

**LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
DATORIKAS NODAĻA**

NOLIKTAVAS PĀRVALDĪBAS SISTĒMA

KVALIFIKĀCIJAS DARBS

Autors: **Raimonds Spricis**

Stud. apl RS05017

Darba vadītājs: docents Dr. dat. **Ģirts Karnītis**

Rīga 2007

Anotācija

Šajā kvalifikācijas darbā ir aprakstīta MS Windows lietotne „Noliktavas pārvaldības sistēma”. Darba ietvaros tika izstrādāta pati lietotne, kā arī tās dokumentācija. Programmatūras lietotāji ir noliktavas menedžeri. Lietotne izmanto MS SQL Server datu bāzi, lai uzglabātu un parādītu lietotājiem noliktavas datus. Tās galvenās funkcijas ir preču kataloga, noliktavas partneru un pasūtījumu datu pārvaldība.

Аннотация

В этой квалификационной работе описана MS Windows аппликация «Управления склада». В рамках работы разработана сама аппликация а также её документация. Пользователи программатурой являются менеджеры складов. Аппликация использует базу данных MS SQL Server для хранения и показа пользователям данных склада. Главными её функциями являются управление каталога товаров, партнеров склада и данных по заказам.

Abstract

In this qualification work is described MS Windows application “Warehouse Management System”. Scope of this work includes developed application and its documentation. Software users are warehouse managers. Application uses MS SQL Server database to store and show warehouse data. Its primary functions are product catalog, warehouse partner and order data management.

SATURS

IEVADS	1
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	3
1.1. Ievads.....	3
1.1.1. Nolūks.....	3
1.1.2. Darbības sfēra	3
1.1.3. Definīcijas, akronīmi un saīsinājumi	3
1.1.4. Saistība ar citiem dokumentiem	4
1.1.5. Dokumenta pārskats	4
1.2. Programmatūras vispārējs apraksts	5
1.2.1. Programmatūras funkcijas	5
1.2.2. Programmatūras perspektīvas.....	5
1.2.3. Lietotāju raksturiezīmes	6
1.2.4. Vispārēji ierobežojumi	6
1.2.5. Pieņēmumi un atkarības.....	6
1.3. FUNKCIONĀLĀS PRASĪBAS	7
1.3.1. Darba uzsākšana	7
1.3.2. Darba beigšana	8
1.3.3. Preču kataloga apskate	9
1.3.4. Kategoriju pievienošana, labošana un dzēšana.....	11
1.3.5. Preču pievienošana, labošana un dzēšana.....	12
1.3.6. Partnera piesaistīšana produktam vai pasūtījumam.....	14
1.3.7. Partneru datu apskate.....	15
1.3.8. Partneru datu pievienošana, labošana un dzēšana	16
1.3.9. Pasūtījumu datu apskate	18
1.3.10. Pasūtījumu datu pievienošana, labošana un dzēšana.....	20
1.3.11. Preču piesaistīšana pasūtījumam	22
1.3.12. Noliktavas rekvizītu apskate.....	23
1.3.13. Noliktavas rekvizītu pievienošana, labošana un dzēšana	24

1.3.14. Informācijas par sistēmu apskate.....	25
1.4. Ārējā saskarne.....	26
1.4.1. Lietotāja saskarne	26
1.4.2. Programmatūras saskarne	32
1.5. NEFUNKCIONĀLĀS PRASĪBAS.....	33
1.5.1. Veiktspējas prasības	33
1.5.2. Pieejamība	33
1.5.3. Datu drošība.....	33
1.5.4. Aizsardzība pret nesankcionētu piekļuvi.....	33
2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	34
2.1. Ievads.....	34
2.1.1. Nolūks.....	34
2.1.2. Darbības sfēra	34
2.1.3. Izstrādes metodes.....	34
2.1.4. Definīcijas, akronīmi un saīsinājumi	35
2.1.5. Saistība ar citiem dokumentiem	35
2.1.6. Dokumenta pārskats	35
2.2. Klašu detalizēts projektējums	36
2.3. Datu detalizēts projektējums	39
2.3.1. Tabula <i>Warehouse_details</i>	39
2.3.2. Tabula <i>Partners</i>	39
2.3.3. Tabula <i>Categories</i>	40
2.3.4. Tabula <i>Products</i>	40
2.3.5. Tabula <i>Orders</i>	41
2.3.6. Tabula <i>Orders_items</i>	42
2.4. Atkarību apraksts.....	43
2.4.1. Datu atkarības – ER realizācijas modelis	43
3. TESTĒŠANAS DOKUMENTĀCIJA	44
3.1. Testpiemēru specifikācija	44
3.2. Testēšanas žurnāls	46
4. PROJEKTA REALIZĀCIJA.....	48
4.1. Projekta organizācija	48
4.2. Kvalitātes nodrošināšana	48
4.3. Konfigurāciju pārvaldība.....	48
4.4. Darbietilpības novērtējums.....	49

5. PROGRAMMAS KODS	51
5.1. Fails „database.sql”	51
5.2. Fails „App_main.vb”	53
5.3. Fails „Login_form.vb”	55
5.4. Fails „Edit_product.vb”	57
SECINĀJUMI	60
IZMANTOTĀ LITERATŪRA UN AVOTI.....	61

IEVADS

Noliktavu darba specifika nosaka darbības ar dažāda veida datiem – detalizētu noliktavā esošo preču jeb produktu sarakstu uzturēšanu, partneru kartotēkas veidošanu, ienākošo un izejošo pasūtījumu datu reģistrēšanu. Ātrākam un efektīvākam darbam šos procesus ir nepieciešams elektronizēt, turklāt ir vēlams, lai visus minētos darba uzdevumus varētu veikt ar vienu, centralizētu programmu sistēmu.

Kvalifikācijas darba ietvaros tika izstrādāta MS Windows lietotne „Noliktavas pārvaldības sistēma” (angl. val. „Warehouse Management System”, saīsināti WMS), kā arī nepieciešamā datu bāze un lietotnes dokumentācija. Programmatūras lietotāji – noliktavas menedžeri – šajā programmā var veidot preču katalogu, noliktavas partneru (klientu un piegādātāju) kartotēku, veikto ienākošo un izejošo pasūtījumu reģistru. Papildus ir iespēja saglabāt sistēmā noliktavas rekvizītus. Lietotni un datu bāzi var lietot tikai lietotāji, kam ir atļauta pieeja datiem datu bāzē, tādējādi nodrošinot sistēmas datu drošību.

Darba pirmajā nodaļā ir iekļauta izstrādājamās programmatūras prasību specifikācija. Otrajā nodaļā ir programmatūras projektējuma apraksts. Trešā nodaļa satur testēšanas dokumentāciju. Ceturtajā nodaļā ir aprakstīta projekta tapšana. Piektajā nodaļā ir iekļauti izstrādātā programmas koda fragmenti.

Darba mērķis:

- Izstrādāt MS Windows lietotni „Noliktavas pārvaldības sistēma” un tās datu bāzi.

Darba uzdevumi:

- Izstrādāt programmas kodu.
- Izstrādāt programmatūras prasību specifikāciju un projektējuma aprakstu.
- Izstrādāt zema līmeņa (datu struktūru un algoritmu) projektējumu.
- Izstrādāt testēšanas dokumentāciju.
- Uzrakstīt paskaidrojošu tekstu, kurā atspoguļota projekta organizācija, kvalitātes nodrošināšana, konfigurāciju pārvaldība un dots darbietilpības novērtējums saskaņā ar izplatītām metodēm.

Izmantotā metodika:

- Projekts tika izstrādāts individuāli. Gan programmas kodu, gan visu dokumentāciju izstrādāja darba autors.
- Lietotne tika realizēta programmēšanas valodā Visual Basic .NET, izmantojot izstrādes vidi Visual Studio .NET (2003) un platformu .NET Framework 1.1. Ekrāna formas tika veidotas ar Visual Studio Windows Form Designer; to programmas kods tika ģenerēts automātiski. Savukārt visa formu funkcionalitāte tika programmēta, rakstot kodu. Datu bāzes izveidei tika izmantots MS SQL Server.
- Programmas koda izstrādē informācija tika iegūta no avotiem [1-3, 5-6, 8].
- Programmatūras prasību specifikācijas izstrādē tika izmantoti avoti [4, 7].
- Programmatūras projektējuma apraksta izstrādē tika izmantots avots [7].

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Šī dokumenta nolūks ir aprakstīt programmatūras prasības programmu sistēmai „Noliktavas pārvaldības sistēma” (angl. val. „Warehouse Management System”, saīsināti WMS).

Dokuments ir paredzēts šādiem lietotājiem:

- Programmatūras izstrādātājiem, kas veic WMS izstrādi, lai noformulētu izstrādājamās programmas prasības.
- Potenciālajiem papildu programmatūras izstrādātājiem sistēmas papildmoduļu izstrādē.

1.1.2. Darbības sfēra

Programmu sistēma „Noliktavas pārvaldības sistēma” ir vienlietotāja MS Windows lietotne, kuras darbības mērķis ir noliktavas datu pārvaldība. Lietotne pieslēdzas kopējai noliktavas datu bāzei un ļauj iegūt un rediģēt nepieciešamos datus. Projekta ietvaros tiek izstrādāta gan lietotne, gan datu bāze. Programmatūra nav paredzēta lielām slodzēm, tās potenciālie izmantotāji ir maza un vidēja izmēra noliktavas. Noliktavas menedžeris ar programmas palīdzību var kontrolēt gan preču jeb produktu katalogu, gan noliktavas partneru sarakstu, gan operatīvi iegūt nepieciešamo informāciju priekš klientiem, pavadzīmju veidošanas u.c., gan reģistrēt ienākošos un izejošos pasūtījumus. WMS pamatuzdevumi ir datu pievienošana un rediģēšana preču katalogā, partneru pārvaldība, pasūtījumu reģistrēšana, papildinformācijas sniegšana par noliktavu un pašu sistēmu, kā arī datu aizsardzība pret nesankcionētu piekļuvi.

1.1.3. Definīcijas, akronīmi un saīsinājumi

WMS – saīsinājums no „Warehouse Management System” (latv. val. „Noliktavas pārvaldības sistēma”).

MS – Microsoft.

VPN (Virtual Private Network) – privāts sakaru tīkls, ko izmanto drošai attālinātu sistēmu saziņai caur publisko tīklu.

Windows NT Integrated Security – MS SQL Server lietotāja autentifikācijas veids, ar kura palīdzību serveris lietotājus atpazīst pēc Windows NT lietotāja ID koda.

Pieteikšanās – lietotāja identificēšanās procedūra, pievienojoties datoram pa sakaru līniju. Šīs procedūras laikā dators parasti pieprasa lietotāja vārdu un paroli.

Atteikšanās – datorsakaru sesijas pārtraukšanas procedūra.

Izņēmuma situācija (angl. val. Exception) – kļūdas situācija programmas darbības laikā, kas maina vai pārtrauc tās normālu izpildes gaitu.

Modālais dialoglogs – lietotāja saskarnes dialoga logs, pēc kura atvēršanas lietotājs nevar aktivizēt pamatlogu, kamēr dialoga logā nav nospiesta kāda no kontroles pogām.

Funkcionālo prasību aprakstā lietotie tekstlodziņu ievades datu tipi:

Text(x–y) – simbolu virkne, kas satur jebkādu ASCII kodējuma simbolus. x – minimālais virknes garums; y – maksimālais virknes garums. Ja garumu ierobežojumi nav norādīti, simbolu virknei var būt jebkāds garums.

uText(x–y) – simbolu virkne, kas satur jebkādu Unicode kodējuma simbolus. x – minimālais virknes garums; y – maksimālais virknes garums.

Integer(x) – vesels skaitlis; x – minimālā pieļaujamā vērtība.

Float(x) – decimāldaļskaitlis; x – minimālā pieļaujamā vērtība.

1.1.4. Saistība ar citiem dokumentiem

Dokuments sagatavots atbilstoši standartam LVS 68:1996 „Informācijas tehnoloģija – Programminženierija – Programmatūras prasību specifikācijas ceļvedis”.

1.1.5. Dokumenta pārskats

Dokuments ir sadalīts 5 nodaļās. Pirmajā nodaļā ir ievadinformācija par dokumentu un programmatūru. Otrajā nodaļā ir aprakstītas vispārējās prasības izstrādājamajai programmatūrai. Trešajā nodaļā ir aprakstītas programmatūras funkcionālās prasības. Ceturtajā nodaļā ir aplūkota ārējā saskarne. Piektajā nodaļā ir aprakstītas nefunkcionālās prasības.

1.2. Programmatūras vispārējs apraksts

1.2.1. Programmatūras funkcijas

Sarakstā ir uzskaitītas WMS funkcijas, grupējot pēc uzdevumiem, kuri noteikti nodaļā *1.1.2. Darbības sfēra*.

- Aizsardzība pret nesankcionētu piekļuvi
 - Darba uzsākšana
 - Darba beigšana
- Preču kataloga pārvaldība
 - Preču kataloga apskate
 - Kategoriju pievienošana, labošana un dzēšana
 - Preču pievienošana, labošana un dzēšana
- Noliktavas partneru pārvaldība
 - Partnera piesaistīšana produktam vai pasūtījumam
 - Partneru datu apskate
 - Partneru datu pievienošana, labošana un dzēšana
- Pasūtījumu pārvaldība
 - Pasūtījumu datu apskate
 - Pasūtījumu datu pievienošana, labošana un dzēšana
 - Preču piesaistīšana pasūtījumam
- Noliktavas rekvizītu apskate un rediģēšana
 - Noliktavas rekvizītu apskate
 - Noliktavas rekvizītu pievienošana, labošana un dzēšana
- Papildinformācijas ieguve
 - Informācijas par sistēmu apskate

1.2.2. Programmatūras perspektīvas

WMS ir jauna, patstāvīga sistēma ar datu bāzi. Tās vienīgā programmatūras saskarne ir starp WMS lietotni un MS SQL Server datu bāzi. Perspektīvā ir iespēja izveidot programmatūru, kas izmantos šo pašu datu bāzi, piemēram, tīmekļa lietotni priekš klientiem preču kataloga apskatei un preču pasūtīšanai.

1.2.3. Lietotāju raksturiezīmes

WMS lietotāji ir noliktavas menedžeri. Lietotājiem jāmaks darboties ar datoru un standarta lietotnēm MS Windows vidē.

1.2.4. Vispārēji ierobežojumi

WMS izstrādē jāievēro šādi ierobežojumi:

- Programmatūrai jāfunkcionē operētājsistēmās MS Windows 2000, XP
- Programmatūra jārealizē programmēšanas valodā Visual Basic .NET un tai jāfunkcionē platformā .NET Framework 1.1. Datu bāzei jādarbojas MS SQL Server 2000 un 2005 datu bāzu serveros.
- Lietotāju pieslēgšanās datu bāzei var notikt gan ar datu bāzes lietotājvārdu un paroli, gan izmantojot Windows NT Integrated Security.

Sistēmas lietošanas ierobežojumi:

- WMS jābūt pastāvīgai pieejai datu bāzes serverim (MS SQL Server 2000 vai 2005). Serverim jāatrodas lokālajā tīklā, iespējama arī pieslēgšanās caur VPN.
- Lietotāja datorā jābūt uzstādītai operētājsistēmai MS Windows 2000 vai XP un platformai .NET Framework 1.1., kā arī pašai WMS lietotnei.
- Lietotāja datora ekrāna izšķirtspējai jābūt iestatītai 600x800 pikseļu vai lielākai.
- Aparatūrai, kas darbinās datu bāzi, jābūt atbilstošai izvēlētajai datu bāzes pārvaldības sistēmai un paredzamajai sistēmas noslodzei. Sīkāku informāciju meklēt MS SQL Server 2000 un 2005 dokumentācijās.

1.2.5. Pieņēmumi un atkarības

- Tiek pieņemts, ka produktu un pasūtījumu reģistrā tiks lietota viena naudas vienība. Konkrētas naudas vienības vai to apzīmējumi sistēmā netiek lietoti.
- Decimāldaļskaitļos kā decimālais atdalītājs tiek izmantots punkts.

1.3. FUNKCIONĀLĀS PRASĪBAS

Šajā nodaļā ir aprakstītas WMS funkcionālās prasības. Atbilstošās prasības lietotāja saskarnei un paziņojumu numurus skat. nodaļā *1.4.1. Lietotāja saskarne*. Paziņojumu nr. 1, 4 un 5 lietošana funkcionālo prasību aprakstos nav minēta, jo tie ir kopīgi daudzām lietotnes funkcijām un ir aprakstīti nodaļā *1.4.1. Lietotāja saskarne*. Tekstlodziņu pieļaujamo ievades datu tipu apzīmējumu skaidrojumus skat. nodaļā *1.1.3. Definīcijas, akronīmi un saīsinājumi*. Atbilstību datu tipiem ievades formās funkcijas pārbauda secīgi no augšas uz leju; ja kādā laukā dati neatbilst datu tipam, pārbaudi pārtrauc.

Sistēmā lietotie objektu statusi un tipi:

- Produkti
 - Statusi – aktīvs, anulēts
- Noliktavas partneri
 - Statusi – aktīvs, anulēts
 - Tipi – klients, piegādātājs
- Pasūtījumi
 - Statusi – aktīvs, anulēts, pabeigts
 - Tipi – izejošs, ienākošs

1.3.1. Darba uzsākšana

Ievads. Funkcija nodrošina pieslēgšanos datu bāzei un aizsardzību pret nesankcionētu piekļuvi, parādot pieteikšanās formu un pieprasot no lietotāja datu bāzes servera un WMS datu bāzes nosaukumus, kā arī pieteikšanās datus – datu bāzes lietotāja vārdu un paroli vai Windows NT Integrated Security lietošanas apstiprinājumu. Funkcija tiek izsaukta katru reizi, kad tiek palaista WMS lietotne, un pēc komandpogas „Logout” nospiešanas galvenajā programmas logā.

Ievade.

1.1. tabula

Pieteikšanās formas ievades lauki

Lauka nosaukums	Datu tips	Obligāts aizpildījums	Apraksts
Server Name	Text	Jā	Datu bāzes servera nosaukums
User	Text	Nē	Datu bāzes lietotāja vārds

Password	Text	Nē	Datu bāzes lietotāja parole
Database	Text	Jā	WMS datu bāzes nosaukums

Citi formas kontroles elementi:

- Izvēles rūtiņa „Use Windows NT integrated security”. Ja tā ir atzīmēta, autentifikācijai tiks izmantots Windows NT integrated security; „User” un „Password” lauki tiek atspējoti. Pretējā gadījumā „User” un „Password” lauki tiek iespējoti, un tiem ir obligāts aizpildījums.
- Komandpoga „Test” – pārbauda, vai ar ievadītajiem datiem ir iespējams pieslēgties datu bāzei.
- Komandpoga „Login” – pieteikšanās un pieslēgšanās datu bāzei.
- Komandpoga „Exit” – programmas aizvēršana.

Apstrāde. Komandpogu „Test” un „Login” nospiešana:

- Pārbauda, vai ievadītas visas obligātās vērtības. Ja nav, parāda kļūdas paziņojumu nr. 27 – 30 un beidz funkcijas izpildi.
- Mēģina pieslēgties datu bāzei ar lietotāja ievadītajiem datiem.
- „Test” – veiksmīgas pieslēgšanās gadījumā parāda paziņojumu nr. 31; kļūdas gadījumā nr. 24 – 26.
- „Login” – veiksmīgas pieslēgšanās gadījumā parāda WMS galveno logu; kļūdas gadījumā parāda paziņojumu nr. 24 – 26.

Komandpogas „Exit” nospiešana: programma beidz darbu.

Izvade. Komandpogas „Test” nospiešanas gadījumā parāda paziņojumu ar rezultātu. Pēc komandpogas „Login” nospiešanas veiksmīgas pieslēgšanās gadījumā parāda WMS galveno logu; kļūdas gadījumā parāda kļūdas paziņojumu.

1.3.2. Darba beigšana

Ievads. Funkcijas mērķis ir lietotnes darba beigšana. Pēc lietotāja izvēles programma vai nu beidz darbu, vai arī parāda pieteikšanās formu. Funkcija tiek izsaukta pēc komandpogu „Exit” un „Logout” nospiešanas galvenajā programmas logā.

Ievade. Komandpogas WMS galvenā loga galvenajā izvēlnē:

- Komandpoga „Logout” – atteikšanās.

- Komandpoga „Exit” – programmas aizvēršana.

Apstrāde.

- Parāda atiecīgu vaicājuma paziņojumu nr. 2 – 3.
- Apstiprinošas atbildes gadījumā:
 - „Logout” – izsauc funkciju *1.3.1. Darba uzsākšana*.
 - „Exit” – aizver programmu.
- Noraidošas atbildes gadījumā tiek turpināts darbs ar galveno logu.

Izvade. Parāda vaicājuma paziņojumu. Apstiprinošas atbildes gadījumā parāda pieteikšanās formu (komandpoga „Logout”) vai aizver programmu (komandpoga „Exit”).

1.3.3. Preču kataloga apskate

Ievads. Funkcija nodrošina noliktavas preču kataloga (kategoriju un preču saraksta) un konkrēta produkta datu apskati. Preču katalogu ir iespējams filtrēt pēc produkta statusa, kā arī var meklēt preces pēc preces nosaukuma. Funkcija tiek izsaukta pēc komandpogas „Product Catalog” nospiešanas galvenajā programmas logā.

Pa produktu katalogu var pārvietoties, izvēloties kategoriju un nospiežot komandpogu „View” vai izpildot dubultklikšķi uz kategorijas nosaukuma. Priekš pārvietošanās uz augšu sarakstā ir speciāla rinda, kuras nosaukuma laukā ir „..”, bet tipa laukā – „Up”.

Ievade. Komandpogas galvenajā programmas logā:

- „View” – iezīmētās preces datu parādīšana atsevišķā formā vai iezīmētās kategorijas atvēršana. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „Edit” – iezīmētās preces vai kategorijas datu labošana. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda (izņēmums – rinda ar „..”).
- „Delete” – iezīmētās preces vai kategorijas dzēšana. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda (izņēmums – rinda ar „..”).
- „New” – jaunas preces vai kategorijas pievienošana. Iespējota vienmēr.
- „Filter” – kataloga saraksta filtrēšana pēc preces statusa, kas norādīts izkrītošajos sarakstlodziņos. Filtrētas tiek preces šobrīd atvērtajā kategorijā.
- „Search” – atlasa sarakstā visas kataloga preces, kuru nosaukumā ir teksta fragments, kas ievadīts meklēšanas tekstlodziņā.

Komandpogas noliktavas produkta apskates formā:

- „Close” – formas aizvēršana.

Apstrāde. Pēc funkcijas izsaukšanas tiek veidots preču kataloga saknes kategorijas preču un kategoriju saraksts. Pēc komandpogu „Filter” un „Search” nospiešanas veido sarakstu, kura ieraksti atbilst norādītajiem kritērijiem.

Preču kataloga saraksts:

- No datu bāzes pieprasa nepieciešamos datus preču kataloga sarakstam.
- Ja ir atrasts kaut viens ieraksts, izveido datu tabulu ar atrastajiem datiem.
- Ja neviens ieraksts nav atrasts, datu tabulas vietā tiek parādīts uzraksts „No Items Found”. Ja atvērtā kategorija nav saknes kategorija, tad šī uzraksta vietā parāda datu tabulas rindu ar „..”.

Komandpogas „View” nospiešana, ja iezīmēta kategorija:

- Veido preču kataloga sarakstu izvēlētajai kategorijai.

Komandpogas „View” nospiešana, ja iezīmēta prece:

- No datu bāzes pieprasa datus par iezīmēto preci
- Ja ieraksts atrasts:
 - Parāda preces apskates formu.
 - Komandpoga „Close” – tiek aizvērta informācijas apskates forma.
- Ja ieraksts nav atrasts, parāda kļūdas paziņojumu nr. 9.

Komandpogu „Edit”, „Delete”, „New” nospiešana:

- Ja nospieš „New”, parāda formu, kurā ar radiopogām var izvēlēties, vai veidot jaunu kategoriju, vai preci.
- Tiek izsaukta atbilstoša funkcija *1.3.4. Kategoriju pievienošana, labošana un dzēšana* vai *1.3.5. Preču pievienošana, labošana un dzēšana*.

Izvade.

Virs datu tabulas, blakus nosaukumam „Product Catalog” tiek parādīts šobrīd atvērtās kategorijas nosaukums.

Preču kataloga sarakstā viena rinda reprezentē vienu kategoriju vai produktu. Ja atvērtā kategorija nav saknes kategorija, pirmajā rindā parāda „..”. Pēc tam sarakstā parāda apakškategorijas, tad preces. Datu tabulas kolonnas:

- "Item Name" – Kategorijas vai produkta nosaukums
- "Item Type" – Tips (kategorija vai produkts)
- "Category" – Produkta kategorija
- "Product Status" – Produkta statuss

- "Arriving Date" – Produkta atvešanas datums
- "Expire Date" – Produkta derīguma termiņš
- "Measurement" – Produkta mērvienība
- "Stock" – Produkta vienību skaits noliktavā

Preces apskates formā ir šādi tikai lasīšanai paredzēti tekstlodziņi:

- "Product Title" (vairākrindu tekstlodziņš) – Produkta nosaukums
- "Product Description" (vairākrindu tekstlodziņš) – Produkta apraksts
- "Location" – Produkta atrašanās vieta
- "Model Number" – Produkta modeļa numurs
- "Batch Number" – Produkta partijas numurs
- "Manufacturer" – Produkta ražotājs
- "Country" – Valsts, kurā produkts ražots
- "Measurement" – Produkta mērvienība
- "Stock" – Produkta vienību skaits noliktavā
- "Arriving Price" – Produkta cena, ievadot noliktavā
- "Current Price" – Produkta pašreizējā cena
- "Partner Name" – Partnera nosaukums
- "Arriving Date" – Produkta atvešanas datums
- "Expire Date" – Produkta derīguma termiņš
- "Product Status" – Produkta statuss

1.3.4. Kategoriju pievienošana, labošana un dzēšana

Ievads. Funkcija nodrošina jaunu preču kataloga kategoriju pievienošanu, jau esošo kategoriju labošanu un dzēšanu. Funkcija tiek izsaukta no funkcijas 1.3.3. *Preču kataloga apskate* pēc komandpogu „Edit”, „Delete” vai „New” nospiešanas, ja sarakstā bijusi iezīmēta kategorija.

Ievade.

1.2. tabula

Preču kataloga kategorijas pievienošanas un labošanas formu ievades lauki

Lauka nosaukums	Datu tips	Obligāts aizpildījums	Apraksts
"Category Name"	uText(1-50)	Jā	Kategorijas nosaukums

Komandpogas **kategorijas pievienošanas un labošanas formās**:

- „Save” – saglabāt ievadītos datus datu bāzē.
- „Cancel” – atcelt izmaiņas un aizvērt formu.

Apstrāde. Kategorijas pievienošana un labošana:

- Labošanas gadījumā – no datu bāzes pieprasa datus par galvenā loga datu tabulā iezīmēto kategoriju. Ja ieraksts nav atrasts, parāda kļūdas paziņojumu nr. 7 un darbību neturpina.
- Parāda **kategorijas** pievienošanas vai labošanas formu.
- Ja nospiesta komandpoga „Save”:
 - Pārbauda ievadīto datu pareizību. Kļūdas gadījumā parāda atbilstošu paziņojumu nr. 17.
 - Saglabā ievadītos datus datu bāzē. Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Kategorijas dzēšana:

- Ja iezīmētajā kategorijā ir kaut viens produkts, kategorija netiek dzēsta un tiek parādīts paziņojums nr. 6.
- Pretējā gadījumā no datu bāzes dzēš galvenā loga datu tabulā iezīmēto kategoriju.
- Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Izvade. Pēc veiksmīgas kategorijas saglabāšanas vai dzēšanas atgriežas galvenajā logā un izsauc funkciju *1.3.3. Preču kataloga apskate*, atverot to kataloga kategoriju, no kuras tika izsaukts tikko aizvērtais dialoga logs. Ja darbība atcelta, vienkārši atgriežas galvenajā logā.

1.3.5. Preču pievienošana, labošana un dzēšana

Ievads. Funkcija nodrošina jaunu preču pievienošanu preču katalogam, jau esošo preču labošanu un dzēšanu. Funkcija tiek izsaukta no funkcijas *1.3.3. Preču kataloga apskate* pēc komandpogu „Edit”, „Delete” vai „New” nospiešanas, ja sarakstā bijusi iezīmēta prece.

Ievade.

1.3. tabula

Preču pievienošanas un labošanas formu ievades lauki

Lauka nosaukums	Datu tips	Obligāts aizpildījums	Apraksts
"Product Title"	uText(1-200)	Jā	Produkta nosaukums (vairākrindu tekstlodziņš)
"Product Description"	uText(1-200)	Jā	Produkta apraksts (vairākrindu tekstlodziņš)
"Location"	uText(0-100)	Nē	Produkta atrašanās vieta
"Model Number"	Text(0-50)	Nē	Produkta modeļa numurs
"Batch Number"	Text(0-50)	Nē	Produkta partijas numurs
"Manufacturer"	uText(1-100)	Jā	Produkta ražotājs
"Country"	uText(0-50)	Nē	Valsts, kurā produkts ražots
"Measurement"	uText(1-25)	Jā	Produkta mērvienība
"Stock"	Integer(0)	Jā	Produkta vienību skaits noliktavā (vienība – product_measurement)
"Arriving Price"	Float(0)	Jā	Produkta cena, ievadot noliktavā
"Current Price"	Float(0)	Jā	Produkta pašreizējā cena
„Partner Name”	Text	Jā	Produkta partnera vārds (tekstlodziņš tikai lasīšanai)

Preces statusu izvēlas ar radiopogām. Pievienošanas formā statusa radiopogu grupa ir atspējota un noklusētais statuss ir „Aktīvs”.

Preces atvešanas datumu un derīguma termiņu izvēlas ar datuma izvēles kalendāru. Derīguma termiņa kalendāru var iespējot un atspējot ar izvēles rūtiņu, jo šis preces atribūts nav obligāts.

Komandpogas **partnera pievienošanas un labošanas formās**:

- „Save” – saglabāt ievadītos datus datu bāzē.
- „Cancel” – atcelt izmaiņas un aizvērt formu.
- „Change Partner” – pievieno vai maina preces partneri. Izsauc funkciju *1.3.6. Partnera piesaistīšana produktam vai pasūtījumam.*

Apstrāde. Preces pievienošana un labošana:

- Labošanas gadījumā – no datu bāzes pieprasa datus par galvenā loga datu tabulā iezīmēto precī. Ja ieraksts nav atrasts, parāda kļūdas paziņojumu nr. 9 un darbību neturpina.
- Parāda **preces** pievienošanas vai labošanas formu.

- Ja nospiesta komandpoga „Save”:
 - Pārbauda ievadīto datu pareizību. Kļūdas gadījumā parāda atbilstošu paziņojumu nr. 17 – 20.
 - Pārbauda, vai ir izvēlēts partneris. Ja nav, parāda paziņojumu nr. 21.
 - Saglabā ievadītos datus datu bāzē. Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Preces dzēšana:

- Ja iezīmētā prece ir piesaistīta kaut vienam pasūtījumam, prece netiek dzēsta un tiek parādīts paziņojums nr. 8.
- Pretējā gadījumā no datu bāzes dzēš galvenā loga datu tabulā iezīmēto preci.
- Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Izvade. Pēc veiksmīgas preces saglabāšanas vai dzēšanas atgriežas galvenajā logā un izsauc funkciju *1.3.3. Preču kataloga apskate*, atverot to kataloga kategoriju, no kuras tika izsaukts tikko aizvērtais dialoga logs. Ja darbība atcelta, vienkārši atgriežas galvenajā logā.

1.3.6. Partnera piesaistīšana produktam vai pasūtījumam

Ievads. Funkcija veic partnera piesaisti precei vai pasūtījumam. Funkcija tiek izsaukta no preču un pasūtījumu pievienošanas un labošanas formām. Lietotājam tiek parādīts partneru saraksts, no kura izvēlēties vajadzīgo.

Ievade. Komandpogas pie partneru saraksta:

- „Accept” – iezīmētā partnera piesaistīšana precei vai pasūtījumam. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „Cancel” – atgriešanās preces vai pasūtījuma pievienošanas vai labošanas formā.

Apstrāde.

- Veido partneru sarakstu.
- Ja ir atrasts kaut viens ieraksts, izveido datu tabulu ar atrastajiem datiem.
- Ja neviens ieraksts nav atrasts, datu tabulas vietā tiek parādīts uzraksts „No Items Found”.

Izvade. Atgriešanās preces vai pasūtījuma pievienošanas vai labošanas formā. Ja nospiesta komandpoga „Accept”, laukā „Partner Name” parādās izvēlētais partnera nosaukums un forma

saņem tā datu bāzes ID numuru. Ja nospiesta komandpoga „Cancel”, minētais lauks nemainās.

1.3.7. Partneru datu apskate

Ievads. Funkcija nodrošina noliktavas partneru saraksta un konkrēta partnera datu apskati. Partneru sarakstu ir iespējams filtrēt pēc partnera statusa un tipa, kā arī var meklēt partnerus pēc partnera nosaukuma. Funkcija tiek izsaukta pēc komandpogas „Partners” nospiešanas galvenajā programmas logā.

Ievade. Komandpogas galvenajā programmas logā:

- „View” – iezīmētā noliktavas partnera datu parādīšana atsevišķā formā. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „Edit” – iezīmētā noliktavas partnera datu labošana. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „Delete” – iezīmētā noliktavas partnera dzēšana. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „New” – jauna noliktavas partnera pievienošana. Iespējota vienmēr.
- „Filter” – partneru saraksta filtrēšana pēc partnera statusa un tipa, kas norādīti izkrītošajos sarakstlodziņos.
- „Search” – atlasa sarakstā visus partnerus, kuru nosaukumā ir teksta fragments, kas ievadīts meklēšanas tekstlodziņā.

Komandpogas noliktavas partnera apskates formā:

- „Close” – formas aizvēršana.

Apstrāde. Pēc funkcijas izsaukšanas tiek veidots noliktavas partneru saraksts. Pēc komandpogu „Filter” un „Search” nospiešanas veido sarakstu, kura ieraksti atbilst norādītajiem kritērijiem.

Noliktavas partneru saraksts:

- No datu bāzes pieprasa nepieciešamos datus noliktavas partneru sarakstam.
- Ja ir atrasts kaut viens ieraksts, izveido datu tabulu ar atrastajiem datiem.
- Ja neviens ieraksts nav atrasts, datu tabulas vietā tiek parādīts uzraksts „No Items Found”.

Komandpogas „View” nospiešana:

- No datu bāzes pieprasa datus par iezīmēto noliktavas partneri

- Ja ieraksts atrasts:
 - Parāda noliktavas partnera apskates formu.
 - Komandpoga „Close” – tiek aizvērtā informācijas apskates forma.
- Ja ieraksts nav atrasts, parāda kļūdas paziņojumu nr. 11.

Komandpogu „Edit”, „Delete”, „New” nospiešana:

- Tiek izsaukta funkcija *1.3.8. Partneru datu pievienošana, labošana un dzēšana*.

Izvade.

Noliktavas partneru sarakstā viena rinda reprezentē vienu partneri. Datu tabulas kolonnas:

- "Partner Name" – Partnera nosaukums
- "Partner Type" – Partnera tips
- "Partner Status" – Partnera statuss

Noliktavas partnera apskates formā ir šādi tikai lasīšanai paredzēti tekstlodziņi:

- "Partner Name" – Partnera nosaukums
- "Domicile" (vairākrindu tekstlodziņš) – Partnera juridiskā adrese
- "Office Address" (vairākrindu tekstlodziņš) – Partnera biroja adrese
- "Delivery Address" (vairākrindu tekstlodziņš) – Partnera piegādes adrese
- "Contact Person" – Partnera kontaktpersona
- "Phone" – Partnera telefons
- "Fax" – Partnera fakss
- "E-mail" – Partnera e-pasts
- "Registration Code" – Partnera uzņēmuma reģistrācijas numurs
- "VAT Code" – Partnera PVN reģistrācijas numurs
- "Bank" – Partnera bankas nosaukums
- "Bank Code" – Partnera bankas kods
- "Bank Account" – Partnera bankas norēķinu konts
- "Partner Type" – Partnera tips
- "Partner Status" – Partnera statuss

1.3.8. Partneru datu pievienošana, labošana un dzēšana

Ievads. Funkcija nodrošina jaunu noliktavas partneru pievienošanu, jau esošo partneru labošanu un dzēšanu. Funkcija tiek izsaukta no funkcijas *1.3.7. Partneru datu apskate* pēc komandpogu „Edit”, „Delete” vai „New” nospiešanas.

Ievade.

1.4. tabula

Noliktavas partnera pievienošanas un labošanas formu ievades lauki

Lauka nosaukums	Datu tips	Obligāts aizpildījums	Apraksts
"Partner Name"	uText(1-100)	Jā	Partnera nosaukums
"Domicile"	uText(1-200)	Jā	Partnera juridiskā adrese (vairākrindu tekstlodziņš)
"Office Address"	uText(0-200)	Nē	Partnera biroja adrese (vairākrindu tekstlodziņš)
"Delivery Address"	uText(0-200)	Nē	Partnera piegādes adrese (vairākrindu tekstlodziņš)
"Contact Person"	uText(1-50)	Jā	Partnera biroja adrese kontaktpersona
"Phone"	Text(0-25)	Nē	Partnera telefons
"Fax"	Text(0-25)	Nē	Partnera fakss
"E-mail"	Text(0-50)	Nē	Partnera e-pasts
"Registration Code"	Text(0-25)	Nē	Partnera uzņēmuma reģistrācijas numurs
"VAT Code"	Text(0-25)	Nē	Partnera PVN reģistrācijas numurs
"Bank"	uText(0-50)	Nē	Partnera bankas nosaukums
"Bank Code"	Text(0-50)	Nē	Partnera bankas kods
"Bank Account"	Text(0-50)	Nē	Partnera bankas norēķinu konts

Partnera statusu un tipu izvēlas ar radiopogām. Pievienošanas formā statusa radiopogu grupa ir atspējota un noklusētais statuss ir „Aktīvs”.

Komandpogas **partnera pievienošanas un labošanas formās**:

- „Save” – saglabāt ievadītos datus datu bāzē.
- „Cancel” – atcelt izmaiņas un aizvērt formu.

Apstrāde. Noliktavas **partnera** pievienošana un labošana:

- Labošanas gadījumā – no datu bāzes pieprasa datus par galvenā loga datu tabulā iezīmēto noliktavas **partneri**. Ja ieraksts nav atrasts, parāda kļūdas paziņojumu nr. 11 un darbību neturpina.
- Parāda **partnera** pievienošanas vai labošanas formu.
- Ja nospiesta komandpoga „Save”:

- Pārbauda ievadīto datu pareizību. Kļūdas gadījumā parāda atbilstošu paziņojumu nr. 17 – 18.
- Saglabā ievadītos datus datu bāzē. Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Noliktavas **partnera** dzēšana:

- Ja iezīmētajam partnerim ir piesaistīts kaut viens produkts vai pasūtījums, partneris netiek dzēsts un tiek parādīts paziņojums nr. 10.
- Pretējā gadījumā no datu bāzes dzēš galvenā loga datu tabulā iezīmēto noliktavas partneri.
- Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Izvade. Pēc veiksmīgas noliktavas partnera saglabāšanas vai dzēšanas atgriežas galvenajā logā un izsauc funkciju *1.3.7. Partneru datu apskate*. Ja darbība atcelta, vienkārši atgriežas galvenajā logā.

1.3.9. Pasūtījumu datu apskate

Ievads. Funkcija nodrošina pasūtījumu saraksta un konkrēta pasūtījuma datu apskati. Pasūtījumu sarakstu ir iespējams filtrēt pēc pasūtījuma statusa un tipa, kā arī var meklēt pasūtījumus pēc pasūtījuma partnera nosaukuma. Funkcija tiek izsaukta pēc komandpogas „Orders” nospiešanas galvenajā programmas logā.

Ievade. Komandpogas galvenajā programmas logā:

- „View” – iezīmētā pasūtījuma datu parādīšana atsevišķā formā. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „Edit” – iezīmētā pasūtījuma datu labošana. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „Delete” – iezīmētā pasūtījuma dzēšana. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „New” – jauna pasūtījuma pievienošana. Iespējota vienmēr.
- „Filter” – pasūtījumu saraksta filtrēšana pēc pasūtījuma statusa un tipa, kas norādīti izkrītošajos sarakstlodziņos.
- „Search” – atlasa sarakstā visus pasūtījumus, kuriem piesaistītā partnera nosaukumā ir teksta fragments, kas ievadīts meklēšanas tekstlodziņā.

Komandpogas pasūtījuma apskates formā:

- „Close” – formas aizvēšana.

Apstrāde. Pēc funkcijas izsaukšanas tiek veidots pasūtītāju saraksts. Pēc komandpogu „Filter” un „Search” nospiešanas veido sarakstu, kura ieraksti atbilst norādītajiem kritērijiem.

Pasūtītāju saraksts:

- No datu bāzes pieprasa nepieciešamos datus pasūtītāju sarakstam.
- Ja ir atrasts kaut viens ieraksts, izveido datu tabulu ar atrastajiem datiem.
- Ja neviens ieraksts nav atrasts, datu tabulas vietā tiek parādīts uzraksts „No Items Found”.

Komandpogas „View” nospiešana:

- No datu bāzes pieprasa datus par iezīmēto pasūtījumu
- Ja ieraksts atrasts:
 - Parāda pasūtītāja apskates formu.
 - Komandpoga „Close” – tiek aizvēta informācijas apskates forma.
- Ja ieraksts nav atrasts, parāda kļūdas paziņojumu nr. 13.

Komandpogu „Edit”, „Delete”, „New” nospiešana:

- Tiek izsaukta funkcija *1.3.10. Pasūtījumu datu pievienošana, labošana un dzēšana*.

Izvade.

Pasūtījumu sarakstā viena rinda reprezentē vienu pasūtījumu. Datu tabulas kolonnas:

- "Order ID" – Pasūtījuma unikāls identifikators
- "Order Date" – Pasūtījuma datums
- "Partner Name" – Pasūtījuma partnera nosaukums
- "Order Type" – Pasūtījuma tips
- "Order Status" – Pasūtījuma statuss
- "Total Price" – Pasūtījuma kopējā summa (ar nodokļiem)

Pasūtījuma apskates formā ir šādi tikai lasīšanai paredzēti tekstlodziņi:

- "Order ID" – Pasūtījuma unikāls identifikators
- "Order Date" – Pasūtījuma datums
- "Partner Name" – Pasūtījuma partnera nosaukums
- "Order Invoice" – Pasūtījuma pavadzīmes numurs
- "Transport Details" (vairākrindu tekstlodziņš) – Pasūtījuma transporta informācija
- "Package" – Pasūtījuma iepakojums

- "Order Item Count" – Pasūtījumam piesaistīto produktu skaits
- "Total Price (Net)" – Pasūtījuma kopējā summa (bez nodokļiem)
- "Excise Tax" – Pasūtījuma akcīzes nodoklis
- "Nature Resource Tax" – Pasūtījuma dabas resursu nodoklis
- "Value Added Tax" – Pasūtījuma pievienotās vērtības nodoklis
- "Discount" – Pasūtījuma atlaide
- "Total Price (Gross)" – Pasūtījuma kopējā summa (ar nodokļiem)
- "Order Type" – Pasūtījuma tips
- "Order Status" – Pasūtījuma statuss

1.3.10. Pasūtījumu datu pievienošana, labošana un dzēšana

Ievads. Funkcija nodrošina jaunu pasūtījumu pievienošanu, jau esošo pasūtījumu labošanu un dzēšanu. Funkcija tiek izsaukta no funkcijas 1.3.3. *Preču kataloga apskate* pēc komandpogu „Edit”, „Delete” vai „New” nospiešanas.

Ievade.

1.5. tabula

Pasūtījumu pievienošanas un labošanas formu teksta lauki

Lauka nosaukums	Datu tips	Obligāts aizpildījums	Apraksts
"Order ID"	Text	Jā	Pasūtījuma unikāls identifikators (Tekstlodziņš tikai lasīšanai).
"Partner Name"	Text	Jā	Pasūtītāja partnera nosaukums (Tekstlodziņš tikai lasīšanai).
"Order Invoice"	Text(0-50)	Nē	Pasūtījuma pavadzīmes numurs
"Transport Details"	uText(0-200)	Nē	Pasūtījuma transporta informācija (vairākrindu tekstlodziņš)
"Package"	uText(0-100)	Nē	Pasūtījuma iepakojums
"Order Item Count"	Integer(0)	Jā	Pasūtījumam piesaistīto produktu skaits (Tekstlodziņš tikai lasīšanai).
"Total Price (Net)"	Float(0)	Jā	Pasūtījuma kopējā summa (bez nodokļiem) (Tekstlodziņš tikai lasīšanai).
"Excise Tax"	Float(0)	Jā	Pasūtījuma akcīzes nodoklis (noklusētā vērtība – 0)

"Nature Resource Tax"	Float(0)	Jā	Pasūtījuma dabas resursu nodoklis (noklusētā vērtība – 0)
"Value Added Tax"	Float(0)	Jā	Pasūtījuma pievienotās vērtības nodoklis (noklusētā vērtība – 0)
"Discount"	Float(0)	Jā	Pasūtījuma atlaide (noklusētā vērtība – 0).
"Total Price (Gross)"	Float(0)	Jā	Pasūtījuma kopējā summa (pasūtījuma produktu cenas + nodokļi); noklusētā vērtība – 0 (Tekstlodziņš tikai lasīšanai).

Pasūtījumu statusu un tipu izvēlas ar radiopogām. Pievienošanas formā statusa radiopogu grupa ir atspējota un noklusētais statuss ir „Aktīvs”.

Pasūtījuma datumu izvēlas ar datuma izvēles kalendāru. To var iespējot un atspējot ar izvēles rūtiņu, jo šis pasūtījuma atribūts nav obligāts.

Komandpogas **pasūtījumu pievienošanas un labošanas formās**:

- „Save” – saglabāt ievadītos datus datu bāzē.
- „Cancel” – atcelt izmaiņas un aizvērt formu.
- „Change Partner” – pievieno vai maina pasūtījuma partneri. Izsauc funkciju *1.3.6. Partnera piesaistīšana produktam vai pasūtījumam*.
- „Add Order Item” – pievieno pasūtījumam produktus. Izsauc funkciju *1.3.11. Preču piesaistīšana pasūtījumam*. Iespējota tikai labošanas formā.
- „View Order Items” – parāda pasūtījuma produktu sarakstu. Šajā sarakstā piesaistītās preces iespējams apskatīt, labot un dzēst. Komandpoga iespējota tikai labošanas formā.
- „Calculate Total Price” – aprēķina pasūtījuma kopējo summu ar nodokļiem un rezultātu parāda laukā "Total Price (Gross)".

Apstrāde. Pasūtījuma pievienošana un labošana:

- Labošanas gadījumā – no datu bāzes pieprasa datus par galvenā loga datu tabulā iezīmēto pasūtījumu. Ja ieraksts nav atrasts, parāda kļūdas paziņojumu nr. 13 un darbību neturpina.
- Parāda **pasūtījuma** pievienošanas vai labošanas formu.
- Ja nospiesta komandpoga „Save”:
 - Pārbauda ievadīto datu pareizību. Kļūdas gadījumā parāda atbilstošu paziņojumu nr. 17 – 20.
 - Pārbauda, vai ir izvēlēts partneris. Ja nav, parāda paziņojumu nr. 22.
 - Aprēķina pasūtījuma kopējo summu ar nodokļiem.

- Saglabā ievadītos datus datu bāzē. Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Pasūtījuma dzēšana:

- Ja iezīmētajam pasūtījumam ir piesaistīts kaut viens produkts, pasūtījums netiek dzēsts un tiek parādīts paziņojums nr. 12.
- Pretējā gadījumā no datu bāzes dzēš galvenā loga datu tabulā iezīmēto preci.
- Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Izvade. Pēc veiksmīgas pasūtījuma saglabāšanas vai dzēšanas atgriežas galvenajā logā un izsauc funkciju *1.3.9. Pasūtījumu datu apskate*. Ja darbība atcelta, vienkārši atgriežas galvenajā logā.

1.3.11. Preču piesaistīšana pasūtījumam

Ievads. Funkcija veic preces piesaisti pasūtījumam. Funkcija tiek izsaukta no pasūtījumu un labošanas formas. Lietotājam tiek parādīta preču piesaistīšanas forma.

Ievade.

1.6. tabula

Preču piesaistīšanas formas ievades lauki

Lauka nosaukums	Datu tips	Obligāts aizpildījums	Apraksts
"Product Quantity"	Integer(0)	Jā	Pasūtījumam piesaistītās preces vienību skaits
"Unit Price"	Float(0)	Jā	Pasūtījumam piesaistītās preces vienas vienības cena

Preci, ko pievienot pasūtījumam, izvēlas no izkrītošā sarakstlodziņā.

Komandpogas **pasūtījumu pievienošanas un labošanas formās**:

- „Save” – saglabāt ievadītos datus datu bāzē.
- „Cancel” – atcelt izmaiņas un aizvērt formu.

Apstrāde.

- Parāda **preces piesaistīšanas formu**.
- Izkrītošajā sarakstlodziņā no datu bāzes ielasa preču nosaukumus.
- Ja nospiesta komandpoga „Save”:

- Pārbauda ievadīto datu pareizību. Kļūdas gadījumā parāda atbilstošu paziņojumu nr. 19 – 20.
- Saglabā ievadītos datus datu bāzē. Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Izvade. Atgriešanās pasūtījuma labošanas formā. Ja piesaistītā prece saglabāta, tiek pārrēķinātas summas lauku vērtības. Ja nospiesta komandpoga „Cancel”, pasūtījumu formā nekas nemainās.

1.3.12. Noliktavas rekvizītu apskate

Ievads. Funkcija nodrošina noliktavas rekvizītu saraksta un atsevišķa rekvizīta datu apskati. Funkcija tiek izsaukta pēc komandpogas „Warehouse Details” nospiešanas galvenajā programmas logā.

Ievade. Komandpogas galvenajā programmas logā:

- „View” – iezīmētā noliktavas rekvizīta datu parādīšana atsevišķā formā. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „Edit” – iezīmētā noliktavas rekvizīta datu labošana. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „Delete” – iezīmētā noliktavas rekvizīta dzēšana. Iespējota tikai tad, ja ir iezīmēta kāda datu tabulas rinda.
- „New” – jauna noliktavas rekvizīta pievienošana. Iespējota vienmēr.

Komandpogas noliktavas rekvizīta apskates formā:

- „Close” – formas aizvēršana.

Apstrāde. Pēc funkcijas izsaukšanas tiek veidots noliktavas rekvizītu saraksts.

Noliktavas rekvizītu saraksts:

- No datu bāzes pieprasa nepieciešamos datus noliktavas rekvizītu sarakstam.
- Ja ir atrasts kaut viens ieraksts, izveido datu tabulu ar atrastajiem datiem.
- Ja neviens ieraksts nav atrasts, datu tabulas vietā tiek parādīts uzraksts „No Items Found”.

Komandpogas „View” nospiešana:

- No datu bāzes pieprasa datus par iezīmēto noliktavas rekvizītu
- Ja ieraksts atrasts:

- Parāda noliktavas rekvizīta apskates formu.
- Komandpoga „Close” – tiek aizvērta informācijas apskates forma.
- Ja ieraksts nav atrasts, parāda kļūdas paziņojumu nr. 16.

Komandpogu „Edit”, „Delete”, „New” nospiešana:

- Tiek izsaukta funkcija *1.3.13. Noliktavas rekvizītu pievienošana, labošana un dzēšana*.

Izvade.

Noliktavas rekvizītu sarakstā viena rinda reprezentē vienu rekvizītu. Datu tabulas kolonnas:

- "Detail Title" – noliktavas rekvizīta nosaukums.
- "Detail Value" – noliktavas rekvizīta vērtība.

Noliktavas rekvizīta apskates formā ir šādi tikai lasīšanai paredzēti tekstlodziņi:

- "Detail Title" – noliktavas rekvizīta nosaukums.
- "Detail Value" (vairākrindu tekstlodziņš) – noliktavas rekvizīta vērtība.

1.3.13. Noliktavas rekvizītu pievienošana, labošana un dzēšana

Ievads. Funkcija nodrošina jaunu noliktavas rekvizītu pievienošanu, jau esošo rekvizītu labošanu un dzēšanu. Funkcija tiek izsaukta no funkcijas *1.3.12. Noliktavas rekvizītu apskate* pēc komandpogu „Edit”, „Delete” vai „New” nospiešanas.

Ievade.

1.7. tabula

Noliktavas rekvizīta pievienošanas un labošanas formu ievades lauki

Lauka nosaukums	Datu tips	Obligāts aizpildījums	Apraksts
Detail Title	uText(1-50)	Jā	Noliktavas rekvizīta nosaukums
Detail Value	uText(1-200)	Jā	Noliktavas rekvizīta vērtība (vairākrindu tekstlodziņš)

Komandpogas **rekvizīta pievienošanas un labošanas formās:**

- „Save” – saglabāt ievadītos datus datu bāzē.
- „Cancel” – atcelt izmaiņas un aizvērt formu.

Apstrāde. Noliktavas rekvizīta pievienošana un labošana:

- Labošanas gadījumā – no datu bāzes pieprasa datus par galvenā loga datu tabulā iezīmēto noliktavas rekvizītu. Ja ieraksts nav atrasts, parāda kļūdas paziņojumu nr. 16 un darbību neturpina.
- Parāda rekvizīta pievienošanas vai labošanas formu.
- Ja nospiesta komandpoga „Save”:
 - Pārbauda ievadīto datu pareizību. Kļūdas gadījumā parāda paziņojumu nr. 17.
 - Saglabā ievadītos datus datu bāzē. Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Noliktavas rekvizīta dzēšana:

- No datu bāzes dzēš galvenā loga datu tabulā iezīmēto noliktavas rekvizītu.
- Kļūdas gadījumā parāda atbilstošu datu bāzes kļūdas paziņojumu.

Izvade. Pēc veiksmīgas noliktavas rekvizīta saglabāšanas vai dzēšanas atgriežas galvenajā logā un izsauc funkciju *1.3.12. Noliktavas rekvizītu apskate*. Ja darbība atcelta, vienkārši atgriežas galvenajā logā.

1.3.14. Informācijas par sistēmu apskate

Ievads. Funkcija parāda formu ar informāciju par programmatūras izstrādātāju un lietotnes versiju. Funkcija tiek izsaukta pēc komandpogas „About This Software” nospiešanas galvenajā programmas logā.

Ievade.

- Komandpoga „OK” informācijas apskates formā – formas aizvēršana.

Apstrāde.

- Parāda formu.
- Komandpoga „OK” – tiek aizvērta informācijas apskates forma.

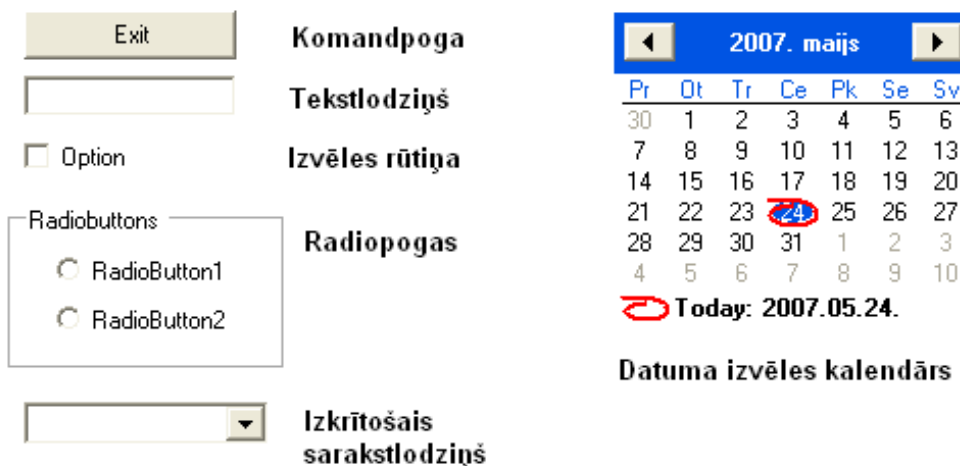
Izvade. Formā ar informāciju par programmatūru parāda:

- Programmatūras nosaukumu un versiju.
- Programmatūras izstrādātāju.

1.4. Ārējā saskarne

1.4.1. Lietotāja saskarne

WMS lietotāja saskarne tiek veidota pēc vispārpieņemtiem MS Windows lietotņu standartiem. Saskaņā ir angļu valodā, lai programmatūra būtu pieejama plašākam lietotāju lokam. 1.1. attēlā ir parādīti izmantotie kontroles elementi. Izņemot datuma izvēles kalendāru, visi pārējie kontroles elementi ir bieži sastopami MS Windows lietotņu lietotāju saskarnēs.

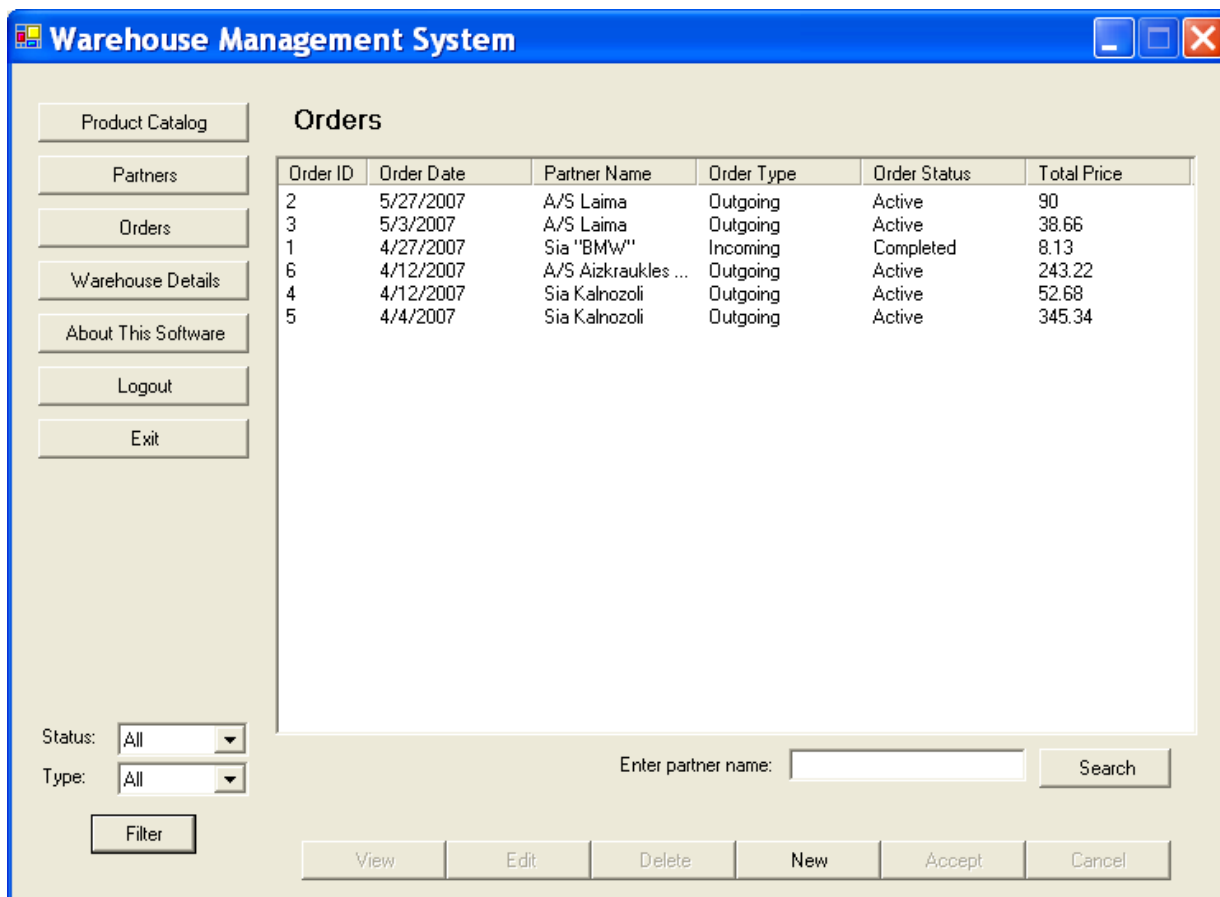


1.1. att. Lietotāja saskarnē izmantotie formu kontroles elementi

- Komandpoga izpilda darbību vai sistēmas funkciju, kas norādīta uz tās.
- Tekstlodziņā lietotājs var ievadīt datus, kurus sistēma pēc attiecīga pieprasījuma apstrādā. Tekstlodziņi var būt vienrindas vai vairākrindu. Vairākrindu tekstlodziņos teksts tiek attēlots vairākās rindās, taču tajos netiek atļauta pārvešana jaunā rindā ar taustiņu ENTER. Tekstlodziņš var būt arī paredzēts tikai lasīšanai, neļaujot lietotājam tajā neko ievadīt. Paroļu tekstlodziņos ievadīto simbolu vietā drošības nolūkos tiek rādītas zvaigznītes.
- Izvēles rūtiņa ir formas kontroles elements, ar kura palīdzību tiek aktivizēta vai atslēgta kāda papildiespēja.
- Ar radiopogām var izvēlēties tieši vienu izvēli no vairākām piedāvātajām.
- Izkrītošais sarakstlodziņš ir saraksts, kura apskatīšanai ir jānospiež uz leju vērsta bultiņa. No tā var izvēlēties vienu no piedāvātajiem vienumiem. Lietotājs no tastatūras šajā kontroles elementā patvaļīgu tekstu ievadīt nevar.
- Ar datuma izvēles kalendāru lietotājs uzstāda datumu. Ar bultiņām iespējams pārvietoties pa mēnešiem. Gadu var mainīt, noklikšķinot uz tā. Pašreizējās dienas

datums tiek atzīmēts ar aplīti. Atšķirībā no datuma ievadīšanas tekstlodziņos, nav jāveic datuma formāta pareizības pārbaude, un lietotājam nav jāiegaumē, kādu datuma pierakstu sistēma akceptē.

1.2. un 1.3. attēlā ir redzama lietotnes galvenais logs, savukārt 1.4. attēlā – datu ievades formas paraugs. Visiem logiem ir loga nosaukuma josla, kuras labajā malā ir standarta MS Windows logu vadības pogas. Visiem lietotnes logiem ir fiksēts izmērs, lietotājs to nevar mainīt; poga loga izplešanai pa visu ekrānu ir atspējota.

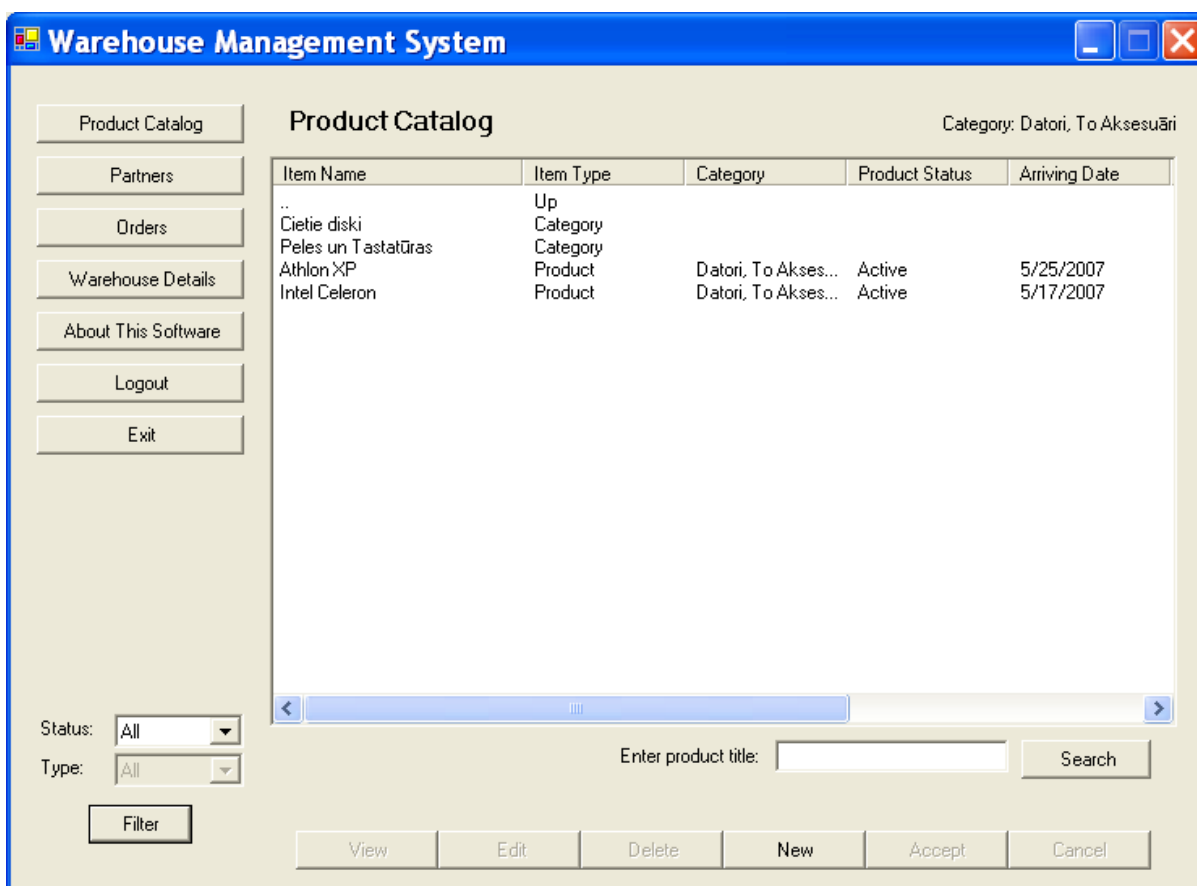


1.2. att. WMS galvenais logs: skats „Pasūtījumi”

Galvenā loga izkārtojums:

- Galvenā loga kreisajā pusē atrodas galvenā izvēlne, kas realizēta ar komandpogām. Ar tām var izvēlēties kādu no sistēmas funkcijām.
- Pa labi no izvēlnes ir datu tabula, kurā tiek parādīti datu pieprasījuma rezultāti.
- Virs datu tabulas ir virsraksts, kas reprezentē tabulā redzamos datus un šobrīd izvēlēto sistēmas funkciju. Blakus tam var tikt attēlota papildinformācija, piemēram, pārlūkojamās kategorijas nosaukums preču katalogā.

- Ja pieprasījuma rezultātā netiek atrasts neviens ieraksts, datu tabulas vietā tiek parādīts uzraksts „No Items Found”.
- Zem datu tabulas atrodas meklēšanas tekstlodziņš ar atbilstošu komandpogu. Tam blakus ir norāde par lauku, pēc kura tiks veikta meklēšana. Tekstlodziņš ir iespējots vai atspējots atkarībā no izvēlētās sistēmas funkcijas.
- Pašā loga apakšā ir komandpogas darbībām ar datu tabulas ierakstiem. Tās tiek aktivizētas un atslēgtas atkarībā no iezīmētās tabulas rindas un izvēlētās sistēmas funkcijas. Komandpogas „Skatīt” („View”) funkciju var arī izsaukt, izpildot dubultklikšķi uz kādas no datu tabulas rindām.
- Kreisajā apakšējā stūrī ir datu filtrēšanas parametru izkrītošie sarakstlodziņi. Tie ir iespējoti vai atspējoti atkarībā no izvēlētās sistēmas funkcijas.



1.3. att. WMS galvenais logs: skats „Preču katalogs”

Datu pieprasījuma laikā visas galvenā lietotnes loga kontroles tiek atspējotas un datu tabulas virsrakstā tiek parādīts uzraksts „Retrieving Data...” Tādējādi lietotājam nav iespējas vienas funkcijas izpildes laikā izsaukt kādu citu funkciju un izraisīt sistēmas konfliktu. Šī paša iemesla dēļ no galvenā loga izsauktās datu apstrādes un apskates formas tiek parādītas kā modālie dialoglogi, vai arī to parādīšanas laikā galvenais logs tiek noslēpts.

Edit Partner

Partner Name: Sia LatRos

Domicile: Rīga, Brīvības iela 111

Office Address: Kr. Barona iela 324

Delivery Address: Kr. Valdemāra iela 3

Contact Person: Visvaldis Graudiņš

Phone: 7656324

Fax: 7234234

E-mail: asd@asd.lv

Registration Code: 40008887775

VAT Code: 40008887777

Bank Name: Hansabanka

Bank Code: 435435674375

Bank Account: HABA-43257432532543254

Partner Type: Client Supplier

Partner Status: Active Void

Save Cancel

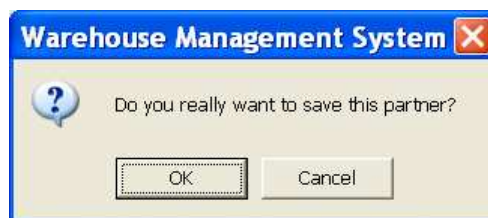
1.4. att. Datu ievades formas paraugs – „Edit Partner”

Sistēmas paziņojumi arī atbilst MS Windows standartiem. Sistēmā ir trīs veidu paziņojumi:

- Vaicājuma paziņojumi – sistēma prasa lietotājam apstiprināt kādu darbību (piemēru skatīt 1.5. attēlā).
- Informatīvie paziņojumi – informē lietotāju par veiksmīgu darbības norisi vai sniedz specifisku informāciju par pieprasītās darbības izpildes problēmām (piemēru skatīt 1.6. attēlā).
- Kļūdu paziņojumi – sniedz informāciju par lietotāja kļūdām datu ievadē, sistēmas kļūdām un izņēmuma situācijām (piemēru skatīt 1.7. attēlā).

Vaicājuma, kļūdu un informatīvajiem paziņojumiem tiek rādītas atbilstošās operētājsistēmā lietotās ikonas.

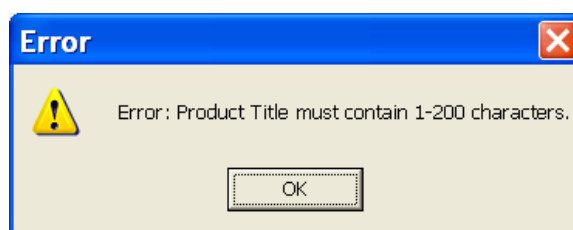
Ja WMS galvenajā logā datu pieprasījuma laikā ir radusies sistēmas kļūda, tiek parādīts attiecīgs paziņojums, un kļūdas teksts tiek parādīts arī datu tabulas vietā.



1.5. att. Vaicājuma paziņojuma piemērs



1.6. att. Informatīvā paziņojuma piemērs



1.7. att. Kļūdas paziņojuma piemērs

Fiksēto sistēmas paziņojumu saraksts ir dots 1.8. tabulā. Paziņojumu tekstā kvadrātisko iekavu vietā ir iespējamās dažādas vērtības (ierakstu ID numuri, ievadāmā teksta pieļaujamā garuma ierobežojumi u.tml.); [item] vietā var būt: *katgorija (category)*, *prece (product)*, *noliktavas partneris (partner)*, *pasūtījums (order)*, *pasūtījumam piesaistīts produkts (order item)*, *noliktavas rekvizīts (warehouse detail)*.

Paziņojumi, kas kopīgi daudzām lietotnes funkcijām:

- Pie katras datu saglabāšanas un dzēšanas operācijas sistēma lietotājam parāda vaicājuma paziņojumu, vai tiešām veikt pieprasītās izmaiņas datu bāzē (paziņojumi nr. 4 – 5).
- Pirms katras datu ievades formas aizvēršanas, ja dati nav tikuši saglabāti, parāda vaicājuma paziņojumu nr. 1.

Izņēmuma situāciju kļūdu paziņojumi ir atkarīgi no konkrētās izņēmuma situācijas.

WMS fiksētie paziņojumi

Paziņojuma numurs	Paziņojuma veids	Paziņojuma teksts
1	Vaicājums	All changes in this form will be lost. Do you want to continue?
2	Vaicājums	Do you really want to exit?
3	Vaicājums	Do you really want to logout?
4	Vaicājums	Do you really want to save this [item]?
5	Vaicājums	Do you really want to delete selected [item]?
6	Informatīvs	Selected category contains at least one product. It cannot be deleted.
7	Kļūda	Error: Category with ID [category_id] not found.
8	Informatīvs	Selected product is associated with at least one order. It cannot be deleted. Selected product's status can be changed to 'Void'.
9	Kļūda	Error: Product with ID [product_id] not found.
10	Informatīvs	Selected partner is associated with at least one order or product. It cannot be deleted. Selected partner's status can be changed to 'Void'.
11	Kļūda	Error: Partner with ID [partner_id] not found.
12	Informatīvs	Selected order has at least one order item. It cannot be deleted. Selected order's status can be changed to 'Void'.
13	Kļūda	Error: Order with ID [order_id] not found.
14	Kļūda	Error: Item count for order with ID [order_id] failed.
15	Kļūda	Error: Order Item with ID [order_item_id] not found.
16	Kļūda	Error: Warehouse Detail with ID [warehouse_detail_id] not found.
17	Kļūda	Error: [Text Field Title] must contain [x-y] characters.
18	Kļūda	Error: [Text Field Title] cannot contain more than [x] characters.
19	Kļūda	Error: [Numeric Field Title] must be non-negative integer.
20	Kļūda	Error: [Numeric Field Title] must be non-negative decimal.
21	Kļūda	Error: Product partner not specified.
22	Kļūda	Error: Order partner not specified.

23	Kļūda	Error: Discount too large. Order Total Price (Gross) cannot be negative number.
24	Kļūda	Error: Server not found.
25	Kļūda	Error: Database not found.
26	Kļūda	Error: Incorrect username or password.
27	Kļūda	Error: Server name field is blank. Enter server name and try again.
28	Kļūda	Error: User field is blank. Enter user name and try again.
29	Kļūda	Error: Password field is blank. Enter password and try again.
30	Kļūda	Error: Database field is blank. Enter database name and try again.
31	Informatīvs	Connection successful.

1.4.2. Programmatūras saskarne

WMS lietotne sazinās ar MS SQL Server 2000 vai 2005 datu bāzi, izmantojot platformas .NET Framework funkciju bibliotēkas.

1.5. NEFUNKCIONĀLĀS PRASĪBAS

1.5.1. Veiktspējas prasības

Visu datu tabulu un formu parādīšanai jānotiek 5 sekunžu laikā vai ātrāk, bet sistēmas pārslodzes gadījumā – 20 sekunžu laikā. Sistēmas pārslodze nedrīkst būt biežāk kā 3% no sistēmas darbības laika. Datu bāzes nepieejamības gadījumā attiecīgs kļūdas paziņojums ir jāparāda 15 sekunžu laikā. Ar datu bāzi bez pārslodzes vienlaicīgi jāvar strādāt 1 – 20 lietotājiem.

1.5.2. Pieejamība

Sistēmai jābūt pieejamai 24 stundas dienā, 7 dienas nedēļā, jeb 97% no kopējā sistēmas ekspluatācijas laika.

1.5.3. Datu drošība

Pieslēgums starp WMS lietotni un datu bāzi tiek veidots lokālajā tīklā, kas pazemina datu pārsūtīšanas drošības riskus, salīdzinot ar publisko tīklu. Ja pieslēgšanās notiek caur VPN, datu plūsmu šifrē VPN programmatūra.

1.5.4. Aizsardzība pret nesankcionētu piekļuvi

WMS var lietot tikai lietotāji, kuriem ir pieeja datu bāzei. Pieejas iespējas nosaka datu bāzes servera uzstādījumi. Lietotāju autentifikāciju var veikt divos veidos:

- Ar datu bāzes serverī izveidotiem lietotāju kontiem, kuriem piešķirtas paroles.
- Ar Windows NT Integrated Security – lietotājs datu bāzē tiek autentificēts ar operētājsistēmas pieteikšanās datiem (lietotāja ID kods). Datu bāzes serverī iespējams noteikt, no kuriem tīkla datoriem vai domēna lietotāju kontiem drīkst pieslēgties datu bāzei.

Lietotāja kontam jābūt tiesībām lasīt un rakstīt WMS datu bāzē.

Lietotāju pārbaudi veic funkcija *1.3.1. Darba uzsākšana*.

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Nolūks

Šī dokumenta nolūks ir aprakstīt programmatūras projektējumu programmu sistēmai „Noliktavas pārvaldības sistēma” (angl. val. „Warehouse Management System”, saīsināti WMS). Dokuments ir paredzēts programmatūras izstrādātājiem, kas veic WMS izstrādi, lai parādītu, kā programmatūras prasību specifikācijā noteiktās prasības jārealizē izvēlētajā programmēšanas vidē.

2.1.2. Darbības sfēra

Programmu sistēma „Noliktavas pārvaldības sistēma” ir vienlietotāja MS Windows lietotne, kuras darbības mērķis ir noliktavas datu pārvaldība. Lietotne pieslēdzas kopējai noliktavas datu bāzei un ļauj iegūt un rediģēt nepieciešamos datus. Projekta ietvaros tiek izstrādāta gan lietotne, gan datu bāze. Programmatūra nav paredzēta lielām slodzēm, tās potenciālie izmantotāji ir maza un vidēja izmēra noliktavas. Noliktavas menedžeris ar programmas palīdzību var kontrolēt gan preču jeb produktu katalogu, gan noliktavas partneru sarakstu, gan operatīvi iegūt nepieciešamo informāciju priekš klientiem, pavadzīmju veidošanas u.c., gan reģistrēt ienākošos un izejošos pasūtījumus. WMS pamatuzdevumi ir datu pievienošana un rediģēšana preču katalogā, partneru pārvaldība, pasūtījumu reģistrēšana, papildinformācijas sniegšana par noliktavu un pašu sistēmu, kā arī datu aizsardzība pret nesankcionētu piekļuvi.

2.1.3. Izstrādes metodes

Projekts tiek realizēts programmēšanas valodā Visual Basic .NET, izmantojot izstrādes vidi Visual Studio .NET (2003) un platformu .NET Framework 1.1. Ekrāna formas tiek veidotas ar Visual Studio Form Designer; to programmas kods tiek ģenerēts automātiski. Savukārt visa formu funkcionalitāte tiek programmēta, rakstot kodu. Datu bāzes izveidei tiek izmantots MS SQL Server 2005, nodrošinot savietojamību ar MS SQL Server 2000. Datu bāzes operācijas veic WMS lietotnes Visual Basic funkcijas; datu bāzes procedūras netiek izmantotas.

2.1.4. Definīcijas, akronīmi un saīsinājumi

WMS – saīsinājums no „Warehouse Management System” (latv. val. „Noliktavas pārvaldības sistēma”).

MS – Microsoft.

PK (Primary Key) – datu bāzes tabulas primārā atslēga

FK (Foreign Key) – datu bāzes tabulas ārējā atslēga

PPS – programmatūras prasību specifikācija

Pieteikšanās – lietotāja identificēšanās procedūra, pievienojoties datoram pa sakaru līniju. Šīs procedūras laikā dators parasti pieprasa lietotāja vārdu un paroli.

Apzīmējumi klašu projektējumā:

class – klase

sub – procedūra

function – funkcija

2.1.5. Saistība ar citiem dokumentiem

- Dokuments sagatavots atbilstoši standartam LVS 72:1996 „Informācijas tehnoloģija – Programminženierija – Ieteicamā prakse programmatūras projektējuma aprakstīšanai”.
- WMS programmatūras prasību specifikācija.

2.1.6. Dokumenta pārskats

Dokuments ir sadalīts 4 nodaļās. Pirmajā nodaļā ir ievadinformācija par dokumentu un programmatūru. Otrajā nodaļā ir detalizēts klašu projektējums. Trešajā nodaļā ir datu bāzes detalizēts projektējums. Ceturtajā nodaļā ir atkarību apraksts.

2.2. Klašu detalizēts projektējums

Nodaļā ir uzskaitītas nozīmīgākajās programmatūras klases, funkcijas un procedūras, attēlojot to savstarpējās pakļautības. Izstrādes procesā pēc vajadzības jāpievieno jaunas funkcijas un procedūras, ja tās nepieciešamas PPS prasību izpildei. Programmējot klases, jāpievieno arī papildus funkcijas lietotāja saskarnes prasību nodrošināšanai (formu aizvēršana, ielāde, komandpogu nospiešanas kontrole u.tml.); skat. PPS nodaļas 1.3. *Funkcionālās prasības* un 1.4.1. *Lietotāja saskarne*. Saskarnes elementu funkcionalitātes nodrošināšanai pieļaujama arī jaunu klašu pievienošana.

Katra klase atrodas atsevišķā failā, kura nosaukumu veido „klases_vārds.vb”. Visi klašu faili atrodas lietotnes pirmkoda saknes mapē. Projekta starta klase ir *App_main*. Mantošana starp klasēm netiek izmantota.

Ja datu objektam ir vajadzīga gan pievienošanas, gan labošanas forma, tās realizē ar vienu klasi (rediģēšanas formu). Objekta pievienošanas gadījumā parāda tukšu formu, bet labošanas gadījumā – pirms formas parādīšanas pieprasa no datu bāzes attiecīgā ieraksta datus un ar tiem aizpilda formu. Labošanas un apskates formām datus iegūst ar funkcijām „load_[object_name]_data”, kuras atrodas klasē „Manager_form”.

Nozīmīgāko klašu, funkciju un procedūru saraksts

Apzīmējumi: class – klase; sub – procedūra; function – funkcija.

- Programmas vadības klase (Class *App_main*) – satur globālos mainīgos, „Main” procedūru un vairākas sistēmā bieži izmantotas funkcijas.
 - Sub *Main* – galvenā programmas procedūra.
 - Sub *error_message* – parāda standarta kļūdas paziņojumu ar tekstu, kas nodots kā parametrs.
 - Sub *inform_message* – parāda standarta informatīvo paziņojumu ar tekstu, kas nodots kā parametrs.
 - Function *save_cancel_message* – parāda atgādinājumu par ievadīto datu zaudēšanu datu ievades formās pirms to aizvēršanas, ja nav veikta saglabāšanas darbība.
 - Function *execute_transaction* – izpilda datu bāzes transakciju, kas neatgriež ierakstus (INSERT, UPDATE, DELETE). Kā parametru saņem izpildāmi SQL pieprasījumu. Atgriežamā vērtība – izmainīto ierakstu skaits datu bāzē. Ja transakcija laikā notiek kļūda, atgriež -1.

- Function `is_non_negative_integer` – pārbauda, vai parametā nodotais `STRING` mainīgais ir nenegatīvs vesels skaitlis.
- Function `is_non_negative_decimal` – pārbauda, vai parametā nodotais `STRING` mainīgais ir nenegatīvs decimāldaļskaitlis.
- Pieteikšanās forma (Class `Login_form`)
 - Function `create_conn_str` – no formas ievadlauku datiem veido „connection string”. Ja kāds lauks ir tukšs, parāda kļūdas paziņojumu.
 - Function `test_connection` – pārbauda, vai ar izveidoto „connection string” ir iespējams pieslēgties datu bāzei.
- Programmas galvenā loga klase (Class `Manager_form`)
 - Forma ar informāciju par programmatūru (Class `About_software`)
 - Preču katalogs
 - Kategorijas rediģēšanas forma (Class `Edit_category`)
 - Preces rediģēšanas forma (Class `Edit_product`)
 - Jauna kataloga vienuma veida izvēles forma (Class `New_catalog_item`)
 - Preces datu apskates forma (Class `View_product`)
 - Sub `load_catalog_list` – izveido kataloga sarakstu un parāda to datu tabulā. Kā parametru saņem atveramās kategorijas identifikatoru.
 - Function `load_category_data` – iegūst datus par konkrētu kategoriju un atgriež tos references parametros.
 - Function `load_product_data` – iegūst datus par konkrētu produktu un atgriež tos references parametros.
 - Function `delete_category` – dzēš konkrētu kategoriju
 - Function `delete_product` – dzēš konkrētu produktu
 - Pasūtījumu pārvaldība
 - Pasūtījuma rediģēšanas forma (Class `Edit_order`)
 - Pasūtījumam piesaistīta produkta datu rediģēšanas forma (Class `Edit_order_item`)
 - Pasūtījuma apskates forma (Class `View_order`)
 - Pasūtījumam piesaistīta produkta datu apskates forma (Class `View_order_item`)
 - Sub `load_orders_list` – izveido pasūtījumu sarakstu un parāda to datu tabulā.
 - Function `load_order_data` – iegūst datus par konkrētu pasūtījumu un atgriež tos references parametros.

- Function count_order_items – references parametros atgriež konkrēta pasūtījuma piesaistīto produktu skaitu un to kopējo summu.
 - Function delete_order – dzēš konkrētu pasūtītāju
 - Sub load_order_items_list – izveido pasūtījumam piesaistīto produktu sarakstu un parāda to datu tabulā.
 - Function load_order_item_data – iegūst datus par konkrētu pasūtījumam piesaistītu produktu un atgriež tos references parametros.
 - Function delete_order_item – dzēš konkrēta produkta piesaisti pasūtījumam.
- Noliktavas partneru datu pārvaldība
 - Partnera rediģēšanas forma (Class Edit_partner)
 - Partnera apskates forma (Class View_partner)
 - Sub load_partners_list – izveido noliktavas partneru sarakstu un parāda to datu tabulā.
 - Function load_partner_data – iegūst datus par konkrētu partneri un atgriež tos references parametros.
 - Function delete_partner – dzēš konkrētu partneri
- Noliktavas rekvizīti
 - Noliktavas rekvizīta rediģēšanas forma (Class Edit_detail)
 - Noliktavas rekvizīta apskates forma (Class Edit_detail)
 - Sub load_details_list – izveido noliktavas rekvizītu sarakstu un parāda to datu tabulā.
 - Function load_detail_data – iegūst datus par konkrētu noliktavas rekvizītu un atgriež tos references parametros.
 - Function delete_detail – dzēš konkrētu noliktavas rekvizītu.

2.3. Datu detalizēts projektējums

Šī nodaļa satur datu bāzes tabulu aprakstu. Tabulu primāro atslēgu laukiem automātiski piešķir unikālus identifikatorus ar IDENTITY(1, 1).

2.3.1. Tabula *Warehouse_details*

Tabulā glabājas dati par noliktavas rekvizītiem.

2.1. tabula

Tabulas *Warehouse_details* apraksts

Kolonnas nosaukums, atslēgas (PK, FK)	Datu tips	NULL vērtības	Apraksts
detail_id (PK)	int	NOT NULL	Noliktavas rekvizīta unikāls identifikators
detail_title	nvarchar(50)	NOT NULL	Noliktavas rekvizīta nosaukums
detail_value	nvarchar(200)	NOT NULL	Noliktavas rekvizīta vērtība

2.3.2. Tabula *Partners*

Tabulā glabājas dati par noliktavas partneriem.

2.2. tabula

Tabulas *Partners* apraksts

Kolonnas nosaukums, atslēgas (PK, FK)	Datu tips	NULL vērtības	Apraksts
partner_id (PK)	int	NOT NULL	Partnera unikāls identifikators
partner_type	int	NOT NULL	Partnera tips (0 – klients, 1 – piegādātājs)
partner_status	int	NOT NULL	Partnera statuss (0 – aktīvs, 1 – anulēts)
partner_name	nvarchar(100)	NOT NULL	Partnera nosaukums
partner_domicile	nvarchar(200)	NOT NULL	Partnera juridiskā adrese
partner_office_address	nvarchar(200)	NULL	Partnera biroja adrese
partner_delivery_address	nvarchar(200)	NULL	Partnera piegādes adrese
partner_person	nvarchar(50)	NOT NULL	Partnera kontaktpersona
partner_phone	varchar(25)	NULL	Partnera telefons
partner_fax	varchar(25)	NULL	Partnera fakss
partner_email	varchar(50)	NULL	Partnera e-pasts
partner_reg_code	varchar(25)	NULL	Partnera uzņēmuma reģistrācijas numurs
partner_vat_code	varchar(25)	NULL	Partnera PVN reģistrācijas numurs
partner_bank	nvarchar(50)	NULL	Partnera bankas nosaukums
partner_bank_code	varchar(50)	NULL	Partnera bankas kods

partner_bank_account	varchar(50)	NULL	Partnera bankas norēķinu konts
----------------------	-------------	------	--------------------------------

2.3.3. Tabula *Categories*

Tabulā glabājas dati par preču kataloga kategorijām.

2.3. tabula

Tabulas *Categories* apraksts

Kolonnas nosaukums, atslēgas (PK, FK)	Datu tips	NULL vērtības	Apraksts
category_id (PK)	int	NOT NULL	Kategorijas unikāls identifikators
category_name	nvarchar(50)	NOT NULL	Kategorijas nosaukums
category_parent_id	int	NOT NULL	Kategorijas vecāka identifikators (0 – kategorija atrodas kataloga saknes kategorijā)

2.3.4. Tabula *Products*

Tabulā glabājas dati par noliktavas partneriem.

2.4. tabula

Tabulas *Products* apraksts

Kolonnas nosaukums, atslēgas (PK, FK)	Datu tips	NULL vērtības	Apraksts
product_id (PK)	int	NOT NULL	Produkta unikāls identifikators
category_id (FK)	int	NULL	Kategorijas identifikators (NULL – produkts atrodas kataloga saknes kategorijā); ārējā atslēga Categories (category_id)
partner_id (FK)	int	NOT NULL	Partnera identifikators; ārējā atslēga Partners (partner_id)
product_status	int	NOT NULL	Produkta statuss (0 – aktīvs, 1 – anulēts)
product_title	nvarchar(200)	NOT NULL	Produkta nosaukums
product_description	nvarchar(200)	NOT NULL	Produkta apraksts
product_location	nvarchar(100)	NULL	Produkta atrašanās vieta
product_model_number	varchar(50)	NULL	Produkta modeļa numurs
product_batch_number	varchar(50)	NULL	Produkta partijas numurs
product_manufacturer	nvarchar(100)	NOT NULL	Produkta ražotājs
product_country	nvarchar(50)	NULL	Valsts, kurā produkts ražots
product_measurement	nvarchar(25)	NOT NULL	Produkta mērvienība

product_stock	int	NOT NULL	Produkta vienību skaits noliktavā (vienība – product_measurement)
product_arriving_price	float	NOT NULL	Produkta cena, ievēdot noliktavā
product_current_price	float	NOT NULL	Produkta pašreizējā cena
product_arriving_date	datetime	NOT NULL	Produkta atvešanas datums
product_expire_date	datetime	NULL	Produkta derīguma termiņš

2.3.5. Tabula *Orders*

Tabulā glabājas dati par preču pasūtījumiem.

2.5. tabula

Tabulas *Orders* apraksts

Kolonnas nosaukums, atslēgas (PK, FK)	Datu tips	NULL vērtības	Apraksts
order_id (PK)	int	NOT NULL	Pasūtījuma unikāls identifikators
order_type	int	NOT NULL	Pasūtījuma tips (0 – izejošs, 1 – ienākošs)
order_status	int	NOT NULL	Pasūtījuma statuss (0 – aktīvs, 1 – anulēts, 2 – pabeigts)
order_date	datetime	NULL	Pasūtījuma datums
order_invoice	varchar(50)	NULL	Pasūtījuma pavadzīmes numurs
partner_id (FK)	int	NOT NULL	Partnera identifikators; ārējā atslēga Partners (partner_id)
order_transport_details	nvarchar(200)	NULL	Pasūtījuma transporta informācija
order_package	nvarchar(100)	NULL	Pasūtījuma iepakojums
order_excise_tax	float	NOT NULL	Pasūtījuma akcīzes nodoklis (noklusētā vērtība – 0)
order_nature_tax	float	NOT NULL	Pasūtījuma dabas resursu nodoklis (noklusētā vērtība – 0)
order_vat	float	NOT NULL	Pasūtījuma pievienotās vērtības nodoklis (noklusētā vērtība – 0)
order_discount	float	NOT NULL	Pasūtījuma atlaide (noklusētā vērtība – 0)
order_total_price	float	NOT NULL	Pasūtījuma kopējā summa (pasūtījuma produktu cenas + nodokļi); noklusētā vērtība – 0

2.3.6. Tabula *Orders_items*

Tabulā glabājas dati par pasūtījumiem piesaistītajiem produktiem.

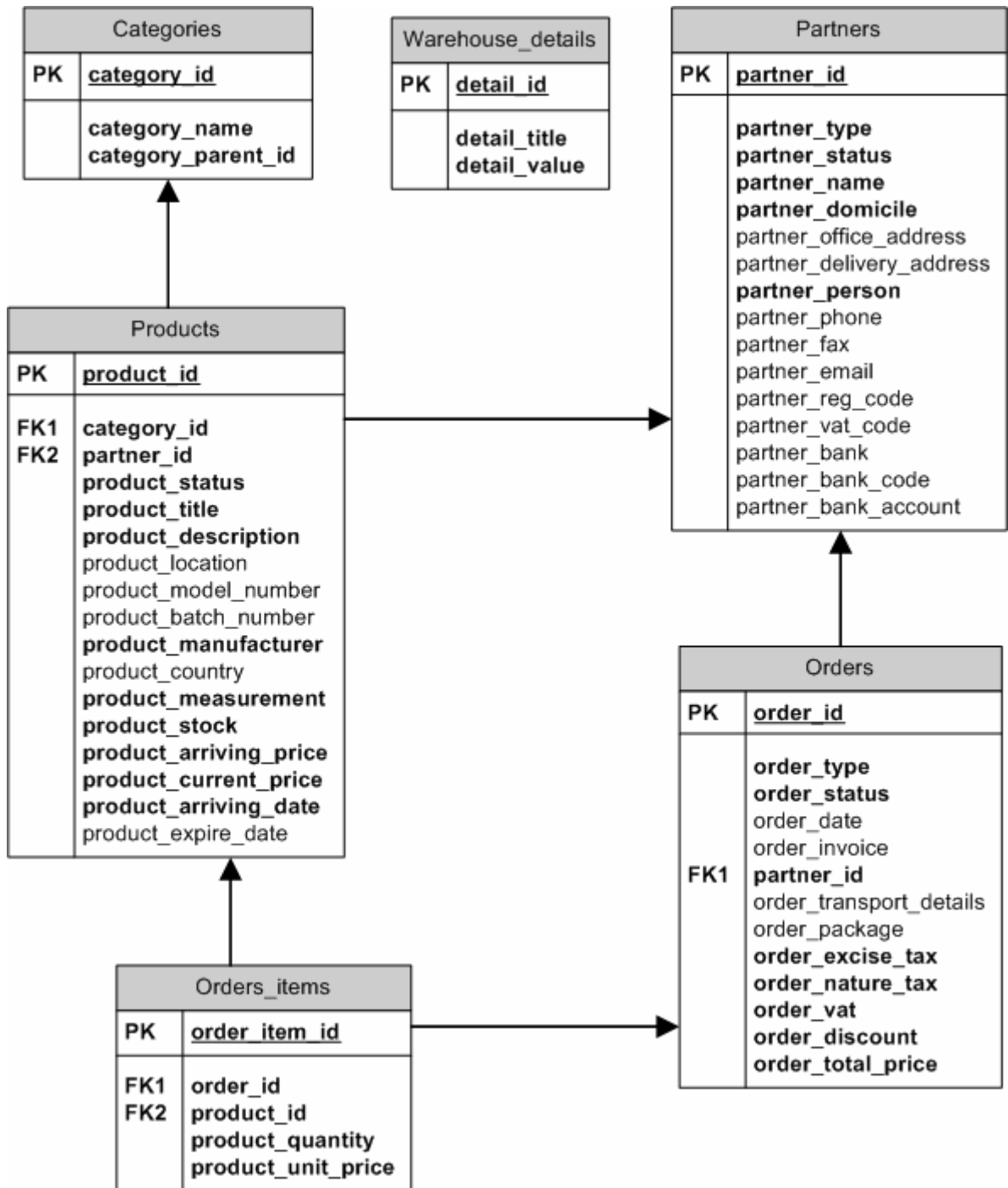
2.6. tabula

Tabulas *Orders_items* apraksts

Kolonnas nosaukums, atslēgas (PK, FK)	Datu tips	NULL vērtības	Apraksts
order_item_id (PK)	int	NOT NULL	Pasūtījumam piesaistīt unikāls identifikators
order_id (FK)	int	NOT NULL	Pasūtījuma identifikators; ārējā atslēga Orders (order_id)
product_id (FK)	int	NOT NULL	Produkta identifikators; ārējā atslēga Products (product_id)
product_quantity	int	NOT NULL	Produkta vienību skaits
product_unit_price	float	NOT NULL	Produkta vienas vienības cena

2.4. Atkarību apraksts

2.4.1. Datu atkarības – ER realizācijas modelis



2.1. att. WMS ER realizācijas modelis

3. TESTĒŠANAS DOKUMENTĀCIJA

Šajā dokumentā ir iekļauta vienībtestēšanas testpiemēru specifikācija un testēšanas žurnāls. Vienībtestēšana tiek veikta pēc melnās kastes principa. Tiek pārbaudīts, vai sistēmai izpilda programmatūras prasību specifikācijā noteitās funkcijas. Datu ievades formas tiek testētas uz ievaddatu datu tipu atbilstību un robežgadījumiem. Paziņojumu numuru atšifrējumus skat. programmatūras prasību specifikācijas nodaļā 1.4.1. *Lietotāja saskarne*.

3.1. Testpiemēru specifikācija

3.1. tabula

Testpiemēru specifikācija

Testējamais vienums	Testa ID	Tests	Sagaidāmie izvaddati
Pieteikšanās forma	1	Nepareizs severa vārds	Paziņojumi nr. 24, 27
	2	Nepareizs datu bāzes lietotājvārds un parole	Paziņojumi nr. 26, 28, 29
	3	Nepareiza datu bāze	Paziņojumi nr. 25, 30
	4	Pareizi pieteikšanās dati	Atveras WMS galvenais logs (komandpoga „Test” – paziņojums nr. 31)
Preču kataloga saraksts	5	Vai parāda datus datu tabulā?	Datu tabulā redzams preču kataloga saraksts
	6	Vai var pārvietoties pa kategorijām?	Atverot kategoriju, parāda tās saturu
	7	Vai var filtrēt sarakstu pēc preces statusa?	Parāda saraksta filtrēšanas rezultātus
	8	Vai var meklēt pēc preces nosaukuma?	Parāda meklēšanas rezultātus
Preces apskates forma	9	Vai parāda preces datus?	Preces apskates formā redzami preces dati
Preces rediģēšanas forma	10	Vai var pievienot jaunu preci?	Prece parādās katalogā
	11	Vai var labot preci?	Var apskatīt izmainītos datus
	12	Nepareizi ievaddati	Paziņojumi nr. 17-20
	13	Vai var piesaistīt partneri?	Partnera vārds parādās preces rediģēšanas formā
Kategorijas rediģēšanas forma	14	Vai var pievienot jaunu kategoriju?	Kategorija parādās katalogā
	15	Vai var labot kategoriju?	Var apskatīt izmainītos datus

	16	Nepareizi ievaddati	Paziņojums nr. 17
Partneru saraksts	17	Vai parāda datus datu tabulā?	Datu tabulā redzams partneru saraksts
	18	Vai var filtrēt sarakstu pēc partnera statusa un tipa?	Parāda saraksta filtrēšanas rezultātus
	19	Vai var meklēt pēc partnera nosaukuma?	Parāda meklēšanas rezultātus
Partnera apskates forma	20	Vai parāda partnera datus?	Partnera apskates formā redzami partnera dati
Partnera rediģēšanas forma	21	Vai var pievienot jaunu partneri?	Partneris parādās partneru sarakstā
	22	Vai var labot partnera datus?	Var apskatīt izmainītos datus
	23	Nepareizi ievaddati	Paziņojumi nr. 17, 18
Pasūtījumu saraksts	24	Vai parāda datus datu tabulā?	Datu tabulā redzams pasūtījumu saraksts
	25	Vai var filtrēt sarakstu pēc pasūtījuma statusa un tipa?	Parāda saraksta filtrēšanas rezultātus
	26	Vai var meklēt pasūtījumu pēc partnera nosaukuma?	Parāda meklēšanas rezultātus
Pasūtījuma apskates forma	27	Vai parāda pasūtījuma datus?	Pasūtījuma apskates formā redzami pasūtījuma dati
Pasūtījuma rediģēšanas forma	28	Vai var pievienot jaunu pasūtījumu?	Pasūtījums parādās katalogā
	29	Vai var labot pasūtījumu?	Var apskatīt izmainītos datus
	30	Nepareizi ievaddati	Paziņojumi nr. 17-20
	31	Vai var piesaistīt partneri?	Partnera vārds parādās pasūtījuma rediģēšanas formā
Pasūtījumam piesaistītas preces apskates forma	32	Vai parāda piesaistītas preces datus?	Pasūtījumam piesaistītas preces apskates formā redzami preces dati
Pasūtījumam piesaistītas preces rediģēšanas forma	33	Vai var piesaistīt pasūtījumam jaunu preci?	Piesaistītā prece parādās piesaistīto preču sarakstā
	34	Vai var labot jau piesaistītas preces datus?	Var apskatīt izmainītos datus
	35	Nepareizi ievaddati	Paziņojumi nr. 17, 18
Noliktavas rekvizītu saraksts	36	Vai parāda datus datu tabulā?	Datu tabulā redzams noliktavas rekvizītu saraksts
Noliktavas rekvizīta apskates forma	37	Vai parāda rekvizīta datus?	Noliktavas rekvizīta apskates formā redzami rekvizīta dati
Noliktavas rekvizīta rediģēšanas forma	38	Vai var pievienot jaunu rekvizītu?	Rekvizīts parādās rekvizītu sarakstā

	39	Vai var labot rekvizīta datus?	Var apskatīt izmainītos datus
	40	Nepareizi ievaddati	Paziņojums nr. 17
Forma ar informāciju par programmatūru	41	Vai parāda informāciju par programmatūru?	Formā redzama informācija par izstrādātāju un programmas versijas numurs

3.2. Testēšanas žurnāls

Iespējamās bīstamības pakāpes testēšanas žurnālā: zema, vidēja, augsta, kritiska.

3.2. tabula

Testēšanas žurnāls

WMS versija	Testa ID	Rezultāts	Bīstamības pakāpe	Problēmas ziņojums	Testēšanas datums
1.00	1	+			20.05.2005.
1.00	2	+			20.05.2005.
1.00	3	+			20.05.2005.
1.00	4	-	Vidēja	Pirms galvenā loga parādīšanas parāda paziņojumu nr. 31, kas nav vajadzīgs.	20.05.2005.
1.00	5	+			20.05.2005.
1.00	6	-	Vidēja	Iezīmējot tabulas rindu, ar kuru pārvietojas katalogā uz augšu par vienu kategoriju, netiek atspējotas komandpogas „Edit” un „Delete”	20.05.2005.
1.00	7	+			20.05.2005.
1.00	8	+			20.05.2005.
1.00	9	+			20.05.2005.
1.00	10	+			20.05.2005.
1.00	11	+			20.05.2005.
1.00	12	-	Zema	Ja nav ievadīta preces mērvienība, parāda nepareizu paziņojumu: "Error: Manufacturer must contain 1-25 characters."	20.05.2005.
1.00	13	-	Vidēja	Neparāda partnera vārdu preces rediģēšanas formā, taču pašu preci ir iespējams saglabāt.	20.05.2005.
1.00	14	+			20.05.2005.
1.00	15	+			20.05.2005.
1.00	16	+			20.05.2005.

1.00	17	+			20.05.2005.
1.00	18	+			20.05.2005.
1.00	19	+			20.05.2005.
1.00	20	+			20.05.2005.
1.00	21	-	Zema	Saglabājot partneri, parāda dialoglogu ar SQL koda fragmentu (ieviests atklūdošanas vajadzībām)	20.05.2005.
1.00	22	+			20.05.2005.
1.00	23	+			20.05.2005.
1.00	24	+			20.05.2005.
1.00	25	+			20.05.2005.
1.00	26	+			20.05.2005.
1.00	27	+			20.05.2005.
1.00	28	+			20.05.2005.
1.00	29	-			20.05.2005.
1.00	30	+			20.05.2005.
1.00	31	+			20.05.2005.
1.00	32	+			20.05.2005.
1.00	33	+			20.05.2005.
1.00	34	+			20.05.2005.
1.00	35	+			20.05.2005.
1.00	36	+			20.05.2005.
1.00	37	+			20.05.2005.
1.00	38	-	Augsta	Dati netiek saglabāti	20.05.2005.
1.00	39	+			20.05.2005.
1.00	40	+			20.05.2005.
1.00	41	+			20.05.2005.
1.01	4	+			24.05.2005.
1.01	6	+			24.05.2005.
1.01	12	+			24.05.2005.
1.01	13	+			24.05.2005.
1.01	21	+			24.05.2005.
1.01	38	+			24.05.2005.

4. PROJEKTA REALIZĀCIJA

4.1. Projekta organizācija

Projekts tika izstrādāts individuāli, tāpēc darbu sadales problēma dažādiem izstrādātājiem nebija aktuāla. Gan programmas kodu, gan visu dokumentāciju izstrādāja darba autors. Projekta tapšanas gaita:

- Idejas konkretizēšana.
- Izstrādes vides un valodas izvēle. Tika izmēģināta gan valoda Visual C#, gan datu bāzu pārvaldības sistēma MySQL. Tomēr tika nolemts izmantot valodu Visual Basic un datu bāzu pārvaldības sistēmu MS SQL Server, jo darba autoram tajās bija lielāka izstrādes pieredze, turklāt MS SQL Server strādā labāk ar lieliem datu apjomiem nekā MySQL.
- Programmatūras prasību specifikācijas uzmetums.
- Datu bāzes struktūras izveidošana un programmēšana.
- Lietotnes programmēšana un paralēla programmatūras projektējuma apraksta veidošana un prasību precizēšana prasību specifikācijā.
- Testēšanas dokumentācijas izstrāde
- Pēc pirmās lietotnes versijas pabeigšanas tika veikta vienībtestēšana.
- Atrasto kļūdu labošana un papildinātas lietotnes versijas izlaišana.

4.2. Kvalitātes nodrošināšana

- Tika izstrādāta Latvijas standartiem atbilstoša programmatūras prasību specifikācija un programmatūras projektējuma apraksts.
- Programmas kodā visiem datu bāzes pieprasījumiem tika veikta izņēmuma situāciju kontrole. Kodā netika lietotas tādas potenciāli bīstamas konstrukcijas kā GOTO. Garākās funkcijas pēc iespējas tika sadalītas mazākās, tādējādi uzlabojot koda pārskatāmību.

4.3. Konfigurāciju pārvaldība

Lietotnei versiju numuri tiek piešķirti formā „X.YY”. Cipara „X” maiņa nozīmē nozīmīgas izmaiņas lietotnes saskarnē un funkcionalitātē. Ciparu „YY” maiņa teik veikta kļūdu labojumu un maznozīmīgu saskarnes un funkcionalitātes izmaiņu gadījumā.

WMS lietotnei tika izlaistas divas versijas: 1.00 un 1.01. Versijā 1.01 bija izlabotas testēšanā atrastās kļūdas.

Veriju vadībai tiek izmatota versionēšanas sistēma Subversion ar MS Windows klienta programmatūru TortoiseSVN. Subversion WMS projekta mapē atrodas:

- Mape TRUNC – uzglabā jaunākās, vēl nepabeigtās lietotnes verijas pirmkodus.
- Mape VERSIONS – uzglabā iepriekšējo lietotnes versiju pirmkodus.

Tā kā projekts tika izstrādāts individuāli, dati Subversion datu bāzē tika atjaunoti tikai līdz ar jaunas lietotnes versijas izlaišanu.

4.4. Darbietilpības novērtējums

Darbietilpības novērtēšanai tika izmantota COCOMO metode.

4.1. tabula

Ievadu un izvadu sarežģītības novērtējums

Lietotājam redzamās ievades un izvades formas	Formas veids	Datu elementu skaits	Datu failu skaits	Sarežģītība
Pieteikšanās forma	Ievads	4	1	Vienkāršs
Preču kataloga saraksts	Izvads	8	2	Vidējs
Preces apskates forma	Izvads	15	3	Vidējs
Preces rediģēšanas forma	Ievads	15	3	Vidējs
Kategorijas rediģēšanas forma	Ievads	1	1	Vienkāršs
Partneru saraksts	Izvads	3	1	Vienkāršs
Partnera apskates forma	Izvads	15	1	Vienkāršs
Partnera rediģēšanas forma	Ievads	15	1	Vienkāršs
Pasūtījumu saraksts	Izvads	6	2	Vidējs
Pasūtījuma apskates forma	Izvads	15	3	Vidējs
Pasūtījuma rediģēšanas forma	Ievads	15	3	Vidējs
Pasūtījumam piesaistītas preces apskates forma	Izvads	3	1	Vienkāršs
Pasūtījumam piesaistītas preces rediģēšanas forma	Ievads	3	1	Vienkāršs
Noliktavas rekvizītu saraksts	Izvads	2	1	Vienkāršs
Noliktavas rekvizīta apskates forma	Izvads	2	1	Vienkāršs
Noliktavas rekvizīta rediģēšanas forma	Ievads	2	1	Vienkāršs
Forma ar informāciju par programmatūru	Izvads	2	0	Vienkāršs

5 vienkārši ievadi, 2 vidēji ievadi, 6 vienkārši izvadi un 4 vidēji izvadi kopā dod 77 funkciju punktus. Valodas Visual Basic vidējais koda rindiņu skaitu uz vienu funkciju punktu ir 40 rindiņas. Sareizinot funkciju punktu skaitu ar šo skaitli, iegūst kopējo paredzamo lietotnes koda rindiņu skaitu – 3080 koda rindiņas. Noslēgumā pēc personmēnešu aprēķināšanas formulas iegūst, ka paredzamais personmēnešu skaits programmas izstrādei ir 5,2 personmēneši.

Ja koda rindiņu skaits tika prognozēts diezgan precīzi (programmai kopā tika uzrakstītas nedaudz virs 2800 rindiņām), tad aprēķinātais personmēnešu skaits bija daudz neobjektīvāks. Projekta īstenošanai tika veltīti apmēram 3 personmēneši. Šī atšķirība ir skaidrojama ar to, ka COCOMO metode ir paredzēta vidējiem un lieliem programmizstrādes projektiem.

5. PROGRAMMAS KODS

Šajā nodaļā ir dots ieskats izstrādātās lietotnes pirmkodā. Ievietoti 4 faili ar kopējo garumu 500 rindiņas – MS SQL Server datu bāzes izveides skripts un trīs Visual Basic .NET klases. Kopējais uzrakstīto koda rindiņu skaits ir nedaudz virs 2800 (neskaitot MS Visual Studio Windows Forms Designer automātiski ģenerēto kodu). Automātiski ģenerētais kods ievietotajos failos ir izlaists.

5.1. Fails „database.sql”

Šajā failā ir MS SQL Server datu bāzes izveides skripts.

```
/* *****  
NPS datu baazes izveides skripts  
Datu baazu paarvaldiibas sistema: MS SQL Server 2005  
*/  
  
CREATE DATABASE WMS  
GO  
  
USE WMS  
GO  
  
CREATE TABLE Warehouse_details  
(  
    detail_id INT NOT NULL IDENTITY(1, 1),  
    detail_title NVARCHAR(50) NOT NULL,  
    detail_value NVARCHAR(200) NOT NULL,  
    PRIMARY KEY (detail_id)  
)  
GO  
  
CREATE TABLE Partners  
(  
    partner_id INT NOT NULL IDENTITY(1, 1),  
    partner_type INT NOT NULL,  
    partner_status INT NOT NULL,  
    partner_name NVARCHAR(100) NOT NULL,  
    partner_domicile NVARCHAR(200) NOT NULL,  
    partner_office_address NVARCHAR(200) DEFAULT NULL,  
    partner_delivery_address NVARCHAR(200) DEFAULT NULL,  
    partner_person NVARCHAR(50) NOT NULL,  
    partner_phone VARCHAR(25) DEFAULT NULL,  
    partner_fax VARCHAR(25) DEFAULT NULL,  
    partner_email VARCHAR(50) DEFAULT NULL,  
    partner_reg_code VARCHAR(25) DEFAULT NULL,  
    partner_vat_code VARCHAR(25) DEFAULT NULL,  
    partner_bank NVARCHAR(50) DEFAULT NULL,  
    partner_bank_code VARCHAR(50) DEFAULT NULL,  
    partner_bank_account VARCHAR(50) DEFAULT NULL,  
    PRIMARY KEY (partner_id)  
)  
GO
```

CREATE TABLE Categories

```
(
    category_id INT NOT NULL IDENTITY(1, 1),
    category_name NVARCHAR(50) NOT NULL,
    category_parent_id INT NOT NULL,
    PRIMARY KEY (category_id)
)
GO
```

CREATE TABLE Products

```
(
    product_id INT NOT NULL IDENTITY(1, 1),
    category_id INT NOT NULL,
    partner_id INT NOT NULL,
    product_status INT NOT NULL,
    product_title NVARCHAR(200) NOT NULL,
    product_description NVARCHAR(200) NOT NULL,
    product_location NVARCHAR(100) DEFAULT NULL,
    product_model_number VARCHAR(50) DEFAULT NULL,
    product_batch_number VARCHAR(50) DEFAULT NULL,
    product_manufacturer NVARCHAR(100) NOT NULL,
    product_country NVARCHAR(50) DEFAULT NULL,
    product_measurement NVARCHAR(25) NOT NULL,
    product_stock INT NOT NULL,
    product_arriving_price FLOAT NOT NULL,
    product_current_price FLOAT NOT NULL,
    product_arriving_date DATETIME NOT NULL,
    product_expire_date DATETIME DEFAULT NULL,
    PRIMARY KEY (product_id),
    FOREIGN KEY (category_id) REFERENCES Categories (category_id),
    FOREIGN KEY (partner_id) REFERENCES Partners (partner_id)
)
GO
```

CREATE TABLE Orders

```
(
    order_id INT NOT NULL IDENTITY(1, 1),
    order_type INT NOT NULL,
    order_status INT NOT NULL,
    order_date DATETIME DEFAULT NULL,
    order_invoice VARCHAR(50) DEFAULT NULL,
    partner_id INT NOT NULL,
    order_transport_details NVARCHAR(200) DEFAULT NULL,
    order_package NVARCHAR(100) DEFAULT NULL,
    order_excise_tax FLOAT NOT NULL DEFAULT 0,
    order_nature_tax FLOAT NOT NULL DEFAULT 0,
    order_vat FLOAT NOT NULL DEFAULT 0,
    order_discount FLOAT NOT NULL DEFAULT 0,
    order_total_price FLOAT NOT NULL DEFAULT 0,
    PRIMARY KEY (order_id),
    FOREIGN KEY (partner_id) REFERENCES Partners (partner_id),
)
GO
```

CREATE TABLE Orders_items

```
(
    order_item_id INT NOT NULL IDENTITY(1, 1),
    order_id INT NOT NULL,
    product_id INT NOT NULL,
    product_quantity INT NOT NULL,
    product_unit_price FLOAT NOT NULL,
    PRIMARY KEY (order_item_id),
)
```

```

FOREIGN KEY (order_id) REFERENCES Orders (order_id),
FOREIGN KEY (product_id) REFERENCES Products (product_id)
)
GO

```

5.2. Fails „App_main.vb”

Šajā failā ir lietotnes vadības klase, kas satur globālos mainīgos, „Main” procedūru un vairākas sistēmā bieži izmantotas funkcijas.

```

Imports System.Data
Imports System.Data.SqlClient
Imports System.Threading
Imports System.Globalization
Imports System.Text

Friend Class App_main
    'Globaalie mainiigie
    Friend Shared app_action As String
    Friend Shared change_active_form As Boolean
    Friend Shared action_completed As Boolean
    Friend Shared db_conn As SqlConnection
    Friend Shared db_command As SqlCommand
    'Ieziimeetaas datu tabulas rindas ID numurs (0 - neviena rinda nav ieziiemeeta)
    Friend Shared list_item_id As Integer
    'Noraada, vai peec ievades formas datu saglabaashanas meegjinaajuma
    'datu baazes izmainju deelj ir jaaatjauno datu tabula (TRUE - ir jaaatjauno)
    Friend Shared refresh_list As Boolean

    <STAThread(> _
    Friend Shared Sub Main()
        App_main.app_action = "Login Form"
        change_active_form = False
        action_completed = False
        db_conn = New SqlConnection
        db_command = New SqlCommand
        db_command.Connection = db_conn
        db_command.CommandType = CommandType.Text
        list_item_id = 0
        refresh_list = False

        Thread.CurrentThread.CurrentUICulture = New CultureInfo("en-US", False)
        Thread.CurrentThread.CurrentCulture = New CultureInfo("en-US", False)

        Do While App_main.app_action = "Login Form"
            Dim loginfrm As New Login_form
            Application.Run(loginfrm)
            Select Case app_action
                Case "Manager Form"
                    Dim managfrm As New Manager_form
                    Application.Run(managfrm)
                Case "exit"
                    db_conn.Dispose()
                    Return
                Case Else
                    db_conn.Dispose()
                    Return
            End Select

```

```

    Loop
    db_conn.Dispose()
End Sub

Friend Shared Sub error_message(ByVal message_text As String)
    MessageBox.Show(message_text, "Error", _
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
End Sub

Friend Shared Sub inform_message(ByVal message_text As String)
    MessageBox.Show(message_text, "Warehouse Management System", _
        MessageBoxButtons.OK, MessageBoxIcon.Information)
End Sub

Friend Shared Function save_cancel_message() As Boolean
    If MessageBox.Show("All changes in this form will be lost. Do you want to continue?", "Warehouse
Management System", _
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question) = DialogResult.OK Then
        Return True
    Else
        Return False
    End If
End Function

Friend Shared Function execute_transaction(ByVal sql As String) As Integer
    Dim db_transact As SqlTransaction
    Dim affected_rows As Integer = -1
    db_command.CommandText = sql
    Try
        db_conn.Open()
        db_transact = db_conn.BeginTransaction(IsolationLevel.Serializable)
        db_command.Transaction = db_transact
        affected_rows = db_command.ExecuteNonQuery()
        db_transact.Commit()
    Catch ex As Exception
        If Not db_transact Is Nothing Then db_transact.Rollback()
        affected_rows = -1
        App_main.error_message(ex.Message)
    Finally
        App_main.db_conn.Close()
    End Try
    Return affected_rows
End Function

Friend Shared Function is_non_negative_integer(ByVal str As String) As Boolean
    If IsNumeric(str) = True And InStr(str, ".") = False And InStr(str, ",") = False And InStr(str, "-") = False
Then
        Return True
    Else
        Return False
    End If
End Function

Friend Shared Function is_non_negative_decimal(ByVal str As String) As Boolean
    If IsNumeric(str) = True And InStr(str, "-") = False And InStr(str, ",") = False Then
        Return True
    Else
        Return False
    End If
End Function
End Class

```

5.3. Fails „Login_form.vb”

Šajā failā ir klase, kas veido lietotāja pieteikšanās formu un tās funkcionalitāti.

```
Imports System.Data
Imports System.Data.SqlClient

Public Class Login_form
    Inherits System.Windows.Forms.Form

    Private new_conn_str As String

    #Region " Windows Form Designer generated code "
    'Windows Form Designer ģenerētais kods
    #End Region

    Private Sub integr_security_CheckStateChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles integr_security.CheckStateChanged
        If integr_security.CheckState = CheckState.Checked Then
            GroupBox1.Enabled = False
        Else
            GroupBox1.Enabled = True
        End If
    End Sub

    Private Sub testbutton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles testbutton.Click
        Dim result As Boolean = test_connection()
        If result = True Then
            App_main.inform_message("Connection successful.")
        End If
    End Sub

    Private Sub login_button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles login_button.Click
        Dim result As Boolean = test_connection()
        If result = True Then
            App_main.app_action = "Manager Form"
            Me.Close()
        End If
    End Sub

    Private Sub exit_button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles exit_button.Click
        App_main.app_action = "exit"
        Me.Close()
    End Sub

    Private Function test_connection() As Boolean
        Dim result As Boolean = create_conn_str()

        If result = True Then
            'Piesleegšana datu baazei
            Try
                App_main.db_conn.ConnectionString = new_conn_str
                App_main.db_conn.Open()
            Catch exSQL As SqlException
                Dim errorMsg As String
                Dim se As SqlError
```

```

se = exSQL.Errors.Item(0)
Select Case se.Number
    Case 17
        ErrorMsg = "Error: Server not found."
    Case 4060
        ErrorMsg = "Error: Database not found."
    Case 18456
        ErrorMsg = "Error: Incorrect username or password."
    Case Else
        ErrorMsg = se.Message
End Select
App_main.error_message(ErrorMsg)
result = False
Catch ex As Exception
    App_main.error_message(ex.Message)
    result = False
Finally
    App_main.db_conn.Close()
End Try
End If
Return result
End Function

Private Function create_conn_str() As Boolean
    Dim Servername As String = server_name.Text
    Dim DBname As String = db_name.Text
    Dim username As String = db_user_name.Text
    Dim passwd As String = db_password.Text

    'Ievadlauku paarbaude
    If Servername = "" Then
        App_main.error_message("Error: Server name field is blank. Enter server name and try again.")
        Return False
    End If
    If integr_security.CheckState = CheckState.Unchecked Then
        If username = "" Then
            App_main.error_message("Error: User field is blank. Enter user name and try again.")
            Return False
        ElseIf passwd = "" Then
            App_main.error_message("Error: Password field is blank. Enter password and try again.")
            Return False
        End If
    End If
    If DBname = "" Then
        App_main.error_message("Error: Database field is blank. Enter database name and try again.")
        Return False
    End If

    'Connection String veidoshana
    If integr_security.CheckState = CheckState.Checked Then
        new_conn_str = "Data Source = " & Servername & _
            "; Initial Catalog = " & DBname & "; Integrated Security=SSPI; " & _
            "Connection Timeout=15"
    Else
        new_conn_str = "Data Source = " & Servername & _
            "; Initial Catalog = " & DBname & "; User ID = " & username & _
            "; Password = " & passwd & "; Connection Timeout=15"
    End If
    Return True
End Function
End Class

```

5.4. Fails „Edit_product.vb”

Šajā failā ir klase, kas veido preces rediģēšanas formu un tās funkcionalitāti.

```
Imports System.Data
Imports System.Data.SqlClient

Public Class Edit_product
    Inherits System.Windows.Forms.Form

    Friend form_type As String
    Friend product_id As Integer
    Friend category_id As Integer
    Friend partner_id As Integer

    #Region " Windows Form Designer generated code "
    'Windows Form Designer ģenerētais kods
    #End Region

    Private Sub Edit_product_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
        If form_type = "new" Then
            Me.Text = "New Product"
            status_panel.Enabled = False
            active_button.Checked = True
            exp_date_not_specified.Checked = True
        ElseIf form_type = "edit" Then
            Me.Text = "Edit Product"
        End If
    End Sub

    Private Sub Edit_product_Closing(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
        'Ja nav jaaatjauno datu tabula, paraada izmainju atcelšanas paziņojumu
        If App_main.refresh_list = False Then
            If App_main.save_cancel_message = False Then
                e.Cancel = True
            End If
        End If
    End Sub

    Private Sub Edit_product_Closed(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Closed
        App_main.change_active_form = True
        App_main.action_completed = True
    End Sub

    Private Sub cancel_button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cancel_button.Click
        Me.Close()
    End Sub

    Private Sub save_button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles save_button.Click
        If validate_product_form() = False Then Return

        If MessageBox.Show("Do you really want to save this product?", "Warehouse Management System", _
            MessageBoxButtons.OKCancel, MessageBoxIcon.Question) = DialogResult.Cancel Then
            Return
        End If
    End Sub
End Class
```

```

Dim sql As String
Dim affected_rows As Integer
Dim status_str As String
Dim expire_date_str As String

If active_button.Checked Then status_str = "0"
If void_button.Checked Then status_str = "1"
If exp_date_not_specified.CheckState = CheckState.Checked Then
    expire_date_str = "NULL"
Else
    expire_date_str = "" & exp_date_calendar.SelectionStart & ""
End If

If form_type = "new" Then
    Dim prm_category_id As New SqlParameter("@category_id", SqlDbType.Int, 4)
    App_main.db_command.Parameters.Add(prm_category_id)
    If category_id <> 0 Then
        App_main.db_command.Parameters("@category_id").Value = category_id
    Else
        App_main.db_command.Parameters("@category_id").Value = DBNull.Value
    End If
    sql = "INSERT INTO Products " & _
        "VALUES (@category_id, " & _
        partner_id.ToString & ", " & status_str & ", " & _
        title_field.Text & ", " & description_field.Text & ", " & _
        location_field.Text & ", " & model_field.Text & ", " & _
        batch_field.Text & ", " & manufacturer_field.Text & ", " & _
        country_field.Text & ", " & measurement_field.Text & ", " & _
        stock_field.Text & ", ROUND(" & arr_price_field.Text & ", 2), ROUND(" & _
        curr_price_field.Text & ", 2), " & arr_date_calendar.SelectionStart & ", " & _
        expire_date_str & ")"
    affected_rows = App_main.execute_transaction(sql)
    App_main.db_command.Parameters.Clear()
    If affected_rows > 0 Then
        App_main.refresh_list = True
    End If
ElseIf form_type = "edit" Then
    sql = "UPDATE Products SET partner_id = " & partner_id.ToString & _
        ", product_status = " & status_str & _
        ", product_title = " & title_field.Text & _
        ", product_description = " & description_field.Text & _
        ", product_location = " & location_field.Text & _
        ", product_model_number = " & model_field.Text & _
        ", product_batch_number = " & batch_field.Text & _
        ", product_manufacturer = " & manufacturer_field.Text & _
        ", product_country = " & country_field.Text & _
        ", product_measurement = " & measurement_field.Text & _
        ", product_stock = " & stock_field.Text & _
        ", product_arriving_price = ROUND(" & arr_price_field.Text & _
        ", 2), product_current_price = ROUND(" & curr_price_field.Text & _
        ", 2), product_arriving_date = " & arr_date_calendar.SelectionStart & _
        ", product_expire_date = " & expire_date_str & _
        " WHERE product_id = " & product_id.ToString
    affected_rows = App_main.execute_transaction(sql)
    If affected_rows >= 0 Then
        App_main.refresh_list = True
        If affected_rows = 0 Then
            App_main.error_message("Error: Product with ID " & _
                App_main.list_item_id.ToString & " not found.")
        End If
    End If
End If
End If

```

```

    If App_main.refresh_list = False Then Return
    Me.Close()
End Sub

Private Sub partner_button_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
partner_button.Click
    Me.Hide()
    App_main.change_active_form = True
End Sub

Private Sub exp_date_not_specified_CheckStateChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles exp_date_not_specified.CheckStateChanged
    If exp_date_not_specified.CheckState = CheckState.Checked Then
        exp_date_calendar.Visible = False
    Else
        exp_date_calendar.Visible = True
    End If
End Sub

Private Function validate_product_form() As Boolean
    If title_field.Text.Length < 1 Or title_field.Text.Length > 200 Then
        App_main.error_message("Error: Product Title must contain 1-200 characters.")
        Return False
    ElseIf description_field.Text.Length < 1 Or description_field.Text.Length > 200 Then
        App_main.error_message("Error: Product Description must contain 1-200 characters.")
        Return False
    ElseIf location_field.Text.Length > 100 Then
        App_main.error_message("Error: Location cannot contain more than 100 characters.")
        Return False
    ElseIf model_field.Text.Length > 50 Then
        App_main.error_message("Error: Model Number cannot contain more than 50 characters.")
        Return False
    ElseIf batch_field.Text.Length > 50 Then
        App_main.error_message("Error: Batch Number cannot contain more than 50 characters.")
        Return False
    ElseIf manufacturer_field.Text.Length < 1 Or manufacturer_field.Text.Length > 100 Then
        App_main.error_message("Error: Manufacturer must contain 1-100 characters.")
        Return False
    ElseIf country_field.Text.Length > 50 Then
        App_main.error_message("Error: Country cannot contain more than 50 characters.")
        Return False
    ElseIf measurement_field.Text.Length < 1 Or measurement_field.Text.Length > 25 Then
        App_main.error_message("Error: Measurement must contain 1-25 characters.")
        Return False
    ElseIf App_main.is_non_negative_integer(stock_field.Text) = False Then
        App_main.error_message("Error: Stock must be non-negative integer.")
        Return False
    ElseIf App_main.is_non_negative_decimal(arr_price_field.Text) = False Then
        App_main.error_message("Error: Arriving Price must be non-negative decimal.")
        Return False
    ElseIf App_main.is_non_negative_decimal(curr_price_field.Text) = False Then
        App_main.error_message("Error: Current Price must be non-negative decimal.")
        Return False
    ElseIf partner_id = Nothing Then
        App_main.error_message("Error: Product partner not specified.")
        Return False
    End If
    Return True
End Function
End Class

```

SECINĀJUMI

- Lietotne „Noliktavas pārvaldības sistēma” ir izstrādāta un gatava lietošanai. Tika sasniegta visa sākotnēji plānotā funkcionalitāte.
- Projekta gaitā ir izstrādāta programmatūras prasību specifikācija, programmatūras projektējuma apraksts un testēšanas dokumentācija.
- Izstrādes laikā ir veikta programmatūras vienībtestēšana.
- Izstrādātā programmatūra nav izmēģināta reālos noliktavas darba apstākļos.
- Novērtējot projekta darbietilpību, var secināt, ka izmantotā COCOMO metode nepieciešamo personmēnešu noteikšanai nav objektīva priekš maziem projektiem.
- Projektam ir vairākas perspektīvas – pilnveidot izstrādātās „Noliktavas pārvaldības sistēmas” funkcionalitāti, kā arī izveidot tīmekļa lietotni priekš klientiem preču kataloga apskatei un preču pasūtīšanai, kas izmantotu šo pašu datu bāzi.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. *.NET Class Library*. **In:** Microsoft Developer Network Library [tiešsaiste] - [atsauce 20.05.2007.]. Pieejams: <http://msdn2.microsoft.com/en-us/library/aa388745.aspx>
2. *About SelectedIndexChanged Event in ListView Control*. **In:** TheScripts Developer Network Forum [tiešsaiste] - [atsauce 16.05.2007.]. Pieejams: <http://www.thescripts.com/forum/thread374880.html>
3. **Crowley, J.** *ListView Control*. **In:** Developer Fusion [tiešsaiste] - [atsauce 11.05.2007.]. Pieejams: <http://www.developerfusion.co.uk/show/74/>
4. *Software Requirements Specification Template* [tiešsaiste] - [atsauce 25.03.2007.]. Pieejams: http://www.processimpact.com/process_assets/srs_template.doc
5. *SQL Server 2005 Books Online: Transact-SQL Reference*. **In:** Microsoft Developer Network Library [tiešsaiste] - [atsauce 20.05.2007.]. Pieejams: <http://msdn2.microsoft.com/en-us/library/ms189826.aspx>
6. **Sawyer, S. A.** *Casting Reference Types in Visual Basic .NET*. **In:** Code Magazine, 2003 [tiešsaiste] - [atsauce 14.04.2007.]. Pieejams: <http://www.code-magazine.com/Article.aspx?quickid=0301111>
7. **Šmite, D., Dosbergs, D., Borzovs, J.** *Informācijas un komunikācijas tehnoloģijas nozares tiesību un standartu pamati*. Rīga : LU Akadēmiskais apgāds, 2005. 207 lpp.
8. *Visual Studio .NET: Visual Basic Reference*. **In:** Microsoft Developer Network Library [tiešsaiste] - [atsauce 20.05.2007.]. Pieejams: [http://msdn2.microsoft.com/en-us/library/25kad608\(VS.71\).aspx](http://msdn2.microsoft.com/en-us/library/25kad608(VS.71).aspx)

DOKUMENTĀRĀ LAPA

Kvalifikācijas darbs „Noliktavas pārvaldības sistēma” izstrādāts LU Fizikas un matemātikas fakultātē, Datorikas nodaļā.

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

Autors: Raimonds Spricis _____

Rekomendēju darbu aizstāvēšanai

Vadītājs: docents Dr. dat. Ģirts Karnītis _____

Recenzents: _____

Darbs iesniegts Datorikas nodaļā _____

Sekretārs: _____

Darbs aizstāvēts kvalifikācijas darbu aizstāvēšanas komisijas sēdē

Komisijas sekretārs: