

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

PASŪTĪJUMU PĀRVALDĪBAS SISTĒMA

KVALIFIKĀCIJAS DARBS

Autors: Aleksejs Grigorjevs
Studenta apliecības Nr.: ag17082
Darba vadītājs: Dr. dat. Jānis Zuters

RĪGA 2019

ANOTĀCIJA

Pasūtījumu pārvaldības sistēma ir publiska tiešsaistes sistēma, kas ļauj uzņēmumu (tipiski e-veikalu) klientiem ērti noformēt un pašiem uzņēmumiem saņemt pasūtījumus uz pakalpojumiem vai produktiem. Integrācija ar trešajām sistēmām ļauj izstrādāt vai izmantot trešo pušu lietotnes sistēmas lietošanai un/vai pārvaldībai. Sistēma ļauj klientiem noformēt jaunus pasūtījumus, atsekot esošo pasūtījumu statusus, kā arī uzņēmumiem pārvaldīt ienākošos pasūtījumus.

ATSLĒGVĀRDI: Pasūtījumi, pārvaldība, Laravel, Bootstrap, PHPUnit.

ABSTRACT

ORDER MANAGEMENT SYSTEM

Order management system is a public online system, which allows businesses (typically e-stores) to receive orders for products or services. Integration with third-party systems enables any third-party applications like warehouse systems to be used with the system for automatic actions, like warehouse refills. Order management system allows individuals to conduct orders, track existing orders and their statuses, and lets businesses handling incoming orders.

KEYWORDS: Orders, management, Laravel, Bootstrap, PHPUnit.

SATURS

APZĪMĒJUMU SARAKSTS	7
1. IEVADS	8
1.1. Nolūks	8
1.2. Darbības sfēra	8
1.3. Saistība ar citiem dokumentiem	8
1.4. Pārskats	8
2. VISPĀRĒJAIS APRAKSTS	10
2.1. Produkta perspektīva	10
2.2. Sistēmas lietotāju lomas	11
2.3. Produkta funkcijas	12
2.4. Vispārējie ierobežojumi	17
2.5. Pieņēmumi un atkarības	17
3. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	18
3.1. Funkcionālās prasības	18
3.2. Funkciju sadalījums pa moduļiem/komponentiem	18
3.2.1. Lietotāju modulis	19
3.2.1.1. Lietotāja reģistrācija	20
3.2.1.2. Lietotāja autentifikācija	21
3.2.1.3. Lietotāja paroles atjaunošana	22
3.2.1.4. Lietotāju saraksta skatīšana	23
3.2.1.5. Lietotāja izveide	24
3.2.1.6. Lietotāja rediģēšana	25
3.2.1.7. Lietotāja API piekļuves marķiera ģenerēšana	26
3.2.1.8. Lietotāja dzēšana	27
3.2.2. Kategoriju modulis	28
3.2.2.1. Kategoriju saraksta skatīšana	29
3.2.2.2. Kategorijas izveide	29
3.2.2.3. Kategorijas rediģēšana	30
3.2.2.4. Kategorijas dzēšana	31
3.2.3. Preču modulis	32
3.2.3.1. Preces pievienošana	33
3.2.3.2. Preces rediģēšana	34

3.2.3.3.	Preces dzēšana	35
3.2.3.4.	Preču saraksta skatīšana	36
3.2.3.5.	Preču saraksta skatīšana (administratīvā)	37
3.2.3.6.	Preču saraksta skatīšana (noteikta kategorija)	38
3.2.3.7.	Preces informācijas skatīšana	38
3.2.4.	Pasūtījumu modulis	40
3.2.4.1.	Pasūtījuma izveide	41
3.2.4.2.	Iepirkšanās groza saraksta skatīšana	42
3.2.4.3.	Preces pievienošana iepirkšanas grozam	43
3.2.4.4.	Preces noņemšana no iepirkšanas groza	44
3.2.4.5.	Pasūtījuma apstiprināšana	45
3.2.4.6.	Pasūtījuma skatīšana	46
3.2.4.7.	Pasūtījuma rediģēšana	47
3.2.4.8.	Pasūtījuma dzēšana	48
3.2.4.9.	Pasūtījuma izsūtīšana	49
3.2.4.10.	Pasūtījumu saraksta skatīšana	50
3.2.5.	Maksājumu modulis	51
3.2.5.1.	Maksājuma izveide	51
3.2.5.2.	Maksājuma apstiprināšana	52
3.2.6.	Noliktavu modulis	54
3.2.6.1.	Noliktavu saraksta skatīšana	54
3.2.6.2.	Noliktavas izveide	55
3.2.6.3.	Noliktavas rediģēšana	56
3.2.6.4.	Noliktavas dzēšana	57
3.2.7.	Sistēmas iestatījumu modulis	58
3.2.7.1.	Sistēmas iestatījumu saraksta skatīšana	58
3.2.7.2.	Sistēmas iestatījumu rediģēšana	59
3.3.	Nefunkcionālās prasības	60
4.	STANDARTA KĻŪDU APSTRĀDE	62
5.	PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	64
5.1.	Konceptuālais datu bāzes apraksts	64
5.2.	Datu bāzes projektējums	65
5.3.	Datubāzes tabulu projektējums un apraksts	66
5.4.	Daļējs funkciju projektējums	74
5.5.	Daļējs lietotāja saskarņu projektējums	77
6.	TESTĒŠANAS DOKUMENTĀCIJA	85

7.	PROJEKTA ORGANIZĀCIJA	95
8.	KVALITĀTES NODROŠINĀŠANA	96
9.	KONFIGURĀCIJU PĀRVALDĪBA	97
10.	DARBIETILPĪBAS NOVĒRTĒJUMS	98
	SECINĀJUMI	101
	IZMANTOTĀ LITERATŪRA UN AVOTI	102
	PIELIKUMI	103
1.	pielikums. Lietotāja izveides funkcijas vienībtests	103
2.	pielikums. Lietotāja izveides funkcionālais tests	104
3.	pielikums. Pasūtījuma izveides funkcija	105
4.	pielikums. Iepirkšanas groza apskates funkcija	107

APZĪMĒJUMU SARAKSTS

REST API - Sistēmas programmēšanas saskarne.

DB - Datu bāze.

DPD - Datu plūsmu diagramma.

Autentifikācija - process, kurā veic subjekta (lietotāja) identitātes pārbaudi datorsistēmā.

Autorizācija - process, kurā datorsistēma lietotājam nosaka noteiktas pilnvaras un resursus sistēmā.

BCRYPT- datu šifrēšanas funkcija.

CRUD - apzīmē četras pamata funkcijas, ko var veikt ar datiem:

- Create – izveidot;
- Read – nolasīt;
- Update – atjaunināt;
- Delete – izdzēst.

CSRF - vairāku vietņu pieprasījuma viltošana (tulkots no angļu valodas - cross-site request forgery).

HTTPS - hiperteksta transporta protokols HTTP, kurš strādā caur transporta slāņa šifrēšanas mehānismiem TLS/SSL, tādējādi papildinot standarta HTTP sakarus ar TLS/SSL drošības iespējām.

MySQL - relāciju datubāzu vadības sistēma.

PPA – programmatūras projektējuma apraksts.

PPS – programmatūras prasību specifikācija.

Git - atvērtā pirmkoda versiju kontroles sistēma

SQL injekcija – datorsistēmu ielaušanās veids, kurā uzbrucējs ar klienta tiesībām mēģina piekļūt datubāzei, kas ir dotās tīmekļa lapas vai lietotnes pamatā.

PayPal - Starptautiska maksāšanas sistēma.

SDK - bibliotēku kopums, kas tiek izmantots, lai izstrādātu lietotnes/rīkus.

1. IEVADS

1.1. Nolūks

Šī programmatūras prasību specifikācija (PPS) ir izstrādāta un paredzēta informācijas sistēmas (IS) „Pasūtījumu pārvaldības sistēma” prasību aprakstīšanai. Atbilstoši šai specifikācijai izstrādāta sistēma. Šajā prasību specifikācijā tiek formulētas sistēmas prasības un raksturota funkcionalitāte.

1.2. Darbības sfēra

Produkta nosaukums: “Pasūtījumu pārvaldības sistēma”. Programmatūra ļauj elektroniski veikt preču uzskaiti, pasūtījumu noformēšanu dažādām precēm vai pakalpojumiem, atsekot esošu pasūtījumu stāvokli, kā arī administrēt esošos pasūtījumus no uzņēmuma puses. Programma ir domāta e-veikaliem jebkuru preču vai pakalpojumu tirdzniecībai un uzskaitēi. Programma ir orientēta uz preču vai pakalpojumu tirdzniecību privātpersonām. Programma nav paredzēta automātiskai preču uzskaitēi un pasūtījumu izpildei, bet ir paredzēta kā palīgrieks šo uzdevumu veikšanai. Produkts nodrošina RESTful API, kas ļauj to integrēt ar trešo pušu sistēmām.

1.3. Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standartu LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis” [1] prasības un LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai” [2].

1.4. Pārskats

Šis dokuments sastāv no 10 daļām:

- Pirmā (šī) dokumenta daļa satur ievadinformāciju par sistēmu un aprakstošo dokumentu.
- Otrajā daļā tiek aprakstīta produkta perspektīva, lietotāju lomas un to tiesības sistēmā, kā arī sistēmas funkciju saraksts un sadalījums starp lietotāju lomām.

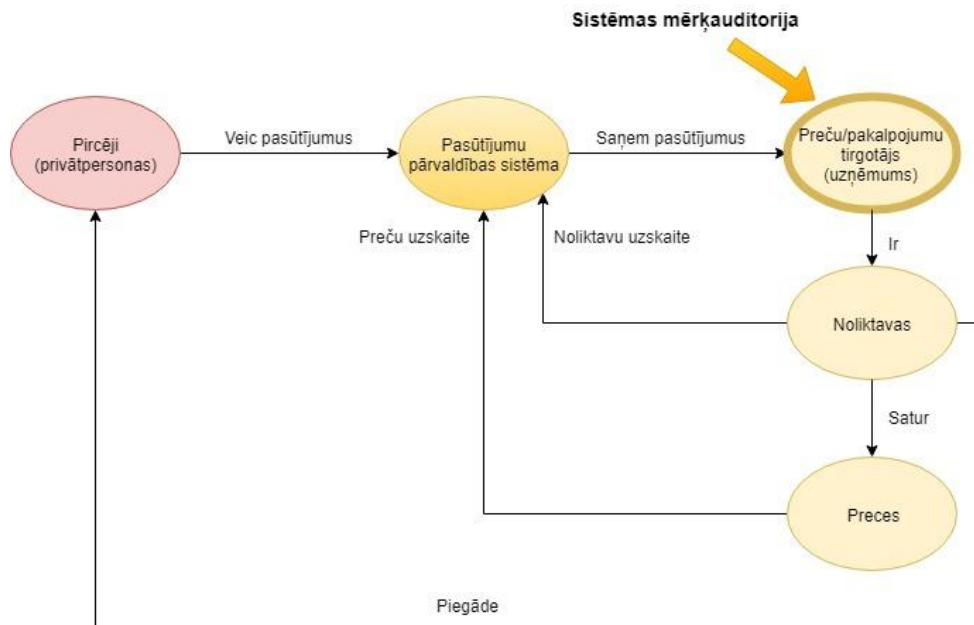
- Trešā daļa ietver funkcionālās un nefunkcionālās prasības. Tā satur detalizētu funkciju aprakstu ar ievaddatiem, apstrādi, un izvadi.
- Ceturtajā daļā tiek aprakstīti sistēmas standarta kļūdu paziņojumu šabloni.
- Piektajā daļā tiek attēloti dažādu sistēmas daļu projektējumi, tādi, kā datu bāzes projektējumi un tabulu apraksts, sarežģītāko funkciju projektējumi un lietotāja saskarne.
- Sestā daļa apskata izmantoto testēšanas procedūru, rīkus, un apraksta izstrādātos automatizētos testus.
- Septītajā daļā tiek izklāstīta projekta organizācija – kādā veidā sistēma tika izstrādāta.
- Astotajā daļā aprakstīts, kā tika nodrošināta projekta kvalitāte.
- Devītā nodaļa dod ieskatu par projekta konfigurāciju pārvaldību.
- Desmitajā nodaļā tiek novērtēta darbietilpība, un tiek pamatota izstrādātās sistēmas atbilstība kvalifikācijas darba prasībām.

2. VISPĀRĒJAIS APRAKSTS

2.1. Produkta perspektīva

Pasūtījumu pārvaldības sistēma ir sistēma, kas domāta uzņēmumiem preču uzskaitēi un tirdzniecībai privātpersonām. Kā redzams attēlā 2.1.1, pasūtījumu pārvaldības sistēma ir veidota preču vai pakalpojumu tirgotājiem pasūtījumu saņemšanai, preču un noliktavu uzskaitēi. Pasūtījuma process ir vienkāršs:

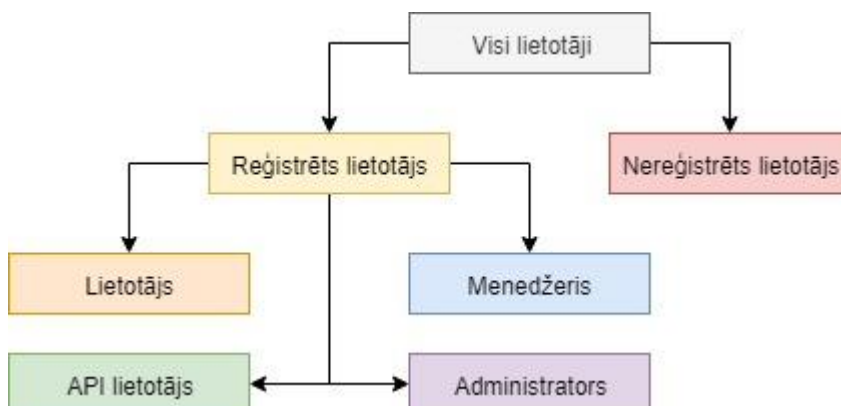
1. Pircējs veic pasūtījumu un pasūtījuma apmaksu, izmantojot pasūtījumu pārvaldības sistēmu.
2. Preču/pakalpojumu tirgotājs sistēmā saņem jaunizveidoto pasūtījumu, un organizē tā piegādi. Sistēma automātiski atjaunina preču skaitu noliktavās.



2.1.1. att. Sistēmas darbības shēma

Aprakstītā sistēma ir pati par sevi neatkarīga, bet to ir iespējams integrēt ar pasūtītāju sistēmām, izmantojot programmatūras API, piemēram noliktavas uzskaites sistēmām. Maksāšanas modulis ir atkarīgs no PayPal sistēmas izmaiņām API SDK bibliotēkās [7] un PayPal sistēmas politikas izmaiņām attiecībā uz to. PayPal sistēmu var aizstāt ar jebkuru citu pasaulē lietojamu maksāšanas sistēmu atbilstoši tirgum, kur tiks izmantota aprakstītā sistēma.

2.2. Sistēmas lietotāju lomas



2.2.1. att. Lietotāju hierarhija

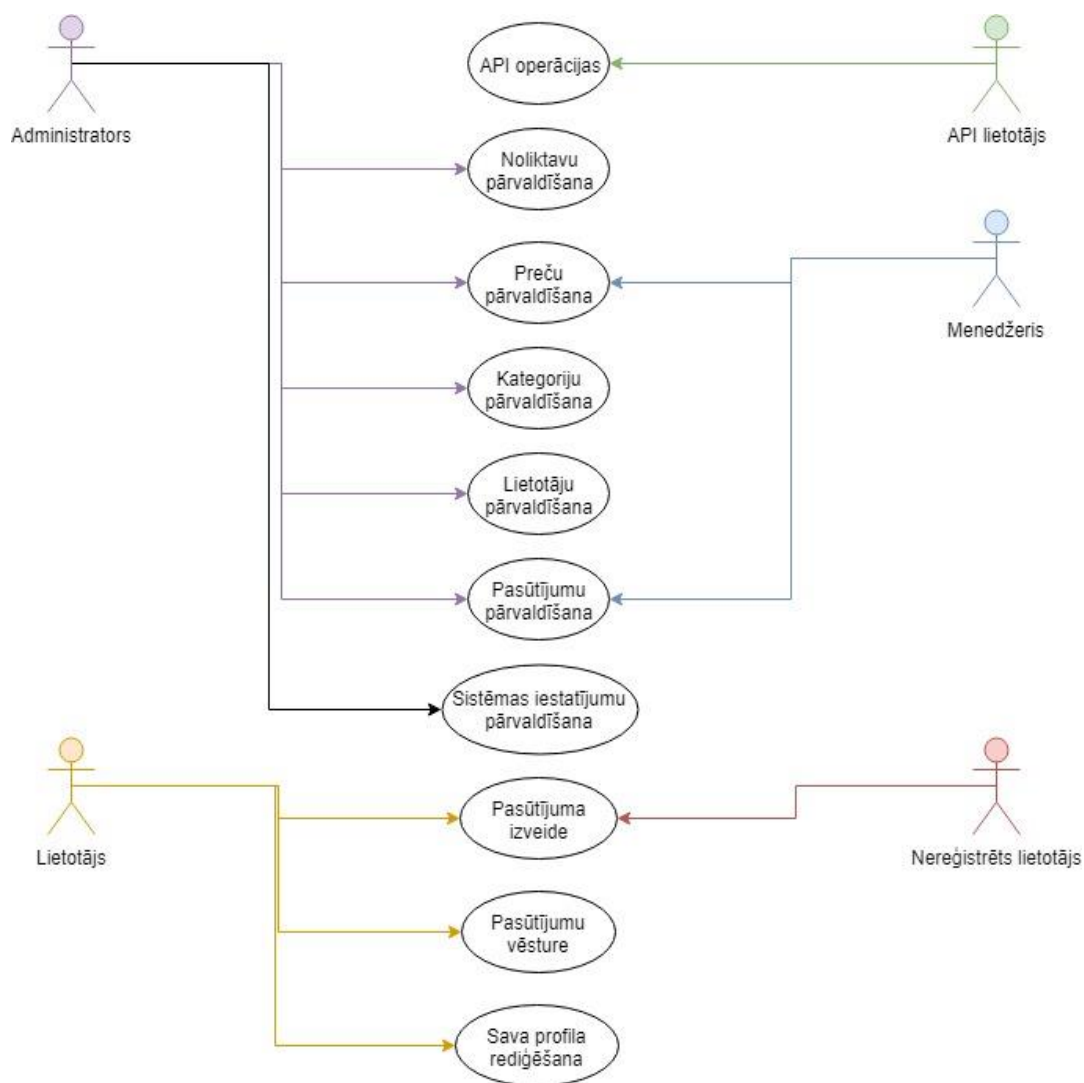
Sistēmā ir noteiktas 4 dažādas lietotāju lomas. Reģistrētiem lietotājiem var piederēt viena vai vairākas lomas, kas nosaka viņu tiesību līmeni sistēmā.

Lomas iedalās:

- Lietotājs;
- API lietotājs;
- Menedžeris;
- Administrators.

Jaunam lietotājam reģistrējoties, tam automātiski tiek nozīmēta “lietotājs” loma, kas ļauj veikt pasūtījumus un redzēt savu pasūtījumu vēsturi un statusu. Pārējās lietotāju lomas var nozīmēt tikai sistēmas lietotājs ar lomu “administrators”. Administratora līmeņa lietotājs var veidot un pārvaldīt lietotājus, noliktavas, kategorijas un sistēmas iestatījumus. Piekļuve sistēmas API resursiem ir tikai lietotājiem ar lomu “API lietotājs”. Menedžera loma ļauj pārvaldīt preces un pasūtījumus.

2.3. Produkta funkcijas



2.3.1. att. Funkcionalitātes sadalījums pa lietotāju lomām

Sistēmā lietotāju lomām ir strikti noteiktas darbības, kuras lietotāji drīkst veikt. Vienam lietotājam var būt vairākas lomas, kas paplašina viņa pilnvaras sistēmā. Darbības veic noteiktas sistēmas funkcijas. Funkcijas ir sadalītas moduļos. Katrs modulis atbild par konkrētu sistēmas objektu, un tas sastāv no funkcijām, kuras mijiedarbojas ar to. Vispārējo funkcionalitātes sadalījumu var redzēt attēlā 2.3.1. Detalizētāku funkciju sarakstu pa moduļiem un tiesības uz šīm funkcijām var novērot, sākot no tabulas 2.3.1. Ar krustiņu atzīmētais nozīmē, ka funkcija ir pieejama norādītajam lietotājam, ar mīnusu – nav pieejama.

2.3.1. Tabula

Lietotāju moduļa funkciju saraksts

Funkcijas ID	Nosaukums	Administrators	Menedžeris	API lietotājs	Parasts lietotājs	Neregistrēts lietotājs
UserRegister	Lietotāja reģistrācija	-	-	-	-	X
UserLogin	Lietotāja autentifikācija	X	X	X	X	-
UserPasswordReset	Lietotāja paroles atjaunošana	X	X	X	X	-
UserIndex	Lietotāju saraksta skatīšana	X	-	-	-	-
UserCreate	Lietotāja izveide	X	-	-	-	-
UserEdit	Lietotāja rediģēšana	X	-	-	-	-
UserGenerateToken	Lietotāja API piekļuves marķiera ģenerēšana	X	-	X	-	-
UserDelete	Lietotāja dzēšana	X	-	-	-	-

2.3.2. Tabula

Kategoriju moduļa funkciju saraksts

Funkcijas ID	Nosaukums	Administrators	Menedžeris	API lietotājs	Parasts lietotājs	Neregistrēts lietotājs
CategoryIndex	Kategoriju saraksta skatīšana	X	-	-	-	-
CategoryCreate	Kategorijas izveide	X	-	-	-	-
CategoryEdit	Kategorijas rediģēšana	X	-	-	-	-
CategoryDelete	Kategorijas dzēšana	X	-	-	-	-

2.3.3. Tabula

Preču moduļa funkciju saraksts

Funkcijas ID	Nosaukums	Administrators	Menedžeris	API lietotājs	Parasts lietotājs	Neregistrēts lietotājs
ItemCreate	Preces pievienošana	X	X	-	-	-
ItemShow	Preces skatīšana	X	X	X	X	X
ItemEdit	Preces rediģēšana	X	X	-	-	-
ItemRemove	Preces dzēšana	X	X	-	-	-
ItemIndex	Preču saraksta skatīšana	X	X	X	X	X
ItemIndex_Admin	Preču saraksta skatīšana (administratīvā)	X	X	-	-	-
ItemIndex_Category	Preču saraksta skatīšana (noteikta kategorija)	X	X	X	X	X

2.3.4. Tabula

Pasūtījumu moduļa funkciju saraksts

Funkcijas ID	Nosaukums	Administrators	Menedžeris	API lietotājs	Parasts lietotājs	Neregistrēts lietotājs
CartIndex	Iepirkšanās groza saraksta skatīšana	X	X	X	X	X
CartAdd	Preces pievienošana iepirkšanas grozam	X	X	X	X	X
CartRemove	Preces noņemšana no iepirkšanas groza	X	X	X	X	X
OrderCreate	Pasūtījuma izveide	X	X	X	X	X
OrderConfirm	Pasūtījuma apstiprināšana	X	X	X	X	X

OrderShow	Pasūtījuma skatīšana	X	X	X	X	-
OrderEdit	Pasūtījuma rediģēšana	X	X	-	-	-
OrderRemove	Pasūtījuma dzēšana	X	X	-	-	-
OrderShip	Pasūtījuma izsūtīšana	X	X	-	-	-
OrderIndex	Pasūtījumu saraksta skatīšana	X	X	-	-	-

2.3.5. Tabula

Maksājumu moduļa funkciju saraksts

Funkcijas ID	Nosaukums	Administrators	Menedžeris	API lietotājs	Parasts lietotājs	Neregistrēts lietotājs
PaymentCreate	Maksājuma izveide	X	X	X	X	X
PaymentConfirm	Maksājuma apstiprināšana	X	X	X	X	X

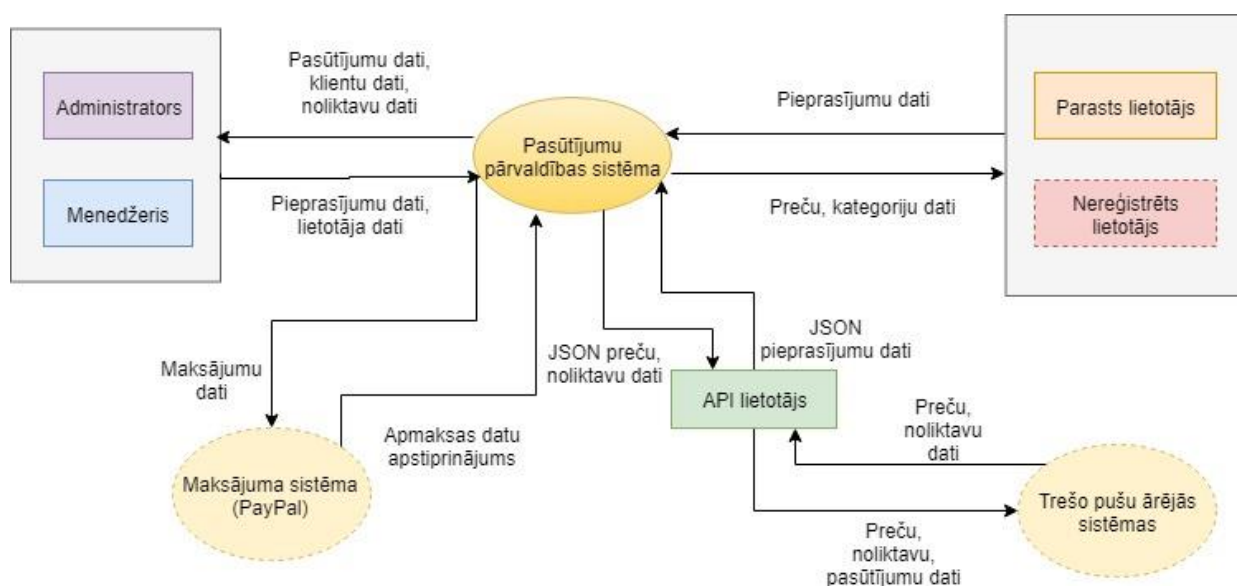
2.3.6. Tabula

Noliktavu moduļa funkciju saraksts

Funkcijas ID	Nosaukums	Administrators	Menedžeris	API lietotājs	Parasts lietotājs	Neregistrēts lietotājs
WarehouseIndex	Noliktavu saraksta skatīšana	X	-	-	-	-
WarehouseCreate	Noliktavas izveide	X	-	-	-	-
WarehouseEdit	Noliktavas rediģēšana	X	-	-	-	-
WarehouseRemove	Noliktavas dzēšana	X	-	-	-	-

Sistēmas iestatījumu moduļa funkciju saraksts

Funkcijas ID	Nosaukums	Administrators	Menedžeris	API lietotājs	Parasts lietotājs	Neregistrēts lietotājs
SettingsIndex	Sistēmas iestatījumu saraksta skatīšana	X	-	-	-	-
SettingsUpdate	Sistēmas iestatījumu rediģēšana	X	-	-	-	-



2.3.2. att. 0. līmeņa datu plūsmas diagramma

Pasūtījumu pārvaldības sistēma izmanto maksājumu sistēmu PayPal, lai nodrošinātu maksājumu izpildi par pasūtījumiem. Komunikācija ar PayPal notiek, izmantojot PayPal RESTful API. No pasūtījumu pārvaldības sistēmas tiek sagatavoti maksājumu pieprasījumi, kas tiek nosūtīti uz maksājumu sistēmu, bet no maksājumu sistēmas tiek saņemti maksājumu apmaksas dati ar apstiprinājumu vai noraidījumu.

Tāpat pasūtījumu pārvaldības sistēma var komunicēt ar trešo pušu sistēmām, izmantojot API. Aprakstītai sistēmai ir izstrādāta API saskarne, kas dublē visu sistēmas funkcionalitāti, tāpēc tā netiek atsevišķi aprakstīta. To var izmantot, piemēram, lai ārēja noliktavas programmatūra automātiski atjaunina datus pasūtījumu pārvaldības sistēmā par preču daudzumu noliktavās, vai arī automātisku preču pievienošanu/atjaunināšanu no ārējiem avotiem. Visa komunikācija notiek,

izmantojot API saskarni, autentificējoties kā lietotājs ar API lomu un iekļaujot pieprasījuma virsrakstā piekļuves marķieri.

2.4. Vispārējie ierobežojumi

- Darbam ar sistēmu nepieciešama jebkura viedierīce ar piekļuvi internetam.
- Sistēmas darbība ir atbalstīta uz četrām populārākajām pārlūkprogrammām: Google Chrome, Mozilla Firefox, Microsoft Edge, Opera.
- Sistēma izmanto šifrēšanu datu pārraidē, jo paredzēts, ka tā būs pieejama no publiskā tīkla.
- Sistēmas lietotāju un pakalpojumu datu uzglabāšana. Sistēmai ir spējīga glabāt sensitīvus sistēmas lietotāju (klientu) datus. Uzglabātās lietotāju paroles glabājas datu bāzē šifrētā veidā, kā arī veic regulāras rezerves kopijas.

2.5. Pieņēmumi un atkarības

Pieņēmumi faktoriem, kas spēj ietekmēt sistēmas programmatūras prasības:

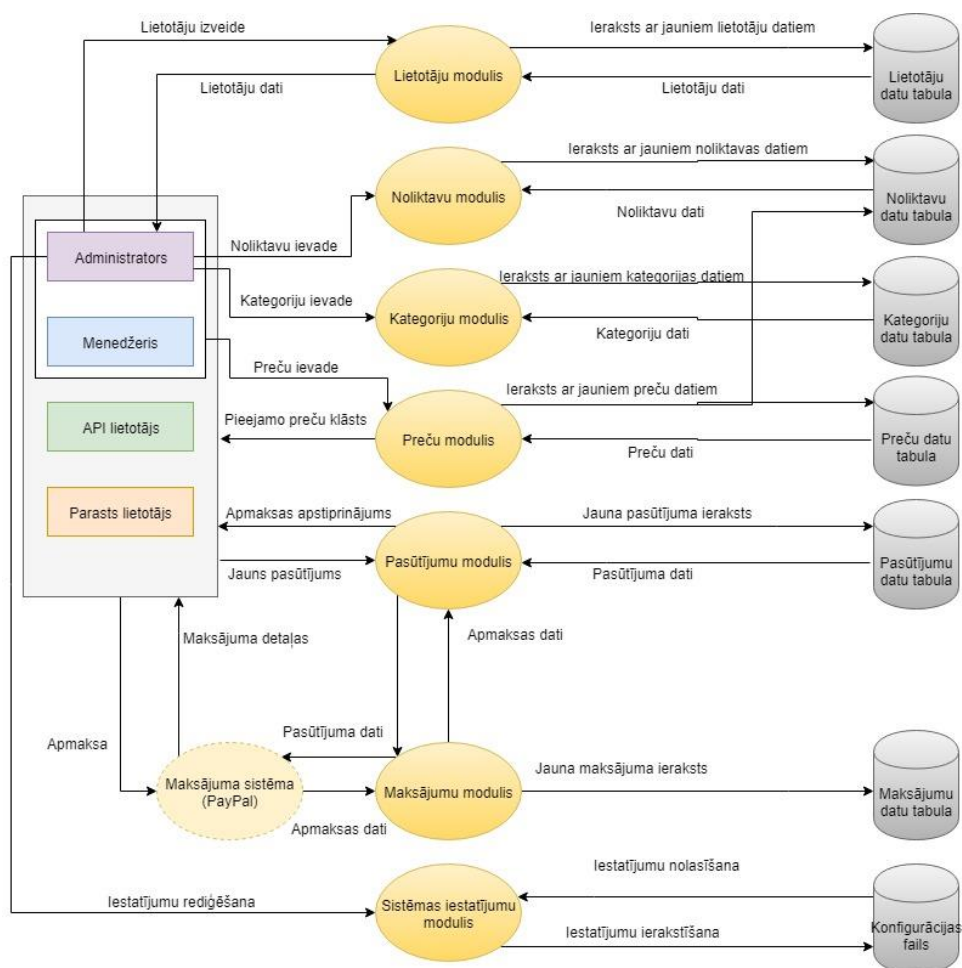
- Sistēmas lietotāji programmatūras darbināšanai izmantos aktuālas operētājsistēmu un pārlūkprogrammu versijas, jo, izmantojot vecās operētājsistēmu vai pārlūkprogrammu versijas, tās var ietekmēt programmatūras darbību un to vizuālo izskatu.

3. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

3.1. Funkcionālās prasības

Visas funkcionālās prasības ir aprakstītas atbilstoši sistēmas funkcionalitātei un funkciju ievades dati atbilst nepieciešamajiem ievades datiem no sistēmas viedokļa. Funkciju aprakstā nav norādīts, kādām lietotāju lomām ir tiesības izmantot šo funkciju – tas ir norādīts attiecīgā modulī, sākot no tabulas 2.3.1. Katrai funkcijai tiek piešķirti unikāli identifikatori, lai varētu vieglāk identificēt, kura funkcija tiek izsaukta kādā konkrētā gadījumā. Funkciju sadalījumu pēc moduļiem var apskatīt 2.3.1. attēlā.

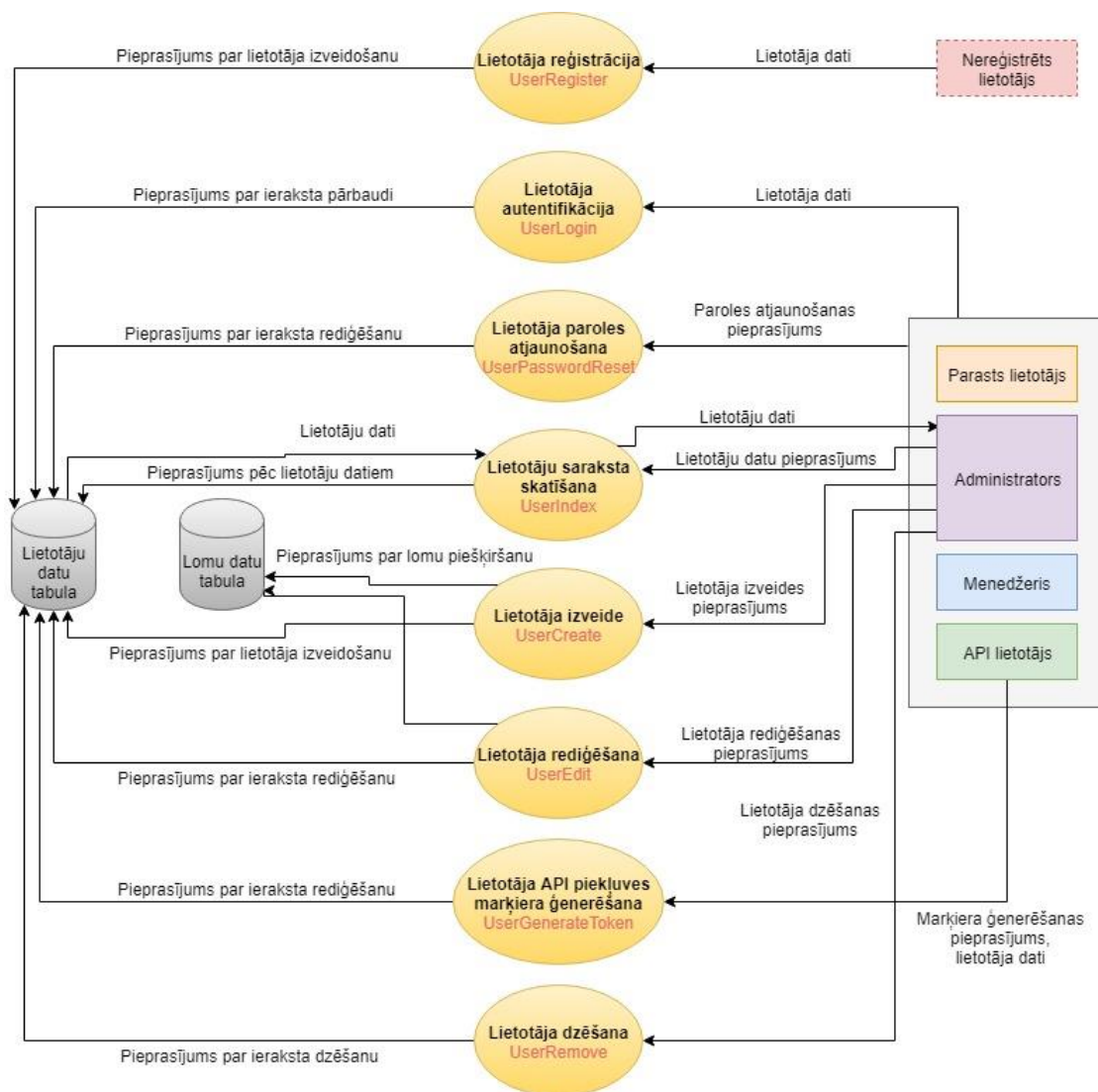
3.2. Funkciju sadalījums pa moduļiem/komponentiem



3.2.1. att. 1. Ieņēmumu datu plūsmas diagramma

Kā var novērot 3.2.1. attēlā, sistēma tiek sadalīta septiņos moduļos - lietotāju modulis, kas atbild par lietotāju pārvaldību, noliktavu modulis, kas atbild par noliktavu pārvaldību, kategoriju modulis, kas atbild par kategoriju pārvaldību, preču modulis, kas atbild par preču uzskaiti un cenām, pasūtījumu modulis, kas atbild par pasūtījumu izveidi un pārvaldīšanu, maksājumu modulis, kas atbild par maksājumu veikšanu, un sistēmas iestatījumu modulis, kas atbild par pasūtījumu pārvaldības sistēmas parametriem. Katrs modulis veic ierakstus savā datu tabulā (sistēmas iestatījumu modulis - konfigurācijas failā), kā arī saistīto moduļu tabulās. Detalizētāka informācija atrodama attiecīgo funkciju aprakstā.

3.2.1. Lietotāju modulis



3.2.1.1. att. 2. Īmeņa datu plūsmas diagramma, lietotāju modulis

3.2.1.1. attēlā ir grafiski attēlots lietotāju modulis, visas tā funkcijas, un to funkciju datu plūsmas. Lietotāja moduļu detalizēto funkciju aprakstu var apskatīt, sākot no sadaļas 3.2.1.1.

3.2.1.1. Lietotāja reģistrācija

Identifikators
UserRegister
Ievads
Funkcija paredzēta jauna lietotāja reģistrācijai
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none">● pilnais vārds. Simbolu virkne, maksimālais garums - 100 simboli;● Vēlamais lietotājvārds. Unikāla simbolu virkne, minimālais garums - 5 simboli, maksimālais garums - 30 simboli;● E-pasta adrese. Unikāla simbolu virkne, e-pasta adrese, maksimālais garums - 50 simboli;● Telefona numurs; Simbolu virkne, maksimālais garums - 15 simboli.● Parole. Simbolu virkne, minimālais garums - 8 simboli. Jāsatur vismaz vienu lielu burtu, vienu mazu burtu, vismaz vienu ciparu un vismaz vienu speciālo simbolu. <p>Neobligātie lauki:</p> <ul style="list-style-type: none">● Adrese. Simbolu virkne, minimālais garums - 10 simboli.
Apstrāde
<p>Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Pārbauda, vai ievadītā e-pasta adrese vai lietotājvārds jau neeksistē sistēmā. Ja visi pārbaudījumi ir veiksmīgi, funkcija izveido lietotāju, veic lietotāja autentifikāciju, un pāradresē uz sākuma lapu.</p>
Izvade

- Veiksmīga funkcijas izpilde - lietotāju pāradresē uz sākuma lapu ar paziņojumu nr. 1.
- Neveiksmīga funkcijas izpilde - atgriežas pie reģistrācijas datu ievades formas, un izvada attiecīgo kļūdas paziņojumu.

Paziņojumi

1. Welcome to the [Sistēmas nosaukums], [Lietotāja pilnais vārds]!

3.2.1.2. Lietotāja autentifikācija

Identifikators

UserLogin

Ievads

Funkcija paredzēta, lai lietotājs varētu autentificēties sistēmā ar saviem datiem

Ievade

Obligātie lauki un to pieļaujamās vērtības:

- Lietotājvārds.
- Parole.

Apstrāde

Pārbauda:

- vai norādītais lietotājvārds eksistē sistēmas datubāzē;
- vai norādītā parole sakrīt ar paroli norādītajam lietotājvārdam datubāzē.

Ja visi pārbaudījumi ir veiksmīgi, funkcija autentificē lietotāju, un pāradresē uz sākuma lapu.

Izvade

- Veiksmīga funkcijas izpilde - notiek lietotāja autentifikācija, un lietotājs tiek pāradresēts uz sākuma lapu.
- Neveiksmīga funkcijas izpilde - atgriežas pie autentifikācijas datu ievades formas, un

izvada kļūdas paziņojumu nr. 1.
Paziņojumi
1. These credentials do not match our records.

3.2.1.3. Lietotāja paroles atjaunošana

Identifikators
UserPasswordReset
Ievads
Funkcija paredzēta, lai lietotājs varētu atjaunot sava profila paroli gadījumā, ja tā ir nozaudēta vai piemirsta.
Ievade
Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none"> e-pasta adrese. E-pasta adrese, maksimālais garums - 50 simboli.
Apstrāde
Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Pārbauda, vai ievadītā e-pasta adrese eksistē sistēmas datubāzē un pieder kādam no lietotājiem. Ja visi pārbaudījumi ir veiksmīgi, nosūta paroles atjaunošanas saiti uz norādīto e-pasta adresi.
Izvade
<ul style="list-style-type: none"> Veiksmīga funkcijas izpilde - uz norādīto e-pasta adresi tiek aizsūtīta e-pasta vēstule ar lietotāja paroles atjaunošanas instrukcijām, un lietotājam tiek izvadīts paziņojums nr. 1. Neveiksmīga funkcijas izpilde - atgriežas pie paroles atjaunošanas datu ievades formas, un izvada kļūdas paziņojumu nr. 2.

Paziņojumi

1. We have e-mailed your password reset link!
2. We can't find a user with that e-mail address.

3.2.1.4. Lietotāju saraksta skatīšana

Identifikators

UserIndex

Ievads

Funkcija paredzēta administratoram, lai varētu apskatīt reģistrēto lietotāju sarakstu.

Ievade

Nav

Apstrāde

Atlasa visus reģistrētos lietotājus no datubāzes un padod datus uz skatu.

Izvade

Veiksmīga funkcijas izpilde - Tiek atlasīti visi reģistrētie lietotāji un attēloti tabulas veidā. Ja neeksistē neviens lietotājs izvada paziņojumu nr. 1.

Paziņojumi

1. No items to display

3.2.1.5. Lietotāja izveide

Identifikators
UserCreate
Ievads
Administrators funkcija, kas paredzēta jauna sistēmas lietotāja izveidei
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none">• pilnais vārds. Simbolu virkne, maksimālais garums - 100 simboli;• vēlamais lietotājvārds. Unikāla simbolu virkne, minimālais garums - 5 simboli, maksimālais garums - 30 simboli;• e-pasta adrese. Unikāla simbolu virkne, maksimālais garums - 50 simboli;• telefona numurs; Simbolu virkne, maksimālais garums - 15 simboli.• parole. Simbolu virkne, minimālais garums - 8 simboli. Jāsatur vismaz vienu lielu burtu, vienu mazu burtu, vismaz vienu ciparu un vismaz vienu speciālo simbolu.• loma. Atzīmes lodziņš. Var sastāvēt no vairākām iespējamām vērtības: admin, api, user. <p>Neobligātie lauki:</p> <ul style="list-style-type: none">• adrese. Simbolu virkne, minimālais garums - 10 simboli.
Apstrāde
<p>Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Pārbauda:</p> <ul style="list-style-type: none">• vai ievadītā e-pasta adrese vai lietotājvārds jau neeksistē sistēmā;• Vai ievadītās lomas vērtības eksistē sistēmā. <p>Ja visi pārbaudījumi ir veiksmīgi, jauns lietotājs tiek izveidots un ierakstīts datubāzē.</p>
Izvade
<ul style="list-style-type: none">• Veiksmīga funkcijas izpilde - izveido lietotāju, un pāradresē uz lietotāju sarakstu lapu ar

<p>paziņojumu nr. 1.</p> <ul style="list-style-type: none"> • Neveiksmīga funkcijas izpilde - atgriežas pie lietotāja datu ievades formas, un izvada attiecīgo kļūdas paziņojumu.
Paziņojumi
1. User created succesfully!

3.2.1.6. Lietotāja rediģēšana

Identifikators
UserEdit
Ievads
Funkcija paredzēta lietotāja datu rediģēšanai
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none"> • Rediģējamā lietotāja ID; • pilnais vārds. Simbolu virkne, maksimālais garums - 100 simboli; • lietotājvārds. Unikāla simbolu virkne, minimālais garums - 5 simboli, maksimālais garums - 30 simboli (tikai administratora skatam); • e-pasta adrese. Unikāla simbolu virkne, maksimālais garums - 50 simboli; • telefona numurs; Simbolu virkne, maksimālais garums - 15 simboli. • parole. Simbolu virkne, minimālais garums - 8 simboli. • loma. Atzīmes lodziņš. Var sastāvēt no vairākām iespējamām vērtībām: admin, api, user. (tikai administratora skatam) <p>Neobligātie lauki:</p> <ul style="list-style-type: none"> • adrese. Simbolu virkne, minimālais garums - 10 simboli.

Apstrāde
<p>Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Pārbauda, vai ievadītās lomas vērtības eksistē sistēmā.</p> <p>Ja visi pārbaudījumi ir veiksmīgi, atrod lietotāju pēc ID un atjaunina ierakstu datubāzē.</p>
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - notiek lietotāja datu rediģēšana, un lietotājs tiek pārdresēts uz lietotāju sarakstu lapu (administrators), vai profila lapu (parasts lietotājs) ar paziņojumu nr. 1. • Neveiksmīga funkcijas izpilde - atgriežas pie lietotāja datu rediģēšanas formas, un izvada attiecīgo kļūdas paziņojumu. Ja lietotājs pēc ID nav atrasts, izvada kļūdas paziņojumu nr. 2.
Paziņojumi
<ol style="list-style-type: none"> 1. User profile edited succesfully! 2. 404 Not Found.

3.2.1.7. Lietotāja API piekļuves marķiera ģenerēšana

Identifikators
UserGenerateToken
Ievads
Funkcija paredzēta, lai lietotājs varētu uzģenerēt piekļuves marķieri datu piekļuvei, izmantojot API.
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none"> • Lietotāja ID

Apstrāde
Atrod lietotāju pēc ID. Pārbauda, vai norādītajam lietotājam ir piešķirta loma “api”. Ja tā ir piešķirta, uzģenerē simbolu virkni no 60 simboliem, nošifrē, izmantojot SHA256 algoritmu un saglabā datubāzē, pārrakstot iepriekšējo simbolu virkni, ja tāda ir bijusi. Ja “api” loma nav piešķirta, izmet kļūdas paziņojumu.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - tiek uzģenerēts API piekļuves marķieris un attēlots lietotāja rediģēšanas lapā. • Neveiksmīga funkcijas izpilde - izmet kļūdas paziņojumu nr. 1.
Paziņojumi
1. 404 Not Found.

3.2.1.8. Lietotāja dzēšana

Identifikators
UserDelete
Ievads
Funkcija paredzēta sistēmas lietotāja dzēšanai
Ievade
Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none"> • Lietotāja ID
Apstrāde
Sistēma atrod lietotāju pēc ID un dzēš no datubāzes.

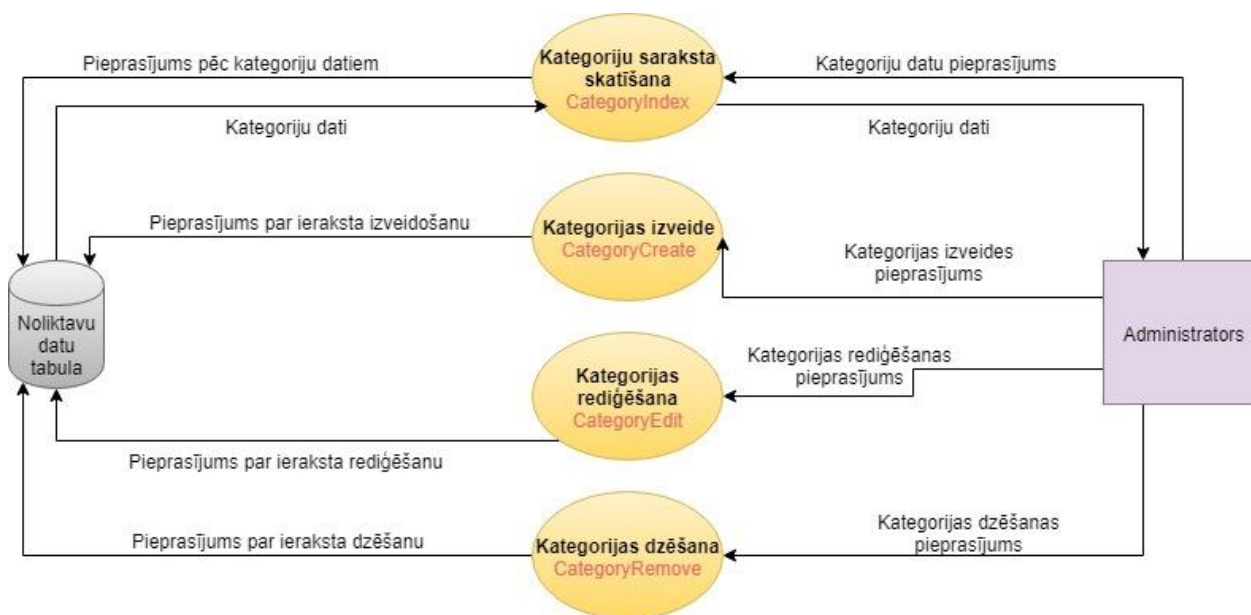
Izvade

- Veiksmīga funkcijas izpilde - lietotājs tiek dzēsts un administrators tiek pārdresēts uz lietotāju saraksta skatu ar paziņojumu nr. 1.
- Neveiksmīga funkcijas izpilde - ja sistēma neatrod lietotāju pēc ID, izmet kļūdas paziņojumu nr. 2.

Paziņojumi

1. User succesfully deleted!
2. 404 | Not Found

3.2.2. Kategoriju modulis



3.2.2.1. att. 2. Īmeņa datu plūsmas diagramma, kategoriju modulis

3.2.2.1. attēlā ir grafiski attēlots kategoriju modulis, visas tā funkcijas, un to funkciju datu plūsmas. Kategoriju moduļa detalizēto funkciju aprakstu var apskatīt, sākot no sadaļas 3.2.2.1.

3.2.2.1. Kategoriju saraksta skatīšana

Identifikators
CategoryIndex
Ievads
Funkcija paredzēta kategoriju saraksta attēlošanai
Ievade
Nav
Apstrāde
Atlasa visas kategorijas no datubāzes.
Izvade
Attēlo visas kategorijas tabulas veidā. Ja neeksistē neviena kategorija, izvada paziņojumu nr. 1.
Paziņojumi
1. No items to display.

3.2.2.2. Kategorijas izveide

Identifikators
CategoryCreate
Ievads
Funkcija paredzēta, lai izveidotu jaunu preču kategoriju
Ievade

<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none"> • Nosaukums. Unikāla simbolu virkne. Maksimālais virknes garums - 15 simboli • Pozīcija. Unikāla skaitliska vērtība. Maksimālā vērtība - 9.
Apstrāde
<p>Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Ja dati ir pareizi, izveido jaunu kategorijas objektu, aizpilda ar ievadītajām vērtībām un saglabā datubāzē.</p>
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - atgriež kategoriju saraksta skatu ar paziņojumu nr. 1. • Neveiksmīga funkcijas izpilde - atgriežas pie kategorijas datu ievades formas, un izvada attiecīgo kļūdas paziņojumu.
Paziņojumi
<ol style="list-style-type: none"> 1. Category successfully created!

3.2.2.3. Kategorijas rediģēšana

Identifikators
CategoryEdit
Ievads
Funkcija paredzēta, lai rediģētu eksistējošas preču kategorijas datus
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none"> • Rediģējamās kategorijas ID. • Nosaukums. Unikāla simbolu virkne. Maksimālais virknes garums - 15 simboli • Pozīcija. Unikāla skaitliska vērtība. Maksimālā vērtība - 9.

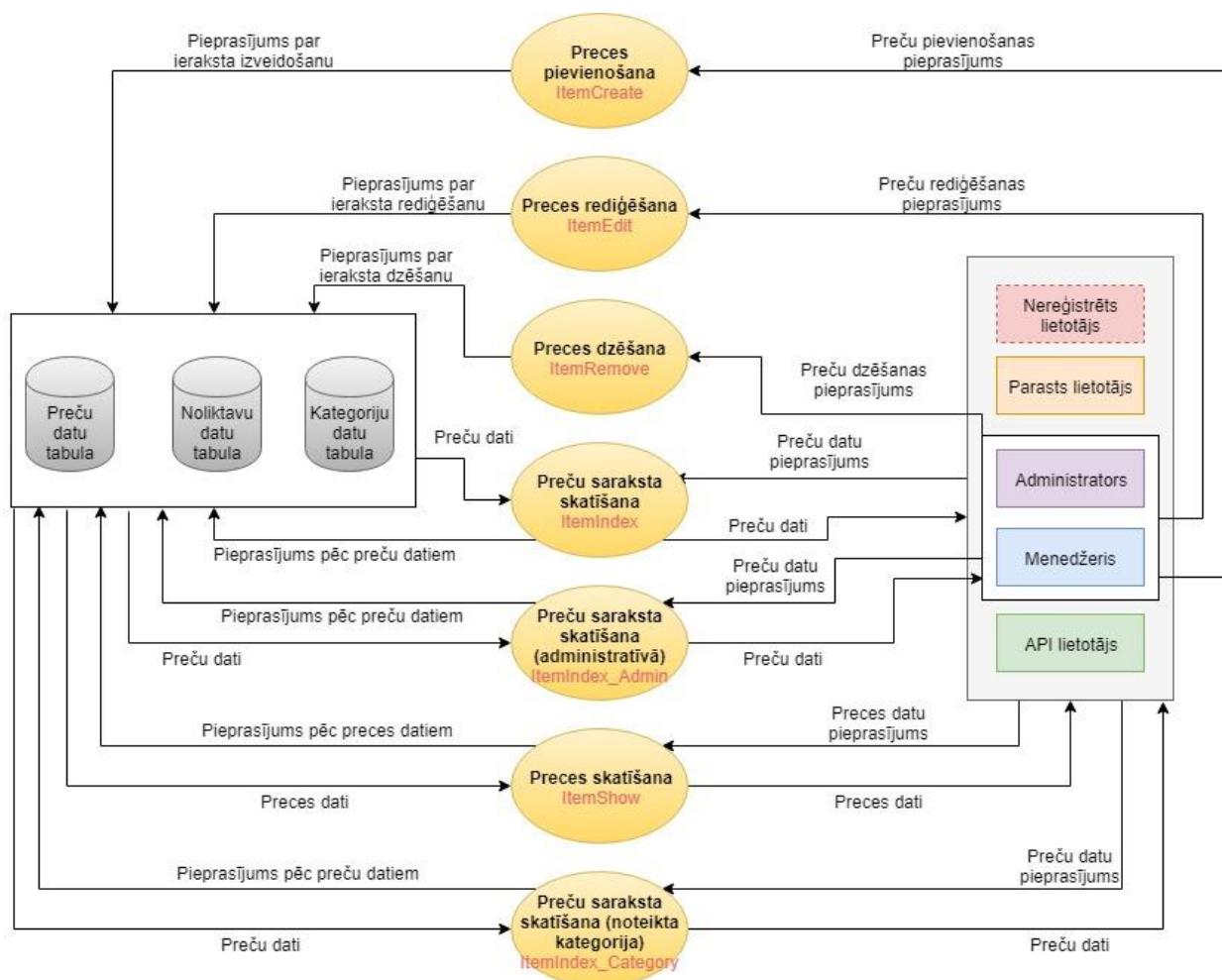
Apstrāde
Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Ja dati ir pareizi, atrod rediģējamo kategorijas objektu pēc ID, un atjaunina ierakstu datubāzē.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - atgriež kategoriju saraksta skatu ar paziņojumu nr. 1. • Neveiksmīga funkcijas izpilde - atgriežas pie kategorijas datu ievades formas, un izvada attiecīgo kļūdas paziņojumu. Ja kategorija pēc norādītā ID nav atrasta, attēlo kļūdas paziņojumu nr. 2.
Paziņojumi
<ol style="list-style-type: none"> 1. Category was successfully edited! 2. 404 Not Found

3.2.2.4. Kategorijas dzēšana

Identifikators
CategoryDelete
Ievads
Funkcija paredzēta preču kategorijas dzēšanai
Ievade
Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none"> • Kategorijas ID
Apstrāde
Sistēma atrod kategoriju pēc ID un dzēš no datubāzes.

Izvide
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - kategorija tiek dzēsta un administrators tiek pārdresēts uz kategoriju saraksta skatu ar paziņojumu nr. 1. • Neveiksmīga funkcijas izpilde - izmet kļūdas paziņojumu nr. 2.
Paziņojumi
<ol style="list-style-type: none"> 1. Successfully deleted category! 2. 404 Not Found

3.2.3. Preču modulis



3.2.3.1. att. 2. Īmeņa datu plūsmas diagramma, preču modulis

3.2.3.1. attēlā ir grafiski attēlots preču modulis, visas tā funkcijas, un to funkciju datu plūsmas. Preču moduļu detalizēto funkciju aprakstu var apskatīt, sākot no sadaļas 3.2.3.1.

3.2.3.1. Preces pievienošana

Identifikators
ItemCreate
Ievads
Funkcija paredzēta jaunas preces pievienošanai
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none">• Nosaukums. Simbolu virkne. Minimālais virknes garums - 3 simboli. Maksimālais virknes garums - 50 simboli.• Apraksts. Simbolu virkne. Minimālais virknes garums - 15 simboli.• Realizācijas cena. Skaitliska decimāla vērtība ar diviem cipariem aiz komatiem.• Iepirkuma cena. Skaitliska decimāla vērtība ar diviem cipariem aiz komatiem.• Preces bildes. Attēlu faila/failu augšupielāde. Pieļaujamie attēlu formāti: jpeg, png, jpg, gif, svg. Pieļaujamais 1 attēla izmērs - 2048 kb. <p>Neobligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none">• Kategorija. Skaitlisks ID. Kategorijai ar norādīto ID ir jāeksistē.• Skaits katrā noliktavā. Skaitliska vērtība.• Piegādes maksas no noliktavām. Skaitliska decimāla vērtība ar diviem cipariem aiz komatiem.
Apstrāde
Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Ja pārbaudes ir izturētas, izveido jaunu preces objektu, ievada datus. Augšupielādētos failus pārvieto uz publisko attēlu direktoriju un failus nosauc sekojošā formātā

- '%datums%_augšupielādētā_faila_nosaukums', un asociē bildes ar precī datubāzē. Izveido preces relāciju ar katru noliktavu datubāzē un ievada datus par preces skaitu un piegādes maksu katrā noliktavā.

Izvade

- Veiksmīga funkcijas izpilde - notiek preces izveide, un lietotājs tiek pāradresēts uz preču saraksta skatu ar paziņojumu nr. 1.
- Neveiksmīga funkcijas izpilde - atgriežas pie preces datu ievades formas, un izvada attiecīgo kļūdas paziņojumu.

Paziņojumi

1. Successfully created item!

3.2.3.2. Preces rediģēšana

Identifikators

ItemEdit

Ievads

Funkcija paredzēta esošās preces datu rediģēšanai

Ievade

Obligātie lauki un to pieļaujamās vērtības:

- Rediģējamās preces ID.
- Nosaukums. Simbolu virkne. Minimālais virknes garums - 3 simboli. Maksimālais virknes garums - 50 simboli.
- Apraksts. Simbolu virkne. Minimālais virknes garums - 15 simboli.
- Realizācijas cena. Skaitliska decimāla vērtība ar diviem cipariem aiz komatiem.
- iepirkuma cena. Skaitliska decimāla vērtība ar diviem cipariem aiz komatiem.

Neobligātie lauki un to pieļaujamās vērtības:

<ul style="list-style-type: none"> • Kategorija. Skaitlisks ID. Kategorijai ar norādīto ID ir jāeksistē. • Skaits katrā noliktavā. Skaitliska vērtība. • Piegādes maksas no noliktavām. Skaitliska decimāla vērtība ar diviem cipariem aiz komatiem.
Apstrāde
Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Ja pārbaudes ir izturētas, atrod preces objektu pēc ID, atjaunina datus par preces skaitu un piegādes maksu katrā noliktavā datubāzē.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - notiek preces izveide, un lietotājs tiek pāradresēts uz preču saraksta skatu ar paziņojumu nr. 1. • Neveiksmīga funkcijas izpilde - atgriežas pie preces datu ievades formas, un izvada attiecīgo kļūdas paziņojumu. Ja prece pēc ID nav atrasta, attēlo kļūdas paziņojumu nr. 2.
Paziņojumi
<ol style="list-style-type: none"> 1. Successfully edited item! 2. 404 Not Found.

3.2.3.3. Preces dzēšana

Identifikators
ItemRemove
Ievads
Funkcija paredzēta preces dzēšanai
Ievade
Obligātie lauki un to pieļaujamās vērtības:

<ul style="list-style-type: none"> • Preces ID
Apstrāde
Pārbauda, vai datubāzē eksistē prece ar norādīto ID un ielādē to. Pēc definētajām relācijām nodzēš attēlu failus un izdzēš preci no datubāzes.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - notiek preces un attēlu failu dzēšana. Lietotājs tiek pāradresēts uz preču saraksta skatu ar paziņojumu nr 1. • Neveiksmīga funkcijas izpilde - izmet kļūdas paziņojumu nr 2.
Paziņojumi
<ol style="list-style-type: none"> 1. Successfully deleted item! 2. 404 Not Found

3.2.3.4. Preču saraksta skatīšana

Identifikators
ItemIndex
Ievads
Funkcija paredzēta preču saraksta attēlošanai
Ievade
Nav
Apstrāde
Atlasa visas preces, kas eksistē datubāzē.
Izvade

<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - atgriež visu pieejamo preču skatu
Paziņojumi
Nav.

3.2.3.5. Preču saraksta skatīšana (administratīvā)

Identifikators
ItemIndex_Admin
Ievads
Funkcija paredzēta preču saraksta attēlošanai administratīvo darbību veikšanai
Ievade
Nav
Apstrāde
Atlasa visas preces, kas eksistē datubāzē.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - atgriež visu pieejamo preču skatu tabulas veidā ar pārvaldības darbībām. Ja neeksistē neviena prece, attēlo paziņojumu nr. 1.
Paziņojumi
1. No items to display.

3.2.3.6. Preču saraksta skatīšana (noteikta kategorija)

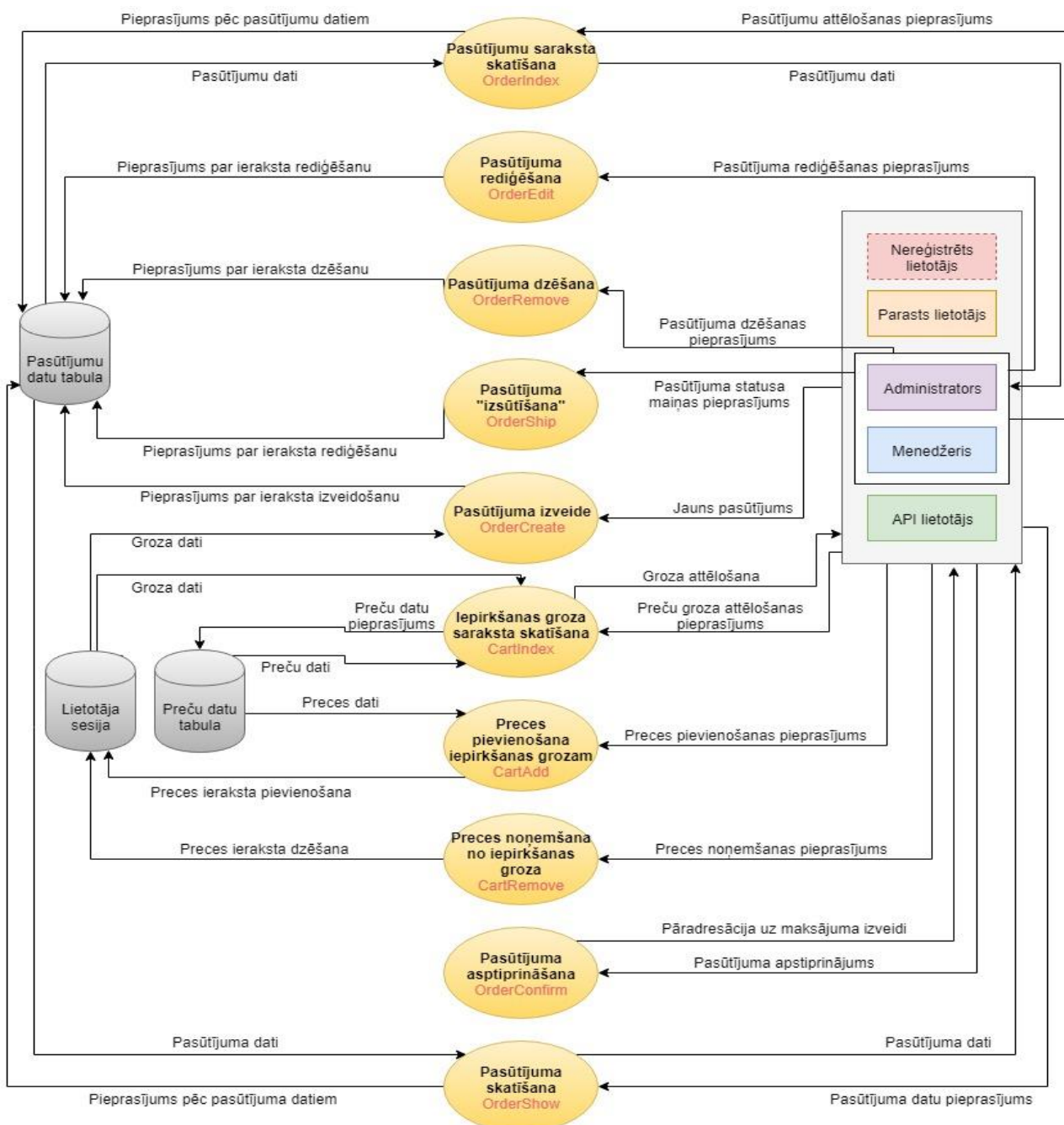
Identifikators
ItemIndex_Category
Ievads
Funkcija paredzēta konkrētas kategorijas preču saraksta attēlošanai lietotājam.
Ievade
Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none">• Kategorijas ID
Apstrāde
Atlasa visas preces, kas eksistē datubāzē ar ievadīto kategorijas ID
Izvade
<ul style="list-style-type: none">• Veiksmīga funkcijas izpilde - atgriež pieprasītās kategorijas pieejamo preču skatu. Ja neeksistē neviena prece, attēlo paziņojumu nr. 1.
Paziņojumi
1. No items to display.

3.2.3.7. Preces informācijas skatīšana

Identifikators
ItemShow
Ievads
Funkcija paredzēta preces informācijas attēlošanai

Ievade
Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none">• Preces ID
Apstrāde
Atlasa precī pēc norādītās ID no datubāzes.
Izvade
<ul style="list-style-type: none">• Veiksmīga funkcijas izpilde - atgriež pieprasītās preces informācijas skatu. Ja neeksistē prece ar norādīto ID, attēlo paziņojumu nr. 1.
Paziņojumi
1. 404 Not Found.

3.2.4. Pasūtījumu modulis



3.2.4.1. att. 2. Īmeņa datu plūsmas diagramma, pasūtījumu modulis

3.2.4.1. attēlā ir grafiski attēlots pasūtījumu modulis, visas tā funkcijas, un to funkciju datu plūsmas. Pasūtījumu moduļu detalizēto funkciju aprakstu var apskatīt, sākot no sadaļas 3.2.4.1.

3.2.4.1. Pasūtījuma izveide

Identifikators
OrderCreate
Ievads
Funkcija paredzēta jauna pasūtījuma izveidei.
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none">• Lietotāja sesijas dati.• Pasūtītāja pilnais vārds. Simbolu virkne. Maksimālais simbolu virknes garums - 100 simboli.• Pasūtītāja e-pasta adrese. E-pasta adrese. Maksimālais e-pasta garums - 50 simboli.• Piegādes adrese. Simbolu virkne. Minimālais simbolu virknes garums - 10 simboli. <p>Neobligātie lauki:</p> <ul style="list-style-type: none">• Lietotāja ID• Komentārs. Brīvs teksta lauks.
Apstrāde
<p>Ielādē lietotāja sesijas datus un pārbauda, vai lietotāja sesijā eksistē masīvs 'cart'. Ja neeksistē, izmet kļūdas paziņojumu. Ja eksistē, veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde).</p> <p>Pēc formu datu pārbaudes, pārbauda vai pasūtījums tiek izveidots no reģistrēta lietotāja profila, vai arī no neregistrēta lietotāja. Ja tas ir neregistrēts lietotājs, tad automātiski izveido lietotāja profilu, bāzējoties uz aizpildītiem pasūtījuma datiem, par lietotājevārdu ņemot pilno vārdu, atstarpes aizvietojo ar “.” un visus burtus pārveidojot mazos, bet paroli - uzģenerējot nejaušu simbolu virkni no 16 simboliem, un nokriptējot ar 'bcrypt' funkciju. Lietotāja loma tiek ielikta 'user'. Pēc lietotāja izveides, uz norādīto e-pastu tiek nosūtīts paziņojums par lietotāja izveidi un jaunizveidotā lietotāja pieejas dati. Jaunais lietotājs tiek automātiski autentificēts.</p>

Tiek izveidots jauns pasūtījuma objekts, tajā tiek ierakstīti ievadītie pasūtījuma dati un tiek noteikts pasūtījuma statuss 'new' un 'shipped' - 'no'. Pasūtījums tiek asociēts ar konkrēto lietotāju un saglabāts datubāzē. Preces preču grozā tiek asociētas ar jaunizveidoto pasūtījumu. Lietotājam uz e-pasta adresi tiek nosūtīts paziņojums par pasūtījuma izveidi. No lietotāja sesijas tiek dzēsta informācija par preču grozu, piegādes maksu un pasūtījuma kopsummā.

Izvade

- Veiksmīga funkcijas izpilde - lietotājs tiek pāradresēts uz pasūtījuma apstiprinājuma funkciju.
- Neveiksmīga funkcijas izpilde - atgriežas pie pasūtījuma datu ievades formas, un izvada attiecīgo kļūdas paziņojumu.

Paziņojumi

Nav.

3.2.4.2. Iepirkšanās groza saraksta skatīšana

Identifikators
CartIndex
Ievads
Funkcija paredzēta iepirkšanas groza attēlošanai
Ievade
Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none"> • Lietotāja sesijas dati.
Apstrāde
No lietotāja sesijas tiek ielādēti dati ar pievienoto preču sarakstu. Katrai pievienotai precei atrod noliktavu ar zemāko piegādes maksu, no kuras sūtīt. Ja pievienoto preču daudzums pārsniedz

skaitu noliktavā ar zemāko piegādes maksu, tad nepietiekamās preces tiek ņemtas no otrās lētākās noliktavas un kā piegādes maksa par šo precī tiek aprēķināta dārgākās noliktavas maksa. Ja vairākām pievienotajām precēm sakrīt zemākās noliktavas piegādes maksas, tad tiek aprēķināta tikai dārgākā piegādes maksa. Katra ielādētā prece tiek ievadīta masīvā attēlošanai iepirkšanas grozā, pirms tam cenu pareizinot ar sistēmā iestatīto PVN.

Lietotāja sesijā tiek ierakstīti dati par:

- kopsummu
- piegādes maksu
- preču daudzumu un noliktavu, no kuras tiks sūtītas preces

Izvade

- Veiksmīga funkcijas izpilde - lietotājam tiek attēlots pievienoto preču saraksts, daudzums, cena, piegādes maksa, un kopsumma. Ja preču grozā nav preču, attēlo paziņojumu nr. 1.

Paziņojumi

1. No items to display.

3.2.4.3. Preces pievienošana iepirkšanas grozam

Identifikators

CartAdd

Ievads

Funkcija paredzēta preces pievienošanai preču grozam.

Ievade

Obligātie lauki un to pieļaujamās vērtības:

- Preces ID
- Lietotāja sesijas dati

Apstrāde
<p>No lietotāja sesijas tiek ielādēti dati ar preču grozu. Atrod precī pēc ID. Ja norādītā prece jau eksistē preču grozā:</p> <ul style="list-style-type: none"> • Pārbauda, vai noliktavā pieejamais preču skaits ir lielāks par preču grozā norādīto. Ja ir lielāks, tad pārraksta lietotāja sesijas datus, palielinot konkrētās preces skaitu par 1. Ja pieejamais preču skaits noliktavās ir mazāks par preču skaitu iepirkšanas grozā, izvada kļūdas paziņojumu <p>Ja norādītā prece neeksistē preču grozā:</p> <ul style="list-style-type: none"> • Pārbauda, vai noliktavās pieejamais preču skaits ir lielāks par 0. Ja ir lielāks, tad ieraksta pievienojamās preces ID un skaitu 1 lietotāja sesijā. Ja noliktavās pieejamais preču skaits ir mazāks vai vienāds ar 0, izvada kļūdas paziņojumu.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - prece tiek pievienota grozam un tiek attēlots veiksmīgs paziņojums nr. 1. • Neveiksmīga funkcijas izpilde - ja preču skaits noliktavās ir mazāks par preču skaitu iepirkšanas grozā, attēlo kļūdas paziņojumu nr. 2. Ja neatrod precī pēc ID, attēlo paziņojumu nr. 3.
Paziņojumi
<ol style="list-style-type: none"> 1. Item succesfully added to your cart! 2. There are no more items in the warehouse! 3. 404 Not Found

3.2.4.4. Preces noņemšana no iepirkšanas groza

Identifikators
CartRemove
Ievads

Funkcija paredzēta, lai noņemtu preci no preču groza.
Ievade
Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none"> • Preces ID • Lietotāja sesijas dati
Apstrāde
No lietotāja sesijas tiek ielādēti dati ar preču grozu. Atrod preci pēc ID. Ja norādītā prece jau eksistē preču grozā ar skaitu lielāku par 0 - samazina preces skaitu par 1.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - preces skaits tiek samazināts par 1 un attēlots veiksmīgs paziņojums nr. 1. • Neveiksmīga funkcijas izpilde - ja prece ar norādīto ID neeksistē - attēlo kļūdas paziņojumu nr. 2.
Paziņojumi
<ol style="list-style-type: none"> 1. Item succesfully removed from your cart! 2. 404 Not Found

3.2.4.5. Pasūtījuma apstiprināšana

Identifikators
OrderConfirm
Ievads
Funkcija paredzēta pasūtījuma apstiprināšanai.
Ievade

Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none"> • Pasūtījuma ID
Apstrāde
Atrod pasūtījuma ierakstu datubāzē pēc ID. Attēlo lietotājam pasūtījuma apstiprinājuma skatu.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde – atgriež lietotājam pasūtījuma ID un pasūtījuma kopsummu ar iespēju izvēlēties veikt maksājumu tagad vai vēlāk. Ja lietotājs izvēlas veikt maksājumu tagad, viņš tiek pāradresēts uz maksājuma izveides funkciju. • Neveiksmīga funkcijas izpilde - ja neatrod ierakstu datubāzē pēc ID, attēlo paziņojumu nr. 1.
Paziņojumi
1. 404 Not Found

3.2.4.6. Pasūtījuma skatīšana

Identifikators
OrderShow
Ievads
Funkcija paredzēta pasūtījuma detaļu attēlošanai lietotājam.
Ievade
Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none"> • Pasūtījuma ID
Apstrāde

Atrod pasūtījuma ierakstu datubāzē pēc ID. Attēlo lietotājam pasūtījuma detalizētāku skatu.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde – atgriež lietotājam konkrētā pasūtījuma detalizētu skatu – preces, kādas tika pasūtītas, summas, un pasūtītāja informāciju. • Neveiksmīga funkcijas izpilde - ja neatrod ierakstu datubāzē pēc ID, attēlo paziņojumu nr. 1.
Paziņojumi
1. 404 Not Found

3.2.4.7. Pasūtījuma rediģēšana

Identifikators
OrderEdit
Ievads
Funkcija paredzēta eksistējoša pasūtījuma rediģēšanai.
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none"> • Pasūtītāja pilnais vārds. Simbolu virkne. Maksimālais simbolu virknes garums - 100 simboli. • Pasūtītāja e-pasta adrese. E-pasta adrese. Maksimālais e-pasta garums - 50 simboli. • Piegādes adrese. Simbolu virkne. Minimālais simbolu virknes garums - 10 simboli. • Statuss - izvēlne. Iespējamās vērtības: new,pending,cancelled,closed. <p>Neobligātie lauki:</p> <ul style="list-style-type: none"> • Komentārs. Brīvs teksta lauks.
Apstrāde

Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde).

Ja dati atbilst pieļaujamām vērtībām, atrod pasūtījuma ierakstu datubāzē pēc ID. Datubāzes ieraksts tiek atjaunināts.

Izvade

- Veiksmīga funkcijas izpilde - notiek pasūtījuma informācijas atjaunināšana un administrators tiek pāradresēts uz pasūtījuma saraksta skatīšanas funkciju ar paziņojumu.
- Neveiksmīga funkcijas izpilde - atgriežas pie pasūtījuma datu rediģēšanas formas, un izvada attiecīgo kļūdas paziņojumu. Ja neatrod ierakstu datubāzē pēc ID, attēlo paziņojumu nr. 2.

Paziņojumi

2. Order was succesfully edited!
3. 404 | Not Found

3.2.4.8. Pasūtījuma dzēšana

Identifikators

OrderRemove

Ievads

Funkcija paredzēta lai dzēstu konkrētu pasūtījumu.

Ievade

Obligātie lauki un to pieļaujamās vērtības:

- Pasūtījuma ID

Apstrāde

Pārbauda, vai eksistē pasūtījums ar norādīto ID un dzēš pasūtījumu no datubāzes.

Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - notiek pasūtījuma dzēšana, un lietotājs tiek pāradresēts uz pasūtījumu saraksta skatīšanas funkciju ar paziņojumu nr 1. • Neveiksmīga funkcijas izpilde - ja pasūtījums ar norādīto ID netiek atrasts - attēlo kļūdas paziņojumu nr 2.
Paziņojumi
<ol style="list-style-type: none"> 1. Order succesfully deleted! 2. 404 Not Found

3.2.4.9. Pasūtījuma izsūtīšana

Identifikators
OrderShip
Ievads
Funkcija paredzēta pasūtījuma statusa maiņai un paziņošanai to lietotājam
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none"> • Pasūtījuma ID
Apstrāde
<p>Pārbauda, vai pasūtījums ar norādīto ID eksistē. Nomaina konkrētā pasūtījuma atribūtu 'shipped' uz 'true', ielādē no datubāzes sarakstu ar pasūtītājam precēm, to skaitu, un no kuras noliktavas tās preces tika sūtītas. Atbilstoši sarakstam tiek atjaunināts preču skaits katrā noliktavā. Pēc tam izsūta uz pasūtītāja e-pasta adresi paziņojumu par pasūtījuma izsūtīšanu.</p>
Izvade

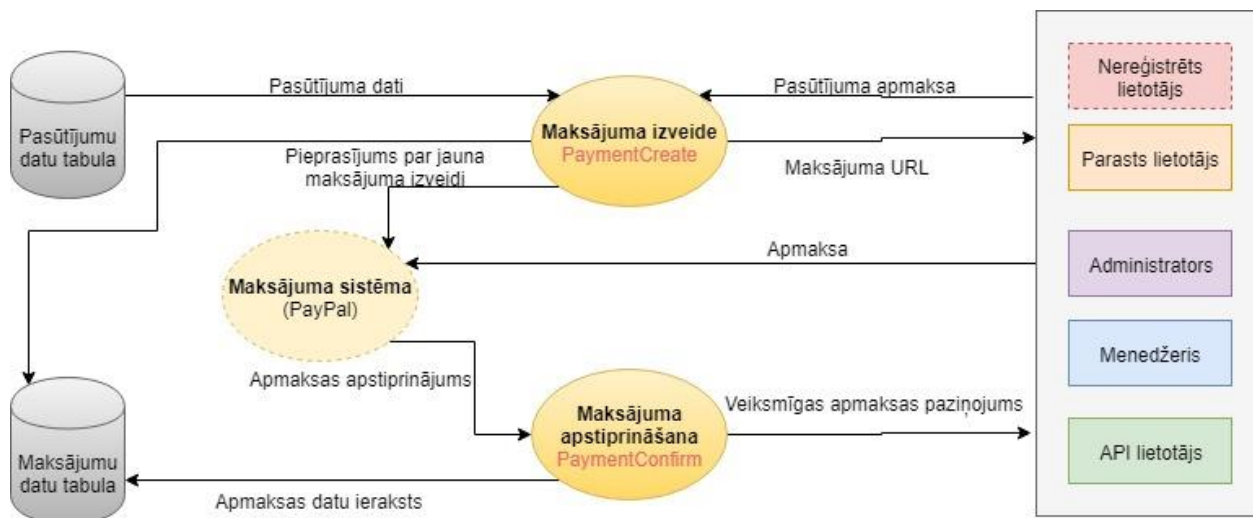
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - lietotāju pāradresē uz pasūtījumu sarakstu skatīšanas funkciju ar paziņojumu nr. 1. • Neveiksmīga funkcijas izpilde - ja pasūtījums ar norādīto ID neeksistē - attēlo paziņojumu nr. 2.
Paziņojumi
<ol style="list-style-type: none"> 1. Order #[Pasūtījuma ID] shipped! 2. 404 Not Found

3.2.4.10. Pasūtījumu saraksta skatīšana

Identifikators
OrderIndex
Ievads
Funkcija paredzēta pasūtījumu saraksta attēlošanai
Ievade
Nav
Apstrāde
Nosaka autentificētā lietotāja tipu - ja autentificēts lietotājs ar lomu 'admin', ielādē visus pasūtījumus no datubāzes. Ja autentificētajam lietotājam nav piešķirta 'admin' grupa, ielādē tikai pasūtījumus, kas pieder šim autentificētajam lietotājam.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - izvada pasūtījumu sarakstu tabulas veidā. Ja neviens pasūtījums neeksistē - attēlo paziņojumu nr. 1.
Paziņojumi

1. No items to display.

3.2.5. Maksājumu modulis



3.2.5.1. att. 2. Īmeņa datu plūsmas diagramma, maksājumu modulis

3.2.5.1. attēlā ir grafiski attēlots pasūtījumu modulis, visas tā funkcijas, un to funkciju datu plūsmas. Pasūtījumu moduļu detalizēto funkciju aprakstu var apskatīt, sākot no sadaļas 3.2.5.1.

3.2.5.1. Maksājuma izveide

Identifikators
PaymentCreate
Ievads
Funkcija paredzēta jauna Paypal maksājuma izveidei
Ievade
Obligātie lauki un to pieļaujamās vērtības: <ul style="list-style-type: none">• Pasūtījuma ID;• Lietotāja sesijas 'shipping' dati.

Apstrāde
<p>Pārbauda, vai eksistē pasūtījums ar norādīto ID. Ja eksistē, ievada maksājumam pasūtīto preču sarakstu, cenu, skaitu, piegādes maksu. Izveido jaunu maksājumu, izmantojot Paypal API SDK. Ja maksājuma izveide notiek veiksmīgi, no Paypal atbildes nolasa maksājuma URL, izveido jaunu sistēmas maksājuma objektu, un saglabā datubāzē, piesaistot pasūtījuma ID un Paypal maksājuma ID.</p>
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - lietotājs tiek pāradresēts uz Paypal maksājuma veikšanai. • Neveiksmīga funkcijas izpilde - ja pasūtījums ar norādīto ID netika atrasts - attēlo paziņojumu nr. 1. Ja rodas kļūda ar savienojumu ar PayPal API, attēlo kļūdu un paziņojumu nr. 2. Ja rodas kļūda ar pasūtījuma datiem (piem. neatbilst preču un pasūtījuma kopsummas), attēlo kļūdu un paziņojumu nr. 3. Ja nav iestatīts atgriezenisks URL, attēlo paziņojumu nr. 4.
Paziņojumi
<ol style="list-style-type: none"> 1. 404 Not Found 2. Some error occurred with connecting to PayPal! 3. Some error occurred with the payment generation! 4. Some error occurred with redirection!

3.2.5.2. Maksājuma apstiprināšana

Identifikators
PaymentConfirm
Ievads
Funkcija paredzēta Paypal maksājuma apstiprināšanai
Ievade

Obligātie lauki un to pieļaujamās vērtības:

- Maksājuma ID
- Maksātāja ID
- Maksājuma marķieris (token)

Apstrāde

Pārbauda, vai visi obligātie lauki ir aizpildīti. Ja kāds no laukiem nav aizpildīts - maksājums nav veikts un funkcija atgriež kļūdas paziņojumu. Ja visi obligātie lauki ir aizpildīti - pieprasa maksājuma izpildi no Paypal. Ja maksājums ir apstiprināts, ielādē sistēmas maksājumu no datubāzes, aizpilda ierakstu ar maksātāja ID un statusu 'paid' - 'true'.

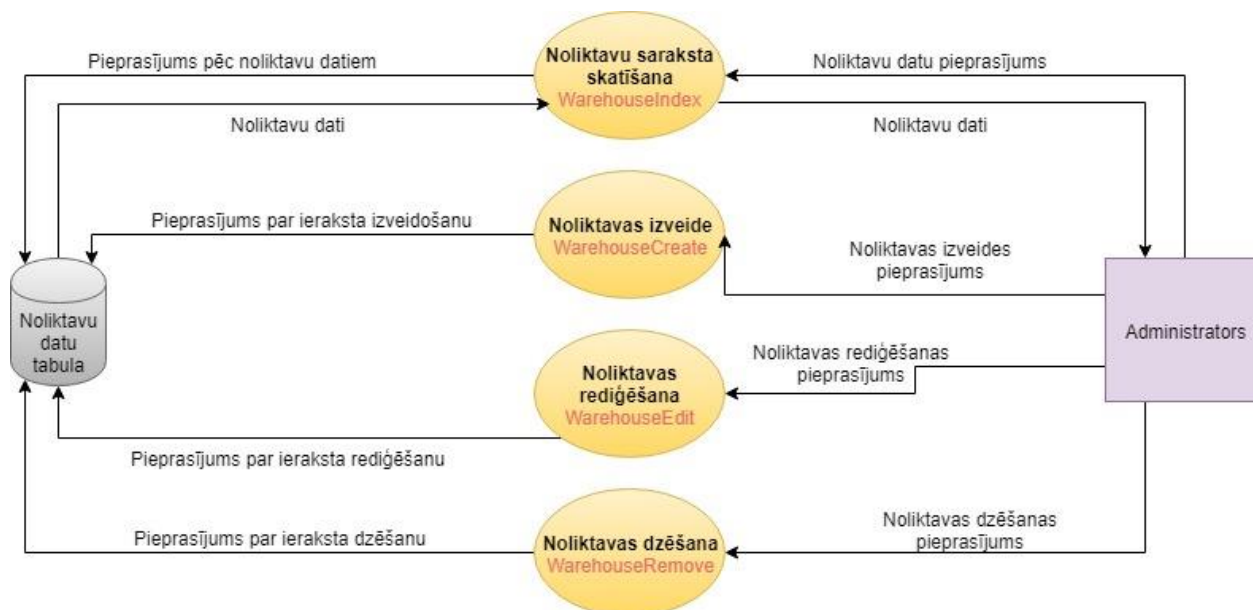
Izvade

- Veiksmīga funkcijas izpilde - notiek maksājuma pārbaude un lietotājs tiek pāradresēts uz preču saraksta skatīšanas funkciju ar paziņojumu nr 1.
- Neveiksmīga funkcijas izpilde - ja kāds no obligātiem laukiem nav aizpildīts - attēlo paziņojumu nr 2. Ja no PayPal saņemta atbilde par to, ka maksājums nav apstiprināts, tiek attēlots paziņojums nr. 3.

Paziņojumi

1. Payment made successfully! Our manager will contact you if needed, and you will receive an e-mail when your items are shipped!
2. Payment was not made.
3. Payment was not successful.

3.2.6. Noliktavu modulis



3.2.6.1. att. 2. Īmeņa datu plūsmas diagramma, noliktavu modulis

3.2.6.1. attēlā ir grafiski attēlots pasūtījumu modulis, visas tā funkcijas, un to funkciju datu plūsmas. Pasūtījumu moduļu detalizēto funkciju aprakstu var apskatīt, sākot no sadaļas 3.2.6.1.

3.2.6.1. Noliktavu saraksta skatīšana

Identifikators
WarehouseIndex
Ievads
Funkcija paredzēta noliktavu saraksta attēlošanai.
Ievade
Nav
Apstrāde
Ielādē no datubāzes visus noliktavu ierakstus.

Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - attēlo informāciju par visām sistēmā esošām noliktavām tabulas veidā. Ja neeksistē neviena noliktava, tiek attēlots paziņojums nr. 1.
Paziņojumi
1. No items to display.

3.2.6.2. Noliktavas izveide

Identifikators
WarehouseCreate
Ievads
Funkcija paredzēta jaunas noliktavas izveidei
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none"> • Nosaukums. Unikāla simbolu virkne. Maksimālais simbolu virknes garums - 50 simboli. • Adrese. Simbolu virkne. Maksimālais simbolu virknes garums - 50 simboli. • Tālruna numurs. Simbolu virkne. Maksimālais simbolu virknes garums - 15 simboli.
Apstrāde
<p>Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Ja pārbaudījumi ir veiksmīgi, izveido jaunu noliktavas ierakstu, aizpilda ar ievadīto informāciju, un saglabā datubāzē. Jauno noliktavu piesaista visām precēm datubāzē.</p>
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - notiek noliktavas izveide, un lietotājs tiek pāradresēts uz noliktavu saraksta skatīšanas funkciju ar paziņojumu nr. 1.

- Neveiksmīga funkcijas izpilde - atgriežas pie datu ievades formas un attēlo attiecīgo kļūdas paziņojumu.

Paziņojumi

1. Warehouse successfully created!

3.2.6.3. Noliktavas rediģēšana

Identifikators

WarehouseEdit

Ievads

Funkcija paredzēta eksistējošas noliktavas rediģēšanai.

Ievade

Obligātie lauki un to pieļaujamās vērtības:

- Nosaukums. Unikāla simbolu virkne. Maksimālais simbolu virknes garums - 50 simboli.
- Adrese. Simbolu virkne. Maksimālais simbolu virknes garums - 50 simboli.
- Tālruņa numurs. Simbolu virkne. Maksimālais simbolu virknes garums - 15 simboli.

Apstrāde

Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Atrod noliktavu pēc ID un atjaunina noliktavas ierakstu datubāzē.

Izvade

- Veiksmīga funkcijas izpilde - notiek noliktavas datu atjaunināšana, un lietotājs tiek pāradresēts uz noliktavu saraksta skatīšanas funkciju ar paziņojumu nr. 1..
- Neveiksmīga funkcijas izpilde - atgriežas pie datu ievades formas un attēlo attiecīgu kļūdas paziņojumu. Ja noliktava pēc norādītā ID nav atrasta - attēlo paziņojumu nr. 2.

Paziņojumi
<ol style="list-style-type: none"> 1. Warehouse was successfully edited! 2. 404 Not Found

3.2.6.4. Noliktavas dzēšana

Identifikators
WarehouseRemove
Ievads
Funkcija paredzēta noliktavas dzēšanai
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none"> • Noliktavas ID
Apstrāde
<p>Pārbauda, vai eksistē noliktava ar norādīto ID. Dzēš noliktavas ierakstu datubāzē, kā arī visu informāciju par preču skaitu un piegādes maksu konkrētajā noliktāvā.</p>
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - notiek noliktavas dzēšana, un lietotājs tiek pāradresēts uz noliktavu saraksta skatīšanas funkciju ar paziņojumu nr. 1. • Neveiksmīga funkcijas izpilde - ja noliktava ar norādīto ID nav atrasta - attēlo paziņojumu nr. 2.
Paziņojumi
<ol style="list-style-type: none"> 1. Warehouse was successfully deleted!

3.2.7. Sistēmas iestatījumu modulis



3.2.7.1. att. 2. Īmeņa datu plūsmas diagramma, sistēmas iestatījumu modulis

3.2.6.1. attēlā ir grafiski attēlots pasūtījumu modulis, visas tā funkcijas, un to funkciju datu plūsmas. Pasūtījumu moduļu detalizēto funkciju aprakstu var apskatīt, sākot no sadaļas 3.2.6.1.

3.2.7.1. Sistēmas iestatījumu saraksta skatīšana

Identifikators
SettingsIndex
Ievads
Funkcija paredzēta sistēmas iestatījumu saraksta attēlošanai
Ievade
Nav
Apstrāde
Ielādē iestatījumus no sistēmas konfigurācijas faila.
Izvade

<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - attēlo sistēmas iestatījumus
Paziņojumi
Nav.

3.2.7.2. Sistēmas iestatījumu rediģēšana

Identifikators
SettingsUpdate
Ievads
Funkcija paredzēta sistēmas iestatījumu rediģēšanai un saglabāšanai
Ievade
<p>Obligātie lauki un to pieļaujamās vērtības:</p> <ul style="list-style-type: none"> • PVN procentos. Skaitliska vērtība. Maksimālā vērtība - 50; • Piegādes uzcelojums procentos. Skaitliska vērtība. Maksimālā vērtība - 100. • Sistēmas nosaukums. Simbolu virkne. Maksimālais simbolu skaits - 30.
Apstrāde
Veic ievadītu datu formāta un obligāti aizpildāmo lauku pārbaudi (skatīt nodaļu 4. Standarta kļūdu apstrāde). Pārraksta konfigurācijas failu ar jauniem iestatījumiem.
Izvade
<ul style="list-style-type: none"> • Veiksmīga funkcijas izpilde - notiek iestatījumu ierakstīšana, un lietotājs tiek pārdresēts uz sistēmas iestatījumu saraksta skatīšanas funkciju ar paziņojumu nr. 1. • Neveiksmīga funkcijas izpilde - ja kaut kādu iemeslu dēļ nav iespējams pārrakstīt iestatījumus failā, attēlo paziņojumu nr. 2.
Paziņojumi

1. Settings were updated successfully!
2. Cannot write system configuration!

3.3. Nefunkcionālās prasības

3.3.1. Tabula

Nefunkcionālo prasību saraksts

Prasības numurs	Prasības apraksts
NP. 1	Drošība - Lietotāju parolēm ir jābūt drošām (vismaz 8 simboli, viens cipars, spec simbols, mazais un lielais burts) un datubāzē jāglabājas šifrētā veidā, izmantojot bcrypt funkciju. Datu pārraide notiek, izmantojot TLS versijas 1.3 šifrēšanu. Lietotnes WEB serveris pārraida HSTS, X-XSS-Protection, X-Content-Type-Options un Content-Security Policy galvenes, lai izvairītos no dažādu veidu uzbrukumiem. Lietotāju sesijas dati tiek dzēsti pēc nopietnas lietotnes statusa maiņas, piemēram, paroles maiņas vai atjaunošanas. Formās tiek izmantoti CSRF marķieri un visi dati tiek filtrēti un validēti gan no front-end, gan no back-end, lai izvairītos no nevēlamiem datiem, kā arī SQL injekciju uzbrukumiem.
NP. 2	Veiktspēja - Pieprasījumiem, kas nav saistīti ar ārēju sistēmu izsaukšanu, jābūt izpildītiem laikā, kas nav lielāks par 1 sekundi. Savukārt pieprasījumiem, kas ir saistīti ar ārēju sistēmu izsaukšanu, ir jābūt izpildītiem laikā, kas nav lielāks par 5 sekundēm.
NP. 3	Izmantojamība - sistēma ir pieejama no jebkuras ierīces, kurai ir pieejams internets. Tā kā izstrādātā sistēma ir WEB lietotne, tai nav ierobežojumu attiecībā uz ierīču operētājsistēmām - tā strādā ar jebkuru modernu viedierīci, piemēram, telefonu, planšetdatoru, datoru, un jebkurām operētājsistēmām - Windows, MacOS, Linux, Android, iOS, Windows Phone, utt.
NP. 4	Uzturamība - sistēma ir izstrādāta, izmantojot MVC arhitektūru. Par modeļiem kalpo sistēmas objekti, biznesa loģika tiek aprakstīta modeļu kontrolleros, bet lietotāja saskarne atrodas skatos, kas satur HTML kodu, savukārt par maršrutēšanu atbild maršrutēšanas konfigurācija. Tādējādi, izmantojot šādu arhitektūru, lietotnes tālākā uzturēšana ir vienkārša.
NP. 5	Pielāgojamība - sistēma ir pielāgojama izmantošanai jebkurā valstī un nozarē - tai nav

	<p>iekodēti parametri, kas ir atkarīgi no valsts vai nozares. Sistēma ir tieši paredzēta kā universāla pasūtījumu pārvaldīšanas sistēma, kas var tikt izmantota jebkur, līdz ar to sistēmā eksistē dažādi rediģējami parametri.</p>
--	---

4. STANDARTA KĻŪDU APSTRĀDE

Šī nodaļa apraksta standarta kļūdu paziņojumu šablonus. Sistēmā ir vairākas funkcijas un skati, kas ietver datu ievadi. Lai pārbaudītu šos ievades datu un dotu lietotājam kļūdas skaidrojumu, ja tāda ir radusies, ir paredzēti universāli kļūdu paziņojumu teksta šabloni dažādiem gadījumiem, kurus var apskatīt tabulā 4.1.

4.1. Tabula

Kļūdu paziņojumu šabloni

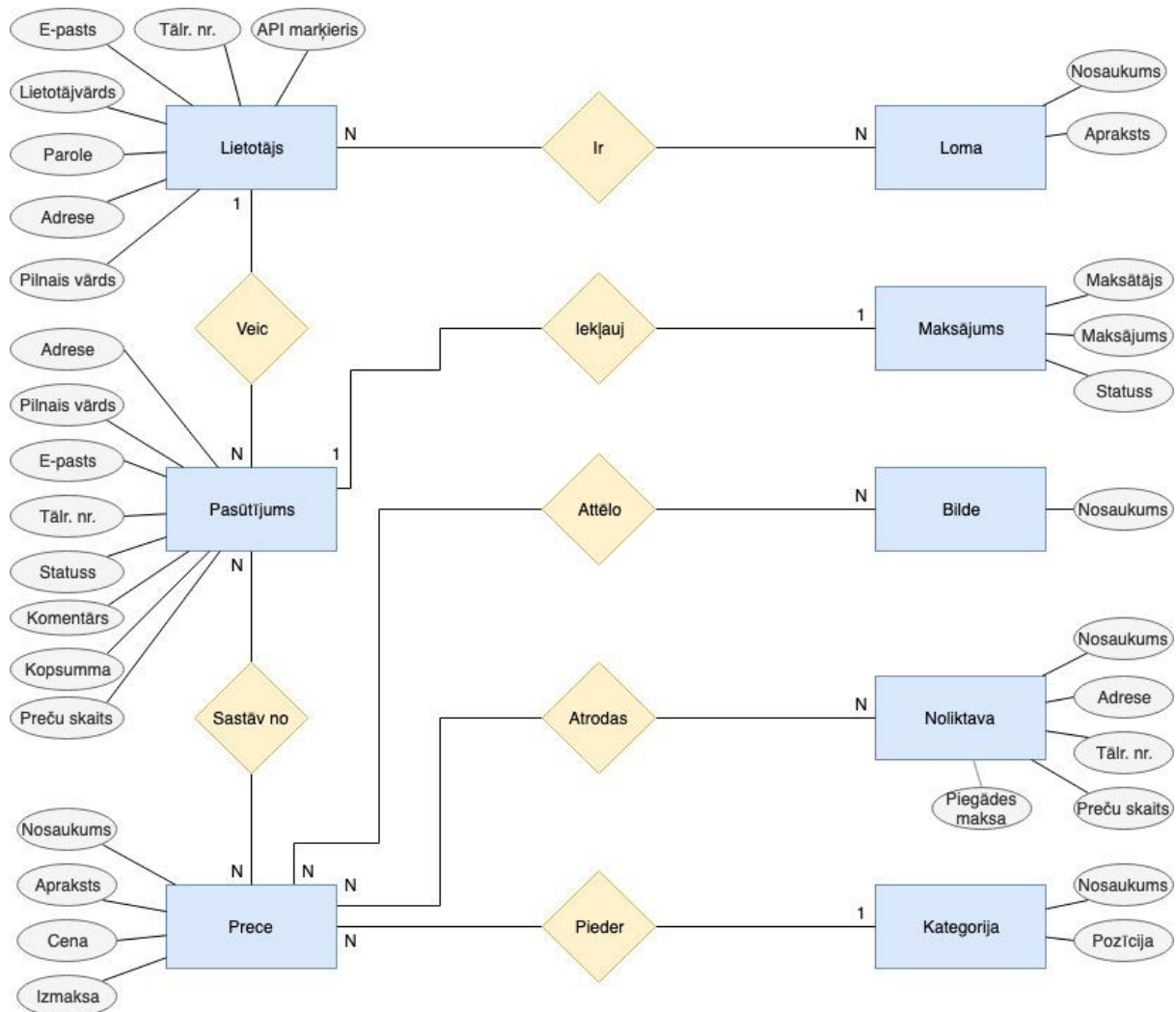
Iemesls	Kļūdas paziņojums
Obligātais lauks nav ievadīts	The [Lauka_nosaukums] field is required.
Ievadītā simbolu virkne nesasniedz pieļauto simbolu skaitu	The [Lauka_nosaukums] must be at least [minimālais garums] characters.
Ievadītā simbolu virkne pārsniedz pieļauto simbolu skaitu	The [Lauka_nosaukums] may not be greater than [maksimālais garums] characters.
Ievadītais e-pasts neatbilst e-pasta adreses formātam	The [Lauka_nosaukums] must be a valid email address.
Paroles nesakrīt	The [Lauka_nosaukums] confirmation does not match.
Ievadītā skaitliskā vērtība nesasniedz pieļauto vērtību	The [Lauka_nosaukums] must be at least [minimālā vērtība].
Ievadītā skaitliskā vērtība pārsniedz pieļauto vērtību	The [Lauka_nosaukums] may not be greater than [maksimālā vērtība].
Augšupielādējamais fails nav attēls	The [Lauka_nosaukums] must be an image.
Ievadītā vērtība neeksistē datubāzē (relācijās)	The selected [Lauka_nosaukums] is invalid.

Fails pārsniedz atļauto izmēru	The [Lauka_nosaukums] may not be greater than [maksimālais izmērs kilobaitos] kilobytes.
Ievadītais lauks nav simbolu virkne	The [Lauka_nosaukums] must be a string.
Ievadītā vērtība nav cipars	The [Lauka_nosaukums] must be a number.
Ievadītā vērtība neksistē izvēlnē	The selected [Lauka_nosaukums] is invalid.
Ievadītā vērtība nav unikāla.	The [Lauka_nosaukums] has already been taken.
Ievadītā vērtība neatbilst prasību kopumam (parole)	The [Lauka_nosaukums] format is invalid.

5. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

5.1. Konceptuālais datu bāzes apraksts

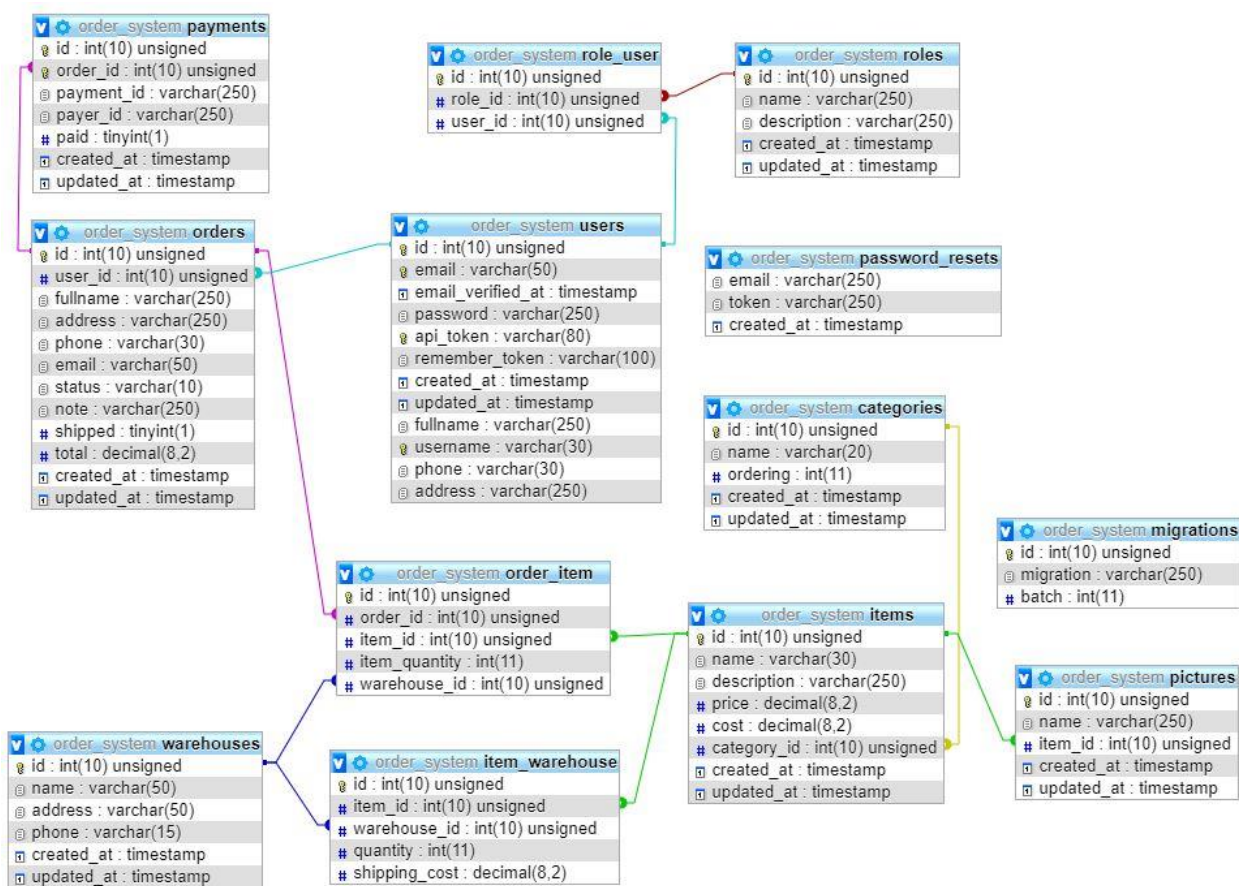
Kā ir attēlots 5.1.1. attēlā, konceptuālajā ER modelī, pasūtījumu pārvaldības sistēmā galvenie objekti ir lietotājs, prece, un pasūtījums. Lietotājam ir viena vai vairākas lomas sistēmā, kas nosaka viņa tiesības. Lietotājs sistēmā veic pasūtījumu, ievadot pasūtījuma datus, un pasūtījums, savukārt, sastāv no lietotāja iepriekš izvēlētām precēm, kur katrai ir savs apraksts un cena, un tās sistēmā tiek attēlotas ar bildēm. Preces fiziski atrodas noliktavās, katrā noliktavā ir kāds konkrēts preču skaits, un preces pieder kādai no preču kategorijām. Visbeidzot, pasūtījums iekļauj maksājumu.



5.1.1. att. Konceptuālais ER modelis

5.2. Datu bāzes projektējums

Fiziskajā datu modelī (attēls 5.2.1.) datubāze tika projektēta, izmantojot MySQL platformu. Projektējumam pārejot uz fizisko datu bāzes modeli, parādījās vairākas starptabulas N:N relāciju realizācijai, kā arī šajās starptabulās glabājas papildus informācija, tāda, kā, piemēram, tabulā `item_warehouse` – preču skaits un piegādes maksa, un tabulā `order_item` – konkrētās preces skaits un noliktava, no kuras preces tika pasūtītas.



5.2.1. att. Fiziskais datu bāzes modelis

5.3. Datubāzes tabulu projektējums un apraksts

Visi sistēmas dati tiek glabāti vienā datubāzē, kas ir sadalīta 12 tabulās.

Datubāzes tabulu uzskaitījums un to apraksts ir apkopots tabulā 5.3.1, savukārt detalizēts katras tabulas projektējums ir aprakstīts, sākot no 5.3.2. tabulas. Papildinformācijā lietotie apzīmējumi: FK – atsauces atslēga, PK – primārā atslēga, AI – automātiski pieaugošā vērtība, Index – indekssēts lauks.

5.3.1. Tabula

Datubāzes tabulu uzskaitījums

Datubāzes objekta nosaukums	Datubāzes objekta apraksts	Objekta tips
users	Sistēmas lietotāji	Tabula
roles	Sistēmas lomas un to apraksti	Tabula
role_user	Tabula N:N relācijas izveidošanai starp lomām un lietotājiem	Tabula
orders	Veiktie pasūtījumi	Tabula
order_item	Tabula N:N relācijas izveidošanai starp pasūtījumiem un precēm	Tabula
payments	Veiktie maksājumi	Tabula
warehouses	Preču noliktavas	Tabula
item_warehouse	Tabula N:N relācijas izveidošanai starp noliktavām un precēm	Tabula
categories	Preču kategorijas	Tabula
pictures	Preču attēli	Tabula
password_resets	Paroles atjaunošanas marķieri	Tabula
items	Preces	Tabula

5.3.2. Tabula

Tabulas "users" apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
email	varchar(50)	Unikāls	E-pasta adrese
password	varchar(250)		Parole
api_token	varchar(80)		API piekļuves marķieris
remember_token	varchar(100)		Lietotāja atceršanās marķieris
fullname	varchar(100)		Lietotāja pilnais vārds
username	varchar(30)		Lietotājvārds
phone	varchar(15)		Tālr. numurs
address	varchar(250)		Adrese
created_at	timestamp		Ieraksta izveides datums un laiks
updated_at	timestamp		Ieraksta atjaunināšanas datums un laiks

5.3.3. Tabula

Tabulas "roles" apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
name	varchar(50)	Unikāls	Lomas nosaukums
description	varchar(250)		Lomas apraksts
created_at	timestamp		Ieraksta izveides

			datums un laiks
updated_at	timestamp		Ieraksta atjaunināšanas datums un laiks

5.3.4. Tabula

Tabulas "role_user" apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
role_id	int(10)	FK	Atsauces atslēga uz ierakstu "roles" tabulā
user_id	int(10)	FK	Atsauces atslēga uz ierakstu "users" tabulā

5.3.5. Tabula

Tabulas "orders" apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
user_id	int(10)	FK	Atsauces atslēga uz ierakstu "users" tabulā
fullname	varchar(100)		Pasūtītāja pilnais vārds
address	varchar(80)		Piegādes adrese
phone	varchar(15)		Tālr. nr.
email	varchar(100)		Pasūtītāja e-pasta adrese
status	ENUM('new',	noklusētā vērtība -	Pasūtījuma statuss

	'pending', 'cancelled', 'closed')	'new'	
note	varchar(250)	Var būt tukšs	Pasūtījuma komentārs
shipped	boolean	noklusētā vērtība - false	Vai prece ir izsūtīta
total	decimal(8,2)		Pasūtījuma kopsumma
created_at	timestamp		Ieraksta izveides datums un laiks
updated_at	timestamp		Ieraksta atjaunināšanas datums un laiks

5.3.6. Tabula

Tabulas "order_item" apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
order_id	int(10)	FK	Atsauces atslēga uz ierakstu "orders" tabulā
item_id	int(10)	FK	Atsauces atslēga uz ierakstu "items" tabulā
item_quantity	int(10)		Pasūtīto preču skaits
warehouse_id	int(10)	FK	Atsauces atslēga uz ierakstu "warehouses" tabulā

5.3.7. Tabula

Tabulas “payments” apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
order_id	int(10)	FK	Atsauces atslēga uz ierakstu “orders” tabulā
payment_id	varchar(250)	Var būt NULL	PayPal maksājuma ID
payer_id	varchar(250)	Var būt NULL	PayPal maksātāja ID
paid	boolean	Noklusējuma vērtība - false	Vai apmaksa ir veikta
created_at	timestamp		Ieraksta izveides datums un laiks
updated_at	timestamp		Ieraksta atjaunināšanas datums un laiks

5.3.8. Tabula

Tabulas “warehouses” apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
name	varchar(15)		Noliktavas nosaukums
address	varchar(250)		Noliktavas adrese
phone	varchar(15)		Noliktavas tālr. nr.
created_at	timestamp		Ieraksta izveides datums un laiks
updated_at	timestamp		Ieraksta atjaunināšanas datums un laiks

5.3.9. Tabula

Tabulas "item_warehouse" apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
item_id	int(10)	FK	Atsauces atslēga uz ierakstu "items" tabulā
warehouse_id	int(10)	FK	Atsauces atslēga uz ierakstu "warehouses" tabulā
quantity	int(10)	Noklusētā vērtība - 0	Preču skaits
shipping_cost	decimal(8,2)	Noklusētā vērtība - 0.00	Preces piegādes maksa

5.3.10. Tabula

Tabulas "categories" apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
name	varchar(15)		Kategorijas nosaukums
ordering	tinyint(4)		Kategorijas pozīcija
created_at	timestamp		Ieraksta izveides datums un laiks
updated_at	timestamp		Ieraksta atjaunināšanas datums un laiks

5.3.11. Tabula

Tabulas “pictures” apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
name	varchar(250)		Attēla nosaukums
item_id	int(10)	FK	Atsauces atslēga uz ierakstu “items” tabulā
created_at	timestamp		Ieraksta izveides datums un laiks
updated_at	timestamp		Ieraksta atjaunināšanas datums un laiks

5.3.12. Tabula

Tabulas “password_resets” apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
email	varchar(100)	Index	E-pasta adrese
token	varchar(250)		Marķieris
created_at	timestamp		Ieraksta izveides datums un laiks

5.3.13. Tabula

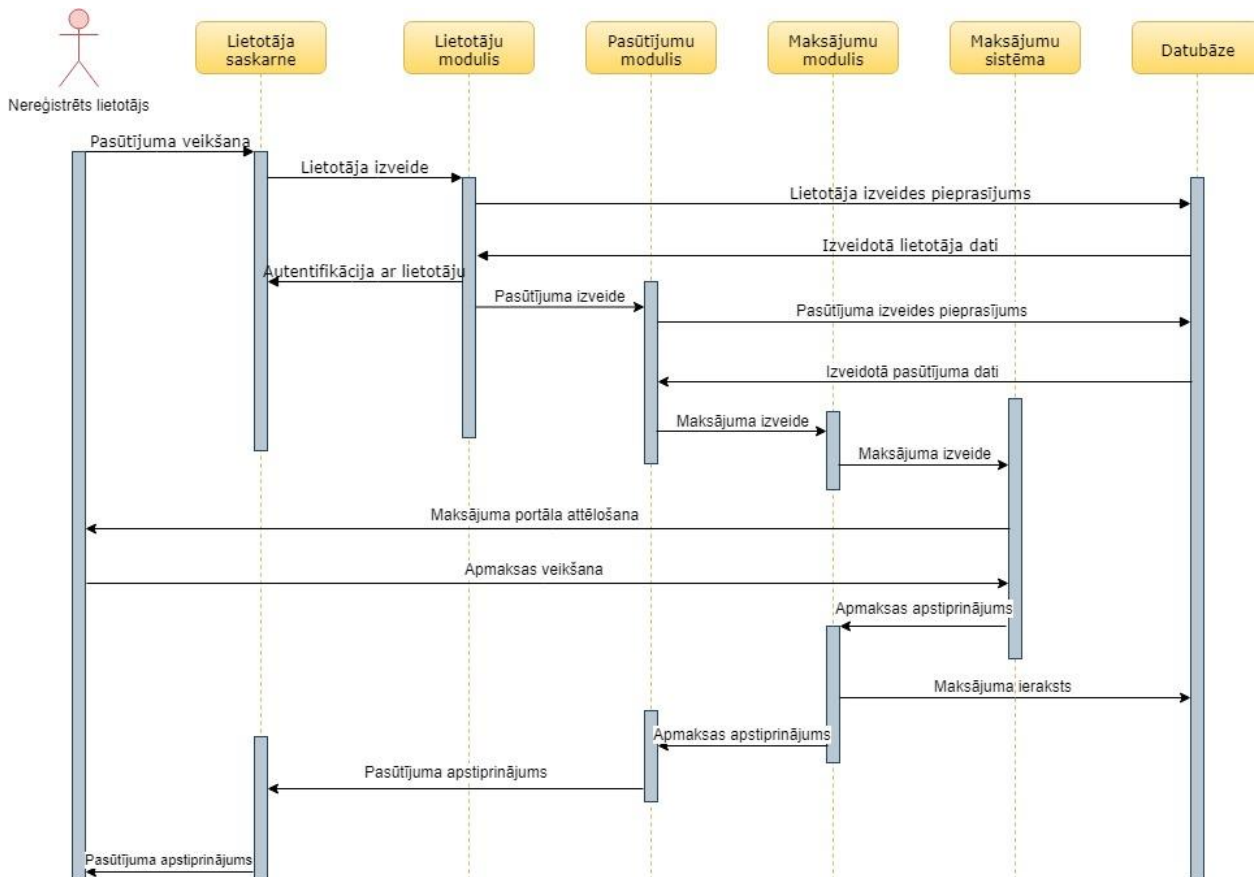
Tabulas “items” apraksts

Kolonnas nosaukums	Datu tips	Papildinformācija	Apraksts
id	int(10)	PK / AI	Unikāls identifikators
name	varchar(50)	Unikāls	Lomas nosaukums
description	varchar(250)		Lomas apraksts

price	decimal(8,2)		Preces cena
cost	decimal(8,2)		Preces izmaksa
category_id	int(10)	FK	Atsauces atslēga uz ierakstu "categories" tabulā
created_at	timestamp		Ieraksta izveides datums un laiks
updated_at	timestamp		Ieraksta atjaunināšanas datums un laiks

5.4. Daļējs funkciju projektējums

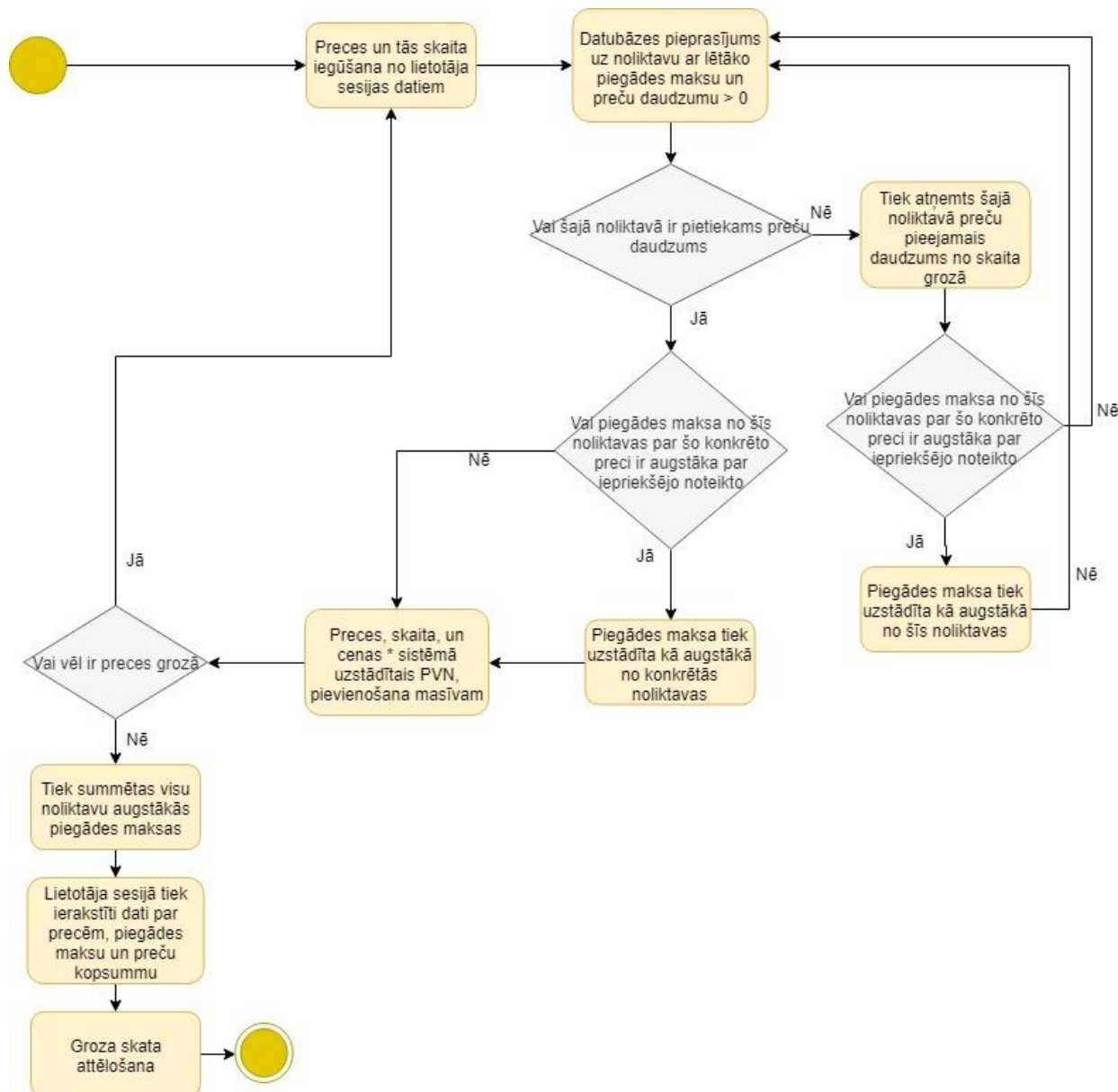
Lai vieglāk uztvertu sistēmas sarežģītākas funkcijas, ir izstrādāti to funkciju projektējumi, kas vizuāli attēlo funkcijas algoritmu.



5.4.1.att. Pasūtījuma izveides (OrderCreate) funkcijas secību diagramma

Attēlā 5.4.1 ir vizuāli parādīta funkcijas OrderCreate darbība gadījumā, ja pasūtījumu veic neregistrēts lietotājs. Lietotājs pirms pasūtījuma izveides ievada visu nepieciešamo informāciju, kura ir detalizēti aprakstīta OrderCreate funkcijas aprakstā. Pēc tam, kad informācija ir ievadīta, sistēma, pārbaudot, ka lietotājs nav autentificēts, nosūta pieprasījumu uz lietotāju izveidi uz lietotāju moduli ar lietotāja ievadītajiem datiem. Tas tiek pārveidots datubāzes pieprasījumā un nosūtīts uz izveidi datubāzē. No datubāzes saņemot jaunā lietotāja datus, par jaunu izveidotu lietotāju tiek nosūtīta e-pasta ziņa uz pasūtījumā norādīto e-pasta adresi ar lietotājvārdu un paroli, un automātiski notiek autentifikācija ar jauno lietotāju. Tālāk seko pasūtījuma ieraksta izveidošana pasūtījumu modulī. Jauns ieraksts tiek izveidots datubāzē un datubāze atgriez izveidotā pasūtījuma datus, uz kā bāzes maksājumu modulī tiek veikta maksājuma izveide datubāzē. Ja maksājuma

izveide ir veiksmīga, maksājums tiek izveidots arī maksājumu sistēmā, un lietotājs tiek pārdresēts uz maksājuma sistēmas saskarni, kur viņš veic pasūtījuma apmaksu. Pēc pasūtījuma apmaksas maksājuma sistēma nodod apstiprinājumu maksājumu modulim, kurš atjaunina datus datubāzē un apstiprina apmaksu pasūtījumu modulim. Tālāk pasūtījumu modulis atgriež pasūtījuma apstiprinājumu lietotājam lietotāja saskarnē.

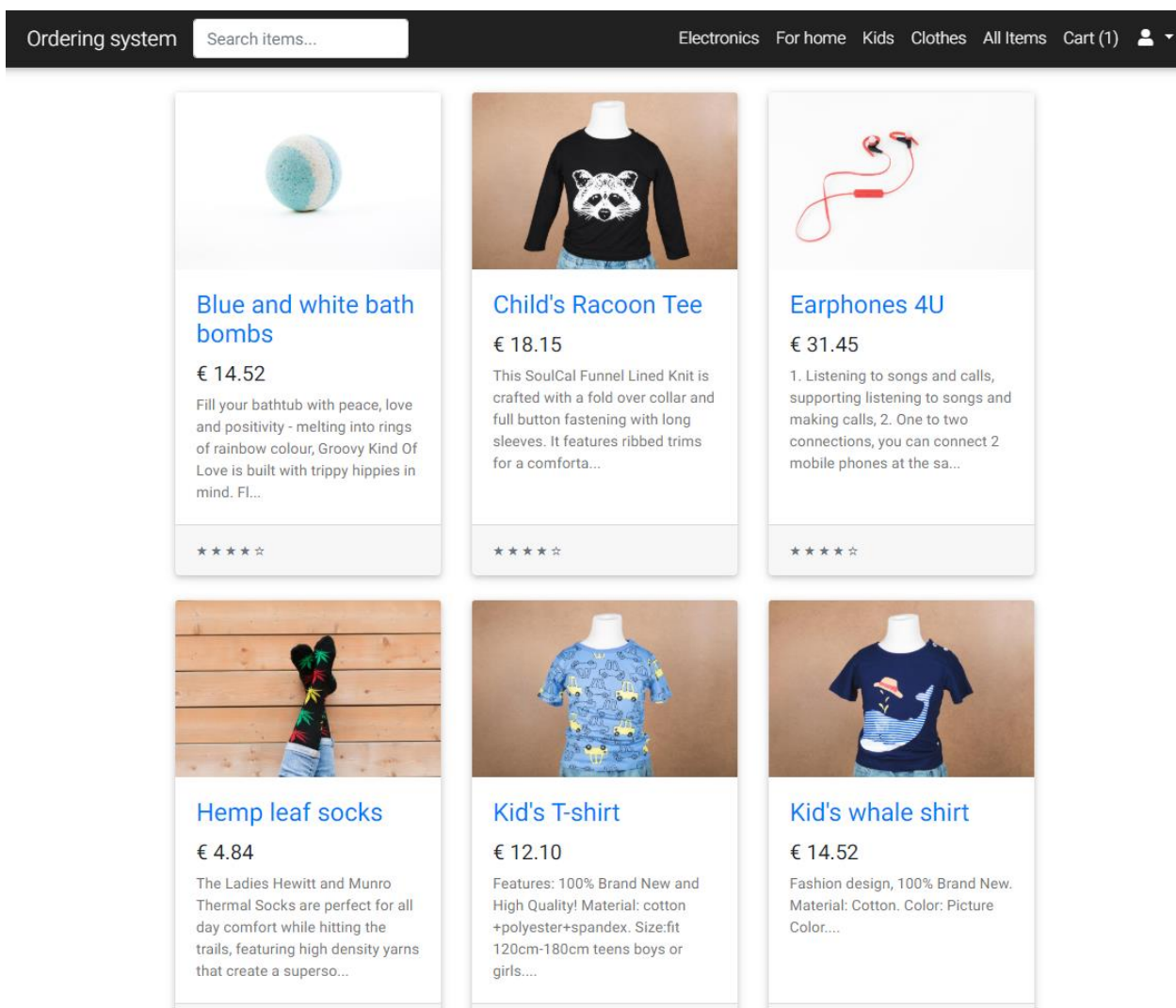


5.4.2.att. Preču groza skatīšanas funkcijas aktivitāšu diagramma

Attēlā 5.4.2 ir parādīta preču groza skatīšanas funkcijas darbības algoritms. Sākumā no lietotāja sesijas datiem tiek iegūtas grozam pievienotās preces un to skaits. Tālāk sistēma katrai precei no lietotāja sesijas veic pieprasījumu uz datubāzi pēc noliktavas ar lētāko piegādes maksu un nepieciešamo preču skaitu. Ja norādītais preču skaits neeksistē nevienā noliktavā, sistēma norezervē esošu skaitu lētākajā noliktavā, fiksē piegādes maksu no šīs noliktavas un atkal veic pieprasījumu uz datubāzi pēc noliktavas ar lētāko piegādes maksu un nepieciešamo preču skaitu, vienlaikus piefiksējot piegādes maksu no katras noliktavas, tā veicot ciklu līdz tiek sasniegts nepieciešamais preču daudzums. Tad preces skaits, noliktavu ID un cena * sistēmā uzstādītais PVN tiek pievienotas masīvam kurš tiks padots uz groza attēlošanas skatu. Ja grozā vēl ir palikušas preces, sistēma veic jaunu ciklu ar iepriekšējām darbībām, pārbaudot piegādes maksas no noliktavām un pārrakstot tos, gadījumā, ja iepriekšējās ir zemākas, tādējādi iegūstot masīvu ar augstāko piegādes maksu no katras noliktavas. Kad visas preces grozā ir apstrādātas, visu noliktavu piegādes maksas tiek summētas un ierakstītas lietotāja sesijā kopā ar preču kopsummu un precēm, un informācija tiek attēlota groza skatā.


5.5. Daļējs lietotāja saskarņu projektējums

Sistēmai tika izstrādāta lietotāju saskarne, kur pircēji var apskatīt un pasūtīt preces, menedžeriem apstrādāt pasūtījumus, kā arī administratoriem pārvaldīt sistēmas iestatījumus. Zemāk attēloti dažādi skati, kas ir pieejami sistēmā.



5.5.1. att. Preču saraksta skats

Attēlā 5.5.1 ir redzams skats, kas ir pieejams jebkurai lietotājai – preču saraksts. Tas attēlo preču bildes, nosaukumus, cenu, un saīsināto aprakstu versiju (līdz 100 simboliem). Svarīgi norādīt, ka šeit preces cena klientam parādās, ieskaitot sistēmā iestatīto PVN. Uzspiežot uz jebkuru preci, atvērsies tās detalizētāka informācija, kā ir redzams attēlā 5.5.2.



Red LED shoes

Gender:UnisexUpper. Material:PUInsole. Material:PU. Feature:Lighted. Pattern Type:Solid. Outsole Material:Rubber. Closure Type:Lace-UpLining Material:PU.




Price (With VAT): € 24.20

Category: Clothes

Stock in warehouse Riga: 1

Stock in warehouse Liepaja: 13


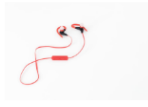

[ADD TO CART](#)

5.5.2. att. Preces skats

Šeit var redzēt jau pilnu preces aprakstu, cenu, kategoriju, preču daudzumu noliktavās, un visas pieejamās preces bildes. Šeit arī ir iespējams pievienot preci iepirkšanas grozam.

Shopping Cart

Item	Description	Price	Count	Total
	Child's Raccoon Tee This SoulCal Funnel Lined Knit is crafted with a fold over collar and full button fastening with lon...	€18.15	- 1 +	€18.15
	Earphones 4U 1. Listening to songs and calls, supporting listening to songs and making calls, 2. One to two conn...	€31.45	- 1 +	€31.45
	Red LED shoes Gender:UnisexUpper. Material:PUInsole. Material:PU. Feature:Lighted. Pattern Type:Solid. Outsole Mat...	€24.20	- 1 +	€24.20
SubTotal				€73.80
Shipping				€2.20
Total				€76.00

[CHECKOUT](#)

5.5.3. att. Preču groza skats

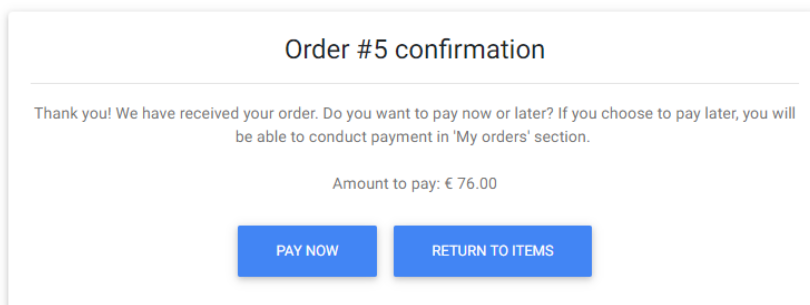
Preču grozā (att. 5.5.3) ir attēlots preču saraksts ar cenām un skaitu. Tiek aprēķināta arī piegādes maksa, ņemot vērā augstāko piegādes maksu no katras noliktavas. Skatā ir pogas, lai pievienotu vai samazinātu preču skaitu, kā arī veiktu pasūtījumu (Checkout).

The screenshot shows a dark navigation bar at the top with the text 'Ordering system' on the left, a search bar with the placeholder 'Search items...', and a list of menu items: 'Electronics', 'For home', 'Kids', 'Clothes', 'All Items', 'Cart (2)', 'Login', and 'Register'. Below the navigation bar is a large white box with the text 'Submit an order'. Underneath this box are several form fields: 'Your full name' with a sub-field 'Full name', 'E-mail' with a sub-field 'Your e-mail address', 'Phone number' with a sub-field 'Phone number', 'Shipping address' with a sub-field 'Shipping address', and 'Note' with a sub-field 'Note'. At the bottom of the form is a blue button labeled 'SUBMIT'. Below the button is a small line of text: 'By clicking "SUBMIT", you agree that a system user is automatically created for you, and your data is used for order processing purposes.'

5.5.4. att. Pasūtījuma izveides skats

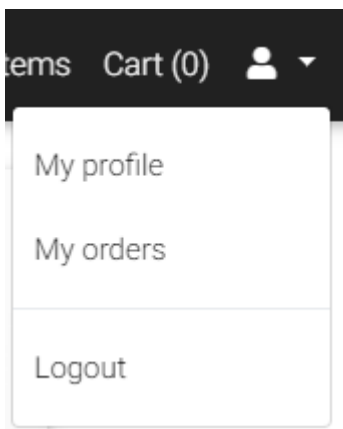
Lai izveidotu pasūtījumu (att. 5.5.4), ir jānorāda sava kontaktinformācija – pilnais vārds, e-pasts, telefona numurs, piegādes adrese, un komentāru, ja tas ir nepieciešams. Ja pasūtījums notiek no autentificēta lietotāja, šie lauki tiek aizpildīti automātiski ar autentificētā lietotāja informāciju. Ja pasūtījums notiek no neautentificēta lietotāja – pēc informācijas ievades un pasūtījuma izveides lietotājs tiks izveidots automātiski – par to arī tiek brīdināts lietotājs pirms “Submit” pogas nospiešanas. Pēc pasūtījuma izveides lietotājs tiek pārvirzīts uz pasūtījuma apstiprinājuma skatu (att. 5.5.5).

Order was succesfully submitted!



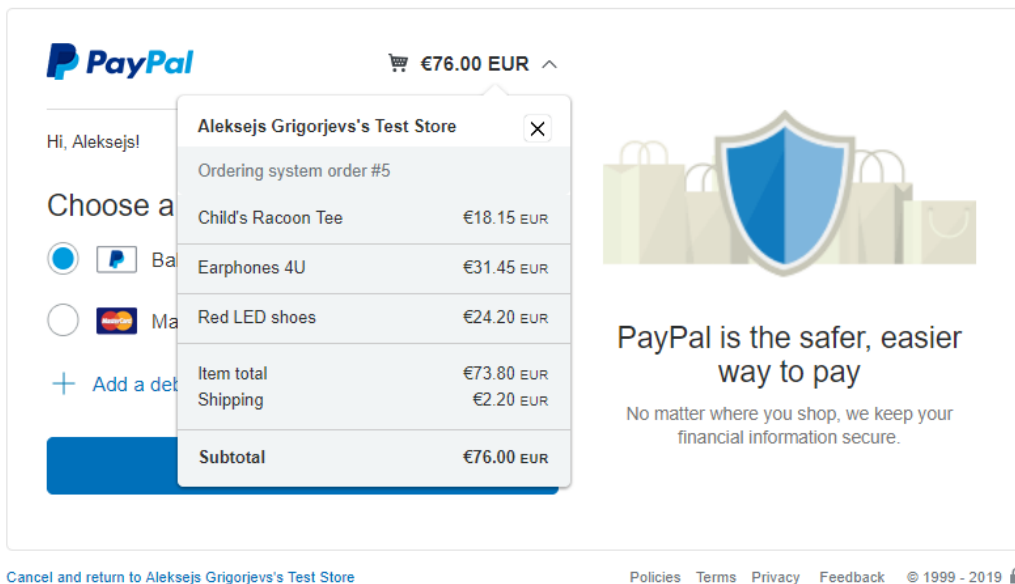
5.5.5. att. Pasūtījuma apstiprinājuma skats

Šajā skatā lietotājs saņem apstiprinājumu par pasūtījuma izveidošanu, un var izvēlēties apmaksāt pasūtījumu tagad, vai arī vēlāk. Apmaksāt pasūtījumu var jebkurā brīdī no sadaļas “My orders”, kas atrodama lietotāja izvēlnē (att 5.5.6).



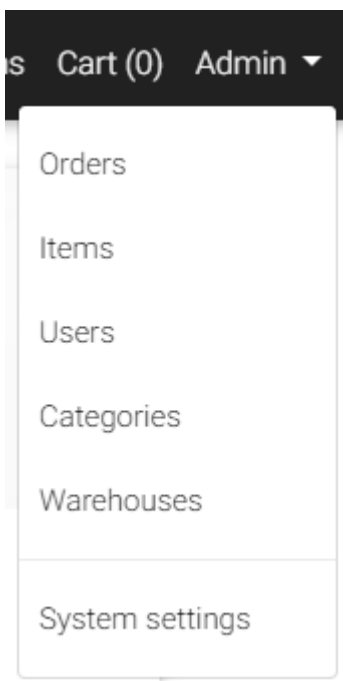
5.5.6. att. Lietotāja izvēlne

Izvēloties apmaksāt pasūtījumu, lietotājs tiek pārvirzīts uz maksājuma sistēmu PayPal (att. 5.5.7), kur viņam tiek aprēķināta maksājamā summa ar visām detaļām – preču nosaukumiem, skaitiem, kopsummām, un piegādes maksu. Lietotājs apmaksā pasūtījumu atbilstoši PayPal maksājuma procedūrai.



5.5.7. att. Maksājuma sistēmas PayPal skats

Pierakstoties sistēmā kā administrators vai menedžeris, tiek aktivizēta slēptā “Admin” izvēlne (att. 5.5.8), kas satur sistēmas pārvaldības funkcijas. Katrai lietotāju lomai ir pieejamas tikai savas opcijas.



5.5.8. att. Administratora izvēlnes funkcijas

Piemēram, sadaļā “Orders” (att. 5.5.9) ir redzami sistēmā veiktie pasūtījumi. Tos var apskatīt, rediģēt, izpildīt, dzēst, apmaksāt, ja pasūtījums ir izveidots uz autentificētā lietotāja. Pasūtījuma “izsūtīšana” ir pieejama tikai pasūtījumiem, kuri ir apmaksāti.

ID	Status	Name	Phone	E-mail	Total	Paid?	Shipped?	Show	Edit	Delete
6	closed	Zane	25946749	zane@example.com	€ 18.15	Yes	Yes	SHOW	EDIT	DELETE
7	cancelled	Jānis	25490605	janis@example.com	€ 44.55	No	No	SHOW	EDIT	DELETE
8	new	Olegs	26543698	olegs@example.com	€ 16.72	PAY	No	SHOW	EDIT	DELETE
9	pending	Brigita	29435696	brigita@example.com	€ 4.84	Yes	SHIP	SHOW	EDIT	DELETE

Showing 1 to 4 of 4 entries

Previous **1** Next

5.5.8. att. Pasūtījumu saraksta skats

Attēlā 5.5.9 ir dots piemērs ievades formām sistēmā. Attēlotā forma ir preces izveides forma, kur ietilpst gan statiski lauki, gan arī, piemēram, noliktavu lauki, kuri tiek ģenerēti katrai noliktavai sistēmā. Šajā formā ir pieejama arī vairāku bilžu augšupielāde.

The screenshot shows a web interface for an 'Ordering system'. At the top, there is a navigation bar with a search bar and menu items: Electronics, For home, Kids, Clothes, All Items, Cart (0), Admin, and a user profile icon. Below the navigation bar is a large white box with the text 'Add item'. The main form contains the following fields and controls:

- Name:** A text input field with the placeholder 'The product name'.
- Description:** A larger text input field with the placeholder 'Product description'.
- Price (Without VAT):** A text input field with a currency symbol '€' and the placeholder 'Product price without VAT'.
- Cost:** A text input field with a currency symbol '€' and the placeholder 'Product cost'.
- Warehouse Riga:** Two text input fields. The first has the placeholder 'Stock', and the second has a currency symbol '€' and the placeholder 'Shipping cost'.
- Warehouse Liepaja:** Two text input fields. The first has the placeholder 'Stock', and the second has a currency symbol '€' and the placeholder 'Shipping cost'.
- Category:** A dropdown menu with 'Electronics' selected.
- Add images:** Two file upload fields. Each has a 'Choose File' button and the text 'No file chosen'. The first field has a green 'ADD' button, and the second has a red 'REMOVE' button.
- SUBMIT:** A blue button at the bottom of the form.

5.5.9. att. Preces izveides forma

Attēlā 5.5.10 var apskatīt pieejamos sistēmas iestatījumus. Uz kvalifikācijas darba izstrādes brīdī tādas ir tikai trīs:

- Sistēmas nosaukums, kurš tiek izmantots lietotnē un visos izsūtāmajās e-pasta vēstulēs.
- PVN procentos, kurš tiek izmantots gala pircēju cenas noteikšanai.
- Piegādes uzcenojums – piegādes maksas uzcenojums procentos gala pircēju piegādes maksas aprēķināšanai.

Ordering system Search items... Electronics For home Kids Clothes All Items Cart (0) Admin

Edit system settings

System name

VAT in %

Shipping margin in %

SUBMIT

5.5.10. att. Sistēmas iestatījumu skats

6. TESTĒŠANAS DOKUMENTĀCIJA

Šajā nodaļā tiek aprakstīta pielietotā testēšanas procedūra izstrādātajai lietotnei. Visi testi ir automatizēti un tika izstrādāti, izmantojot PHPUnit ietvaru. Izstrādātie testi iedalās vienībtestos, kas pārbauda tikai konkrētās klases un konkrētās klases funkcijas, kā arī funkcionālajos testos, kas pārbauda visu komponentu (maršrutēšanu, apstrādi, izvadi, sesijas datus, autentifikāciju, u.c.) mijiedarbību. Testi tika izstrādāti tādā veidā, lai pārbaudītu katras funkcijas veiksmīgu darbību un atbilstību šajā dokumenta aprakstītajām prasībām. Visi testi ir izpildāmi atsevišķā testa vidē, ar savu datubāzi, netraucējot produkcijas sistēmai.

Izstrādes laikā kopumā ir izstrādāti 45 automatizēti testi, no kuriem 15 ir vienībtesti un 30 ir funkcionālie testi. Testu laikā tiek veiktas 346 pārbaudes (att. 6.1).

```
C:\wamp64\www\order_system>vendor\bin\phpunit
PHPUnit 7.5.8 by Sebastian Bergmann and contributors.

..... 45 / 45 (100%)

Time: 2.25 minutes, Memory: 32.00 MB

OK (45 tests, 346 assertions)
```

6.1. att. PHPUnit testu rezultāti

Vairākumam funkciju ir izstrādāti gan vienībtesti, gan funkcionālie testi, bet ir funkcijas, kurām vienībtesti nav pielietojami, un tāpēc dažas funkcijas tika notestētas tikai, izmantojot funkcionālos testus. Dažos funkcionālajos testos tika pārbaudītas vairākas funkcijas, kas ir savstārpēji savienotas. Testējot funkcijas, kurām ir nepieciešami ievaddati, tika izmantots Laravel ietvarā iekļautā bibliotēka Faker, kas ļauj viegli ģenerēt nejauši izvēlētus datus, tādā veidā testus padarot par efektīviem un neatkarīgiem. Pārsvārā tika pielietota t. s. “pozitīvā” testēšana – pārbaudot, vai viss izpildās, kā iecerēts, bet dažos testos tika pielietota arī “negatīvā” pieeja – pārbaudot, vai pie noteiktajiem kritērijiem sistēma izdos kļūdu, kā tam arī vajadzētu būt.

Zemāk, testi, tabulas veidā, ir uzskaitīti pa moduļiem, ar norādi par testa tipu, funkciju, kuru tas testē, kā arī testa aprakstu.

Lietotāju moduļa testi

Testa funkcijas nosaukums	Testa tips	Testa apraksts	Testējamās funkcijas ID
It_can_create_a_user	Vienībtests	Pārbauda, vai lietotājs tiek izveidots datubāzē, vai tā vērtības tiek uzdotas korekti, un vai piešķirtās lomas tiek pareizi ierakstītas datubāzes starptabulā.	3.2.1.5. UserCreate
It_can_update_the_user	Vienībtests	Pārbauda, vai rediģējamam lietotājam datubāzē tiek pareizi atjaunināti dati un lomas.	3.2.1.6. UserEdit
It_can_delete_the_user	Vienībtests	Pārbauda, vai lietotājs patiešām tiek dzēsts no datubāzes.	3.2.1.8. UserRemove
It_can_show_users	Funkcionālais tests	Pārbauda, vai tiek attēloti lietotāji tabulas veidā administratora lietotājam, un pārbauda, ka citu lomu lietotāji nevar piekļūt šim sarakstam.	3.2.1.4. UserIndex
It_can_create_user	Funkcionālais tests	Pārbauda, vai izveides forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc lietotāja izveides tiek izvadīts atbilstošs paziņojums.	3.2.1.5. UserCreate

It_can_edit_user	Funkcionālais tests	Pārbauda, vai rediģēšanas forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc lietotāja rediģēšanas tiek izvadīts atbilstošs paziņojums.	3.2.1.6. UserEdit
It_can_delete_user	Funkcionālais tests	Pārbauda, vai lietotāja dzēšana, izmantojot maršrutus, strādā un vai tiek izvadīts atbilstošs paziņojums.	3.2.1.8. UserRemove
It_can_generate_user_token	Funkcionālais tests	Pārbauda, vai lietotāja API marķieris tiek ģenerēts un attēlots skatā.	3.2.1.7. UserGenerateToken

6.2. Tabula

Kategoriju moduļa testi

Testa funkcijas nosaukums	Testa tips	Testa apraksts	Testējamās funkcijas ID
It_can_create_a_category	Vienībtests	Pārbauda, vai kategorija tiek izveidota datubāzē, un pārbauda, vai kategorijas dati ir ierakstīti korekti.	3.2.2.2. CategoryCreate
It_can_update_the_category	Vienībtests	Pārbauda, vai rediģējamai kategorijai datubāzē tiek pareizi atjaunināti dati.	3.2.2.3. CategoryEdit
It_can_delete_the_category	Vienībtests	Pārbauda, vai kategorija patiešām tiek dzēsta no datubāzes.	3.2.2.4. CategoryRemove
It_can_show_categories	Funkcionālais tests	Pārbauda, vai kategorijas izvēlnē tiek attēlotas	3.2.2.1. CategoryIndex

		korekti, un pārbauda, vai kategoriju administratīvais skats ir pieejams tikai administratora lietotājam.	
It_can_create_category	Funkcionālais tests	Pārbauda, vai izveides forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc kategorijas izveides tiek izvadīts atbilstošs paziņojums.	3.2.2.2. CategoryCreate
It_can_edit_category	Funkcionālais tests	Pārbauda, vai rediģēšanas forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc kategorijas rediģēšanas tiek izvadīts atbilstošs paziņojums.	3.2.2.3. CategoryEdit
It_can_delete_category	Funkcionālais tests	Pārbauda, vai kategorijas dzēšana, izmantojot maršrutus, strādā un vai tiek izvadīts atbilstošs paziņojums.	3.2.2.4. CategoryRemove

6.3. Tabula

Preču moduļa testi

Testa funkcijas nosaukums	Testa tips	Testa apraksts	Testējamās funkcijas ID
It_can_create_an_item	Vienībtests	Pārbauda, vai prece tiek izveidota datubāzē, pārbauda bilžu augšupielādes formu, un pārbauda, vai preces dati	3.2.3.1. ItemCreate

		datubāzē, tai skaitā informācija starptabulās ir ierakstīta korekti.	
It_can_update_the_item	Vienībtests	Pārbauda, vai rediģējamai precei datubāzē tiek pareizi atjaunināti dati.	3.2.3.2. ItemEdit
It_can_delete_the_item	Vienībtests	Pārbauda, vai prece patiešām tiek dzēsta no datubāzes.	3.2.3.3. ItemRemove
It_can_show_items	Funkcionālais tests	Pārbauda, vai visas preces tiek attēlotas, un skats satur vajadzīgo informāciju.	3.2.3.4. ItemIndex
It_can_show_items_in_category	Funkcionālais tests	Pārbauda, vai katrā kategorijā tiek attēlotas tikai tai kategorijai piederošās preces.	3.2.3.6. ItemIndex_Category
It_can_show_items_admin	Funkcionālais tests	Pārbauda, vai administratīvā preču informācija tiek attēlota korekti, un tā tiek attēlota tikai lietotājam ar administratora līmeņa tiesībām.	3.2.3.5. ItemIndex_Admin
It_can_show_specific_item	Funkcionālais tests	Pārbauda, vai tiek izvadīta korekta	3.2.3.7. ItemShow

		informācija, apskatot konkrētu preci.	
It_can_create_item	Funkcionālais tests	Pārbauda, vai izveides forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc preces izveides tiek izvadīts atbilstošs paziņojums.	3.2.3.1. ItemCreate
It_can_edit_item	Funkcionālais tests	Pārbauda, vai rediģēšanas forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc preces rediģēšanas tiek izvadīts atbilstošs paziņojums.	3.2.3.2. ItemEdit
It_can_delete_item	Funkcionālais tests	Pārbauda, vai preces dzēšana, izmantojot maršrutus, strādā un vai tiek izvadīts atbilstošs paziņojums.	3.2.3.3. ItemRemove

6.4. Tabula

Pasūtījumu moduļa testi

Testa funkcijas nosaukums	Testa tips	Testa apraksts	Testējamās funkcijas ID
It_can_create_an_order	Vienībtests	Pārbauda, vai pasūtījums tiek izveidots datubāzē, un	3.2.4.1. OrderCreate

		pārbauda, vai pasūtījuma dati datubāzē atbilst ievadītajiem.	
It_can_update_the_order	Vienībtests	Pārbauda, vai rediģējamam pasūtījumam datubāzē tiek pareizi atjaunināti dati.	3.2.4.7. OrderEdit
It_can_delete_the_order	Vienībtests	Pārbauda, vai pasūtījums patiešām tiek dzēsts no datubāzes.	3.2.4.8. OrderRemove
It_can_show_cart	Funkcionālais tests	Pārbauda, vai sistēma uzdod lietotāja sesijas datus.	3.2.4.2. CartIndex
It_can_add_to_cart	Funkcionālais tests	Pārbauda, vai informācija par precī un tās skaitu tiek pievienota lietotāja sesijā un ka pievienotā informācija ir pareiza.	3.2.4.3. CartAdd
It_can_remove_from_cart	Funkcionālais tests	Pārbauda, vai informācija par precī tiek noņemta no lietotāja sesijas.	3.2.4.4. CartRemove
It_can_show_orders	Funkcionālais tests	Pārbauda, vai pasūtījumu saraksts attēlo korektu informāciju, un vai piekļūstot sarakstam kā parasts lietotājs, tiek attēloti tikai šī konkrētā lietotāja pasūtījumi.	3.2.4.10. OrderIndex
It_can_show_specific_order	Funkcionālais tests	Pārbauda, vai tiek izvadīta korekta informācija, apskatot konkrētu pasūtījumu.	3.2.4.6. OrderShow
It_can_create_order	Funkcionālais tests	Pārbauda, vai izveides forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc pasūtījuma izveides tiek izvadīts atbilstošs paziņojums.	3.2.4.1. OrderCreate

It_can_edit_order	Funkcionālais tests	Pārbauda, vai rediģēšanas forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc pasūtījuma rediģēšanas tiek izvadīts atbilstošs paziņojums.	3.2.4.7. OrderEdit
It_can_delete_order	Funkcionālais tests	Pārbauda, vai pasūtījuma dzēšana, izmantojot maršrutus, strādā un vai tiek izvadīts atbilstošs paziņojums.	3.2.4.8. OrderRemove
Order_can_be_shipped	Funkcionālais tests	Pārbauda, ka informācija datubāzē tiek atjaunota korekti, un tiek izvadīts atbilstošs paziņojums.	3.2.4.9. OrderShipped
Order_can_be_confirmed	Funkcionālais tests	Pārbauda, vai tiek attēlota pareiza pasūtījuma dati un maksājamā summa.	3.2.4.5. OrderConfirm

6.5.Tabula

Maksājumu moduļa testi

Testa funkcijas nosaukums	Testa tips	Testa apraksts	Testējamās funkcijas ID
It_can_create_payment	Vienībtests	Pārbauda, ka sistēmā tiek izveidots jauns maksājuma ieraksts datubāzē ar maksājuma ID un notiek pāradresācija uz maksājuma sistēmu.	3.2.5.1. PaymentCreate

Noliktavu moduļa testi

Testa funkcijas nosaukums	Testa tips	Testa apraksts	Testējamās funkcijas ID
It_can_create_a_warehouse	Vienībtests	Pārbauda, vai noliktava tiek izveidota datubāzē, un pārbauda, vai noliktavas dati ir ierakstīti korekti.	3.2.6.2. WarehouseCreate
It_can_update_the_warehouse	Vienībtests	Pārbauda, vai rediģējamai noliktavai datubāzē tiek pareizi atjaunināti dati.	3.2.6.3. WarehouseEdit
It_can_delete_the_warehouse	Vienībtests	Pārbauda, vai noliktava patiešām tiek dzēsta no datubāzes.	3.2.6.4. WarehouseRemove
It_can_show_warehouses	Funkcionālais tests	Pārbauda, vai noliktavas un to detaļas tiek attēlotas korekti.	3.2.6.1. WarehouseIndex
It_can_create_warehouse	Funkcionālais tests	Pārbauda, vai izveides forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc noliktavas izveides tiek izvadīts atbilstošs paziņojums.	3.2.6.2. WarehouseCreate
It_can_edit_warehouse	Funkcionālais tests	Pārbauda, vai rediģēšanas forma tiek attēlota un strādā korekti, kā arī pārbauda, vai pēc noliktavas	3.2.6.3. WarehouseEdit

		rediģēšanas tiek izvadīts atbilstošs paziņojums.	
It_can_delete_warehouse	Funkcionālais tests	Pārbauda, vai noliktavas dzēšana, izmantojot maršrutus, strādā un vai tiek izvadīts atbilstošs paziņojums.	3.2.6.4. WarehouseRemove

6.7.Tabula

Sistēmas iestatījumu moduļa testi

Testa funkcijas nosaukums	Testa tips	Testa apraksts	Testējamās funkcijas ID
It_can_show_settings_page	Funkcionālais tests	Pārbauda, vai iestatījumu skats attēlo korektu informāciju, un ka tam var piekļūt tikai lietotājs ar administratora lomu, savukārt parastam lietotājam tiek izvadīts “401 Unauthorized”.	3.2.7.1. SettingsIndex
It_can_edit_settings	Funkcionālais tests	Pārbauda, vai iestatījumu vērtības tiek pareizi ierakstītas failā, un ka lietotājam tiek izvadīts atbilstošs paziņojums.	3.2.7.2. SettingsUpdate

7. PROJEKTA ORGANIZĀCIJA

Sistēma tika izstrādāta vienpersoniski, bez citu cilvēku līdzdalības, izmantojot PHP valodu, Laravel ietvaru, un MVC pieeju.

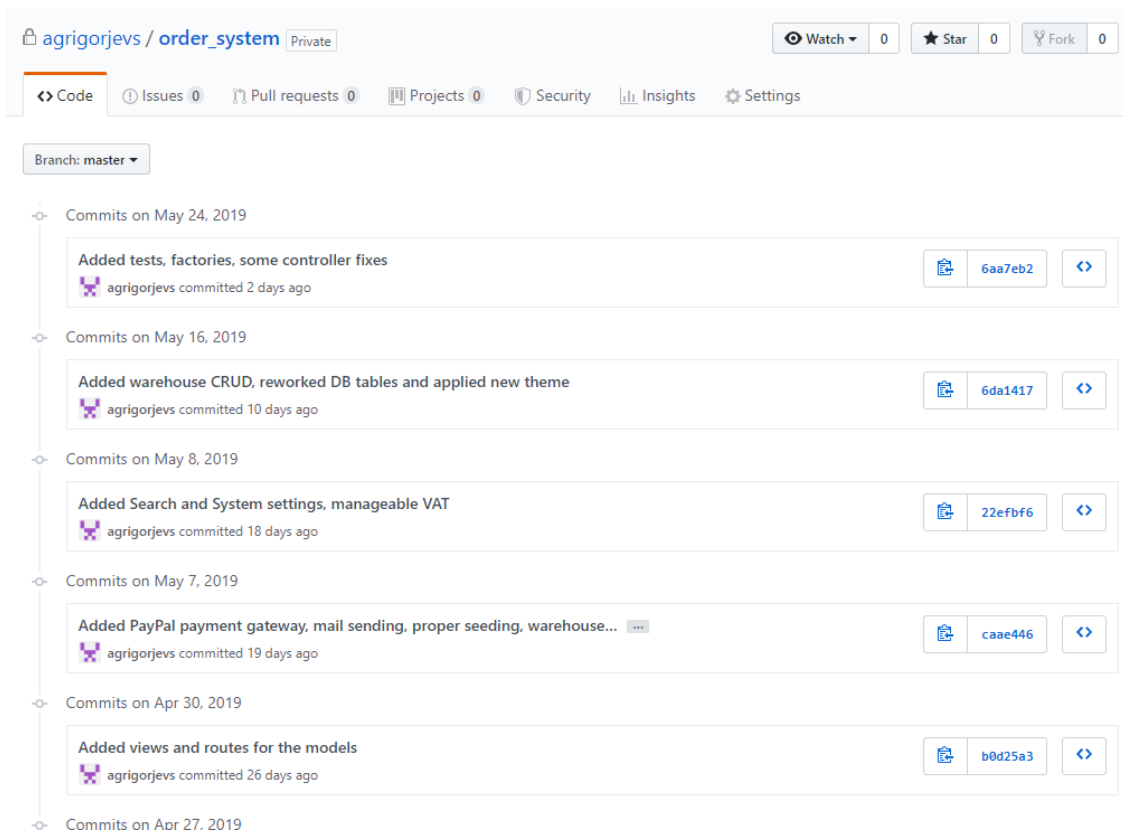
Projekta izstrādes sākumā bija zināma aptuvenā kopējā tehniskā risinājuma arhitektūra, galvenie sistēmas objekti, kuriem sistēmā ir jāpastāv, bet trūka saprašanas par konkrēto moduļu funkcijām un to mijiedarbību. Tika uzzīmēta principiāla risinājuma shēma. Projekts tika izstrādāts pakāpeniski, pa moduļiem, izstrādes laikā topot vairākām papildus idejām un funkcionalitātei, kuras tika ieviestas papildus sākotnējai idejai. Izstrādājot katru moduli, tika piedomāts par lietotāja ērtību un par to, lai sistēma būtu universāla un plaši pielietojama, līdz ar to mainoties arī sākotnēji iepļānotām funkcijām.

8. KVALITĀTES NODROŠINĀŠANA

Darbs tika izstrādāts, izmantojot PHP 7.1.9 valodu un Laravel 5.8 ietvaru. Lietotāja saskarnes izstrādei tika izmantots Bootstrap Material versijas 4.8.0. ietvars. Kods tika komentēts pēc iespējas sīkāk un saprotamāk, tika izmantoti Laravel versijas 5.8 ietvara izstrādātāju efektīvākās metodes kā modeļu injekcija, grupēšana, kas atvieglo gan koda uztveri, gan arī paša koda apjomu, kā arī koda strukturēšanas ieteikumi. Tika izmantota programmēšanas valodas labā prakse, jēgpilni funkciju un mainīgo nosaukumi. Izstrādājot sistēmu, tika stingri ievērotas visas ievaddatu pārbaudes, izmantojot lietotāja pārlūkprogrammas iespējas (front-end), gan servera pusē, kas pilnībā izslēdz nekorektu lietotāja ievadi sistēmā. Vienībtesti un funkcionālie testi tika izstrādāti, izmantojot PHPUnit 7.5.8 bibliotēkas. Visi testi ir automatizēti un pārbauda sistēmas funkciju pareizu darbību.

9. KONFIGURĀCIJU PĀRVALDĪBA

Programmatūras koda glabāšanai un versiju kontrolei tika izmantots versiju kontroles rīks “Git” un repozitoriju glabātuve “GitHub”, kas ļauj glabāt koda dažādas versijas un programmas atzarus. Pēc katras nopietnas funkcionalitātes pievienošanas un tās testēšanas, kods tika sinhronizēts repozitorijā.



9.1.att. GitHub projekta daļējs izmaiņu saraksts

Repozitorijā tiek glabāts tikai lietotnes kods, un composer.json fails, kas satur aprakstu par lietotnes atkarībām (pakotnēm), kas nepieciešamas lietotnes darbībai. Repozitorijā netiek glabātas pašas pakotnes, jo tās ir pieejamas lejupielādei caur Composer pakotņu menedžeri. Datubāzes un SMTP pieslēguma parametri tika aprakstīti, izmantojot vides mainīgos, līdz ar to nekādi parametri, kas atkarīgi no sākotnējās vides, netiek iekopēti repozitorijā, un lietotni ir viegli nodublēt uz jebkura cita servera. Visi iestatījumi, kas attiecas uz pašu sistēmu (PVN likme, uzcenojumi, utt.) tiek glabāti sistēmas konfigurācijā.

10. DARBIETILPĪBAS NOVĒRTĒJUMS

Darbietilpības novērtējums tika veikts, par pamatu nosakot pirmkoda rindiņu skaitu. PHP Pirmkoda rindiņu skaitīšana notika, izmantojot atvērtā pirmkoda programmu PHPLOC 5.0 no PHPUnit izstrādātāja, kas ir speciāli izstrādāts rīks PHP programkoda rindiņu skaitīšanai darbietilpības un statistikas vākšanai.

Koda skaitīšanā tika izslēgtas visas mapes un faili, kas satur Laravel un PayPal pakotnes, kā arī faili, kas tiek automātiski izveidoti Laravel projekta sākumā. Skaitīšana notika divos piegājos - ar lietotāja saskarnes (front-end) un servera puses (back-end) koda rindiņu skaitīšanu, un tikai servera puses koda rindiņu skaitīšanu. Šeit jāņem vērā, ka abos gadījumos tika skaitīts arī izstrādāto automatizēto testu koda rindiņas.

```
phploc 5.0.0 by Sebastian Bergmann.  
  
Directories          18  
Files                98  
  
Size  
Lines of Code (LOC)      5400  
Comment Lines of Code (CLOC) 1198 (22.19%)  
Non-Comment Lines of Code (NCLOC) 4202 (77.81%)  
Logical Lines of Code (LLOC) 1579 (29.24%)
```

10.1. att. PHPLOC 5.0 rīka rezultāts, analizējot servera puses kodu rindiņu skaitu

Attēlā 10.1. var redzēt statistiku par servera puses kodu. Tika uzrakstītas 5400 rindiņas koda, no kurām 1198, jeb 22% ir komentāru rindiņas, pārējās 4202 rindiņas ir koda rindiņas, no tām 1579 rindiņas jeb 29% rīks uzskatīja par loģiskām rindiņām.

```
phploc 5.0.0 by Sebastian Bergmann.  
  
Directories          23  
Files                132  
  
Size  
Lines of Code (LOC)      7559  
Comment Lines of Code (CLOC) 1198 (15.85%)  
Non-Comment Lines of Code (NCLOC) 6361 (84.15%)  
Logical Lines of Code (LLOC) 1579 (20.89%)
```

10.2. att. PHPLOC 5.0 rīka rezultāts, analizējot servera puses + lietotāja saskarnes kodu rindiņu skaitu

Analizējot kopējo lietotnes kodu, tad, kā var novērot 10.2. attēlā, kopējais rindiņu daudzums sasniedz 7559, un visu pieaugumu veido saskarnes kods. Līdz ar to, var secināt, ka lietotāja saskarnes kods sastāda 2159 rindiņas.

Pēc QSM (Quantitative Software Management), kas ir pasaulē atzītākā komercsabiedrība, kas nodarbojas ar programmatūras darbietilpības prognozēšanu (effort estimation), atklātā piekļuvē izliktās etalontabulas [8] ir secināms, ka vidēji 3.2 mēnešu ilgušam projektam ar izstrādes komandu vidēji 1.57 cilvēki, šāda projekta programkoda rindiņu vidējais skaits ir 1889. Līdz ar to var secināt, ka izstrādātā sistēma atbilst 3 cilvēkmēnešu ieguldījumam.

Lai darbietilpība tiktu novērtēta daudzpusīgi, tā tika aprēķināta arī pēc “PERT” [9] metodes. PERT (Program Evaluation and Review Technique) metode ir balstīta uz 3 izstrādes prognozēm:

- Optimiskā prognoze
- Reālā prognoze
- Pesimistiskā prognoze

Darbietilpību rēķina, šīs vērtības iekļaujot sekojošā formulā:

$$\frac{\text{Optimistiskā prognoze} + 4 * \text{Reālā prognoze} + \text{Pesimistiskā prognoze}}{6}$$

10.1. Tabula

Darbietilpības novērtējums pēc PERT metodes

Darbība	Optimiskā prognoze	Reālā prognoze	Pesimistiskā prognoze	PERT
Dokumentācija	10	14	17	13.83
Lietotāju modulis	3	5	9	5.33
Noliktavu modulis	2	4	7	4.16
Maksājumu modulis	5	7	9	7
Iestatījumu modulis	3	5	7	5

Pasūtījumu modulis	4	6	9	6.16
Preču modulis	5	8	12	8.16
Kategoriju modulis	2	4	7	4.16
API daļas izstrāde	4	6	10	6.33
Testēšana	4	6	10	6.33
Kopā	42	65	97	66.5

Tabulā 10.1. ir norādīts darbietilpības novērtējums cilvēkdienās. Novērtējumā ietilpa gan servera puses izstrāde konkrētajam moduļim, lietotāja saskarnes izstrāde, kā arī datubāzes projektējums.

Kopējā darbietilpība pēc PERT formulas ir 66.5 cilvēkdienas, kas ir apmēram 3 cilvēkmēneši, kas nozīmē, ka izstrādātā sistēma atbilst kvalifikācija darba prasībām.

SECINĀJUMI

Kvalifikācijas darba ietvaros tika veiksmīgi izstrādāta pasūtījumu pārvaldības sistēma, kā arī izstrādāta programmatūras prasību specifikācija, programmatūras projektējuma apraksts, un testēšanas dokumentācija.. Sistēmas izstrādes sākumā darba autoram nebija vērā ņemamas pieredzes strādājot ar PHP ietvariem. Izstrādājot sistēmu, darba autors ir padziļināti izpētījis “Laravel” ietvara funkcionalitāti un tā piedāvātās iespējas, kuras arī veiksmīgi ir pielietojis izstrādātajā sistēmā.

Tika iegūtas vērtīgas zināšanas par programmatūras vienību un funkcionālo testēšanu, un konkrēti izmantojot PHPUnit testēšanas ietvaru.

Projekta gaitā radās dažādi šķēršļi, taču visi tika veiksmīgi atrisināti. Izstrādātā sistēma tika veidota kā pielāgojama izmantošanai jebkurā nozarē un valstī. Nākotnē ir plānots attīstīt sistēmu un pievienot jaunu funkcionalitāti, atstājot pielāgojamību.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis [tiešsaiste – pārbaudīts 26.05.2019] Pieejams: <https://estudijas.lu.lv/mod/resource/view.php?id=131427>
2. LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai [tiešsaiste – pārbaudīts 26.05.2019] Pieejams: <https://estudijas.lu.lv/mod/resource/view.php?id=131428>
3. Laravel ietvara dokumentācija [tiešsaiste – pārbaudīts 26.05.2019] Pieejams: <https://laravel.com/docs/5.8/>
4. PHPUnit ietvara dokumentācija [tiešsaiste – pārbaudīts 26.05.2019] Pieejams: <https://phpunit.readthedocs.io/en/8.1/>
5. Bootstrap ietvara dokumentācija [tiešsaiste – pārbaudīts 26.05.2019] Pieejams: <https://getbootstrap.com/docs/4.1/>
6. Bootstrap Material ietvara dokumentācija [tiešsaiste – pārbaudīts 26.05.2019] Pieejams: <https://mdbootstrap.com/>
7. PayPal PHP SDK bibliotēkas [tiešsaiste – pārbaudīts 26.05.2019] Pieejams: <https://github.com/paypal/PayPal-PHP-SDK>
8. QSM Benchmark tables [tiešsaiste – pārbaudīts 26.05.2019] Pieejams: <http://www.qsm.com/resources/qsm-benchmark-tables>
9. “PERT” darbietplības metodes apraksts [tiešsaiste – pārbaudīts 26.05.2019] Pieejams: <https://www.slideshare.net/tomeh/program-evaluation-review-technique-pert>

PIELIKUMI

1. pielikums.

Lietotāja izveides funkcijas vienībtests

```
/** @test */
public function it_can_create_a_user()
{
    $password = $this
->faker->regexify('/^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*~]).{6,}$/')

    $roles = [];
    $all_roles = Role::all();

    foreach ($all_roles as $role) {
        $roles[] = $role->id;
    }

    $request = Request::create('/users', 'POST', [
        'fullname' => $this->faker->firstName,
        'username' => $this->faker->unique()->userName,
        'email' => $this->faker->unique()->safeEmail,
        'password' => $password,
        'password_confirmation' => $password,
        'address' => $this->faker->text(50),
        'phone' => $this->faker->e164PhoneNumber,
        'role' => $roles,
    ]);

    $controller = new UserController();
    $response = $controller->store($request);
    $this->assertEquals(302, $response->getStatusCode());

    $user = User::where('username', $request['username'])->first();

    $this->assertNotNull($user);
    $this->assertEquals($request['fullname'], $user->fullname);
    $this->assertEquals($request['username'], $user->username);
    $this->assertEquals($request['email'], $user->email);
    $this->assertEquals($request['address'], $user->address);
    $this->assertEquals($request['phone'], $user->phone);

    foreach ($request['role'] as $role_user)
        $this->assertDatabaseHas('role_user', ['user_id' => $user->id, 'role_id' => $role_user]);
}
```

```

/** @test */
public function it_can_create_user()
{
    $role = Role::where('name', 'admin')->get()->first();
    $user = factory(User::class)->create();
    $user->roles()->attach($role->id);

    // If it shows the create form
    $this
    ->actingAs($user)->get('/users/create')
    ->assertStatus(200)
    ->assertSee('Username')
    ->assertSee('Password')
    ->assertSee('Confirm password')
    ->assertSee('Full name')
    ->assertSee('E-mail address')
    ->assertSee('Phone')
    ->assertSee('Address');

    // If create works
    $password = $this
    ->faker->regexify('/^(?=.*?[A-Z]) (?=.*?[a-z]) (?=.*?[0-9]) (?=.*?[\#?!@$$%^&*~]).{6,}$/' );

    $roles = [];
    $all_roles = Role::all();

    foreach ($all_roles as $role) {
        $roles[] = $role->id;
    }

    $data = [
        'fullname' => $this->faker->firstName,
        'username' => $this->faker->unique()->userName,
        'email' => $this->faker->unique()->safeEmail,
        'password' => $password,
        'password_confirmation' => $password,
        'address' => $this->faker->text(50),
        'phone' => $this->faker->e164PhoneNumber,
        'role' => $roles,
    ];

    $this
    ->actingAs($user)
    ->post('/users/', $data)
    ->assertStatus(302)
    ->assertRedirect('/users')
    ->assertSessionHas('success', 'User created succesfully!');
}

```

3. pielikums.
Pasūtījuma izveides funkcija

```
public function store(Request $request)
{
    $data = $request->all();

    // Form validation rules
    $rules = array(
        'fullname' => 'required|string|max:100',
        'email' => 'required|email|max:50',
        'phone' => 'required|string|max:15',
        'address' => 'required|string|min:10'
    );

    $this->validate($request, $rules);

    /* If user not registered, a new one will be created,
    password generated and sent, and authenticated automatically */
    if (!Auth::check()) {
        $password = Str::random(16);
        $user = userCreate($data, $password);
        Mail::to($user)->send(new UserAutoCreated($user, $password));
        Auth::login($user);
    }

    $order = new Order();

    if (Auth::check())
        $order->user()->associate(User::findOrFail(Auth::id()));
    else $order->user()->associate($user);

    // Creating new order record in DB
    $order->fullname = $data['fullname'];
    $order->email = $data['email'];
    $order->address = $data['address'];
    $order->phone = $data['phone'];
    $order->total = session('total');
    $order->shipping = session('shipping');
    $order->note = $data['note'];
    $order->save();

    $deduct = $request->session()->get('deduct', array());

    // Attaching items for this order
    // deduct = array ( item => [warehouse => count] )
    foreach ($deduct as $item => $warehouses) {
        foreach ($warehouses as $warehouse => $count) {
            $order->items()->attach($item, ['item_quantity' => $count, 'warehouse_id' => $warehouse]);
        }
    }
    $order->save();

    // Sending notification to the user
    if (Auth::check())
        Mail::to($request->user()->send(new OrderSubmitted($order));
}
```

```
else
    Mail::to($user)->send(new OrderSubmitted($order));

// Cleaning session data
$request->session()->forget('cart', 'total', 'deduct', 'shipping');

$system_payment = new SystemPayment();
$system_payment->order()->associate($order);
$system_payment->save();

return redirect()->action('OrderController@confirmOrder', [
    'order' => $order
])->withSuccess('Order was successfully submitted!');
}
```

```

public function index(Request $request)
{
    // Cleaning session data of old values, in case there are
    $request->session()->forget('total', 'shipping', 'deduct');

    // Initializing variables
    $total = 0.0;
    $shipping = [];
    $deduct = [];
    $shipping_total = 0;

    // products array will go in the view
    $products = array();
    $current_cart = $request->session()->get('cart', array());

    // Looping through each item in the cart
    foreach ($current_cart as $id => $count)
    {
        if ($count > 0)
        {
            // Resetting previous product, if present
            $product = array();

            /* Default lowest shipping must be the highest possible
            to be overwritten by the first occasion */
            $lowest_shipping = 99999;

            // Variables for item count and shipping in previous warehouse if not enough
            $partial_quantity = 0;
            $partial_shipping = 0;

            $item = Item::findOrFail($id);

            // Determine lowest available shipping cost for specific item
            foreach ($item->warehouses as $warehouse) {
                if(!isset($shipping[$warehouse->id]))
                    $shipping[$warehouse->id]=0;

                $query = DB::table('item_warehouse')->where([
                    ['item_id', $item->id],
                    ['warehouse_id', $warehouse->id],
                    ['quantity', '>=', $count-$partial_quantity],
                ]->orderBy('shipping_cost', 'asc')->first();

                // If item with such quantity exists in any warehouse
                if($query) {

                    if($lowest_shipping > $query->shipping_cost) {

                        //Set lowest item shipping cost
                        $lowest_shipping = $query->shipping_cost;

                        /* Set warehouse item deduction array
                        (item => [warehouse => item count to deduct from that warehouse]) */
                    }
                }
            }
        }
    }
}

```

```

        if(isset($deduct[$item->id]))
            $deduct[$item->id] += array($warehouse->id => $count-$partial_quantity);
        else $deduct[$item->id] = array($warehouse->id => $count);

        // Set highest shipping cost from specific warehouse in this cart
        if ($shipping[$warehouse->id]<$lowest_shipping)
            $shipping[$warehouse->id] = $lowest_shipping;
    }

    // fix for items that can be gathered from more than 1 warehouse
} else if ($warehouse->pivot->quantity>0) {

    $partial_quantity += $warehouse->pivot->quantity;
    $partial_shipping += $warehouse->pivot->shipping_cost;

    // Set highest shipping cost from specific warehouse in this cart
    if ($shipping[$warehouse->id]<$partial_shipping)
        $shipping[$warehouse->id] = $partial_shipping;

    if(isset($deduct[$item->id]))
        $deduct[$item->id] += array($warehouse->id => $warehouse->pivot->quantity);
    else $deduct[$item->id] = array($warehouse->id => $warehouse->pivot->quantity);
}
}

$product['item'] = $item;
$product['count'] = $count;
$product['total'] = round($item->price *(1+config('settings.vat')*0.01),2) * $count;
$total += $product['total'];

$products[] = $product;
}
}

$warehouses = Warehouse::all();

// Calculating shipping total from highest warehouse shippings
foreach ($warehouses as $warehouse) {
    if (isset($shipping[$warehouse->id]))
        $shipping_total += round($shipping[$warehouse->id]*(1+config('settings.shipping_margin')*0.01),2);
}

// Fill session with data
$request->session()->put('total', $total);
$request->session()->put('shipping', $shipping_total);
$request->session()->put('deduct', $deduct);

return view('cart_show', array('products' => $products));
}

```

Kvalifikācijas darbs „*Pasūtījumu pārvaldības sistēma*” izstrādāts Latvijas Universitātes Datorikas fakultātē.

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

Autors: *Aleksejs Grigorjevs* _____ .05.2019.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *Dr. dat. Jānis Zuters* _____ .05.2019.

Recenzents: *Agnis Škuškovniks*

Darbs iesniegts 27.05.2019.

Kvalifikācijas darbu pārbaudījumu komisijas sekretāre: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2019. prot. Nr. _____

Komisijas sekretārs(-e): _____