēram, polises, transakcijas, atlīdzības pieprasījuma u.c. līmeņos. Uz maksājuma metodes ir

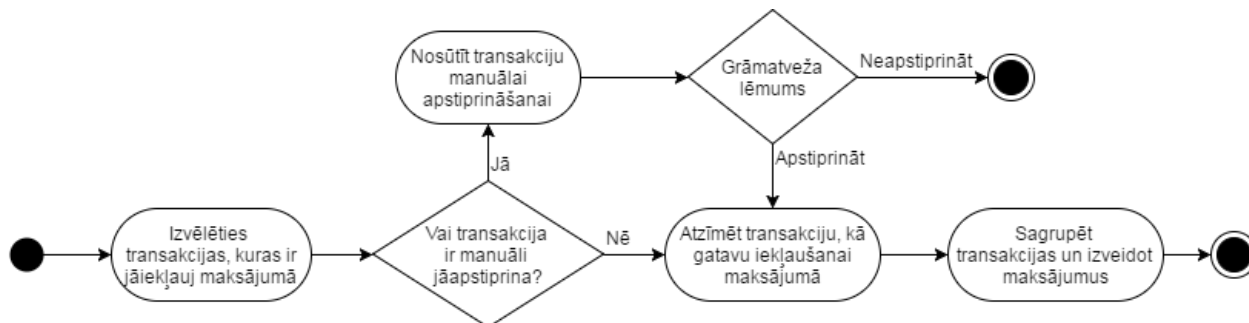
jānorāda dienu skaits pirms transakcijas apmaksas termiņa, kurā transakcija ir jāiekļauj maksājumā. Papildu šai informācijai formā var norādīt arī to vai maksājuma metodi var izmantot gan ienākošajiem maksājumiem, gan izejošajiem maksājumiem, vai maksājumi ir manuāli jāpastiprina, vai maksājuma metodei ir minimālā summa, lai maksājums tiktu izveidots, vai saņēmējam ir nepieciešamas maksājumam atbilstošas maksājuma detaļas u.c. Šeit var definēt noklusēto sistēmas uzvedību ar šo maksājumu metodi, kuru var pielāgot sistēmas biznesa pielāgojumu slānī. Papildu tam ir jādefinē maksājuma kanāli, kas identificē finanšu institūciju, kurai maksājums tiks nosūtīts, un veidu, kādā tas tiks nosūtīts.



3.1. att. Sistēmas TIA maksājumu apakšmoduļa entītiju relācijas diagramma

Attēlā 3.1. ir redzamas sistēmas TIA grāmatvedības moduļa maksājumu apakšmoduļa entītiju relācijas diagramma. Šī diagramma ir vienkāršota un tā satur tikai entītijas un to laukus, kas ir nozīmīgi šajā nodaļā aprakstītajam risinājumam. Galvenā entītija sistēmas maksājumu apakšmodulī ir finanšu transakciju entītija. Šajā tabulā glabājas pilnīgi visas apdrošināšanas sistēmā TIA izveidotās finanšu transakcijas. Uz finanšu transakcijas tiek glabāta informācijas par finanšu transakcijas seguma periodu, apmaksas statusu, kāds modulis šo finanšu transakciju ir radījis, informācija par sistēmas transakciju, kas ir izveidojusi doto finanšu transakciju un informāciju par finanšu transakcijas atmaksas termiņu. Finanšu transakcijas tiek izveidotas balstoties uz polises iemaksu tabulu, kura glabā prēmijas sadalījumu par polises līnijām, seguma periodiem un polises transakcijām

Maksājumu tabula glabā visus sistēmas maksājumus, kuri tiek nosūtīti klientiem. Maksājumā iekļautās finanšu transakcijas tiek glabātas atsevišķā tabulā, jo viena finanšu transakcija var būt iekļauta vairāk nekā vienā maksājumā un to apmaksas summa dažādos maksājumos var atšķirties. Maksājumu tabulā tiek glabāta informācija par maksājuma atmaksas metodi, tā seguma periodu un informāciju par to sedzošo ienākošo maksājumu. Atsevišķā tabulā ir definētas maksājumu metodes un maksājumu kanāli, kā arī informācija par tiem. Samaksas termiņi tiek definēti atsevišķās tabulās un tie tiešā veidā netiek saistīti ar citām sistēmas tabulām.



3.2. att. Apdrošināšanas sistēmas TIA maksājumu procesa aktivitāšu diagramma

Pēc tam, kad kāds no sistēma moduļiem ir radījis transakciju, uz kuras ir uzstādīta sistēmā definēta maksājuma metode un maksājuma kanāls, un kura ir jāiekļauj maksājumā, sākas maksājumu process. Attēlā 3.2. ir redzama augsta līmeņa maksājumu procesa aktivitāšu diagramma, kura parāda galvenos soļus, lai apdrošināšanas sistēma TIA varētu izveidot maksājumus. Šis process sākas tad, kad pienāk datums, kurā transakcija ir jāiekļauj maksājumā. Šo datumu nosaka izmantojot samaksu termiņu nosacījumu, kas norādīts uz klienta konta vai, ja tas nav norādīts, izmanto sistēmas noklusēto samaksu termiņa nosacījumu. Pirmais solis maksājuma procesā ir izvēlēties transakcijas, kuras ir jāiekļauj maksājumā. Maksājumā var iekļaut tikai atvērtas transakcijas jeb transakcijas, uz kurām apmaksai atlikusī summa nav nulle un kuras šobrīd nav iekļautas nevienā maksājumā. Šajā brīdī sistēmā iestrādāta lietotāja izeja, kurā izstrādātāji var realizēt papildu biznesa noteikumus, pēc kuriem izvēlētas transakcijas var vēl neiekļaut maksājumos, nosūtīt transakciju manuālai apstiprināšanai vai arī atzīmēt transakciju kā gatavu iekļaušanai maksājumā. Šī lietotāja izeja tiek izsaukta katrai maksājumam izvēlētajai transakcijai.

Ja transakcija tiek nosūtīta manuālai apstiprināšanai, tad kompānijas grāmatvežiem transakcija ir jāizvērtē manuāli un jāpieņem lēmums, ko ar šo transakciju darīt. Ja transakcija tiek apstiprināta, tad tā tiek atzīmēta kā gatava iekļaušanai maksājumā. Ja transakcija netiek apstiprināta, tad ir jānoskaidro iemesls, kāpēc šī transakcija tika radīta un noraidīta. Iemesli tam, kāpēc transakcija ir radīta un noraidīta iekļaušanai maksājumā, var būt dažādi, piemēram,

lietotājam, kurš ir izveidojis šo transakciju, nav autorizācijas izveidot tik lielus maksājumu vai arī maksājums neatbilst kompānijas biznesa noteikumiem. Transakcijas, kuras netiek apstiprinātas, ir iespējams pielāgot vai norakstīt.

Pēc tam, kad transakcijas tiek atzīmētas kā gatavas iekļaušanai maksājumā, tās tiek sagrupētas līmenī, kāds ir norādīts uz transakciju maksājuma metodes un datu bāzē tiek izveidots maksājums. Pēc tam, kad maksājums ir izveidots, tiek izsauktas lietotāja funkcija, kas ļauj izstrādātājiem realizēt papildu darbības, kuras veikt pēc maksājuma izveides vai arī atsaukt izveidoto maksājumu. Šī lietotāja izeja tiek izsaukta katram maksājumam, kas šajā procesā tiek apstrādāts.

Papildu minētajām lietotāja izejām, apdrošināšanas sistēma TIA ļauj definēt lietotāja izejas pirms un pēc brīžiem, kad datu bāzē tiek izveidoti jauni ieraksti vai izmainīti jau eksistējoši ieraksti, izmantojot entītijū pakotnes. Šīs lietotāja izejas var būt noderīgas, lai realizētu biznesa noteikumus, kas attiecas uz visām transakcijām, kas strādā ar doto datubāzes tabulu, vai arī veikt izmaiņas ierakstos, kas tiek izveidoti datu bāzēs, ja tam nav izveidotas piemērotas lietotāja izejas. Šīs lietotāja izejas nav ieteicams izmantot, lai realizētu sarežģītu loģiku, jo tādā veidā var rasties procesi, kurus ir grūti izsekot un grūti uzturēt.

Katrs no attēlā 3.2. redzamajiem procesiem ir jāsāk atsevišķi, lai ļautu lietotājam radīt lietotājam specifiskus biznesa procesus, ja tādi ir nepieciešami. Šos procesus var sākt gan manuāli, gan izveidojot pakešuzdevumus, kas veic šīs darbības automātiski. Šiem procesiem ir iespējams norādīt arī papildu parametrus, lai izveidotu procesus, kas apstrādā kādu noteiktu maksājumu grupu. Pēc tam, kad maksājumu process ir pabeigts, sistēma izveidos maksājumus sagrupē un nosūta apstrādei finanšu institūcijai, izmantojot maksājuma kanālā norādītos noteikumus.

3.2. Biznesa prasības

Automašīnu parka apdrošināšanas produkts ir specifisks produkts ar to, ka to pārsvarā iegādājas komerciāli klienti un ka tajā bieži tiek veiktas izmaiņas, piemēram, pievienotas vai izņemtas automašīnas. Šī iemesla dēļ business izstrādāja šim produktam specifiskus biznesa noteikumus attiecībā uz prēmijas iekasēšanu. Galvenā biznesa prasība šī produkta maksājumu procesam ir tāda, ka klientam ir jāsaņem maksājumi tikai tad, kad to paredz maksājumu kalendārs. Ja klients ir izvēlējis apmaksāt polisi reizi mēnesī, tad rēķins ir jānosūta tikai reizi mēnesī. Ja klients ir izvēlējis uzreiz apmaksāt visu polisi gadu, tad klients nedrīkst saņemt rēķinu polisi

gada vidū. Tas nozīmē to, ka visas izmaiņas, kuras tiek veiktas polisei un izmaina polises prēmiju, ir jāiekļauj nākamajā regulārajā rēķinā par polisi.

Tas rada problēmas ar transakcijām, kuras tiek radītas laika periodā starp brīdi, kad ir izgatavots pēdējais rēķins polises gadā un polises gada beigām. Šīs transakcijas nevar iekļaut rēķinā par nākamo polises gadu, jo biznesam ir prasība, kad vienā rēķinā var būt tikai transakcijas par vienu polises gadu vienai polisei. Šī iemesla dēļ tika izveidota papildu prasība, ka, ja polises gadā vairs nav palicis neviens regulārais rēķins, tad 15 dienas pēc nākamā polises gada sākuma tiek izgatavots vēl viens rēķins par iepriekšējo polises gadu, kurā tiek iekļautas transakcijas, kuras ir izveidotas pēc tam, kad pēdējais rēķina par polises gadu ir izveidots. Šādā veidā tiek nosegtas visas iespējamās transakcijas visā polises gadā.

Vēl viena klienta prasība par automašīnu parka apdrošināšanas produkta rēķiniem ir, ka rēķinā klientam ir jābūt redzamām visām transakcijām, kas ir veiktas polisei. Tas nozīmē to, ka ir jāizmaina biznesa standarta maksājumu process, ka brīdī, kad polisei ir veikta izmaiņa, izmaiņas dēļ radītā prēmijas dzēšanas transakcija, kas radīta par polises periodu, kurš vairs nav spēkā, bet tam jau ir uzģenerēta prēmija, netiek savstarpēji izlīdzināta ar veco prēmiju. Tas nozīmē to, ka visas transakcijas, kuras vēl nav iekļautas rēķinā un apmaksātas, ir atvērtas transakcijas un transakcijām vienmēr ir jābūt saskaņotiem atmaksas termiņiem, lai nerastos gadījumi, kad sistēma klientam nosūta nepareizu rēķinu.

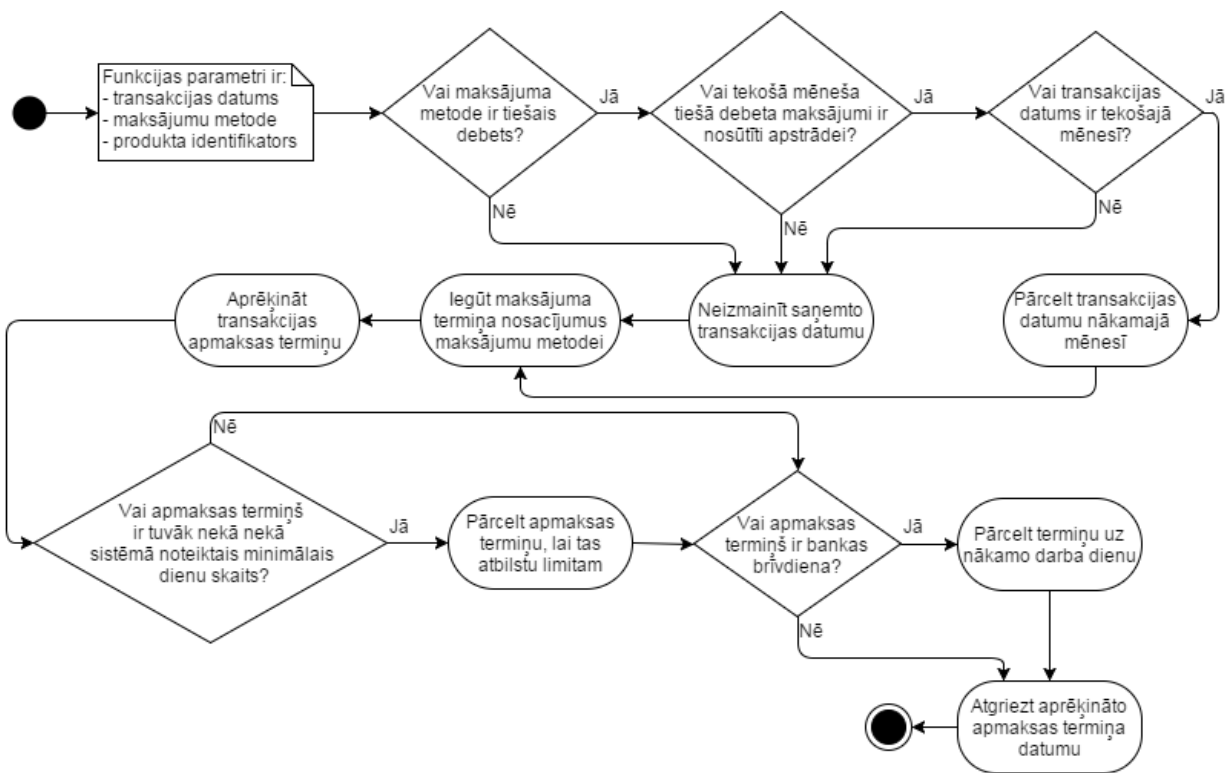
3.3. Risinājums transakciju pārvešanai uz plānotajiem rēķiniem

Kā iepriekš minēts, tad TIA kodols transakcijas apmaksas termiņu izvēlas pēc maksājuma metodes, kas ir uzstādīta uz transakcijas. Šajā gadījumā šis risinājums nav derīgs, jo klients vēlas, lai transakcijas tiktu grupētas kopā vienā maksājumā, kas iekļaujas polises maksājumu kalendārā, bet standarta risinājums piedāvā radīt jaunu rēķinu par katru izveidoto transakciju. Kompānijas klientiem tiek piedāvāti sekojoši maksājumu kalendāri:

- ikgadējs, klients polisi apmaksā polises gada sākumā;
- pusgada, klients saņem maksājumu reizi pusgadā par pusi polises gada;
- ceturkšņa, klients saņem maksājumu reizi ceturksnī par polises gada ceturksni;
- ikmēneša, klients saņem maksājumu katru mēnesi par polises gada mēnesi.

Izstrādātajam risinājumam ir jāstrādā visiem šiem maksājumu kalendāriem. Tas nozīmē to, ka izstrādājot transakciju pārvešanu uz plānotajiem rēķiniem ir jāņem vērā klienta izvēlētais

maksājumu kalendārs. Lai realizētu transakciju pārvešanu uz plānotajiem rēķiniem, transakcijas apmaksas termiņam uz izveidotajām transakcijām ir jāsakrīt ar transakcijas apmaksas termiņiem uz transakcijām, kas tiek izveidotas plānotajiem maksājumiem.



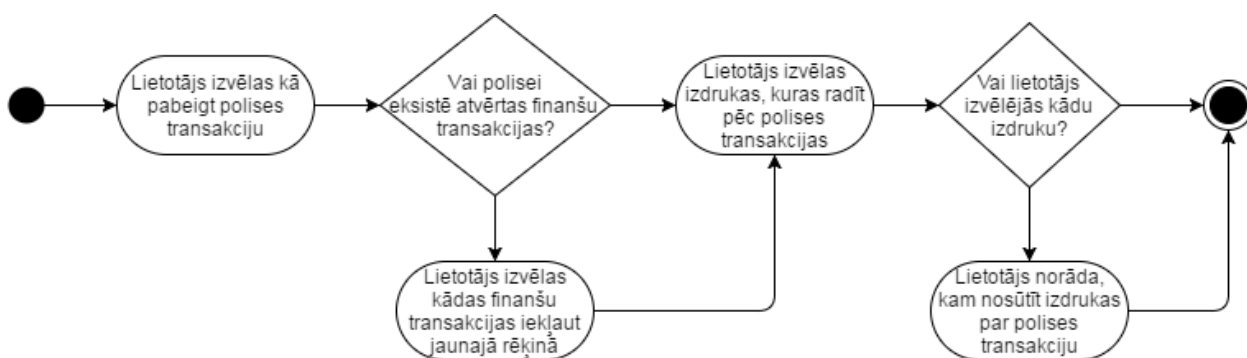
3.3. att. Maksājuma termiņa aprēķināšanas procesa aktivitāšu diagramma

Attēlā 3.3. ir redzama maksājuma termiņa automašīnu parka apdrošināšanas polisēm aprēķināšanas procesa aktivitāšu diagramma. Šis process kā parametrus saņem transakcijas datumu, maksājumu metodi un produkta identifikatoru. Automašīnu parka apdrošināšanas produktam var tik izmantotas divas maksājumu metodes: tiešais debets un bankas pārskaitījums. Šī apdrošināšanas produkta tiešā debeta maksājumi apstrādei tiek nosūtīti pēdējā mēneša 15.dienā pirms maksājuma termiņa, lai klients varētu nodrošināt to, ka viņa kontā maksājuma izpildes dienā ir pietiekoši daudz naudas rēķina apmaksai. Šī iemesla dēļ, ja tiešā debeta maksājuma transakcijas datums ir vēlāks par mēneša 15 dienu, tad šīm transakcijām transakcijas datums tiek pārcelts uz nākamo mēnesi.

Transakcijas apmaksas termiņa noteikšana notiek pēc funkcijā noteiktā transakcijas datuma, izmantojot samaksu termiņa nosacījumus, kuri tiek noteikti pēc uz transakcijas uzstādītās maksājumu metodes. Šis risinājums atšķiras no TIA kodola standarta procesa, kurā samaksas termiņa nosacījumi tiek atrasti izmantojot klienta kontu. Bankas pārskaitījuma samaksas termiņa nosacījums ir uzstādīt samaksas termiņu, kas ir vienāds ar transakcijas datumu. Tiešā debeta

Samaksas termiņa nosacījums ir uzstādīt samaksas termiņu nākamā mēneša pirmajā darba dienā pēc transakcijas datuma. Procesa turpinājumā tiek pārbaudīts, vai aprēķinātais atmaksas termiņš datums ir vairāk dienas nākotnē nekā sistēmā noteikts. Šis dienu skaits tiešā debeta maksājuma metodei ir 0 dienas, bet bankas pārskaitījuma maksājuma metodei tas ir 17 dienas. Ja aprēķinātais termiņš ir tuvāk nekā noteiktais dienu skaits, tad tas tiek pārcelts nākotnē, lai atbilstu šim limitam.

Pēdējais solis atmaksas termiņa aprēķināšanas procesā ir pārbaudīt, vai iegūtais datums ir banku brīvdiena. Banku brīvdienas tiek definētas lietojuma iestatījumos. Ja aprēķinātais datums ir banku brīvdiena, tas tiek pārcelts uz nākamo darba dienu pēc šī datuma. Pēc tam, kad šis datums ir pielāgots, tas tiek atgriezts un to var uzstādīt uz transakcijas kā atmaksas termiņu. Lai realizētu to, ka transakcijas atmaksas termiņš sakrīt ar standarta maksājumu kalendāru, šai funkcijai kā transakcijas datums ir jānorāda pirmais datums polises gadā, par kuru vēl nav uzģenerēts maksājums.



3.4. att. Lietotāja darbības, lai pabeigtu polises transakciju

Attēlā 3.4. ir redzamas darbības, kuras lietotājs veic, lai pabeigtu polises transakciju. Pirmais solis ir izvēlēties, kā pabeigt polises transakciju. Šajā brīdī lietotājs var izvēlēties izveidot polisi, izveidot polises piedāvājumu, saglabāt transakciju, lai pabeigtu to vēlāk, vai arī atcelt polises transakciju. Pēc tam, kad lietotājs ir izvēlēties izveidot polisi, ja polisei eksistē finanšu transakcijas, kuras vēl nav apmaksātas, tad lietotājam tiek parādīta forma, kurā lietotājs var izvēlēties vai šīs transakcijas iekļaut jaunā rēķinā vai nē, un norādīt atmaksas termiņa dienu skaitu. Nākamā lietotāja darbība ir izvēlēties, kādas izdrukās pievienot polises transakcijas. Katra apdrošināšanas produkta katrai polises transakcijai tiek definēts, kādas izdrukās ir iespējams pievienot un kuras no šīm izdrukām ir obligātas. Ja lietotājs ir izvēlēties kaut vienu izdrukā, tad lietotājam tiek parādīta forma, kurā norādīt saņēmēja informāciju. Pēc tam, kad šī forma tiek aizvērta, polises transakcija tiek pabeigta un saglabāta datu bāzē. Starp visām šīm darbībām ir pieejamas lietotāja izejas, kurās izstrādātāji var pielāgot sistēmas darbību.

Ir divas iespējas, kur sistēmā TIA ir iespējams realizēt transakcijas atmaksas termiņa manuālu pārrēķināšanu. Viena no iespējām ir realizēt termiņa pārrēķinu brīdī, kad finanšu transakcija tiek ierakstīta datu bāzē. Tas notiek pēc tam, kad lietotājs ir izvēlējis kā pabeigt polises transakciju. Otra šāda iespēja ir izmantot sistēmā TIA iestrādāto lietotāja funkciju, kas tiek izsaukta pēc tam, kad polises transakcija ir pabeigta, bet pirms tā ir saglabāta datu bāzē. Šī lietotāja funkcija ir paredzēta biznesam specifiskai loģikai, kuru izpildīt pēc tam, kad polises transakcija ir pabeigta. Lai realizētu transakciju atmaksas termiņa pārrēķinu automašīnu parka apdrošināšanas produktam, galvenokārt, ir izmantota lietotāja funkcija polises transakcijas beigās.

Šim risinājumam ir jāstrādā visa veida polises transakcijām. Galvenās no šīm transakcijām, kuras tiks apskatītas atsevišķi, ir jaunas polises izveidošana, izmaiņu izdarīšana jau eksistējošā polisē un polises pārtraukšanas transakcijas. Standarta gadījumā jaunai automašīnu parka apdrošināšanas polisei rēķins tiek izveidots kopā ar polisi, un šis rēķins ir par pirmo vai pirmajiem diviem polises maksājumiem, atkarībā no polises sākuma datuma. Ja automašīnu parka apdrošināšanas polisei tiek izdarītas izmaiņas, tad tās radītās finansiālās transakcijas tiek pārceltas uz nākamo norēķinu periodu. Ja automašīnu parka apdrošināšanas polise tiek pārtraukta, tad rēķins tiek izveidots uzreiz pēc polises transakcijas, un tajā tiek iekļautas visas atvērtās finanšu transakcijas.

Jaunas polises transakcijas gadījumā maksājuma procesa biznesa loģika automašīnu parka apdrošināšanas produkta gadījumā ir tāda pati kā citiem kompānijas piedāvātajiem apdrošināšanas produktiem. Tas nozīmē to, ka gadījumā, ja polisei radīto iemaksu ierakstu seguma perioda beigas ir mazāk nekā 20 dienas nākotnē, tiek uzģenerēts vēl viens iemaksas ieraksts, kurš tiek iekļauts pirmajā klientam piegādātajā rēķinā. Šī ir klientam specifiska biznesa loģika, kura lietotāja izejā, kura tiek izsaukta pēc tam, kad klients izvēlas izveidot polisi, pārbauda vai seguma perioda beigas ir pietiekoši tālā nākotnē, un, ja tā nav, tad tiek uzstādīts karogs pakotnes globālajā mainīgajā, kurš ziņo, ka lietotāja izejā pēc polises transakcijas beigām un pirms dati tiek saglabāti datu bāzē ir jāuzģenerē nākamais norēķinu periods. Šāds risinājums tika izvēlēts, lai gadījumos, kad polises transakcija ir jāatceļ, sistēmā nepaliktu lieki ieraksti, jo nākamā samaksas perioda radīšana ir atsevišķa transakcija, kuru TIA kodola risinājums polises transakcijas atcelšanai ignorētu. Lai noteiktu dienu skaitu, kuram samaksas perioda beigām ir jābūt nākotnē tika radīta jauns sistēmas iestatījums, kurš glabā šo vērtību, un kuru sistēmas lietotājs var izmainīt jebkurā brīdī.

Lai nodrošinātu automašīnu parka apdrošināšanas produkta jaunas polises transakcijas radīto finanšu transakciju pareizu atmaksas termiņu, tiek izmantota lietotāja funkcija polises transakcijas

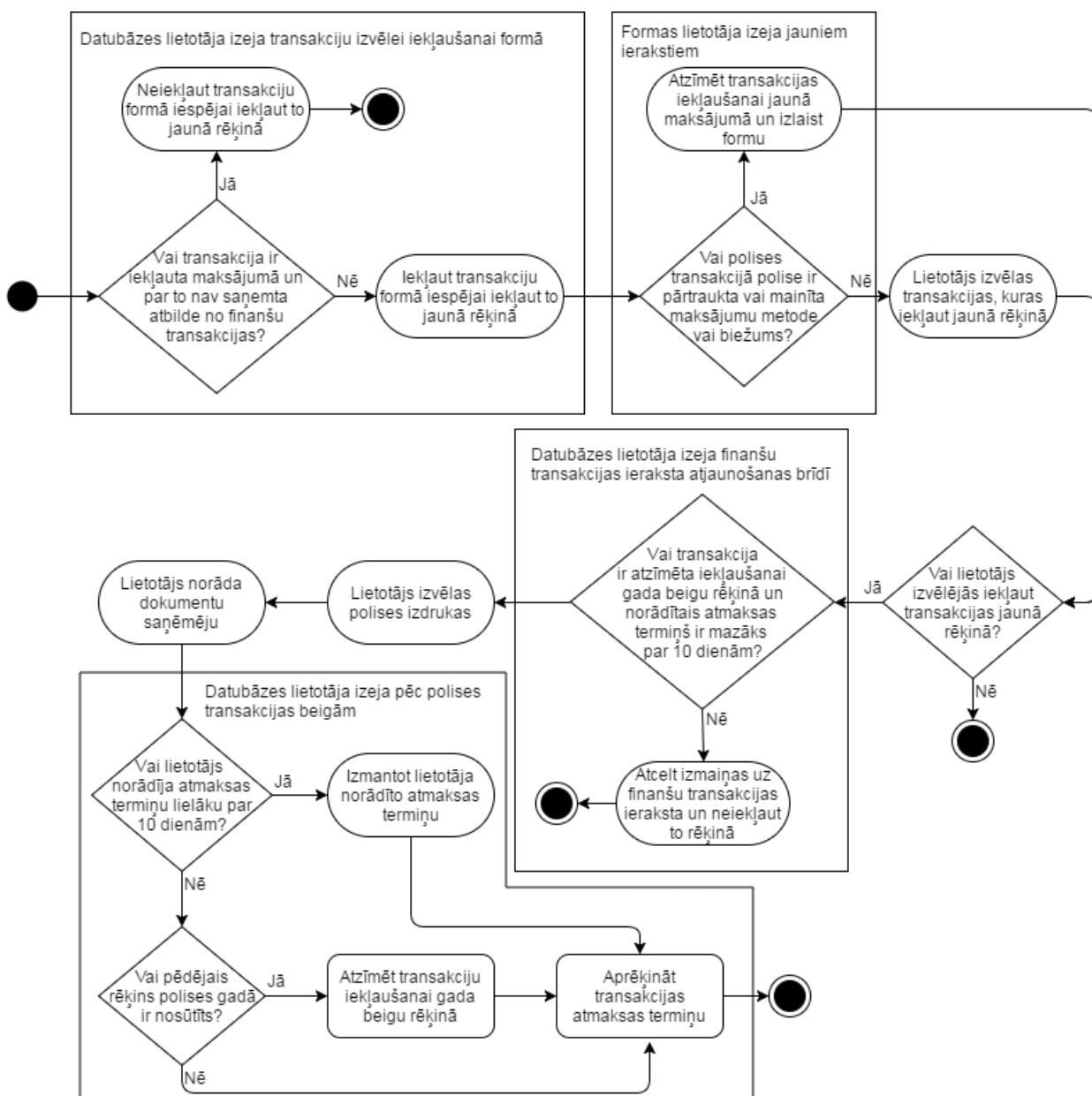
beigās. Šajā funkcijā jaunai automašīnas parka apdrošināšanas polisei no sākuma tiek pārrēķināts atmaksas termiņš jau izveidotajām finanšu transakcijām un tas tiek uzstādīts uz tām. Pēc tam tiek atjaunots rēķina izdrukas pieprasījuma ieraksts ar datumiem, starp kuriem ir jāatrodas finanšu transakcijas apmaksas termiņam, lai tas tiktu iekļauts maksājumā. Tā kā beigu datums tiek uzstādīts tāds pats kā pārrēķinātais termiņa datums izveidotajai transakcijai, tad tā tiks iekļauta maksājumā, kuru šis izdrukas pieprasījums uzģenerēs. Pēc tam, kad izdrukas pieprasījums ir pielāgots, tiek uzģenerēta finanšu transakcijas par nākamo norēķinu periodu un uz tās tiek uzstādīts tāds pats apmaksas termiņš kā uz pārējām jaunās polises transakcijām, kas nozīmē to, ka tās tiks iekļautas tajā pašā maksājumā. Šādā veidā tiek apstrādātas jaunās automašīnu parka apdrošināšanas polises.

Nākamais transakciju veids, kas tiks apskatīts atsevišķi ir automašīnu parka apdrošināšanas polises izmaiņu veikšana. Vispārīgajā gadījumā, veicot izmaiņas automašīnu parka apdrošināšanas polisē, finanšu transakcijas, kuras jau ir iekļautas kādā maksājumā, netiek izmainītas, un pēc polises transakcijas pabeigšanas jaunās finanšu transakcijas, kuras dzēš iepriekšējo polises prēmiju par periodu, kurš tika izmainīts, un rada jaunu prēmiju par šo periodu, tiek pārnestas uz nākamo maksājumu periodu. Klientam attiecībā uz šo polises transakciju ir papildu prasības, ka gadījumos, kad uz polises tiek mainīta maksājumu metode vai nomaksas kalendārs iepriekšējās atvērtās finanšu transakcijas ir jāizlīdzina ar vecās prēmijas dzēšanas transakciju. Papildu tam, klients vēlas, lai neparedzētos gadījumos lietotājam būtu iespējas izveidot jaunus rēķinus automašīnu parka apdrošināšanas produkta polisēm.

Klienta prasības automašīnu parka apdrošināšanas produkta polišu izmaiņām ir pietiekoši sarežģītas, un to realizācija nav triviāls process. Šo prasību realizācija nedrīkst būt pretrunā ar apdrošināšanas sistēmas TIA kodola standarta procesiem. Tā kā šis process galvenokārt tiek veikts pēc manuālas polišu apstrādes, tad izstrādājot šo procesu biznesa loģikas darbības ātrums nav primārais faktors. Klientam ir svarīgi, lai šis process būtu stabils, jo bieži vien izmaiņas automašīnu parka apdrošināšanas produkta polisēs, var radīt lielas polises prēmijas izmaiņas.

Tas nozīmē to, ka šo transakciju nepareiza apstrāde var nozīmēt lielu naudas summu nepareizu izmaksu vai iekasēšanu. Šādas darbības ir ne tikai nevēlamas biznesam, bet arī var radīt sliktu slavu kompānijai, tāpēc šī risinājuma stabilitāte ir ļoti svarīga. Šī iemesla dēļ, šis process tiek regulāri uzlabots, jo dažādu iemeslu dēļ, šajā procesā tiek atrastas kļūdas, kas rodas dēļ neparedzētām lietotāja darbībām vai iepriekš neparedzētām sistēmas procesu mijiedarbībām. Pēdējais uzlabojums, kas tika veikts procesā, ir automātiska finanšu transakciju iekļaušana jaunā

rēķinā bez lietotāja lēmuma gadījumos, kad tiek mainīta polises maksājumu metode vai maksājumu biežums.



3.5 att. Automašīnu parka apdrošināšanas produkta polises izmaiņas maksājumu procesa diagramma

Attēlā 3.5. ir redzama automašīnu apdrošināšanas parka apdrošināšanas produkta polišu izmaiņas maksājumu procesa aktivitāšu diagramma. Šajā diagrammā ir iekļautas tikai darbības, kas tiek veiktas, lai nodrošinātu maksājumu procesa atbilstību klienta prasībām, un lietotāja izejas, un to izsaukuma brīži procesā, kurās šīs darbības ir realizētas. Lai realizētu šo procesu tiek izmantotas lietotāja izejas gan datubāzē, gan sistēmas formās.

Pirmā darbība, kas tiek veikta pēc tam, kad lietotājs pabeidz polises transakciju ir to finanšu transakciju atase, kuras tiks parādītas lietotājam formā, kurā lietotājs var izvēlēties iekļaušanai

jaunā maksājumā. Lai izvēlētos, kādas transakcijas parādīt šajā formā, apdrošināšanas sistēmā TIA ir iestrādāta lietotāja izeja, kurā izstrādātāji var definēt savus noteikumus, pēc kuriem izvēlēties kādas transakcijas iekļaut formā, kurā lietotājs izvēlas, kādas finanšu transakcijas iekļaut maksājumā par polises transakciju. Ja iekļaušanai formā netiek izvēlēta neviena transakcija, tad sistēma šo formu izlaiž. Šī produkta gadījumā parādīšanai formā tiek izvēlētas visas polises atvērtās transakcijas, kuras nav iekļautas rēķinā, kurš šobrīd tiek apstrādāts ārējā finanšu institūcijā. To vai transakcija ir iekļauta maksājumā, kurš vēl ir apstrādē finanšu institūcijā var pēc nosūtīšanas apstrādei datuma un atbildes saņemšanas datuma laukiem uz maksājuma ieraksta. Ja nosūtīšanas apstrādei datums ir aizpildīts, bet atbildes saņemšanas datuma lauks ir tukšs, tad atbilde no finanšu institūcijas vēl nav saņemta, un šo maksājumu formā neparāda.

Tā kā sistēmas lietotāji bieži nepareizi izvēlējas finanšu transakcijas, kuras iekļaut jaunā maksājumā gadījumos, kad uz polises ir nomainīta maksājuma metode vai maksājumu biežums, tad tika izstrādāta funkcionalitāte, kas šajos gadījumos finanšu transakcija automātiski atzīmēja iekļaušanai jaunā rēķinā ārpus norēķinu kalendāra. Šādos gadījumos šī forma tika izlaista. Šo funkcionalitāti nav iespējams vienkāršā veidā realizēt iepriekšējā datu bāzes lietotāja izejā, jo šādā gadījumā nenostādāja sistēmas funkcionalitāte transakciju iekļaušanai jaunā rēķinā. Tāpēc šī funkcionalitāte tika realizēta formas trigerī, kurš nostrādā brīdī, kad formas fokuss pārslēdzās uz jaunu lauku. Lai realizētu šo funkcionalitāti, šajā trigerī tiek pārbaudīts, vai polises transakcijas laikā uz šīs polises ir mainījusies maksājumu metode vai maksājumu biežums. Ja kāda no šīm lietām ir mainījusies, tad visas transakcijas tiek atzīmētas iekļaušanai jaunā rēķinā un dienu skaits līdz atmaksas termiņa tiek norādīts kā 11. Šīs vērtības formas laukiem tiek uzstādītas izmantojot *Oracle Forms* procedūru *copy()*. Pēc tam, izmantojot *do_key()* procedūru, forma tiek aizvērta un tiek izpildīta visa formā iekļautā biznesa loģika.

Automašīnu parka apdrošināšanas polises izmaiņu transakcijas standarta gadījumā, transakcijas, kuras lietotājs ir izvēlējis iekļaut jaunā maksājumā tiks pārnestas uz nākamo norēķinu periodu. Tā kā klients vēlas, lai ārkārtas gadījumos būtu iespējams izveidot rēķinus ārpus norēķinu kalendāra, tad tika izstrādāta funkcionalitāte, kura, ja lietotājs norāda atmaksas termiņu lielāku par 10 dienām, tad šīm transakcijām tiek izveidots atsevišķs rēķins ārpus norēķinu perioda. Tā kā šis lauks standarta procedūrās netiek izmantots, tad klients piekrita šādam risinājumam.

Pēc tam, kad lietotājs beidz darbu šajā formā, TIA standarta process, katrai izvēlētajai transakcijai nomaina maksājumu statusu uz vēl neiekļautu maksājumā un atjauno tās atmaksas termiņu. Izvēlētas finanšu transakcijas tiek nodotas datu bāzei izmantojot pagaidu tabulu, kurā tiek

norādīti finanšu transakciju numuri, kopā ar polises transakcijas numuru un procedūrai unikālu atslēgu. Tā kā tiek atjaunota katra finanšu transakcija, kura ir izvēlēta iekļaušanai maksājumā, tad šo transakciju izmaiņu ir droši pārtvert lietotāja funkcijā, kura tiek izsaukta pirms šie ieraksti tiek atjaunoti datu bāzē. Šajā lietotāja izejā visas iekļaušanai jaunā rēķinā izvēlētas transakcijas tiek ievietotas pagaidu tabulā ar jaunu atslēgu, lai tās apstrādātu lietotāja izejā pēc polises transakcijas beigām. Gadījumos, kad transakcijas ir atzīmētas iekļaušanai gada beigu rēķinā un lietotājs nav norādījis atmaksas termiņa dienu skaitu lielāku par 10, un maksājumu metode vai biežums polises transakcijā nav mainīts, uz transakcijām tiek uzstādīts vecais atmaksas termiņš un maksājuma statuss. Šīs vērtības šajā brīdī vēl var iegūt no datu bāzes, jo šajā brīdī datu bāzes ieraksti vēl nav izmainīti. Šāds risinājums tika izvēlēts, jo no šīs lietotāja izejas nav iespējams apturēt datubāzes ierakstu atjaunošanu, tāpēc datubāzes ieraksti tiek atjaunoti ar tādām pašām vērtībām, kādas tās jau ir datu bāzē. Šis risinājums strādā, jo lietotāja izejā pēc polises transakcijas tiek izvēlētas tikai finanšu transakcijas, kuras nav iekļautas maksājumos.

Šajā brīdī lietotājs var izvēlēties kādas izdrukas radīt šai polises transakcijai, un kam tās nosūtīt. Pēc tam, kad lietotājs ir izdarījis šīs darbības, polises transakcija tiek pabeigta un tiek izsaukta lietotāja izeja pēc polises transakcijas. Šajā lietotāja izejā sistēma iegūst lietotāja norādīto atmaksas termiņu un pārbauda, vai tas ir vairāk nekā 10 dienas nākotnē, un, ja tā ir, tad tas tiek saglabāts pakotnes globālajā mainīgajā, un tiek izmantots tālākajā procesā. Gadījumā, ja lietotājs nav norādījis nepieciešamo atmaksas termiņa dienu skaitu, tiek atrasta pirmā diena, kurai nav izveidots maksājums, un tā tiek izmantota apmaksas termiņa aprēķinos. Pretējā gadījumā tiek izmantots lietotāja norādītais datums. Katra rēķinā iekļaujamā finanšu transakcija, tiek atrastāa izmantojot PL/SQL kursoru, kurš atlasa visas atvērtās polises finanšu transakcijas, kuru maksājuma statuss ir, ka tās nav iekļautas maksājumā un tās ir izveidotas tekošajā polises transakcijā, izmantojot polises transakcijas identifikatoru, vai ir iekļautas pagaidu tabulā brīdī, kad transakcijas tika apstrādātas pirms izlabošanas datu bāzē. Katrai transakcijai tiek aprēķināts transakcijas atmaksas datums, izmantojot pirmo datumu bez rēķina vai lietotāja norādīto datumu, kā transakcijas datumu, un aprēķinātais datums tiek uzstādīts uz finanšu transakcijas.

Pēdējā polises transakcija, kas tiks apskatīta, saistībā ar automašīnu parka apdrošināšanas produkta polišu maksājumu procesu ir polises pārtraukšanas transakcija. Pārtraucot automašīnu parka apdrošināšanas polisi visas atvērtās finanšu transakcijas ir jāiekļauj jaunā maksājumā un jānosūta klientam. Izmantotā loģika, lai atlasītu finanšu transakcijas, kuras ir jāiekļauj rēķinā, ir

līdzīga automašīnu parka apdrošināšanas polises izmaiņu transakcijai. Pēc tam šīs transakcijas tiek atzīmētas iekļaušanai maksājumā.

Tāpat kā uz polises izmaiņas transakciju, pārtraucot polisi, iekļaušanai formā, kurā lietotājs izvēlas finanšu transakcijas, kuras iekļaut jaunā rēķinā, tiek izvēlētas pilnīgi visas polises atvērtās transakcijas, kuras nav iekļautas maksājumā, kuru šobrīd apstrādā finanšu institūcija. Pēc tam šī forma tiek izlaista, un visas transakcijas, kuras tika izvēlētas parādīšanai formā, tiek atzīmētas iekļaušanai jaunā maksājumā un atmaksas termiņš tiek uzstādīts 11 dienas nākotnē. Lietotāja izeja pirms finanšu transakciju ierakstu atjaunošanas datu bāzē neatceļ šo transakciju iekļaušanu jaunā rēķinā, jo atmaksas termiņš ir uzstādīts vairāk nekā 10 dienas nākotnē.

Lietotāja funkcijā pēc polises transakcijas beigām, tiek aprēķināta maksājumā iekļaujamo transakciju apmaksājamā summa, un, ja šī summa ir mazāka par minimālo summu, kas ir definēta uz maksājuma metodes, lai izveidotu maksājumu, tad šīs transakcijas tiek savā starpā izlīdzinātas un atlikums tiek norakstīts, un rēķins klientam netiek nosūtīts. Ja maksājumā iekļaujamo finanšu transakcija apmaksas summa ir lielāka, par minimālo summu, kas ir definēta uz maksājuma metodes, tad līdzīgi, kā polises izmaiņas transakcijas gadījumā, tiek atlasītas visas transakcijas, kuras ir izvēlētas iekļaušanai jaunā maksājumā un tām tiek pārrēķināts atmaksas termiņš. Atmaksas termiņš tiek pārrēķināts izmantojot sistēmas datumu, kā transakcijas datumu, tādējādi nodrošinot to, ka rēķins klientam tiks nogādāts pēc iespējas ātrāk.

Automātisko šī produkta polises transakciju apstrāde notiek tāpat, kā manuālu polises transakciju apstrāde, tikai šīm transakcijām nav lietotāja izeju formās. Tas nozīmē to, ka šīs transakcijas vienmēr tiek apstrādātas kā šī produkta standarta gadījumi. Automātisku polišu transakciju gadījumā visas izveidotās jaunās finanšu transakcijas tiek pārvietotas iekļaušanai rēķinā par nākamo norēķinu periodu. Gadījumā, ja polise tiek pārtraukta neatmaksāta rēķina dēļ, tad finanšu transakcijas par nākotnes periodiem tiek izlīdzinātas ar uzģenerētajām prēmijas atsaukšanas transakcijām un atlikušās transakcijas tiek atstātas sistēmā parāda piedziņai. Ja automātiskas transakcijas dēļ ir nepieciešam nosūtīt klientam jaunu rēķinu, tad tajā tiek iekļautas visas atvērtās finanšu transakcijas un rēķins tiek nosūtīts ārpus norēķinu kalendāra.

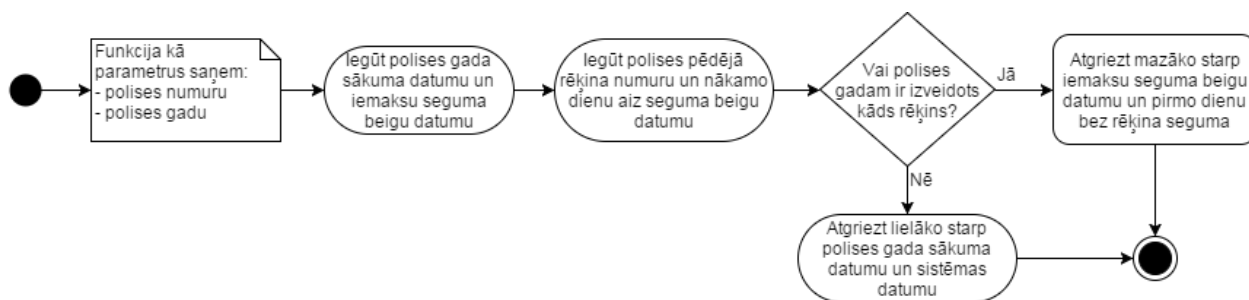
3.4. Risinājums gada pēdējā rēķina sagatavošanai

Tā kā biznesa klientam ar šo risinājumu ir jāvar nosūtīt klientam rēķinu par izmaiņām, kas veiktas periodā starp pēdējo maksājumu kalendāra rēķinu un jauna polises gada sākšanos, tad ir

nepieciešams risinājums, kas izveido rēķinu par šo periodu pēc polises gada beigām. Šī iemesla dēļ tika izstrādāts risinājums, kas finanšu transakcijas, kas atzīmē finanšu transakcijas, kuras jau ir iekļautas kādā rēķinā, bet kāda iemesla dēļ nav atmaksātas, un transakcijas, kuras ir izveidotas starp pēdējo regulāro rēķinu un polises gada beigām, tiek iekļautas jaunā rēķinā. Lai apstrādātu šādas transakcijas tika izstrādāta jauna procedūra, kura apstrādā šīs transakcijas.

Pirmais solis, kas ir realizēts, lai īstenotu gada pēdējā rēķina funkcionalitāti ir transakciju atzīmēšana pēdējam rēķinam. Transakciju atzīmē iekļaušanai gada beigu rēķinā izmantojot pielāgojamu lauku uz finanšu transakcijas ieraksta datu bāzē. Šim laukam var būt trīs vērtības. Tas var būt tukšs, kas nozīmē to, ka transakcija ir iekļaujama jebkurā rēķinā. Tā vērtība var būt viens, kas nozīmē to, ka finanšu transakcija ir iekļauta pēdējā norēķinu kalendāra maksājumā un to var apstrādāt sistēmas maksājumu atgādinājumu funkcionalitāte. Pēdējā šī lauka iespējamā vērtība ir divi, kas nozīmē to, ka finanšu transakcija ir iekļaujama tikai gada pēdējā maksājumā, kas tiek izveidots pēc polises gada beigām.

Ir divi veidi, kā var rasties transakcijas, kuras ir jāatzīmē iekļaušanai gada pēdējā rēķinā. Iekļaušanai gada pēdējā rēķinā tiek atzīmētas pilnīgi visas finanšu transakcijas, kuras tiek izvēlētas iekļaušanai jaunā rēķinā, automašīnu parka apdrošināšanas polises izmaiņas transakcijā, kura tiek veikta pēc tam, kad ir izveidots pēdējais rēķins. Šajā gadījumā uz finanšu transakcijas tiek uzstādīta vērtība divi, kas nozīmē to, ka šī transakcija ir iekļaujama tikai polises gada pēdējā rēķinā. Tas tiek darīts lietotāja izejā pēc polises transakcijas beigām, brīdī, kad tiek pārrēķināts finanšu transakciju atmaksas datums un lietotājs nav norādījis atmaksas termiņu vairāk kā 10 dienas nākotnē. Šajā gadījumā tiek izsaukta funkcija, kura atgriež pirmo datumu polises gadā, kuram vēl nav izveidots rēķins, un, ja šis datums sakrīt ar nākamā polises gada sākuma datumu, tad šī transakcija tiek atzīmēta iekļaušanai gada beigu rēķinā.



3.6. att. Pirmā datuma bez rēķina polises gadā iegūšanas aktivitāšu diagramma

Attēlā 3.6. ir redzama funkcijas pirmā datuma polises gadā, kuram nav izveidots rēķins iegūšanai. Šī funkcija kā parametrus saņem polises numuru un polises gadu, kuram šis datums ir jāiegūst. Polises gads ir nepieciešams, jo biznesa klients vēlas, lai rēķini par katru gadu klientam

tiktu nosūtīti atsevišķi, un vienā polises transakcijā var tikt radītas gan finanšu transakcijas, kuras ir nepieciešams iekļaut gada beigu rēķinā, gan finanšu transakcijas, kuras klientam ir jānosūta ar nākamo maksājuma kalendāra rēķinu. Pirmais solis šīs funkcijas darbībā ir iegūt pēdējās polises versijas dotajā polises gadā iemaksu seguma beigu datumu un polises gada sākuma datumu. Polises pēdējās versijas iemaksu seguma beigu datums ir nepieciešams, jo polisei var būt mainīts polises maksājumu kalendārs. Gadījumā, ja polise ir izveidota ar ikgadēju maksājumu kalendāru, un tas mēnesi vēlāk ir nomainīts uz ikmēneša maksājumu kalendāru, tad uz pirmās polises versijas iemaksu beigu datums būs polises gada beigas, bet uz vēlākās versijas tas būs tikai 3 mēnešus pēc polises gada sākuma. Lai iegūtu šos datus tiek izmantots PI/SQL kursori, kurš iegūst polises iemaksu beigu periodu no jebkuras polises līnijas, kura ir aktīva polises pēdējā versijā. To var darīt, jo iemaksu beigu periodiem ir jābūt vienādiem uz visām polises līnijām polisē.

Pēc tam, kad ir iegūts polises iemaksu beigu periods, tiek izsaukts nākamais kursori, kurš, izmantojot polises iemaksu beigu periodu polises gadā, atrod pēdējo maksājumu, kurā ir iekļautas finanšu transakcijas, kuru seguma beigu datums ir mazāks vai vienāds ar iepriekš iegūto polises iemaksu seguma beigu datumu. Šis kursori atgriež šī maksājuma numuru un lielāko finanšu transakcijas seguma beigu datumu, kurš ir mazāks vai vienāds ar polises iemaksu seguma beigu datumu, un pievieno tam vienu dienu, lai iegūtu pirmo dienu polises gadā, kurai nav izveidots rēķins. Lai to varētu izdarīt, kursora rezultāti tiek grupēti pēc maksājuma numura un rezultāti tiek kārtoti pēc maksājuma numura dilstošā secībā

Tā kā maksājums, kurā ir iekļautas finanšu transakcijas, kuru seguma perioda beigu datums ir mazāks par polises seguma beigu datumu, var neeksistēt, tad ir jāpārbauda vai tas eksistē. Ja kursori maksājumu seguma beigu perioda noteikšanai nav atgriezis nevienu vērtību, tad var uzskatīt, ka izveidojamais maksājums ir pirmais maksājums polises gadā, un kā rezultāts tiek atgriezts lielākais datums starp polises gada sākuma datumu un sistēmas datumu. Tiek atgriezts lielākais no šiem datumiem, jo šis rezultāts vēlāk tiek izmantots finanšu transakcijas apmaksas termiņa izmantošanai, un ir jānodrošina, lai finanšu transakcijas apmaksas termiņš nav pagātnē vai pirms polises gada sākuma. Ja kursori maksājumu seguma beigu datuma noteikšanai ir atradis rezultātu, tad tiek atgriezta mazākā vērtība starp pēdējo maksājuma seguma datumu un polises seguma pēdējo datumu. Salīdzināšana ar polises seguma pēdējo datumu notiek, lai vēlreiz pārlicinātos, ka netiek atgriezts datums no maksājuma, kurš izveidots kamēr polisei ir bijis cits maksājumu kalendārs.

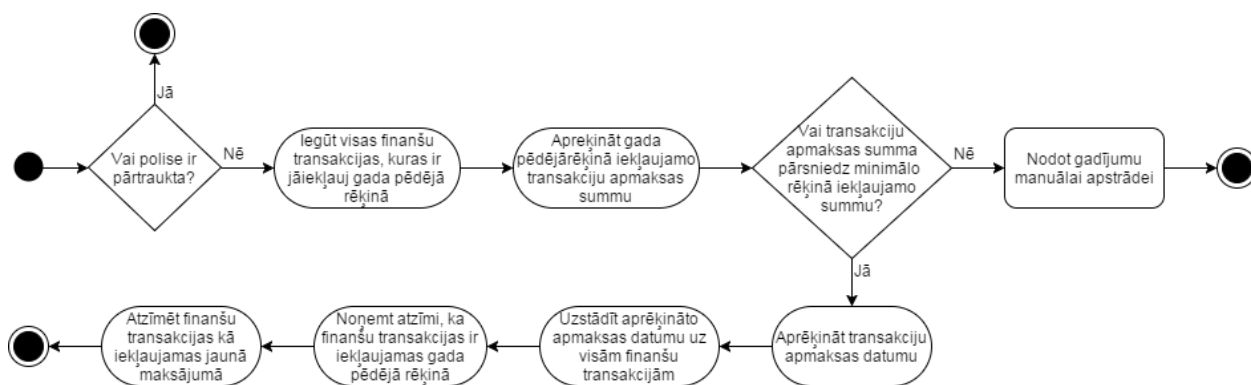
Kad funkcija pirmā datuma polises gadā, kuram nav izveidots maksājums, iegūšanai beidz darbu, tās atgrieztais datums tiek izmantots, lai noteiktu vai pēdējais maksājums polises gadā ir izveidots un atmaksas termiņa noteikšanai. Lai noteiktu, vai finanšu transakcijas ir jāatzīmē iekļaušanai gada pēdējā rēķinā, atgrieztais datums tiek salīdzināts ar nākamā polises gada sākuma datumu. Ja tie ir vienādi, tad var secināt, ka pēdējais maksājums polises gadā ir izveidots, un šī finanšu transakcija ir jāatzīmē iekļaušanai polises gada pēdējā rēķinā. Gadījumos, kad gada beigu rēķins par doto polises gadu jau ir uzģenerēts, šīs finanšu transakcijas netiek atzīmētas iekļaušanai gada beigu rēķinā, un tiek apstrādātas pēc kompānijas standarta maksājumu procesa citiem apdrošināšanas produktiem. Lai aprēķinātu finanšu transakcijas apmaksas termiņu, funkcijai, kas aprēķina finanšu transakcijas apmaksas termiņu kā transakcijas datums tiek nodots lielākais datums starp pirmo datumu, kuram nav rēķina, un jaunākās dotā polises gada polises versijas sākuma datumu. Šī salīdzināšana tiek veikta, lai vēlreiz nodrošinātu, ka apmaksas termiņš netiek uzstādīts pirms polises versija, par kuru ir radīta finanšu transakcija, stājas spēkā.

Otrs iespējamais veids, kā var rasties transakcijas, kuras ir jāatzīmē iekļaušanai gada pēdējā rēķinā ir pēdējā regulārā polises gada rēķina izveidošana. Finanšu transakcijas, kuras tiek iekļautas gada pēdējā rēķinā, tiek atzīmētas iekļaušanai gada pēdējā rēķinā, lai nodrošinātu to, ka gadījumos, kad šīs transakcijas, tiek atzīmētas iekļaušanai jaunā rēķinā, tām netiek uzģenerēts jauns rēķins ārpus maksājumu kalendāra, bet tās tiek iekļautas gada beigu rēķinā. Šajā gadījumā uz finanšu transakcijas tiek uzstādīta vērtība viens, lai gadījumā, ja kompānijas klients šo rēķinu neatmaksā, finanšu transakciju varētu apstrādāt sistēmas atgādinājumu funkcionalitāte. Šī vērtība tiek nomainīta uz 2 pēc tam, kad atgādinājumu process ir beidzis strādāt ar šīm finanšu transakcijām.

Lai nodrošinātu to, ka finanšu transakcijas, kuras ir atzīmētas iekļaušanai polises gada pēdējā rēķinā, netiek iekļautas nevienā citā rēķinā, tiek izmantota apdrošināšanas sistēmā TIA iestrādātā lietotāja izeja, kura tiek izpildīta brīdī, kad sistēma ir izvēlējusies atvērtās transakcijas, kuras iekļaut maksājumā. Šī lietotāja izeja tiek izsaukta katrai finanšu transakcijai, kuru sistēma ir izvēlējusies iekļaut maksājumā. Lietotāja izejas mērķis ir ļaut izstrādātājiem radīt papildu noteikumus finanšu transakciju iekļaušanai maksājumā un tā ļauj finanšu transakciju neiekļaut maksājumā, nosūtīt to manuālai apstiprināšanai vai atzīmēt transakciju, kā gatavu iekļaušanai maksājumā. Šajā gadījumā šī lietotāja izeja tiek izmantota, lai neiekļautu maksājumos finanšu transakcijas, kuras ir atzīmētas iekļaušanai automašīnu parka apdrošināšanas polises gada pēdējā maksājumā.

Gada pēdējā rēķina izveide ir realizēta, kā atsevišķs izdrukas pieprasījums. Izdrukas pieprasījumus var definēt gan kā izdruku, kas izveido papīra formāta failu, kuru var nosūtīt izdrukai

vai nosūtīt klientam elektroniskā formātā, gan kā PL/SQL procedūru, kas izpilda kādu procesu. Gada pēdējais rēķins ir realizēts kā izdrukas pieprasījums, kas izpilda procedūru, kura apstrādā finanšu transakcijas, kuras ir atzīmētas iekļaušanai automašīnu parka apdrošināšanas polises pēdējā rēķinā. Izdrukas pieprasījums tiek izveidots konkrētai apdrošināšanas polises, un uz tā tiek norādīts polises gads, par kuru šim pieprasījumam ir jāapstrādā polises finanšu transakcijas. Šis pieprasījums tiek uzģenerēts 75 dienas pirms jauna polises gada sākuma brīdī, kad polisei tiek sākts polises atjaunošanas cikls nākamajam polises gadam, kopā ar citiem polises dokumentiem, kas ir nepieciešami polises atjaunošanai. Atšķirībā no pārējiem izdruku pieprasījumiem, kuri tiek uzģenerēti polises atjaunošanas cikla sākumā un tiek izpildīti 45 dienas pirms polises gada sākuma, izdrukas pieprasījuma polises gada pēdējā rēķina radīšanai tiek izpildīts 15 dienas pēc jaunā polises gada sākuma.



3.7. att. Finanšu transakciju apstrādes, iekļaušanai gada pēdējā rēķinā, aktivitāšu diagramma

Attēlā 3.7. ir redzama polises finanšu transakciju apstrādes procedūras, lai tās iekļautu automašīnu parka apdrošināšanas polises gada pēdējā rēķinā, aktivitāšu diagramma. Pirmais solis, kas procedūrā tiek veikts ir pārbaude, vai polise ir pārtraukta. Šī pārbaude tiek veikta, jo izdrukas pieprasījums tiek uzģenerēts 90 dienas pirms tā izpildes, un šajā laika periodā polise var tikt pārtraukta. Gadījumā, ja polise ir pārtraukta visas finanšu transakcijas tiek iekļautas rēķinā, kas tiek radīts pēc polises pārtraukšanas transakcijas, kas nozīmē to, ka gada beigu rēķins šai polisei nav nepieciešams.

Ja polise nav pārtraukta, tad procedūra atrod visas finanšu transakcijas, kas ir atzīmētas iekļaušanai polises gada pēdējā rēķinā dotajam polises gadam. Tas ir realizēts izmantojot PL/SQL kursoru, kurš atrod visas atvērtās finanšu transakcijas, kuras ir atzīmētas iekļaušanai gada pēdējā rēķinā, izmantojot polises numuru un polises gadu. Kad visas gada pēdējā rēķinā iekļaujāmās finanšu transakcijas ir atrastas, tiek aprēķināta šo transakciju kopējā atmaksas summa. Ja šī summa ir mazāka par uz maksājumu metodes definēto minimālo maksājumu iekļaujamo summu, tad šīs

transakcijas tiek savā starpā izlīdzinātas, un tiek izveidots polises manuālas apstrādes pieteikums, kurš tiek nodots par polisi atbildīgajai kompānijas filiālei. Šajā gadījumā sistēmas lietotājam ir jāpieņem lēmums par to, ko darīt ar šo polises maksājumu, kas satur atlikušo summu. Visbiežāk atlikums pēc transakciju savstarpējas izlīdzināšanas tiek norakstīts, jo atlikusī summa ir pārāk maza, lai to iekasētu no klienta vai izmaksātu tam. Ja transakciju summa pārsniedz uz maksājumu metodes noteikto minimālo summu maksājuma izveidei, tad tiek turpināta finanšu transakciju sagatavošana iekļaušanai jaunā rēķinā.

Maksājumā iekļaujamajām finanšu transakcijām tiek pārrēķināts to apmaksas termiņš. Apmaksas termiņa aprēķinam kā transakcijas datums tiek nodots sistēmas datums, kas nozīmē to, ka transakcijas apmaksas termiņš tiks aprēķināts izmantojot sistēmas noklusētos noteikumus un, tas būs ārpus polises maksājumu kalendāra. Aprēķinātais apmaksas termiņš tiek uzstādīts uz visām finanšu transakcijām, kuras ir iekļaujam gada beigu maksājumā. Lai maksājumu varētu izveidot, finanšu transakcijām tiek noņemta atzīme par iekļaušanu polises gada pēdējā maksājumā, un transakcijas tiek izņemtas no maksājumiem, kuros tās šobrīd ir iekļautas. Pēc procedūras darba beigām finanšu transakcijas ir sagatavotas iekļaušanai jaunā rēķinā un nākamajā dienā šis rēķins tiek izveidots un nosūtīts klientam apmaksai.

3.5. Risinājums visu transakciju redzamībai rēķinā

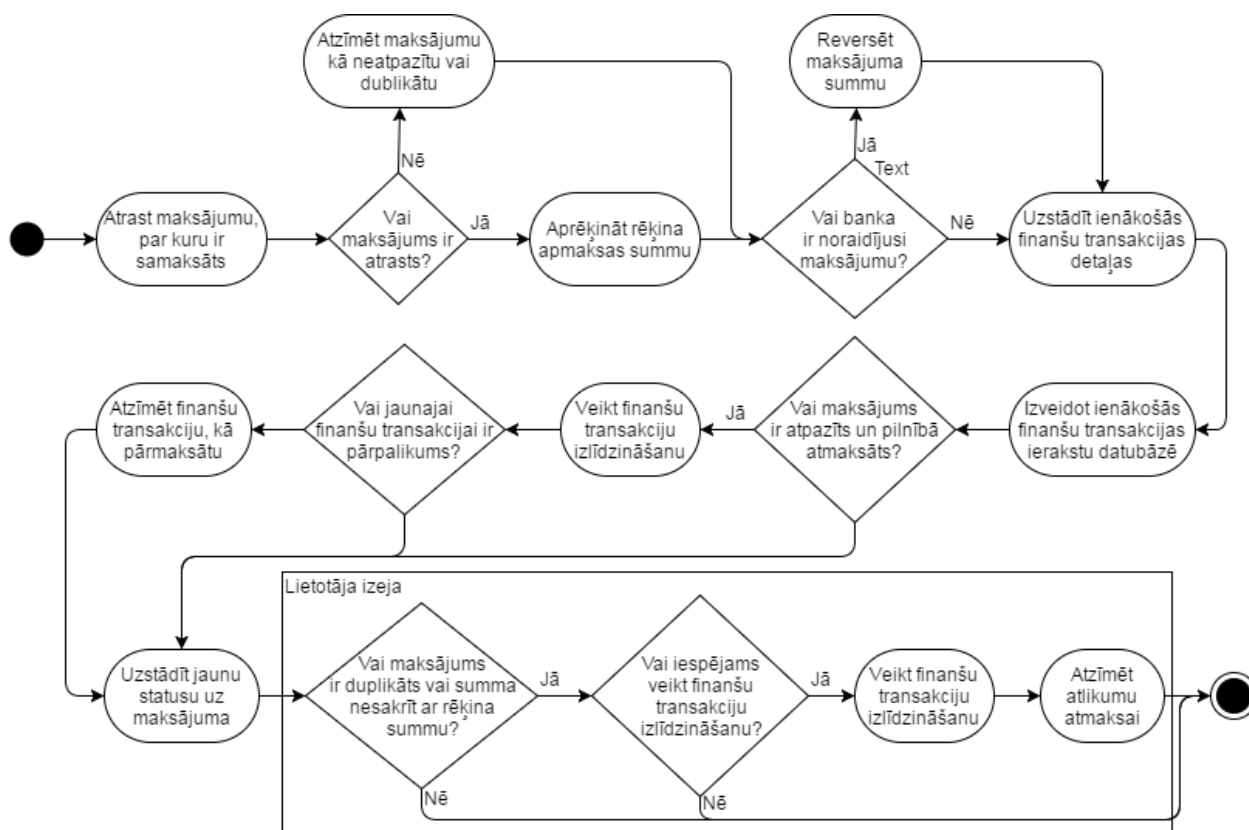
Pēdējais risinājums, kas ir nepieciešams, lai nodrošinātu klientam nepieciešamo automašīnu parka apdrošināšanas produkta polišu maksājumu procesa funkcionalitāti, ir visu transakciju par doto norēķinu periodu redzamība rēķinā. Rēķinos ir redzamas tikai atvērtās transakcijas, kuras tiek iekļautas rēķinā. Apdrošināšanas sistēmas TIA standarta maksājumu procesā pēc polises izmaiņas uzģenerētais prēmijas reverss tiek automātiski izlīdzināts ar iepriekšējo polises prēmiju par doto periodu. Tas nozīmē to, ka rēķinā, kurš tiek radīts pēc polises transakcijas, ir redzamas tikai jaunās finanšu transakcijas un jaunā prēmija. Šis risinājums ir derīgs gadījumos, kad rēķini tiek nosūtīti pēc katras polises transakcijas, un kompānijas klients šādā veidā var izsekot polises prēmijas izmaiņām. Tā kā automašīnu parka apdrošināšanas polisēm rēķini tiek radīti tikai sekojot norēķinu kalendāram, tad, ja tiktu izmantota TIA standarta funkcionalitāte, klients redzētu tikai prēmiju, kas ir jāmaksā par polisi tikai rēķina uzģenerēšanas brīdī, un klients nevarētu redzēt, kā polises prēmiju ir ietekmējušas izmaiņas, kas izdarītas starp radītajiem maksājumiem.

Apdrošināšanas sistēmas TIA kodola risinājumā finanšu transakcijas ir iespējams izlīdzināt vairākos līmeņos. Sistēmas piedāvātie finanšu transakciju izlīdzināšanas līmeņi ir: transakcija par vienu polises līniju un vienādu summu, transakcijas par vienu polises līniju, transakcijas par vienu polisi un vienādu summu, transakcijas par vienu polisi, transakcijas par vienādu summu vienā un transakcijas vienā kontā. Šāda funkcionalitāte ir nepieciešama, lai apdrošināšanas kompānija varētu kontrolēt kā finanšu transakcijas tiek savā starpā izlīdzinātas, kā arī, izsaucot zemākos izlīdzināšanas līmeņus pirms augstākiem, var panākt to, ka augstākas sakritības finanšu transakcijas tiek izlīdzinātas savā starpā.

Lai realizētu šo funkcionalitāti, ir nepieciešams pielāgot finanšu transakciju izlīdzināšanas procesu. Šie pielāgojumi ir izdarīti tikai automašīnu parka apdrošināšanas produktam, jo citi apdrošināšanas kompānijas produkti, izmanto maksājumu procesu, kas ir tuvāk TIA standarta maksājumu procesam, un tiem rēķini tiek radīti pēc katras polises izmaiņas. Sistēmas TIA standarta finanšu transakciju izlīdzināšana tiek izsaukta pēc tam, kad lietotājs ir izvēlējies pabeigts polises transakciju, kā aktīvu polisi. Pirms finanšu transakciju izlīdzināšanas ir iestrādā lietotāja izeja, kas ļauj realizēt klientam specifisku finanšu transakciju izlīdzināšanu, un izlaist sistēmas kodola finanšu transakciju izlīdzināšanas procesu. Tas šajā lietotāja izejā arī tiek darīts, lai nodrošinātu pareizu automašīnu parka apdrošināšanas produkta apstrādi un, lai nodrošinātu to, ka ir iespējams izlīdzināt finanšu transakcijas, gadījumos, ja mainās polisi apkalpojošā kompānijas filiāle. Gadījumā, ja tiek apstrādāta automašīnu parka apdrošināšanas polise, finanšu transakciju izlīdzināšana netiek veikta, un TIA kodola finanšu transakciju izlīdzināšanas process tiek izlaists.

Risinājums transakciju pārņemšanai uz plānotajiem rēķiniem polises transakcijas beigās pārrēķina šo transakciju atmaksas termiņus, lai netiktu radīti lieki rēķini. Lai šis risinājums veiksmīgi strādātu ir svarīgi, lai risinājums transakciju pārņemšanai uz plānotajiem rēķiniem strādātu korekti. Tas ir tāpēc, ka visas atvērtās transakcijas ir pieejamas iekļaušanai jaunā rēķinā, un kompānijas kredīta un debeta finanšu transakcijām atmaksas termiņš tiek aprēķināts dažādi. Ja risinājums transakciju pārņemšanai uz plānotajiem rēķiniem pieļauj kļūdu, tad klients var saņemt divus rēķinus par polises transakciju no kuriem vienā ir iekļauts tikai radītais prēmijas reverss, bet otrā jaunā prēmija. Gadījumos, kad klients ir izvēlējies tiešā debeta maksājuma metodi, lai norēķinātos par polisi, šādas kļūdas var sagādāt problēmas apdrošināšanas kompānijai, jo šīs finanšu transakcijas sistēmā tiek izmaksātas automātiski, un rada kompānijai sliktu slavu. Šī iemesla dēļ šāds risinājums ir riskants, un ir svarīgi nodrošināt tā stabilitāti.

Tagad visas polises finanšu transakcijas ir pieejamas iekļaušanai nākamajā regulārajā rēķinā, jo tās visas ir atvērtas transakcijas, t.i. to apmaksas summa nav nulle, un risinājums transakciju pārņemšanai uz plānotajiem rēķiniem ir pārrēķinājis šo transakciju atmaksas termiņus. Šīs transakcijas netiek savstarpēji izlīdzinātas arī pēc to iekļaušanas rēķinā, lai gadījumā, ja klientam ir kādas neskaidrības par saņemto rēķinu, visas šīs transakcijas varētu tikt iekļautas jaunā rēķinā. Visas finanšu transakcijas, kas ir iekļautas rēķinā par automašīnu parka apdrošināšanas polisi, savstarpēji tiek izlīdzinātas tikai pēc tam, kad klients ir samaksājis par tam nosūtīto rēķinu vai kad kompānija ir veikusi rēķina summas izmaksu, ja tas ir izejošs maksājums.



3.8. att. Ienākošā maksājuma apstrādes procesa aktivitāšu diagramma

Attēlā 3.8. ir redzama ienākošā maksājuma apstrādes aktivitāšu diagramma. Šis process tiek sākts pēc tam, kad ienākošais maksājums ir ielādēts sistēmā un tiek atpazīts. Ienākošais maksājums ir atpazīts, ja sistēma pēc tajā definētās biznesa loģikas var noteikt, ko šis maksājums apmaksā. Pēc šī procesa ienākošā maksājuma statuss var būt pilnībā atmaksāts, atpazīts kā maksājuma dublikāts, neatpazīts maksājums, rēķins ir daļēji atmaksāts vai pārmaksāts un banka maksājumu ir noraidījusi. Šie maksājuma statusi ietekmē to, kā sistēma apstrādā maksājumus iekļautās finanšu transakcijas.

Pirmais solis, ko sistēma dara ar ienākošo maksājumu, ir atrast rēķinu, par kuru šis maksājums ir izveidots. Ja sistēma šo rēķinu nevar atrast, tad sistēma ienākošo maksājumu atzīmē

kā neatpazītu, kas nozīmē to, ka sistēmas lietotājam, tas ir jāapstrādā vēlāk, bet, ja atrastajam rēķinam jau ir izveidots ienākošais maksājums, tad jaunais ienākošais maksājums tiek atzīmēts par dublikātu un tiks atmaksāts klientam. Ja rēķins, par kuru ir izveidots ienākošais maksājums ir atrasts, tad tiek aprēķināta šī brīža atlikusī neatmaksātā summa uz rēķinā iekļautajām finanšu transakcijā, jo starp rēķina izveides brīdi un atmaksas brīdi dažādu iemeslu dēļ šī summa var būt mainījies.

Ienākošais maksājums var tikt noraidīts bankā. Šādā gadījumā ienākošā maksājuma summa tiek reversēta un nodota tālāk manuālai apstrādei. Pēc tam, kad ir apstrādāti gadījumi, kad banka ir noraidījusi maksājuma pieprasījumu, tiek izveidota ienākošā maksājuma finanšu transakcija. Lai to varētu izdarīt tiek aizpildīti finanšu transakcijas standarta lauki, kas nepieciešami, lai sistēma varētu strādāt ar šo transakciju. Pēc finanšu transakcijas ieraksta izveides datu bāzē, tiek sāka finanšu transakciju savstarpēja izlīdzināšana. Finanšu transakciju izlīdzināšana notiek tikai tad, ja klients ir samaksājis visu summu, kas ir norādīta rēķinā vai pārmaksājis rēķinā norādīto summu. Sistēma no sākuma veic rēķinā iekļauto transakciju savstarpēju izlīdzināšanu. Savstarpējā rēķinā iekļauto finanšu transakciju izlīdzināšana tiek veikta līdz starp rēķinā iekļautajām finanšu transakcijām atvērtas paliek tikai debeta vai tikai kredīta finanšu transakcijas. Pēc tam, kad no maksājumā iekļautajām finanšu transakcijā atvērtas ir palikušas tikai debeta vai kredīta finanšu transakcijas, šīs transakcijas tiek izlīdzinātas ar ienākošo finanšu transakciju. Ja pēc ienākošās finanšu transakcijas izlīdzināšanas ar rēķinā iekļautajām finanšu transakcijām, uz tās ir palikusi apmaksas summas, kas nav vienāda ar nulli, šī finanšu transakcija tiek atzīmēta, kā pārmaksāta. Ja uz finanšu transakcijas atlikusī summa ir nulle, tad rēķins tiek atzīmēts kā pilnībā atmaksāts.

Ienākošā maksājuma apstrādes procesa beigās ir iestrādāta lietotāja izeja, kurā ir iespējams veikt papildu finanšu transakciju izlīdzināšanu, kuru sistēmas kodols nav veicis. Šajā lietotāja izejā var izmainīt maksājumu statusu, kuru uz finanšu transakcijas ir uzstādījis sistēmas TIA kodols, mainīt kontu, kurā nonāk neatpazītie maksājumi, izstrādāta papildu biznesa loģiku u.c. Automašīnu parka apdrošināšanas polišu maksājumu gadījumā šī lietotāja izeja tiek izmantota, lai veiktu papildu finanšu transakciju izlīdzināšanu, ja klienta maksājums ir mazāks par rēķinā norādīto summu. Šī finanšu transakciju izlīdzināšana notiek tāpat kā sistēmas kodola finanšu transakciju izlīdzināšanas process. Ja klients ir pārmaksājis rēķinā norādīto summu, starpība starp klienta maksājumu un rēķinā norādīto summu klientam tiek atmaksāta.

REZULTĀTI

Darba izstrādes rezultātā ir izstrādāti un ieviesti divi klientam nozīmīgi risinājumi: grāmatvedības perioda beigu virsgrāmatas grāmatojumu optimizācija un risinājums automašīnu parka apdrošināšanas produkta maksājumu apstrādei. Šie risinājumi ir realizēti un piegādāti klientam, un tiek izmantoti, lai realizētu klienta prasības. Pie mēneša beigu grāmatojumu optimizācijas analīzes un realizācijas tika pavadīts aptuveni cilvēk mēnesis, bet pie risinājuma automašīnu parka apdrošināšanas produkta maksājumu apstrādei analīzes un izstrādes tika pavadīti aptuveni 3 cilvēk mēnesi.

Papildu tam ir radīts šo risinājumu detalizēts apraksts, kas ļauj gūt ieskatu sistēmas kodola procesos un dažādos viedos, kā šos sistēmas procesus var pielāgot. Šis apraksts atspoguļo to, kā izstrādātāji var manipulēt ar šiem procesiem un vajadzības gadījumā izlaist sistēmas kodola procesus.

Darbā ir izveidoti apdrošināšanas sistēmas TIA kodola procesu apraksti sistēmas grāmatvedības perioda beigu grāmatojumu procesam un maksājumu procesam. Šie apraksti ir izveidoti, lai parādītu to, kas tiek darīts sistēmas kodolā, un kā tas atšķiras no klientam piegādātajiem procesiem. Aprakstā ir atrodama informācija par to, kā sistēmas uzstādīšanas darbi ir jāveic, lai panāktu sistēmas kodola korektu darbību.

Lai radītu vispārēju priekšstatu par sistēmu, tika izstrādāts augsta līmeņa apdrošināšanas sistēmas TIA apraksts. Šajā aprakstā ir īsi aprakstīta sistēmas arhitektūra un tās tehniskais risinājums. Papildu tam, tajā ir iekļauts īss apraksts par veidiem, kā izstrādātāji var strādāt ar apdrošināšanas sistēmu TIA.

SECINĀJUMI

Apdrošināšanas sistēma TIA ir augstā mērā pielāgojama un tajā ir iespējams izstrādāt jebkādas klienta prasības, kas ir attiecināmas uz apdrošināšanas biznesa pārvaldību. Sistēmas izstrādātāji to ir panākuši, iestrādājot lietotāja izejas dažādos sistēmas procesu kritiskos izpildes brīžos. Izmantojot šīs lietotāja izejas ir iespējams pielāgot sistēmas kodola darbību vai izlaist sistēmas kodola procesus pa visam.

Tā kā lietotāja izejas ir pieejamas tikai iepriekš noteiktās vietās un to skaits ir ierobežots, bieži vien nav praktiski realizēt visus nepieciešamos pielāgojumus tieši lietotāja izejā. Šī iemesla dēļ šo funkcionalitāti ir ieteicams pārnest uz PL/SQL pakotnēm, kuras tiek izsauktas no lietotāja izejām. Šādā veidā var panākt gan to, ka sistēmas kods ir lasāmāks un uzturamāks, gan to, ka šo funkcionalitāti ir iespējams izmantot vairākās lietotāja izejās.

Lai nodotu informāciju par to, ko sistēmai vajadzētu darīt, starp lietotāja izejām viena procesa ietvaros, var izmantot PL/SQL globālos mainīgos. Gadījumos, kad informācija ir jānodod ārpus viena procesa ietvariem, jeb, kad process norit vairākās klienta sesijās, var izmantot datu bāzes brīvos laukus, kas paredzēti klientam specifisku datu glabāšanai.

IZMANTOTĀLITERATŪRA

1. S.Moore, Oracle Database PL/SQL Language Reference, 11g Release 2 (11.2), 2014, <https://docs.oracle.com/cloud/latest/db112/LNPLS/LNPLS.pdf>, 12-4.
2. Immanuel Chan, Lance Ashdown, Oracle Database Performance Tuning Guide, 11g Release 2 (11.2), 2014, https://docs.oracle.com/cd/E11882_01/server.112/e41573.pdf , 12-17 – 12-20.

PIELIKUMI

1. pielikums. Mēneša beigu grāmatojumu skata izpildes plāns pirms optimizācijas

Apraksts	Objekta vārds	Procesora izmaksas	Ievades/Izvades izmaksas	Kardinalitāte	Balti
SELECT STATEMENT, GOAL = ALL_ROWS		5.68655E+11	4910425	2237568	691804776
UNION-ALL					
CONCATENATION					
MERGE JOIN		2.81834E+11	2547470	1114279	344312211
SORT JOIN		2.7757E+11	2547456	960873	189291981
HASH JOIN		2.76038E+11	2506286	960873	189291981
INDEX FULL SCAN	VIRSGRAMATAS_KONTA_PRIMARA_ATSLEGA	44921		1	54
HASH JOIN		2.75913E+11	2506285	1174400	223136000
TABLE ACCESS FULL	IEMAKSU_TABULA	28259492748	145472	23487999	1996479915
TABLE ACCESS FULL	PREMIJAS_SADALIJUMS_PAR_RISKIEM	1.9252E+11	1174291	195141968	20489906640
FILTER					
SORT JOIN		15296496	14	47	5264
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1077908	14	47	5264
NESTED LOOPS		2.8682E+11	2362955	2	618
HASH JOIN		2.86819E+11	2362941	1	197
TABLE ACCESS FULL	IEMAKSU_TABULA	28259492748	145472	23487999	1996479915
HASH JOIN		2.12041E+11	1174292	159661610	17882100320
INDEX FULL SCAN	VIRSGRAMATAS_KONTA_PRIMARA_ATSLEGA	44921		1	54
TABLE ACCESS FULL	PREMIJAS_SADALIJUMS_PAR_RISKIEM	1.9252E+11	1174291	195141968	20489906640
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1097818	14	1	112
CONCATENATION					
MERGE JOIN		2.81834E+11	2547470	1114279	344312211
SORT JOIN		2.7757E+11	2547456	960873	189291981
HASH JOIN		2.76038E+11	2506286	960873	189291981
INDEX FULL SCAN	VIRSGRAMATAS_KONTA_PRIMARA_ATSLEGA	44921		1	54
HASH JOIN		2.75913E+11	2506285	1174400	223136000
TABLE ACCESS FULL	IEMAKSU_TABULA	28259492748	145472	23487999	1996479915
TABLE ACCESS FULL	PREMIJAS_SADALIJUMS_PAR_RISKIEM	1.9252E+11	1174291	195141968	20489906640
FILTER					
SORT JOIN		15296496	14	47	5264
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1077908	14	47	5264
NESTED LOOPS		2.8682E+11	2362955	2	618
HASH JOIN		2.86819E+11	2362941	1	197
TABLE ACCESS FULL	IEMAKSU_TABULA	28259492748	145472	23487999	1996479915
HASH JOIN		2.12041E+11	1174292	159661610	17882100320
INDEX FULL SCAN	VIRSGRAMATAS_KONTA_PRIMARA_ATSLEGA	44921		1	54
TABLE ACCESS FULL	PREMIJAS_SADALIJUMS_PAR_RISKIEM	1.9252E+11	1174291	195141968	20489906640
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1097818	14	1	112
NESTED LOOPS		1.809E+11	1846845	4503	1589559
HASH JOIN		1.78649E+11	1824376	1765	425365
INDEX FULL SCAN	VIRSGRAMATAS_KONTA_PRIMARA_ATSLEGA	44921		1	54
HASH JOIN		1.78641E+11	1824375	1765	413010
TABLE ACCESS FULL	PRODUKTA_LINIJA	146960	4	249	2739
NESTED LOOPS					
NESTED LOOPS		1.78633E+11	1824371	9079	2024617
HASH JOIN		1.78326E+11	1787059	18656	3787168
TABLE ACCESS FULL	PROD_LINIJAS_RISKI	1602865	16	2739	35607
HASH JOIN		1.782E+11	1787043	1174400	223136000
TABLE ACCESS FULL	IEMAKSU_TABULA	28259492748	145472	23487999	1996479915
TABLE ACCESS FULL	PREMIJAS_SADALIJUMS_PAR_RISKIEM	1.28448E+11	1174291	65047323	6829968915
INDEX UNIQUE SCAN	POL_LINIJAS_PRIMARA_ATSL	9021		1	1
TABLE ACCESS BY INDEX ROWID	POLISES_LINIJA	16433		2	1
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1275388		13	3
NESTED LOOPS		1.809E+11	1846845	4503	1589559
HASH JOIN		1.78649E+11	1824376	1765	425365
INDEX FULL SCAN	VIRSGRAMATAS_KONTA_PRIMARA_ATSLEGA	44921		1	54
HASH JOIN		1.78641E+11	1824375	1765	413010
TABLE ACCESS FULL	PRODUKTA_LINIJA	146960	4	249	2739
NESTED LOOPS					
NESTED LOOPS		1.78633E+11	1824371	9079	2024617
HASH JOIN		1.78326E+11	1787059	18656	3787168
TABLE ACCESS FULL	PROD_LINIJAS_RISKI	1602865	16	2739	35607
HASH JOIN		1.782E+11	1787043	1174400	223136000
TABLE ACCESS FULL	IEMAKSU_TABULA	28259492748	145472	23487999	1996479915
TABLE ACCESS FULL	PREMIJAS_SADALIJUMS_PAR_RISKIEM	1.28448E+11	1174291	65047323	6829968915
INDEX UNIQUE SCAN	POL_LINIJAS_PRIMARA_ATSL	9021		1	1
TABLE ACCESS BY INDEX ROWID	POLISES_LINIJA	16433		2	1
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1275388		13	3

2. pielikums. Mēneša beigu grāmatojumu skata izpildes plāns pēc optimizācijas

Apraksts	Objekta vārds	Procesora izmaksas	Ievades/Izvades Izmaksas	Kardinalitāte	Baiti
SELECT STATEMENT, GOAL = ALL_ROWS		27040333768	145492	4	9382
UNION-ALL					
NESTED LOOPS		27040333768	145492	1	2323
NESTED LOOPS		27038903940	145478	1	2208
NESTED LOOPS		27038896269	145477	1	2201
MERGE JOIN CARTESIAN		27038859533	145473	1	2096
TABLE ACCESS BY INDEX ROWID	PARAMETRU_SKATS	41457	1	1	2015
INDEX RANGE SCAN	PARAMETRU_SKATS_C1	9971	1	1	
BUFFER SORT		27038818076	145472	16769825	1358355825
TABLE ACCESS FULL	IEMAKSU_TABULA	27038818076	145472	16769825	1358355825
TABLE ACCESS BY INDEX ROWID	PREMIJAS_SADALIJUMS_PAR_RISKIEM	36736	4	1	105
INDEX RANGE SCAN	PREMIJAS_SADALIJUMS_PAR_RISKIEM_AGR_TR_RT	23814	3	8	
INDEX RANGE SCAN	VIRSGRAMATAS_KONTA_KODS	7671	1	1	7
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1429828	14	1	115
NESTED LOOPS		27040333768	145492	1	2327
NESTED LOOPS		27038903940	145478	1	2212
NESTED LOOPS		27038896269	145477	1	2205
MERGE JOIN CARTESIAN		27038859533	145473	1	2100
TABLE ACCESS BY INDEX ROWID	PARAMETRU_SKATS	41457	1	1	2015
INDEX RANGE SCAN	PARAMETRU_SKATS_C1	9971	1	1	
BUFFER SORT		27038818076	145472	16769825	1425435125
TABLE ACCESS FULL	IEMAKSU_TABULA	27038818076	145472	16769825	1425435125
TABLE ACCESS BY INDEX ROWID	PREMIJAS_SADALIJUMS_PAR_RISKIEM	36736	4	1	105
INDEX RANGE SCAN	PREMIJAS_SADALIJUMS_PAR_RISKIEM_AGR_TR_RT	23814	3	8	
INDEX RANGE SCAN	VIRSGRAMATAS_KONTA_KODS	7671	1	1	7
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1429828	14	1	115
NESTED LOOPS		27040363703	145496	1	2366
NESTED LOOPS		27038933876	145482	1	2251
NESTED LOOPS		27038926204	145481	1	2244
NESTED LOOPS		27038902220	145478	1	2139
NESTED LOOPS		27038884527	145476	1	2126
NESTED LOOPS		27038875966	145475	1	2115
MERGE JOIN CARTESIAN		27038859533	145473	1	2100
TABLE ACCESS BY INDEX ROWID	PARAMETRU_SKATS	41457	1	1	2015
INDEX RANGE SCAN	PARAMETRU_SKATS_C1	9971	1	1	
BUFFER SORT		27038818076	145472	16769825	1425435125
TABLE ACCESS FULL	IEMAKSU_TABULA	27038818076	145472	16769825	1425435125
TABLE ACCESS BY INDEX ROWID	POLISES_LINIJA	16433	2	1	15
INDEX UNIQUE SCAN	POL_LINIJAS_PRIMARA_ATSL	9021	1	1	
TABLE ACCESS BY INDEX ROWID	PRODUKTA_LINIJA	8561	1	1	11
INDEX UNIQUE SCAN	PROD_LINIJAS_PRIMARA_ATSL	1050	0	1	
TABLE ACCESS BY INDEX ROWID	PROD_LINIJAS_RISKI	17693	2	3	39
INDEX RANGE SCAN	PROD_LIN_RISKU_UNIKALS_INDEKSS	8771	1	3	
TABLE ACCESS BY INDEX ROWID	PREMIJAS_SADALIJUMS_PAR_RISKIEM	23984	3	1	105
INDEX UNIQUE SCAN	PREMIJAS_SADALIJUMS_PAR_RISKIEMPRIMARA_ATS	16203	2	1	
INDEX RANGE SCAN	VIRSGRAMATAS_KONTA_KODS	7671	1	1	7
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1429828	14	1	115
NESTED LOOPS		27040363703	145496	1	2366
NESTED LOOPS		27038933876	145482	1	2251
NESTED LOOPS		27038926204	145481	1	2244
NESTED LOOPS		27038902220	145478	1	2139
NESTED LOOPS		27038884527	145476	1	2126
NESTED LOOPS		27038875966	145475	1	2115
MERGE JOIN CARTESIAN		27038859533	145473	1	2100
TABLE ACCESS BY INDEX ROWID	PARAMETRU_SKATS	41457	1	1	2015
INDEX RANGE SCAN	PARAMETRU_SKATS_C1	9971	1	1	
BUFFER SORT		27038818076	145472	16769825	1425435125
TABLE ACCESS FULL	IEMAKSU_TABULA	27038818076	145472	16769825	1425435125
TABLE ACCESS BY INDEX ROWID	POLISES_LINIJA	16433	2	1	15
INDEX UNIQUE SCAN	POL_LINIJAS_PRIMARA_ATSL	9021	1	1	
TABLE ACCESS BY INDEX ROWID	PRODUKTA_LINIJA	8561	1	1	11
INDEX UNIQUE SCAN	PROD_LINIJAS_PRIMARA_ATSL	1050	0	1	
TABLE ACCESS BY INDEX ROWID	PROD_LINIJAS_RISKI	17693	2	3	39
INDEX RANGE SCAN	PROD_LIN_RISKU_UNIKALS_INDEKSS	8771	1	3	
TABLE ACCESS BY INDEX ROWID	PREMIJAS_SADALIJUMS_PAR_RISKIEM	23984	3	1	105
INDEX UNIQUE SCAN	PREMIJAS_SADALIJUMS_PAR_RISKIEMPRIMARA_ATS	16203	2	1	
INDEX RANGE SCAN	VIRSGRAMATAS_KONTA_KODS	7671	1	1	7
TABLE ACCESS FULL	VIRSGRAMATAS_PAKESUZDEVUMS	1429828	14	1	115

Bakalaura darbs „Risinājumi apdrošināšanas sistēmas TIA darbības uzlabošanai” izstrādāts LU Datorikas fakultātē.

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

Autors: (*personiskais paraksts*) Oskars Kalniņš

Rekomendēju/nerekomendēju darbu aizstāvēšanai

Vadītāja: profesore Dr. dat. Aivars Niedrītis (*personiskais paraksts*) 29.05.2017.

Recenzents: Artis Mednis

Darbs iesniegts Datorikas fakultātē 29.05.2017.

Dekāna pilnvarotā persona: metodiķe Ārija Sproģe (*personiskais paraksts*)

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

Komisijas sekretāre: