

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**Bankas klientu priekšrocības tarifu piešķiršanas un
aprēķināšanas sistēma**

KVALIFIKĀCIJAS DARBS

Autors: Vladislavs Kļevickis

Studenta apliecības Nr.: vk14063

Darba vadītājs: Dr. Sc. Ing. Vadims Žuravļovs

RĪGA 2016

ANOTĀCIJA

Šajā dokumentā ir aprakstīta sistēma, kas bija izstrādāta vienai no Latvijas bankām, lai optimizētu atlaižu tarifu piešķiršanas un komisijas aprēķināšanas procesus.

Dokumentā ir iekļauta sistēmas tehniskā dokumentācija, kas ietver prasību specifikāciju, projektējumu, testēšanas dokumentāciju, pirmkoda piemērus, kā arī informāciju par projekta organizāciju.

Sistēma izstrādāta OpenEdge 10.2 vidē.

Atslēgvārdi: priekšrocības tarifi, banku sistēmas, OpenEdge.

ABSTRACT

This document describes a system which was developed for one Latvian bank in order to optimize preferential rate assignment and comission calculation processes.

The document contains technical documentation of the system, which includes requirements specification, software design description, testing review as well as information about project organisation.

The system was developed in OpenEdge 10.2 environment.

Keywords: preferential rates, OpenEdge software, banking systems

SATURA RĀDĪTĀJS

ANOTĀCIJA	2
ABSTRACT.....	3
SATURA RĀDĪTĀJS.....	4
1. APZĪMĒJUMU SARAKSTS	7
2. IEVADS	8
3. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA.....	9
3.1. Ievads	9
3.1.1. Nolūks.....	9
3.1.2. Darbības sfēra	9
3.1.3 Saistība ar citiem dokumentiem.....	9
3.1.4. Pārskats	9
3.2 Vispārējais apraksts	10
3.2.1. Produkta perspektīva.....	10
3.2.2. Produkta funkcijas	10
3.2.3. Lietotāja raksturozīmes	11
3.2.4. Pieņemumi un ierobežojumi	11
3.3. Funkcionālās prasības	12
3.3.1 Komisijas aprēķināšana (PTS-COM)	12
3.3.2 Klienta kategoriju pārskats un rediģēšana (PTS-CUST)	14
3.3.3 Kategoriju pārskats un rediģēšana (PTS-CAT)	17
3.3.4 Kategorijas tarifu pārskats un rediģēšana (PTS-RATE).....	20
3.3.5. Klienta karšu kategoriju pārskats un rediģēšana (PTS-CARD).....	23
3.3.6. Automātiskās priekšrocības tarifu piešķiršanas modulis (PTS-AUTO)	23
4. PROGRAMMATŪRAS PROJEKTĒJUMA APRĀKSTS	26

4.1. Ievads	26
4.1.1. Nolūks	26
4.1.2. Darbības sfēra	26
4.1.3. Saistība ar citiem dokumentiem.....	26
4.1.4. Dokumenta pārskats.....	26
4.2. Dekompozīcijas apraksts	26
4.3. Atkarības apraksts.....	29
4.3.1 DPD 0. līmenis.....	29
4.3.2. DPD 1. līmenis sistēmas lietotājam	30
4.3.3. Komisijas aprēķināšanas DPD 2. līmenis.....	31
4.4 Datu atkarības	32
4.4.1. Konceptuālais ER modelis.....	32
4.4.2. Fiziskais ER modelis.....	33
4.5. Datubāzes detalizēts projektējums.....	34
4.5.1. Piezīmes datu tabulām	34
4.5.2 „customer” Tabula	34
4.5.3. “account” Tabula	34
4.5.4. “card” Tabula.....	35
4.5.5. “category” Tabula	35
4.5.6. “object_categ” Tabula.....	36
4.5.7. “prefer_rate” Tabula	37
4.5.8. “standard_rate” Tabula	38
4.6. Lietotāja grafisko saskarņu projektējumi.....	39
4.6.1. Piezīmes.....	39
4.6.2. Klienta kategoriju apakšsistēmas sakārņu projektējums.....	39
4.6.3. Kategoriju apakšsistēmas sakārņu projektējums	40
4.6.4. Kategorijas tarifu apakšsistēmas sakārņu projektējums	41
4.6.5 Karšu tarifu apakšsistēmas sakārņu projektējums	43

5. TESTĒŠANĀS DOKUMENTĀCIJA	44
5.1. Apraksts	44
5.2. Testēšanas plāns	44
5.3 Testēšanas žurnāls.....	44
5.3.1. Komisiju aprēķināšana.....	44
5.3.2 Kategoriju modulis.....	46
6. PROGRAMMATŪRAS PROJEKTA ORGANIZĀCIJA	50
7. KVALITĀTES NODROŠINĀŠANA	51
8. KONFIGURĀCIJU PĀRVALDĪBA	52
9. DARBIETILPĪBAS NOVĒRTĒJUMS	53
10. REZULTĀTI.....	55
11. IZMANTOTI LITERATŪRA UN AVOTI.....	56
12. Pielikums.....	57

1. APZĪMĒJUMU SARAKSTS

PTS – priekšrocības tarifu piešķiršanas un aprēķināšanas sistēma.

Banka – uzņēmums, kurā tiek izmantota PTS.

Lietotājs – sistēmas lietotājs.

Klients – bankas klients.

PPS – programmatūras prasību specifikācija.

PPA – programmatūras projektējuma apraksts.

DB – datu bāze.

Saskarne – funkciju kopa, kas nodrošina datu apmaiņu starp lietotāju un lietojumprogrammu

ER (Entity relationship) – diagrammas tips, kurš attēlo datubāzes entītiņu savstarpējas atkarības.

OpenEdge – Progress OpenEdge relāciju datu bāzes pārvaldības sistēma.

Tarifs – funkcija, kas nosaka komisiju noteiktām pakalpojumam.

Kategorija – priekšrocības tarifu komplekts.

Karte – bankas norēķinu karte.

Konts – bankas konts.

2. IEVADS

Banka piedāvā saviem klientiem dažādus pakalpojumus, par kuriem tiek ieturētas komisijas. Pēdējos gados liela klientu skaitam noteiktiem pakalpojumiem bija piešķirtas atlaides, jeb priekšrocības tarifi. Pie tam, dažādiem klientiem atlaides par vienu un to pašu pakalpojumu var ievērojami atšķirties – gan ar derīguma termiņu, gan ar komisijas aprēķināšanas formulu. Dažādi tarifi var būt piešķirti arī klienta konkrētām norēķinu kartēm.

Pārsvārā, priekšrocības tarifi tiek piešķirti klientiem pēc kaut kādām pazīmēm, jeb kategorijām. Piemēram, dažādi atlaižu komplekti ir pusaudžiem un senioriem. Tāpēc bankai arī jā rūpējas par savlaicīgu tarifu piešķiršanu un noņemšanu.

Lai optimizētu priekšrocības tarifu piešķiršanas un komisiju aprēķināšanas procesus, bija izstrādāta “Bankas klientu priekšrocības tarifu piešķiršanas un aprēķināšanas sistēma” (priekšrocības tarifu sistēma, PTS). Sistēma realizē gan saskarni bankas biznesa nodaļas darbiniekiem, gan automātisko funkcionalitāti.

Priekšrocības tarifu sistēmu bija nolemts integrēt centrālajā bankas uzskaites sistēmā, izveidojot jaunus izvēlnes punktus. Sistēma bija izstrādāta OpenEdge 10.2 vidē.

Dokuments ietver PTS programmatūras prasību specifikāciju, programmatūras projektējuma aprakstu, testēšanas dokumentāciju, projekta organizāciju, programmatūras kvalitātes nodrošināšanas aprakstu, konfigurāciju pārvaldību, darbietilpības novērtējumu, ka arī darba rezultātus un secinājumus.

3. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

3.1. Ievads

3.1.1. Nolūks

Šis dokuments izstrādāts, lai definētu PTS prasības. Turpmāk sistēma tiks izstrādāta atbilstoši šīm prasībām.

Dokuments ir paredzēts programmatūras izstrādātājiem, projekta vadītājam un pasūtītājam.

3.1.2. Darbības sfēra

Bankas klientu priekšrocības tarifu sistēma ir izstrādāta paredzēta bankas darbiniekiem, lai optimizētu visus procesus, kas ir saistīti ar priekšrocības tarifu piešķiršanu.

Sistēmai ir 2 galvenie mērķi:

Nodrošināt lietotāju saskarni, kurā dot iespēju manuālajā režīmā veikt visas vajadzīgas operācijas.

Nodrošināt automatisko funkcionalitāti operācijām, kuras neprasa lietotāja tiešo dalību.

3.1.3 Saistība ar citiem dokumentiem

Dokuments ir izstrādāts pēc standarta LVS 68:1996 „Programmatūras prasības specifikācijas ceļvedis” vadlīnijām.

3.1.4. Pārskats

Dokuments sastāv no 3 nodaļām:

Ievads – satur specifikācijas nolūku, darbības sfēras aprakstu, saistību ar citiem dokumentiem.

Vispārējais apraksts – vispārējā informācija par produkta perspektīvu, funkcijām, lietotāju raksturiezīmēm un citiem pieņēmumiem.

Funkcionālās prasības – programmatūras funkciju apraksts (katras funkcijas apraksts, ievaddati, apstrāde, izvaddati)

3.2 Vispārējais apraksts

3.2.1. Produkta perspektīva

Produkts tiek pozicionēts kā Centrālās Bankas uzskaites sistēmas sastāvdaļa.

3.2.2. Produkta funkcijas

PTS apakšsistēmu tabula

Apakšsistēma	Apakšsistēmas apraksts	Apakšsistēmas funkciju kodi
Komisiju aprēķināšana (PTS-COM)	Apakšsistēma nodrošina komisiju aprēķināšanu par pakalpojumiem pēc dažādu veidu tarifiem.	PTS-COM-SR PTS-COM-PR PTS-COM-CC
Klienta kategoriju pārskats un rediģēšana (PTS-CUST)	Apakšsistēma dod iespēju lietotājam apskatīt, pievienot un slēgt noteikta klienta tarifu kategorijas, apskatīt noteiktas kategorijas tarifus, apskatīt karšu kategorijas un tarifus	PTS-CUST-NEW PTS-CUST-CLS PTS-CUST-HIS PTS-CUST-CRD
Kategoriju pārskats un rediģēšana (PTS-CAT)	Apakšsistēma dod iespēju lietotājam apskatīt, pievienot, dzēst un rediģēt klientu un karšu kategorijas.	PTS-CAT-FIND PTS-CAT-EDIT PTS-CAT-NEW PTS-CAT-CLS PTS-CAT-HIS
Kategorijas tarifu pārskats un rediģēšana (PTS-RATE)	Apakšsistēma dod iespēju apskatīt noteiktās kategorijas tarifus, kā arī pievienot, dzēst un rediģēt tos. Apakšsistēmā var strādāt pārskata un rediģēšanas režīmos	PTS-RATE-ADD PTS-RATE-EDIT PTS-RATE-DEL PTS-RATE-HIS
Klienta karšu kategoriju pārskats un rediģēšana PTS-CARD	Apakšsistēma dod iespēju apskatīt klienta karšu sarakstu ar ieturēšanas tarifiem, kā arī pievienot un dzēst klienta karšu	PTD-CARD-INF PTS-CARD-ADD PTS-CARD-CAT-CLS

	kategorijas, apskatīt kategoriju tarifus.	
Automātiskās piešķiršanas/rediģēšanas funkcijas PTS-AUTO)	Šajā apakšsistēmā ir funkcijas, kuras nostrādā automātiski dienas sākumā vai dienas beigās.	PTS-AUTO-CLS PTS-AUTO-SEN PTS-AUTO-IRAT PTS-AUTO-ICAT

3.2.3. Lietotāja raksturiezīmes

Sistēma paredzēta bankas darbiniekiem, kuriem ir augsta darba pieredze un kvalifikācija. Sistēmas lietotājam jābūt plašām zināšanām produkta darbības sfērā. Nepieciešams augsts atbildības līmenis, jo visas darbības tiek veiktas ar reāliem bankas klientu datiem.

3.2.4. Pieņēmumi un ierobežojumi

Sistēmai jābūt pieejamai visas darbadienas laikā.

Sistēmas automātiskai funkcionalitātei jābūt pieejamai arī naktī.

Vienlaicīgi sistēmu var izmantot vairāki lietotāji.

Sistēmai jābūt izstrādātai OpenEdge vidē, lietotāju saskarnēm jābūt realizētām konsoles režīmā.

3.3. Funkcionālās prasības

3.3.1 Komisijas aprēķināšana (PTS-COM)

3.3.1.1. Standarta tipa komisijas aprēķināšana (PTS-COM-SR)

Mērķis

Aprēķināt komisiju, izmantojot standarta tipa tarifa parametrus.

Ievade

Funkcija pieņem vērtības sekojošiem parametriem:

Pakalpojuma valūtas

Operācijas summa

Tarifa tips

Tarifa parametri – min, max, proc, ost

Apstrāde

Pakalpojuma summa tiek konvertēta no valūtas 1 uz nacionālo valūtu (Eiro).

Balstoties uz tipa (1 vai 2) ir izvēlēta aprēķināšanas funkcija:

1. tips: $\text{Min}(\text{Max}(\text{sum} * \text{proc}), \text{min}), \text{max}) + \text{ost}$

2. tips: $(\text{Sum} - \text{min}) * \text{proc} + \text{ost}$

Izvade

Komisija

3.3.1.2 Komisijas aprēķināšana pēc priekšrocības tarifa (PTS-COM-PR)

Mērķis

Aprēķināt komisiju, izmantojot priekšrocības tarifa parametrus

Ievade

Klienta vai kartes ID

Pakalpojuma valūtas

Operācijas summa

Tarifa ID

Apstrāde

No visiem piešķirtiem priekšrocības tarifiem pēc ID izvēlās vienu ar lielāko prioritāti.

Ja izvēlētā priekšrocības tarifa tips ir 3:

- 1) Aprēķina komisiju ar standarta tarifa parametriem.
- 2) Pareizina rezultātu ar proc vērtību.

Citādi aprēķina komisiju ar priekšrocības tarifa parametriem

Izvade

Komisija, Prioritāte

3.3.1.3 Komisijas par pakalpojumu aprēķināšana klientam vai kartei (PTS-COM-CC)

Mērķis

Aprēķināt komisiju, ņemot vērā gan klientu, gan karšu tarifus.

Ievade

Klienta vai kartes ID

Operācijas valūtas

Summa

Tarifa ID

Apstrāde

-
- 1) Aprēķina komisiju pēc standarta tarifa.
 - 2) Aprēķina komisiju pēc priekšrocības tarifa.
 - 3) Pēc ID pārbauda, vai tas ir klients vai karte. Ja tā ir karte, tad aprēķina komisiju pēc priekšrocības tarifa, izmantojot kartes īpašnieka ID.
 - 4) No visiem aprēķinātiem rezultātiem izvēlās vienu ar lielāko prioritāti (uzskatam, ka standarta veida tarifam prioritāte ir 0) vai ar lielāko komisiju, ja prioritātes ir vienādas.

Izvade

Bankas sistēmas konts

Komisija

Prioritāte

3.3.2 Klienta kategoriju pārskats un rediģēšana (PTS-CUST)

3.3.2.1. Kategorijas piesaiste klientam (PTS-CUST-NEW)

Mērķis

Piešķirt kategoriju klientam

Ievade

Lietotājs izvēlās pogu <Jauns> un aizpilda sekojošas vērtības:

- Kategorijas kods
- Ir spēkā no
- Ir spēkā līdz
- Piezīme

Jānospiež <Saglabāt>

Apstrāde

Pēc <Saglabāt> pogas nospiešanas tiek veiktas sekojošas darbības:

1. Funkcija pārbauda kategorijas lauku. Ja tāds kategorijas kods neeksistē, tiek radīts paziņojums E1.
2. Funkcija pārbauda, vai lietotājam ir tiesības piešķirt šo kategoriju. Kļūdas gadījumā parādās paziņojums E2.
3. Funkcija pārbauda, vai lauka “Ir spēkā no” datums nav mazāks par šodienas datumu. Kļūdas gadījumā parādās paziņojums E3.
4. Funkcija pārbauda, vai lauka “Ir spēkā līdz” datums nav mazāks par “Ir spēkā no” datumu. Kļūdas gadījumā parādās paziņojums E4.

Izvade

Kategorija parādās sarakstā.

Kļūdu paziņojumi:

- E1 – Kategorija ar kodu “<kods>” neeksistē
- E2 – Kategorija ar kodu "<kods>" vēl nevar būt piesaistīta. Nav tiesību!
- E3 - Nedrīkst piesaistīt kategoriju postfactum!
- E4 – Kategorijas periods ievadīts nekorekti!

3.3.2.2. Kategorijas noņemšana (PTS-CUST-CLS)

Mērķis

Noņemt klientam kategoriju.

Ievade

Lietotājs nospiež pogu “Slēgt”.

Apstrāde

Funkcija pārbauda, vai lietotājam ir tiesības veikt šo darbību. Kļūdas gadījumā parādās paziņojums E1.

Izvade

Ja funkcija nostrādā sekmīgi, kategoriju saraksts uz ekrāna atjaunojas.

Kļūdu paziņojumi

E1 - Jums nav tiesību mainīt šo ierakstu

3.3.2.3. Tarifu apskatīšana (PTS-CUST-RATE)

Identifikators:

Mērķis

Apskatīt kategorijas tarifus

Ievade

Kategoriju sarakstā lietotājs izvēlās attiecīgu kategoriju un nospiež pogu <Tarifi> vai "T" uz tastatūras.

Apstrāde

Funkcija pārbauda vai kategorijai ir atlaižu tarifi.

Izvade

Atveras saskarne ar kategorijas tarifu sarakstu bez rediģēšanas iespējas.

3.3.2.4. Karšu tarifu apskatīšana (PTS-CUST-CARD)

Mērķis

Apskatīt klienta karšu kategorijas

Ievade

Kategoriju sarakstā lietotājs izvēlās attiecīgu kategoriju un nospiež pogu <Kartes> vai "K" uz tastatūras.

Apstrāde

Funkcija pārbauda, vai klientam ir norēķinu kartes.

Izvade

Atveras saskarne, kas atbilst klienta karšu tarifu modulim.

3.3.3 Kategoriju pārskats un rediģēšana (PTS-CAT)

3.3.3.1. Kategorijas meklēšana (PTS-CAT-FIND)

Mērķis

Atrast kategoriju pēc nosaukuma vai apraksta.

Ievade

1. Lietotājs nospiež pogu <Meklēt>
2. Aizpilda vismaz vienu no ievadlaukiem - “Kods sākas ar: ”, “Apraksts ietver” .
3. Nospiež <Labi> pogu vai ENTER.

Apstrāde

Ja neviena kategorija nebija atrasta, tiek izvadīts kļūdu paziņojums E1.

Izvade

Galvenajā apakšsistēmas formā kategoriju sarakstā paliek tikai tās kategorijas, kuras atbilst meklēšanas kritērijiem. Lai atgriezties pie pilnā saraksta, ir jānospiež F4.

Kļūdu paziņojumi

E1 – Nekas nav atrasts.

3.3.3.2. Kategorijas rediģēšana (PTS-CAT-EDIT)

Mērķis

Rediģēt eksistējošās kategorijas parametrus.

Ievade

1. Lietotājs nospiež vienu no trim pogām:
 - <Mainīt aprakstu>
 - <Saīsināt termiņu>
 - <Izmainīt prioritāti>
 2. Aizpilda atbilstošu lauku.
 3. Nospiež <Saglabāt> pogu.
-

Apstrāde

Parādās kategorijas rediģēšanas saskarne, kurā, atkarībā no nospiestas pogas, būs aktīvs atbilstošs ievadlauks.

Gadījumā, ja lietotājs izmaina termiņu, tiek pārbaudīti nosacījumi:

- Bija ievadīts korekts datums.
- Datums nav mazāks par šodienas datumu. Kļūdas gadījumā tiek izvadīts paziņojums E1.

Gadījumā, ja lietotājs izmaina aprakstu, pārbauda, vai apraksts nav tukšs (E2).

Izvade

Atjaunojas informācija kategoriju sarakstā.

Kļūdu paziņojumi

E1 – Nedrīkst saīsināt termiņu līdz pagātnes

E2 – Aizpildiet aprakstu.

3.3.3.3. Jaunas kategorijas pievienošana (PTS-CAT-NEW)

Mērķis

Pievienot jaunu kategoriju.

Ievade

1. Lietotājs nospiež <Jauns> pogu.
2. Ievada sekojošus laukus:
 - a) Kods
 - b) Apraksts
 - c) No (Periods)
 - d) Līdz (Periods)
 - e) Prioritāte
3. Nospiež <Saglabāt> pogu.

Apstrāde

Pēc kārtēja lauka aizpildes tiek pārbaudīti atbilstošie nosacījumi:

Kategorijas kods ir unikāls. Kļūdas paziņojums E1.

Apraksts nav tukšs. Kļūdas paziņojums E2.

Datums ir korekts.

Datums ir korekts un nav mazāks par lauka "No" datumu. Kļūdas paziņojums E3.

Prioritāte nav vienāda ar 0. Kļūdas paziņojums E4.

Izvade

Jaunā kategorija parādās kategoriju sarakstā.

Kļūdu paziņojumi

E1 – Ieraksts ar šo kodu jau eksistē.

E2 – Ievadiet aprakstu

E3 – Periods nav ievadīts korekti

E4 – Ievadiet prioritāti $\diamond 0$

3.3.3.4. Kategorijas tarifu apskatīšana (PTS-CAT-RATE)

Mērķis

Noteiktai kategorijai apskatīt/rediģēt tarifus.

Ievade

Kategoriju sarakstā lietotājs izvēlās attiecīgu kategoriju un nospiež pogu <Tarifi> vai "T" uz tastatūras.

Apstrāde

Ja lietotājam ir tiesības rediģēt šīs kategorijas tarifus, atvert saskarni rediģēšanas režīmā, citādi – pārskata režīmā.

Izvade

Parādās saskarne, kas atbilst PTS-RATE apakšsistēmai.

3.3.3.5. Dzēšana (PTS-CAT-DEL)

Mērķis

Dzēst kategoriju

Ievade

Nospieš DEL vai <Dzēst>

Apstrāde

Funkcija saglabā vēsturi par dzēšanu.

Izvade

Atjaunojas kategoriju saraksts.

3.3.4 Kategorijas tarifu pārskats un rediģēšana (PTS-RATE)

3.3.4.1. Jauna tarifa pievienošana (PTS-RATE-ADD)

Mērķis

Pievienot tarifu kategorijai.

Ievade

1. Lietotājs nospiež pogu <Jauns>.
2. Ievada standarta tipa kodu.

Apstrāde

1. Pārbauda, vai kods atbilst standarta tipa tarifam (kļūdas paziņojums E1).
2. Pārbauda, vai tarifs ar tādu kodu vēl nav pievienots kategorijai (kļūdas paziņojums E2).

Tarifs būs pievienots datubāzei tikai pēc <Saglabāt> pogas nospiešanas.

Izvade

E1 – Pārbaudiet tarifa kodu

E2 – Ieraksts ar šo tarifu dotai kategorijai jau eksistē

3.3.4.2. Tarifa rediģēšana (PTS-RATE-EDIT)

Mērķis

Rediģēt tarifa laukus.

Ievade

1. Lietotājs izvēlās sarakstā tarifu un nospiež ENTER.
2. Lietotājam ir iespēja rediģēt sekojošas vērtības:
 - a) Tips
 - b) Konts (bankas sistēmas konts, uz kuru būs pārskaitīta komisija)
 - c) Pakalpojums (apraksts)
 - d) Sum, proc, max, min – komisijas aprēķināšanas parametri
3. Nospiež <Saglabāt> pogu.

Apstrāde

Funkcija atbilstošiem laukiem pārbauda sekojošus nosacījumus:

- a) Tips ir vērtība no {1, 2, 3} (kļūdas paziņojums E3).
- b) Ir ievadīts viens no bankas sistēmas kontiem (kļūdas paziņojums E4).
- c) Pakalpojuma apraksts nav tukšs (kļūdas paziņojums E5)
- d) Min nepārsniedz max (kļūdas paziņojums E6)

Izvade

Informācija sarakstā atjaunojas.

Kļūdu paziņojumi

- E1 – Ievadiet vērtību no 1 līdz 3
- E2 – Konts neeksistē
- E3 – Ievadiet pakalpojuma aprakstu
- E4 – Nekorekti aizpildītas min un max parametri
-

3.3.4.3. Tarifa dzēšana (PTS-RATE-DEL)

Identifikators:

Mērķis

Nodzēst kategorijas tarifu.

Ievade

Lietotājs nospiež DEL.

Apstrāde

Nav apstrādes.

Izvade

Tarifu saraksts atjaunojas.

3.3.4.4. Tarifu vēstures apskatīšana (PTS-RATE-HIS)

Mērķis

Apskatīt tarifu izmaiņu vēsturi.

Ievade

Nospieš <Vēsture> pogu.

Apstrāde

Nav apstrādes.

Izvade

Saraksts ar rediģēšanas vēsturi.

3.3.5. Klienta karšu kategoriju pārskats un rediģēšana (PTS-CARD)

3.3.5.1. Klienta kartes kategorijas pievienošana (PTS-CARD-NEW)

Mērķis

Piešķirt klienta konkrētai kartei kategoriju.

Ievade

Lietotājs izvēlas karti sarakstā, nospiež ENTER un <Add> pogu, ievada parametrus analogiski funkcijai PTS-CUST-NEW

Apstrāde

Analogiskā PTS-CUST-NEW funkcijai

3.3.5.2 Klienta kartes kategorijas dzēšana (PTS-CARD-DEL)

Mērķis

Dzēst kategoriju

Ievade

Lietotājs nospiež <Dzēst> pogu.

Izvade

Kategoriju saraksts atjaunojas.

3.3.6. Automātiskās priekšrocības tarifu piešķiršanas modulis (PTS-AUTO)

3.3.6.1 Senioru kategorijas piešķiršanas funkcija (PTS-AUTO-SEN)

Mērķis

Automātiski piešķirt atlaižu kategoriju klientiem, kuri sasniedz senioru vecumu.

Ievade

Funkcija nostrādā automātiski dienas sākumā, dati tiek ņemti no datubāzes.

Apstrāde

Funkcija papildus pārbauda, ka klients ir fiziska nevis juridiska persona, kā arī saglabā vēsturi par katru piešķiršanas procedūru.

Izvade

Funkcija pievieno informāciju par jaunu senioru skaitu batch atskaitē.

3.3.6.2 Kategoriju ielāde no faila datiem (PTS-AUTO-ICAT)

Mērķis

Funkcija izveido jaunās kategorijas pēc informācijas no .csv faila

Ievade

Funkcija kā parametru pieņem CSV faila nosaukumu. CSV faila katra rindiņa apraksta viena tarifa parametrus.

Apstrāde

Funkcija pārbaude kategorijas parametrus analogiski funkcijai PTS-CAT-NEW

Izvade

Funkcija izveido divus informācijas failus:

- 1) imp_cat_ddmmgg_err.scv – informācija par kļūdām
 - 2) imp_cat_ddmmgg_log.scv – informācija par veiksmīgi pievienotām kategorijām
-

3.3.6.3 Jaunu tarifu ielāde no faila (PTS-AUTO-IRAT)

Mērķis

Funkcija izveido jaunus standarta tipa un priekšrocības tarifus pēc informācijas .csv failā.

Ievade

Funkcija kā parametru pieņem CSV faila nosaukumu. CSV faila katra rindiņa

apraksta viena tarifa parametrus.

Apstrāde

- 1) Funkcija pārbauda, vai katra rindiņa satur korektus datus.
- 2) Ja lauks "category" rindiņā ir aizpildīts, tad funkcija pievieno priekšrocības tarifu, citādi – standarta tipa.

Izvade

Funkcija izveido divus informācijas failus:

- 3) imp_rate_ddmmgg_err.scv – informācija par kļūdām
- 4) imp_rate_ddmmgg_log.scv – informācija par veiksmīgi pievienotiem tarifiem

3.3.6.4. Kategoriju automātiskā noņemšana (PTS-RATE-CLS)

Mērķis

Noņemt klientiem vai kartēm kategorijas, kurām ir beidzies termiņš.

Ievade

Funkcija nostrādā automātiski dienas beigās, dati tiek ņemti no datubāzes.

Apstrāde

Katram klientam un katrai kartei funkcija pārbauda visas piešķirtās kategorijas un noņem, ja izpildās viens no nosacījumiem:

1. Kategorijai ir beidzies termiņš.
2. Konkrētām klientam/kartei ir beidzies termiņš šai kategorijai.

Visas noņemšanas procedūras tiek arī saglabāti vēsturē.

Izvade

Funkcija pievieno informāciju par noņemtu kategoriju skaitu batch atskaitē.

4. PROGRAMMATŪRAS PROJEKTĒJUMA APRĀKSTS

4.1. Ievads

4.1.1. Nolūks

Šī dokumenta nolūks ir aprakstīt Priekšrocības tarifu sistēmas projektējumu atbilstoši prasību specifikācijā izvirzītām prasībām. Dokuments ir paredzēts sistēmas izstrādātājiem. Programmatūras izstrāde tiek veikta stingri pēc šī dokumenta vadlīnijām.

4.1.2. Darbības sfēra

Priekšrocības tarifu sistēma ir domāta bankas darbiniekiem, kuri nodarbojas ar jaunu tarifu ieviešanu un piešķiršanu klientiem. Sistēmas automātiskā funkcionalitāte nozīmīgi atvieglo šo procesu.

4.1.3. Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standarta LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai prasības.

4.1.4. Dokumenta pārskats

Dokumentā ir 6 nodaļas:

1. nodaļa ir dokumenta ievads. Nodaļa satur nolūku, darbības sfēras aprakstu, saistību ar citiem dokumentiem un dokumenta pārskatu.
2. nodaļā ir aprakstīts projekta sadalījums moduļos un funkcijās.
3. nodaļa satur atkarības aprakstu – datu plūsmas diagrammas.
4. nodaļa satur konceptuālo un fizisko ER modeļus.
5. nodaļā ir aprakstīts datubāzes detalizēts projektējums.
6. nodaļā ir attēloti visu moduļu grafisko saskarņu projektējumi.

4.2. Dekompozīcijas apraksts

Sistēma bija sadalīta 6 moduļos atbilstoši programmatūras prasību specifikācijai:

- Komisiju aprēķināšanas modulis (PTS-COM)
- Klienta kategoriju modulis (PTS-CUST)

- Kategoriju modulis (PTS-CAT)
- Tarifu modulis (PTS-RATE)
- Karšu kategoriju modulis (PTS-CARD)
- Automātiskais modulis (PTS-AUTO)

Komisiju aprēķināšanas moduļa funkcijas:

Apraksts	Atbilst PPS funkcijām
Standarta tipa komisijas aprēķināšana	PTS-COM-SR
Priekšrocības tarifa komisijas aprēķināšana	PTS-COM-PR
Komisijas par pakalpojumu aprēķināšana klientam vai kartei	PTS-COM-CC

Klienta kategoriju moduļa funkcijas:

Apraksts	Atbilst PPS funkcijām
Moduļa inicializēšana	nav aprakstīta PPS

Kategoriju moduļa funkcijas:

Apraksts	Atbilst PPS funkcijām
Moduļa inicializēšana	nav aprakstīta PPS
Kategorijas piesaiste objektam	PTS-CUST-NEW, PTS-CARD-NEW
Kategorijas noņemšana objektam	PTS-CUST-CLS, PTS-CARD-DEL
Kategorijas meklēšana	PTS-CAT-FIND
Kategorijas pievienošana/rediģēšana	PTS-CAT-EDIT, PTS-CAT-NEW
Kategorijas dzēšana	PTS-CAT-DEL

Tarifu moduļa funkcijas:

Apraksts	Atbilst PPS funkcijām
Moduļa inicializēšana	PTS-CAT-RATE, PTS-CUST-RATE
Tarifa pievienošana/rediģēšana	PTS-RATE-ADD, PTS-RATE-EDIT
Tarifa vēstures apskatīšana	PTS-RATE-HIS
Tarifa dzēšana	PTS-RATE-DEL

Karšu kategoriju moduļa funkcijas:

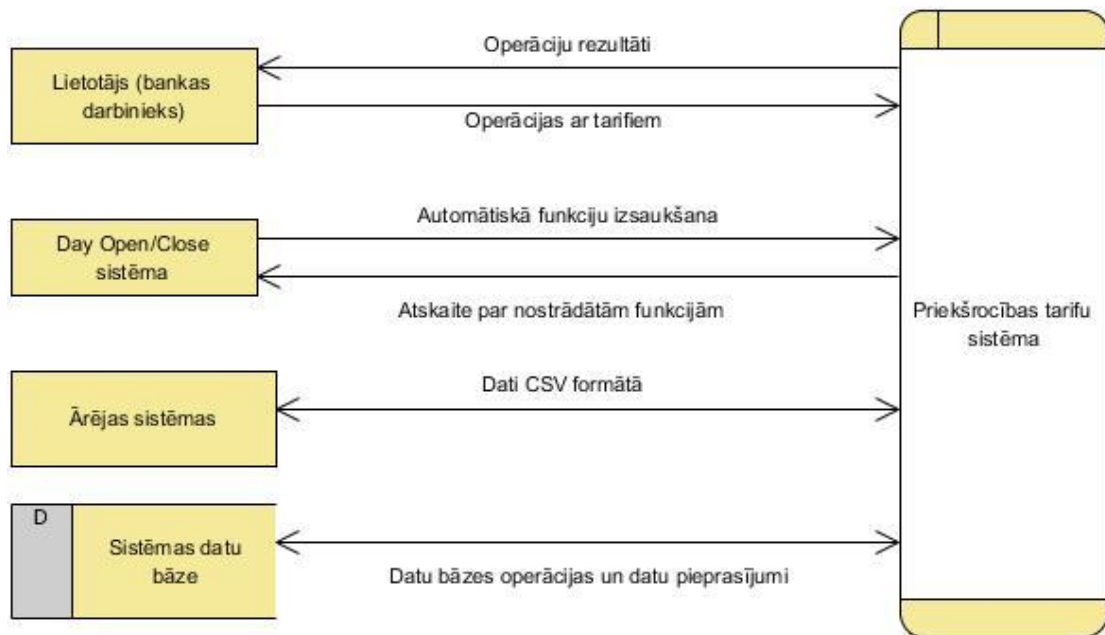
Apraksts	Atbilst PPS funkcijām
Moduļa inicializēšana	PTS-CUST-CARD
Kartes ieturēšanas komisiju apskatīšana	nav aprakstīta PPS

Automātiskā moduļa funkcijas:

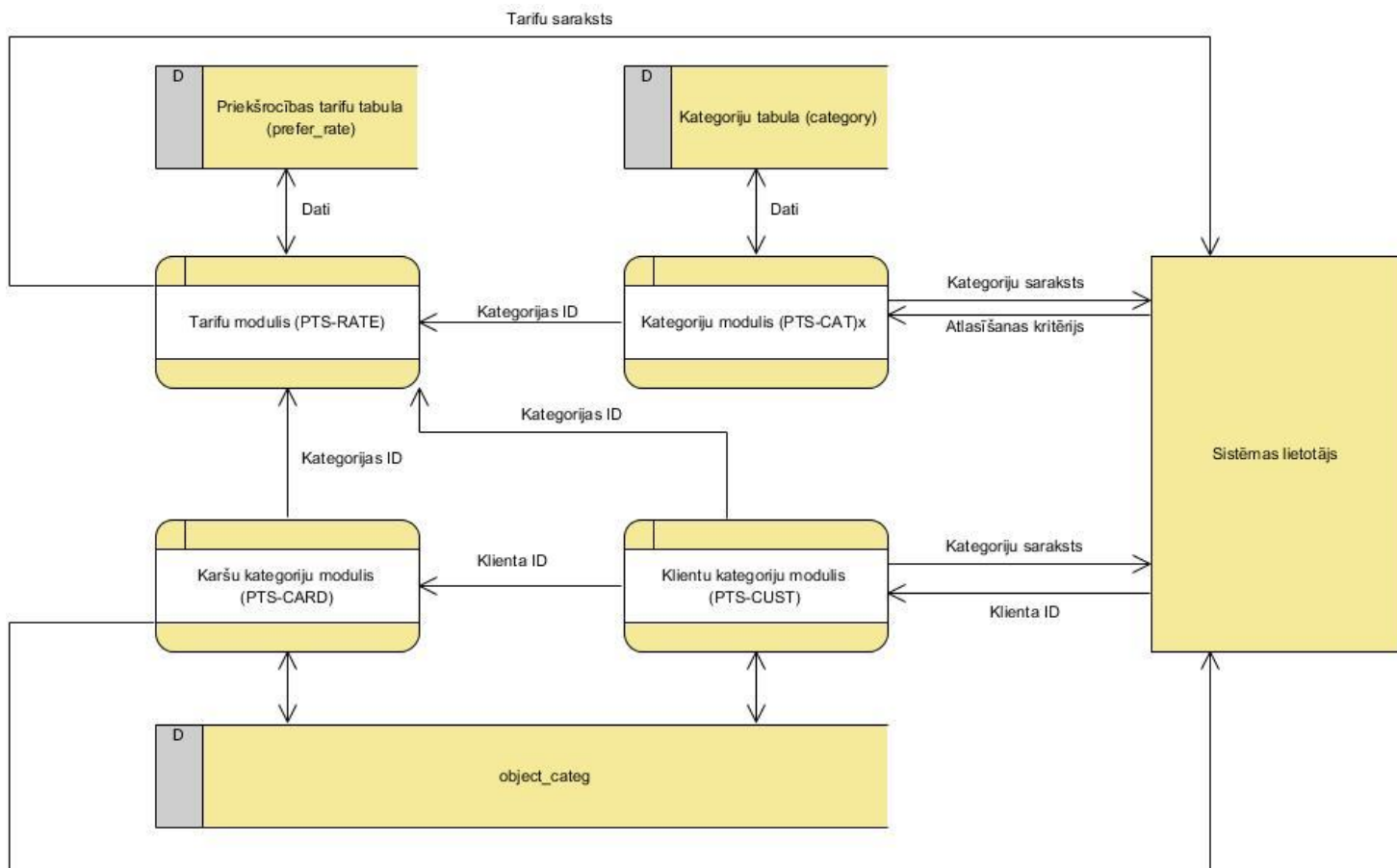
Apraksts	Atbilst PPS funkcijām
Senioru kategorijas piešķiršanas funkcija	PTS-AUTO-SEN
Kategoriju ielāde no faila	PTS-AUTO-ICAT
Tarifu ielāde no faila	PTS-AUTO-IRAT

4.3. Atkarības apraksts

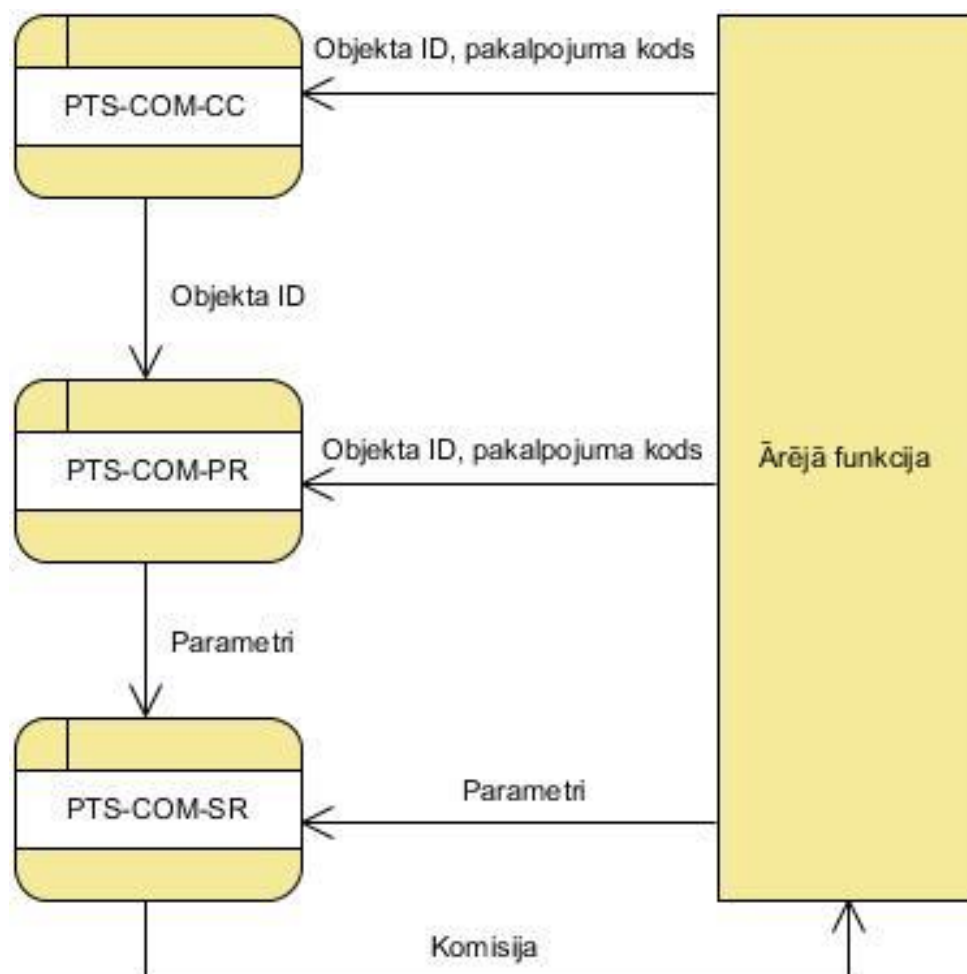
4.3.1 DPD 0. līmenis.



4.3.2. DPD 1. līmenis sistēmas lietotājam

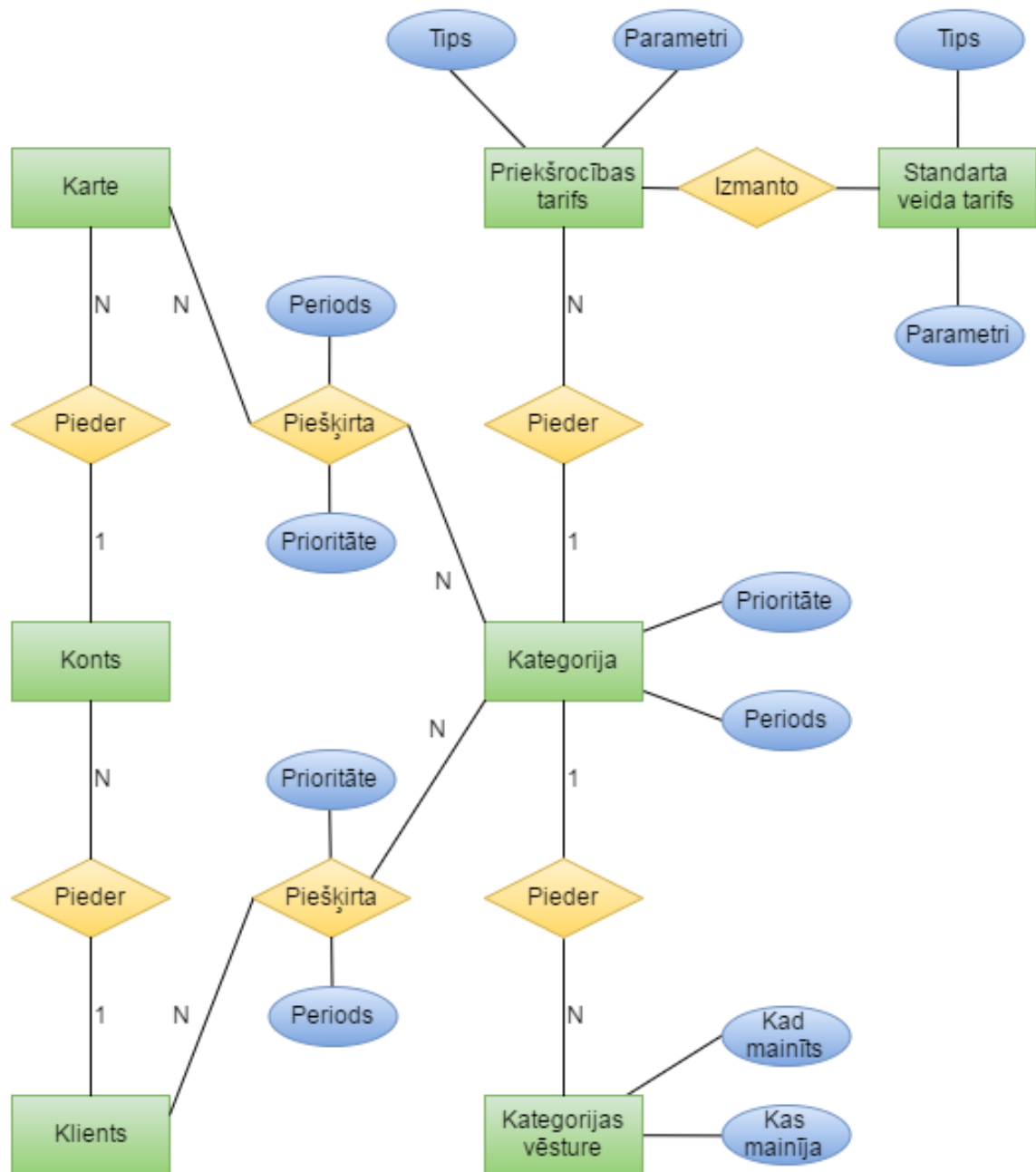


4.3.3. Komisijas aprēķināšanas DPD 2. līmenis.

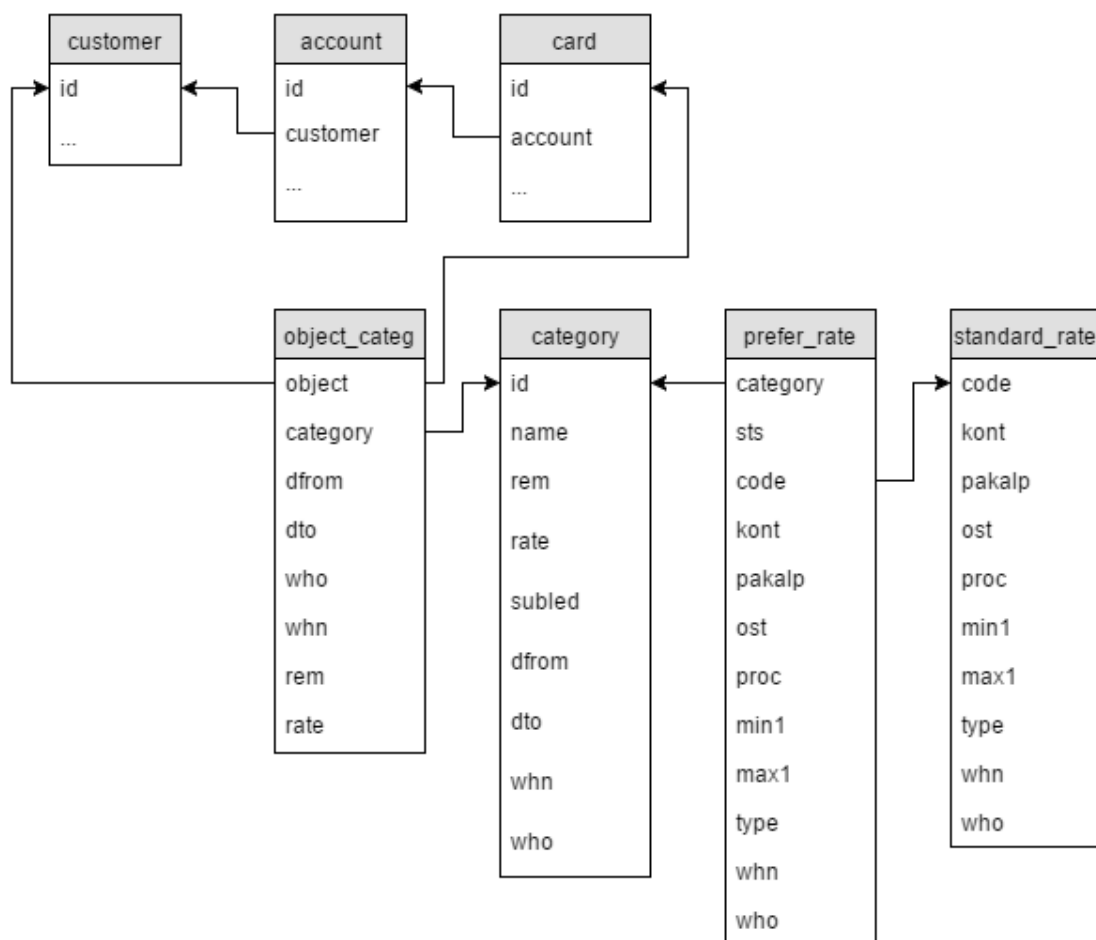


4.4 Datu atkarības

4.4.1. Konceptuālais ER modelis



4.4.2. Fiziskais ER modelis



4.5. Datubāzes detalizēts projektējums

4.5.1. Piezīmes datu tabulām

- Tabulas “customer”, “account” un “card” ir globālas tabulas, kuras ir izmantotas arī citās banka sistēmās, tāpēc šeit būs aprakstīti tikai tie lauki, kuri ir nepieciešami konkrēti šai sistēmai.
- Visu tabulu un lauku nosaukumi bija izmainīti konfidencialitātes dēļ.

4.5.2 „customer” Tabula

Tabulas projektējums

Lauks	datu tips	Izvades formāts pēc noklusējuma	Atļautās vērtības	Obligāts	Apraksts
id	integer (7)	>>9		Visiem klientiem	Klienta ID

4.5.3. “account” Tabula

Tabulas projektējums

Lauks	datu tips	Izvades formāts pēc noklusējuma	Atļautās vērtības	Obligāts	Apraksts
id	Integer (9)	>>9		Visiem kontiem	Konta ID
currency	Integer(2)	>9	0..60	Visiem kontiem	Konta valūta
customer	Integer (7)	>>9		Visiem kontiem	Konta īpašnieks (id)

Tabulas saites

Ārējā atslēga	Saite uz tabulu	Saite uz lauku
customer	customer	id

4.5.4. "card" Tabula

Tabulas projektējums

Lauks	datu tips	Izvades formāts pēc noklusējuma	Atļautās vērtības	Obligāts	Apraksts
id	Int(9)	>>9		Visām kartēm	Kartes unikālais identifikācijas numurs
account	Int(9)	>>9		Visām kartēm	Kartes konts

Tabulas saites

Ārējā atslēga	Saite uz tabulu	Saite uz lauku
account	account	id

4.5.5. "category" Tabula

Tabulas projektējums

Lauks	datu tips	Izvades formāts pēc noklusējuma	Atļautās vērtības	Obligāts	Apraksts
int	int (3)	>>9		Visām kategorijām	Kategorijas unikālais identifikācijas numurs
name	Char	x(20)		Visām kategorijām	Kategorijas nosaukums
rem	Char	x(20)		Visām kategorijām	Kategorijas apraksts
rate	Int(3)	>>>9	0..200	Visām kategorijām	Kategorijas prioritāte
subled	Char	x(8)	CLI, CRD	Nav obligāts	Kategorija klientam/kartei

dfrom	date	99/99/99		Visām kategorijām	Kategorijas iestāšanas datums
dto	date	99/99/99		Nav obligāts	Kategorijas beigšanas datums
whn	date	99/99/99		Visām kategorijām	Kad kategorija izveidota
who	char	x(8)		Visām kategorijām	Kurš izveidoja

4.5.6. "object_categ" Tabula

Tabulas projektējums

Lauks	datu tips	Izvides formāts pēc noklusējuma	Atļautās vērtības	Obligāts	Apraksts
object	Int(9)	>>9		Visiem ierakstiem	Klienta vai kartes numurs
category	Int(3)	>>9		Visiem ierakstiem	Kategorija
dfrom	Date	99/99/99		Visiem ierakstiem	Kategorijas iestāšanas datums noteiktam klientam
dto	Date	99/99/99		Nav obligāts	Kategorijas beigšanas datums noteiktam klientam
who	Char	x(8)		Nav obligāts	Kurš izveidoja ierakstu
whn	Date	99/99/99		Visiem ierakstiem	Kad tika izveidots ieraksts
rem	Char	x(20)		Nav obligāts	Apraksts

rate	Int	->, >>>, >>>9		Nav obligāts	Prioritāte
------	-----	---------------	--	--------------	------------

Tabulas saites

Ārējā atslēga	Saite uz tabulu	Saite uz lauku
object	customer/card	id
object	category	id

4.5.7. "prefer_rate" Tabula

Tabulas projektējums

Lauks	datu tips	Izvades formāts pēc noklusējuma	Atļautās vērtības	Obligāts	Apraksts
category	Int(3)	>>>9		Visiem tarifiem	Tarifa kategorija (id)
code	char	x(4)		Visiem tarifiem	Standarta tarifa identifikātors
kont	Int	zzzzz9		Visiem tarifiem	Bankas sistēmas konts
pakalp	Char	x(60)		Visiem tarifiem	Pakalpojuma apraksts
ost	Decimal 2	->>9.99		1 un 2 tipa tarifiem	Tarifa aprēķināšanas parametrs
proc	Decimal 2	->9.99	0..100	Visiem tarifiem	Tarifa aprēķināšanas parametrs
min1	Decimal 2	->>, >>>9.99		1 un 2 tipa tarifiem	Tarifa aprēķināšanas parametrs
max1	Decimal	->>, >>>9.99		1 un 2	Tarifa

	2			tipa tarifiem	aprēķināšanas parametrs
whn	Date	99/99/99		Visiem tarifiem	Kas mainīja
who	Char	x(20)		Visiem tarifiem	Kad mainīja
sts	char	x(1)		Visiem tarifiem	Statuss
type	int	x(20)		Visiem tarifiem	Tarifa tips

Tabulas saites

Ārējā atslēga	Saite uz tabulu	Saite uz lauku
category	category	id
code	standard_rate	code

4.5.8. "standard_rate" Tabula

Tabulas projektējums

Lauks	datu tips	Izvades formāts pēc noklusējuma	Atļautās vērtības	Obligāts	Apraksts
code	char	x(4)		Visiem tarifiem	Tarifa unikālais identifikators
kont	int	zzzzz9		Visiem tarifiem	Bankas sistēmas konts
pakalp	char	x(60)		Visiem tarifiem	Pakalpojuma apraksts
ost	decimal 2	->>9.99	>0	Nav obligāts	Tarifa aprēķināšanas parametrs
proc	Decimal 2	->9.99	0..100	Visiem tarifiem	Tarifa aprēķināšanas parametrs
max1	Decimal 2	->>, >>9.99	>0	1. tipa tarifiem	Tarifa aprēķināšanas parametrs

min1	Decimal 2	->>, >>9.99	>0	Visiem tarifiem	Tarifa aprēķināšanas parametrs
whn	Date	99/99/99		Visiem tarifiem	Kas mainīja
who	Char	x(8)		Visiem tarifiem	Kad mainīja
type	Char	x(2)	1,2,3	Visiem tarifiem	Tarifa tips

4.6. Lietotāja grafisko saskarņu projektējumi

4.6.1. Piezīmes.

Visā konfidenciālā informācija no saskarņu projektējuma ir noslēpta.

4.6.2. Klienta kategoriju apakšsistēmas sakārņu projektējums

4.6.2.1. Klienta kategoriju saraksta saskarne

```

Edit      Joint  Kateg  Ref.   NewAcc Kodif  EditAc Acc.  Agreem Sign  >
CFENTE   Jaunu klientu un kontu atvēršana          PRAGMA  klevicki 18/04/16
          Klienta  Kategorijas

```

Kods.	Kategorijas nosaukums	No	Līdz	Prio	Piezīme
>VP25Sepa	standarta SEPA maksa	A 01/03/16	31/01/17	150	
KONTS25	norekinu konta apkal	A 01/03/16	31/01/17	160	
IEKSB04	maksājumi DNB bankas	A 01/03/16	31/01/17	170	

< Jauns > < Slēgt > <Tarifi/T> <Karšu tarifi/K> <Vēsture> < Iziet >

F3 +slēgtas

Enter data or press F4 to end.

4.6.2.2. Kategorijas piesaistes klientam saskarne

```

Edit      Joint  Kateg  Ref.   NewAcc Kodif  EditAc Acc.   Agreem Sign  >
CFENTE   Jaunu klientu un kontu atvēršana          PRAGMA  klevicki 18/04/16
      Klienta      Kategorijas

      Teraksts
K
-
>V      Kategorija: PROFES  Profesionāļi
K      Stavoklis: A      (A - aktīvs, C - Slēgts)
I      Ir spēkā no: 18/04/16
      Līdz: 12/11/16
      Piezīme:
      Prioritāte: 100
      Rediģēja:
      Kad:
      < Saglabāt > < Atcelt >

< Jauns > < Slēgt > <Tarifi/T> <Karšu tarifi/K> <Vēsture>
      < Iziet >
  
```

F3 +slēgtas

Enter data or press F4 to end.

4.6.3. Kategoriju apakšsistēmas sakārņu projektējums

4.6.3.1. Kategoriju saraksta saskarne

```

CIFC      Klientu kategoriju ievads un rediģēšana          PRAGMA  klevicki 06/05/16
      Klientu kategorijas

      Kategorijas
      Apraksts
      Pieslēg. Pieslēg. Per Prio Tie P
      Kods      Apraksts      sākums      beigas      iod rit. sība i
      -----
      PROFES  Profesionāļi      10/03/16      0 100 B 1
      > KK BUSCO Business kartes-0.00 maksa 04/03/16      0 100 0
      JURNIEKS Atlaižu komplekts jūrniekiem 03/03/16      0 130 B 0
      VisGolM0 VISA Gold monthly fee bez ma 03/03/16      0 200 0
      VisDMon0 VISA debit monthly bez maksa 03/03/16      0 200 0
      aptieka2 aptieka pagarinājums      02/03/16      0 173 2
      VP75in  ienakosie maksājumi      01/03/16      0 152 2

      < Meklēt > < Rediģēt > < Jauns > <Tarifu piesaiste/T > < Vēsture > < Dzēst >
      < Saīsināt termiņu > <Izmainīt prioritāti > < Iziet >
  
```

4.6.3.2. Kategorijas meklēšanas saskarne

Klientu kategorijas

Kateg. Kods	Kategorijas Apraksts	Pieslēg. sākums	Pieslēg. beigas	Per iod	Prio rit.	Tie sība	P i
> SENIORS	Seniori	26/02/16		0	100		2

Meklēt

Kods sākas ar: _____
Apraksts ietver: senior _____

< Labi > < Atcelt >

4.6.3.3. Kategorijas rediģēšanas/pievienošanas saskarne

Kategorija

Kods : █ _____
Apraksts : _____
No : / /
Līdz : / /
Periods : _____ dienas
Prioritāte : _____
Tiesības rēģistrēt : _____ (B -filiālēs; D -fil.tikai slēgt)

ir piesaistīta: klientiem(-am)
un saistīta ar: tarifiem(-u)

< Saglabāt > < Atcelt >

4.6.4. Kategorijas tarifu apakšsistēmas sakārņu projektējums

4.6.4.1. Tarifu saraksta saskarne

Ar rediģēšanas iespēju

Klientu kategorijas
Tarifu piesaiste

Ktg kod	Trf kod	Konts	Pakalpojumi	Type	Summa	Proc	Mi
> PROFES	110		Visa Debit izgat.(47	1	0.00	0.00	0.0
PROFES	113		Visa Gold izgat.	1	0.00	0.00	0.0
PROFES	164		VisaClassic izgatavo	1	0.00	0.00	0.0
PROFES	601		VISA debit monthly f	1	0.00	0.00	0.0
PROFES	603		VISA Classic monthly	1	1.00	0.00	0.0
PROFES	604		VISA Gold monthly fe	1	2.75	0.00	0.0

< Jaunu > < Dzēst > < Vēsture > < Iziet >

Bez rediģēšanas iespējas

Klienta karšu kategorijas
Tarifu piesaiste

Ktg kod	Trf kod	Konts	Pakalpojumi	Type	Summa	Proc	Mi
SET0001	607		. Galactico VISA Clasi	1	0.30	0.00	0.0
SET0001	609		. MC Standard monthly	1	0.00	0.00	0.0
SET0001	610		. MC Gold monthly fee	1	2.75	0.00	0.0
SET0001	621		. VISA debit annual fe	1	0.00	0.00	0.0
>SET0001	623		. VISA Classic annual	1	0.00	0.00	0.0
SET0001	624		. VISA Gold annual fee	1	2.75	0.00	0.0
SET0001	625		. VISA Platinum annual	1	0.00	0.00	0.0

< Vēsture > < Iziet >

4.6.4.2. Tarifa pievienošanas/rediģēšanas saskarne

Klientu kategorijas
Tarifu piesaiste
Jaunā piesaiste

Kategorija: PROFES
Kods: 169
Type: 2
Konts: [redacted]
Pakalpojumi: VISA sk.nauda
Summa: 0.00
Proc: 0.20
Min: 700.00 Max: 0.00
when: ofc:

< Saglabāt > < Atcelt >

< Jaunu > < Dzēst > < Vēsture > < Iziet >

4.6.4.3. Tarifa rediģēšanas vēstures saskarne

Klientu kategorijas
Izmaiņu vēsture :110 PROFES

Kateg. kods	Objekts	Izpild.	Datums	Apraksts
>PROFES	110	[redacted]	10/03/16 12:31:17	New. Type: 1 Konts:

< Iziet >

4.6.5 Karšu tarifu apakšsistēmas sakārņu projektējums

4.6.5.1. Klienta karšu saraksta saskarne

[redacted] klienta karšu komisijas

Karte	Produkts	Ieturēšanas datums	Apkalp. maksa (EUR)	Aizviet maksa (EUR)	Uzņēmuma Rēģ.Nr.
>442752*****8939	VISA BUSINESS DEBIT	Y 22/01/17	10.00	10.00	[redacted]

<Informācija/<Enter>> <Karšu kategorijas/C> < Iziēt >

4.6.5.2. Kartes komisiju informācijas saskarne

Karte : [redacted] [redacted]
Produkts : VISA DEBIT
Konta numurs: [redacted]

Kl.darba vieta :

Kods	Komisija	Ieturēš. pazīme	Maksas summa(EUR)	Kategorija	Ieturēšanas datums
601	VISA debit monthly fee	Mēneša	0.80		22/05/2016
120	Visa Debit aizviet.		2.00		

< Aizvērt >

4.6.5.3 Klienta karšu kategoriju saskarne

[redacted] klienta karšu kategorijas

Karte	Kartes kategor.	No	Līdz	Prio	Konts	Uzņ.R.Nr. (kartē)
442752*****8166	SET0001	C	01/02/16	05/02/16	101	
>442752*****8166	TestCard	C	01/02/16	01/02/16	100	[redacted]
442752*****8166	TestCard	A	01/02/16		100	-----

<Informācija <Enter>> <Tarifi/T> < Vēsture > < Iziēt >

5. TESTĒŠANĀS DOKUMENTĀCIJA

5.1. Apraksts

Šajā nodaļā ir aprakstīta PTPAS testēšana. Kvalifikācijas darba autora uzdevums bija notestēt komisijas aprēķināšanas un kategoriju moduļus. Vislielākā uzmanība bija pievērsta komisijas aprēķināšanas moduļa testēšanai, jo šīs funkcijas tiek ļoti bieži izmantotas dažādās bankas sistēmās. Testēšanas process ir balstīts uz programmatūrās prasību specifikācijā definētām funkcijām un to sagaidāmiem rezultātiem.

5.2. Testēšanas plāns

Testēšanas plāns ietver:

1. Nepieciešamu testu noteikšana
2. Testu aprakstu sastādīšana
3. Testu ģenerēšana pēc apraksta (tikai komisijas aprēķināšanai).
4. Sagaidāmo rezultātu noteikšana.
5. Sistēmas pārbaude pēc nodefinētiem testiem.
6. Kļūdu izlabošana.

5.3 Testēšanas žurnāls

5.3.1. Komisiju aprēķināšana

Nr	Testa apraksts	Sagaidāmais rezultāts	Testēšanas rezultāts
1.	Abas operāciju valūtas ir EUR, 1. tips	Konvertācija nenotiek, izmantota 1. tipa formula	Izpildās pareizi
2.	Vienā valūta ir EUR, otrā USD, 1. tips	Valūtas konvertējas uz EUR, izmantota 1. tipa formula	Izpildās pareizi
3.	Abas valūtas ir USD, 1. tips	Konvertācija nenotiek, izmantota 1. tipa formula	Izpildās pareizi
4.	Viena valūta ir USD,	Valūtas konvertējas uz	Izpildās pareizi

	otrā RUB, 1. tips	EUR, izmantota 1. tipa formula	
5.	Abas operāciju valūtas ir EUR, 2. tips	Konvertācija nenotiek, izmantota 2. tipa formula	Izpildās pareizi
6.	Vienā valūta ir EUR, otrā USD, 2. tips	Valūtas konvertējas uz EUR, izmantota 2. tipa formula	Izpildās pareizi
7.	Abas valūtas ir USD, 2. tips	Konvertācija nenotiek, izmantota 2. tipa formula	Izpildās pareizi
8.	Viena valūta ir USD, otrā RUB, 2. tips	Valūtas konvertējas uz EUR, izmantota 2. tipa formula	Izpildās pareizi
9.	3. tipa priekšrocības tarifs, 2. tipa standarts tarifs	Tiek ņemti priekšrocības tarifā aprakstīti procenti no standarta 2. tipa tarifa.	Izpildās pareizi
10.	3. tipa priekšrocības tarifs, 1. tipa standarts tarifs	Tiek ņemti priekšrocības tarifā aprakstīti procenti no standarta 2. tipa tarifa.	Izpildās pareizi
11.	Klienta ID, kuram nav piešķirts priekšrocības tarifs šim pakalpojumam.	Komisija aprēķināta pēc standarta tarifa	Izpildās pareizi
12.	Klienta ID, kuram ir piešķirti vairāki priekšrocības tarifi šim pakalpojumam.	Komisija aprēķināta pēc priekšrocības tarifa ar lielāko prioritāti	Izpildās pareizi
13.	Kartes ID, ne kartei, ne kartes īpašniekam nav priekšrocības tarifu pakalpojumam	Komisija aprēķināta pēc standarta tarifa	Izpildās pareizi

14.	Kartes ID, kurai ir priekšrocības tarifs, bet kartes īpašniekam nav	Komisija aprēķināta pēc kartes priekšrocības tarifa	Izpildās pareizi
15.	Gan kartei, gan īpašniekam ir priekšrocības tarifi kategorijai, prioritāte kartei ir lielāka	Komisija aprēķināta pēc kartes priekšrocības tarifa	Izpildās pareizi
16.	Gan kartei, gan īpašniekam ir priekšrocības tarifi kategorijai, prioritāte klientam ir lielāka	Komisija aprēķināta pēc klienta priekšrocības tarifa	Izpildās pareizi
17.	Gan kartei, gan īpašniekam ir priekšrocības tarifi kategorijai, prioritātes vienādas, komisija kartei mazāka	Komisija aprēķināta pēc kartes priekšrocības tarifa	Izpildās pareizi
18.	Gan kartei, gan īpašniekam ir priekšrocības tarifi kategorijai, prioritātes vienādas, komisija klientam mazāka	Komisija aprēķināta pēc klienta priekšrocības tarifa	Izpildās pareizi

5.3.2 Kategoriju modulis

5.3.2.1. FIND

Nr	Testa apraksts	Sagaidāmais rezultāts	Testēšanas rezultāts
19.	Aizpildīt laukus sekojoši: Kods sākas ar – “senio”	Sarakstā paliek vienīga kategorija – ‘SENIORS’	Izpildās pareizi

	Apraksts ietver – “”		
20.	Kods sākas ar – “senio” Apraksts ietver – “jurniek”	Saraksts tukšs, kļūdas paziņojums	Izpildās pareizi
21.	Kods sākas ar – “visa” Apraksts ietver – “”	Sarakstā paliek vairākas kategorijas – visa1, visa2, visa3 u.t.t.	Izpildās pareizi
22.	Kods sākas ar – “visa’ Apraksts ietver “platinum”	Sarakstā paliek tikai kategorijas priekš VISA Platinum kartēm.	Izpildās pareizi
23.	Kods sākas ar – “” Apraksts ietver – “jurniekiem”	Sarakstā paliek vienīga kategorija – “JURNIEKI”	Izpildās pareizi
24.	Pēc meklēšanas nospiest F4	Sarakstā parādās visas kategorijas	Izpildās pareizi

5.3.2.2. NEW/ EDIT

Nr	Testa apraksts	Sagaidāmais rezultāts	Testēšanas rezultāts
25.	Pamēģināt ievadīt eksistējošu kodu un nospiest TAB	Kļūdas paziņojums	Izpildās pareizi
26.	Atstāt koda lauku tukšu un nospiest TAB	Kļūdas paziņojums	Izpildās pareizi
27.	Atstāt apraksta lauku tukšu un nospiest TAB	Kļūdas paziņojums	Izpildās pareizi
28.	Ievadīt datumu "Līdz" mazāku par "No" un nospiest "Saglabāt"	Kļūdas paziņojums	Izpildās pareizi

29.	Atstāt lauku "Prioritāte" tukšu un nospieš TAB	Kļūdas paziņojums	Izpildās pareizi
30.	Nospieš F4	Saskarne aizveras, izmaiņas nenotiek	Izpildās pareizi
31.	Nospieš <Atcelt>	Saskarne aizveras, izmaiņas nenotiek	Izpildās pareizi
32.	Ievadīt korektus datus un nospieš "Saglabāt"	Saskarne aizveras, jaunā kategorija parādās saraksta beigās	Izpildās pareizi

5.3.2.3 DEL

33.	Nospieš DEL	Parādās apstiprinājuma dialoglodziņš	Izpildās pareizi
34.	Nospieš <Dzēst>	Parādās apstiprinājuma dialoglodziņš	Izpildās pareizi
35.	Nospieš <Dzēst> un pēc tam <Atcelt>	Apstiprinājuma dialoglodziņš aizveras, izmaiņas nenotiek.	Izpildās pareizi
36.	Nospieš <Dzēst> un pēc tam F4	Apstiprinājuma dialoglodziņš aizveras, izmaiņas nenotiek.	Izpildās pareizi
37.	Nospieš <Dzēst> un pēc tam <Labi>	Kategorija pazūd no saraksta.	Izpildās pareizi

5.3.2.4 Citi

38.	Pamēģināt nospieš <Tarifu piesaiste> pogu	Atveras tarifu saskarne ar iespēju rediģēt	Izpildās pareizi
39.	Pamēģināt nospieš T uz tastatūras	Atveras tarifu saskarne ar iespēju rediģēt	Izpildās pareizi

40.	Pamēģināt nospiest <Vēsture> pogu	Atveras kategoriju vēstures saskarne	Izpildās pareizi
41.	Nospiest <Iziet>	Saskarne aizveras	Izpildās pareizi

6. PROGRAMMATŪRAS PROJEKTA ORGANIZĀCIJA

Priekšrocības tarifu sistēma bija izstrādāta atbilstoši ūdenskrituma modelim, kas ir labi piemērots bankas sistēmas izstrādei, jo sistēmas prasības tiek precīzi definētas.

Pirmajā fāzē, kurā notika pasūtītāju un IT nodaļas vadītāju sapulces, bija noformulētas sistēmas prasības un izstrādāta programmatūras prasību specifikācija.

Otrajā fāzē notika darbietilpības novērtēšana un sistēmas projektēšana. Bija nedefinēti sistēmas moduļi, izstrādāti datubāzes un lietotāju grafisko saskarņu projektējumi. Programmatūras projektējums tika izstrādāts, ņemot vērā sistēmas tālāku integrēšanu galvenajā bankas uzskaites sistēmā.

Nākamajā fāzē tika veikta sistēmas programmēšana, kurā piedalījās 3 cilvēki. Visas funkcijas bija realizētas, balstoties uz PPA aprakstīto projektējumu.

Programmatūras testēšanā piedalījās gan bankas darbinieki no pasūtītāju puses, gan paši programmētāji. Kad visas atklātās kļūdas bija novērstas, produkts bija uzstādīts produkcijā.

7. KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu produkta kvalitāti, tika veiktas vairākas aktivitātes.

Programmatūras prasību specifikācija un programmatūras projektējuma apraksts bija izstrādāti atbilstoši valsts standartiem.

Programmas koda izstrāde notika pēc programmēšanas labās prakses. Tika ievērots vienots stils, izmantoti saprotami funkciju un mainīgo nosaukumi. Funkciju svarīgākās vietās bija pievienoti komentāri.

Lietotāju saskarnes bija izstrādātas vienotā formātā.

Sistēma bija labi notestēta. Testēšanā piedalījās gan programmētāji, gan pasūtītāji. Sistēmas automātiskā funkcionalitāte arī bija notestēta testa serverī, kas ir identisks reālas sistēmas serverim.

8. KONFIGURĀCIJU PĀRVALDĪBA

Programmatūras pēdējās versijas pirmkodi un visu izmaiņu vēsture bija glabāti centrālā uzņēmuma servera noteiktajā direktoriņā, kurā tiesības rakstīt bija tikai nodaļas vadītājam. Savukārt, programmētāji izstrādei un testēšanai izmantoja citu tam paredzēto serveri, kurā bija instalēta reālas sistēmas kopija. Šajā serverī bija izmantota GIT versijas kontroles sistēma. Izmaiņas centrālajā serverī tika ievietotas tikai pēc svarīgām izmaiņām un tikai pēc testēšanas.

9. DARBIETILPĪBAS NOVĒRTĒJUMS

Darbietilpības novērtējumam bija izmantots algoritmiskais modelis COCOMO Model 2: Intermediate. Ņemot vērā funkciju skaitu, sarežģītību un pieredzi ar līdzīgām OpenEdge programmām, katram modulim bija aprēķināts prognozējams pirmkoda rindiņu skaits.

- Automātiskās piešķiršanas modulis – 500 rindiņas.
- Komisijas aprēķināšanas modulis – 300 rindiņas.
- Klienta kategoriju modulis – 1200 rindiņas.
- Kategoriju modulis – 1200 rindiņas.
- Tarifu modulis – 800 rindiņas.
- Karšu tarifu modulis – 1000 rindiņas.

Kopā – 5000 koda rindiņas.

Izstrādes faktors	Vērtējums	Koeficients
Produkta atribūti		
Produkta drošība	1,15	Augsts
Datu bāzes izmērs	1	Vidējs
Sistēmas sarežģītība	1	Vidējs
Aparatūras saskarnes raksturojumi		
Izpildes laika ierobežojumi	1	Vidējs
Atmiņas ierobežojumi	1	Vidējs
Vides nepastāvība	1	Vidējs
Datora apgrozījuma laiks	1	Vidējs
Personāla atribūti		
Analītiskas spējas	0,86	Augsts
Programmatūras izstrādes pieredze	1	Vidējs

Programmētāja izstrādes spējas	0,86	Augsts
Izstrādes vides pieredze	1	Vidējs
Programmēšanas valodas pieredze	1	Vidējs
Projekta atribūti		
Moderno programmēšanas paņēmieni izmantošana	1	Vidējs
Izstrādes rīku lietojums	1	Vidējs
Izstrādes laika ierobežojumi	1	Vidējs

Darbietlības regulējošā faktora koeficients pēc tabulas datiem ir:

$$EAF = 1.15 * 0.86 * 0.86 = 0.85$$

Projekta tips – organic ($a_i = 3.2$, $b_i = 1.05$).

$$KloC = 5 \text{ (rindiņu skaits / 1000)}$$

Darbietlība personmēnešos –

$$E = a_i * KloC^{b_i} * EAF = 3.2 * 5^{1.05} * 0.85 = 14.74$$

$$14.74 / 3 = 4.9 \text{ (uz vienu personu)}$$

Reālais izstrādes laiks bija aptuveni 4 mēneši, kas par ~20% atšķiras no darbietlības novērtējuma.

10. REZULTĀTI

Darba ietvaros tika izstrādāta un integrēta sistēma, kuras funkcionalitāte atbilst izvirzītām prasībām, kā arī izstrādāta tās pavadošā dokumentācija.

Kvalifikācijas darba autors piedalījās programmatūras projektēšanas, programmēšanas, testēšanas un uzturēšanas posmos.

Izstrādes laikā autors paplašināja zināšanas gan par produkta darbības sfēru – banku sistēmām, gan par sistēmas izstrādē iesaistītajām tehnoloģijām. Izstrādes laikā bija apgūta OpenEdge vide (Openedge ABL, Pacific application server), UNIX operētājsistēma, Git versiju kontroles sistēma.

Tika apgūta darbietilpības novērtēšanas metode COCOMO, ar kuru tika konstatēts, ka darbietilpība atbilst 4,9 personmēnešiem.

11. IZMANTOTI LITERATŪRA UN AVOTI

1. LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis [tiešsaiste].

Pieejams: <http://estudijas.lu.lv/mod/resource/view.php?id=131427>

2. LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai [tiešsaiste].

Pieejams: <http://estudijas.lu.lv/mod/resource/view.php?id=131428>

3. Lekcijas prezentācija [tiešsaiste].

Pieejams: <http://estudijas.lu.lv/mod/resource/view.php?id=197434>

4. Lekcijas prezentācija [tiešsaiste].

Pieejams: <http://estudijas.lu.lv/mod/resource/view.php?id=131444>

5. Elektroniskā terminu vārdnīca [tiešsaiste].

Pieejams: <http://termini.lza.lv/>

6. COCOMO Model Manual [tiešsaiste].

Pieejams:

http://sunset.usc.edu/csse/affiliate/private/COCOMOII_2000_3/modelman.pdf

12. Pielikums

Tarifa aprēķināšanas funkcijas

```
// Tarifu aprēķināšanas funkcijas

def var rr as dec.
def var sum1 like rem.payment.
def var sum2 like rem.payment.
def var sum3 like rem.payment.
def var v-sumkom as dec.
def var v-fee_min as dec.
def var tproc like standart_rate.proc .
def var tmin1 like standart_rate.min1 .
def var tmax1 like standart_rate.max1 .
def var tost like standart_rate.ost .
def var ttype like standart_rate.type .

def var tar2_sum like rem.payment.
def var card_sum as dec .
def var card_catcod like prefer_rate.catcod .
def var card_rate like cust_categ.rate .
def var cust_sum as dec .
def var cust_catcod like prefer_rate.catcod .
def var cust_rate like cust_categ.rate .
def var card_found as logic .
def var cust_found as logic .

// tiek meklēts standarta tarifs pēc koda
find first standart_rate where standart_rate.code = komis no-lock
no-error.
if available standart rate then do :
    pakal = standart_rate.pakal.
    konts = standart_rate.kont .
    tproc = standart_rate.proc .
    tmin1 = standart_rate.min1 .
    tmax1 = standart_rate.max1 .
    tost = standart_rate.ost .
    ttype = standart_rate.type .

    // tiek izsaukta komisijas aprēķināšanas funkcijas ar standarta
parametriem
    {perv.i}
    tar2_sum = v-sum . // v-sum ir funkcijas rezultāts
    res_rate = 0 . // prioritāte standarta komisijai ir 0
    res_cod = "" .

    cust_found = false .
    card_found = false .
    if tcust <> "" then do:
        if length(tcust) eq 16 then do: // pārbauda vai tā ir
karte
        /* card rate comission calculation*/
        run calc_prefer_rate(input tcust, output card_rate,
            output card_sum, output card_catcod, output card_found)
    .
```

```

        /* cust rate comission calculation*/
        for each card where card.card = tcust no-lock,
            each account where account.account = card.bacc no-lock:
                run calc_prefer_rate(input account.cust, output
cust_rate,
                                output cust_sum, output cust_catcod, output
cust_found) .
                    leave .
                end.

        if (cust_found or card_found) then do: // nosacījums, kad
ir izmantots priekšrocības tarifs
            if (not cust_found) or
                (card_found and cust_found and
                (card_rate > cust_rate or
                (card_rate = cust_rate and card_sum < cust_sum )))
// nosacījums, kad rezultāts ir kartes tarifs
            then do:
                /* disp "card" . */
                res_rate = card_rate .
                res_cod = card_catcod .
                v-sum = card_sum .
            end .
            else do: // klienta tarifs
                /* disp "cust" .*/
                res_rate = cust_rate .
                res_cod = cust_catcod .
                v-sum = cust_sum .
            end.
        end.
        else v-sum = tar2_sum . // rezultāts ir standarta veida
tarifs.
    end.
    else do: // nav karte
        /* cust */
        run calc_prefer_rate(input tcust, output cust_rate,
output cust_sum, output cust_catcod, output cust_found) .
        if cust_found then do :
            res_rate = cust_rate .
            res_cod = cust_catcod .
            v-sum = cust_sum .
        end.
        else v-sum = tar2_sum.
    end.
end.
end.
end.

```

```

// priekšrocības tarifa komisijas aprēķināšana

procedure calc_prefer_rate .

    def input parameter ccust like cust.cust .
    def output parameter crate like cust_categ.rate init 0.
    def output parameter csum as dec init 0.
    def output parameter ccatcod like prefer_rate.catcod init "".
    def output parameter cfound as logic init false .

    // meklē aktīvu kategoriju ar lielāko prioritāti
    for each cust_categ where cust_categ.cust eq ccust and
        cust_categ.sts eq "A" no-lock,
        each prefer_rate where prefer_rate.code eq standart_rate.code
and
    prefer_rate.catcod eq cust_categ.catcod and
    prefer_rate.sts ne "C" no-lock break by cust_categ.rate
descending :

        if cust_categ.dfrom le g-today and
            (cust_categ.dto ge g-today or cust_categ.dto eq ?) then do:

                cfound = true .
                ttype = prefer_rate.type .
                if ttype = "3" then do: // 3 tipa komisija
                    csum = round(tar2_sum * prefer_rate.proc / 100, 2) .
                end.
                else do: // standarta tipa komisijas
                    pakal = prefer_rate.pakal.
                    kots = prefer_rate.kots .
                    tproc = prefer_rate.proc .
                    tmin1 = prefer_rate.min1 .
                    tmax1 = prefer_rate.max1 .
                    tost = prefer_rate.ost .
                    {perv.i}
                    csum = v-sum .
                end.

                crate = cust_categ.rate .
                ccatcod = prefer_rate.catcod .

        leave.
    end.
end.

/* debug
display "calculated " ccust crate csum ccatcod cfound.
*/
end .

```

```

//Standarta tipa komisijas aprēķināšana

if kod12 <> kod11 then do: // valūtas atšķiras
    find crc where crc.crc = kod12 no-lock no-error.
    if available crc then do:
// konvertācija

        sum1 = round( crc2nat(crc.crc, paym, 1), crc.decpnt).
        if tproc <> 0 then do :
// 2 tipa formula

            if ttype = "2" then do:
                rr = round((sum1 - tmin1) / 100 * tproc,2) .
                if rr < 0 then rr = 0.
            end.
// 1 tipa formula
            else do:
                rr = round(tproc * sum1 / 100,2).
                if rr < tmin1 then rr = tmin1.
                if rr > tmax1 and tmax1 <> 0 then rr = tmax1.
            end.

        v-sumkom = tost + rr.
    end.
    else v-sumkom = tost.
end.
// konvertācija atpakaļ
find crc where crc.crc = kod11 no-lock no-error.
if available crc then do:
    v-sumkom = round(nat2crc(crc.crc, v-sumkom, 1), crc.decpnt).
    v-sum = v-sumkom.
end.
end. /*kod11<>kod12*/
else do: // valūtas vienādas
    find crc where crc.crc = kod12 no-lock no-error.
    if available crc then do:
        sum1 = round(nat2crc(crc.crc, tmin1, 1), crc.decpnt).
        sum2 = round(nat2crc(crc.crc, tmax1, 1), crc.decpnt).
        sum3 = round(nat2crc(crc.crc, tost, 1), crc.decpnt).
        if tproc<> 0 then do :

            if ttype = "2" then do:
                rr = round((paym - sum1) / 100 * tproc,2) .
                if rr < 0 then rr = 0.
            end.
            else do:
                rr = round(tproc * paym / 100,2).
                if rr < sum1 then rr = sum1.
                if rr > sum2 and sum2 <> 0 then rr = sum2.
            end.
            v-sumkom = sum3 + rr.
        end.
        else v-sumkom = sum3.
    end.
    v-sum = v-sumkom.

end. /*kod11 = kod12*/

```

```

// import rates from csv file

{bfapi.i} // batch library

function runcommand returns char ( v-cmd as char) forwards.
def var v-msg as char.

def var f_path as char .
def var f_name as char.
def var f_dir as char.
def var f_pref as char.

def stream v-in.
def stream v-err.
def stream v-ok.
def stream v-copy.

def var head as char .
def var field_cnt as int init 6 .
def var v-row as char .
def var idx as int .
def var cnt as int.
def var cnt_ok as int.
def var err as char.
def var msg as char.

def var v-code like prefer_rate.code .
def var v-catcod like prefer_rate.catcod .
def var v-ost as char .
def var v-proc as char .
def var v-min as char .
def var v-max as char .

// faila meklēšana
def var k as int.
find first sysc where sysc.sysc = "tarifi" no-lock no-error.
if available sysc then do:
  f_path = sysc.chval .
  k = num-entries(f_path, "/" ) .
  f_name = trim( entry(k, f_path, "/" ) ).
  k = index(f_path, f_name) .
  f_dir = substring(f_path, 1, k - 1) .
  FILE-INFO:FILE-NAME = f_path.
  if FILE-INFO:FILE-TYPE <> ? then
    input stream v-in from value(f_path) .
  else
    return .
end.
else do:
  bf-err("'tarifi' not found in sysc").
  return.
end.
// atskaites failu izveidošana
FILE-INFO:FILE-NAME = f_dir + "arch".
if FILE-INFO:FILE-TYPE = ? then do:
  v-msg = runcommand("mkdir " + f_dir + "arch") .
  if (v-msg ne "") then bf-warn(v-msg) .
end.
FILE-INFO:FILE-NAME = f_dir + "log".
if FILE-INFO:FILE-TYPE = ? then do:

```

```

        v-msg = runcommand("mkdir " + f_dir + "log") .
        if (v-msg ne "") then bf-warn(v-msg) .
end.

// creating directory and file names
f_pref = string(now, "99/99/99 hh:mm:ss").
f_pref = replace(f_pref, "/", "").
f_pref = replace(f_pref, ":", "").
f_pref = replace(f_pref, " ", "-").
output stream v-err to value(f_pref + "err.csv") .
put stream v-err unformatted
"Time;Row;Column;Error;code;catcod;ost;proc;min;max" skip .

import stream v-in unformatted head.

// check file format
if num-entries(head, ";") ne field_cnt then do:
    put stream v-err unformatted "[" now "];;;Wrong file format.
Column number should be "
        string(field_cnt) ";;;" .
    bf-err("Wrong file format") .
    v-msg = runcommand("mv " + f_path + " " + f_dir + "arch/" +
f_pref + f_name) .
    if (v-msg ne "") then bf-warn(v-msg) .
    v-msg = runcommand("mv " + f_pref + "err.csv " + f_dir +
"log/").
    if (v-msg ne "") then bf-warn(v-msg) .
    output stream v-err close.
    input stream v-in close.
    return .
end.

// if format correct, create log file.
output stream v-ok to value(f_pref + "ok.csv") .
put stream v-ok unformatted ";;;Values in database after update;;;"
skip .
put stream v-ok unformatted "Time;Row;code;catcod;ost;proc;min;max"
skip .

/***** main part */
// check decimal number
procedure upd_dec_field.
    def input parameter p-col as int.
    def input parameter p-in as char.
    def input-output parameter p-out as dec.
    def var tmp as dec .
    if (err eq "") and (p-in ne "") then do:
        tmp = dec(p-in) no-error .
        if error-status:error then do:
            err = string(p-col) + ";Not decimal;" .
        end.
        else p-out = tmp .
    end.
end.

idx = 1 . /* first for head */
cnt = 0 .
cnt_ok = 0.

```

```

repeat :
  idx = idx + 1 .
  err = "".
  import stream v-in unformatted v-row. // read csv file line
  if (v-row ne "") then do:
    v-code = trim(entry(1,v-row,";")).
    v-catcod = trim(entry(2,v-row,";")).
    v-ost = trim(entry(3,v-row,";")).
    v-proc = trim(entry(4,v-row,";")).
    v-min = trim(entry(5,v-row,";")).
    v-max = trim(entry(6,v-row,";")).
    if v-catcod eq "" then do: // rate is standart, category not
specified.
// check all fields for correct format
    find first standart_rate where standart_rate.code = v-
code exclusive-lock no-error .
    if available standart_rate then do:
      run upd_dec_field(input 3, input v-ost, input-output
standart_rate.ost) .
      run upd_dec_field(input 4, input v-proc, input-output
standart_rate.proc) .
      run upd_dec_field(input 5, input v-min, input-output
standart_rate.min) .
      run upd_dec_field(input 6, input v-max, input-output
standart_rate.max) .
    end.
    else do:
      err = ";code not found;" .
    end.
  end .
  else do:
    find first prefer_rate where prefer_rate.code eq v-code
and
      prefer_rate.catcod eq v-catcod exclusive-lock no-error
.
    if available prefer_rate then do:
      run upd_dec_field(input 3, input v-ost, input-output
prefer_rate.ost) .
      run upd_dec_field(input 4, input v-proc, input-output
prefer_rate.proc) .
      run upd_dec_field(input 5, input v-min, input-output
prefer_rate.min) .
      run upd_dec_field(input 6, input v-max, input-output
prefer_rate.max) .
    end.
    else do:
      err = ";code or catcod not found;" .
    end.
  end.

  if (err ne "") then do:
    put stream v-err unformatted "[" now "]" idx ";" err v-
row skip.
  end.
  else do:
    put stream v-ok unformatted "[" now "]" idx ";" v-code
";" v-catcod ";".
    if (v-catcod eq "") then
      put stream v-ok unformatted standart_rate.ost ";"
standart_rate.proc ";"

```

```

                standart_rate.min ";" standart_rate.max "" skip .
            else
                put stream v-ok unformatted prefer_rate.ost ";"
prefer_rate.proc ";"
                prefer_rate.min ";" prefer_rate.max skip .
                cnt_ok = cnt_ok + 1 .
            end.
            cnt = cnt + 1 .
        end.
    end.
end.
put stream v-ok unformatted "Total rows (without head):;" cnt
";;;;;;" skip.
put stream v-ok unformatted "Updated rows:;" cnt_ok ";;;;;;" skip.
put stream v-err unformatted "Total rows (without head):;" cnt
";;;;;;" skip.
put stream v-err unformatted "Skipped rows:;" (cnt - cnt_ok)
";;;;;;" skip.
/***** end of main part *****/

// move log files to log directory
v-msg = runcommand("mv " + f_path + " " + f_dir + "arch/" + f_pref +
f_name) .
if (v-msg ne "") then bf-warn(v-msg) .
v-msg = runcommand("mv " + f_pref + "err.csv " + f_pref + "ok.csv "
+ f_dir + "log/").
if (v-msg ne "") then bf-warn(v-msg) .

if (cnt = cnt_ok) then bf-log("All " + string(cnt) + " rows
updated.").
else bf-warn(string(cnt_ok) + " rows updated, " + string(cnt -
cnt_ok) + " errors found.").

```

Kvalifikācijas darbs „**Bankas klientu priekšrocības tarifu piešķiršanas un aprēķināšanas sistēma**” izstrādāts Latvijas Universitātes Datorikas fakultātē.

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

Autors: *Vladislavs Kļevickis* _____ .05.2016.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *Dr. sc. ing. Vadims Žuravļovs* _____ .05.2016.

Recenzents: *M. Dat. Aleksandrs Zeļenkovs*

Darbs iesniegts 30.05.2016.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2016. prot. Nr. _____

Komisijas sekretārs(-e): _____