

LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

**MAKSĀJUMU IMPORTS/EKSPORTS
HANSAWORLD SISTĒMĀ FIDAVISTA
FORMĀTĀ**

KVALIFIKĀCIJAS DARBS

Autors: Edgars Dzenis(ed07008)

Stud. apl. nr.: ed07008

Darba vadītājs: lektors Dainis Dosbergs, Latvijas Universitāte

RĪGA 2011

ANOTĀCIJA

HansaWorld sistēmā līdz šim bija problemātiski ievadīt bankā veiktos maksājumus sistēmas datubāzē un otrādi- visi ieraksti bija manuāli jāievada katrā sistēmā.

Mērķis ir izstrādāt neatkarīgu moduli, ko varētu pievienot jau gatavai HansaWorld instalācijai un ko izmantotu, lai atvieglotu sistēmas lietotājam iepriekšminēto darbu, ļaujot pārnest datus no vienas sistēmas uz otru, izmantojot failu un automatizētu ierakstu sasaisti.

ABSTRACT

It has so far been difficult to enter payments in HansaWorld system which already were entered in bank's system and vice versa- all entries had to be manually entered in each system.

Objective was to create independent module, which would be easy to attach to existing HansaWorld installation. Module would be used to automatize previously mentioned problem, thereby making it easier for user to enter payments.

SATURS

1. IEVADS	6
2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA.....	7
2.1. Ievads.....	7
2.2. Vispārējs apraksts.....	8
2.3. Funkcionālās prasības	9
2.4. Ārējās saskarnes prasības	20
2.5. Nefunkcionālās prasības.....	21
3. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....	23
3.1. Ievads.....	23
3.2. Atkarību apraksts.....	25
3.3. Datu modelis	27
3.4. Lietotāja saskarnes apraksts.....	34
4. TESTĒŠANAS DOKUMENTĀCIJA.....	48
4.1 Ievads.....	48
4.2. Testa datu sagatavošana	48
4.3. Testēšanas rezultāti	48
5. PROJEKTA ORGANIZĀCIJA.....	53
6. KVALITĀTES NODROŠINĀŠANA.....	54
7. KONFIGURĀCIJU PĀRVALDĪBA	55
8. DARBIETILPĪBAS NOVĒRTĒJUMS.....	56
9. IZMANTOTĀ LITERATŪRA	58
10. PROGRAMMATŪRAS KODA DAĻA	59
10.1. Datu definīcijas	59
10.2. Logu definīcijas.....	64
10.3. Importa funkcijas fragments	69
11. DOKUMENTĀRĀ LAPA.....	76

AUTOREFERĀTS

Šajā darbā esmu izstrādājis maksājumu importu, eksportu un datu pareizības pārbaudi HAL(Hansa Application Language) valodā.

1. IEVADS

Maksājumu importēšanas un eksportēšanas mērķis HansaWorld sistēmā ir darba atvieglošana un automatizēšana cilvēkiem, kam nepieciešams bankā veiktos maksājumus fiksēt arī HansaWorld sistēmā un otrādi(HansaWorld sistēmā izveidotos maksājumu uzdevumus vēlāk apmaksāt ar bankas starpniecību).

Lai uzstādītu produktu, lietotājam nepieciešamas zināšanas par HansaWorld sistēmu un Hansa Application Language valodas zināšanas.

Produkts ir izveidots kā papildu modulis, ko var pievienot jebkurai HansaWorld instalācijai.

Lietotājam būs iespēja importēt/eksportēt maksājumus uz/no HansaWorld sistēmas/u, kā arī vēlāk veikt to apstrādi(piesaistīt pie nepieciešamajiem rēķiniem, kontaktiem) un atskaitē aplūkot sevi interesējošos izveidotos ierakstus.

Darbs sastāv no astoņām daļām:

1. Programmatūras prasību specifikācija
2. Programmatūras projektējuma apraksts
3. Testēšanas dokumentācija
4. Projekta organizācija
5. Kvalitātes nodrošināšana
6. Konfigurāciju pārvaldība
7. Darbietilpības novērtējums
8. Programmatūras kods

2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

2.1. Ievads

2.1.1. Nolūks

Dokuments ir paredzēts “Bankas palīga” prasību specifikācijai. To izmantos “Bankas palīga” pasūtītājs un izstrādātājs, lai saskaņotu abu prasības un iespējas.

2.1.2. Darbības sfēra

Programmatūra ir paredzēta ikvienam, kas izmanto HansaWorld sistēmu un ar internetbankas starpniecību apmaksā rēķinus un ir paredzēta, lai atvieglotu un automatizētu šīs darbības lietotājam.

2.1.3. Definīcijas un saīsinājumi

Bankas palīgs – kvalifikācijas darba laikā izstrādātais produkts

Sistēma – HansaWorld Enterprise produkts, kurā iekļauts standarta risinājums un “Bankas palīgs”

Modulis – atsevišķa identificējama programmatūras daļa, kuru var pievienot jau gatavai sistēmai, lai veiktu darbības atbilstošajai sfērai

Reģistrs – ierakstu kopas, kas piesaistītas moduļiem un paredzētās nepieciešamo datu uzglabāšanai. Datu atainošanai tiek lietoti lietotāju saskarnes logi

Ieraksts – viens ieraksts no reģistra, kas glabās konkrētu informāciju. Katram reģistram ir unikāls identifikācijas numurs

Bāzes logs – logs, kas atveras palaižot sistēmu. No bāzes loga var piekļūt pie visiem pārējiem logiem un ierakstiem

Pārskata logs – šajā logā tiek atainoti konkrēti ieraksti

Saraksta logs – šajā logā tiek atainots viss reģistra saturs(visi ieraksti)

Specifikācijas logs – logs, kurā tiek norādīti parametri veicamajai darbībai(eksportam, importam, apkalpošanai vai atskaitei)

Lauks – lauks lietotāja saskarnē, kuru lietotājs var gan būt spējīgs modificēt gan arī nebūt. Atkarībā no sistēmas prasībām

Matrica – tabula no kolonnām, rindām un šūnām datu ievadei

HW – HansaWorld Enterprise, produkts kuram jāpievieno izgatavotā izstrāde

HAL – Hansa Application Language. Uz C bāzēta valoda, kurā tika izstrādāts projekts

FIDAVISTA – Finansiālu datu apmaiņas vienotais standarts. Formāts kuru atbalsta lielāka daļa Latvijas banku. Šī dokumenta ietvaros FIDAVISTA apzīmē arī ierakstu, kas izveidojas veicot datu importu

Importa fails – bankas izraksts fīdavista formātā

Ievietot speciāli – HWE standarta funkcionalitāte- uzspiežot taustiņu kombināciju Cmd+Enter, tiek atvērts logs ar vērtībām ko var ievietot attiecīgajā laukā

2.1.4. Saistība ar citiem dokumentiem

Programmatūras prasību specifikācija tika sastādīta saskaņā ar “LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis” noteiktajām prasībām.

Finansiālu Datu Apmaiņas Vienotais Standarts(FiDAViSta) specifikācija, Versija 1.2.

2.1.5. Dokumenta pārskats

Dokuments sastāv no 3 nodaļām. 1. Nodaļā tiek aprakstīta dotā dokumenta mērķauditorija, tā mērķi un paskaidroti dokumentā izmantotie vārdi. 2. Nodaļā aprakstītas izstrādātās programmatūras funkcijas un prasības. 3. Nodaļā aprakstītas funkcionālās prasības, ārējā saskarne un ar to saistītās pasūtītāja prasības.

2.2. Vispārējs apraksts

2.2.1. Produkta funkcijas

1. Maksājumu imports sistēmā
2. Maksājumu eksports no sistēmas
3. Pareizā sistēmas ieraksta, kontakta atrašana un piesaistīšana ieimportētajiem ierakstiem
3. Izveidoto ierakstu apstrāde un sasaiste ar nepieciešamajiem sistēmas ierakstiem
4. Ieimportēto ierakstu atlase pēc vairākiem parametriem

2.2.2. Prasības lietotājam

Lietotājam nepieciešamas zināšanas darbam ar HansaWorld sistēmu, ar datoram uzstādītās operētājsistēmas specifiku, kā arī neliela apmācība

“Bankas palīga” lietošanai.

2.2.3. Produkta perspektīva

“Bankas palīgs” nav neatkarīga sistēma, bet papildus modulis HW standarta sistēmai, kas ir savietojams arī ar citām izstrādēm kas uzstādītas uz HW sistēmas.

2.2.4. Vispārēji ierobežojumi

“Bankas palīga” lietošanai nepieciešama uzstādīta HW sistēma ar HAL licenci.

2.2.5. Pieņēmumi un atkarības

Sistēma sastāv no divām daļām- servera un klienta, no kurām katra var darboties uz populārākajām operētājsistēmām(Windows, Linux, Mac Os). Klients pie servera var pievienoties caur jebkuru brīvu portu. Klienta un servera programmatūra atšķiras tikai ar uzstādījumiem.

2.3. Funkcionālās prasības

2.3.1. Importa faila datu pārbaude

2.3.1.1. Ievads

Importa faila datu pārbaude, veiks pārbaudi vai importa fails ir pareizā formātā

2.3.1.2. Ievade

Datu ievade notiek failā, nevis caur lietotāja saskarni. Importa failu iespējams izveidot manuāli, izmantojot sev ērtāku teksta editoru(šādā gadījumā pašam jā rūpējas, lai ieraksts būtu atbilstošā formātā) vai arī izmantot bankas izrakstu, kas radies eksporta no bankas laikā. Fails jā saglabā .xml vai .txt formātā, lai varētu veikt importu HWE sistēmā

2.3.1.3. Apstrāde

Funkcijā tiks secīgi pārbaudīts vai fails atbilst FIDAVISTA standartam:

- ierakstā tiks meklēts teksts “/AccountSet[” , kas norāda, ka beigusies faila daļa, kurā glabājas visu maksājumu un dati
- ierakstā tiks meklēts teksts “/CcyStmt[” , kas norāda, ka beigusies faila daļa, kura ir “/AccountSet[” apakšsadaļa, kurā tiek norādīta valūta
- ierakstā tiek meklēts teksts “/TrxSet[” , kas ir iepriekšējos

punktos minēto faila daļu apakšsadaļa, kas apzīmē, ka ir beidzies viens importa failā pierakstīts maksājums(tādējādi apzīmē arī importējamā ieraksta beigas)

2.3.1.4. Izvads

Ja failā būs nepareizi ievadīta informācija, nekādi paziņojumi lietotājam nav paredzēti, jo importējamajam failam ir jābūt FIDAVISTA formātā. Ja importējamais fails būs bankas izraksts, kļūdains fails nav iespējams, taču, ja lietotājs būs manuāli ko mainījis, sistēmai nav iespējams prognozēt kādas izmaiņas lietotājs būs veicis

2.3.2. FIDAVISTA ieraksta izveide importa laikā

2.3.2.1. Ievads

Funkcija nodrošina, ka importa laikā automātiski tiek izveidots jauns FIDAVISTA ieraksts

2.3.2.2. Ievade

Pēc noklusējuma lietotājam nav ļauts mainīt nevienu no FIDAVISTA ieraksta laukiem, tādējādi pasargājot sistēmu no nepareizu datu ievades. Pastāv iespēja nomainīt kontaktu(kontaktu no sistēmas datubāzes- maksājuma veicēju vai saņēmēju), kas piesaistīts ierakstam, izmantojot iestatījumu FIDAVISTA parametri. Šī iespēja paredzēta, lai ierakstam varētu piesaistīt kādu jau sistēmā reģistrētu kontaktu

2.3.2.3. Apstrāde

Jaunu ierakstu veidošana notiks automātiski importa laikā. Ieraksts tiks izveidots tajā brīdī, kad tiks secināts, ka tas ir pierakstīts pareizā formātā

- tiks izveidots unikāls ierakstu identificējošs identifikators md5hash
- veidojot ierakstu, tam tiks aizpildīti lauki, kas identificēs konkrēto importu- importa datums, importa laiks un importa veicējs
- pārējie ieraksta dati tiks aizpildīti ar attiecīgo informāciju no importa faila(kā jau iepriekš minēts, vienu maksājumu ierobežo "TrxSet")

2.3.2.4. Izvads

Lietotājam netiek izvadīts nekāds paziņojums pēc katra ieraksta izveides(ja lietotājam tiktu izvadīts paziņojums, tas būtu visai neērti lietotājam, jo lielā daļā gadījumu tiek importēts vairāk par 100 ierakstiem vienlaicīgi), taču izveidotajā ierakstā tiks ierakstīta informācija, kas noticis ar ierakstu importa laikā(vai tas veiksmīgi izveidots, atrasts kontakts sistēmā, utt). Pēc visa faila importa tiks izvadīts paziņojums par to cik ierakstiem vajadzēja tikt izveidotiem un cik ieraksti tika izveidoti. Kā arī ir iespējams aplūkot importa laikā izveidotos ierakstus, izmantojot atskaiti “FIDAVISTA statistika”

2.3.3. FIDAVISTA ieraksta piesaista pareizajam sistēmas ierakstam

2.3.3.1. Ievads

Funkcijai importa laikā automātiski jāatrod sistēmā reģistrēti maksājumi, kas atbilst ieimportētajā ierakstā norādītajai informācijai, atkarībā no lietotāja norādītajiem iestatījumiem

2.3.3.2. Ievade

- Iestatījumā “FIDAVISTA – aprakstu uzstādījumi” lietotājs var norādīt kādā formātā ieraksta laukā “Maks. mērķis” var būt norādīts rēķina vai pavadzīmes numurs.
- Iestatījumā “FIDAVISTA – Bankas kontu piesaiste kontiem” lietotājs var norādīt kādam bankas kontam jāpiesaista kāds finanšu kots(grāmatojumiem)

2.3.3.3. Apstrāde

Funkcija, katra ieraksta izveides laikā, mēģinās sameklēt ierakstu sistēmā, ko ieimportētais ieraksts apmaksā. Meklēšana notiks pēc augstāk norādītajiem parametriem, tādējādi katrā instalācijā tie var atšķirties

2.3.3.4. Izvads

Lietotājam netiks izvadīts nekāds paziņojums interfeisā, taču tiks ierakstīta informācija, kas notikusi ar ierakstu, pašā ieraksta “logā”

2.3.4. FIDAVISTA ieraksta piesaiste pareizajam sistēmas klientam

2.3.4.1. Ievads

Funkcijai jāatrod ierakstā norādītais kontakts sistēmas ierakstos, ja tur tāds ir reģistrēts. Funkcijai jāizpildās tikai gadījumos, ja nav jau atrasts

atbilstošais ieraksts sistēmā(aprakstīts 2.3.3. punktā)

2.3.4.2. Ievade

Lietotājam nav paredzēta nekāda datu ievade šai funkcionalitātei

2.3.4.3. Apstrāde

Importa laikā, kad tiek izveidots konkrētais ieraksts, gadījumā, ja netika atrasts konkrētajam maksājumam atbilstošs rēķins, tiek meklēts sistēmā esošs kontakts vai nu pēc bankas konta vai reģistrācijas numura

- Bankas konts tiek meklēts tāds pats kā FIDAVISTA ierakstā
- Reģistrācijas numurs tiks meklēts pēc sistēmas pamatiestatījuma "Sistēma>>PVN reģ. nr. šabloni". Funkcijai jānodrošina, ka tiks atpazīti reģistrācijas numuri, kuros " " būs aizstāts ar "-" un otrādi kā arī reģistrācijas numuri, kuros nesakrīt simbolu skaits(sistēmas iestatījumā un importa failā)

2.3.4.4. Izvads

Paziņojums tiks izvadīts FIDAVISTA ieraksta "logā"- attiecīgi vai nu kontakts sistēmā ir atrasts vai nav

2.3.5. Maksājumu uzdevumu eksports

2.3.5.1. Vairāku maksājumu uzdevumu eksports

2.3.5.1.1. Ievads

Jāizveido eksports ar kura palīdzību varēs eksportēt maksājumu uzdevumus no HWE sistēmas uz failu

2.3.5.1.2. Ievade

Lietotājs varēs ievadīt eksporta specifikācijā attiecīgu informāciju(kā arī norādīt faila atrašanās vietu un tā nosaukumu):

- Maksājumu uzdevumi- maksājumu uzdevumu periods(jāatdala ar ":",")
- Periods- laika periods, kurā reģistrēties maksājumi jāpievieno eksporta failam
- Piegādātājs- ja norādīts, tad tiks eksportēti tikai norādītā piegādātāja maksājumu uzdevumi
- Tikai izrakstītos- ja atzīmēts ķeksis, tad tiks eksportēti

tikai tie maksājumu uzdevumi kuri ir izrakstīti

- Tikai neeksportētos- tiks eksportēti tikai tie maksājumu uzdevumi, kuri vēl nav eksportēti. Šim laukam sistēmas ierakstā jātiek mainītai vērtībai, kad tam tiek veikts FIDAVISTA eksports, lai būtu iespējams noteikt vai konkrētais ieraksts jau ticis eksportēts

2.3.5.1.3. Apstrāde

Tiks palaista eksporta funkcija, kas izveidos jaunu failu pēc FIDAVISTA standartiem (tādējādi failu, ja nepieciešams, būs iespējams arī ieimportēt citā vai tajā pašā HWE instalācijā) ar visiem maksājumu uzdevumu ierakstiem, kas atbilst eksporta specifikācijā norādītajiem parametriem

2.3.5.1.4. Izvads

Eksporta laikā atvērsies logs “Uzgaidiet: tiek aprēķināta atskaite”. Kad logs aizvērsies, dati būs tikuši eksportēti un lietotāja norādītajā direktoriņā būs fails ar lietotāja piešķirto nosaukumu

2.3.5.2. Viena maksājumu uzdevuma eksports

2.3.5.2.1. Ievads

Nepieciešama funkcionalitāte, kas ļautu lietotājam eksportēt maksājumu uzdevumu no tā pārskata loga. Šī funkcionalitāte paredzēta, lai atvieglotu eksportu, ja nepieciešams tajā iekļaut tikai vienu konkrētu maksājuma uzdevumu

2.3.5.2.2. Ievade

Jauna poga maksājuma uzdevuma ierakstā, kuru uzspiežot tiks veikts maksājumu uzdevuma eksports, kā arī norādīt eksporta laikā izveidotā faila atrašanās vietu un nosaukumu

2.3.5.2.3. Apstrāde

Nospiežot pogu, tiks izveidots eksporta fails, kurā būs tikai konkrētā maksājumu uzdevuma dati.

2.3.5.2.4. Izvads

Tiks atvērts logs “Uzgaidiet: tiek aprēķināta atskaite”. Kad logs aizvērsies, būs izveidots fails ar maksājuma uzdevuma informāciju norādītajā direktoriņā ar norādīto nosaukumu

2.3.6. Atskaite, kas attēlos uz ekrāna lietotājam nepieciešamos

FIDAVISTA ierakstus

2.3.6.1. Ievads

Nepieciešams izveidot atskaiti ar kuras palīdzību lietotājam būtu iespējams aplūkot importa laikā izveidotos ierakstus un to statusus

2.3.6.2. Ievade

Atskaites specifikācijas logā lietotājs varēs ievadīt sekojošus parametrus:

- Importa periods- laika periods kurā veikts imports
- Importa laiks- laukā var vai nu manuāli ierakstīt kādu laiku(ja tiek ievadīts laiks, tad tiks rādīti tikai ieraksti, kas ir veikti TIEŠI norādītajā laikā ar precizitāti līdz sekundeī) vai arī izmantot Paste special(šajā gadījumā izvēles logā jātiek ievietotiem pēdējiem 10 importiem) un izvēloties kādu no vērtībām, automātiski tiks aizpildīti lauki Importa periods(ar import datumu) un Importa laiks(ar izvēlētajā importa laiku)
- Oper. Tips- manuāli aizpildāms lauks. Ja tiks aizpildīts, tiks rādīti tikai tie FIDAVISTA ieraksti, kuriem laukā “Op.tips” būs tāds pats kods kā šeit norādītajam
- Rādīt- trīs ķekšu izvēlne. Atskaitē jārāda tikai tie ieraksti, kuru statuss būs tāds pats kā kāds no izvēlētajiem ķekšiem. Tādējādi, ja lietotājs, nebūs atzīmējis nevienu, atskaite neuzrādīs nevienu ierakstu

2.3.6.3. Apstrāde

Funkcijai jāskrien cauri visiem FIDAVISTA ierakstiem, meklējot tos, kas atbilst atskaites specifikācijai un jāparāda tie visi atskaitē.

Uzspiežot uz kāda ieraksta numura, lietotājam jāatveras konkrētajam ierakstam, lai varētu ar to veikt nepieciešamās darbības

2.3.6.4. Izvads

2.3.7. Pārbaude, kas seko līdzī vai importējot tiek ieimportēti visi maksājumi, kas atrodas importa failā

2.3.7.1. Ievads

Nepieciešama funkcija, kas informētu lietotāju par importa

sekmīgumu. Tas ir, cik ieraksti tika atrasti importa failā un cik ieraksti tika izveidoti importa laikā

2.3.7.2. Ievade

Funkcionalitātei nav nepieciešama datu ievade no lietotāja puses

2.3.7.3. Apstrāde

Tā kā pastāv iespēja, ka faila formātu konvertācijas(importa laikā faila informācija tiek konvertēta no area tipa uz xml tipu) laikā var tikt pazaudēta informācija, tiks skaitīts ierakstu daudzums kāds ir brīdī kad fails ir area formātā kā arī izveidoto ierakstu skaits

2.3.7.4. Izvads

Pēc importa funkcijas izpildes, lietotājam tiks izvadīts paziņojums pa to cik ieraksti tika atrasti importa failā un cik ieraksti tika izveidoti importa laikā

2.3.8. Iestatījums ar kura palīdzību noteikt kādi bankas konti jāpiesaista kādiem finanšu kontiem

2.3.8.1. Ievads

Nepieciešams izveidot iestatījumu, ar kura palīdzību varēs noteikt kādiem bankas kontiem jāpiesaista kāds finanšu konts, veidojot grāmatojumus

2.3.8.2. Ievade

Iestatījumā “FIDAVISTA – bankas kontu piesaiste kontiem” aizpildāmi lauki:

- Bankas konts- bankas konta nr., kas tiks salīdzināts ar FiDAViSta ieraksta konta nr.
- Raksts(pattern)- 'maska' kāda tiks meklēta FiDAViSta ieraksta laukā 'Maks.mērķis'. Formāts ir sekojošs:
 - A – jebkurš latīņu alfabēta burts
 - X – skaitlis no 0 līdz 9
 - ‘ - zvaigznīte (*)
 - * - jebkurš simbols
 - jebkurš cits simbols – tieši norādītais simbols
- Konts- ar paste special iespējams izvēlēties kādu kontu piesaistīt izveidotajam grāmatojumam

- Objekti- ar paste special iespējams izvēlēties kādu objektu piesaistīt izveidotajam grāmatojumam
- Komentārs- manuāli aizpildāms lauks, lai identificētu konkrēto rindu

2.3.8.3. *Apstrāde*

Ja pēc FIDAVISTA ieraksta lauka “Op.tips” ierakstam nebūs jāveido ne maksājums, ne maksājuma uzdevums un būs ieslēgta grāmatojumu veidošana, tiks veidots jauns sistēmas grāmatojums. Veidojot grāmatojumu, tiks skatīts FIDAVISTA ieraksta bankas konts un pēc tā noteiks kura iestatījuma rinda jāizmanto, lai piesaistītu ierakstam atbilstošo kontu un objektu

2.3.8.4. *Izvads*

Ja tiks izveidots jauns grāmatojums, FIDAVISTA ieraksta “logā” tiks ievadīts atbilstošs paziņojums. Ja ierakstā atrastajam bankas kontam nebūs piesaistīts finanšu konts, arī tad “logā” tiks ievadīta atbilstoša informācija(ka nav norādīta darījuma pretējā puse)

2.3.9. Iestatījums kurā norādīt kā var atpazīt sistēmā reģistrētu rēķinu vai pavadzīmi

2.3.9.1. *Ievads*

Nepieciešams izveidot jaunu iestatījumu, kas nodrošinātu funkcionalitāti, lai importa laikā sistēmā tiktu meklēts konkrētajam FIDAVISTA ierakstam atbilstošs rēķins HWE sistēmā

2.3.9.2. *Ievade*

Iestatījumā “FIDAVISTA – aprakstu uzstādījumi” aizpildāmi lauki:

- Raksts (pattern) – šeit tiek definēts kāds raksts tiks meklēts.

Formāts ir sekojošs:

- A – jebkurš latīņu alfabēta burts
- X – skaitlis no 0 līdz 9
- ‘ - zvaigznīte (*)
- * - jebkurš simbols
- jebkurš cits simbols – tieši norādītais simbols
- Tips – kurā rēķina laukā jāmeklē nedefinētais raksts. Ja tips būs “Rēķins”, tad raksts tiks meklēts rēķina laukā “Nr.”, ja tips būs

“Pavadzīme” – meklēts tiks laukā “Pavadzīmes Nr.”

- Komentārs – lauks, kas paredzēts lietotājam, lai aprakstītu konkrēto iestatījuma rindu

2.3.9.3. Apstrāde

Ja iestatījumos būs ieslēgta rēķina meklēšana, tad izveidojot katru ierakstu, tam tiks meklēts tam piesaistīts rēķins HWE sistēmā.

FIDAVISTA laukā “Maks.mērķis” pēc iestatījumā norādītajiem parametriem tiks meklēts rēķina vai pavadzīmes numurs. Ja tāds tiks atrasts, tad tāds pats numurs tiks meklēts HWE sistēmā. Ja arī sistēmā, tāds tiks atrasts, ieraksti tiks sasaistīti

2.3.9.4. Izvads

FIDAVISTA ieraksta “logā” tiks ievadīts vai ierakstam tika vai netika atrasts atbilstošais rēķins un pēc kādiem parametriem rēķins atbilst FIDAVISTA ierakstam(piem, sakrīt rēķina summa)

2.3.10. Iestatījums kurā var norādīt kādos gadījumos un kādi sistēmas ieraksti ir jāveido importa laikā vai veicot konkrētas darbības

2.3.10.1. Ievads

Nepieciešams izveidot iestatījumu, kurā varēs ievadīt nozīmīgākos ierakstu veidošanas parametrus. Gan importa ierakstiem, gan eksporta ierakstiem, gan jauna kontakta izveides parametrus

2.3.10.2. Ievade

Iestatījumā “FIDAVISTA – parametri” lietotājs varēs aizpildīt sekojošus laukus:

2.3.10.2.1. Importa parametri

- Importa mape– mape, kurā glabāsies importa faili(visu mapju atrašanās vieta tiek norādīta attiecībā pret HWE sistēmas direktoriju)
- Mape pēc importa- mape, kurā glabāsies importa laikā izveidotie faili
- Eksporta mape- mape, kurā glabāsies eksporta laikā izveidotie faili
- Importa failu kodējums- faila kodējums kādā tiks mēģināts lasīt import failu

2.3.10.2.2. Maksājumi

- Grāmatot- ja atzīmēts, tiks veidoti grāmatojumi visiem izveidotajiem maksājumiem
- Piesaistīt, ja sakrīt rēķina summa - tiks izveidots maksājums, ja FiDAViSta ieraksta summa būs vienāda ar kāda neapmaksāta rēķina summu
- Piesaistīt, ja sakrīt kopējā parāda summa - tiks izveidots maksājums, ja FiDAViSta ieraksta summa sakrītīs ar atrastā klienta kopējā parāda summu
- Piesaistīt, ja nesakrīt valūta identificētam rēķinam - rēķins tiks atpazīts arī gadījumos, kad nesakrītīs valūta ar izveidoto FiDAViSta ierakstu
- Neatšifrēti maksājumi apmaksā vecākos rēķinus - tiks izveidots maksājums, kas apmaksās tos atrastā klienta neapmaksātos rēķinus(sākot no vecākā), kas iekļaujas FiDAViSta ieraksta summā. Ja FiDAViSta ieraksta summa būs lielāka kā visi neapmaksātie klienta rēķini, maksājumā tiks iekļauta priekšapmaksa par atlikušo summu

2.3.10.2.3. Maksājumu uzdevumi

- Grāmatot - ja atzīmēts, tiks veidoti grāmatojumi visiem izveidotajiem maksājuma uzdevumiem
- Piesaistīt, ja sakrīt rēķina summa - tiks izveidots maksājuma uzdevums, ja FiDAViSta ieraksta summa būs vienāda ar kāda neapmaksāta rēķina summu
- Piesaistīt, ja sakrīt kopējā parāda summa - tiks izveidots maksājuma uzdevums, ja FiDAViSta ieraksta summa sakrītīs ar atrastā klienta kopējā parāda summu
- Neatšifrēti maksājumu uzdevumi apmaksā vecākos rēķinus - tiks izveidots maksājuma uzdevums, kas apmaksās tos atrastā klienta neapmaksātos rēķinus(sākot no vecākā), kas iekļaujas FiDAViSta ieraksta summā. Ja FiDAViSta ieraksta summa būs

lielāka kā visi neapmaksātie klienta rēķini, maksājuma uzdevumā tiks iekļauta priekšapmaksā par atlikušo summu

- Importējot veidot- tiek norādīts kāda veida ieraksti drīkst tikt veidoti importa laikā
- Maksājumu uzdevumus - tiks veidota priekšapmaksā, ja tiks atpazīts klients, bet netiks atrasts ierakstam atbilstošs rēķins
- Maksājumu uzdevumā eksportēt- kādu informāciju eksportēt maksājumu uzdevuma komentāra laukā

2.3.10.2.4. Kontaktu veidošana

- Veidot kontaktu, ja FV ierakstā nav reģ.nr. - no ieraksta tiks ļauts veidot jaunu kontaktu, ja FV ierakstā nebūs norādīts kontakta reģ.nr.
- Veidot kontaktu, ja FV ierakstā nav bankas konta - no ieraksta tiks ļauts veidot jaunu kontaktu, ja FV ierakstā nebūs norādīts kontakta bankas konts
- Pēc izveidošanas atvērt kontakta ierakstu - pēc jauna kontakta izveidošanas tiks atvērts jaunizveidotais kontakta ieraksts, ļaujot veikt tūlītējus labojumus
- Ļaut piesaistīt ierakstam esošu kontaktu - ļauj mainīt FV ieraksta lauku "klienta kods"

2.3.10.3. Apstrāde

- Visu mapju ceļš jānorāda attiecībā pret HWE sistēmas direktoriju!
- Mapes ir jāizveido manuāli, sevis norādītajā direktorijā!
- Ja mapes nebūs norādītas un izveidotas, dati netiks importēti!
- Ja lauks "Importa failu kodējums" netiks aizpildīts, tiks izmantots UTF-8 kodējums
- Lai importējot tiktu veidoti maksājumi, jābūt atzīmētiem laukiem – Importējot veidot>>Maksājumus un Ja nekas nav uzminēts, veidot priekšapmaksas>>Maksājumus
- Lai importējot tiktu veidoti maksājumu uzdevumi, jābūt

atzīmētiem laukiem – Importējot veidot>>Maksājumu uzdevumus un Ja nekas nav uzminēts, veidot priekšapmaksas>>Maksājumu uzdevumus

2.3.10.4. Izvads

Saglabājot ierakstu, tiks saglabāti tābrīža iestatījumi. Šie iestatījumi arī noteiks, kas vēlāk veidojot FIDAVISTA ierakstus tiks rakstīts FIDAVISTA ierakstu “logos”

2.4. Ārējās saskarnes prasības

2.4.1. Lietotāja saskarnes prasības

Bāzes logā, moduļu sarakstā jābūt pievienotam modulim “Bankas palīgs”

2.4.1.1. Reģistri

Bankas palīga modulī jābūt sekojošiem reģistriem:

- FIDAVISTA- jauns reģistrs, kurā jābūt laukiem, kas parāda informāciju par maksājuma veicēju, maksājuma saņēmēju, ieraksta statusu un importa datiem
- Kontakti- standarta reģistrs
- Rēķini- standarta reģistrs
- Maksājumi- standarta reģistrs
- Ienākošie rēķini- standarta reģistrs
- Maksājumu uzdevumi- standarta reģistrs

2.4.1.2. Iestatījumi

Bankas palīga modulī jābūt sekojošiem iestatījumiem:

- FIDAVISTA – aprakstu uzstādījumi- iestatījumā jābūt laukiem atbilstošiem punktam 2.3.9.
- FIDAVISTA – parametri- iestatījumā jābūt laukiem atbilstošiem punktam 2.3.10.
- FIDAVISTA – bankas kontu piesaiste kontiem- iestatījumā jābūt laukiem atbilstošiem punktam 2.3.8.

2.4.1.3. Eksporti

Bankas palīga modulī jābūt sekojošiem eksportiem:

- Maksājumu uzdevumu eksports(FIDAVISTA)- eksportā jābūt laukiem atbilstoši punktam 2.3.5.

2.4.1.4. Importi

Bankas palīga modulī jābūt sekojošiem importiem:

- FIDAVISTA

2.4.1.5. Atskaites

Bankas palīga modulī jābūt sekojošām atskaitēm:

- FIDAVISTA statistika- atskaitē jābūt laukiem atbilstoši punktam 2.3.6.

2.4.1.6. Kļūdu paziņojum logi

Kļūdu paziņojumu logiem jāatbilst kļūdas būtība

2.4.1.7. Speciālās ievietošanas logs importa specifikācijas logā specifiska importa izvēlei

Logā jāredzama ne vairāk kā pēdējie 10 sistēmā veiktie importi un jānodrošina, ka izvēloties konkrētu importu tiktu ievietota tād informācija, kas viennozīmīgi identificē izvēlēto importu

2.4.2. Aparatūras saskarne

Sistēma vienlīdzīgi darbojas gan uz Windows, gan Linux un MacOS operētājsistēmām. Jaunizveidotos reģistrus tāpat kā jebkuru citu HansaWorld sistēmas reģistru jāvar lietot arī no darbam atbilstošajām mobilajām ierīcēm

2.4.3. Programmatūras saskarne

Programmatūra prot eksportēt un iekopēt atmiņā datus no sistēmas tā, lai tos varētu ērti caurskatīt uz MS Excel un citām līdzīgām programmām

2.5. Nefunkcionālās prasības

2.5.1. Veiktspējas prasības

Piekļuve sistēmai tiek veikta ne ilgāk kā 10 sekundēs.

Datu filtrēšana un kārtošana pēc noteiktajiem parametriem tiek veikta ne ilgāk kā 2 sekundēs(ar kopējo ierakstu skaitu līdz 1000, lokālajā serverī(lai izpildes laiku neietekmētu datu pārraides ātrums)).

Datu saglabāšanas, kopēšanas darbības tiek izpildītas ne ilgāk kā 1 sekundes laikā(attiecas tikai uz darbībām, kas tiek veiktas tikai vienam ierakstam nevis vairākiem uzreiz, lokālajā serverī(lai izpildes laiku neietekmētu datu pārraides ātrums)).

Datu meklēšana, atlasīšana un drukāšana uz atskaitēm 90% gadījumos neaizņem vairāk par 2 sekundēm, pārējos 10% - ne vairāk par 5 sekundēm.

2.5.2. Datu drošība

Pie sistēmas datiem iespējams piekļūt tikai personām, kam ir pieeja konkrētajai HW sistēmai. Par importa un eksporta failu drošību atbildīgs lietotājs, kas tos izveidojis.

3. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

3.1. Ievads

3.1.1. Nolūks

Dokuments ir paredzēts programmatūras “Bankas palīgs” projektējuma uzlabojumu aprakstam. Dokuments ir paredzēts produkta izstrādātājiem, izstrādātāju vadībai, kas kontrolē pakalpojuma sniegšanu un programmatūras pasūtītājam.

3.1.2. Darbības sfēra

Programmatūra ir paredzēta ikvienam, kas izmanto HansaWorld sistēmu un ar internetbankas starpniecību apmaksā rēķinus un ir paredzēta, lai atvieglotu un automatizētu šīs darbības lietotājam.

3.1.3. Definīcijas un saīsinājumi

Bankas palīgs – kvalifikācijas darba laikā izstrādātais produkts

Sistēma – HansaWorld Enterprise produkts, kurā iekļauts standarta risinājums un “Bankas palīgs”

Modulis – atsevišķa identificējama programmatūras daļa, kuru var pievienot jau gatavai sistēmai, lai veiktu darbības atbilstošajai sfērai

Reģistrs – ierakstu kopas, kas piesaistītas moduļiem un paredzētās nepieciešamo datu uzglabāšanai. Datu atainošanai tiek lietoti lietotāju saskarnes logi

Ieraksts – viens ieraksts no reģistra, kas glabās konkrētu informāciju. Katram reģistram ir unikāls identifikācijas numurs

Bāzes logs – logs, kas atveras palaižot sistēmu. No bāzes loga var piekļūt pie visiem pārējiem logiem un ierakstiem

Pārskata logs – šajā logā tiek atainoti konkrēti ieraksti

Saraksta logs – šajā logā tiek atainots viss reģistra saturs(visi ieraksti)

Specifikācijas logs – logs, kurā tiek norādīti parametri veicamajai darbībai(eksportam, importam, apkalpošanai vai atskaitei)

Lauks – lauks lietotāja saskarnē, kuru lietotājs var gan būt spējīgs modificēt gan arī nebūt. Atkarībā no sistēmas prasībām

Matrica – tabula no kolonnām, rindām un šūnām datu ievadei

HW – HansaWorld Enterprise, produkts kuram jāpievieno izgatavotā izstrāde
HAL – Hansa Application Language. Uz C bāzēta valoda, kurā tika izstrādāts projekts

FIDAVISTA – Finansiālu datu apmaiņas vienotais standarts. Formāts kuru atbalsta lielāka daļa Latvijas banku. Šī dokumenta ietvaros FIDAVISTA apzīmē šī ierakstu, kas izveidojas veicot datu importu

Importa fails – bankas izraksts fīdavista formātā

Ievietot speciāli – HWE standarta funkcionalitāte- uzspiežot taustiņu kombināciju Cmd+Enter, tiek atvērts logs ar vērtībām ko var ievietot attiecīgajā laukā

3.1.4. Datu tipu apraksts

M4Code – string tipa mainīgais, kas var saturēt līdz 255 simboliem. Visi simboli lietotājam un datu bāzē tiek reprezentēti ar lielajiem burtiem

M4Str – simbolu virkne ar maksimālo garumu 255 simboli

M4Date – datu tips, kas attēlo un saglabā ievadīto vērtību kā datumu(nav noteikta principa kā tiks saglabāts, jo sistēmā iespējams mainīt datuma formātu)

M4Val – skaitlis ar precizitāti 0.01. Tiek izmantots, lai atainotu mainīgos, kas glabā naudas summas

M4Time – datu tips, kas attēlo un saglabā ievadīto vērtību kā laiku(formātā hh:mm:ss)

M4Int – vesels skaitlis robežās no $-2^{31}+1$ līdz $+2^{31}-1$

M4Matrix – matrica

M4Mark – integer tipa mainīgais, kas tiek izmantots, lai sistēmā vajadzētu ko vizuāli atainot(izvēloties konkrētu skaitli, sistēmā tiek rādīts konkrēts vizuāls simbols, piemēram, ķeksis)

M4Long – long integer mainīgais

M4Set – ataino iepriekš definētas vērtības ar tipu string

M4UStr – Uppercase string- rāda string mainīgo tikai ar lielajiem burtiem

3.1.5. Saistība ar citiem dokumentiem

Programmatūras projektējuma apraksts izveidots saskaņā ar “LVS 72:1996 Ieteicamo praksi programmatūras projektējuma aprakstīšanai un PPS” dotajiem norādījumiem.

Finansiālu Datu Apmaiņas Vienotais Standarts(FiDAViSta)

specifikācija, Versija 1.2.

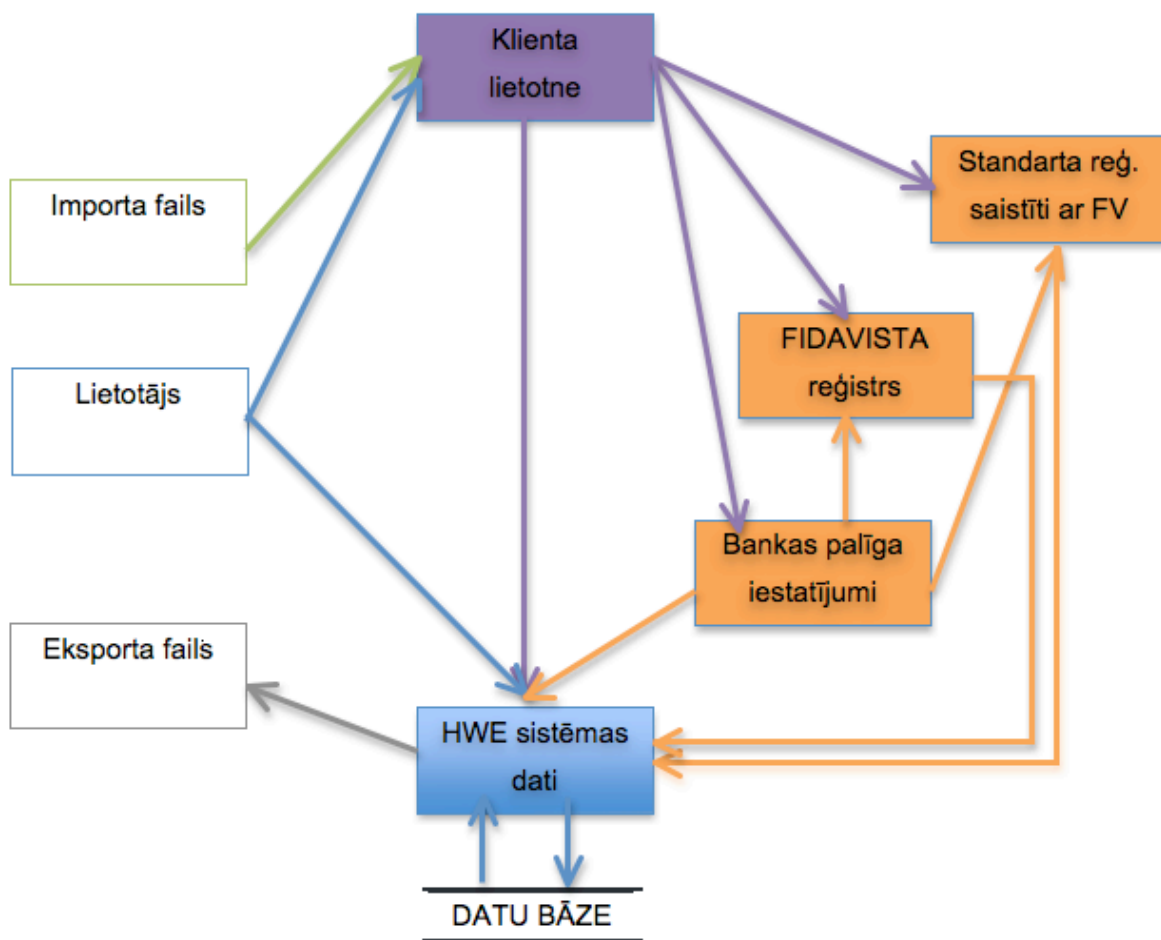
3.1.6. Pārskats

1. nodaļā ir aprakstīts dokuments, tā mērķis un tiek identificēts programmatūras projekts;
2. nodaļā aprakstītas atkarības;
3. nodaļā aprakstīts datu modelis;
4. nodaļā aprakstīta lietotāja saskarne;

3.2. Atkarību apraksts

Sistēma sastāv no sešiem moduļiem:

- Datubāze, kurā glabāsies sistēmā ievadītie lietotāja dati un sistēmas konfigurācijas dati
- FIDAVISTA reģistrs, kas tiek integrēts abās lietotnēs
- Sistēmas standarta reģistri, kas tiek saistīti ar FIDAVISTA ierakstiem (grāmatojumi, maksājumi, maksājumu uzdevumi, rēķini, ienākošie rēķini, kontakti)
- Bankas palīga iestatījumi, kas tiek integrēti abās lietotnēs
- HWE sistēma, kas sasaista bankas palīga datus ar pārējiem sistēmas pamatdatiem
- Klienta lietotne, kas apstrādā ievadīto informāciju uz servera un pareizas ievadītās informācijas gadījumā, saglabā datus sistēmas datubāzē. Kā arī veic pieprasījumus no bāzes, lai attēlotu tajā saglabātos datus lietotājam



3.3. Datu modelis

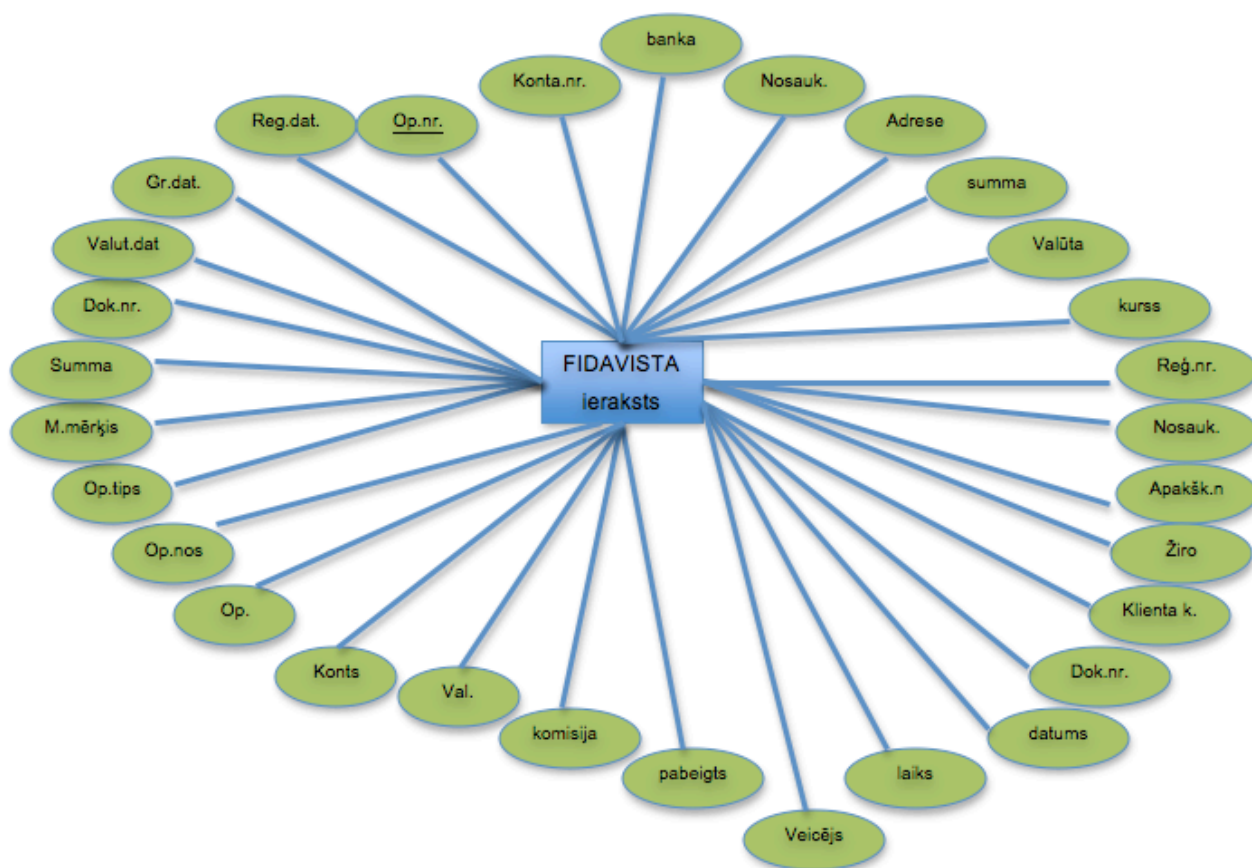
HWE sistēmā dati tiek glabāti datu bāzē, kuras struktūra daudz atšķiras no populārajiem SQL tipa datu bāžu variantiem. Sistēmā parasti tiek izmantotas vismaz divas datubāzes. Viena no tām ir saistīta ar servera puses lietotni, bet pārējās – lietotāja puses (katram lietotājam ir sava datu bāze). Lietotāju datubāzēs tiek glabāti dažādi uzstādījumi, kas ir specifiski lietotājam. Galvenokārt dati tiek glabāti uz servera, uz kura arī tiek rēķinātas funkcijas, kuras sadarbojas ar datu bāzi, jo pieprasījumu izpildīšana no klienta lietotnes lieki noslogotu tīklu. Pastāv arī iespēja, ka tiek izmantota tikai viena datubāze – nav klienta-servera savienojuma, bet ir tikai viena lokāla bāze, kas kalpotu gan kā klienta, gan servera datubāze, taču tādos gadījumos darbs ar HWE sistēmu kļūtu neērtāks, tāpēc turpmāk tiek pieņemts, ka instalācijā, uz kuras tiek likts Bankas palīgs, būs klienta-servera režīms.

Šajā nodaļā tiks aprakstīta servera datu bāzes uzbūve. Aprakstīta ir tikai jaunizveidotā funkcionalitāte, kuras datu definīcijas ir integrētas HWE sistēmā un tās datu bāzes struktūrā. Attēlotas tiks arī daļas no pamata funkcionalitātes, taču tās netiks atainotas detalizēti un to apzīmējošie bloki tiks iekrāsoti dzeltenā krāsā, lai atšķirtos no Bankas palīgu apzīmējošajiem blokiem.

3.3.1. Datu bāzes loģiskais ER modelis

Fidavista ieraksts detalizētāk tiks apzīmēts nākamajā lpp.





3.3.2. Datu bāzes tabulu apraksts

3.3.2.1. FIDAVISTA ieraksta tabula

Lauka nosaukums	Datu tips(garums)	Obligāts	Unikāls	Apraksts
BankRef	M4Code(60)	1	1	FIDAVISTA ieraksta kods(tiek ņemts no importa faila)
TypeCode	M4Str(5)	0	0	Operācijas tips
TypeName	M4Str(100)	0	0	Operācijas nosaukums
RegDate	M4Date	0	0	Reģistrācijas datums
BookDate	M4Date	0	0	Grāmatojuma datums
ValueDate	M4Date	0	0	Valut. Datums
ExtId	M4Str(10)	0	0	Netiek izmantots
BenExtId	M4Str(5)	0	0	Netiek izmantots
DocNo	M4Str(60)	0	0	Dokumenta numurs
CorD	M4Str(5)	0	0	Operācija
AccAmt	M4Val	0	0	summa
FeeAmt	M4Val	0	0	komisija
PmtInfo	M4Str(200)	0	0	Maksājuma mērķis
CPSAccNo	M4Str(60)	0	0	Konta numurs
CPSSubAccNo	M4Str(60)	0	0	Apakškonta numurs
CPSAHName	M4Str(200)	0	0	Nosaukums
CPSAHLegalld	M4Str(20)	0	0	Reģistrācijas numurs
CPSAHAddresses	M4Str(70)	0	0	Adrese
CPSAHExtension	M4Str(200)	0	0	Netiek izmantots
CPSBankCode	M4Str(20)	0	0	Banka kods
CPSBankName	M4Str(200)	0	0	Bankas nosaukums
CPSCcy	M4Str(5)	0	0	Valūta
CPSAmt	M4Val	0	0	Summa
CPSCurRate	M4Val	0	0	Kurss
CPSGiro	M4Str(5)	0	0	Žiro
CPSExtension	M4Str(255)	0	0	Netiek izmantots

Extension	M4Str(255)	0	0	Netiek izmantots
ImportDate	M4Date	0	0	Importa datums
ImportTime	M4Time	0	0	Importa laiks
ImportUser	M4Code(5)	0	0	Importa veicējs
ImportStatus	M4Int	0	0	Statuss
Math	M4Matrix	0	0	Matrica
CUCode	M4Code(20)	0	0	Klienta kods
DoneFlag	M4Mark	0	0	Apstrāde pabeigta
Ccy	M4Str(5)	0	0	Valūta
FileName	M4Str(20)	0	0	Faila nosaukums
TransNr	M4Long	0	0	Faila numurs
md5hash	M4Str(32)	0	1	Unikāls identifikators

3.3.2.2. FIDAVISTA ieraksta matrica

Lauka nosaukums	Datu tips(garums)	Obligāts	Unikāls	Apraksts
Code	M4Int	1	1	Rindas numurs
LogDate	M4Date	0	0	Loga datums
LogTime	M4Time	0	0	Loga laiks
LogMessage	M4Str(200)	0	0	Loga teksts

3.3.2.3. FIDAVISTA aprakstu uzstādījumi

Lauka nosaukums	Datu tips(garums)	Obligāts	Unikāls	Apraksts
Math	M4Matrix	0	0	Matrica
Pattern	M4Str(80)	0	0	Raksts
Type	M4Set	0	0	Tips
Comment	M4Str(100)	0	0	Komentārs

3.3.2.4. FIDAVISTA parametri

Lauka nosaukums	Datu tips(garums)	Obligāts	Unikāls	Apraksts
ImportFolder	M4Str(20)	0	0	Importa mape
AfterFolder	M4Str(20)	0	0	Mape pēc importa
ExportFolder	M4Str(20)	0	0	Eksporta mape
OKIPFlag	M4Int	0	0	Grāmatot maksājumus
Encoding	M4Int	0	0	Importa faila

				kodejums
ImpSales	M4Int	0	0	Importēt maksājumus
ImpPurch	M4Int	0	0	Importēt MU
ImpTrans	M4Int	0	0	Importēt grāmatojumus
SalesPrepay	M4Int	0	0	Veidot priekšapmaksas maksājumus
PurchPrepay	M4Int	0	0	Veidot priekšapmaksas MU
SalesSum4Match	M4Int	0	0	Piesaistīt, ja sakrīt rēķina summa
SalesDebtMatch	M4Int	0	0	Piesaistīt, ja sakrīt kopējā parāda summa
AutorunReport	M4Int	0	0	Netiek izmantots
SalesPayOld	M4Int	0	0	Apmaksāt vecākos rēķinus
AllowCcyMismatch	M4Int	0	0	Piesaistīt, ja nesakrīt valūta
PurchSum4Match	M4Int	0	0	Piesaistīt, ja sakrīt rēķina summa MU
PurchDebtMatch	M4Int	0	0	Piesaistīt, ja sakrīt kopējā parāda summa MU
PurchPayOld	M4Int	0	0	Apmaksāt vecākos rēķinus MU
OKOPFlag	M4Int	0	0	Grāmatot MU
NoVATNr	M4Int	0	0	Ļaut veidot kontaktu, ja nav pvn reģ. Nr.
NoBankAcc	M4Int	0	0	Ļaut veidot kontaktu, ja nav bankas konta
OpenCU	M4Int	0	0	Pēc izveidošanas atvērt kontakta ierakstu
AttachExisting	M4Int	0	0	Ļaut piesaistīt

				ierakstam esošu kontaktu
ExpComment	M4Int	0	0	Eksportēt komentāru
ExpDate	M4Int	0	0	Eksportēt datumu

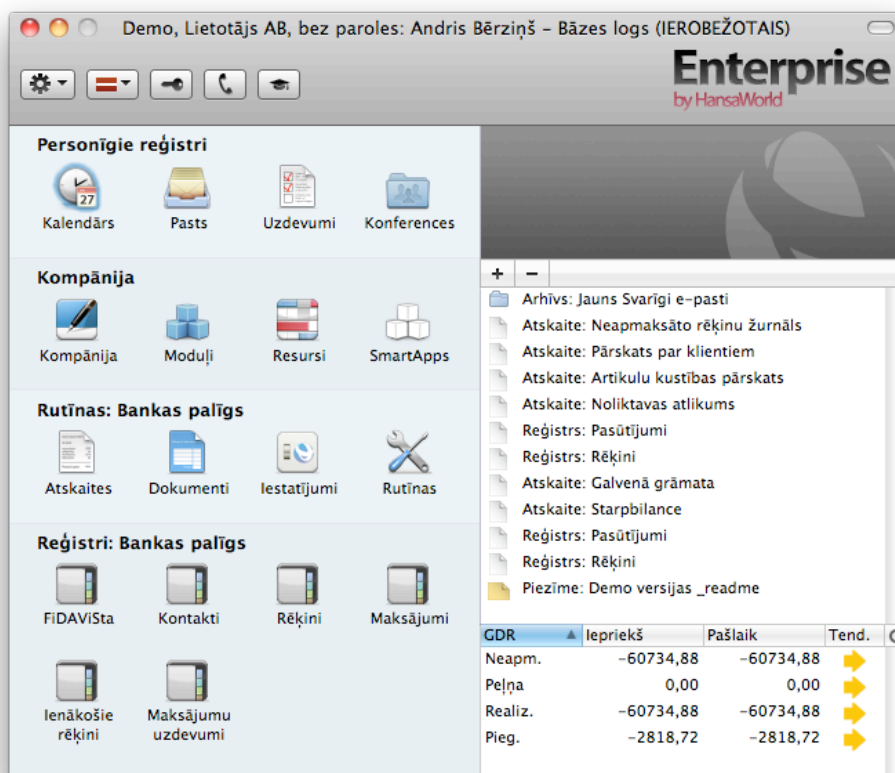
3.3.2.5. FIDAVISTA bankas kontu piesaiste kontiem

Lauka nosaukums	Datu tips(garums)	Obligāts	Unikāls	Apraksts
Math	M4Matrix	0	0	Tabula
BankAcc	M4Str(40)	0	0	Bankas konts
Pattern	M4Str(50)	0	0	Raksts
Acc	M4Str(20)	0	0	Konts
Objects	M4UStr(60)	0	0	Objekts
Comment	M4Str(100)	0	0	Komentārs

3.4. Lietotāja saskarnes apraksts

3.4.1. HWE sistēmas bāzes logs

HWE sistēmas galvenais logs tiek saukts par bāzes logu, ļauj lietotājam ērti orientēties sistēmā un no tā var ātri piekļūt jebkurai sistēmas daļai. Pašā loga augšpusē kreisajā malā atrodas izvēlnes poga, kurā pieejama dažādas speciālās operācijas, kas attiecas uz konkrēto lietotāju, izžurnālēšanās poga, telefona centrāles atvēršanas poga, mācību materiālu poga un produkta logo. Labajā malā ir lietotāja veidots saraksts ar saīsnēm uz ierakstiem vai darbībām sistēmā, kuras lietotājs pats var pievienot un dzēst. Arī personīgie reģistri satur informāciju, kas katram lietotājam ir atšķirīga. Kompānijas sadaļā iespējams izvēlēties moduļus. Attēlā redzamajā logā ir atvērts modulis “Bankas palīgs”, kas arī tiks tālāk detalizēti aprakstīts šajā dokumentā.



3.4.1. attēls

3.4.2. FIDAVISTA ierakstu saraksta logs

Šajā logā iespējams aplūkot visus sistēmā izveidotos FIDAVISTA ierakstus,

kārtot tos pēc kolonnām, kā arī kopēt, dzēst un veidot jaunus ierakstus.

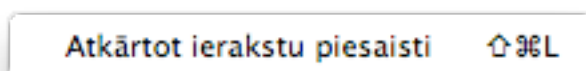
BankRef	BookDate	Done	CPSAHLegalId	CPSAHName	CUCODE	PmtInfo	AccAmt
1183336617	26.04.2011		11066612733	PR. ĀRSTS INETA DOMBROVSKA		PAR TONERA KASETI..P...	36,53 C
1183378380	26.04.2011		45403007675	DATE V SIA		Pavadzime Nr. 1112945 ...	26,01 C
1183486793	26.04.2011		40003084917	A/S RITA		PAV.1112738 BTSC-14...	51,24 C
1183512456	26.04.2011		40003146723	KS JŪRMALAS STŪRE		PAVADZĪME NR.1111988...	12,20 C
1183529303	26.04.2011		41503016220	ALFA LĪ SIA		PAVADZĪMES 1113332 B...	20,18 C
1183580418	26.04.2011		40002043216	ANTOLS-HIM SIA		APMAKSA SASKAŅĀ AR P...	30,50 C
1183633120	26.04.2011		50003690041	I DRAVNIECES GR.BIR. SIA		PPR Nr.1113415BTSC, rē...	72,72 C
1183634950	26.04.2011		40003219730	KNAUF SIA		1109755BTSC	165,09 C
1183661425	26.04.2011		45403009587	ARVITA SIA		IEPIRKŠ.PZ.: 1112763 BT...	54,44 C
1183666649	26.04.2011		40003796236	PRISMA LATVIJA AS		1112155BTSC	36,64 C
1183668693	26.04.2011		40103069401	MELIORATORS-I SIA		P.NR.1113019BTSC	12,70 C

3.4.2. attēls

3.4.3. FIDAVISTAS ierakstu saraksta loga darbību poga

Pārskata loga augšpusē, kreisajā pusē redzama poga, kuru uzspiežot tiek atvērtas darbības ko iespējams veikt ar iezīmētajiem ierakstiem(turpmāk saukta par darbību pogu). Šajā gadījumā ir tikai viena darbība ko izvēlēties(blakus tai norādīta taustiņu kombinācija kuru nospiežot tiktu izpildīta tā pati darbība).

Atkārtot ierakstu piesaisti- visiem iezīmētajiem ierakstiem vēlreiz tiks izpildīta importa funkcija, t.i. vēlreiz tiks mēģināts sameklēt iezīmētajiem ierakstiem sistēmā reģistrētus, saistītos ierakstus.



3.4.3. attēls

3.4.4. FIDAVISTA ieraksta logs

Šajā logā var aplūkot katru konkrēto FIDAVISTA ierakstu ar to saturošo informāciju. Informācija(izņemot klienta kodu) drošības apsvērumu dēļ ierakstā nevar tikt mainīta manuāli. Ieraksta kreisajā pusē tiks attēlota maksātāja un saņēmēja informācija, labajā pusē- faila statuss un importa informācija. Loga lejasdaļā tiks rādīta ieraksta vēsture- darbības kas veiktas ar ierakstu, tā dēvētajā ieraksta “logā”.

The screenshot shows a software window titled "DEMO: FiDAViSta: Jauns". It contains several sections for data entry:

- Operācijas dati:** Fields for Op.numurs, Reģ.dat., Grāmat.dat., Valut.dat., Dok.numurs, Summa, and Maks.mērķis.
- Importa dati:** Fields for Datums, Laiks, Veicējs, and a Status section with radio buttons for "Importēts fails", "Maks. nav izveidots", and "Izveidots maksājums", plus a checkbox for "Apstrāde pabeigta".
- Pretējās puses dati:** Fields for Konta nr., Banka, Nosaukums, Adrese, Summa, Valūta, Kurss, Apakšk.nr., Žiro, Reg.nr., Klienta kods, and Dok.nr.
- Ieraksta importa vēsture:** A table with columns "Datums", "Laiks", and "Ziņojums".

	Datums	Laiks	Ziņojums
1			
2			
3			
4			
5			
6			
7			
8			
9			

3.4.4. attēls

3.4.5. FIDAVISTA ieraksta loga darbību poga

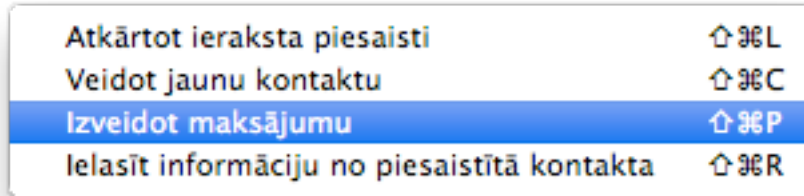
Šī poga attēlo visas darbības, ko iespējams veikt ar konkrēto atvērto ierakstu.

Atkārtot ieraksta piesaisti – ierakstam tiks atkārtoti palaista importa funkcija

Veidot jaunu kontaktu – ja sistēmā netika atrasts ieraksts, kas atbilst informācijai, ko satur ieimportētais FIDAVISTA ieraksts, iespējams izveidot kontaktu, kas saturēs FIDAVISTA ierakstā atrasto informāciju (iespējams papildināt arī ar sev nepieciešamu papildinformāciju)

Izveidot maksājumu – izveidos no ieraksta jaunu maksājumu, pat ja maksājumu veidošana parametros nebūs ieslēgta

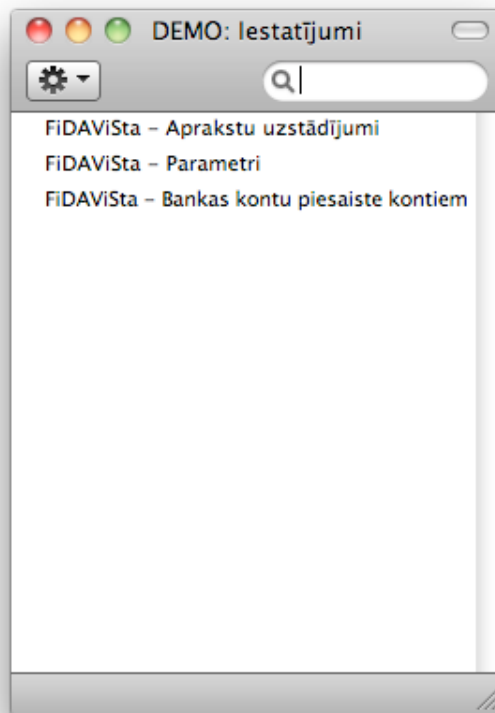
Ielasīt informāciju no piesaistītā kontakta – paredzēts gadījumos, kad lietotājs ir manuāli ievietojis kādu kontaktu no datubāzes. Tad izvēloties šo darbību, ierakstā tiks ielasīta informācija no piesaistītā kontakta



3.4.5. attēls

3.4.6. Iestatījumu izvēlnes logs

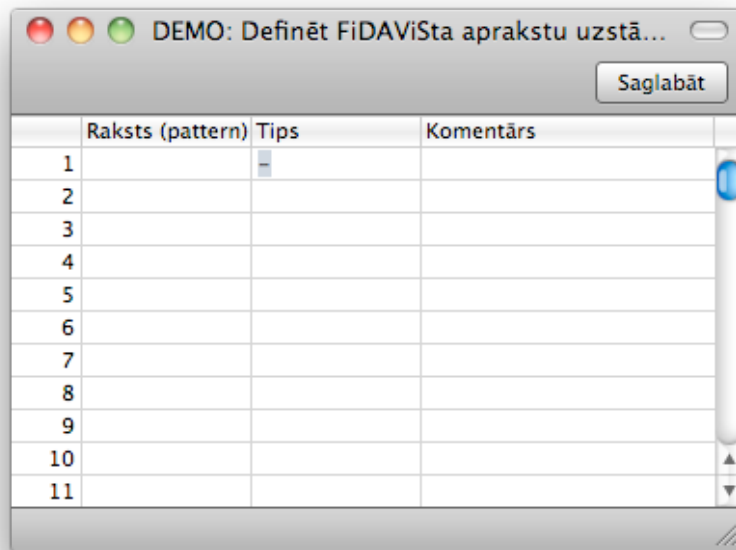
Šajā logā iespējams redzēt un izvēlēties visus konkrētā moduļa iestatījumus.



3.4.6. attēls

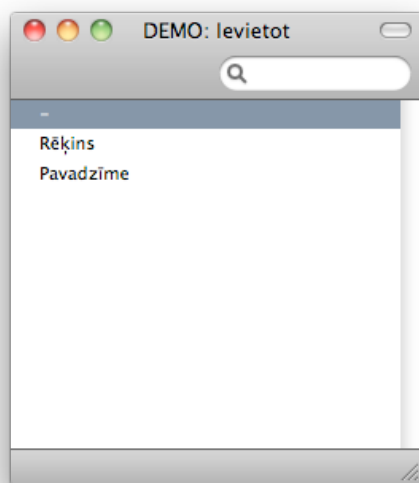
3.4.7. Iestatījuma FIDAVISTA – aprakstu uzstādījumi logs

Šajā logā iespējams aplūkot iepriekšminētā iestatījuma laukus kā arī mainīt tos pēc vajadzības.



3.4.7. attēls

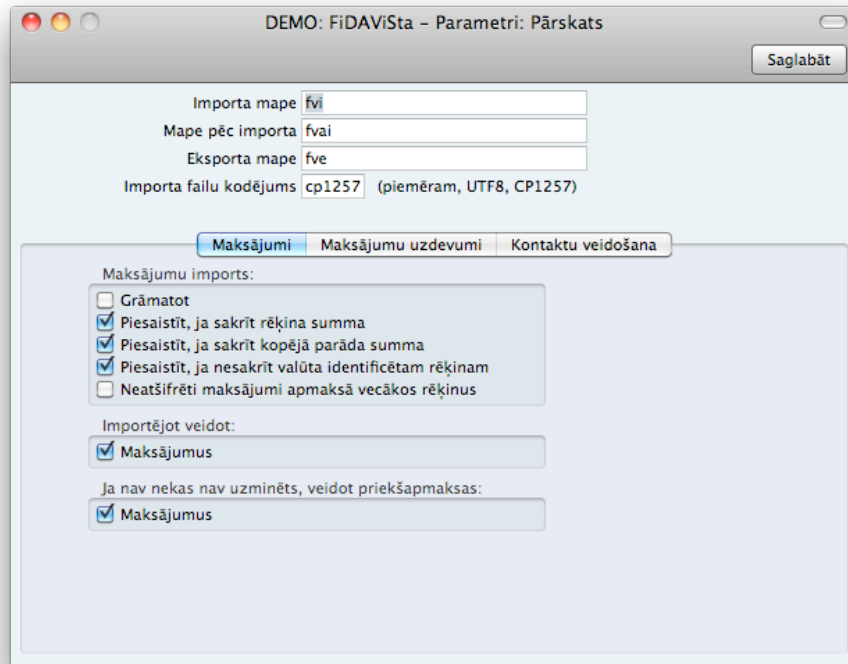
Kolonnā “tips”, izmantojot funkciju “Paste special”, var izvēlēties starp iepriekš definētām vērtībām.



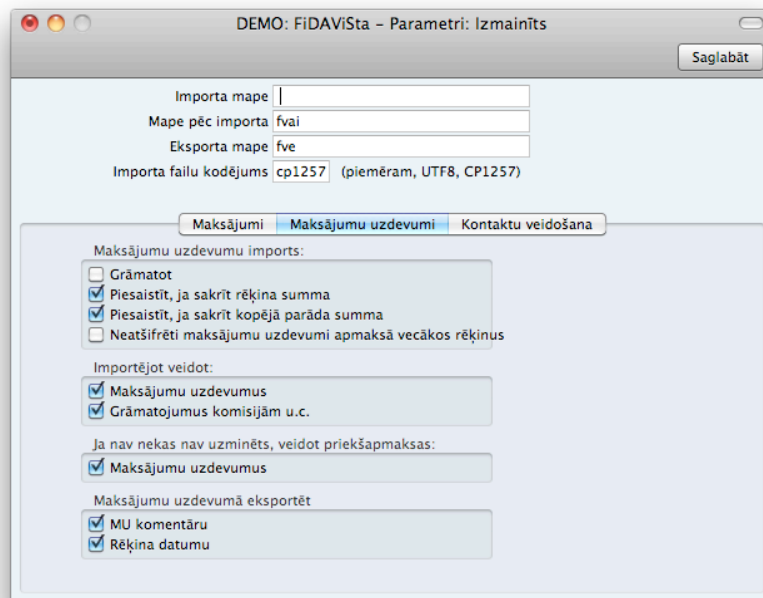
3.4.7.2. attēls

3.4.8. iestatījuma FIDAVISTA – parametri logs

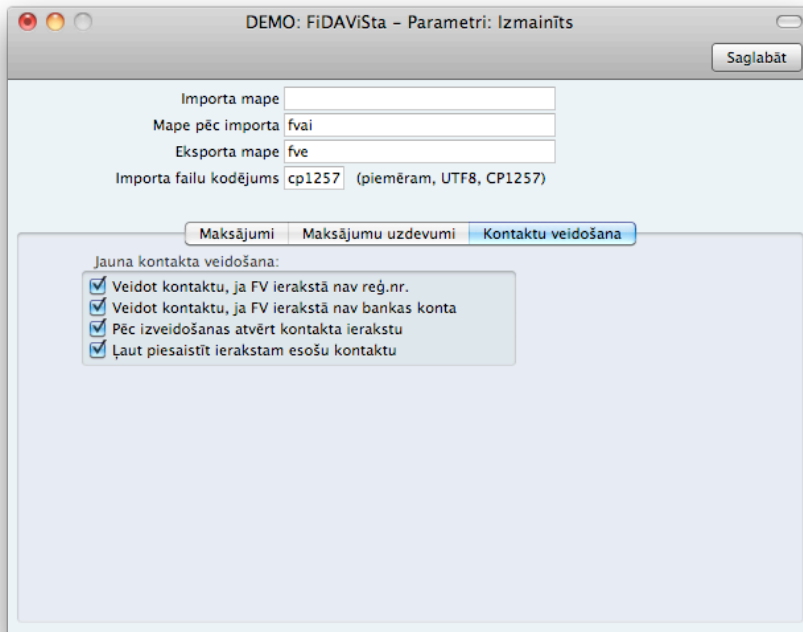
Šajā logā iespējams aplūkot un mainīt atbilstošā iestatījuma parametrus. Iestatījums nosaka kādos gadījumos un kādi jāveido sistēmas ieraksti importa laikā. Tālāk redzamas iestatījumu loga visas sadaļas.



3.4.8. attēls



3.4.8.1. attēls



3.4.8.2. attēls

3.4.9. iestatījuma FIDAVISTA – bankas kontu piesaiste kontiem logs

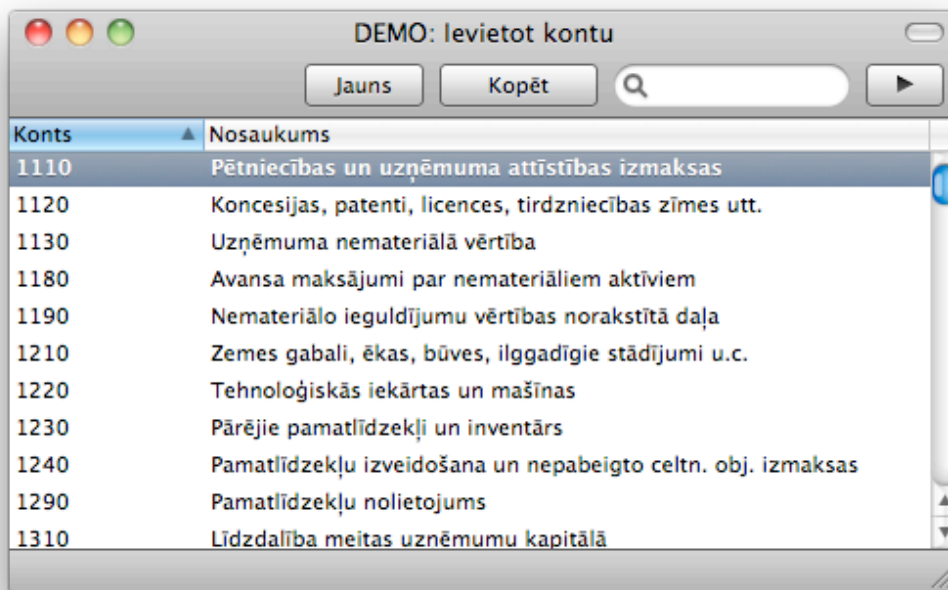
Šajā logā var apskatīt un mainīt augstākminētā iestatījuma laukus. Iestatījums paredzēts, lai veidojot grāmatojumus, tiktu sasaistīti pareizie konti. Kontiem(3.4.9.1. attēls) un objektiem(3.4.9.2. attēls) ir iespēja izmantot funkciju “Paste special” tādējādi izvēloties sev nepieciešamo ierakstu(ieraksts tiek izvēlēts kā jebkurā citā sistēmā- ar dubultklišķi vai ar taustiņu enter).

DEMO: Definēt FiDAViSta bankas kontu piesaisti kontiem: Pārskats

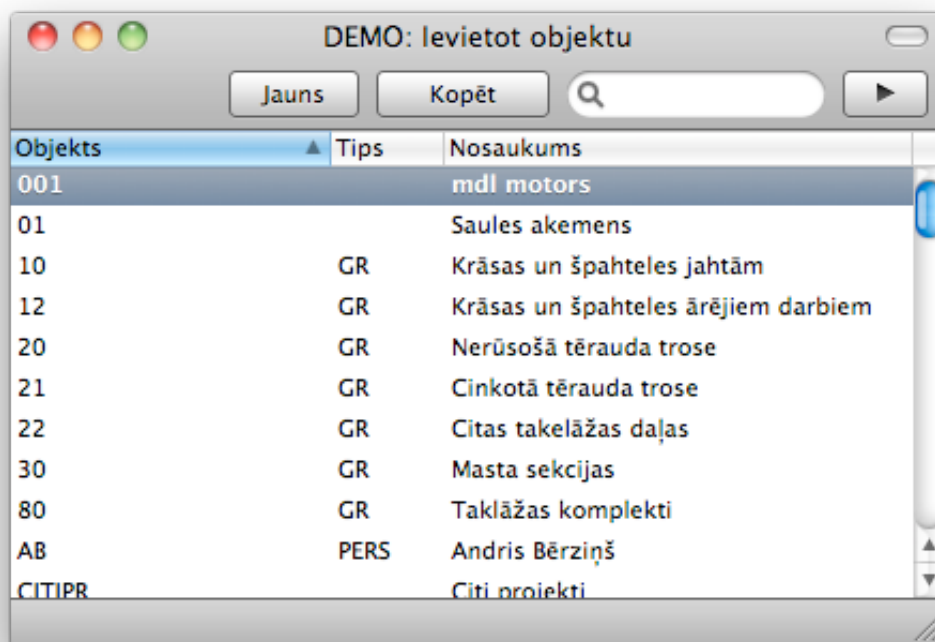
Saglabāt

	Bankas konts	Raksts (pattern)	Konts	Objekti	Komentārs
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

3.4.9. attēls



3.4.9.1. attēls

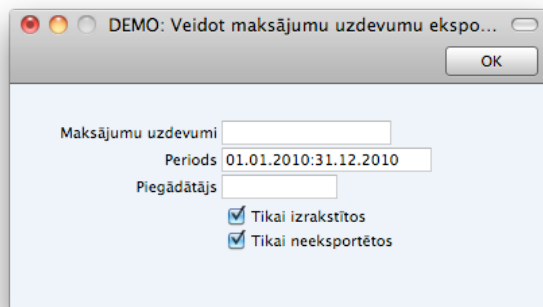


3.4.9.2. attēls

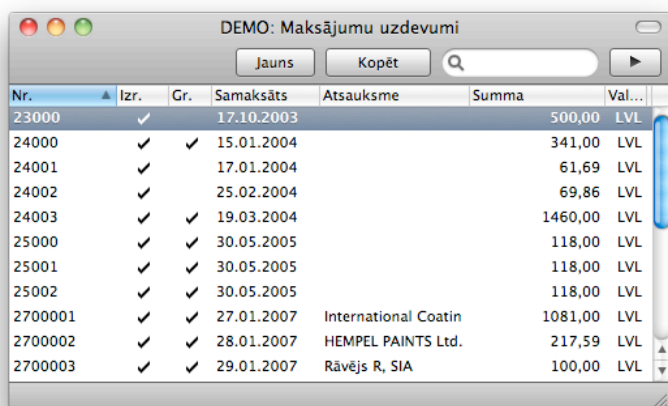
3.4.10. Maksājumu uzdevumu eksporta specifikācijas logs

Šajā logā ir iespējams norādīt nepieciešamos parametrus pēc kādiem tiks atlasīti ieraksti konkrētajam eksportam. Visiem laukiem ir “Paste special” funkcijas

iespēja(attēli 3.4.10.1.-3.4.10.3.), lai atvieglotu lietotājam darbu. Ķekši norāda kādām īpašībām jābūt eksportējamajiem ierakstiem. Pēc “Ok” pogas nospiešanas lietotājam tiks atvērts logs(3.4.10.4. attēls), kurā būs jānorāda kur veidot eksporta failu, kādā formātā un kur to saglabāt(jaunatvērtais logs ir operētājsistēmas standarta pārskata logs).



3.4.10. attēls



Nr.	Izr.	Gr.	Samaksāts	Atsauksme	Summa	Val...
23000	✓		17.10.2003		500,00	LVL
24000	✓	✓	15.01.2004		341,00	LVL
24001	✓		17.01.2004		61,69	LVL
24002	✓		25.02.2004		69,86	LVL
24003	✓	✓	19.03.2004		1460,00	LVL
25000	✓	✓	30.05.2005		118,00	LVL
25001	✓	✓	30.05.2005		118,00	LVL
25002	✓	✓	30.05.2005		118,00	LVL
2700001	✓	✓	27.01.2007	International Coatin	1081,00	LVL
2700002	✓	✓	28.01.2007	HEMPEL PAINTS Ltd.	217,59	LVL
2700003	✓	✓	29.01.2007	Rāvējs R, SIA	100,00	LVL

3.4.10.1. attēls

DEMO: levietot periodu

01.01.2010:31.12.2010	2010
01.01.2004:31.12.2009	
01.01.2009:31.12.2009	2009
01.01.2008:31.12.2008	2008
01.01.2008:31.01.2008	1
01.02.2008:28.02.2008	2
01.03.2008:31.03.2008	4
01.04.2008:30.04.2008	4
01.05.2008:31.05.2008	5
01.06.2008:30.06.2008	6
01.07.2008:31.07.2008	7
01.08.2008:31.08.2008	8

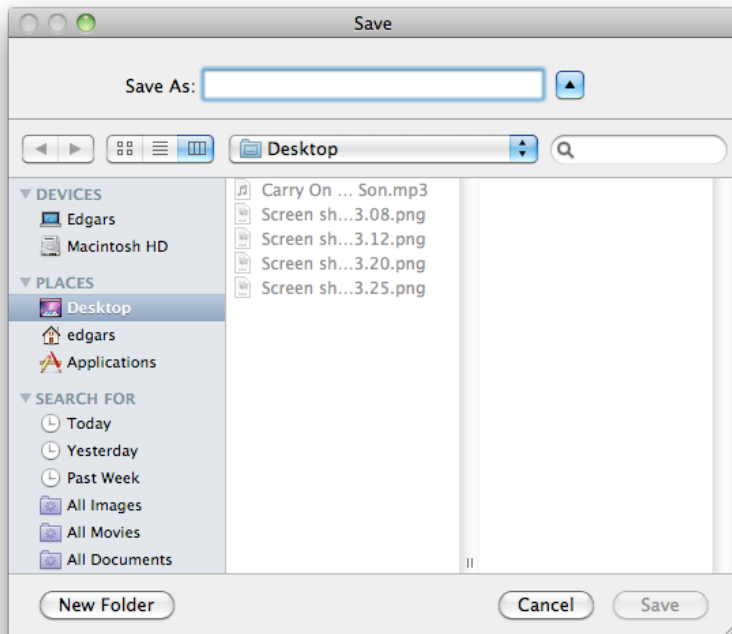
3.4.10.2. attēls

DEMO: levietot piegādātāju

Jauns Kopēt

Nr.	Īsais	Kat.	Nosaukums	Departaments	PVN reģ. Nr.	Bankas konts
10112		CITI	Betons Plus, SIA		LV 4000020...	LV00HAB...
10114		PR	Rāvējs R, SIA		LV 40000201...	LV00HAB...
10115		AN	Rīgas taksometru parks, AS		LV 40012010...	LV00RIET...
10128			AEZ SIA		LV 40003136...	LV50LAPB...
10141			Valsts kase		90000050138	LV37TREL...
20102		PR	HEMPEL PAINTS Ltd.		UK 300031001	155A554...
20103		PR	International Coatings Ltd		UK 400010023	155A123...
20104		PR	Barton Marine Ltd.			
20105	LMT	CITI	Latvijas Mobilais Telefons AS		LV 50004561...	0140805...
20106		PR	Latvijas Finieris AS		LV 40004561...	0140805...
20107		CITI	Auzas SIA		LV 40214161	LV00HAB...

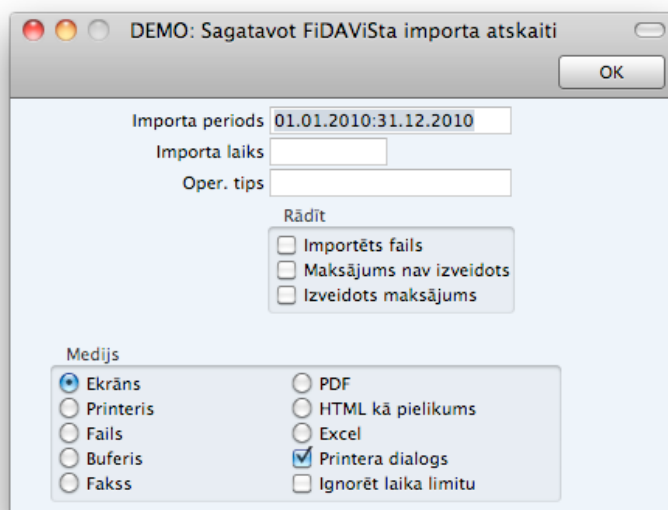
3.4.10.3. attēls



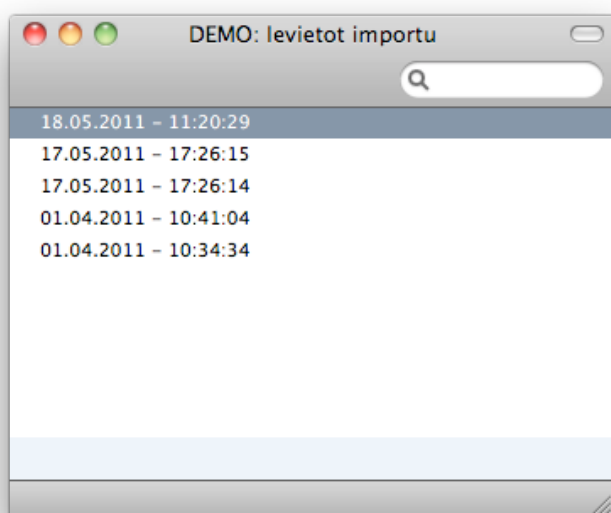
3.4.10.4. attēls

3.4.11. FIDAVISTA statistika atskaites specifikācijas logs

Šajā logā tiek norādīti parametri pēc kuriem tiks atlasīti ieraksti, kas tiks rādīti atskaites pārskata logā. Loga apakšdaļā var norādīt kā veikt atskaiti. Izvēloties, piemēram, uz failu- atskaite tiks izpildīta kā eksports un rezultāti izvadīti failā(šī darbība var būt nepieciešama, ja atskaitē ir liels daudzums rezultātu un tos nevar attēlot ekrānā). Laukā “Importa laiks” ar “Paste special” funkciju(3.4.11.1. attēls) tiks piedāvāti pēdējie 10 eksporti(ja tik daudz, protams, sistēmā jau būs veikti. Ja būs mazāk, tiks rādīti mazāk). Importu izvēles logā importi tiks sakārtoti secībā sākot ar jaunāko un pierakstīti formātā “datums” + ” - ” + ”laiks”.



3.4.11. attēls



3.4.11.1. attēls

3.4.12. atskaites pārskata logs

Šajā logā tiek rādīti tie ieraksti, kas atbilst ieraksta specifikācijā norādītajiem parametriem. Ierakstu numuri ir pasvītroti, jo uzspiežot uz numura, tiks atvērs tā ieraksts.

DEMO: FiDAViSta importa statistika

IEROBEŽOTĀIS FiDAViSta importa statistika
Demo kompānija, SIA

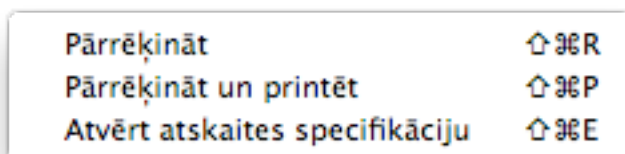
HansaWorld, Izdrukas datums: 22.05.2011 17:14
Periods: 01.01.2010-31.12.2011

Op. numurs	Op. tips	Summa	Statuss
20110301	OTHR	2,06	Importēts fails
20110302	OTHR	1,20	Importēts fails
2011030200419713	OUTP	271,00	Dokuments nav izveidots
2011030300252304	INP	12,10	Izveidots maksājums
1183336617	INP	36,53	Dokuments nav izveidots
1183378380	INP	26,01	Dokuments nav izveidots
1183486793	INP	51,24	Dokuments nav izveidots
1183512456	INP	12,20	Dokuments nav izveidots
1183529303	INP	20,18	Dokuments nav izveidots
1183580418	INP	30,50	Dokuments nav izveidots
1183633120	INP	72,72	Dokuments nav izveidots
1183634950	INP	165,09	Dokuments nav izveidots
1183661425	INP	54,44	Dokuments nav izveidots
1183666649	INP	36,64	Dokuments nav izveidots
1183668693	INP	12,70	Dokuments nav izveidots
1183669291	INP	15,01	Dokuments nav izveidots
1183669427	INP	19,52	Dokuments nav izveidots
1183669895	INP	113,24	Dokuments nav izveidots
1183670467	INP	75,73	Dokuments nav izveidots
1183670489	INP	49,91	Dokuments nav izveidots
1183670865	INP	14,64	Dokuments nav izveidots
1183676912	INP	100,50	Dokuments nav izveidots
1183678485	INP	24,80	Dokuments nav izveidots
1183678568	INP	10,00	Dokuments nav izveidots
1183679870	INP	169,49	Dokuments nav izveidots
1183682126	INP	88,40	Dokuments nav izveidots
1183685864	INP	61,21	Dokuments nav izveidots

3.4.12. attēls

3.2.13. Atskaites darbību izvēlnes logs

Arī atskaitēm ir iespēja izvēlēties darbības. Vai nu pārrēķināt, ja mainīti sistēmas ieraksti vai atvērt atskaites specifikācijas logu vēlreiz un mainīt parametrus.



3.2.13. attēls

3.2.14. Kļūdu un paziņojumu attēlošanas logs

Arī kļūdas un paziņojumi tiek attēloti specifiskos logos. Šiem logiem vizuālas atšķirības nav.



3.2.14. attēls



3.2.14.1. attēls

4. TESTĒŠANAS DOKUMENTĀCIJA

4.1 Ievads

Dokumentā attēloti un aprakstīti sistēmas vienībtestēšanas rezultāti. Testēšana veikta pēc melnās kastes metodes. Testi veidoti, lai pārlicinātos vai funkcijas atrod pareizos ierakstus un piesaista tos pareizajiem ierakstiem, vai visi ieraksti ir atrodami pēc specifikāciju logos norādītās informācijas, kā arī vai izpildās darbības ko veic ar ierakstiem.

4.2. Testa datu sagatavošana

Lai testēšana būtu pilnvērtīga, tika sagatavota demo datubāze ar datiem HWE sistēmā, lai varētu pilnvērtīgi skatīties kā sistēma sadarbojas ar jau gataviem datiem, kā tas notiek arī reālās HWE instalācijās.

4.3. Testēšanas rezultāti

Ar testēšanu nodarbojās arī citi sistēmas lietotāji, kas sniedza savus ziņojumus par datu nepilnībām. Turpmāk redzami tabulās sakārtoti testpiemēru dati.

4.3.1. Ieraksta veidošanas laikā tiek ielasīti visi dati tiem paredzētajos laukos

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Operācijas dati	Tiek atbilstošajā formātā aizpildīti visi lauki ar datiem no importa faila	1
Importa dati	Tiek atbilstošajā formātā aizpildīti visi lauki ar datiem no importa faila	1
Statuss	Statuss norāda, kas noticis ar ierakstu importa laikā	1
Pretējās puses dati	Tiek atbilstošajā formātā aizpildīti visi lauki ar datiem no importa faila	1
Ieraksta importa vēsture	Tiek izvadīts pareizais paziņojums par to kas notic ar ierakstu importa laikā	1

4.3.2. Klienta atrašanas pēc reģistrācijas numura

Sistēmā reģistrētie PVN šabloni(?- jebkurš simbols, #- jebkurš skaitlis):

#####-#####
 LV #####
 EE #####
 LT #####
 EE##?AB

Importa failā	Klientam sistēmā	Nepieciešamais rezultāts	Reālais rezultāts kā nepieciešamais
LV12345678901	LV 12345678901	LV 12345678901	1
LV 12345678901	LV 12345678901	LV 12345678901	1
12345678901	LV 12345678901 vai 123456-78901	LV 12345678901 vai 123456-78901	1
AA 12345678901	LV 12345678901	Neatrod	1
EE 12cAB	EE12cAB	EE12cAB	1
AE12cAB	EE12cAB	Neatrod	1
EE 123456789	EE 123456789	EE 123456789	1
LV 123456789	LV 12345678901 vai LT 123456789	Neatrod	1
LT 123456789	LT 123456789	LT 123456789	1
LT12345678901	LV 12345678901 vai LT 123456789	Neatrod	1
123456 78901	123456-78901	123456-78901	1

4.3.3. Ieraksta datu kārtošana kolonnās

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Ierakstu kārtošana	Nospiežot uz nepieciešamās kolonnas ieraksti tiek kārtoti dilstošā vai augošā secībā	1

4.3.4. Ierakstu meklēšana pārskata logā

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Ierakstu meklēšana	Ierakstot meklēšanas laukā kādu tekstu, tiek sameklēts līdzīgākais ieraksts sistēmā, skatoties pēc kāda lauka tika sakārtoti ieraksti	1

4.3.5. Speciālā ievietošana

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Speciālā ievietošana	Izmantojot “Paste special” funkciju, atbilstošā vērtība ievietojas izvēlētajā laukā	1
Papildus informācijas ievietošana	Logiem no kuriem paredzēts ielasīt informāciju vairāk kā vienā laukā, informācija ielasās visos paredzētajos laukos	1

4.3.6. Iestatījumos ievadītie dati tiek saglabāti sistēmā

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Datu saglabāšana iestatījumos	Saglabājot ieraksta saturu un atverot to vēlāk, iepriekš saglabātie dati tiek piedāvāti kā noklusējuma dati	1

4.3.7. Ieraksta statusa izmaiņas

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
No ieraksta tiek izveidots maksājums vai	Mainās statuss uz “Maksājums izveidots”	1
Maksājums tiek izdzēsts	Mainās statuss uz “Maksājums nav izveidots”	Pārbaude izieta?
Maksājums tiek grāmatots	Tiek atzīmēts ķeksis “Apstrāde pabeigta”	1

4.3.8. Elementa eksistences pārbaude

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
----------------	-----------------------	------------------

Saglabā ierakstu ar tādu pašu sērijas numuru	Paziņo ka tāds ieraksts jau sistēmā ir reģistrēts	1
--	---	---

4.3.9. Pēc ieraksta dzēšanas, tiek dzēstas saites uz to

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Izdzēš ierakstu, kam bija piesaistīti sistēmas ieraksti	No sistēmas ieraksta pazūd saite uz FIDAVISTA ierakstu	1

4.3.10. Ierakstam piesaistot jaunu ierakstu, abos izveidojas saite

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
No FIDAVISTA izveido jaunu kontaktu un piesaista tam maksājumu	FIDAVISTA ierakstam tiek pievienotas divas saites- katra uz savu ierakstu un katram ierakstam viena saite uz FIDAVISTA ierakstu	1

4.3.11. Atskaite rāda datus atbilstoši specifikācijā norādītajiem

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Specifikācijā norāda specifiskus datus	Palaižot atskaiti, tiek rādīti tikai ieraksti, kas ietilpst specifikācijā norādītajos parametros	1

4.3.12. Importējot tiek paziņots par importējamo ierakstu skaitu un ieimportēto ierakstu skaits

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Veic ierakstu importu	Tiek izvadīts pareizs skaits ar ierakstiem, kas atrodas importa failā un tiek izvadīts pareizs skaits ar ierakstiem kas izveidojušies importa funkcijas laikā	1

4.3.13. Eksporta failā tiek ierakstīti tikai norādītie ieraksti

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Veic ierakstu eksportu	Eksporta failā atrodas tikai ieraksti, kas atbilst eksporta specifikācijā norādītajiem parametriem	1

4.3.14. Tiek izveidots kontakts ar ierakstā atrodamo informāciju

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Izveido no ieraksta jaunu kontaktu	Izveidojas jauns kontakts	1
Veido kontaktu no ieraksta kurā nav reģ.nr. un iestatījumos norādīts, ka nedrīkst veidot tādu ierakstu	Netiek izveidots jauns kontakts	Pārbaude izieta?
Iestatījumos atzīmēts, ka pēc izveidošanas jāatver kontakta ieraksts	Kontakts atveras	1

4.3.15. Atkārtota ieraksta piesaiste

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Veic atkārtotu piesaisti, izmainot datus, lai šajā reizē jāizveidojas maksājumam	Izveidojas maksājums un lietotājs par to tiek informēts	1

4.2.16. Piesaistās informācija no pievienotā kontakta

Testa apraksts	Sagaidāmais rezultāts	Pārbaude izieta?
Pievieno ierakstam kontaktu un ielasa no tā informāciju	Informācija tiek ielasīta ierakstā	1

5. PROJEKTA ORGANIZĀCIJA

Projekts tika izstrādāts, kad jau bija apgūta HAL valoda, tādējādi tika ietaupīts daudz laika, kas būtu jāpatērē valodas apgūšanas, lai varētu pilnvērtīgi veikt darbu.

Projekta izstrāde sākās ar rūpīgu prasību apkopošanu un analīzi. Pēc idejas un mērķu izprašanas tika aptuveni noformulētas prasības par funkcionalitāti un lauku izkārtojumu logos. Ar pasūtītāju un projekta vadītāju tika izveidota aptuvena programmatūras prasību specifikācija, kas pamazām tika precizēta. Tika izveidots programmatūras projektējuma apraksts, kurā tika aprakstīti galvenie sistēmas principi un datu modeļi.

Sākotnēji tika izveidotas datu definīcijas, kas bija nepieciešamas moduļa izstrādei. Vēlāk tika pievienoti logi, kas atspoguļoja informāciju. Pēdējais solis bija pati galvenā funkcionalitāte, kas veidoja ierakstus, sasaistīja tos un veica pārējās dokumentācijā minētās prasības

Projekts tika izstrādāts patstāvīgi, izmantojot projekta vadītāja un pieredzējuša programmētāja, kolēģa konsultācijas.

6. KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu produkta kvalitāti tika sastādīta programmatūras prasību specifikācija, programmatūras projektējuma apraksts un testēšanas dokumentācija. Veidojot programmas kodu tika izmantoti šie dokumenti, kā arī tie tika laboti prasību izmaiņu gadījumos.

Programmas kods ir veidots pēc kompānijas iekšienē noteiktajiem principiem. Komentāru nelielais daudzums skaidrojams ar to, ka mainīgo nosaukumi ir veidoti tā, lai pēc tiem varētu saprast to nozīmi un funkcionalitāti un komentāri pievienoti tikai atsevišķās, specifiskās vietās.

Programmas koda izmaiņas tika glabātas SVN, lai varētu atgriezties pie nepieciešamās revīzijas, ja tas būtu nepieciešams.

Programmatūras testēšanā piedalījās ne tikai izstrādātājs, bet arī klients, kuriem programmatūra nonāca jau ļoti agrīnā versijā, tādējādi prasības bieži tika mainītas pakļaujoties konkrētu klientu interesēm.

7. KONFIGURĀCIJU PĀRVALDĪBA

Produkts tiek numurēts pēc katras izmaiņas- izmaiņu numerācija notiek automātiski- katru veikto izmaiņu pievienojot repozitorijam. Tādējādi sanāk, ka katrs commit's ir produkta nākamā versija.

8. DARBIETILPĪBAS NOVĒRTĒJUMS

Darbietilpības novērtēšanai tika izmantots COCOMO modelis. Turpmākajā tekstā redzams aprēķins.

Nosaukums	Tips	Lauku skaits
FIDAVISTA ieraksta izveide importa laikā	Ievads	~20
FIDAVISTA ierakstu pārskats	Izvads	9
FIDAVISTA parametru definēšana	Ievads	~20
FIDAVISTA parametru attēlošana	Izvads	~20
FIDAVISTA aprakstu uzstādījumu definēšana	Ievads	3
FIDAVISTA apraksta uzstādījumu attēlošana	Izvads	3
FIDAVISTA bankas kontu piesaiste kontiem definēšana	Ievads	5
FIDAVISTA bankas kontu piesaiste kontiem attēlošana	Izvads	5
Eksporta specifikācija	Ievads	5
Eksporta faila izveide	Izvads	~20
Atskaites definēšana	Ievads	6
Atskaites izvade	Izvads	4
FIDAVISTA reģistrs	Iekš. Datu fails	DET – 35 RET – 5

Atskaites datu atlasīšana	Vaicājums	6
Eksporta datu atlasīšana	Vaicājums	5

Vienkāršie ievadi un izvadi = $1+3 = 3*1+3*4 = 15$

Vidēji sarežģītie ievadi un izvadi = $3+1 = 4*3+5*1 = 17$

Sarežģītie ievadi un izvadi = $2+2 = 2*6+2*7 = 28$

Vidēji sarežģītie iekšējo datu faili = $1 = 10*1 = 10$

Vidēji sarežģītie vaicājumi = $2 = 4*2 = 8$

Kopējais nepieskaņoto funkcijpunktu skaits ir 78

Tiek pieņemts, ka HAL konstante funkcijpunktu mērīšanai ir ~22. Sanāk, ka kopējais rindu skaits ir 1716. Aptuvenā darbietilpība ir ~3.20 personmēneši un ~3.90 kalendārie mēneši. Aprēķins ir samērā precīzs, jo produkta izstrādei tika patērēti apmēram trīs personmēneši, neņemot vērā prasību saskaņošanu un sastādīšanu, bet tikai reālo programmētāja darbu.

9. IZMANTOTĀ LITERATŪRA

1. [tiešsaiste] www.termini.lv
2. [tiešsaiste] www.hansamanuals.com
3. [tiešsaiste] www.hansaworld.com/WebHalMain.hal
4. [tiešsaiste] www.bankasoc.lv/lka/komitejas/article.php?id=278

10. PROGRAMMATŪRAS KODA DAĻA

Turpmākajā tekstā nav redzams pilnīgs programmatūras kods. Pievienoti ir tikai pārīsi koda fragmenti, lai gūtu priekšstatu par HAL valodas programmēšanas stilu.

10.1. Datu definīcijas

```
SourceFile("halcust/FIDAVISTA",3);

// datadef
begin

RecordBegin(FidaVistaVc,-,"");
RecordField(BankRef,M4Code,60,0);
RecordField(TypeCode,M4Str,5,0);
RecordField(TypeName,M4Str,100,0);
RecordField(RegDate,M4Date,0,0);
RecordField(BookDate,M4Date,0,0);
RecordField(ValueDate,M4Date,0,0);
RecordField(ExtId,M4Str,10,0);
RecordField(BenExtId,M4Str,5,0);
RecordField(DocNo,M4Str,60,0);
RecordField(CorD,M4Str,5,0);
RecordField(AccAmt,M4Val,0,0);
RecordField(FeeAmt,M4Val,0,0);
RecordField(PmtInfo,M4Str,200,0);
RecordField(CPSAccNo,M4Str,60,0);
RecordField(CPSSubAccNo,M4Str,60,0);
RecordField(CPSAHName,M4Str,200,0);
RecordField(CPSAHLegalId,M4Str,20,0);
RecordField(CPSAHAddress,M4Str,70,0);
RecordField(CPSAHExtension,M4Str,255,0);
RecordField(CPSBankCode,M4Str,20,0);
RecordField(CPSBankName,M4Str,200,0);
RecordField(CPSCcy,M4Str,5,0);
RecordField(CPSAmt,M4Val,0,0);
RecordField(CPSCurRate,M4Val,0,0);
RecordField(CPSGiro,M4Str,5,0);
RecordField(CPSExtension,M4Str,255,0);
```

RecordField(Extension,M4Str,255,0);
RecordField(ImportDate,M4Date,0,0);
RecordField(ImportTime,M4Time,0,0);
RecordField(ImportUser,M4Code,0,UserVc);
RecordField(ImportStatus,M4Int,0,0);
RecordField(Math,M4Matrix,0,0);
RecordField(CUCode,M4Code,20,CUVc);
RecordField(DoneFlag,M4Mark,0,0);
RecordField(Ccy,M4Str,5,0);
RecordField(FileName,M4Str,20,0);
RecordField(TransNr,M4Long,0,0);
RecordField(md5hash,M4Str,32,0);
RecordField(OurAccNo,M4Str,60,0);

ArrayBegin(Math,-,1);
ArrayField(LogDate,M4Date,0,0);
ArrayField(LogTime,M4Time,0,0);
ArrayField(LogMessage,M4Str,200,0);
EndArray;

MainKeyBegin(md5hash,0);
KeySegment(md5hash);

KeyBegin(BankRef);
KeySegment(BankRef);
KeySegment(TypeCode);
KeySegment(CPSAHLegalId);
KeySegment(md5hash);

KeyBegin(CorD);
KeySegment(CorD);
KeySegment(RegDate);
KeySegment(BankRef);
KeySegment(TypeCode);
KeySegment(CPSAHLegalId);
KeySegment(md5hash);

KeyBegin(RegDate);
KeySegment(RegDate);
KeySegment(BankRef);

KeySegment(TypeCode);
KeySegment(CPSAHLegalId);
KeySegment(md5hash);

KeyBegin(CPSAHLegalId);
KeySegment(CPSAHLegalId);
KeySegment(RegDate);
KeySegment(BankRef);
KeySegment(TypeCode);
KeySegment(md5hash);

KeyBegin(CPSBankCode);
KeySegment(CPSBankCode);
KeySegment(RegDate);
KeySegment(BankRef);
KeySegment(TypeCode);
KeySegment(CPSAHLegalId);
KeySegment(md5hash);

KeyBegin(TypeCodeLegalId);
KeySegment(TypeCode);
KeySegment(CPSAHLegalId);
KeySegment(RegDate);
KeySegment(BankRef);
KeySegment(md5hash);

KeyBegin(TypeCodeDate);
KeySegment(TypeCode);
KeySegment(RegDate);
KeySegment(BankRef);
KeySegment(CPSAHLegalId);
KeySegment(md5hash);

KeyBegin(CPSCcy); // -- //
KeySegment(CPSCcy);
KeySegment(BankRef);
KeySegment(TypeCode);
KeySegment(CorD);
KeySegment(CPSAHLegalId);
KeySegment(md5hash);

```
KeyBegin(CUCode); // -- //  
KeySegment(CUCode);  
KeySegment(BankRef);  
KeySegment(TypeCode);  
KeySegment(CorD);  
KeySegment(CPSAHLegalId);  
KeySegment(md5hash);
```

```
KeyBegin(BookDate); // -- //  
KeySegment(BookDate);  
KeySegment(BankRef);  
KeySegment(TypeCode);  
KeySegment(CorD);  
KeySegment(CPSAHLegalId);  
KeySegment(md5hash);
```

```
KeyBegin(CPSAHName); // -- //  
KeySegment(CPSAHName);  
KeySegment(BankRef);  
KeySegment(TypeCode);  
KeySegment(CorD);  
KeySegment(CPSAHLegalId);  
KeySegment(md5hash);
```

```
KeyBegin(ImportDate); // -- //  
KeySegment(ImportDate);  
KeySegment(ImportTime);  
KeySegment(BankRef);  
KeySegment(TypeCode);  
KeySegment(CPSAHLegalId);  
KeySegment(md5hash);
```

```
KeyBegin(FileName);  
KeySegment(FileName);  
KeySegment(TransNr);  
KeySegment(BankRef);  
KeySegment(TypeCode);  
KeySegment(CPSAHLegalId);  
KeySegment(md5hash);
```

```
KeyBegin(DoneFlag);
KeySegment(DoneFlag);
KeySegment(BankRef);
KeySegment(TypeCode);
KeySegment(CPSAHLegalId);
KeySegment(md5hash);
EndRecord;
```

```
BlockBegin(FidaVistaPatBlock,-,"");
//BlockInServer;//edz@07.12.10; FV-17
//BlockForAllCompanies;
RecordField(Math,M4Matrix,0,0);
ArrayBegin(Math,-,1);
ArrayField(Pattern,M4Str,80,0);
ArrayField(Type,M4Set,790,0);
ArrayField(Comment,M4Str,100,0);
EndArray;
EndBlock;
```

```
BlockBegin(FidaVistaParamsBlock,-,"");
//BlockInServer;//edz@07.12.10; FV-17
//BlockForAllCompanies;
RecordField(ImportFolder,M4Str,80,0);
RecordField(AfterFolder,M4Str,80,0);
RecordField(ExportFolder,M4Str,80,0);
RecordField(OKIPFlag,M4Int,0,0);
RecordField(Encoding,M4Str,10,0);
RecordField(ImpSales,M4Int,0,0);
RecordField(ImpPurch,M4Int,0,0);
RecordField(ImpTrans,M4Int,0,0);
RecordField(SalesPrepay,M4Int,0,0);
RecordField(PurchPrepay,M4Int,0,0);
RecordField(SalesSum4Match,M4Int,0,0);
RecordField(SalesDebtMatch,M4Int,0,0);
RecordField(AutorunReport,M4Int,0,0);
RecordField(SalesPayOld,M4Int,0,0);//FV-14; edz
RecordField(AllowCcyMismatch,M4Int,0,0);
RecordField(PurchSum4Match,M4Int,0,0);//FV-4; edz:
RecordField(PurchDebtMatch,M4Int,0,0);
```

```

RecordField(PurchPayOld,M4Int,0,0);
RecordField(OKOPFlag,M4Int,0,0);//:edz
RecordField(NoVATNr,M4Int,0,0);//fv-29; edz
RecordField(NoBankAcc,M4Int,0,0);//fv-29; edz
RecordField(OpenCU,M4Int,0,0);//fv-16; edz
RecordField(AttachExisting,M4Int,0,0);//fv-31; edz
RecordField(ExpComment,M4Int,0,0);//fv-42; edz
RecordField(ExpDate,M4Int,0,0);//fv-42; edz
EndBlock;

```

```

BlockBegin(FidaVistaAccBlock,-,"");
RecordField(Math,M4Matrix,0,0);
ArrayBegin(Math,-,1);
ArrayField(BankAcc,M4Str,40,0);
ArrayField(Pattern,M4Str,50,0);
ArrayField(Acc,M4Str,20,0);
ArrayField(Objects,M4UStr,60,0);
ArrayField(Comment,M4Str,100,0);
EndArray;
EndBlock;

```

```

return;
end;

```

10.2. Logu definīcijas

```

// ui
event definewindows
begin
real h,v,vs,vm,h1;
record FidaVistaParamsBlock FidaVistaParamsb;//fv-31; edz

SetLangMode(LangLatvian,"LAT",0);

BlockLoad(FidaVistaParamsb);//fv-31; edz

WindowBegin("FiDAViSta",FidaVistaLClass,CGmlist,-);
UseView(FidaVistaVc);
SpawnClass(FidaVistaDCClass);
SetWRect(20,80,890,298);
Sizeable(-1,-1,0,0);

```

```

RecordList(4,17,-21,-39);
RlColKey(h=4,h,"BankRef",BankRef,false,BankRef);
RlColKey(h+=120,h,"BookDate",BookDate,false,BookDate);
RlColKey(h+=75,h,"Done",DoneFlag,false,DoneFlag);
RlColKey(h+=40,h,"CPSAHLegalId",CPSAHLegalId,false,CPSAHLegalId);
RlColKey(h+=95,h,"CPSAHName",CPSAHName,false,CPSAHName);
RlColKey(h+=200,h,"CUCode",CUCode,false,CUCode);
RlCol(h+=80,h,"PmtInfo",PmtInfo,false);
RlCol(-70,-70,"AccAmt",AccAmt,true);
RlColKey(-10,-10,"",CorD,false,CorD);
SpecialMenu("Atkārtot ierakstu piesaisti",!0,"FV_LinkDataLsm");//fv-25; edz
SearchField("Meklēt");
EndWindow;

```

```

vs = 20;
WindowBegin("FiDAViSta",FidaVistaDClass,CGview,-);
DynamicWindow;//fv-31; edz
UseView(FidaVistaVc);
SetWRect(20,80,300,650);
Sizeable(-1,-1,0,0);
StaticText(150,v=15,"Operācijas dati",false);
EditField(100,v=25,150,"Op.numurs",ViewOnly,BankRef,false,0);
EditField(100,v+=vs,70,"Reģ.dat.",ViewOnly,RegDate,false,0);
EditField(250,v,50,"Op.tips",ViewOnly,TypeCode,false,0);
EditField(100,v+=vs,70,"Grāmat.dat.",ViewOnly,BookDate,false,0);
EditField(250,v,50,"Op.nosauk.",ViewOnly,TypeName,false,0);
EditField(100,v+=vs,70,"Valut.dat.",ViewOnly,ValueDate,false,0);
EditField(250,v,20,"Operācija",ViewOnly,CorD,false,0);
// EditField(100,v+=vs,100,"ExtId",ViewOnly,ExtId,false,0);
// EditField(100,v+=vs,100,"BenExtId",ViewOnly,BenExtId,false,0);
EditField(100,v+=vs,100,"Dok.numurs",ViewOnly,DocNo,false,0);
EditField(250,v,200,"Konts",ViewOnly,OurAccNo,false,0);
EditField(100,v+=vs,70,"Summa",ViewOnly,AccAmt,true,0);
EditField(250,v,70,"Valūta",ViewOnly,Ccy,false,0);
EditField(380,v,70,"Komisija",ViewOnly,FeeAmt,true,0);
EditField(100,v+=vs,500,"Maks.mērķis",ViewOnly,PmtInfo,false,0);

StaticText(150,v+=40,"Pretējās puses dati",false);
EditField(100,v+=10,150,"Konta nr.",ViewOnly,CPSAccNo,false,0);
EditField(350,v,100,"Apakšk.nr.",ViewOnly,CPSSubAccNo,false,0);

```

```

EditField(500,v,100,"Žiro",ViewOnly,CPSGiro,false,0);
EditField(100,v+=vs,70,"Banka",ViewOnly,CPSBankCode,false,0);
EditField(250,v,200,"Nosauk.",ViewOnly,CPSBankName,false,0);
EditField(100,v+=vs,200,"Nosaukums",ViewOnly,CPSAHName,false,0);
EditField(350,v,100,"Reģ.nr.",ViewOnly,CPSAHLegalId,false,0);
EditField(100,v+=vs,350,"Adrese",ViewOnly,CPSAHAddress,false,0);
if (FidaVistaParamsb.AttachExisting==1) then begin//fv-31; edz:
    EditField(-150,v,110,"Klienta kods",normal,CUCode,false,CUSClass);
end else begin
    EditField(-150,v,110,"Klienta kods",ViewOnly,CUCode,false,CUSClass);
end;//:edz
// EditField(100,v+=vs,100,"PP papildlauks",ViewOnly,CPSAHExtension,false,0);
EditField(100,v+=vs,70,"Summa",ViewOnly,CPSAmt,true,0);
EditField(250,v,50,"Valūta",ViewOnly,CPSCcy,false,0);
EditField(350,v,70,"Kurss",ViewOnly,CPSCurRate,true,0);
// EditField(100,v+=vs,100,"PP papildlauks",ViewOnly,CPSEExtension,false,0);
// EditField(100,v+=vs,100,"Papildlauks",ViewOnly,Extension,false,0);
EditField(-150,v,30,"Dok.nr.",ViewOnly,FileName,false,0);
EditField(-110,v,70,"",ViewOnly,TransNr,false,0);

StaticText(150,v+=45,"Ieraksta importa vēsture",false);
MatrixBegin(27,370,-21,-20,-,99);
MatCol(1,24,"Datums",0,LogDate,false,0);
MatCol(1,99,"Laiks",0,LogTime,false,0);
MatCol(1,154,"Ziņojums",0,LogMessage,false,0);
EndMatrix;

StaticText(510,v=15,"Importa dati",false);
ButtonFrame(-150,v=40,120,3,"Statuss");
RadioButton(-150,v,0,0,"Importēts fails",ImportStatus);
RadioButton(-150,v+=16,1,0,"Maks. nav izveidots",ImportStatus);
RadioButton(-150,v+=16,2,0,"Izveidots maksājums",ImportStatus);
CheckBox(-150,v+=28,1,"Apstrāde pabeigta",DoneFlag);
EditField(470,v=25,70,"Datums",ViewOnly,ImportDate,false,0);
EditField(470,v+=vs,50,"Laiks",ViewOnly,ImportTime,false,0);
EditField(470,v+=vs,40,"Veicējs",ViewOnly,ImportUser,false,0);

SpecialMenu("Atkārtot ieraksta piesaisti",'l',0,"FV_LinkDataDsm");
SpecialMenu("Veidot jaunu kontaktu",'c',0,"FV_CreateNewContact");//FV-16; edz
SpecialMenu("Izveidot maksājumu",'p',0,"FV_CreatePayment");//FV-15; edz

```

```
SpecialMenu("Ielasīt informāciju no piesaistītā kontakta",'r',0,"FV_LinkContactData");//FV-30; edz  
EndWindow;
```

```
WindowBegin("Definēt FiDAViSta aprakstu  
uzstādījumus",FidaVistaPatTClass,CGHbl,HBlRecType);  
SetBlid(FidaVistaPatBlock);  
SetWRect(20,80,400,298);  
Sizeable(-1,-1,0,0);  
h = 8; v = 20;  
h1=34;  
MatrixBegin(h,v,-31,-20,-,999);  
MatCol(1,h1,"Raksts (pattern)",0,Pattern,false,0);  
MatCol(1,h1+=85,"Tips",0,Type,false,SetSClass);  
MatCol(1,h1+=85,"Komentārs",0,Comment,false,0);  
EndMatrix;  
EndWindow;
```

```
WindowBegin("Definēt FiDAViSta bankas kontu piesaisti  
kontiem",FidaVistaAccTClass,CGHbl,HBlRecType);  
SetBlid(FidaVistaAccBlock);  
SetWRect(20,80,950,298);  
Sizeable(-1,-1,0,0);  
h = 8; v = 20;  
h1=34;  
MatrixBegin(h,v,-31,-20,-,999);  
MatCol(1,h1,"Bankas kots",0,BankAcc,false,0);  
MatCol(1,h1+=185,"Raksts (pattern)",0,Pattern,false,0);  
MatCol(1,h1+=185,"Kots",0,Acc,false,AccSClass);  
MatCol(1,h1+=85,"Objekti",0,Objects,false,ObjSClass);  
MatCol(1,h1+=115,"Komentārs",0,Comment,false,0);  
EndMatrix;  
EndWindow;
```

```
vs = 20;
```

```
h = 70;
```

```
h1 = 50;
```

```
v = 0;
```

```
WindowBegin("FiDAViSta - Parametri",FidaVistaParamsTClass,CGsBl,-);  
SetBlid(FidaVistaParamsBlock);  
SetWRect(64,80,650,500);
```

```

Tile(0,125,false,"0",ImportFolder);
EditField(h=200,v=6,200,"Importa mape",Normal,ImportFolder,false,0);
EditField(h,v+=vs,200,"Mape pēc importa",Normal,AfterFolder,false,0);
EditField(h,v+=vs,200,"Eksporta mape",Normal,ExportFolder,false,0);
EditField(h,v+=vs,40,"Importa failu kodējums",Normal,Encoding,false,0);
StaticText(h+50,v+14,"(piemēram, UTF8, CP1257)",false);
TileButton(1,"Maksājumi");
TileButton(2,"Maksājumu uzdevumi");
TileButton(3,"Kontaktu veidošana");
Tile(1,200,false,"1",ImpSales);
ButtonFrame(h1,vs,320,5,"Maksājumu imports:");
CheckBox(h1,vs,0,"Grāmatot",OKIPFlag);
CheckBox(h1,vs+=16,0,"Piesaistīt, ja sakrīt rēķina summa",SalesSum4Match);
CheckBox(h1,vs+=16,0,"Piesaistīt, ja sakrīt kopējā parāda summa",SalesDebtMatch);
CheckBox(h1,vs+=16,0,"Piesaistīt, ja nesakrīt valūta identificētam rēķinam",AllowCcyMismatch);
CheckBox(h1,vs+=16,0,"Neatšifrēti maksājumi apmaksā vecākos rēķinus",SalesPayOld);//FV-14; edz
ButtonFrame(h1,vs+=45,320,1,"Importējot veidot:");
CheckBox(h1,vs,0,"Maksājumus",ImpSales);
ButtonFrame(h1,vs+=45,320,1,"Ja nav nekas nav uzminēts, veidot priekšapmaksas:");
CheckBox(h1,vs,0,"Maksājumus",SalesPrepay);
Tile(2,260,false,"2",ImpPurch);
ButtonFrame(h1,vs=20,320,4,"Maksājumu uzdevumu imports:");//FV-4; edz:
CheckBox(h1,vs,0,"Grāmatot",OKOPFlag);
CheckBox(h1,vs+=16,0,"Piesaistīt, ja sakrīt rēķina summa",PurchSum4Match);
CheckBox(h1,vs+=16,0,"Piesaistīt, ja sakrīt kopējā parāda summa",PurchDebtMatch);
CheckBox(h1,vs+=16,0,"Neatšifrēti maksājumu uzdevumi apmaksā vecākos
rēķinus",PurchPayOld);//:edz
ButtonFrame(h1,vs+=45,320,2,"Importējot veidot:");
CheckBox(h1,vs,0,"Maksājumu uzdevumus",ImpPurch);
CheckBox(h1,vs+=16,0,"Grāmatojumus komisijām u.c.",ImpTrans);
ButtonFrame(h1,vs+=45,320,1,"Ja nav nekas nav uzminēts, veidot priekšapmaksas:");
CheckBox(h1,vs,0,"Maksājumu uzdevumus",PurchPrepay);
ButtonFrame(h1,vs+=45,320,2,"Maksājumu uzdevumā eksportēt");//fv-22; edz:
CheckBox(h1,vs,0,"MU komentāru",ExpComment);
CheckBox(h1,vs+=16,0,"Rēķina datumu",ExpDate);//:edz
Tile(3,200,false,"3",NoVATNr);
ButtonFrame(h1,vs=20,320,4,"Jauna kontakta veidošana:");//FV-4; edz:
CheckBox(h1,vs,0,"Veidot kontaktu, ja FV ierakstā nav reģ.nr.",NoVATNr);//fv-29; edz
CheckBox(h1,vs+=16,0,"Veidot kontaktu, ja FV ierakstā nav bankas konta",NoBankAcc);//fv-29; edz
CheckBox(h1,vs+=16,0,"Pēc izveidošanas atvērt kontakta ierakstu",OpenCU);//fv-16; edz

```

```
CheckBox(h1,v+=16,0,"Ļaut piesaistīt ierakstam esošu kontaktu",AttachExisting);//fv-31; edz  
EndWindow;
```

```
vs = 20;
```

```
WindowBegin("Sagatavot FiDAViSta importa atskaiti",FidaVistaRClass,CGRcW,RcType);
```

```
SetWRect(20,80,450,200);
```

```
PeriodField(h=170,v=6,"Importa periods");
```

```
EditField(h,v+=vs,70,"Importa laiks",Normal,f1,false,PasteFVImportSClass);
```

```
EditField(h,v+=vs,150,"Oper. tips",Normal,f2,false,0);
```

```
ButtonFrame(h1=170,v+=2*vs,150,3,"Rādīt");
```

```
CheckBox(h1=170,v,0,"Importēts fails",flags[0]);
```

```
CheckBox(h1,v+=16,0,"Maksājums nav izveidots",flags[1]);
```

```
CheckBox(h1,v+=16,0,"Izveidots maksājums",flags[2]);
```

```
EndWindow;
```

```
//fv-43; edz:
```

```
WindowBegin("Ievietot importu",PasteFVImportSClass,CGoother,HBIRcType);
```

```
SetWRect(240,80,600,300);
```

```
Sizeable(-1,-1,0,0);
```

```
StringList(0,0,600,-25);
```

```
SearchField("Search");
```

```
EndWindow;
```

```
//:edz
```

```
return;
```

```
end;
```

10.3. Importa funkcijas fragments

```
global
```

```
updating procedure FidaVistaMainIn(area ar)
```

```
begin
```

```
record FidaVistaVc FVr,FV2r;
```

```
record RcVc RepSpec;
```

```
xml xdata;
```

```
Boolean testf;
```

```
string 255 xstr,xstr_statement,xstr_acc,xstr_ccy,xstr_trx,tstr,starttime,typecode,cord,ourbankacc;
```

```
Integer err;
```

```
longint i,c_acc,c_trx,c_ccy;
```

```
date d;
```

```
boolean exists;
```

```

integer counter;
string 32 md5hash;
record FidaVistaParamsBlock FVParamsr;
longint pos,count;//fv-35; edz
string 9 tstr2;//fv-35; edz

        BlockLoad(FVParamsr);

FV_SetImportCodePage;

xdata = ParseXMLArea(ar);

xstr_statement = "FIDAVISTA/Statement";
counter = 0;
starttime = CurrentTime;

c_acc = 0;
while (XMLNodeExists(xdata,xstr_statement & "/AccountSet[" & c_acc & "]")) begin
    xstr_acc = "/AccountSet[" & c_acc & "]";
    ourbankacc = XMLGet(xdata,xstr_statement & xstr_acc & "/AccNo");
    c_ccy = 0;
    while (XMLNodeExists(xdata,xstr_statement & xstr_acc & "/CcyStmt[" & c_ccy & "]")) begin
        xstr_ccy = "/CcyStmt[" & c_ccy & "]";
        c_trx = 0;
        while (XmlNodeExists(xdata,xstr_statement & xstr_acc & xstr_ccy & "/TrxSet[" & c_trx & "]"))
begin
            xstr_trx = "/TrxSet[" & c_trx & "]";
            xstr = xstr_statement & xstr_acc & xstr_ccy & xstr_trx;
            typecode = XMLGet(xdata,xstr & "TypeCode");
            cord = XMLGet(xdata,xstr & "CorD");
            if (blank(typecode)) then begin
                if (cord == "C") then begin
                    typecode = "INP";
                end else begin
                    typecode = "OUTB";
                end;
            end;
        end;

        md5hash = md5string(XMLGet(xdata,xstr & "BankRef") &
typecode & XMLGet(xdata,xstr & "CPartySet/AccHolder/LegalId") & cord & XMLGet(xdata,xstr &
"TypeName") & XMLGet(xdata,xstr & "RegDate") & XMLGet(xdata,xstr & "ExtId") &

```

```

XMLGet(xdata,xstr & "BenExtId") & XMLGet(xdata,xstr & "DocNo"));
if (!FidaVistaRecordExists(md5hash,FVr)) then begin
    RecordNew(FVr);
                                FVr.md5hash = md5hash;
    FVr.BankRef = XMLGet(xdata,xstr & "BankRef");
    FVr.TypeCode = typecode;
    FVr.CPSAHLegalId = XMLGet(xdata,xstr & "CPartySet/AccHolder/LegalId");
    FVr.OurAccNo = ourbankacc;
    exists = false;
end else begin
    RecordCopy(FV2r,FVr);
    exists = true;
end;
if (FVr.DoneFlag == 0) then begin
    FVr.Ccy = XMLGet(xdata,xstr_statement & xstr_acc & xstr_ccy & "/Ccy[0]");
    FVr.TypeName = XMLGet(xdata,xstr & "TypeName");
    tstr = XMLGet(xdata,xstr & "RegDate");
    d.Year = StringToInt(left(tstr,4));
    d.Month = StringToInt(mid(tstr,5,2));
    d.Day = StringToInt(right(tstr,2));
    FVr.RegDate = d;
    tstr = XMLGet(xdata,xstr & "BookDate");
    d.Year = StringToInt(left(tstr,4));
    d.Month = StringToInt(mid(tstr,5,2));
    d.Day = StringToInt(right(tstr,2));
    FVr.BookDate = d;
    tstr = XMLGet(xdata,xstr & "ValueDate");
    d.Year = StringToInt(left(tstr,4));
    d.Month = StringToInt(mid(tstr,5,2));
    d.Day = StringToInt(right(tstr,2));
    FVr.ValueDate = d;
    FVr.ExtId = XMLGet(xdata,xstr & "ExtId");
    FVr.BenExtId = XMLGet(xdata,xstr & "BenExtId");
    FVr.DocNo = XMLGet(xdata,xstr & "DocNo");
    FVr.CorD = XMLGet(xdata,xstr & "CorD");
    FVr.AccAmt = StringToVal(XMLGet(xdata,xstr & "AccAmt"),M4Val);
    FVr.FeeAmt = StringToVal(XMLGet(xdata,xstr & "FeeAmt"),M4Val);
    FVr.PmtInfo = XMLGet(xdata,xstr & "PmtInfo");
    FVr.CPSAccNo = XMLGet(xdata,xstr & "CPartySet/AccNo");
    FVr.CPSSubAccNo = XMLGet(xdata,xstr & "CPartySet/SubAccNo");

```

```

FVr.CPSAHName = XMLGet(xdata,xstr & "CPartySet/AccHolder/Name");
//FVr.CPSAHLegalId = XMLGet(xdata,xstr & "CPartySet/AccHolder/LegalId");
FVr.CPSAHAddress = XMLGet(xdata,xstr & "CPartySet/AccHolder/Address");
FVr.CPSAHExtension = XMLGet(xdata,xstr & "CPartySet/AccHolder/Extension");
FVr.CPSBankCode = XMLGet(xdata,xstr & "CPartySet/BankCode");
FVr.CPSBankName = XMLGet(xdata,xstr & "CPartySet/CPSBankName");
FVr.CPSCcy = XMLGet(xdata,xstr & "CPartySet/Ccy");
FVr.CPSAmt = StringToVal(XMLGet(xdata,xstr & "CPartySet/Amt"),M4Val);
FVr.CPSCurRate = StringToVal(XMLGet(xdata,xstr & "CPartySet/CurRate"),M4Val);
FVr.CPSGiro = XMLGet(xdata,xstr & "CPartySet/Giro");
FVr.CPSExtension = XMLGet(xdata,xstr & "CPartySet/Extension");
FVr.Extension = XMLGet(xdata,xstr & "Extension");

FVr.ImportDate = CurrentDate;
FVr.ImportTime = CurrentTime;
FVr.ImportUser = CurrentUser;
if (!exists) then begin
    FVr.ImportStatus = 0;
end;

LogFVMMessage(FVr,USetStr(31950) & GetImportFileName);

// NEED setting - do we want to link straight away or run maintenance manually?
FidaVista_LinkImportData(FVr,FVParamsr);
counter = counter + 1;

if (!exists) then begin
    RecordInsert(FVr,true);
end else begin
    RecordUpdate(FV2r,FVr,true);
end;
end;
c_trx = c_trx + 1;
end;
c_ccy = c_ccy + 1;
end;
c_acc = c_acc + 1;
end;

//fv-35; edz:

```

```

count = 0;
while (tstr2!="#</FIDAVIS") begin
  tstr2 = GetStringFromArea(ar,pos,9);
  if (tstr2=="</TrxSet>") then begin
    count=count+1;
  end;
  pos = pos+1;
end;
MessageBox(0,USetStr(31979) & count & USetStr(31980) & USetStr(31981) & counter &
USetStr(31980));
//:edz

while (!tsteof()) begin
  nextimportline(false);
end;

return;
end;

global
updating procedure FV_ImportMaster()
BEGIN
  record FidaVistaParamsBlock Setup;
  integer c,i;
  area ar;
  string 255 fn,ofn,nfn,nofn;

  BlockLoad(Setup);
  c = CountFilesInDir(Setup.ImportFolder);
  LogText(0,"FidaVista ImportMaster started (" & c & " files in " & Setup.ImportFolder & ")");
  for (i=c-1;i>=0;i=i-1) begin
    fn = GetFileNameInDir(Setup.ImportFolder,i);
    ofn = Setup.ImportFolder & "\\ " & fn;
      //"
    FV_AreaFromFile(ar,ofn);
      FidaVistaMainIn(ar);
    /*
    nfn = DateToString(CurrentDate,"YYYYMMDD") & TimeToString(CurrentTime) & "_" & fn;
    nofn = Setup.ImportFolder & "\\ " & nfn;
      //"

```

```

if (RenameFile(ofn, nofn)==0) then begin
    MoveFile(nfn,Setup.ImportFolder,Setup.AfterFolder);
end;
*/
nfn = DateToString(CurrentDate,"YYYYMMDD") & TimeToString(CurrentTime) & "_" & fn;
nofn = Setup.AfterFolder & "\\\" & nfn;
    //"
CreateFile(nofn);
CloseFile;
WriteAreaToFile(ar,nofn,0);
Delete_File(ofn);
    end;
    return;
END;

global
updating procedure FidaVistaIn()
BEGIN
    record MailVc Mailr;
    area ar;
    longint pos,textsize;
        integer aaa;
    string 1 tstr;
    string 255 nfn,nofn,tstr2;
    record FidaVistaParamsBlock Setup;

        FV_SetImportCodePage;

BlockLoad(Setup);

while (!testeof()) begin
    ImportIntoText(Mailr);
    nextimportline(false);
end;
    CloseFile;

textsize = SizeTextCnt(Mailr);
while (pos<textsize) begin
    tstr2 = StringFromText(Mailr,pos,250);
    pos = pos + 250;

```

```
AddTextToArea(tstr2,ar);
end;

if (nonblank(Setup.ImportFolder)) then begin
  nfn = DateToString(CurrentDate,"YYYYMMDD") & TimeToString(CurrentTime) & ".xml";
  nofn = Setup.ImportFolder & "\\\" & nfn;
      //"
  CreateFile(nofn);
  CloseFile;
  WriteAreaToFile(ar,nofn,0);
  //FV_ImportMaster;
end else begin
  LogText(0,"FidaVista: set up Import folder!");
end;

FidaVistaMainIn(ar);

//FidaVistaMainIn(GetImportFileName);
END;
```

11. DOKUMENTĀRĀ LAPA

Kvalifikācijas darbs “Maksājumu eksports/imports HansaWorld sistēmā FIDAVISTA formātā” 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: Edgars Dzenis

Rekomendēju darbu aizstāvēšanai

Vadītājs: Dainis Dosbergs

Recenzents:

Darbs iesniegts Datorikas nodaļā 26.05.2011.

Metodiķis:

Darbs aizstāvēts pirmā līmeņa profesionālās augstākās izglītības studiju programmas komisijas sēdē

Komisijas sekretārs: