

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

## **Preču cenu uzraudzības sistēma**

KVALIFIKĀCIJAS DARBS

Autors: **Oskars Kalniņš**

Studenta apliecības numurs: ok13015

Darba vadītājs: prof. Guntis Arnicāns

RĪGA 2015

# ANOTĀCIJA

Preču cenu uzraudzības sistēma “WebApiPrice” ir tīmekļa lietojumprogrammas saskarne, kas ļauj lietotājam uzzināt sev interesējošu produktu cenas lielveikalos neizkustoties no mājām. Sākotnējā sistēmas versija paredzēta tīmekļa klient programmatūras izstrādātājiem, lai varētu to piedāvāt plašākai auditorijai. Sistēma ļauj lietotājam apskatīt dažādu kategoriju produktu cenas veikalos un izveidot savu produktu grozu, lai piekļūtu sev interesējošiem produktiem vēl ātrāk. Sistēma arī piedāvā iespēju lietotājam pievienot jaunus produktus, atjaunot produktu cenas, un izveidot jaunas produktu kategorijas, tādējādi uzlabojot sistēmas funkcionalitāti.

**Atslēgas vārdi:** tīmekļa lietojumprogrammas saskarne, produktu cenas, produktu grozu veidošana, ASP.NET Web API 2.2, C#, .NET

## **ABSTRACT**

Monitoring system for prices of consumer goods “WebApiPrice” is web API that allows user to learn prices of products that they are interested in in supermarkets without moving from home. First version of system is for developers of web clients so that the system could be offered to wider auditory. System allows user to examine prices for products of various categories in various shops and create their own product carts so that they can access their favourite products even faster. System also allows users to add new products, refresh product prices and create new product categories, thus adding to functionality of the system.

**Key words:** web api, product prices, creating of product carts, ASP.NET Web API 2.2, c#, .NET

## CONTENTS

1	APZĪMĒJUMU SARAKSTS .....	1
2	IEVADS.....	2
3	PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA.....	3
3.1	Ievads .....	3
3.1.1	Nolūks.....	3
3.1.2	Darbības sfēra .....	3
3.1.3	Dokumenta pārskats.....	3
3.2	Vispārējs apraksts.....	3
3.2.1	Produkta perspektīva.....	3
3.2.2	Produkta funkcijas .....	3
3.2.3	Lietotāja raksturozīmes .....	4
3.2.4	Vispārējie ierobežojumi .....	4
3.3	Biznesa terminu skaidrojums .....	4
3.4	Lietotāju stāsti .....	5
3.4.1	Reģistrēšanās sistēmā lietotāja stāsti .....	5
3.4.2	Produktu pārvaldības lietotāju stāsti .....	7
3.4.3	Cenu pārvaldības lietotāju stāsti .....	12
3.4.4	Saziņas lietotāju stāsti .....	13
3.4.5	Administrācijas lietotāju stāsti .....	15
3.5	Lietotāju grupas.....	20
4	PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS .....	21
4.1	Ievads .....	21
4.1.1	Nosaukums.....	21
4.1.2	Autors.....	21
4.1.3	Dokumenta izveidošanas datums.....	21
4.1.4	Dokumenta nolūks .....	21
4.1.5	Darbības sfēra .....	21
4.1.6	Saistība ar citiem dokumentiem.....	21
4.2	Datu plūsmu diagrammas.....	22
4.2.1	Nultā līmeņa datu plūsmas diagramma.....	22
4.2.2	Pirmā līmeņa datu plūsmas diagramma .....	23
4.2.3	Otrā līmeņa datu plūsmas diagrammas .....	24
4.3	Moduļu dekompozīcija.....	29

4.3.1	Modulis Reģistrēšanās .....	29
4.3.2	Modulis Produkti .....	32
4.3.3	Modulis Cenas .....	40
4.3.4	Modulis Saziņa.....	42
4.3.5	Modulis Administrācija .....	46
4.4	Datu atkarības.....	52
4.4.1	Konceptuālais ER modelis.....	52
4.4.2	Fiziskais ER modelis.....	53
4.5	Datu dekompozīcija.....	54
4.5.1	Tabulas “User” lauki.....	54
4.5.2	Tabulas “Class” lauki.....	55
4.5.3	Tabulas “Message” lauki .....	55
4.5.4	Tabulas “Category” lauki.....	57
4.5.5	Tabulas “Type” lauki .....	58
4.5.6	Tabulas “Product” lauki.....	59
4.5.7	Tabulas “Cart” lauki .....	60
4.5.8	Tabulas “Shop” lauki .....	60
4.5.9	Tabulas “Price” lauki .....	61
4.6	Lietotnes ieejas punkti.....	63
4.7	Programmatūras koda struktūra .....	67
4.7.1	Mapes WebApiPrice saturs.....	67
4.7.2	Mapes WebApiPrice\src saturs .....	67
4.7.3	Mapes WebApiPrice\src\WebApiPrice.Common saturs .....	68
4.7.4	Mapes WebApiPrice\src\WebApiPrice.Data saturs.....	70
4.7.5	Mapes WebApiPrice\src\WebApiPrice.Data.SqlServer saturs.....	73
4.7.6	Mapes WebApiPrice\src\WebApiPrice.Web.Api saturs.....	75
4.7.7	Mapes WebApiPrice\src\WebApiPrice.Web.Api.Models saturs.....	86
4.7.8	Mapes WebApiPrice\src\WebApiPrice.Web.Common saturs.....	87
4.7.9	Mapes WebApiPrice\src\WebApiPriceDb saturs .....	89
5	TESTĒŠANAS DOKUMENTĀCIJA .....	91
5.1	Ievads .....	91
5.2	Moduļu testēšana.....	91
6	PROJEKTA ORGANIZĀCIJA .....	102
7	KONFIGURĀCIJAS PĀRVALDĪBA .....	103

8	KVALITĀTES NODROŠINĀŠANA .....	104
9	DARBETILPĪBAS NOVĒRTĒŠANA.....	105
9.1	Darbietilpības novērtējums .....	105
9.2	Secinājumi.....	105
10	NOBEIGUMS.....	106
11	IZMANTOTĀ LITERATŪRA.....	107

# 1 APZĪMĒJUMU SARAKSTS

.NET – kompānijas “Microsoft” radīts programmatūras ietvars.

Atkarību injicēšana – programmatūras izstrādes modelis, kas implementē programmatūras bibliotēku vadības inversiju.

Agile – spējā programmatūras izstrādes metode

AutoMapper – bibliotēka, kura kartē vienu objektu citā.

Blockchain – decentralizētas datu bāzes mehānisms. Šāds mehānisms tiek izmantots pakalpojumā *Bitcoin*.

C# - kompānijas Microsoft izstrādāta daudzparadigmu programmēšanas valoda.

Domēna modelis – datu bāzes tabulu lauku reprezentācija lietotnē.

log4net – bibliotēka, kura realizē lietotāju darbību reģistrēšanu dažādos formātos.

NHibernate – atvērta pirmkoda objektu attiecību kartētājs .NET ietvaram.

Ninject – atkarību injicēšanas rīks .NET ietvaram.

Reģistrēšana – sistēmas darbību ierakstīšana.

REST – programmatūras arhitektūras stils, kas nosaka vadlīnijas un labo praksi mērogojamu tīmekļa pakalpojumu radīšanai.

Servisa modelis – datu modelis, kas tiek izmantots datu reprezentācijai lietotnes iekšējos procesos.

Web Api – tīmekļa lietojumprogrammas saskarne.

## 2 IEVADS

Kvalifikācijas darba pamatā bija idejas piedāvājums no darba vadītāja par paregulējošas sistēmas izveidi. Pēc kopējas apspriedes tika nolemts veidot cenu uzraudzības sistēmu. Ideja par paregulējošu sistēmu man šķita interesanta tāpēc arī izvēlējos šo tēmu.

Izstrādātajai sistēmas versijai galvenās funkcijas ir: preču cenu pievienošana, vēlmju grozu izveide un statistikas apskate.

Darba mērķis bija izstrādāt sistēmas pirmo versiju, uz kuras pamata varētu mēģināt veidot paregulējošu sistēmu un tālākā nākotnē mēģināt sistēmu pārnest uz decentralizētu datu bāzi izmantojot *blockchain* mehānismu.

Sistēma tika izstrādāta izmantojot *agile* metodoloģiju. Šāds lēmums tika pieņemts, jo man nav bijusi pieredze tīmekļa lietotņu izstrādē un *agile* pieļauj programmatūras prasību specifikācijas izmaiņas izstrādes procesā.

Sistēmai netika izveidota grafiskā lietotāja saskarne, tāpēc, lai testētu izveidoto sistēmu tiek izmantots rīks “Fiddler4”

## **3 PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA**

### **3.1 Ievads**

#### **3.1.1 Nolūks**

Programmatūras prasību specifikācijas (PPS) nolūks ir apkopot un aprakstīt sistēmas WebApiPrice funkcionalitāti, prasības un specifikāciju.

Šis dokuments ir paredzēts programmatūras izstrādātājiem, kuri veiks sistēmas projektējuma un paša produkta izstrādi.

#### **3.1.2 Darbības sfēra**

Programmprodukts WebApiPrice ir tīmekļa lietojumprogrammas saskarne pakalpojumam, kurš glabā patērētāju produktu cenas veikalos, ļauj lietotājiem tās atjaunot, pievienot jaunus produktus un izveidot vēlmju grozus ar kuru palīdzību lietotājs varētu atrast veikalu, kurā tam ir visizdevīgāk iepirkties.

Programmprodukta mērķa auditorija ir cilvēki, kuri regulāri iepērkas pārtikas veikalos.

#### **3.1.3 Dokumenta pārskats**

Dokuments sastāv no sadaļām:

1. **Ievads** – iepazīstina ar turpmāk izstrādāto produktu un apraksta dokumenta struktūru un mērķi.
2. **Vispārējs apraksts** – apraksta izstrādājamā produkta galvenās funkcijas, potenciālo produkta lietotāju, kā arī ierobežojums produkta lietošanai.
3. **Biznesa terminu skaidrojums** - skaidro dokumentā lietoto terminoloģiju.
4. **Lietotāju stāsti** – neformāli aprakstītas funkcionālās prasības no sistēmas lietotāja viedokļa, kā arī apraksta nosacījumus, kuru izpildes gadījumā var uzskatīt, ka funkcionalitāte ir ieviesta.
5. **Lietotāju grupas** – apraksta lietotāju grupas, kādas tiek izmantotas sistēmā.

### **3.2 Vispārējs apraksts**

#### **3.2.1 Produkta perspektīva**

Izstrādātais produkts WebApiPrice ir tikai produkta pirmās versijas servera daļa.

#### **3.2.2 Produkta funkcijas**

Detalizēts funkciju apraksts ir atrodams nodaļā 3.3 “Lietotāju stāsti”:

- Lietotāju reģistrēšanās – nodrošina lietotājam iespēju reģistrēties sistēmā.

- Administrēšana – nodrošina lietotājam ar lomu administrators meklēt, paaugstināt vai pazemināt lietotāja klasi vai bloķēt lietotāju.
- Cenu pievienošana – nodrošina lietotājam iespēju pievienot produkta cenu sistēmā.
- Preču meklēšana – nodrošina iespēju lietotājam iespēju meklēt preces.
- Lietotāju meklēšana - nodrošina iespēju administratoriem meklēt citus reģistrētus lietotājus.
- Preču groza izveidošana – Lietotājs var izveidot grozu ar precēm, kuras viņš vēlas iegādāties un sistēma atgriež viņam, veikalus ar groza cenām sakārtotus dilstošā secībā pēc kopējās groza cenas veikalā.
- Cenu pievienošana izmantojot čeku – Lietotājs var nobildēt čeku un sistēma no šī čeka automātiski pievieno cenas precēm, kuras ir redzamas čekā.
- Preču atrašana izmantojot svītrkodu – lietotājs var atrast precī sistēmā noskenējot tās svītrkodu.
- Cenu precizitātes noteikšana – sistēma, kas palīdz noteikt cenas precizitāti parādot lietotājam populārāko cenu konkrētajā datumā.
- Negodīgo lietotāju atrašana – sistēma, kas atlasa lietotājus, kuri regulāri negodprātīgi izmanto sistēmu (norāda nepareizas cenas).
- Saziņa - lietotāji var sazināties savā starpā, kā arī sazināties ar administrāciju

### **3.2.3 Lietotāja raksturiezīmes**

Sistēmas lietotājs ir produkta klienta daļas izstrādātājs, kas datoru izmanto programmatūras radīšanai. Šīs darbības veikšanai viņi izmanto tam pielāgotu programmatūru, piemēram, “NetBeans”, “Visual Studio”, “Fiddler”.

### **3.2.4 Vispārējie ierobežojumi**

Sistēmas lietošanai ir nepieciešams dators ar piekļuvi internetam un programmatūra ar kuru ir iespējams nosūtīt HTTP pieprasījumus.

## **3.3 Biznesa terminu skaidrojums**

Cenas tips – norāda kāds cenas tips ir produktam. Sistēmā ir divi cenu tipi – cena gabalā un cena kilogramā.

Kategorija – līdzīgu preču kopums, kas ieviests, lai atvieglotu preču pārskatīšanu, piemēram, “Saldējums”.

Lietotājs – programmatūra vai cilvēks, kas lieto sistēmu.

Lietotāja klase – apzīmē lietotāja klasi sistēmā. Atkarībā no klases lietotājam autentifikācijas procesā tiek piešķirtas noteiktas lomas. Piemēram, lietotājam ar klasi “Administrator” tiek piešķirtas lomas “User” un “Administrator”

Lietotāja loma – lietotājam tiek piešķirtas lomas sistēmā atkarībā no tā klases. Lietotāja loma tiek izmantota autorizācijas procesā, lai noteiktu vai lietotājam ir tiesības izsaukt metodi.

Prece/produkts – ieraksts datu bāzē, kas apraksta kādu dzīvē eksistējošu produktu piemēram “TIO ar plombīra garšu”.

Preču grozs – preču kopa, ko lietotājs ir pievienojis saviem īsceļiem.

Veikals – ieraksts datu bāzē, kas apzīmē veikalu dzīvē.

Ziņa – ziņojums, ar kura palīdzību lietotāji var sazināties savā starpā.

### 3.4 Lietotāju stāsti

Lietotāju stāstu noformēšanas paraugs redzams tabulā 3.1.

Tabula 3.1

#### Lietotāja stāsta paraugs

<b>Identifikators</b>	Lietotāja stāsta identifikators
<b>Nosaukums</b>	Lietotāja stāsta nosaukums
<b>Stāsts</b>	Stāsta veidā neformāli tiek aprakstīta nepieciešamā funkcionalitāte no attiecīgā lietotāja skata punkta.
<b>Detāļas</b>	Tiek detalizēti aprakstīta nepieciešamā funkcionalitāte no izstrādātāja skata punkta.
<b>Akceptēšanas kritēriji</b>	Tiek definēti kritēriji, kuru izpildes gadījumā var uzskatīt, ka lietotāja stāsts ir realizēts.

#### 3.4.1 Reģistrēšanās sistēmā lietotāja stāsti

3.4.1.1 Jauna lietotāja reģistrēšanās lietotāja stāsts

Tabula 3.2

<b>Identifikators</b>	US_REG_01
<b>Nosaukums</b>	Lietotāja reģistrācija.
<b>Stāsts</b>	Es kā neregistrēts sistēmas lietotājs, varu reģistrēties sistēmā, lai varētu sākt lietot tās funkcionalitāti.

<b>Detālas</b>	Lietotājam, lai reģistrētos sistēmā ir jānosūta sistēmai reģistrēšanās pieprasījums ar sekojošiem datiem: <ul style="list-style-type: none"> <li>• Lietotāja vārds [obligāts, vismaz 3 simbolus garš, maksimālais garums 30 simboli, datu bāzē vēl neeksistē];</li> <li>• E-pasta adrese [obligāta, korekta, datu bāzē vēl neeksistē];</li> <li>• Parole [obligāta vismaz 6 simbolus gara, maksimālais garums 30 simboli].</li> </ul>
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas, sistēma pārbauda ievadīto datu atbilstību prasībām. Ja ievadītie dati atbilst prasībām, tad lietotājs tiek saglabāts sistēmā un tiek autentificēts sistēmā.

#### 3.4.1.2 Lietotāja autentifikācijas lietotāja stāsts

Tabula 3.3

<b>Identifikators</b>	US_REG_02
<b>Nosaukums</b>	Lietotāja autentifikācija.
<b>Stāsts</b>	Es kā reģistrēts sistēmas lietotājs, varu autentificēties sistēmā, lai lietu tās funkcionalitāti.
<b>Detālas</b>	Lietotājam, lai autentificētos sistēmā ir jānosūta sistēmai autentifikācijas pieprasījums ar sistēmā jau eksistējoša lietotāja vārda un paroles kombināciju.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas, sistēma pārbauda ievadīto datu atbilstību prasībām. Ja ievadītie dati atbilst prasībām, tad lietotājs tiek autentificēts sistēmā un tam tiek atgriezti viņa dati.

#### 3.4.1.3 Lietotāja datu rediģēšanas lietotāja stāsts

Tabula 3.4

<b>Identifikators</b>	US_REG_03
<b>Nosaukums</b>	Lietotāja datu rediģēšana.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu izmainīt savus datus, lai atjaunotu savu informāciju vai izlabotu kļūdaini ievadīto informāciju par sevi.

<b>Detālas</b>	Lietotājam, lai izmainītu savus datus ir jānosūta sistēmai lietotāju datu rediģēšanas pieprasījums ar atjaunotiem lietotāja datiem
<b>Akceptēšanas kritēriji</b>	<p>Pēc pieprasījuma saņemšanas sistēma pārbauda:</p> <ul style="list-style-type: none"> <li>• vai lietotājs ir autentificēts;</li> <li>• vai nodotais identifikators ir vienāds ar autentificētā lietotāja identifikatoru;</li> <li>• vai ievadītie dati ir sistēmā unikāli.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad sistēma atzīmē, ka lietotājs ir izmainījis attiecīgo lietotāja ierakstu datu bāzē un lietotājam tiek parādīti viņa jaunie dati.</p>

### 3.4.2 Produktu pārvaldības lietotāju stāsti

#### 3.4.2.1 Jauna produkta izveidošanas lietotāja stāsts

Tabula 3.5

<b>Identifikators</b>	US_PROD_01
<b>Nosaukums</b>	Jauna produkta izveidošana.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu izveidot jaunu produktu, lai varētu uz šo produktu izmantot sistēmas funkcionalitāti.
<b>Detālas</b>	<p>Lietotājam, lai izveidotu jaunu produktu ir sistēmai jānosūta produkta izveidošanas pieprasījums ar šādām detaļām:</p> <ul style="list-style-type: none"> <li>• ievadīts produkta nosaukums [obligāts, garumā no 3 līdz 30 simbolu garš];</li> <li>• ievadīta sistēmā eksistējoša produkta kategorija;</li> <li>• ievadīts sistēmā eksistējošs produkta cenas tips.</li> </ul>
<b>Akceptēšanas kritēriji</b>	<p>Pēc pieprasījuma saņemšanas sistēma pārbauda:</p> <ul style="list-style-type: none"> <li>• vai ievadītie dati ir atbilstoši sistēmas prasībām;</li> <li>• vai ievadītā kategorijas, nosaukuma un cenas tipa kombinācija ir unikāla.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad lietotājam tiek parādīts viņa izveidotais produkts.</p>

### 3.4.2.2 Produkta meklēšanas sistēmā lietotāja stāsts

Tabula 3.6

<b>Identifikators</b>	US_PROD_02
<b>Nosaukums</b>	Produkta meklēšana sistēmā.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu meklēt sistēmā eksistējošus produktus, lai pievienotu apskatītu eksistējoša produkta cenu vai pārbaudītu vai konkrētais produkts sistēmā eksistē.
<b>Detāļas</b>	Lietotājam, lai meklētu sistēmā produktu, ir jānosūta sistēmai meklēšanas pieprasījums ar meklēšanas parametriem.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas, sistēma atgriež visus produktus, kuru nosaukumā ir ievadītā simbolu virkne.

### 3.4.2.3 Produkta cenu apskatīšanas lietotāja stāsts

Tabula 3.7

<b>Identifikators</b>	US_PROD_03
<b>Nosaukums</b>	Produkta cenu apskatīšana.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu apskatīt sistēmā izveidotam produktam aktuālās cenas.
<b>Detāļas</b>	Lietotājam, lai apskatītu produkta aktuālās cenas sistēmā, ir jāpieprasa sistēmai konkrētais produkts.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma lietotājam izvada visas aktuālās cenas produktam veikalos, kuros produkts sistēmā ir ievadīts.

### 3.4.2.4 Vienas kategorijas produktu sistēmā apskatīšanas lietotāju stāsti

Tabula 3.8

<b>Identifikators</b>	US_PROD_04
<b>Nosaukums</b>	Vienas kategorijas produktu apskatīšana.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu apskatīt visus noteiktā kategorijā eksistējošus produktus, lai meklētu produktus pēc noteiktas kategorijas.

<b>Detālas</b>	Lietotājam, lai apskatītu visus produktus vienā kategorijā eksistējošus produktus ir jānosūta sistēmai pieprasījums apskatīt konkrēto kategoriju.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas, sistēma pārbauda vai kategorija sistēmā eksistē, un, ja tā eksistē, atgriež lietotājam visus produktus, kuri eksistē šajā kategorijā.

#### 3.4.2.5 Produkta rediģēšanas lietotāja stāsts

Tabula 3.9

<b>Identifikators</b>	US_PROD_05
<b>Nosaukums</b>	Produkta rediģēšana.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu rediģēt sevis izveidotu produktu, lai izlabotu izveidošanas procesā pieļautās kļūdas.
<b>Detālas</b>	Lietotājam, lai rediģētu produktu ir jānosūta sistēmai pieprasījums mainīt sevis izveidotu produktu ar rediģētām produkta detaļām.
<b>Akceptēšanas kritēriji</b>	<p>Pēc pieprasījuma saņemšanas sistēma pārbauda:</p> <ul style="list-style-type: none"> <li>• vai rediģētās detaļas atbilst sistēmas prasībām;</li> <li>• produkta nosaukuma, kategorijas un cenu tipa kombinācija ir sistēmā unikāla.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad sistēma atzīmē, ka lietotājs ir izmainījis attiecīgo produkta ierakstu datu bāzē un lietotājam tiek parādīts produkts ar izlabotajām detaļām.</p>

#### 3.4.2.6 Kategorijas pievienošanas lietotāja stāsts

Tabula 3.10

<b>Identifikators</b>	US_PROD_06
<b>Nosaukums</b>	Kategorijas pievienošana.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu izveidot sistēmā jaunu kategoriju, lai varētu sistēmai pievienot šīs kategorijas preces.
<b>Detālas</b>	Lietotājam, lai izveidotu jaunu kategoriju ir jānosūta sistēmai jaunas kategorijas izveidošanas pieprasījumu ar kategorijas

	nosaukumu [obligāts, sistēmā neeksistējošs, garumā no 3 līdz 30 simboliem garš].
<b>Akceptēšanas kritēriji</b>	<p>Pēc pieprasījuma saņemšanas sistēma pārbauda:</p> <ul style="list-style-type: none"> <li>• vai kategorijas nosaukums atbilst sistēmas prasībām;</li> <li>• vai kategorija sistēmā jau neeksistē.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad lietotājam tiek parādīta viņa izveidotā kategorija.</p>

#### 3.4.2.7 Kategorijas rediģēšanas lietotāja stāsts

Tabula 3.11

<b>Identifikators</b>	US_PROD_07
<b>Nosaukums</b>	Kategorijas rediģēšana.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu rediģēt sevis izveidotu kategoriju, lai izlabotu kategorijas veidošanas procesā radušās kļūdas.
<b>Detālas</b>	Lietotājam, lai rediģētu sevis izveidotu kategoriju, ir jānosūta kategorijas rediģēšanas pieprasījums ar rediģētām kategorijas detaļām.
<b>Akceptēšanas kritēriji</b>	<p>Pēc pieprasījuma saņemšanas sistēma pārbauda:</p> <ul style="list-style-type: none"> <li>• vai lietotājs, kurš izmaina kategoriju, pats ir izveidojis šo kategoriju;</li> <li>• vai izmainītā kategorija atbilst sistēmas prasībām.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad sistēma atzīmē, ka lietotājs ir izmainījis attiecīgo kategorijas ierakstu datu bāzē un lietotājam tiek parādīta viņa rediģētā kategorija.</p>

#### 3.4.2.8 Produkta pievienošanas grozam lietotāja stāsts

Tabula 3.12

<b>Identifikators</b>	US_PROD_08
<b>Nosaukums</b>	Produkta pievienošana grozam
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu pievienot produktu savam grozam, lai varētu piekļūta tam ātrāk.

<b>Detālas</b>	Lietotājam, lai pievienotu produktu savam grozam ir jānosūta sistēmai produkta pievienošanas grozam pieprasījums ar produkta datiem.
<b>Akceptēšanas kritēriji</b>	<p>Pēc pieprasījuma saņemšanas sistēma pārbauda:</p> <ul style="list-style-type: none"> <li>• vai lietotājs ir autentificējies</li> <li>• vai produkts eksistē datu bāzē;</li> <li>• vai pievienojamā produkta un lietotāja kombinācija ir unikāla.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad sistēma pievieno ierakstu datu bāzē un lietotājam tiek parādīts viņa grozs.</p>

#### 3.4.2.9 Produkta dzēšanas no groza lietotāja stāsts

*Tabula 3.13*

<b>Identifikators</b>	US_PROD_09
<b>Nosaukums</b>	Produkta dzēšana no groza
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu dzēst produktu no sava groza, lai izņemt no groza produktus, kurus vairāk nevēlos tur redzēt.
<b>Detālas</b>	Lietotājam, lai dzēst produktu no sava grozam ir jānosūta sistēmai produkta dzēšanas no groza pieprasījums ar produkta datiem.
<b>Akceptēšanas kritēriji</b>	<p>Pēc pieprasījuma saņemšanas sistēma pārbauda:</p> <ul style="list-style-type: none"> <li>• vai lietotājs ir autentificējies</li> <li>• vai dzēšamā lietotāja un produkta kombinācijā eksistē datu bāzē.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad sistēma atzīmē ierakstu datu bāzē kā dzēstu un lietotājam tiek parādīts viņa grozs.</p>

### 3.4.3 Cenu pārvaldības lietotāju stāsti

#### 3.4.3.1 Jaunas cenas pievienošanas lietotāja stāsts

Tabula 3.14

<b>Identifikators</b>	US_PRICE_01
<b>Nosaukums</b>	Jaunas cenas pievienošana.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu pievienot jaunu cenu produktam, lai atjaunotu vai apstiprinātu tā cenu.
<b>Detāļas</b>	Lietotājam, lai pievienotu jaunu cenu ir jānosūta sistēmai jaunas cenas pievienošanas pieprasījums ar: <ul style="list-style-type: none"><li>• veikalu, kurā produktam ir konkrētā cena;</li><li>• cena [decimāldaļskaitlis, obligāts, vērtība no 0,01 līdz 99999,99], kāda produktam ir šajā veikalā.</li></ul>
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas, sistēma pārbauda: <ul style="list-style-type: none"><li>• vai ievadītie dati atbilst sistēmas prasībām</li><li>• vai sistēmā eksistē produkts un veikals ar attiecīgajiem identifikatoriem.</li></ul> Ja datu pārbaude ir veiksmīga, tad lietotājam tiek parādīta viņa izveidotā cena.

#### 3.4.3.2 Cenas rediģēšanas lietotāja stāsts

Tabula 3.15

<b>Identifikators</b>	US_PRICE_02
<b>Nosaukums</b>	Cenas rediģēšana.
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu rediģēt sevis pievienotu cenu, lai izlabotu cenas radīšanas procesā radušās kļūdas.
<b>Detāļas</b>	Lietotājam, lai rediģētu sevis izveidotu cenu, ir jānosūta sistēmai cenas rediģēšanas pieprasījums ar izmainītām cenas detaļām.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas, sistēma pārbauda: <ul style="list-style-type: none"><li>• vai lietotājs, kurš izlabo cenu, šo cenu ir izveidojis;</li><li>• vai ievadītie dati atbilst sistēmas prasībām;</li><li>• vai sistēmā eksistē produkts un veikals ar attiecīgajiem identifikatoriem.</li></ul>

	Ja datu pārbaude ir veiksmīga, tad sistēma atzīmē, ka lietotājs ir izmainījis attiecīgo cenas ierakstu datu bāzē un lietotājam tiek parādīta viņa rediģētā cena.
--	--

### 3.4.4 Saziņas lietotāju stāsti

#### 3.4.4.1 Ziņas nosūtīšanas lietotāja stāsts.

Tabula 3.16

<b>Identifikators</b>	US_MSGE_01
<b>Nosaukums</b>	Ziņas nosūtīšana
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu nosūtīt ziņu administrācijai vai citam lietotājam, lai paziņotu par kļūdām sistēmā, citu lietotāju negodprātīgām darbībām vai sazinātos ar citiem lietotājiem.
<b>Detāļas</b>	Lietotājam, lai nosūtītu ziņu ir jānosūta sistēmai ziņas nosūtīšanas pieprasījums ar: <ul style="list-style-type: none"> <li>• ziņas saņēmēja identifikatoru [obligāts, sistēmā eksistējošs];</li> <li>• ziņas tēmu [obligāta, no 5 līdz 100 simbolu gara];</li> <li>• ziņas tekstu [obligāts, no 5 līdz 2000 simbolu garš].</li> </ul>
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas, sistēma pārbauda: <ul style="list-style-type: none"> <li>• vai ziņas saņēmējs ir sistēmā eksistējošs lietotājs;</li> <li>• vai ievadītie dati atbilst sistēmas prasībām;</li> </ul> Ja datu pārbaude ir veiksmīga, tad lietotājam tiek parādīta viņa nosūtītā ziņa cena.

#### 3.4.4.2 Ziņas apskatīšanas lietotāja stāsts

Tabula 3.17

<b>Identifikators</b>	US_MSGE_02
<b>Nosaukums</b>	Ziņas apskatīšana
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs apskatīt saņemtu ziņu, lai atbildētu uz to vai rīkotos attiecīgi ziņā dotajiem norādījumiem.

<b>Detālas</b>	Lietotājam, lai apskatītu ziņu ir jānosūta pieprasījums sistēmai ar ziņas identifikatoru.
<b>Akceptēšanas kritēriji</b>	Pēc ziņas pieprasījuma saņemšanas sistēma pārbauda vai ziņas saņēmējs ir autentificētais lietotājs, un, ja tā ir, tad atzīmē, ka ziņa ir lasīta un atgriež ziņu lietotājam.

#### 3.4.4.3 Ziņas dzēšanas lietotāja stāsts

*Tabula 3.18*

<b>Identifikators</b>	US_MSGE_03
<b>Nosaukums</b>	Ziņas dzēšana
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu dzēst sevis nosūtītu vai saņemtu ziņu, lai tā neuzrādītos nosūtīto vai saņemto ziņu sarakstā.
<b>Detālas</b>	Lietotājam, lai dzēstu ziņu, ir jānosūta sistēmai ziņas dzēšanas pieprasījums.
<b>Akceptēšanas kritēriji</b>	Pēc ziņas dzēšanas pieprasījuma saņemšanas sistēma pārbauda vai ziņas saņēmējs vai autors ir autentificētais lietotājs, un, ja tā ir, tad atzīmē, ka autors vai saņēmējs ziņu ir izdzēsis.

#### 3.4.4.4 Visu saņemto ziņu apskatīšanas lietotāja stāsts

*Tabula 3.19*

<b>Identifikators</b>	US_MSGE_04
<b>Nosaukums</b>	Visu saņemto ziņu apskatīšana
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu apskatīt visas saņemtās ziņas, lai pārbaudītu vai man nav pienākusi jauna ziņa un apskatīt konkrētas saņemtās ziņas.
<b>Detālas</b>	Lietotājam, lai apskatītu visas saņemtās ziņas, ir jānosūta sistēmai visu saņemto ziņu pieprasījums.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma atgriež lietotājam visas ziņas, kurās kā ziņas saņēmējs ir norādīts autentificētais lietotājs.

### 3.4.4.5 Visu nosūtīto ziņu apskatīšanas lietotāja stāsts

Tabula 3.20

<b>Identifikators</b>	US_MSGE_05
<b>Nosaukums</b>	Visu nosūtīto ziņu apskatīšana
<b>Stāsts</b>	Es kā autentificēts sistēmas lietotājs varu apskatīt visas sevis nosūtītās ziņas, lai pārbaudītu, ko es esmu nosūtījis konkrētam lietotājam vai administrācijai.
<b>Detāļas</b>	Lietotājam, lai apskatīt visas saņemtās ziņas, ir jānosūta sistēmai visu nosūtīto ziņu pieprasījums.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma atgriež lietotājam visas ziņas, kurās kā ziņas autors ir norādīts autentificētais lietotājs.

## 3.4.5 Administrācijas lietotāju stāsti

### 3.4.5.1 Lietotāja klases izmaiņš lietotāja stāsts

Tabula 3.21

<b>Identifikators</b>	US_ADMN_01
<b>Nosaukums</b>	Lietotāja klases izmaiņš
<b>Stāsts</b>	Es kā autentificēts sistēmas administrators varu izmainīt citu lietotāju klasi, lai paaugstinātu citus lietotājus par administratoriem vai bloķētu tos.
<b>Detāļas</b>	Administratoram, lai nomainītu lietotāja klasi, ir jānosūta sistēmai klases izmaiņš pieprasījums.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma pārbauda vai lietotāju drīkst pārcelt jaunajā klasē, ja tā ir, tad sistēma saglabā datubāzē jaunās lietotāja detaļas un atzīmē, kurš lietotājs tās ir izmainījis, un atgriež lietotājam viņa izmainīto lietotāju.

### 3.4.5.2 Produkta rediģēšanas lietotāja stāsts

Tabula 3.22

<b>Identifikators</b>	US_ADMN_02
<b>Nosaukums</b>	Produkta rediģēšana
<b>Stāsts</b>	Es kā autentificēts sistēmas administrators varu izmainīt citu lietotāju radītos produktus, lai precizētu produkta detaļas vai izmainītu produkta radīšanas laikā radušās kļūdas.
<b>Detaļas</b>	Administratoram, lai rediģētu produktu, ir jānosūta sistēmai izmainītas produkta detaļas, kuras atbilst sistēmas prasībām: <ul style="list-style-type: none"> <li>• ievadīts produkta nosaukums [obligāts, garumā no 3 līdz 30 simbolu garš];</li> <li>• ievadīts sistēmā eksistējoša produkta kategorija;</li> <li>• ievadīts sistēmā eksistējošs produkta cenas tips.</li> </ul>
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma pārbauda: <ul style="list-style-type: none"> <li>• vai autentificētais lietotājs ir administrators;</li> <li>• vai ievadītie dati ir ievadīti dati ir atbilstoši sistēmas prasībām;</li> <li>• vai ievadītā kategorijas, nosaukuma un cenas tipa kombinācija ir unikāla.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad sistēma saglabā datubāzē jaunās produkta detaļas un atzīmē, kurš lietotājs tās ir izmainījis, un lietotājam tiek parādīts viņa rediģētais produkts.</p>

### 3.4.5.3 Produkta dzēšanas lietotāja stāsts

Tabula 3.23

<b>Identifikators</b>	US_ADMN_03
<b>Nosaukums</b>	Produkta dzēšana
<b>Stāsts</b>	Es kā autentificēts sistēmas administrators varu izdzēst citu lietotāju radītos produktus, lai attīrītu sistēmu no neatbilstošiem produktiem.
<b>Detaļas</b>	Administratoram, lai dzēstu produktu, ir jānosūta sistēmai produkta dzēšanas pieprasījums.

<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma pārbauda vai autentificētais lietotājs ir administrators, ja tā ir, tad sistēma norādīto produktu atzīmē kā dzestu un datu bāzē atzīmē, kurš lietotājs ir izmainījis produkta ierakstu.
-------------------------------	---

#### 3.4.5.4 Cenu rediģēšanas lietotāja stāsts

Tabula 3.24

<b>Identifikators</b>	US_ADMN_04
<b>Nosaukums</b>	Cenu rediģēšana
<b>Stāsts</b>	Es kā autentificēts sistēmas administrators varu izmainīt citu lietotāju radītās cenas, lai precizētu cenas detaļas vai izmainītu cenas radīšanas laikā radušās kļūdas.
<b>Detaļas</b>	Administratoram, lai rediģētu produktu, ir jānosūta sistēmai izmainītas cenas detaļas, kuras atbilst sistēmas prasībām: <ul style="list-style-type: none"> <li>• jāizvēlas veikals, kurā produktam ir konkrētā cena;</li> <li>• jāpievieno tam cena [decimāldaļskaitlis, obligāts, vērtība no 0,01 līdz 99999,99].</li> </ul>
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma pārbauda: <ul style="list-style-type: none"> <li>• vai autentificētais lietotājs ir administrators;</li> <li>• vai ievadītie dati ir ievadīti dati ir atbilstoši sistēmas prasībām;</li> </ul> Ja datu pārbaude ir veiksmīga, tad sistēma saglabā datubāzē jaunās cenas detaļas un atzīmē, kurš lietotājs tās ir izmainījis, un lietotājam tiek parādīts viņa rediģētā cena.

#### 3.4.5.5 Cenas dzēšanas lietotāja stāsts

Tabula 3.25

<b>Identifikators</b>	US_ADMN_05
<b>Nosaukums</b>	Cenas dzēšana
<b>Stāsts</b>	Es kā autentificēts sistēmas administrators varu izdzēst citu lietotāju radītās cenas, lai attīrītu sistēmu no neatbilstošajām cenām.

<b>Detālas</b>	Administratoram, lai dzēstu cenu, ir jānosūta sistēmai cenas dzēšanas pieprasījums.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma pārbauda vai autentificētais lietotājs ir administrators, ja tā ir, tad sistēma norādīto cenu atzīmē kā dzēstu un datu bāzē atzīmē, kurš lietotājs ir izmainījis cenas ierakstu.

#### 3.4.5.6 Veikala pievienošanas lietotāja stāsts

Tabula 3.26

<b>Identifikators</b>	US_ADMN_06
<b>Nosaukums</b>	Veikala pievienošana
<b>Stāsts</b>	Es kā autentificēts sistēmas administrators varu izveidot sistēmā jaunu veikalu, lai papildinātu sistēmas funkcionalitāti.
<b>Detālas</b>	Administratoram, lai izveidotu jaunu veikalu, ir jānosūta sistēmai veikala izveidošanas pieprasījums ar jaunā veikala vārdu [obligāts, garumā no 1 līdz 30 simboliem].
<b>Akceptēšanas kritēriji</b>	<p>Pēc pieprasījuma saņemšanas sistēma pārbauda:</p> <ul style="list-style-type: none"> <li>• vai autentificētais lietotājs ir administrators;</li> <li>• vai ievadītie dati atbilst sistēmas prasībām.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad sistēmā tiek izveidots ieraksts ar jauno veikalu un lietotājam tiek parādīts viņa izveidotais veikals.</p>

#### 3.4.5.7 Veikala rediģēšanas lietotāja stāsts

Tabula 3.27

<b>Identifikators</b>	US_ADMN_07
<b>Nosaukums</b>	Veikala rediģēšana
<b>Stāsts</b>	Es kā autentificēts sistēmas administrators varu rediģēt sistēmā eksistējošu veikalu, lai izmainītu veikala radīšanas laikā radušās kļūdas.
<b>Detālas</b>	Administratoram, lai rediģētu sistēmā eksistējošu veikalu ir jānosūta sistēmai veikala rediģēšanas pieprasījums ar rediģētām veikala detaļām.

<b>Akceptēšanas kritēriji</b>	<p>Pēc pieprasījuma saņemšanas sistēma pārbauda:</p> <ul style="list-style-type: none"> <li>• vai autentificētais lietotājs ir administrators;</li> <li>• vai ievadītie dati atbilst sistēmas prasībām.</li> </ul> <p>Ja datu pārbaude ir veiksmīga, tad sistēma saglabā datubāzē jaunās veikala detaļas un atzīmē, kurš lietotājs tās ir izmainījis, un lietotājam tiek parādīts viņa rediģētais veikals.</p>
-------------------------------	--

#### 3.4.5.8 Veikala dzēšanas lietotāja stāsts

Tabula 3.28

<b>Identifikators</b>	US_ADMN_08
<b>Nosaukums</b>	Veikala dzēšana
<b>Stāsts</b>	Es kā autentificēts sistēmas administrators varu izdzēst citu lietotāju radītos veikalus, lai attīrītu sistēmu no neatbilstošiem veikaliem.
<b>Detaļas</b>	Administratoram, lai dzēstu veikalus, ir jānosūta sistēmai veikala dzēšanas pieprasījums.
<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma pārbauda vai autentificētais lietotājs ir administrators, ja tā ir, tad sistēma norādīto veikalus atzīmē kā dzēstus un datu bāzē atzīmē, kurš lietotājs ir izmainījis veikala ierakstu.

#### 3.4.5.9 Visu administratoriem nosūtīto ziņu apskatīšanas lietotāja stāsts

Tabula 3.29

<b>Identifikators</b>	US_ADMN_09
<b>Nosaukums</b>	Visu administratoriem nosūtīto ziņu apskatīšana
<b>Stāsts</b>	Es kā autentificēts sistēmas administrators varu apskatīt visas administratoriem adresētās ziņas, lai pārbaudītu, ko lietotāji vēlas paziņot administratoriem.
<b>Detaļas</b>	Administratoram, lai apskatītu visas saņemtās ziņas, ir jānosūta sistēmai visu saņemto ziņu pieprasījums.

<b>Akceptēšanas kritēriji</b>	Pēc pieprasījuma saņemšanas sistēma pārbauda vai autentificētais lietotājs ir administrators un atgriež lietotājam visas ziņas, kurās kā ziņas saņēmējs ir norādīts administrators lietotājs.
-------------------------------	---

### 3.5 Lietotāju grupas

Sistēmā ir 3 lietotāju grupas:

- Reģistrēts lietotājs – lietotājs, kuram ir atļauja piekļūt sistēmai, apskatīt un pievienot preces un cenas un izveidot preču grozus, kā arī sazināties ar administrāciju, lai ziņotu par kļūdām vai paplašinātus resursus, kuriem lietotājam nav pieejas tiesību.
- Administrators – lietotājs, kuram ir piekļuve visai sistēmas funkcionalitātei. Lietotājs papildus reģistrēta lietotāja funkcionalitātei var pievienot arī veikalus un produktu kategorijas, kā arī mainīt citu reģistrētu lietotāju statusus.
- Bloķēts lietotājs – lietotājs, kuram negodīgas sistēmas izmantošanas rezultātā, pieeja sistēmai ir liegta.

## **4 PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS**

### **4.1 Ievads**

#### **4.1.1 Nosaukums**

Sistēmas WebApiPrice programmatūras projektējuma apraksts.

#### **4.1.2 Autors**

Oskars Kalniņš, oskars.kalnins@gmail.com

#### **4.1.3 Dokumenta izveidošanas datums**

17.05.2015

#### **4.1.4 Dokumenta nolūks**

Dokumenta nolūks ir aprakstīt, kā programmatūras prasību specifikācijā aprakstītās prasības tiek realizētas programmaprodukta izstrādē. Programmatūras projektējuma apraksts paredzēts sistēmas WebApiPrice izstrādātājiem, kā tehniskās specifikācijas un apraksta palīgmateriāls.

#### **4.1.5 Darbības sfēra**

Programmaprodukts WebApiPrice ir tīmekļa lietojumprogrammas saskarne pakalpojumam, kurš glabā patērētāju produktu cenas veikalos, ļauj lietotājiem tās atjaunot, pievienot jaunus produktus un izveidot vēlmju grozus ar kuru palīdzību lietotājs varētu atrast veikalu, kurā tam ir visizdevīgāk iepirkties.

Izstrādātajam programmaproduktam nav jāsniedz grafiskā lietotāja saskarne gala lietotājam.

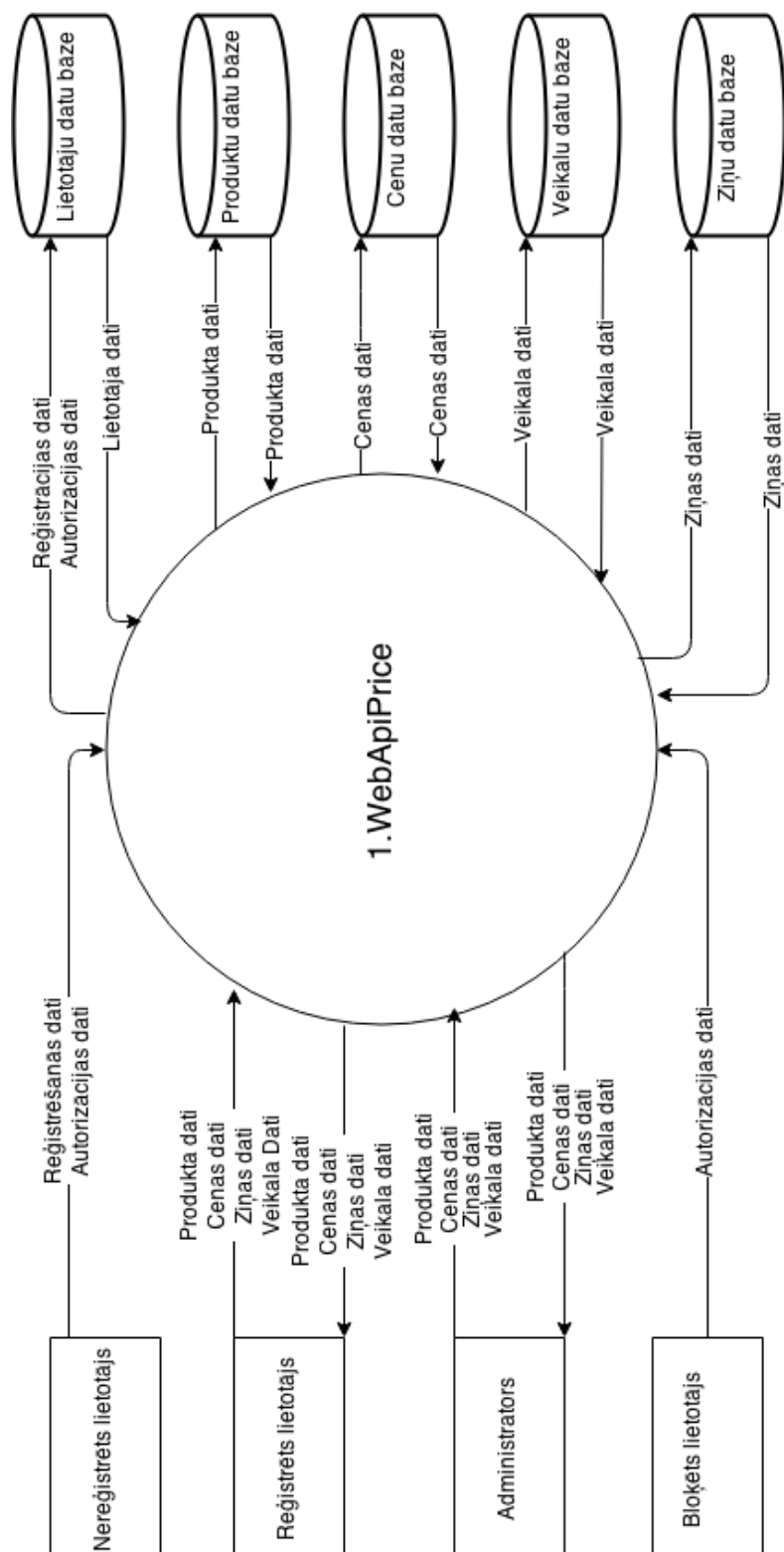
#### **4.1.6 Saistība ar citiem dokumentiem**

Programmatūras projektējuma apraksts ir kvalifikācijas darba “Preču cenu uzraudzības sistēma” sastāvdaļa.

Projektējuma apraksts lietojams kopā ar WebApiPrice programmatūras prasību specifikāciju.

## 4.2 Datu plūsmu diagrammas

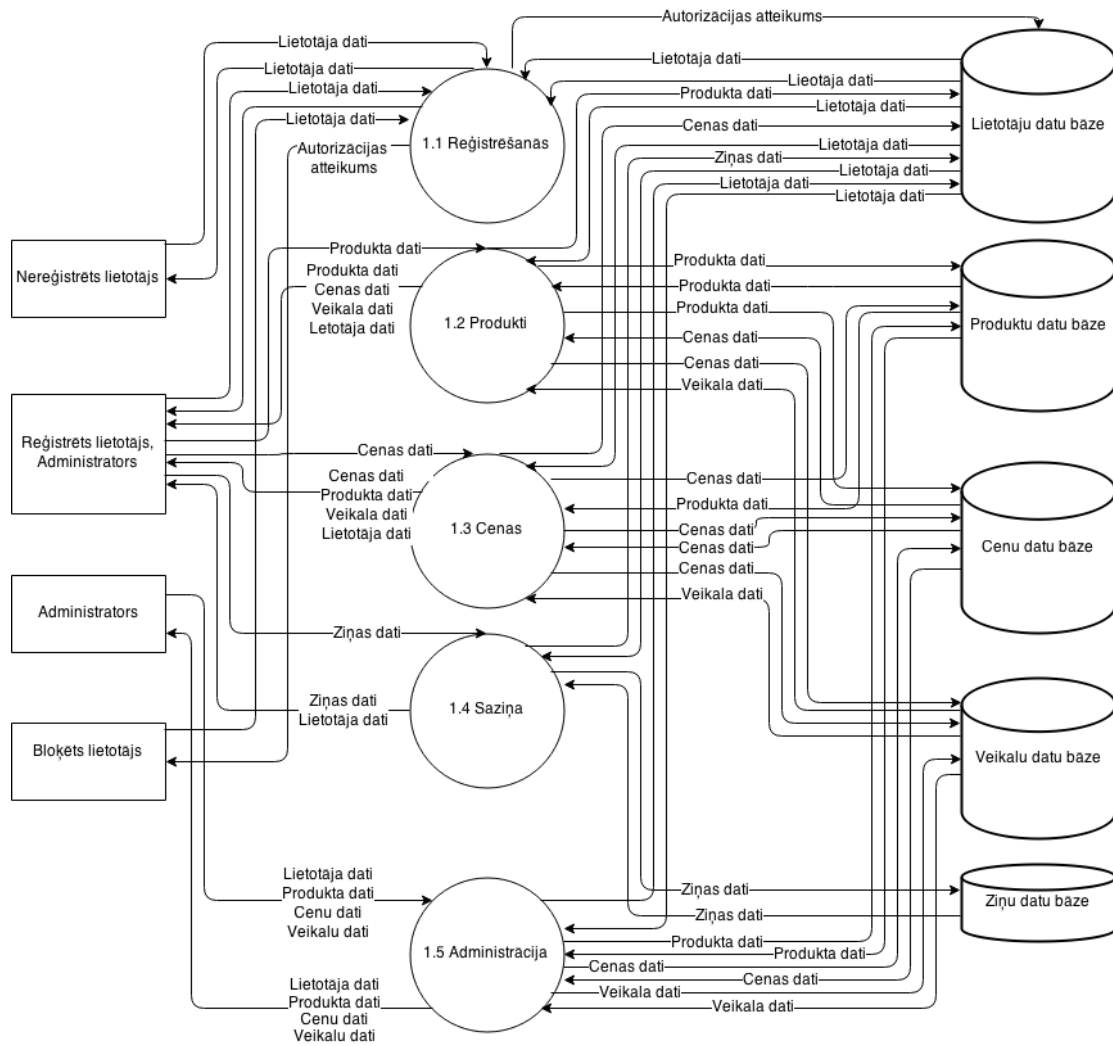
### 4.2.1 Nultā līmeņa datu plūsmas diagramma



Ilustrācija 4.1

Šobrīd nav sistēmu ar ko izveidotā sistēma sadarbojās, bet nākotnē tādas varētu parādīties, piemēram, ārēja veikalu datu bāze vai preču dati.

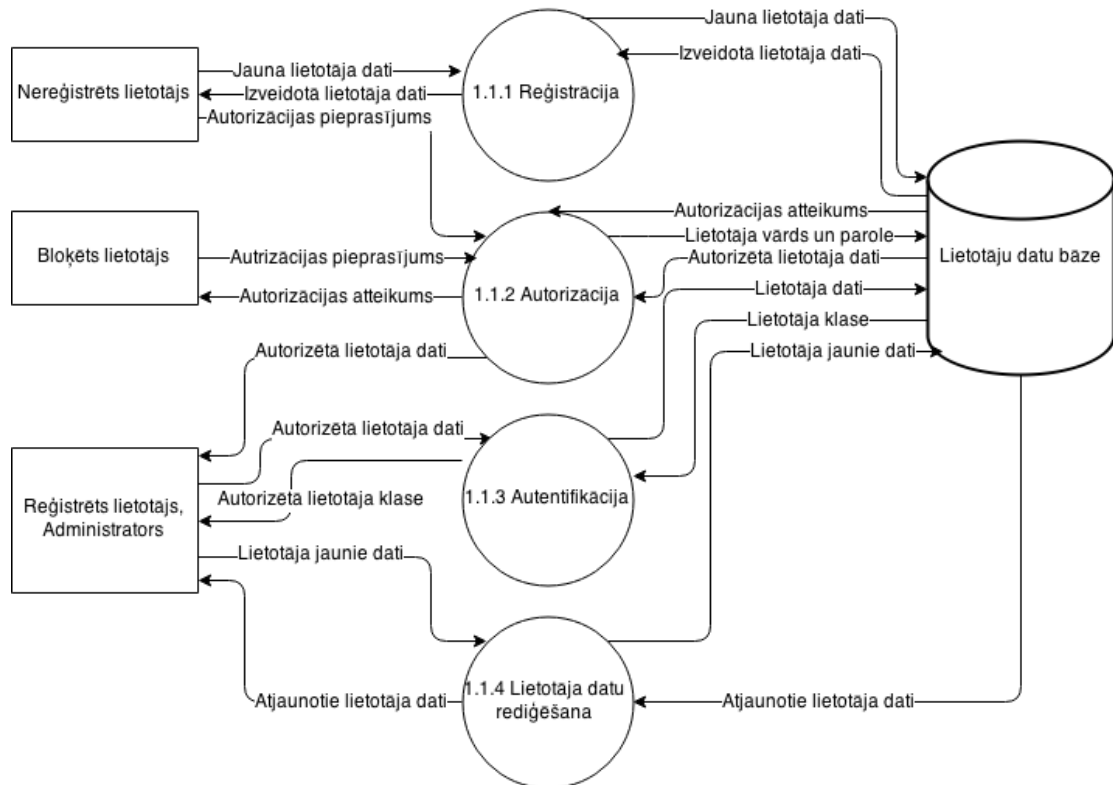
## 4.2.2 Pirmā līmeņa datu plūsmas diagramma



Ilustrācija 4.2

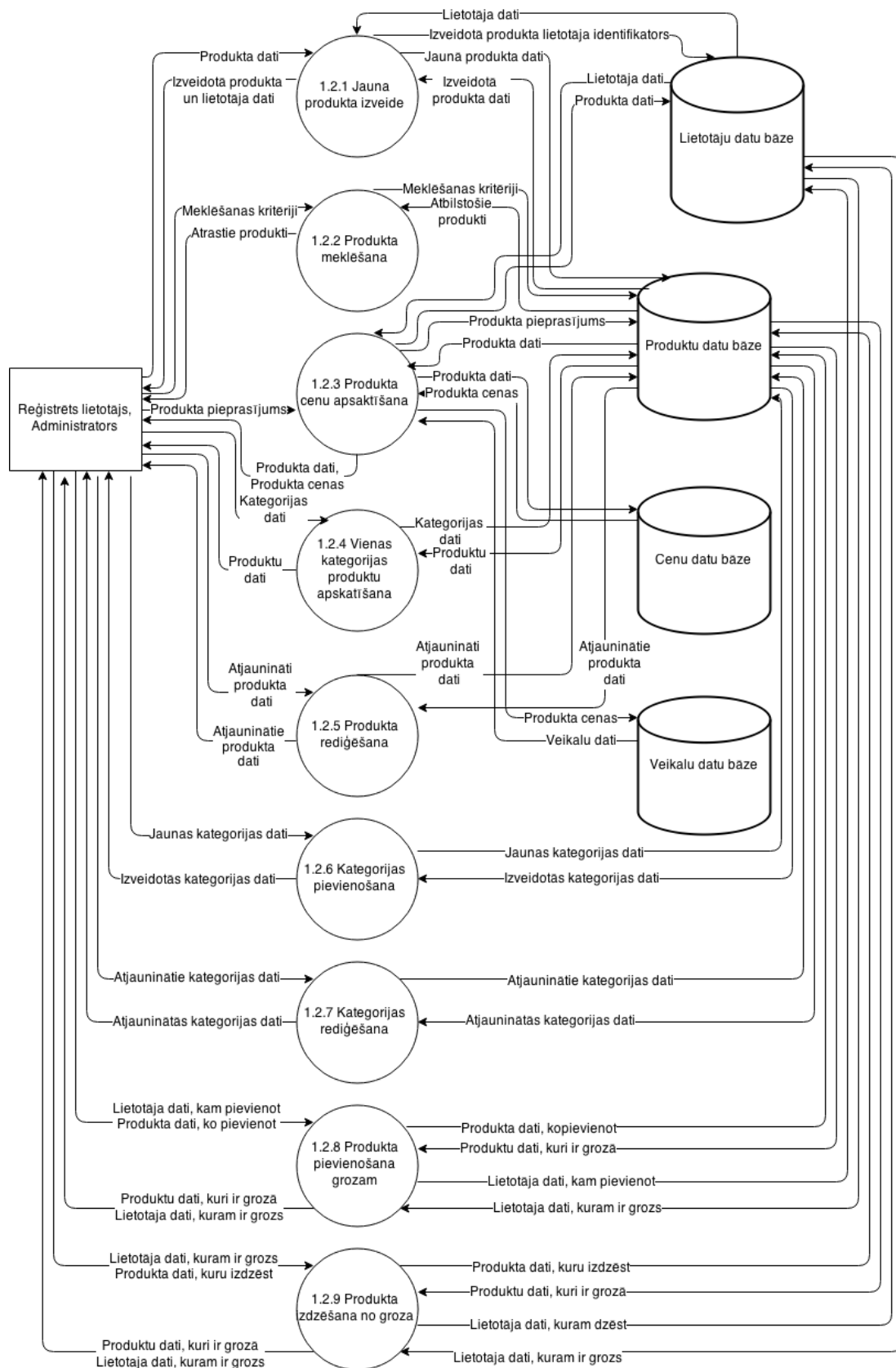
## 4.2.3 Otrā līmeņa datu plūsmas diagrammas

### 4.2.3.1 Moduļa “Reģistrēšanās” datu plūsmas diagramma



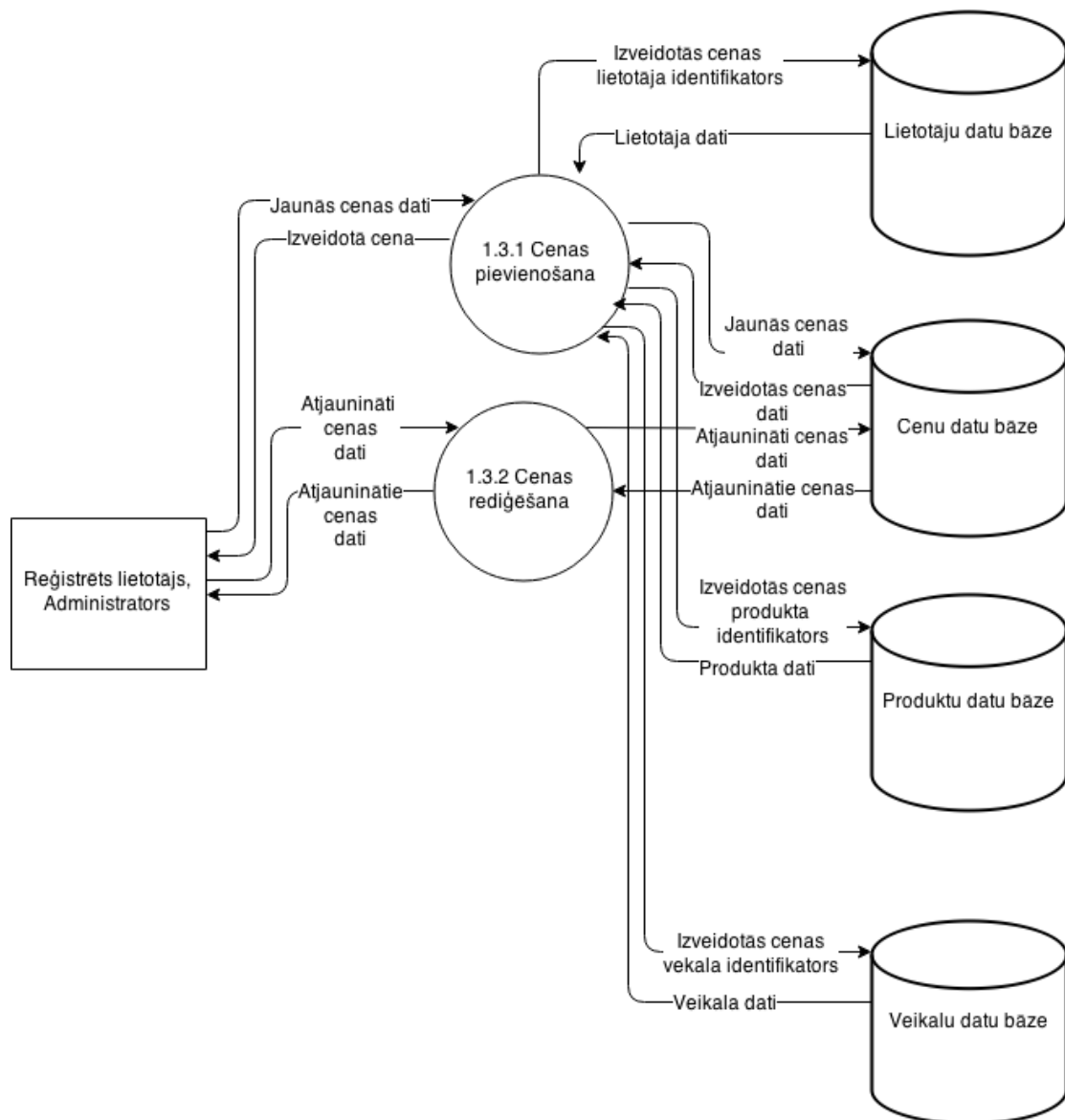
Ilustrācija 4.3

#### 4.2.3.2 Moduļa “Produkti” datu plūsmas diagramma



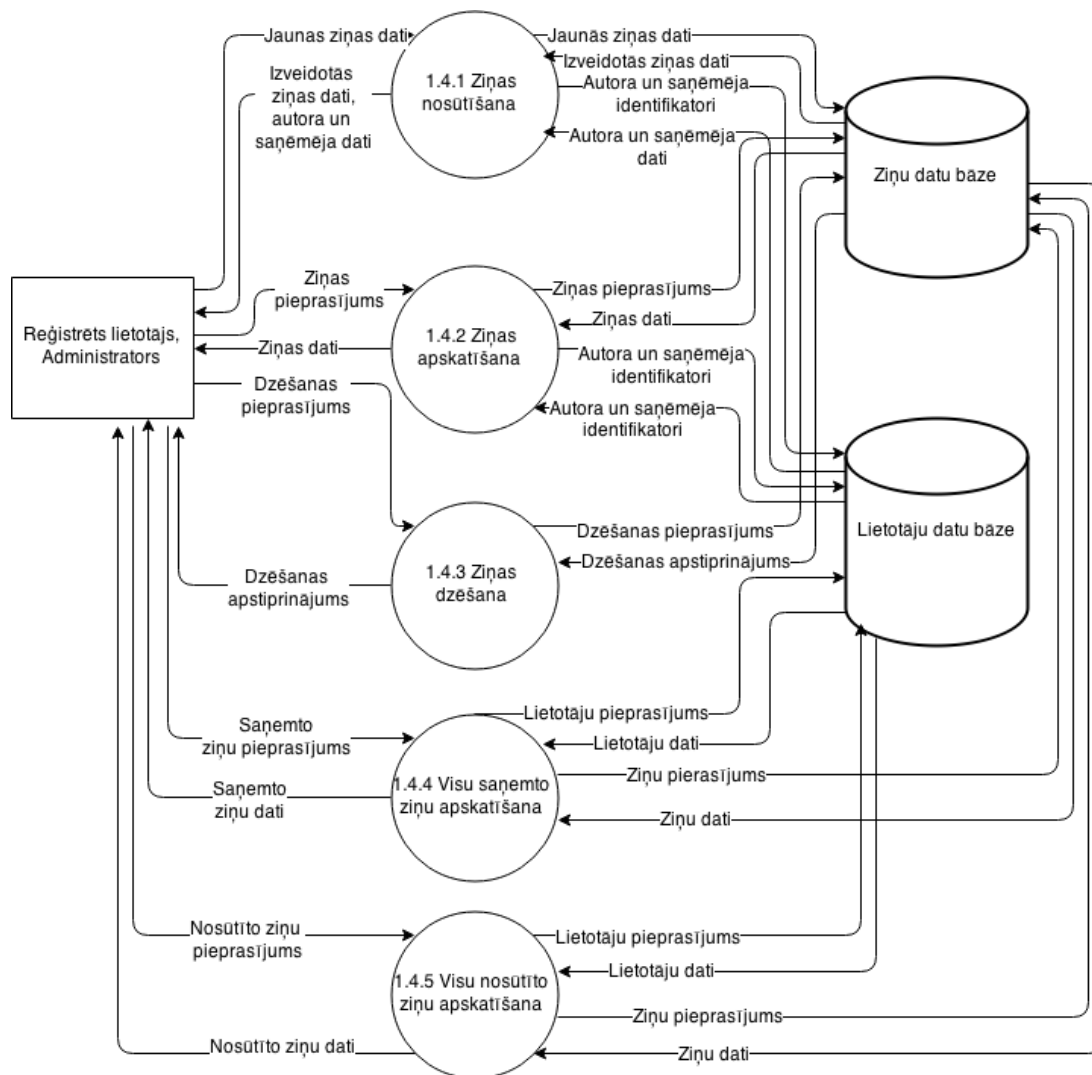
Ilustrācija 4.4

#### 4.2.3.3 Moduļa “Cenas” datu plūsmas diagramma



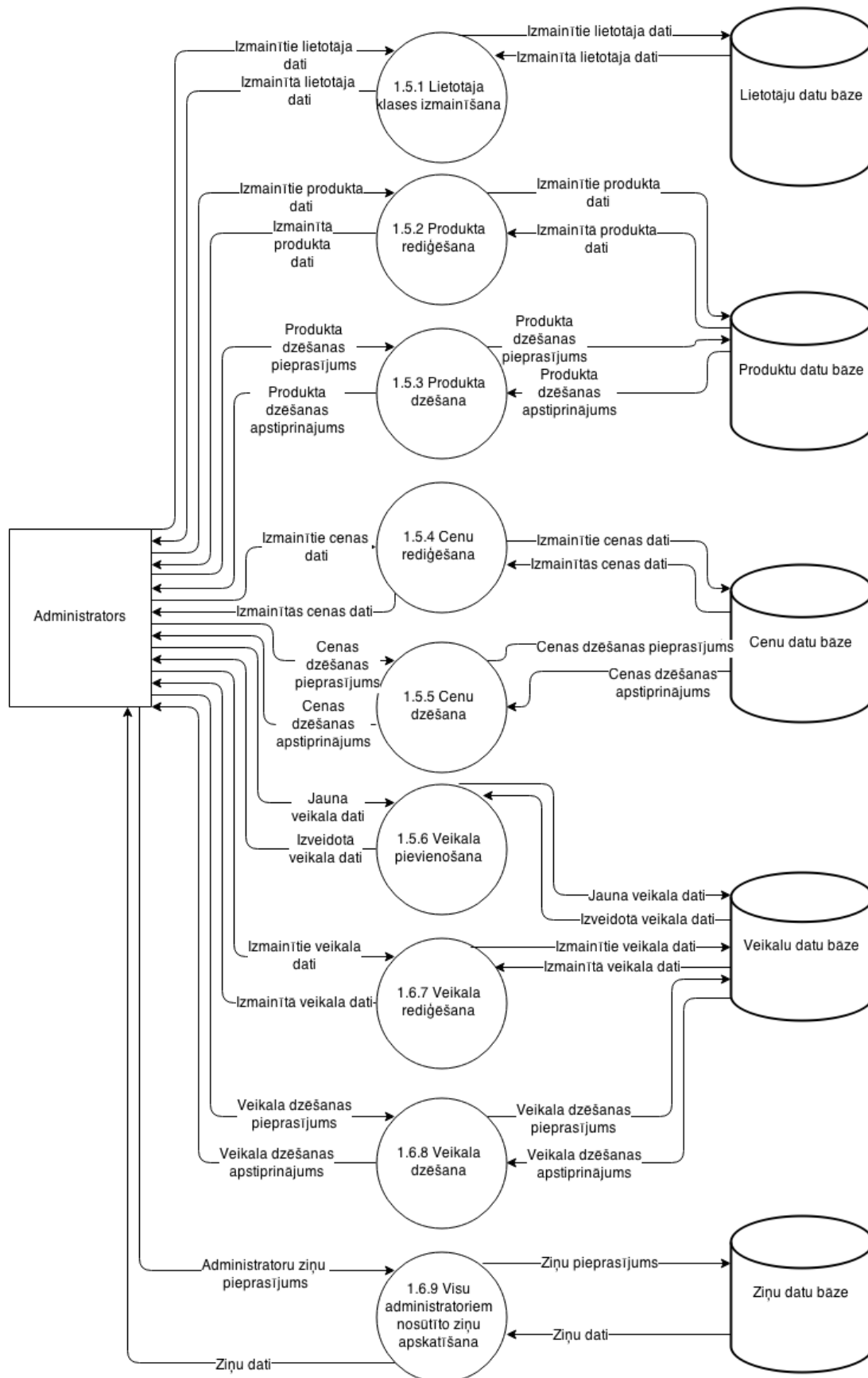
Ilustrācija 4.5

#### 4.2.3.4 Moduļa “Saziņa” datu plūsmas diagramma



Ilustrācija 4.6

#### 4.2.3.5 Moduļa “Administrācija” datu plūsmas diagramma



Ilustrācija 4.7

## 4.3 Moduļu dekompozīcija

Tālāk ir aprakstīta 2. līmeņa datu plūsmu diagrammā redzamo moduļu realizācijas apraksts.

### 4.3.1 Modulis Reģistrēšanās

#### 4.3.1.1 Reģistrācija

Tabula 4.1

<b>Kontrolieris</b>	RegisterController
<b>Metode</b>	RegisterUser
<b>HTTP ceļš</b>	api/v1/register
<b>HTTP metode</b>	POST
<b>Lietotāja stāsts</b>	US_REG_01
<b>Apraksts</b>	Metode saņem JSON objektu “newUser” ar aizpildītiem laukiem: “Username”, “Password”, “Email”. Pirms objekta sistēma pārbauda vai saņemtie dati atbilst sistēmas prasībām, ja šī pārbaude ir neveiksmīga, tiek atgriezts atbilstošs kļūdas paziņojums. Pēc veiksmīgas apstrādes šis objekts tiek padots klases “IAddUserMaintenanceProcessor” metodei “AddUser”, kura kartē servisa modeļa klasi “newUser” uz domēna modeļa klasi “User”. Izveidotais objekts tiek padots “IAddUserQueryProcessor” klases “AddUser” metodei, kura pārbauda datu vai lietotājvārds un e-pasta adreses sistēmā vēl neeksistē. Ja dati ir unikāli objektam tiek aizpildīti lauki, kuriem ir nepieciešamas noklusētās vērtības un objekts tiek saglabāts datu bāzē. Tālāk objekts tiek atgriezts izsaucošajai metodei, kur tas tiek kartēts uz servisa modeļa klasi “User” un atgriezts lietotājam.

#### 4.3.1.2 Autorizācija

Tabula 4.2

<b>Klase</b>	Iebūvēts autorizācijas filtrs
<b>Metode</b>	[Authorize]
<b>HTTP ceļš</b>	-
<b>HTTP metode</b>	-
<b>Lietotāja stāsts</b>	US_REG_02
<b>Apraksts</b>	Autorizācija notiek pirms visām kontrolieru metodēm sistēmā, izņemot pirms RegisterController.RegisterUser metodes, kurai var piekļūt neregistrēts lietotājs. Autorizācijas filtrs pirms kontroliera metodes izsaukšanas pārbauda vai lietotājam ir piešķirta loma, kura norādīta autorizācijas filtrā. Ja lietotājam šāda loma nav piešķirta tad tiek atgriezts kļūdas paziņojums. Ja autorizācija notiek veiksmīgi, tad tiek izpildīta izsauktā kontroliera metode.

#### 4.3.1.3 Autentifikācija

Tabula 4.3

<b>Klase</b>	BasicAuthenticationMessageHandler
<b>Metode</b>	Authenticate
<b>HTTP ceļš</b>	-
<b>HTTP metode</b>	-
<b>Lietotāja stāsts</b>	US_REG_02
<b>Apraksts</b>	Lietotāja autentifikācija notiek saņemot HTTP pieprasījumu ar galvenē ietvertu "Authorization" sadaļu. Sistēmā ir realizēta vienkārša HTTP autorizācija. Sistēma saņem Base64 kodējumā iekodētu lietotājvārda un paroles kombināciju, kura tiek atkodēta un salīdzināta ar datubāzē esošajām lietotājvārda un paroles kombinācijām. Ja lietotājvārda un paroles kombinācija tiek atrasta, tad lietotājs tiek autentificēts sistēmā un tam tiek piešķirtas lomas atkarībā no tā klases. Autentifikāciju realizē "BasicSecurityService" klase.

#### 4.3.1.4 Lietotāja datu rediģēšana

Tabula 4.4

<b>Klase</b>	UsersController
<b>Metode</b>	UpdateUser
<b>HTTP ceļš</b>	api/v1/users/ api/v1/users/{userId}
<b>HTTP metode</b>	PUT/PATCH
<b>Lietotāja stāsts</b>	US_REG_03
<b>Apraksts</b>	<p>Lietotājam ir iespējams izmainīt savu lietotājvārdu, e-pastu un paroli. Pirms metodes sava lietotāja izmaiņšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User", kā arī tiek izsaukta "ValidateUserUpdateRequestAttribute" metode, kura pārbauda vai dati atbilst sistēmas prasībām. Metode saņem JSON objektu "NewUser" ar vismaz vienu no šiem atribūtiem, šis objekts tiek nodots "UpdateUserMaintenanceProcessor" klasei. Šī klase nosaka, kuri lietotāja atribūti ir izmainīti un izveido vārdnīcu ar šiem atribūtiem. Šī vārdnīca tiek nodota "UpdateUserQueryProcessor" klasei, kur tiek pārbaudīts vai atjauninātie dati ir sistēmā unikāli. Ja šādi dati sistēmā atrasti netiek, tad tiek aizpildīti lauki, kuriem ir nepieciešamas noklusētās vērtības, atjauninātais ieraksts tiek saglabāts, kartēts uz servisa modeļa klasi "User" un atgriezts lietotājam. Ja metode tiek izsaukta ar lietotāja identifikatoru, tad tiek izsaukts validācijas filtrs, kurš pārbauda vai lietotājam ir loma "Administrator". Tālākā metodes darbība ir līdzīga aprakstītajai ar nelielām atšķirībām.</p>

## 4.3.2 Modulis Produkti

### 4.3.2.1 Jauna produkta izveidošana

Tabula 4.5

<b>Kontrolieris</b>	ProductsController
<b>Metode</b>	AddProduct
<b>HTTP ceļš</b>	api/v1/products
<b>HTTP metode</b>	POST
<b>Lietotāja stāsts</b>	US_PROD_01
<b>Apraksts</b>	<p>Pirms metodes produkta pievienošanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma “User”, kā arī tiek izsaukta “NewProduct” servisa modeļa validācija, kura pārbauda vai saņemtajā JSON objektā ir visi sistēmai nepieciešamie dati un vai šie dati atbilst sistēmas prasībām, lai izveidotu jaunu produkta ierakstu sistēmā. Saņemtais objekts tiek kartēts uz “Product” domēna modeļa klases objektu, pēc kura izveides tiek pārbaudīts vai norādītā produkta kategorija un cenas tips eksistē sistēmā un vai produkta nosaukuma, kategorijas un cenas tipa kombinācija ir unikāla. Ja pārbaudes ir veiksmīgas tad objektam tiek aizpildīti lauki kuriem ir nepieciešamas noklusētās vērtības, un tas tiek saglabāts “Product” tabulā, kartēts uz servisa modeļa klasi “Product” un atgriezts lietotājam.</p>

#### 4.3.2.2 Produkta meklēšana

Tabula 4.6

<b>Kontrolieris</b>	ProductsController
<b>Metode</b>	GetProducts
<b>HTTP ceļš</b>	api/v1/products
<b>HTTP metode</b>	GET
<b>Lietotāja stāsts</b>	US_PROD_02
<b>Apraksts</b>	Pirms metodes produkta meklēšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User", kā arī tiek izsaukta "SearchString" servisa modeļa validācija, kura pārbauda vai saņemtajā JSON objektā ir aizpildīts "String" atribūts. Metode meklē visus sistēmā esošos produktus, kuru nosaukumā ir padotā simbolu virkne neatkarīgi no mazajiem un lielajiem burtiem. Atrastie produkti tiek sadalīti lapās pēc lietotāja norādītajiem parametriem, vai, ja lietotājs parametrus nav norādījis, izmantojot sistēmas noklusētās vērtības un atgriezti lietotājam.

#### 4.3.2.3 Produkta cenu apskatīšana

Tabula 4.7

<b>Kontrolieris</b>	ProductsController
<b>Metode</b>	GetPrices
<b>HTTP ceļš</b>	api/v1/products/{productId}/prices
<b>HTTP metode</b>	GET
<b>Lietotāja stāsts</b>	US_PROD_03
<b>Apraksts</b>	Pirms metodes sava cenu apskatīšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User". Metode meklē visas sistēmā esošās cenas, kuras ir pievienotas saitē norādītajam produktam. Atrastās cenas tiek sadalītas lapās pēc lietotāja norādītajiem parametriem, vai, ja lietotājs parametrus nav norādījis, izmantojot sistēmas noklusētās vērtības un atgrieztas lietotājam.

#### 4.3.2.4 Vienas kategorijas produktu apskatīšana

Tabula 4.8

<b>Kontrolieris</b>	CategoriesController
<b>Metode</b>	GetProducts
<b>HTTP ceļš</b>	api/v1/categories/{categoryId}
<b>HTTP metode</b>	GET
<b>Lietotāja stāsts</b>	US_PROD_04
<b>Apraksts</b>	Pirms metodes vienas kategorijas produktu apskatīšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User". Metode meklē visas sistēmā esošos produktus, kuri ir pievienoti saitē norādītajai kategorijai. Atrastie produkti tiek sadalīti lapās pēc lietotāja norādītajiem parametriem, vai, ja lietotājs parametrus nav norādījis, izmantojot sistēmas noklusētās vērtības un atgriezti lietotājam.

#### 4.3.2.5 Produkta rediģēšana

Tabula 4.9

<b>Klase</b>	ProductsController
<b>Metode</b>	UpdateProduct
<b>HTTP ceļš</b>	api/v1/products/{productId}
<b>HTTP metode</b>	PUT/PATCH
<b>Lietotāja stāsts</b>	US_PROD_05
<b>Apraksts</b>	<p>Lietotājam ar lomu "User" ir iespējams izmainīt sevis izveidota produkta nosaukumu, bet lietotāji ar lomu "Administrator" var izmainīt jebkura produkta nosaukumu. Pirms kontroliera metodes produkta izmaiņšanai izsaukšanas nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User", kā arī tiek izsaukta "ValidateProductUpdateRequest" metode, kura pārbauda vai dati atbilst sistēmas prasībām. Metode saņem JSON objektu ar aizpildītu atribūtu "Name" un padod to "UpdateUserMaintenanceProcessor" klasei. Šī klase izveido vārdnīcu nodod to "UpdateProductQueryProcessor" klasei, kur tiek pārbaudīts vai "Name" ir vienīgais atribūts, kurš ir padots klasei, vai atjauninātā produkta nosaukuma, kategorijas un cenas tipa kombinācija ir unikāla. Ja jaunā kombinācija sistēmā atrasta netiek, tad tiek aizpildīti lauki, kuriem ir nepieciešamas noklusētās vērtības, atjauninātais ieraksts tiek saglabāts, kartēts uz servisa modeļa klasi "Product" un atgriezts lietotājam</p>

#### 4.3.2.6 Kategorijas pievienošana

Tabula 4.10

<b>Kontrolieris</b>	CategoriesController
<b>Metode</b>	AddCategory
<b>HTTP ceļš</b>	api/v1/categories
<b>HTTP metode</b>	POST
<b>Lietotāja stāsts</b>	US_PROD_06
<b>Apraksts</b>	<p>Pirms metodes kategorijas pievienošanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User", kā arī tiek izsaukta "NewCategory" servisa modeļa validācija, kura pārbauda vai saņemtajā JSON objektā ir aizpildītas atribūts "Name". Ja validācija ir veiksmīga saņemtais objekts tiek kartēts uz "Category" domēna modeļa klases objektu, pēc kura izveides tiek pārbaudīts vai kategorijas nosaukums ir unikāls. Ja pārbaude ir veiksmīga tad objektam tiek aizpildīti lauki kuriem ir nepieciešamas noklusētās vērtības, un tas tiek saglabāts "Category" tabulā, kartēts uz servisa modeļa klasi "Category" un atgriezts lietotājam.</p>

#### 4.3.2.7 Kategorijas rediģēšana

Tabula 4.11

<b>Klase</b>	CategoriesController
<b>Metode</b>	UpdateCategory
<b>HTTP ceļš</b>	api/v1/categories/{categoryId}
<b>HTTP metode</b>	PUT/PATCH
<b>Lietotāja stāsts</b>	US_PROD_07
<b>Apraksts</b>	<p>Lietotājam ar lomu "User" ir iespējams izmainīt sevis izveidotas kategorijas nosaukumu, bet lietotāji ar lomu "Administrator" var izmainīt jebkuras kategorijas nosaukumu. Pirms kontroliera metodes kategorijas izmainīšanai izsaušanas nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User", kā arī tiek izsaukta "ValidateCategoryUpdateRequest" metode, kura pārbauda vai dati atbilst sistēmas prasībām. Metode saņem JSON objektu ar aizpildītu atribūtu "Name" un padod to "UpdateCategoryMaintenanceProcessor" klasei. Šī klase izveido vārdnīcu un nodod to "UpdateCategoryQueryProcessor" klasei, kur tiek pārbaudīts vai "Name" ir vienīgais ieraksts vārdnīcā, vai atjauninātais kategorijas nosaukums ir unikāls. Ja jaunais nosaukums sistēmā atrasts netiek, tad tiek aizpildīti lauki, kuriem ir nepieciešamas noklusētās vērtības, atjauninātais ieraksts tiek saglabāts, kartēts uz servisa modeļa klasi "Category" un atgriezts lietotājam.</p>

#### 4.3.2.8 Produkta pievienošana grozam

Tabula 4.12

<b>Klase</b>	UserProductsController
<b>Metode</b>	AddUserProducts AddUserProduct
<b>HTTP ceļš</b>	api/v1/users/products api/v1/users/products/{productId}
<b>HTTP metode</b>	PUT
<b>Lietotāja stāsts</b>	US_PROD_08
<b>Apraksts</b>	<p>Pirms metodes produkta pievienošanai lietotāja grozam izsaukšanas nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User". Ceļš bez produkta identifikatora ļauj lietotājam aizstāt visu groza saturu ar jauniem produktiem. "AddUserProducts" metode saņem kolekciju ar produkta identifikatoriem, pārbauda vai sistēmā eksistē norādītie produkti, un, ja tie visi eksistē, pievieno lietotāja grozam tos produktus, kuri tajā vēl neeksistē. Ceļš ar produkta identifikatoru ļauj lietotājam pievienot savam grozam ceļā norādīto produktu. "AddUserProduct" metode saņem produkta identifikatoru, pārbauda vai šis produkts eksistē datu bāzē un pievieno to lietotāja grozam, ja šis produkts tajā vēl neeksistē.</p>

#### 4.3.2.9 Produkta izdzēšana no groza

Tabula 4.13

<b>Klase</b>	UserProductsController
<b>Metode</b>	DeleteUserProducts DeleteUserProduct
<b>HTTP ceļš</b>	api/v1/users/products api/v1/users/products/{productId}
<b>HTTP metode</b>	DELETE
<b>Lietotāja stāsts</b>	US_PROD_09
<b>Apraksts</b>	Pirms metodes produkta pievienošanai lietotāja grozam izsaukšanas nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User". Ceļš bez produkta identifikatora izsauc metodi "DeleteUserProducts", kura ļauj lietotājam izdzēst visu groza saturu. Ceļš ar produkta identifikatoru ļauj lietotājam dzēst no sava groza ceļā norādīto produktu. "DeleteUserProduct" metode saņem produkta identifikatoru, pārbauda vai šis eksistē produkts lietotāja grozā un, ja tas ir lietotāja grozā, dzēš to no groza. Abas metodes atgriež lietotāju, kurš ir izsaucis šīs metodes.

### 4.3.3 Modulis Cenas

#### 4.3.3.1 Cenas pievienošana

Tabula 4.14

<b>Kontrolieris</b>	PricesController
<b>Metode</b>	AddPrice
<b>HTTP ceļš</b>	api/v1/prices
<b>HTTP metode</b>	POST
<b>Lietotāja stāsts</b>	US_PRICE_01
<b>Apraksts</b>	<p>Pirms metodes cenas pievienošanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User", kā arī tiek izsaukta "NewPrice" servisa modeļa validācija, kura pārbauda vai saņemtajā JSON objektā ir visi sistēmai nepieciešamie dati un vai šie dati atbilst sistēmas prasībām, lai izveidotu jaunu cenas ierakstu sistēmā. Saņemtais objekts tiek kartēts uz "Price" domēna modeļa klases objektu, pēc kura izveides tiek pārbaudīts vai norādītais produkts un veikals eksistē. Ja pārbaudes ir veiksmīgas tad objektam tiek aizpildīti lauki kuriem ir nepieciešamas noklusētās vērtības, un tas tiek saglabāts "Price" tabulā, kartēts uz servisa modeļa klasi "Price" un atgriezts lietotājam.</p>

#### 4.3.3.2 Cenas rediģēšana

Tabula 4.15

<b>Klase</b>	PricesController
<b>Metode</b>	UpdatePrice
<b>HTTP ceļš</b>	api/v1/prices/{priceId}
<b>HTTP metode</b>	PUT/PATCH
<b>Lietotāja stāsts</b>	US_PRICE_02
<b>Apraksts</b>	<p>Lietotājam ar lomu "User" ir iespējams izmainīt sevis izveidotas cenas vērtību, bet lietotāji ar lomu "Administrator" var izmainīt jebkuras cenas vērtību. Pirms kontroliera metodes kategorijas izmainīšanai izsaukšanas nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User", kā arī tiek izsaukta "ValidatePriceUpdateRequestAttribute" metode, kura pārbauda vai dati atbilst sistēmas prasībām. Metode saņem JSON objektu ar aizpildītu atribūtu "Value" un padod to "UpdatePriceMaintenanceProcessor" klasei. Šī klase izveido vārdnīcu un nodod to "UpdatePriceQueryProcessor" klasei, kur tiek pārbaudīts vai "Value" ir vienīgais ieraksts vārdnīcā. Ja tā ir, tad tiek aizpildīti lauki, kuriem ir nepieciešamas noklusētās vērtības, atjauninātais ieraksts tiek saglabāts, kartēts uz servisa modeļa klasi "Price" un atgriezts lietotājam</p>

### 4.3.4 Modulis Saziņa

#### 4.3.4.1 Ziņas nosūtīšana

Tabula 4.16

<b>Kontrolieris</b>	MessagesController
<b>Metode</b>	AddMessage
<b>HTTP ceļš</b>	api/v1/messages
<b>HTTP metode</b>	POST
<b>Lietotāja stāsts</b>	US_MSGE_01
<b>Apraksts</b>	<p>Pirms metodes ziņas pievienošanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma “User”, kā arī tiek izsaukta “NewMessage” servisa modeļa validācija, kura pārbauda vai saņemtajā JSON objektā ir visi sistēmai nepieciešamie dati un vai šie dati atbilst sistēmas prasībām, lai izveidotu jaunu ziņas ierakstu sistēmā. Saņemtais objekts tiek kartēts uz “Message” domēna klasi, pēc tam tiek pārbaudīts vai datu bāzē eksistē objektā norādītais ziņas saņēmējs un, ja tas eksistē, ziņa tiek saglabāta datubāzē, kartēta uz servisa modeļa klasi “Message” un atgriezta lietotājam.</p>

#### 4.3.4.2 Ziņas apskatīšana

Tabula 4.17

<b>Kontrolieris</b>	MessagesController
<b>Metode</b>	GetMessage
<b>HTTP ceļš</b>	api/v1/messages/{messageId}
<b>HTTP metode</b>	GET
<b>Lietotāja stāsts</b>	US_MSGE_02
<b>Apraksts</b>	<p>Pirms metodes ziņas apskatīšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma “User”. “GetMessage” metode pārbauda vai datu bāzē eksistē ziņa ar padoto identifikatoru, ja tā ir tad, tiek pārbaudīts vai autentificētais lietotājs ir ziņas autors vai saņēmējs, ja tā nav tad atgriež kļūdas paziņojumu. Ja autentificētais lietotājs ir ziņas autors, tad lietotājam tiek atgrieztas ziņas detaļas, bet, ja autentificētais lietotājs ir ziņas saņēmējs, tad ziņa tiek atzīmēta kā lasīta un tad atgriezta lietotājam.</p>

#### 4.3.4.3 Ziņas dzēšana

Tabula 4.18

<b>Kontrolieris</b>	MessagesController
<b>Metode</b>	DeleteMessage
<b>HTTP ceļš</b>	api/v1/messages/{messageId}
<b>HTTP metode</b>	DELETE
<b>Lietotāja stāsts</b>	US_MSGE_03
<b>Apraksts</b>	Pirms metodes ziņas dzēšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma “User”. “DeleteMessage” metode pārbauda vai datu bāzē eksistē ziņa ar padoto identifikatoru, ja tā ir tad, tiek pārbaudīts vai autentificētais lietotājs ir ziņas autors vai saņēmējs, ja tā nav tad atgriež kļūdas paziņojumu. Ja autentificētais lietotājs ir ziņas autors, tad tiek atzīmēts, ka autors ziņu ir dzēsis un lietotājam tiek atgrieztas visas nosūtītās ziņas, bet, ja autentificētais lietotājs ir ziņas saņēmējs, tad tiek atzīmēts, ka ziņas saņēmējs ziņu ir dzēsis un lietotājam tiek atgrieztas visas saņemtās ziņas.

#### 4.3.4.4 Visu saņemto ziņu apskatīšana

Tabula 4.19

<b>Kontrolieris</b>	MessagesController
<b>Metode</b>	GetReceivedMessages
<b>HTTP ceļš</b>	api/v1/messages/received
<b>HTTP metode</b>	GET
<b>Lietotāja stāsts</b>	US_MSGE_04
<b>Apraksts</b>	Pirms metodes saņemto ziņu apskatīšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma “User”. “GetReceivedMessages” metode meklē visas sistēmā esošās ziņas, kurām kā saņēmējs ir norādīts autentificētais lietotājs. Atrastās ziņās tiek sadalītas lapās pēc lietotāja norādītajiem parametriem, vai, ja lietotājs parametrus nav norādījis, izmantojot sistēmas noklusētās vērtības un atgrieztas lietotājam.

#### 4.3.4.5 Visu nosūtīto ziņu apskatīšana

Tabula 4.20

<b>Kontrolieris</b>	MessagesController
<b>Metode</b>	GetSentMessages
<b>HTTP ceļš</b>	api/v1/messages/sent
<b>HTTP metode</b>	GET
<b>Lietotāja stāsts</b>	US_MSGE_04
<b>Apraksts</b>	Pirms metodes nosūtīto ziņu apskatīšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "User". "GetSentMessages" metode meklē visas sistēmā esošās ziņas, kurām kā saņēmējs ir norādīts autentificētais lietotājs. Atrastās ziņas tiek sadalītas lapās pēc lietotāja norādītajiem parametriem, vai, ja lietotājs parametrus nav norādījis, izmantojot sistēmas noklusētās vērtības un atgrieztas lietotājam.

## 4.3.5 Modulis Administrācija

### 4.3.5.1 Lietotāja klases izmaiņšana

Tabula 4.21

<b>Kontrolieris</b>	UsersController
<b>Metode</b>	ActivateUser BlockUser PromoteUser
<b>HTTP ceļš</b>	api/v1/{userId}/activate api/v1/{userId}/block api/v1/{userId}/promote
<b>HTTP metode</b>	POST
<b>Lietotāja stāsts</b>	US_ADMN_01
<b>Apraksts</b>	Pirms metodēm lietotāja klašu izmaiņšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "Administrators". Visas metodes pārbauda vai datu bāzē eksistē norādītais lietotājs, un, ja tas neeksistē, atgriež kļūdas paziņojumu. "ActivateUser" ļauj administratoram nomainīt lietotāja klasi uz "User", ja lietotājam jau ir šī klase, tad tiek atgriezts kļūdas paziņojums. "BlockUser" ļauj administratoram nomainīt lietotāja klasi uz "BlockedUser", ja lietotājam jau ir šī klase, tad tiek atgriezts kļūdas paziņojums. "PromoteUser" ļauj nomainīt lietotāja ar klasi "User" klasi uz "Administrator", ja lietotāja klase nav "User", tad tiek atgriezts kļūdas paziņojums. Visas metodes atgriež lietotāju, kura klase ir izmainīta.

### 4.3.5.2 Produkta rediģēšana

Tabula 4.22

<b>Klase</b>	ProductsController
<b>Metode</b>	UpdateProduct
<b>HTTP ceļš</b>	api/v1/products/{productId}
<b>HTTP metode</b>	PUT/PATCH
<b>Lietotāja stāsts</b>	US_ADMN_02
<b>Apraksts</b>	Skatīt sadaļu "4.3.2.5 Produkta rediģēšana"

#### 4.3.5.3 Produkta dzēšana

Tabula 4.23

<b>Kontrolieris</b>	ProductsController
<b>Metode</b>	DeleteProduct
<b>HTTP ceļš</b>	api/v1/products/{productId}
<b>HTTP metode</b>	DELETE
<b>Lietotāja stāsts</b>	US_ADMN_03
<b>Apraksts</b>	Pirms metodes produkta dzēšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "Administrator". "DeleteProduct" metode pārbauda vai datu bāzē eksistē produkts ar padoto identifikatoru, ja tā tad produkts tiek atzīmēts kā dzēsts un lietotājam tiek atgriezti visi produkti sadalīti lapās pēc lietotāja norādītajiem parametriem, vai, ja lietotājs parametrus nav norādījis, izmantojot sistēmas noklusētās vērtības un atgriezti lietotājam.

#### 4.3.5.4 Cenu rediģēšana

Tabula 4.24

<b>Klase</b>	PricesController
<b>Metode</b>	UpdatePrice
<b>HTTP ceļš</b>	api/v1/prices/{priceId}
<b>HTTP metode</b>	PUT/PATCH
<b>Lietotāja stāsts</b>	US_ADMN_04
<b>Apraksts</b>	Skatīt sadaļu "4.3.3.2 Cenas rediģēšana"

#### 4.3.5.5 Cenas dzēšana

Tabula 4.25

<b>Kontrolieris</b>	PricesController
<b>Metode</b>	DeletePrice
<b>HTTP ceļš</b>	api/v1/prices/{priceId}
<b>HTTP metode</b>	DELETE
<b>Lietotāja stāsts</b>	US_ADMN_05
<b>Apraksts</b>	<p>Pirms metodes cenas dzēšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma “Administrator”.</p> <p>“DeletePrice” metode pārbauda vai datu bāzē eksistē cena ar padoto identifikatoru, ja tā tad cena tiek atzīmēta kā dzēsta un lietotājam tiek atgriezta visas cenas, kuras eksistē produktam, kuram pievienota dzēstā cena, sadalītas lapās pēc lietotāja norādītajiem parametriem, vai, ja lietotājs parametrus nav norādījis, izmantojot sistēmas noklusētās vērtības un atgriezti lietotājam, pretējā gadījumā tiek atgriezts kļūdas paziņojums.</p>

#### 4.3.5.6 Veikala pievienošana

Tabula 4.26

<b>Kontrolieris</b>	ShopsController
<b>Metode</b>	AddShop
<b>HTTP ceļš</b>	api/v1/shops
<b>HTTP metode</b>	POST
<b>Lietotāja stāsts</b>	US_ADMN_06
<b>Apraksts</b>	<p>Pirms metodes veikala pievienošanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "Administrator", kā arī tiek izsaukta "NewShop" servisa modeļa validācija, kura pārbauda vai saņemtajā JSON objektā ir aizpildītas atribūts "Name". Ja validācija ir veiksmīga saņemtais objekts tiek kartēts uz "Shop" domēna modeļa klases objektu, pēc kura izveides tiek pārbaudīts vai veikala nosaukums ir unikāls. Ja pārbaude ir veiksmīga tad objektam tiek aizpildīti lauki kuriem ir nepieciešamas noklusētās vērtības, un tas tiek saglabāts "Shop" tabulā, kartēts uz servisa modeļa klasi "Shop" un atgriezts lietotājam.</p>

#### 4.3.5.7 Veikala rediģēšana

Tabula 4.27

<b>Klase</b>	ShopsController
<b>Metode</b>	UpdateShop
<b>HTTP ceļš</b>	api/v1/shops/{shopId}
<b>HTTP metode</b>	PUT/PATCH
<b>Lietotāja stāsts</b>	US_ ADMN_07
<b>Apraksts</b>	<p>Pirms metodes veikala atjaunošanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "Administrator", kā arī tiek izsaukta "ValidateShopUpdateRequest" metode, kura pārbauda vai dati atbilst sistēmas prasībām. Metode saņem JSON objektu ar aizpildītu atribūtu "Name" un padod to "UpdateShopMaintenanceProcessor" klasei. Šī klase izveido vārdnīcu un nodod to "UpdateShopQueryProcessor" klasei, kur tiek pārbaudīts vai "Name" ir vienīgais ieraksts vārdnīcā un vai atjauninātais veikala nosaukums ir unikāls. Ja jaunais nosaukums sistēmā atrasts netiek, tad tiek aizpildīti lauki, kuriem ir nepieciešamas noklusētās vērtības, atjauninātais ieraksts tiek saglabāts, kartēts uz servisa modeļa klasi "Shop" un atgriezts lietotājam.</p>

#### 4.3.5.8 Veikala dzēšana

Tabula 4.28

<b>Kontrolieris</b>	ShopsController
<b>Metode</b>	DeleteShop
<b>HTTP ceļš</b>	api/v1/shops/{shopId}
<b>HTTP metode</b>	DELETE
<b>Lietotāja stāsts</b>	US_ADMN_08
<b>Apraksts</b>	Pirms metodes veikala dzēšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "Administrator". "DeleteShop" metode pārbauda vai datu bāzē eksistē veikals ar padoto identifikatoru, ja tā tad veikals tiek atzīmēts kā dzēsts un lietotājam tiek atgriezti visi veikali, kuri eksistē datu bāzē sadalīti lapās pēc lietotāja norādītajiem parametriem, vai, ja lietotājs parametrus nav norādījis, izmantojot sistēmas noklusētās vērtības un atgriezti lietotājam, pretējā gadījumā tiek atgriezts kļūdas paziņojums.

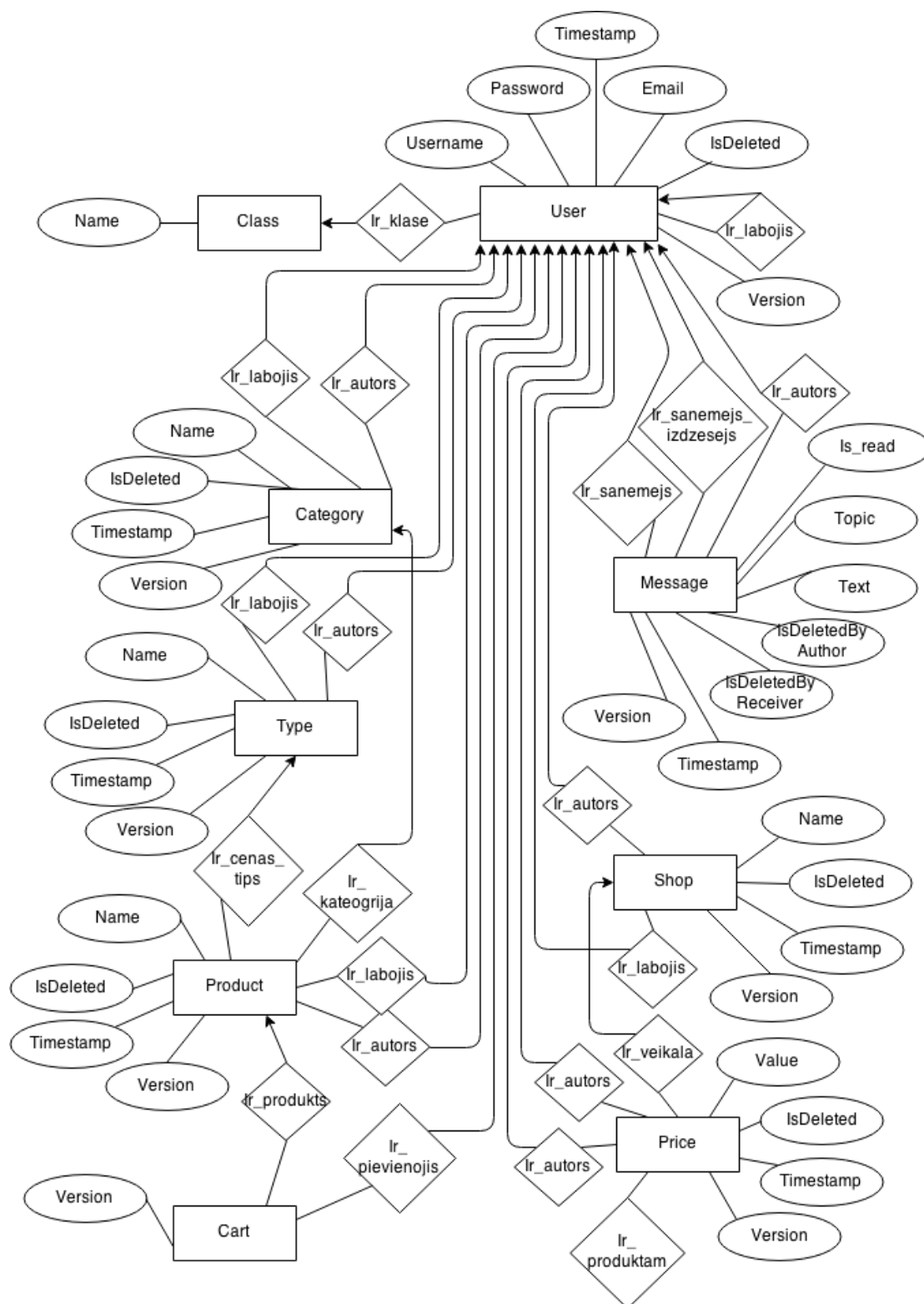
#### 4.3.5.9 Visu administratoriem nosūtīto ziņu apskatīšana.

Tabula 4.29

<b>Kontrolieris</b>	MessagesController
<b>Metode</b>	GetReceivedMessagesAdmin
<b>HTTP ceļš</b>	api/v1/messages/received/admin
<b>HTTP metode</b>	GET
<b>Lietotāja stāsts</b>	US_ADMN_09
<b>Apraksts</b>	Pirms metodes saņemto ziņu apskatīšanai nostrādā autentifikācijas filtrs, kurš pārbauda vai lietotājam ir loma "Administrator". "GetReceivedMessages" metode meklē visas sistēmā esošās ziņas, kurām kā saņēmējs ir norādīts lietotājs ar identifikatoru 1. Atrastās ziņās tiek sadalītas lapās pēc lietotāja norādītajiem parametriem, vai, ja lietotājs parametrus nav norādījis, izmantojot sistēmas noklusētās vērtības un atgrieztas lietotājam.

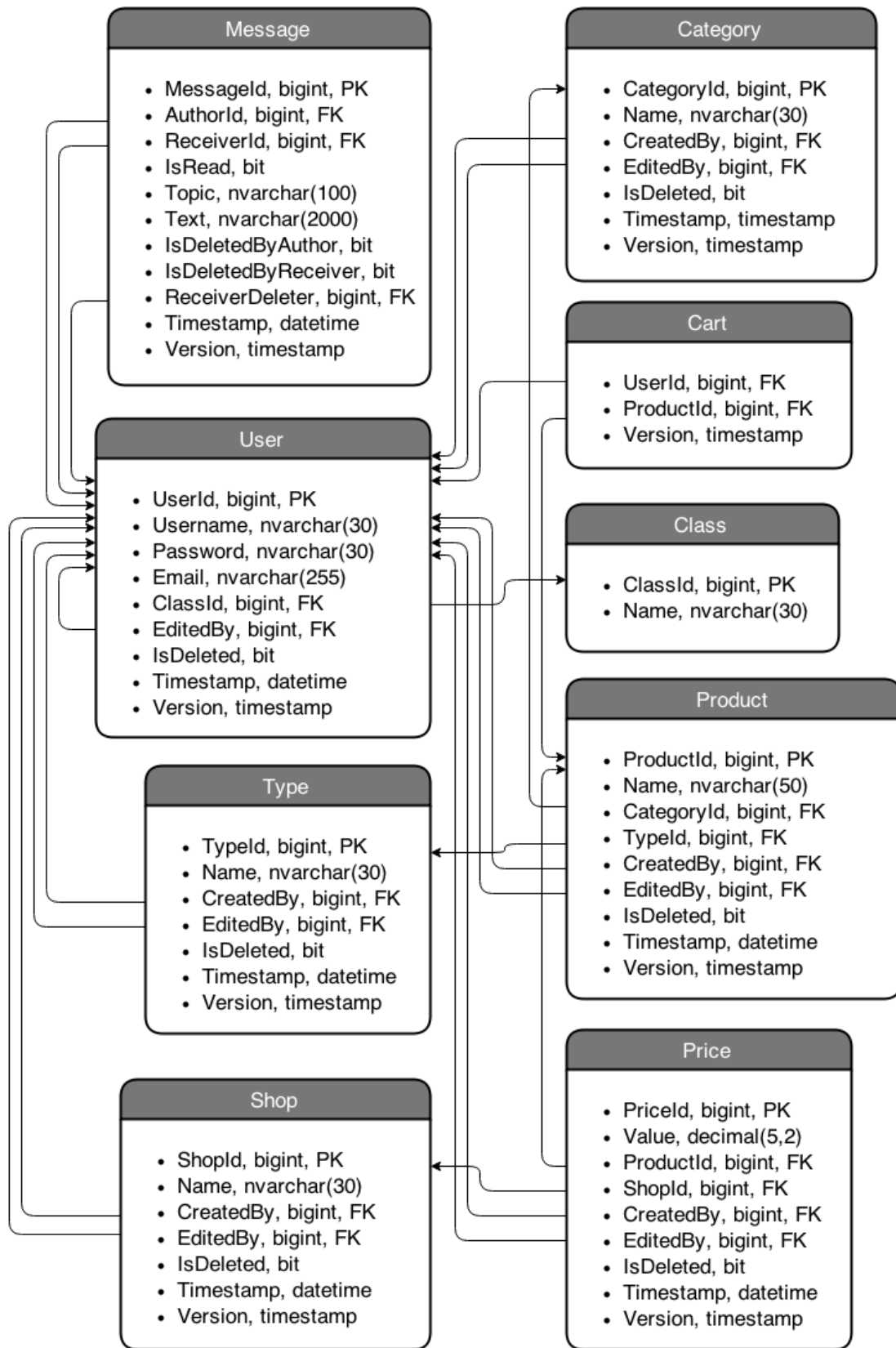
## 4.4 Datu atkarības

### 4.4.1 Konceptuālais ER modelis



Ilustrācija 4.8

#### 4.4.2 Fiziskais ER modelis



Ilustrācija 4.9

## 4.5 Datu dekompozīcija

### 4.5.1 Tabulas “User” lauki

Tabula glabā sistēmas lietotājus. Lauki “Username” un “Email” glabā unikālas vērtības.

Tabula 4.30

LAUKS	TIPS	OBLIGĀTS	ATSLĒGA	APRAKSTS
UserId	bigint	jā	PK	Surogāt identifikators tabulai “User”.
Username	nvarchar(30)	jā		Lietotāja lietotājvārds.
Password	nvarchar(30)	jā		Lietotāja parole.
Email	nvarchar(255)	jā		Lietotāja e-pasta adrese.
ClassId	bigint	jā	FK (Class)	Lietotāja klase. Ārējā atslēga uz tabulu “Class”.
EditedBy	bigint	nē	FK(User)	Lietotāja, kurš ir pēdējais, kas labojis šo ierakstu identifikators.
IsDeleted	bit	jā		Karogs, kurš parāda vai ieraksts ir loģiski dzēsts. Noklusētā vērtība ir false.
Timestamp	datetime	jā		Laiks un datums, kad ieraksts ir izveidots.
Version	timestamp	jā		Sistēmas ģenerēts lauks, ar kura palīdzību nosaka vai ieraksts nav bojāts.

#### 4.5.2 Tabulas “Class” lauki

Tabula glabā lietotāju klašu identifikatoru atšifrējumus. Sistēmā šobrīd ir trīs lietotāju klases: “Blocked User”, “User” un “Administrator”.

Tabula 4.31

LAUKS	TIPS	OBLIGĀTS	ATSLĒGA	APRAKSTS
ClassId	bigint	Jā	PK	Surogāt identifikators tabulai “Class”
Name	Nvarchar(30)	Jā		Klases nosaukums.

#### 4.5.3 Tabulas “Message” lauki

Tabula glabā lietotāju nosūtītos ziņojumus.

Tabula 4.32

LAUKS	TIPS	OBLIGĀTS	ATSLĒGA	APRAKSTS
MessageId	bigint	jā	PK	Surogāt identifikators tabulai “Message”.
AuthorId	bigint	jā	FK (User)	Ziņas autora identifikators. Ārējā atslēga uz tabulu “User”.
ReceiverId	bigint	nē	FK (User)	Ziņas saņēmēja identifikators. Ārējā atslēga uz tabulu “User”. Ja lauks ir tukšs, tad tiek uzskatīts, ka ziņa nosūtīta administrācijai.
IsRead	bit	jā		Karogs, kurš parāda vai ziņa ir izlasīta. Noklusētā vērtība – 0.

Topic	nvarchar(100)	jā		Ziņas tēma.
Text	nvarchar(2000)	jā		Ziņas teksts.
IsDeletedByAuthor	bit	jā		Karogs, kurš parāda vai autors ir izdzēsis nosūtīto ziņu. Noklusētā vērtība ir false.
IsDeletedByReceiver	bit	jā		Karogs, kurš parāda vai saņēmējs ir izdzēsis saņemto ziņu. Noklusētā vērtība ir false.
ReceiverIdentifier	bigint	nē	FK (User)	Saņēmēja identifikators, kurš ir izdzēsis šo ziņu. Lauks domāts administratoru ziņām, bet ir implementēts visām ziņām.
Timestamp	datetime	jā		Laiks un datums, kad ieraksts ir izveidots.
Version	timestamp	jā		Sistēmas ģenerēts lauks, ar kura palīdzību nosaka vai ieraksts nav bojāts.

#### 4.5.4 Tabulas “Category” lauki

Tabula glabā produktu kategorijas. Lietotāji var izveidot produktu kategorijas, lai grupētu produktus.

Tabula 4.33

LAUKS	TIPS	OBLIGĀTS	ATSLĒGA	APRAKSTS
CategoryId	bigint	jā		Surogāt identifikators tabulai “Category”.
Name	nvarchar(30)	jā		Kategorijas nosaukums.
CreatedBy	bigint	jā	FK (User)	Ieraksta izveidotāja identifikators.
EditedBy	bigint	nē	FK(User)	Lietotāja, kurš ir pēdējais, kas labojis šo ierakstu identifikators.
IsDeleted	bit	jā		Karogs, kurš parāda vai ieraksts ir loģiski dzēsts. Noklusētā vērtība ir false.
Timestamp	datetime	jā		Laiks un datums, kad ieraksts ir izveidots.
Version	timestamp	jā		Sistēmas ģenerēts lauks, ar kura palīdzību nosaka vai ieraksts nav bojāts.

#### 4.5.5 Tabulas “Type” lauki

Tabula glabā produktu cenu tipus. Šobrīd sistēmā ir realizēta cena gabalā un cena kilogramā

Tabula 4.34

LAUKS	TIPS	OBLIGĀTS	ATSLĒGA	APRAKSTS
TypeId	bigint	jā		Surogāt identifikators tabulai “Type”.
Name	nvarchar(30)	jā		Cenu tipa nosaukums.
CreatedBy	bigint	jā	FK (User)	Ieraksta izveidotāja identifikators.
EditedBy	bigint	nē	FK(User)	Lietotāja, kurš ir pēdējais, kas labojis šo ierakstu identifikators.
IsDeleted	bit	jā		Karogs, kurš parāda vai ieraksts ir loģiski dzēsts. Noklusētā vērtība ir false.
Timestamp	datetime	jā		Laiks un datums, kad ieraksts ir izveidots.
Version	timestamp	jā		Sistēmas ģenerēts lauks, ar kura palīdzību nosaka vai ieraksts nav bojāts.

#### 4.5.6 Tabulas “Product” lauki

Tabula glabā lietotāju izveidotos produktus. Produktiem ir piesaistīta kategorija un cenas tips. Produkta nosaukuma, kategorijas identifikatora un cenas tipa identifikatora kombinācijai ir jābūt unikālai.

Tabula 4.35

LAUKS	TIPS	OBLIGĀTS	ATSLĒGA	APRAKSTS
ProductId	bigint	jā		Surogāt identifikators tabulai “Product”.
Name	nvarchar(30)	jā		Produkta nosaukums.
CategoryId	bigint	jā	FK (Category)	Produkta kategorijas identifikators. Ārējā atslēga uz tabulu “Category”.
TypeId	bigint	jā	FK (Type)	Produkta cenas tips. Ārējā atslēga uz tabulu “Type”
CreatedBy	bigint	jā	FK (User)	Ieraksta izveidotāja identifikators.
EditedBy	bigint	nē	FK(User)	Lietotāja, kurš ir pēdējais, kas labojis šo ierakstu identifikators.
IsDeleted	bit	jā		Karogs, kurš parāda vai ieraksts ir loģiski dzēsts. Noklusētā vērtība ir false.
Timestamp	datetime	jā		Laiks un datums, kad ieraksts ir izveidots.

Version	timestamp	jā		Sistēmas ģenerēts lauks, ar kura palīdzību nosaka vai ieraksts nav bojāts.
---------	-----------	----	--	--

#### 4.5.7 Tabulas "Cart" lauki

Tabula glabā produktus kurus lietotājs ir pievienojis savam produktu grozam. Tabula ir realizēta kā starptabula starp tabulām "User" un "Product".

Tabula 4.36

LAUKS	TIPS	OBLIGĀTS	ATSLĒGA	APRAKSTS
UserId	bigint	jā	PK (kopā ar ProductId) FK (User)	Lietotāja identifikators, kura grozam pievienota prece. Ārējā atslēga uz tabulu "User".
ProductId	bigint	jā	PK (kopā ar UserId) FK (Product)	Preces identifikators, kura pievienota grozam. Ārējā atslēga uz tabulu "Product".
Version	timestamp	jā		Sistēmas ģenerēts lauks, ar kura palīdzību nosaka vai ieraksts nav bojāts.

#### 4.5.8 Tabulas "Shop" lauki

Tabula glabā sistēmā izveidotos veikalus.

Tabula 4.37

LAUKS	TIPS	OBLIGĀTS	ATSLĒGA	APRAKSTS
ShopId	bigint	jā		Surogāt identifikators tabulai "Shop".

Name	nvarchar(30)	jā		Veikala nosaukums.
CreatedBy	bigint	jā	FK (User)	Ieraksta izveidotāja identifikators.
EditedBy	bigint	nē	FK(User)	Lietotāja, kurš ir pēdējais, kas labojis šo ierakstu identifikators.
IsDeleted	bit	jā		Karogs, kurš parāda vai ieraksts ir loģiski dzēsts. Noklusētā vērtība ir false.
Timestamp	datetime	jā		Laiks un datums, kad ieraksts ir izveidots.
Version	timestamp	jā		Sistēmas ģenerēts lauks, ar kura palīdzību nosaka vai ieraksts nav bojāts.

#### 4.5.9 Tabulas "Price" lauki

Tabula glabā sistēmā pievienotās cenas. Tabula glabā norādi uz produktu un veikalu, lai varētu noteikt kuram produktam un veikalam cena ir pievienota.

Tabula 4.38

LAUKS	TIPS	OBLIGĀTS	ATSLĒGA	APRAKSTS
PriceId	bigint	jā	PK	Surogāt identifikators tabulai "Price".
Value	decimal(5,2)	jā		Produkta cena.
ProductId	bigint	jā	FK (Product)	Produkta, kuram pievienota cena, identifikators. Ārējā

				atslēga uz tabulu "Product".
ShopId	bigint	jā	FK (Shop)	Veikala, kuram produkta cena pievienota, identifikators. Ārējā atslēga uz tabulu "Shop".
CreatedBy	bigint	jā	FK (User)	Ieraksta izveidotāja identifikators.
EditedBy	bigint	nē	FK(User)	Lietotāja, kurš ir pēdējais, kas labojis šo ierakstu identifikators.
IsDeleted	bit	jā		Karogs, kurš parāda vai ieraksts ir loģiski dzēsts. Noklusētā vērtība ir false.
Timestamp	datetime	jā		Laiks un datums, kad ieraksts ir izveidots.
Version	timestamp	jā		Sistēmas ģenerēts lauks, ar kura palīdzību nosaka vai ieraksts nav bojāts.

## 4.6 Lietotnes ieejas punkti

Tīmekļa lietojumprogrammas saskarnē WebApiPrice ir realizēti šādi maršuti:

Tabula 4.39

URI	Metode	Apraksts	Drošība
api/v1/categories	POST	Ļauj lietotājam izveidot jaunu ierakstu "Category" tabulā.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/categories/{categoryId}	GET	Ļauj lietotājam iegūt visus vienas kategorijas produktus.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/messages	POST	Ļauj lietotājam izveidot jaunu ierakstu "Message" tabulā.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/messages/{messageId}	GET	Ļauj lietotājam apskatīt saņemtu vai nosūtītu ziņu "Message" tabulā.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/messages/{messageId}	DELETE	Ļauj lietotājam dzēst saņemtu vai nosūtītu ziņu "Message" tabulā.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/messages/received	GET	Ļauj lietotājam apskatīt visas saņemtās ziņas.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/messages/received/admin	GET	Ļauj lietotājam apskatīt visas administratoriem adresētās ziņas.	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."

api/v1/messages/sent	GET	Ļauj lietotājam izveidot apskatīt visas nosūtītās ziņas.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/prices	POST	Ļauj lietotājam izveidot jaunu ierakstu "Price" tabulā.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/prices/{priceId}	PUT/PATCH	Ļauj lietotājam atjaunot cenas ierakstu.	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/prices/{priceId}	DELETE	Ļauj lietotājam loģiski dzēst cenas ierakstu.	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/products	POST	Ļauj lietotājam izveidot jaunu ierakstu "Product" tabulā.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/products	GET	Ļauj lietotājam iegūt visus sistēmā saglabātos produktus. Atbildi iespējams sadalīt lapās.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/products/{productId}	GET	Ļauj lietotājam iegūt detaļas par vienu produktu	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/products/{productId}	DELETE	Ļauj lietotājam loģiski dzēst vienu produktu	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/products/{productId}	PUT/PATCH	Ļauj lietotājam atjaunot produkta ierakstu.	Ja lietotājs ir autorizēts ar klasi "User", viņš drīkst izmainīt sevi

			izveidotus produktus. Ja lietotājs ir autorizēts ar klasi "Administrator" viņš drīkst mainīt jebkuru ierakstu.
api/v1/products/{productId}/prices	GET	Ļauj lietotājam viena produkta cenas.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/products/search	GET	Ļauj lietotājam meklēt produktus pēc nosaukuma izmantojot pieprasījumā iekļautu simbolu virkni.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/register	POST	Ļauj lietotājam izveidot jaunu ierakstu "User" tabulā.	
api/v1/shops	POST	Ļauj lietotājam izveidot jaunu ierakstu "Shop" tabulā.	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/shops	GET	Ļauj lietotājam iegūt visus veikalus.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/shops/{shopId}	PUT/PATCH	Ļauj lietotājam atjaunot savu lietotāja ierakstu "Shop" tabulā.	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/shops/{shopId}	DELETE	Ļauj lietotājam dzēst vienu veikalu.	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/users	PUT/PATCH	Ļauj lietotājam atjaunot savu lietotāja ierakstu "User" tabulā.	Lietotājam jābūt autorizētam sistēmā ar

			klasi "User" vai "Administrator."
api/v1/users/{userId}	GET	Ļauj lietotājam iegūt detaļas par vienu lietotāju.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/users/{userId}	PUT/PATCH	Ļauj lietotājam atjaunot cita lietotāja ierakstu "User" tabulā	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/users/{userId}/activate	POST	Ļauj lietotājam nomainīt klasi uz "User"	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/users/{userId}/block	POST	Ļauj lietotājam nomainīt klasi uz "Blocked User"	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/users/{userId}/promote	POST	Ļauj lietotājam nomainīt klasi uz "Administrator"	Lietotājam jābūt autorizētam sistēmā ar klasi "Administrator."
api/v1/users/products	PUT	Ļauj lietotājam pievienot savam grozam vairākus produktus.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/users/products	DELETE	Ļauj lietotājam izdzēst visus produktus no sava groza.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/users/products/{productId}	PUT	Ļauj lietotājam pievienot vienu produktu savam grozam.	Lietotājam jābūt autorizētam sistēmā ar klasi "User" vai "Administrator."
api/v1/users/products/{productId}	DELETE	Ļauj lietotājam izdzēst vienu produktu no sava groza.	Lietotājam jābūt autorizētam sistēmā ar

			klasi “User” vai “Administrator.”
--	--	--	--------------------------------------

## 4.7 Programmatūras koda struktūra

Šajā sadaļā tiks aprakstīta projekta pirmkoda struktūra. Šajā sadaļā netiks ietvertas datnes un mapes, kuras tiek izveidotas kopā ar projektu lietojumprogramma “Visual Studio 2013” un projekta laikā netiek izmainītas.

### 4.7.1 Mapes WebApiPrice saturs

Tabula 4.40

Mapes/datnes nosaukums	Satura apraksts
lib	Satur visas trešās puses bibliotēkas un pakotnes, kuras izmanto sistēma WebApiPrice.
log	Satur sistēmas žurnālu failus.
src	Satur sistēmas pirmkodu.

### 4.7.2 Mapes WebApiPrice\src saturs

Tabula 4.41

Mapes/datnes nosaukums	Satura apraksts
WebApiPrice.Common	C# klases bibliotēkas mape. Šī klase ir atbildīga par ietvaram līdzīgu funkcionalitāti, kura nav specifiska ne lietojumprogrammas saskarnei, ne datu bāzei.
WebApiPrice.Data	C# klases bibliotēkas mape. Projekts satur domēna modeli, kā arī datu piekļuves saskarni un palīgklases.
WebApiPrice.Data.SqlServer	C# klases bibliotēkas projekta mape. Satur datu piekļuves realizāciju un <i>NHibernate</i> kartējumu.
WebApiPrice.Web.Api	C# klases bibliotēkas projekta mape. Satur pašas tīmekļa lietojumprogrammas saskarnes realizāciju.
WebApiPrice.Web.Api.Models	C# klases bibliotēkas projekta mape. Satur pakalpojums resursu tipus, jeb modeļus.

WebApiPrice.Web.Common	C# klases bibliotēkas projekta mape. Satur funkcionalitāti, kas kopēja tīmekļa un pakalpojuma lietotnēm.
WebApiPriceDb	Microsoft SQL datubāzes projekts. Satur SQL datubāzes shēmu, kodu un datus.
nuget.config	Satur NuGet konfigurācijas kodu, lai trešās puses bibliotēkas tiktu glabātas lib mapē.
WebApiPrice.sln	“Visual Studio 2013” risinājuma fails.

### 4.7.3 Mapes WebApiPrice\src\WebApiPrice.Common saturs

Tabula 4.42

Mapes/datnes nosaukums	Satura apraksts
Extensions	Satur paplašinājumu klases.
Logging	Satur sistēmas darbību reģistrēšanas realizāciju.
Security	Satur klases, kuras izmantotas, lai nodrošinātu sistēmu.
TypeMapping	Satur tipu kartēšanas konfigurācijas klases.
BusinessRuleViolationsException.cs	Satur kļūdas realizāciju, kura tiek atgriezta gadījumos, kad pārkāpti ierobežojumi, kuri ir noteikti prasībās.
Constants.cs	Satur klasi, kas glabā konstantes šim projektam.
IDateTime.cs	Interfeiss klasei, kas realizē DateTime bibliotēkas funkcionalitāti sistēmas vajadzībām.
DateTimeAdapter.cs	Klase, kas realizē DateTime bibliotēkas funkcionalitāti sistēmas vajadzībām.
PrimitiveTypeParser.cs	Klase datu parsēšanai.

#### 4.7.3.1 Mapes WebApiPrice\src\WebApiPrice.Common\Extensions saturs

Tabula 4.43

Mapes/datnes nosaukums	Satura apraksts
IntExtensions.cs	Paplašinājuma klase metodēm, kuras nepieciešamas, lai sadalītu atbildes lapās.

UriExtensions.cs	Paplašinājuma klase metodēm, kuras nepieciešamas, lai pievienotu saites atbildes ziņojumiem.
------------------	--

#### 4.7.3.2 Mapes WebApiPrice\src\WebApiPrice.Common\Logging saturs

*Tabula 4.44*

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
ILogManager.cs	Interfeiss klasei, kas realizē bibliotēkas log4net izmantošanu sistēmā.
LogManagerAdapter.cs	Klase, kas realizē bibliotēkas log4net izmantošanu sistēmā.

#### 4.7.3.3 Mapes WebApiPrice\src\WebApiPrice.Common\Security saturs

*Tabula 4.45*

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
IUserSession.cs	Interfeiss klasei, kura ļauj piekļūt lietotāja sesijai.

#### 4.7.3.4 Mapes WebApiPrice\src\WebApiPrice.Common\TypeMapping saturs

*Tabula 4.46*

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
IAutoMapper.cs	Interfeiss klasei, kura realizē bibliotēkas “Automapper” izmantošanu sistēmā.
AutoMapperAdapter.cs	Klase, kura realizē bibliotēkas “Automapper” izmantošanu sistēmā.
IAutoMapperTypeConfigurator.cs	Interfeiss klasei, kuru izmanto “AutoMapper”, lai kartētu objektus.

#### 4.7.4 Mapes WebApiPrice\src\WebApiPrice.Data saturs

Tabula 4.47

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
Entities	Satur domēna modeļa realizāciju.
Exceptions	Satur lietotnes kļūdu apstrādi.
QueryProcessors	Satur loģiku mijiedarbībai ar datu bāzi.
PagedDataRequest.cs	Klase, kura implementē parametrus atbilžu dalīšanai lapās.
QueryResult.cs	Klase, kura pārstrādā atgrieztos rezultātus lapās.
ResultsPagingUtility.cs	Klase, kura ierobežo lapu skaitu un izmēru samērīgās vērtības.

##### 4.7.4.1 Mapes WebApiPrice/src/WebApiPrice.Data/Entities saturs

Tabula 4.48

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
IVersionedEntity.cs	Satur versionēšanas klases realizāciju.
Category.cs	Satur domēna modeļa klases Category realizāciju.
Class.cs	Satur domēna modeļa klases Class realizāciju.
Message.cs	Satur domēna modeļa klases Message realizāciju.
Price.cs	Satur domēna modeļa klases Price realizāciju.
Product.cs	Satur domēna modeļa klases Product realizāciju.
Shop.cs	Satur domēna modeļa klases Shop realizāciju.
Type.cs	Satur domēna modeļa klases Type realizāciju.
User.cs	Satur domēna modeļa klases User realizāciju.

#### 4.7.4.2 WebApiPrice/src/WebApiPrice.Data/Exceptions saturs

Tabula 4.49

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
ChildObjectNotFoundException.cs	Apstrādā kļūdu, kad primārā datu modeļa bērns nav atrasts.
DuplicateDataException.cs	Apstrādā kļūdu, kad datu bāzē jau eksistē ieraksts ar norādītiem datiem.
RootObjectNotFoundException.cs	Apstrādā kļūdu, kad primārais datu modelis nav atrasts.
UnauthorizedException.cs	Apstrādā kļūdu, kad lietotājam nav piekļuves noteiktai sistēmas funkcionalitātei.

#### 4.7.4.3 WebApiPrice/src/WebApiPrice.Data/QueryProcessors saturs

Tabula 4.50

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
IAddCategoryQueryProcessor.cs	Interfeiss klasei, kura realizē kategorijas pievienošanu sistēmai.
IAddMessageQueryProcessor.cs	Interfeiss klasei, kura realizē ziņas nosūtīšanu.
IAddPriceQueryProcessor.cs	Interfeiss klasei, kura realizē cenas pievienošanu sistēmai.
IAddProductQueryProcessor.cs	Interfeiss klasei, kura realizē produkta pievienošanu sistēmai.
IAddShopQueryProcessor.cs	Interfeiss klasei, kura realizē veikala pievienošanu sistēmai.
IAddUserQueryProcessor.cs	Interfeiss klasei, kura realizē jauna lietotāja pievienošanu datu bāzei.
IAllProductsQueryProcessor.cs	Interfeiss klasei, kura realizē visu produktu iegūšanu no datu bāzes.
IAllShopsQueryProcessor.cs	Interfeiss klasei, kura realizē visu veikalu iegūšanu no datu bāzes.

IDeleteMessageQueryProcessor.cs	Interfeiss klasei, kura realizē vienas ziņas loģisku dzēšanu no datu bāzes.
IDeleteProductQueryProcessor.cs	Interfeiss klasei, kura realizē viena produkta loģisku dzēšanu no datu bāzes.
IDeleteProductQueryProcessor.cs	Interfeiss klasei, kura realizē viena produkta loģisku dzēšanu no datu bāzes.
IDeleteShopQueryProcessor.cs	Interfeiss klasei, kura realizē viena veikala loģisku dzēšanu no datu bāzes.
IMessageByIdQueryProcessor.cs	Interfeiss klasei, kura realizē vienas ziņas iegūšanu no datu bāzes.
IMessagesReceivedQueryProcessor.cs	Interfeiss klasei, kura realizē saņemto ziņu iegūšanu no datu bāzes.
IMessagesSentQueryProcessor.cs	Interfeiss klasei, kura realizē nosūtīto ziņu iegūšanu no datu bāzes.
IPricesByProductQueryProcessor.cs	Interfeiss klasei, kura realizē cenu iegūšanu produktam no datu bāzes.
IProductByCategoryQueryProcessor.cs	Interfeiss klasei, kura realizē vienas kategorijas visu produktu iegūšanu.
IProductByIdQueryProcessor.cs	Interfeiss klasei, kura realizē viena produkta iegūšanu no datu bāzes izmantojot tā identifikatoru.
ISearchProductQueryProcessor.cs	Interfeiss klasei, kura realizē produktu meklēšanu pēc nosaukuma.
IUpdateCategoryQueryProcessor.cs	Interfeiss klasei, kura realizē kategorijas atjaunošanai nepieciešamās funkcijas.
IUpdatePriceQueryProcessor.cs	Interfeiss klasei, kura realizē cenas atjaunošanai nepieciešamās funkcijas.
IUpdateShopQueryProcessor.cs	Interfeiss klasei, kura realizē veikala atjaunošanai nepieciešamās funkcijas.
IUpdateUserClassQueryProcessor.cs	Interfeiss klasei, kur realizē lietotāja klases izmainīšanu.
IUpdateProductQueryProcessor.cs	Interfeiss klasei, kura realizē produkta ieraksta atjaunināšanai nepieciešamās funkcijas.

IUpdateUserQueryProcessor.cs	Interfeiss klasei, kura realizē groza pārvaldībai un lietotāju ierakstu atjaunināšanai nepieciešamās funkcijas.
IUserByIdQueryProcessor.cs	Interfeiss klasei, kura realizē viena lietotāja iegūšanu no datu bāzes izmantojot tā identifikatoru.

#### 4.7.5 Mapes WebApiPrice\src\WebApiPrice.Data.SqlServer saturs

*Tabula 4.51*

Mapes/datnes nosaukums	Satura apraksts
Mapping	Satur loģiku domēna modeļu klašu kartējumu uz datubāzi.
QueryProcessors	Satur loģiku sistēmas mijiedarbībai ar datu bāzi.

##### 4.7.5.1 Mapes WebApiPrice\src\WebApiPrice.Data.SqlServer\Mapping saturs

*Tabula 4.52*

Mapes/datnes nosaukums	Satura apraksts
VersionedClassMap.cs	Versiju ģenerēšanas implementācija.
CategoryMap.cs	Entītijas Category kartējums.
ClassMap.cs	Entītijas Class kartējums.
MessageMap.cs	Entītijas Message kartējums.
PriceMap.cs	Entītijas Price kartējums.
ProductMap.cs	Entītijas Product kartējums.
ShopMap.cs	Entītijas Shop kartējums.
TypeMap.cs	Entītijas Type kartējums.
UserMap.cs	Entītijas User kartējums.

4.7.5.2 Mapes      WebApiPrice\src\WebApiPrice.Data.SqlServer\QueryProcessors  
saturs

Tabula 4.53

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
AddCategoryQueryProcessor.cs	Klase, kura realizē kategorijas pievienošanu sistēmai.
AddMessageQueryProcessor.cs	Klase, kura realizē ziņas nosūtīšanu.
AddPriceQueryProcessor.cs	Klase, kura realizē cenas pievienošanu sistēmai.
AddProductQueryProcessor.cs	Klase realizē produkta pievienošanu datu bāzei.
AddShopQueryProcessor.cs	Klase realizē veikala pievienošanu datu bāzei.
AddUserQueryProcessor.cs	Klase realizē lietotāja pievienošanu datu bāzei.
AllProductsQueryProcessor.cs	Klase, kas realizē visu produktu iegūšanu no datu bāzes.
AllShopsQueryProcessor.cs	Klase, kas realizē visu veikalu iegūšanu no datu bāzes.
DeleteMessageQueryProcessor.cs	Klase, kura realizē vienas ziņas loģisku dzēšanu no datu bāzes.
DeletePriceQueryProcessor.cs	Klase, kura realizē vienas cenas loģisku dzēšanu no datu bāzes.
DeleteProductQueryProcessor.cs	Klase, kura realizē viena produkta loģisku dzēšanu no datu bāzes.
DeleteShopQueryProcessor.cs	Klase, kura realizē viena veikala loģisku dzēšanu no datu bāzes.
MessageByIdQueryProcessor.cs	Klase, kura realizē vienas ziņas iegūšanu no datu bāzes.
MessagesReceivedQueryProcessor.cs	Klase, kura realizē saņemto ziņu iegūšanu no datu bāzes.

MessagesSentQueryProcessor.cs	Klase, kura realizē nosūtīto ziņu iegūšanu no datu bāzes.
PricesByProductQueryProcessor.cs	Klase, kura realizē cenu iegūšanu produktam no datu bāzes.
ProductByCategoryQueryProcessor.cs	Klase, kura realizē vienas kategorijas visu produktu iegūšanu.
ProductByIdQueryProcessor.cs	Klase realizē viena produkta iegūšanu no datu bāzes izmantojot tā identifikatoru.
SearchProductQueryProcessor.cs	Klase, kura realizē produktu meklēšanu pēc nosaukuma.
UpdateCategoryQueryProcessor.cs	Klase, kura realizē kategorijas atjaunošanai nepieciešamās funkcijas.
UpdatePriceQueryProcessor.cs	Klase, kura realizē cenas atjaunošanai nepieciešamās funkcijas.
UpdateProductQueryProcessor.cs	Klase, kura realizē produkta ieraksta atjaunināšanai nepieciešamās funkcijas.
UpdateShopQueryProcessor.cs	Klase, kura realizē veikala atjaunošanai nepieciešamās funkcijas.
UpdateUserClassQueryProcessor.cs	Klase realizē lietotāja klases izmainīšanu.
UpdateUserQueryProcessor.cs	Klase, kura realizē groza pārvaldība nepieciešamās funkcijas.
UserByIdQueryProcessor.cs	Klase realizē viena lietotāja iegūšanu no datu bāzes izmantojot tā identifikatoru.

#### 4.7.6 Mapes WebApiPrice\src\WebApiPrice.Web.Api saturs

*Tabula 4.54*

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
App_Start	Satur kodu, kas tiek izpildīts, kad lietotne tiek palaista.
AutoMappingConfiguration	Satur lietotnes iekšējo modeļu kartējumu.
Controllers	Satur tīmekļa lietojumprogrammas saskarnes kontrolierus.

InquiryProcessing	Satur klases, kuras apstrādā datu pieprasījumus, kuri prasa datus no datu bāzes.
LinkServices	Satur loģiku, kura nepieciešama, lai pievienotu saites atbildes ziņām.
MaintenanceProcesssing	Satur loģiku, kura tiek izsaukta no kontrolieriem.
Security	Satur klases, kuras ir atbildīgas par sistēmas drošību.
Global.asax	Satur kodu, kas nostrādā pie lietotnes līmeņa notikumiem.

#### 4.7.6.1 Mapes WebApiPrice\src\WebApiPrice.Web.Api\App\_Start saturs

Tabula 4.55

Mapes/datnes nosaukums	Satura apraksts
AutoMapperConfigurator.cs	Satur kodu, kas konfigurē visas <i>IAutoMapperTypeConfigurator</i> instances.
NinjectConfigurator.cs	Satur kodu, kas sasaista klašu deklarācijas ar klašu realizācijām.
NinjectWebCommon.cs	Satur kodu, kas izveido atkarību injekciju konteineri lietotnes ieslēgšanas brīdī.
WebApiConfig.cs	Satur kodu, kas konfigurē pielāgoto lietotnes funkcionalitāti.

#### 4.7.6.2 Mapes WebApiPrice\src\WebApiPrice.Web.Api\AutoMappingConfiguration saturs

Tabula 4.56

Mapes/datnes nosaukums	Satura apraksts
CategoryToCategoryEntityTypeAutoMapperTypeConfigurator.cs	Satur kartējumu no Web.Api.Models.Category uz Data.Entities.Category klases mainīgajiem.
CategoryProductsResolver.cs	Satur kartējumu Data.Entities.Category klases Products sarakstam.

CategoryEntityToCategoryAutoMapperTypeConfigurator	Satur kartējumu no Data.Entities.Category uz Web.Api.Models.Category klases mainīgajiem.
ClassToClassEntityAutoMapperTypeConfigurator.cs	Satur kartējumu no Web.Api.Models.Class uz Data.Entities.Class klases mainīgajiem.
ClassEntityToClassAutoMapperTypeConfigurator.cs	Satur kartējumu no Data.Entities.Class uz Web.Api.Models.Class klases mainīgajiem.
NewMessageToMessageEntityAutoMapperTypeConfigurator	Satur kartējumu no Web.Api.Models.NewMessage uz Data.Entities.Message klases mainīgajiem.
MessageEntityToMessageAutoMapperTypeConfigurator	Satur kartējumu no Data.Entities.Message uz Web.Api.Models.Message klases mainīgajiem.
NewCategoryToCategoryEntityAutoMapperTypeConfigurator.cs	Satur kartējumu no Web.Api.Models.NewCategory uz Data.Entities.Category klases mainīgajiem.
NewProductToProductEntityAutoMapperTypeConfigurator.cs	Satur kartējumu no Web.Api.Models.NewProduct uz Data.Entities.Product klases mainīgajiem.
NewShopToShopEntityAutoMapperTypeConfigurator.cs	Satur kartējumu no Web.Api.Models.NewShop uz Data.Entities.Shop klases mainīgajiem.
NewUserToUserEntityAutoMapperTypeConfigurator.cs	Satur kartējumu no Web.Api.Models.NewUser uz Data.Entities.User klases mainīgajiem.
PriceEntityToPriceAutoMapperTypeConfigurator.cs	Satur kartējumu no Data.Entities.Price uz Web.Api.Models.Price klases mainīgajiem.
NewPriceToPriceEntityAutoMapperTypeConfigurator.cs	Kartējums no Web.Api.Models.NewPrice uz Data.Entities.Price klases mainīgajiem.

ProductEntityToProduct AutoMapperTypeConfigurator.cs	Satur kartējumu no Data.Entities.Product uz Web.Api.Models.Product klases mainīgajiem.
ProductPriceResolver.cs	Satur kartējumu Data.Entities.Product klases Prices sarakstam.
ProductToProductEntity AutoMapperTypeConfigurator.cs	Satur kartējumu no Web.Api.Models.Product uz Data.Entities.Product klases mainīgajiem.
ShopToShopEntityAuto MapperTypeConfigurator.cs	Kartējums no Web.Api.Models.Shop uz Data.Entities.Shop klases mainīgajiem.
ShopEntityToShopAuto MapperTypeConfigurator.cs	Satur kartējumu no Data.Entities.Shop uz Web.Api.Models.Shop klases mainīgajiem.
ShopCategoryeResolver.cs	Satur kartējumu Data.Entities.Shop klases Categories sarakstam.
TypeToTypeEntityAuto MapperTypeConfigurator.cs	Kartējums no Web.Api.Models.Type uz Data.Entities.Type klases mainīgajiem.
TypeEntityToTypeAuto MapperTypeConfigurator.cs	Satur kartējumu no Data.Entities.Type uz Web.Api.Models.Type klases mainīgajiem.
UserToUserEntityAutoM apperTypeConfigurator.cs	Kartējums no Web.Api.Models.User uz Data.Entities.User klases mainīgajiem.
UserEntityToUserAutoM apperTypeConfigurator.cs	Satur kartējumu no Data.Entities.User uz Web.Api.Models.User klases mainīgajiem.
UserProductResolver.cs	Satur kartējumu Data.Entities.User klases Product sarakstam.

#### 4.7.6.3 Mapes WebApiPrice\src\WebApiPrice.Web.Api\Controllers saturs

Tabula 4.57

Mapes/datnes nosaukums	Satura apraksts
V1	Satur tīmekļa lietojumprogrammas saskarnes pirmās versijas kontrolierus.

#### 4.7.6.4 Mapes WebApiPrice\src\WebApiPrice.Web.Api\Controllers\V1 saturs

Tabula 4.58

Mapes/datnes nosaukums	Satura apraksts
CategoriesController.cs	Kontrolieris, kurš atbild par pieprasījumu izpildi, kuri ir saistīti ar kategoriju apstrādi.
MessagesController.cs	Kontrolieris, kurš atbild par pieprasījumu izpildi, kuri ir saistīti ar ziņu apstrādi.
PricesController.cs	Kontrolieris, kurš atbild par pieprasījumu izpildi, kuri ir saistīti ar cenu apstrādi.
ProductsController.cs	Kontrolieris, kurš atbild par pieprasījumu izpildi, kuri ir saistīti ar produkta apstrādi.
RegisterController.cs	Kontrolieris, kurš atbild par pieprasījumu izpildi, kuri ir saistīti ar jauna lietotāja reģistrāciju.
Shopscontroller.cs	Kontrolieris, kurš atbild par pieprasījumu izpildi, kuri ir saistīti ar veikalu pārvaldību.
UserProductsController.cs	Kontrolieris, kurš atbild par pieprasījumu izpildi, kuri ir saistīti ar lietotāja groza pārvaldību.
UsersController.cs	Kontrolieris, kurš atbild par pieprasījumu izpildi, kuri ir saistīti ar lietotāju apstrādi.

#### 4.7.6.5 Mapes WebApiPrice\src\WebApiPrice.Web.Api\InquiryProcessing saturs

Tabula 4.59

Mapes/datnes nosaukums	Satura apraksts
------------------------	-----------------

IAllProductsInquiryProcessor.cs	Interfeiss klasei, kura realizē visu produktu iegūšanai no datu bāzes nepieciešamo kartēšanu.
AllProductsInquiryProcessor.cs	Klase, kura realizē visu produktu iegūšanai no datu bāzes nepieciešamo kartēšanu.
IAllShopsInquiryProcessor.cs	Interfeiss klasei, kura realizē visu veikalu iegūšanai no datu bāzes nepieciešamo kartēšanu.
AllShopsInquiryProcessor.cs	Klase, kura realizē visu veikalu iegūšanai no datu bāzes nepieciešamo kartēšanu.
IDeleteMessageInquiryProcessor.cs	Interfeiss klasei, kura realizē vienas ziņas dzēšanai nepieciešamo kartēšanu.
DeleteMessageInquiryProcessor.cs	Klase, kura realizē vienas ziņas dzēšanai nepieciešamo kartēšanu.
IDeletePriceInquiryProcessor.cs	Interfeiss klasei, kura realizē vienas cenas dzēšanai nepieciešamo kartēšanu.
DeletePriceInquiryProcessor.cs	Klase, kura realizē vienas cenas dzēšanai nepieciešamo kartēšanu.
IDeleteProductInquiryProcessor.cs	Interfeiss klasei, kura realizē viena produkta dzēšanai nepieciešamo kartēšanu.
DeleteProductInquiryProcessor.cs	Klase, kura realizē viena produkta dzēšanai nepieciešamo kartēšanu.
IDeleteShopInquiryProcessor.cs	Interfeiss klasei, kura realizē viena veikala dzēšanai nepieciešamo kartēšanu.
DeleteShopInquiryProcessor.cs	Klase, kura realizē viena veikala dzēšanai nepieciešamo kartēšanu.
IMessageByIdInquiryProcessor.cs	Interfeiss klasei, kura realizē viena ziņojumu iegūšanai no datu bāzes nepieciešamo kartēšanu.
MessageByIdInquiryProcessor.cs	Klase, kura realizē viena ziņojumu iegūšanai no datu bāzes nepieciešamo kartēšanu.
IMessagesReceivedInquiryProcessor.cs	Interfeiss klasei, kura realizē saņemto ziņojumu iegūšanai no datu bāzes nepieciešamo kartēšanu.
MessagesReceivedInquiryProcessor.cs	Klase, kura realizē visu produktu iegūšanai no datu bāzes nepieciešamo kartēšanu.

IMessagesSentInquiryProcessor.cs	Interfeiss klasei, kura realizē nosūtīto ziņojumu iegūšanai no datu bāzes nepieciešamo kartēšanu.
MessagesSentInquiryProcessor.cs	Klase, kura realizē nosūtīto ziņojumu iegūšanai no datu bāzes nepieciešamo kartēšanu.
IPagedDataRequestFactory.cs	Interfeiss klasei, kura izveido lapās sadalītu atbildi no pieprasījuma.
PagedDataRequestFactory.cs	Klase, kura izveido lapās sadalītu atbildi no pieprasījuma.
IPricesByProductInquiryProcessor.cs	Interfeiss klasei, kura realizē viena produkta cenu iegūšanai nepieciešamo kartēšanu.
PricesByProductInquiryProcessor.cs	Klase, kura realizē viena produkta cenu iegūšanai nepieciešamo kartēšanu.
IProductsByCategoryInquiryProcessor.cs	Interfeiss klasei, kura realizē vienas kategorijas produktu iegūšanai nepieciešamo kartēšanu.
ProductsByCategoryInquiryProcessor.cs	Klase, kura realizē vienas kategorijas produktu iegūšanai nepieciešamo kartēšanu.
IProductByIdInquiryProcessor.cs	Interfeiss klasei, kura realizē kartēšanu no Product domēna klases uz Product servisa klasi.
ProductByIdInquiryProcessor.cs	Klase, kura realizē kartēšanu no Product domēna klases uz Product servisa klasi.
ISearchProductInquiryProcessor.cs	Interfeiss klasei, kura realizē produkta meklēšanai pēc nosaukuma nepieciešamo kartēšanu.
SearchProductInquiryProcessor.cs	Klase, kura realizē produkta meklēšanai pēc nosaukuma nepieciešamo kartēšanu.
IUserByIdInquiryProcessor.cs	Interfeiss klasei, kura realizē kartēšanu no User domēna klases uz User servisa klasi.
UserByIdInquiryProcessor.cs	Klase, kura realizē kartēšanu no User domēna klases uz User servisa klasi.

#### 4.7.6.6 Mapes WebApiPrice\src\WebApiPrice.Web.Api\ LinkServices saturs

Tabula 4.60

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
ICommonLinkService.cs	Interfeiss klasei, kura realizē domēnam specifisku saišu pievienošanu ziņām.
CommonLinkService.cs	Klase, kura realizē domēnam specifisku saišu pievienošanu ziņām.
IMessageLinkService.cs	Interfeiss klasei, kura pievieno "Message" servisa modelim specifiskas saites.
MessageLinkService.cs	Klase, kura pievieno "Message" servisa modelim specifiskas saites.
IPriceLinkService.cs	Interfeiss klasei, kura pievieno "Price" servisa modelim specifiskas saites.
PriceLinkService.cs	Klase, kura pievieno "Price" servisa modelim specifiskas saites.
IProductLinkService.cs	Interfeiss klasei, kura pievieno "Product" servisa modelim specifiskas saites.
ProductLinkService.cs	Klase, kura pievieno "Product" servisa modelim specifiskas saites.
IShopLinkService.cs	Interfeiss klasei, kura pievieno "Shop" servisa modelim specifiskas saites.
ShopLinkService.cs	Klase, kura pievieno "Shop" servisa modelim specifiskas saites.
IUserLinkService.cs	Interfeiss klasei, kura pievieno "User" servisa modelim specifiskas saites.
UserLinkService.cs	Klase, kura pievieno "User" servisa modelim specifiskas saites.

4.7.6.7 Mapes WebApiPrice\src\WebApiPrice.Web.Api\MaintenanceProcessing  
saturs

Tabula 4.61

<b>Mapes/datnes nosaukums</b>	<b>Satura apraksts</b>
IActivateUserProcessor.cs	Interfeiss klasei, kura ļauj iestatīt lietotajam klasi "User".
ActivateUserProcessor.cs	Interfeiss klasei, kura ļauj iestatīt lietotajam klasi "User".
IAddCategoryMaintenanceProcessor.cs	Interfeiss klasei, kura realizē NewCategory servisa modeļa klases kartēšanu uz Category domēna modeļa klasi un Category domēna modeļa klases kartēšanu uz Category servisa klasi.
AddCategoryMaintenanceProcessor.cs	Klase, kura realizē NewCategory servisa modeļa klases kartēšanu uz Category domēna modeļa klasi un Category domēna modeļa klases kartēšanu uz Category servisa klasi.
IAddMessageMaintenanceProcessor.cs	Interfeiss klasei, kura realizē NewMessage servisa modeļa klases kartēšanu uz Message domēna modeļa klasi un Message domēna modeļa klases kartēšanu uz Message servisa klasi.
AddMessageMaintenanceProcessor.cs	Klase, kura realizē NewMessage servisa modeļa klases kartēšanu uz Message domēna modeļa klasi un Message domēna modeļa klases kartēšanu uz Message servisa klasi.
IAddPriceMaintenanceProcessor.cs	Interfeiss klasei, kura realizē NewPrice servisa modeļa klases kartēšanu uz Price domēna modeļa klasi un Price domēna modeļa klases kartēšanu uz Price servisa klasi.
AddPriceMaintenanceProcessor.cs	Klase, kura realizē NewPrice servisa modeļa klases kartēšanu uz Price domēna modeļa klasi un Price domēna modeļa klases kartēšanu uz Price servisa klasi.
IAddProductMaintenanceProcessor.cs	Interfeiss klasei, kura realizē NewProduct servisa modeļa klases kartēšanu uz Product domēna modeļa klasi un Product domēna modeļa klases kartēšanu uz Product servisa klasi.

AddProductMaintenanceProcessor.cs	Realizācija klasei, kura realizē NewProduct servisa modeļa klases kartēšanu uz Product domēna modeļa klasi un Product domēna modeļa klases kartēšanu uz Product servisa klasi.
IAddShopMaintenanceProcessor.cs	Interfeiss klasei, kura realizē NewShop servisa modeļa klases kartēšanu uz Shop domēna modeļa klasi un Shop domēna modeļa klases kartēšanu uz Shop servisa klasi.
AddShopMaintenanceProcessor.cs	Klase, kura realizē NewShop servisa modeļa klases kartēšanu uz Shop domēna modeļa klasi un Shop domēna modeļa klases kartēšanu uz Shop servisa klasi.
IAddUserMaintenanceProcessor.cs	Interfeiss klasei, kura realizē NewUser servisa modeļa klases kartēšanu uz User domēna modeļa klasi un User domēna modeļa klases kartēšanu uz User servisa klasi.
AddUserMaintenanceProcessor.cs	Realizācija klasei, kura realizē NewUser servisa modeļa klases kartēšanu uz User domēna modeļa klasi un User domēna modeļa klases kartēšanu uz User servisa klasi.
IBlockUserProcessor.cs	Interfeiss klasei, kura realizē lietotāja bloķēšanu.
BlockUserProcessor.cs	Klase, kura realizē lietotāja bloķēšanu.
IPromoteUserProcessor.cs	Interfeiss klasei, kura realizē lietotāja paaugstināšanu par administratoru.
PromoteUserProcessor.cs	Klase, kura realizē lietotāja paaugstināšanu par administratoru.
IUpdateCategoryMaintenanceProcessor.cs	Interfeiss klasei, kura realizē kategorijas ieraksta atjaunošanai nepieciešamo kartēšanu.
UpdateCategoryMaintenanceProcessor.cs	Klase, kura realizē kategorijas ieraksta atjaunošanai nepieciešamo kartēšanu.
IUpdatePriceMaintenanceProcessor.cs	Interfeiss klasei, kura realizē cenas ieraksta atjaunošanai nepieciešamo kartēšanu.
UpdatePriceMaintenanceProcessor.cs	Klase, kura realizē cenas ieraksta atjaunošanai nepieciešamo kartēšanu.

IUpdateProductMaintenanceProcessor.cs	Interfeiss klasei, kura realizē produkta ieraksta atjaunošanai nepieciešamo kartēšanu.
UpdateProductMaintenanceProcessor.cs	Klase, kura realizē produkta ieraksta atjaunošanai nepieciešamo kartēšanu.
IUpdateShopMaintenanceProcessor.cs	Interfeiss klasei, kura realizē veikala ieraksta atjaunošanai nepieciešamo kartēšanu.
UpdateShopMaintenanceProcessor.cs	Klase, kura realizē veikala ieraksta atjaunošanai nepieciešamo kartēšanu.
IUpdateUserMaintenanceProcessor.cs	Interfeiss klasei, kura realizē lietotāja ieraksta atjaunošanai nepieciešamo kartēšanu.
UpdateUserMaintenanceProcessor.cs	Klase, kura realizē lietotāja ieraksta atjaunošanai nepieciešamo kartēšanu.
IUserProductsMaintenanceProcessor.cs	Interfeiss klasei, kura realizē kartēšanu no User domēna modeļa uz User servisa modeli groza apstrādei.
UserProductsMaintenanceProcessor.cs	Klase, kura realizē kartēšanu no User domēna modeļa uz User servisa modeli groza apstrādei.
LocationLinkCalculator.cs	
MessageCreatedActionResult.cs	Klase, kura izveido atbildes ziņu pēc veiksmīgas ziņas ievietošanas datubāzē.
PriceCreatedActionResult.cs	Klase, kura izveido atbildes ziņu pēc veiksmīgas cenas ievietošanas datubāzē.
ProductCreatedActionResult.cs	Satur loģiku, kas tiek izpildīta pēc tam, kad jauns produkts ir izveidots, lai pievienotu tam saites un pareizu HTTP atbildes kodu.
ValidateCategoryUpdateRequestAttribute.cs	Satur loģiku, kas validē lietotāja pieprasījumu labot kategorijas ierakstu datu bāzē.
ValidatePriceUpdateRequest.cs	Satur loģiku, kas validē lietotāja pieprasījumu labot cenas ierakstu datu bāzē.

ValidateUserUpdateRequestAttribute.cs	Satur loģiku, kas validē lietotāja pieprasījumu labot produkta ierakstu datu bāzē.
ValidateShopUpdateRequestAttribute.cs	Satur loģiku, kas validē lietotāja pieprasījumu labot veikala ierakstu datu bāzē.
ValidateUserUpdateAdminRequestAttribute.cs	Satur loģiku, kas validē administratora pieprasījumu labot lietotāja ierakstu datu bāzē.

#### 4.7.6.8 Mapes WebApiPrice\src\WebApiPrice.Web.Api\Security saturs

Tabula 4.62

Mapes/datnes nosaukums	Satura apraksts
BasicAuthenticationMessageHandler.cs	Klase satur pielāgotu ziņojumu apstrādi vienkāršajai autentifikācijai.
IBasicSecurityService.cs	Satur BasicSecurityService klases deklarāciju.
BasicSecurityService.cs	Klase satur vienkāršās autentifikācijas realizāciju.

#### 4.7.7 Mapes WebApiPrice\src\WebApiPrice.Web.Api.Models saturs

Tabula 4.63

Mapes/datnes nosaukums	Satura apraksts
Category.cs	Satur servisa modeļa klases Category realizāciju.
Class.cs	Satur servisa modeļa klases Class realizāciju.
ILinkContaining.cs	Interfeiss klasei, kura realizē saišu pievienošanu atbildes ziņojumiem.
Link.cs	Satur servisa modeļa klases Link realizāciju.
Message.cs	Satur servisa modeļa klases Message realizāciju.

NewCategory.cs	Satur servisa modeļa klasi NewCategory. Klase realizēta, lai izsargātos no <i>OverPosting</i> uzbrukuma.
NewMessage.cs	Satur servisa modeļa klasi NewMessage. Klase realizēta, lai izsargātos no <i>OverPosting</i> uzbrukuma.
NewPrice.cs	Satur servisa modeļa klasi NewPrice. Klase realizēta, lai izsargātos no <i>OverPosting</i> uzbrukuma.
NewProduct.cs	Satur servisa modeļa klasi NewProduct. Klase realizēta, lai izsargātos no <i>OverPosting</i> uzbrukuma.
NewShop.ce	Satur servisa modeļa klasi NewShop. Klase realizēta, lai izsargātos no <i>OverPosting</i> uzbrukuma.
NewUser.cs	Satur servisa modeļa klasi NewUser. Klase realizēta, lai izsargātos no <i>OverPosting</i> uzbrukuma.
IPageLinkContaining.cs	Interfeiss klasei, kura realizē lapušu saišu pievienošanu atbildes ziņojumiem.
PagedDataInquiryResponse.cs	Klase, kura realizē lapušu saišu pievienošanu atbildes ziņojumiem.
Price.cs	Satur servisa modeļa klases Price realizāciju.
Product.cs	Satur servisa modeļa klases Product realizāciju.
RecentPrices.cs	Satur servisa modeļa klases RecentPrices realizāciju.
Shop.cs	Satur servisa modeļa klases Shop realizāciju.
Type.cs	Satur servisa modeļa klases Type realizāciju.
User.cs	Satur servisa modeļa klases User realizāciju.

#### 4.7.8 Mapes WebApiPrice\src\WebApiPrice.Web.Common saturs

Tabula 4.64

Mapes/datnes nosaukums	Satura apraksts
ErrorHandling	Satur loģiku kļūdu apstrādei.
Routing	Satur tīmekļa lietojumprogrammas maršrutēšanas loģiku.
Security	Satur klases, kuras izmantotas, lai nodrošinātu sistēmu.
Validation	Satur klase, kas atbild par datu validāciju.
IActionTransactionHelper.cs	Satur ActionTranscationHelper klases deklarāciju.

ActionTransactionHelper.cs	Satur kodu, kas pārvalda sesijas kontrolieru metožu izpildes laikā.
IUpdateablePropertyDetector.cs	Intefeiss klasei, kura nosaka, kuri mainīgie objektam ir jāizmaina atkarībā no pieprasījuma.
IObjectUpdateablePropertyDetectors.cs	Klase, kura nosaka, kuri objekta mainīgie ir jāizmaina, atkarībā no pieprasījuma.
NamespaceHttpControllerSelector.cs	Satur pielāgotu kontrolieru izvēles loģiku.
NinjectDependencyResolver.cs	Satur kodu, kas ļauj Ninject servisam pārvaldīt atkarības, kuras ir nepieciešamas izpildes laikā.
SimpleTraceWriter.cs	Satur kodu, kas iespējo trasējuma rakstīšanu failā.
UnitOfWorkActionFilterAttribute.cs	Satur kodu, kas nodrošina, ka kontrolieru klases metodes izmanto konkrētu <i>ISession</i> instanci.
WebContainerManager.cs	Satur kodu, kas atļauj piekļuvi atkarībām, kuras pārvalda <i>IDependencyResolver</i> klase.

#### 4.7.8.1 Mapes WebApiPrice\src\WebApiPrice.Web.Common\ErrorHandling saturs

Tabula 4.65

Mapes/datnes nosaukums	Satura apraksts
GlobalExceptionHandler.cs	Satur kodu, kas aizvieto noklusēto kļūdu apstrādi.
SimpleErrorResult.cs	Satur kodu, kas izveido HTTP atbildi no kļūdu paziņojumiem.
SimpleExceptionHandler.cs	Satur kodu, kas grāmato kļūdas.

#### 4.7.8.2 Mapes WebApiPrice\src\WebApiPrice.Web.Common\Routing saturs

Tabula 4.66

Mapes/datnes nosaukums	Satura apraksts
ApiVersionConstraint.cs	Satur loģiku, kura pārbauda vai tiek izsaukta eksistējoša lietotnes versija.
ApiVersionRoutePrefixAttribute.cs	Satur loģiku, kas automātiski pievieno api/v1 daļu lietotnes ceļiem.

#### 4.7.8.3 Mapes WebApiPrice\src\WebApiPrice.Web.Common\Security saturs

Tabula 4.67

Mapes/datnes nosaukums	Satura apraksts
IWebUserSession.cs	Satur IWebUserSession klases deklarāciju.
UserAuditAttribute.cs	Klase satur implementāciju funkcionalitātei, kas pievieno lietotājvārdu žurnāla ierakstiem par lietotāja izsauktajām funkcijām.
UserSession.cs	Satur UserSession klases realizāciju

#### 4.7.8.4 Mapes WebApiPrice\src\WebApiPrice.Web.Common\Validation saturs

Mapes/datnes nosaukums	Satura apraksts
ValidateModelAttribute.cs	Satur loģiku, kas atļauj izmantot .NET iebūvētu validāciju datu modeļiem.

#### 4.7.9 Mapes WebApiPrice\src\WebApiPriceDb saturs

Tabula 4.68

Mapes/datnes nosaukums	Satura apraksts
Cart.sql	Satur datubāzes tabulas "Cart" izveidošanas skriptu.
Category.sql	Satur datubāzes tabulas "Category" izveidošanas skriptu.
Class.sql	Satur datubāzes tabulas "Class" izveidošanas skriptu.

Message.sql	Satur datubāzes tabulas "Message" izveidošanas skripts.
Price.sql	Satur datubāzes tabulas "Price" izveidošanas skripts.
Product.sql	Satur datubāzes tabulas "Product" izveidošanas skripts.
Shop.sql	Satur datubāzes tabulas "Shop" izveidošanas skripts.
Type.sql	Satur datubāzes tabulas "Type" izveidošanas skripts.
User.sql	Satur datubāzes tabulas "User" izveidošanas skripts.

## 5 TESTĒŠANAS DOKUMENTĀCIJA

### 5.1 Ievads

Testēšana notika paralēli izstrāde, kas ļāva daļu no pamanītajām kļūdām novērst jau izstrādes procesā. Pēc sistēmas izstrādes notika testēšanas atsevišķiem moduļiem, un atrastās kļūdas tika novērstas.

Sistēmā visas ievades tiek pārbaudītas, un, ja sistēma atrod kļūdu ievadītajos laukos tiek izvadīts kļūdas paziņojums.

### 5.2 Moduļu testēšana

Tabula 5.1

<b>Lietotāja reģistrācija</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Ievadīti nepietiekoši dati.	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs lietotāja vārds.	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstoša parole.	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs e-pasta adrese.	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā eksistējošs lietotājvārds	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā eksistējoša e-pasta adrese.	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati.	Datu bāzē tiek izveidots jauns lietotājs un sistēma lietotājam atgriež izveidotā lietotāja datus.	Jā

Tabula 5.2

<b>Lietotāja autorizācija</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Mēģinājums izsaukt metodi bez atbilstošas lomas	Tiek atgriezts kļūdas paziņojums	Jā
Mēģinājums izsaukt metodi ar atbilstošu lomu.	Izsauktā metode tiek izpildīta.	Jā

Tabula 5.3

<b>Lietotāja autentifikācija</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Autentifikācijas mēģinājums ar neeksistējošu lietotāju	Tiek atgriezts kļūdas paziņojums	Jā
Autentifikācijas mēģinājums ar eksistējošu lietotāju	Lietotājs tiek autentificēts un notiek lietotāja autorizācija	Jā

Tabula 5.4

<b>Lietotāja rediģēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Aizpildīts JSON objekts, bez lietotāja laukiem.	Tiek atgriezts neizlabots lietotājs	Jā
Neatbilstošs lietotāja vārds	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstoša parole	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs e-pasts	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā jau eksistējošs lietotājvārds.	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā jau eksistējoša e-pasta adrese.	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek izveidots jauns lietotājs un sistēma lietotājam atgriež izveidotā lietotāja datus.	Jā

Tabula 5.5

<b>Produkta izveidošana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Nepietiekami dati	Tiek atgriezts kļūdas paziņojums	Jā

Neatbilstošs produkta nosaukums	Tiek atgriezts kļūdas paziņojums	Jā
Neeksistējoša kategorija	Tiek atgriezts kļūdas paziņojums	Jā
Neeksistējošs cenas tips	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā jau eksistējoša produkta nosaukuma, kategorijas un cenas tipa kombinācija	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek izveidots jauns produkts un sistēma lietotājam atgriež izveidotā produkta datus.	Jā

Tabula 5.6

<b>Produkta meklēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs JSON objekts	Tiek atgriezts kļūdas paziņojums	Jā
Atbilstošs JSON objekts	Tiek atgriezti atrastie produkti	Jā

Tabula 5.7

<b>Produkta cenu apskatīšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Norādīts izdzēst vai neeksistējošs produkts	Tiek atgriezts kļūdas paziņojums	Jā
Atbilstošs pieprasījums	Tiek atgriezti atrastās cenas	Jā

Tabula 5.8

<b>Vienas kategorijas produktu apskatīšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Norādīta izdzēsta vai neeksistējoša kategorija	Tiek atgriezts kļūdas paziņojums	Jā
Atbilstošs pieprasījums	Tiek atgriezti atrastie produkti	Jā

Tabula 5.9

<b>Produkta rediģēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Aizpildīts JSON objekts, bez produkta laukiem.	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs produkta vārds	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā jau eksistējoša produkta vārda, kategorijas un cenas tipa kombinācija	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek rediģēts produkta ieraksts un sistēma lietotājam atgriež rediģētā produkta datus.	Jā
Administrators labo cita lietotāja izveidoto produktu.	Datu bāzē tiek rediģēts produkta ieraksts un sistēma lietotājam atgriež rediģētā produkta datus.	Jā
Lietotājs labo cita lietotāja izveidoto produktu.	Tiek atgriezts kļūdas paziņojums	Jā

Tabula 5.10

<b>Kategorijas pievienošana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Nepietiekami dati	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs kategorijas nosaukums	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā jau eksistējošs kategorijas nosaukums	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek izveidota jauna kategorija un sistēma lietotājam atgriež izveidotās kategorijas datus.	Jā

Tabula 5.11

<b>Kategorijas rediģēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Aizpildīts JSON objekts, bez kategorijas laukiem.	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs kategorijas vārds	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā jau eksistējoša kategorijas vārds	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek rediģēts kategorijas ieraksts un sistēma lietotājam atgriež rediģētās kategorijas datus.	Jā

Tabula 5.12

<b>Produkta pievienošana grozam</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Neatbilstošs objekts pieprasījuma pievienot vairākus produktus grozam ķermenī.	Tiek atgriezts neizmainīts lietotājs	Jā
Tukšs pieprasījuma pievienot vairākus produktus grozam ķermenis.	Tiek atgriezts neizmainīts lietotājs	Jā
Neeksistējošs produkts pieprasījuma pievienot vairākus produktus grozam ķermenī.	Tiek atgriezts kļūdas paziņojums	Jā
Atbilstošs pieprasījums pievienot vairākus produktus grozam.	Tiek atgriezts lietotājs ar grozam pievienotiem produktiem.	Jā
Neeksistējošs produkts pieprasījumā pievienot vienu produktu	Tiek atgriezts kļūdas paziņojums	Jā

Atbilstošs pieprasījums pievienot vienu produkt grozam.	Tiek atgriezts lietotājs ar grozam pievienotu produktiem.	Jā
---	---	----

Tabula 5.13

<b>Produkta izdzēšana no groza</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Neatbilstošs objekts pieprasījuma dzēst vairākus produktus grozam ķermenī.	Tiek atgriezts neizmainīts lietotājs	Jā
Tukšs pieprasījuma dzēst vairākus produktus grozam ķermenī.	Tiek atgriezts neizmainīts lietotājs	Jā
Neeksistējošs produkts pieprasījuma dzēst vairākus produktus grozam ķermenī.	Tiek atgriezts lietotājs ar izdzēstiem eksistējošiem produktiem, kuri norādīti pieprasījuma ķermenī	Jā
Atbilstošs pieprasījums dzēst vairākus produktus grozam.	Tiek atgriezts lietotājs ar izdzēstiem norādītajiem produktiem no groza	Jā
Neeksistējošs produkts pieprasījumā dzēst vienu produktu	Tiek atgriezts kļūdas paziņojums	Jā
Atbilstošs pieprasījums dzēst vienu produkt grozam.	Tiek atgriezts lietotājs ar no groza izdzēstu norādīto produktu	Jā

Tabula 5.14

<b>Cenas pievienošana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Nepietiekami dati	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstoša cenas vērtība	Tiek atgriezts kļūdas paziņojums	Jā
Neeksistējošs produkts	Tiek atgriezts kļūdas paziņojums	Jā

Neeksistējošs veikals	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek izveidots jauna cena un sistēma lietotājam atgriež izveidotās cenas datus.	Jā

Tabula 5.15

<b>Cenas rediģēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Nepietiekami dati	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstoša cenas vērtība	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek rediģēta norādītā cena un sistēma lietotājam atgriež izveidotās cenas datus.	Jā
Administrators labo cita lietotāja izveidotu cenu	Datu bāzē tiek rediģēts cenas ieraksts un sistēma lietotājam atgriež rediģētās cenas datus.	Jā
Lietotājs labo cita lietotāja izveidoto cenu.	Tiek atgriezts kļūdas paziņojums	Jā

Tabula 5.16

<b>Ziņas nosūtīšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Nepietiekami dati	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstoša ziņas tēma	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs ziņas teksts	Tiek atgriezts kļūdas paziņojums	Jā
Neeksistējošs saņēmējs	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek izveidots jauna ziņa un sistēma lietotājam atgriež izveidotās ziņas datus.	Jā

Tabula 5.17

<b>Ziņas apskatīšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Neeksistējoša ziņa	Tiek atgriezts kļūdas paziņojums	Jā
Ziņa, kurai autorizētais lietotājs nav ne autors, ne saņēmējs	Tiek atgriezts kļūdas paziņojums	Jā
Autors pieprasa ziņu	Tiek atgriezti pieprasītās ziņas dati	Jā
Saņēmējs pieprasa ziņu	Ziņa tiek atzīmēta kā lasīta un atgriezti pieprasītās ziņas dati	Jā

Tabula 5.18

<b>Ziņas dzēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Neeksistējoša ziņa	Tiek atgriezts kļūdas paziņojums	Jā
Ziņa, kurai autorizētais lietotājs nav ne autors, ne saņēmējs	Tiek atgriezts kļūdas paziņojums	Jā
Autors dzēš ziņu	Tiek atzīmēts, ka autors ziņu ir dzēsis un tiek atgrieztas visas nosūtītās ziņas.	Jā
Saņēmējs dzēš ziņu	Tiek atzīmēts, ka saņēmējs ziņu ir dzēsis un tiek atgrieztas visas saņemtās ziņas.	Jā

Tabula 5.19

<b>Visu saņemto ziņu apskatīšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Atbilstošs pieprasījums	Tiek atgrieztas visas ziņas, kuras autentificētais lietotājs ir saņēmis, un saņēmējs nav izdzēsis	Jā

Tabula 5.20

<b>Visu nosūtīto ziņu apskatīšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Atbilstošs pieprasījums	Tiek atgrieztas visas ziņas, kuras autentificētais lietotājs ir nosūtījis, un autors nav izdzēsis	Jā

Tabula 5.21

<b>Lietotāja klases izmaiņšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Neeksistējošs lietotājs	Tiek atgriezts kļūdas paziņojums	Jā
Lietotāja klases "User" nomainīšana uz "User"	Tiek atgriezts kļūdas paziņojums	Jā
Lietotāja klases "Blocked User" nomainīšana uz "Blocked User"	Tiek atgriezts kļūdas paziņojums	Jā
Lietotāja klases "Administrator" nomainīšana uz "Administrator"	Tiek atgriezts kļūdas paziņojums	Jā
Lietotāja klases "Blocked User" nomainīšana uz "Administrator"	Tiek atgriezts kļūdas paziņojums	Jā
Lietotāja klases "User" nomainīšana uz "Blocked User"	Tiek atgrieztas lietotāja detaļas, kuram nomainīta klase.	Jā
Lietotāja klases "User" nomainīšana uz "Administrator"	Tiek atgrieztas lietotāja detaļas, kuram nomainīta klase.	Jā
Lietotāja klases "Administrator" nomainīšana uz "Blocked User"	Tiek atgrieztas lietotāja detaļas, kuram nomainīta klase.	Jā
Lietotāja klases "Administrator" nomainīšana uz "User"	Tiek atgrieztas lietotāja detaļas, kuram nomainīta klase.	Jā
Lietotāja klases "Blocked User" nomainīšana uz "User"	Tiek atgrieztas lietotāja detaļas, kuram nomainīta klase.	Jā

Tabula 5.22

<b>Produkta dzēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Neeksistējošs produkts	Tiek atgriezts kļūdas paziņojums	Jā
Atbilstošs pieprasījums	Norādītais produkts tiek atzīmēts kā dzēsts un tiek atgriezti visi produkti	Jā

Tabula 5.23

<b>Cenas dzēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Neeksistējoša cena	Tiek atgriezts kļūdas paziņojums	Jā
Atbilstošs pieprasījums	Norādītā cena tiek atzīmēta kā dzēsts un tiek atgrieztas visas cenas, kuras pievienotas produktam, kuram pievienota dzēstā cena.	Jā

Tabula 5.24

<b>Veikala pievienošana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Nepietiekami dati	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs veikala nosaukums	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā jau eksistējošs veikala nosaukums	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek izveidots jauns veikals un sistēma lietotājam atgriež izveidotā veikala datus.	Jā

Tabula 5.25

<b>Veikala rediģēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Tukšs pieprasījums	Tiek atgriezts kļūdas paziņojums	Jā
Aizpildīts JSON objekts, bez veikala laukiem.	Tiek atgriezts kļūdas paziņojums	Jā
Neatbilstošs veikala vārds	Tiek atgriezts kļūdas paziņojums	Jā
Sistēmā jau eksistējošs veikala nosaukums	Tiek atgriezts kļūdas paziņojums	Jā
Tiek ievadīti atbilstoši dati	Datu bāzē tiek rediģēts veikala ieraksts un sistēma lietotājam atgriež rediģētā veikala datus.	Jā

Tabula 5.26

<b>Produkta dzēšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Neeksistējošs veikals	Tiek atgriezts kļūdas paziņojums	Jā
Atbilstošs pieprasījums	Norādītais veikals tiek atzīmēts kā dzēsts un tiek atgriezti visi veikali	Jā

Tabula 5.27

<b>Visu nosūtīto ziņu apskatīšana</b>		
<b>Ievadītie dati</b>	<b>Vēlamais rezultāts</b>	<b>Korektums</b>
Lietotājs pieprasa administratoru ziņas	Tiek atgriezts kļūdas paziņojums	Jā
Administrators pieprasa administratoru ziņas	Tiek atgrieztas visas ziņas, kurās kā saņēmējs norādīts administrators un saņēmējs nav dzēsis.	Jā

## 6 PROJEKTA ORGANIZĀCIJA

Šāda kvalifikācijas darba tēma tika izvēlēta, pēc kvalifikācijas darba vadītāja ieteikuma, jo tēmā ir iespējams izvērsties un projektā izstrādātā programmaprodukta sarežģītība ir atkarīga tieši no izstrādātāja prasmēm.

Sistēmas izstrādes sākuma stadijā tika brīvā veidā nodefinētas prasības un lietotāju stāsti. Dokumentācijas izstrāde notika vienlaicīgi ar sistēmas izstrādi, un izstrādes laikā jau izstrādātā dokumentācija tika labota, ja tas bija nepieciešams, jo projektā izmantotais spējās programmatūras izstrādes paveids to atļauj.

Sistēmas izstrāde tika veikta pēc spējās programmatūras izstrādes principa. Sistēmas funkcionalitāte tika aprakstīta ar lietotāja stāstiem. Visas prasības un projektējumus izstrādes laikā bija iespējams mainīt, pielāgojot tās aktuālajai situācijai.

Projekta tehniskā realizācija sastāv no izstrādātās tīmekļa lietojumprogrammas saskarnes, ar kuru iespējams strādāt izmantojot rīkus ar kuriem ir iespējams nosūtīt un saņemt HTTP ziņojumus, šajā projektā tika izmantots "Fiddler4".

Tīmekļa lietojumprogrammas saskarne izstrādātā programmatūras valodā C#, tika izmantots ietvars "APS.NET Web Api2.2" ar tipu kartēšanas pakotni "AutoMapper", datubāzes pieejas pakotni "NHibernate", trasēšanas pakotni "log4net" un atkarību pārvaldības pakotni "Ninject". Datu bāze tika izstrādāta izmantojot Microsoft SQL Server 2012.

Projekts izstrādāts patstāvīgi, konsultējoties ar darba vadītāju.

## **7 KONFIGURĀCIJAS PĀRVALDĪBA**

Izstrādājot sistēmu, tika lietots konfigurācijas pārvaldības rīks *tortoiseSVN*. Tas tika izmantots, lai sekotu līdzi sistēmas izmaiņām, kā, lai nepieciešamības gadījumā varētu atgriezties pie strādājošas projekta versijas.

## **8 KVALITĀTES NODROŠINĀŠANA**

Lai nodrošinātu projekta kvalitāti, tika ievēroti sekojoši principi:

- Sistēmas dokumentācija tika izstrādāta balstoties uz valsts standartiem programminženierija;
- Sistēmas pirmkoda izstrādē tika ievērots vienots stils;
- Tika ievērota objektorientēta programmēšanas pieeja;
- Tika ievērota programmprodukta izstrādes labā prakse;
- Sarežģītākas koda daļas tika komentētas.

## 9 DARBIETILPĪBAS NOVĒRTĒŠANA

### 9.1 Darbietilpības novērtējums

Darbietilpības novērtēšanai ir izvēlēti Basic COCOMO metode.

Sākumā tika plānots, ka risinājuma darba apjoms varētu būt aptuveni 2,5 tūkstoši rindiņu.

Basic COCOMO modelis ir statisks, vienas vērtības modelis, kas aprēķina lietojumprogrammu izstrādes darbietilpību, kā funkciju kas izteikta aplēstajās koda rindiņās.

Basic COCOMO modeļa vienādojumi ir :

$$E = a_b KLOC^{b_b} \text{ un } D = c_b E^{d_b}, \text{ kur}$$

$E$  – darbietilpība personmēnešos,  $KLOC$  – aplēstās koda rindiņas tūkstošos,  $a_b, b_b, c_b$  un  $d_b$  ir konstantes,  $D$  – izstrādes laiks kalendārajos mēnešos.

$a_b, b_b, c_b$  un  $d_b$  vērtības tiek izvēlētas priekš maziem projektiem, jeb “Organiski” projektu koeficienti, kur  $a_b = 2,4, b_b = 1,05, c_b = 2,5$  un  $d_b = 0,38$ .

Aprēķinot sanāk:

$$E = 2,4 * 2,5^{1,05} = 6,3 \text{ personmēneši.}$$

$$D = 2,5 * 6,3^{0,38} = 5 \text{ kalendārie mēneši.}$$

### 9.2 Secinājumi

Lai gan tika aprēķināts, ka sistēmas izstrāde ies 5 personmēnešus, to izdevās izstrādāt aptuveni 3 personmēnešos. To varētu skaidrot ar to, ka ne viss kods darbā bija unikāls un darba beigās daļu no koda varēja pārstrādāt no esošajiem moduļiem.

## 10 NOBEIGUMS

Darba gaitā ir izveidota lietotāja stāstiem atbilstoša sistēma, kā arī tās dokumentācija. Izveidoto sistēmu vēl var pilnveidot ar papildu funkcionalitāti, un turpināt darbu pie tās pārveidošanas par pašregulējošu sistēmu tālākās tās versijās.

Autoram iepriekš nebija pieredzes darbā ar tīmekļa tehnoloģijām. Izstrādājot darbu tika apgūtas iemaņas darbā ar .Net ietvaru un c# programmēšanas valodu. Darba laikā tika iegūtas zināšanas par sistēmas izstrādi izmantojot *agile* metodoloģiju, kā arī labo praksi tīmekļa lietojumprogrammas saskarnes izstrādāšanā.

## 11 IZMANTOTĀ LITERATŪRA

1. LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis, Latvijas Valsts standarts.
2. LVS 72:1996 Ieteicamā prakse PPA aprakstīšanai, Latvijas Valsts standarts.
3. Jamie Kurtz, Brian Wortman “ASP.NET Web API 2: Building a REST Service from Start to Finish” New York, Apress, 2014.
4. “A Brief Summary of the Original COCOMO Model” [tiešsaiste] – [skatīts 31.05.2015.]  
Pieejams:  
<http://www.mhhe.com/engcs/compsci/pressman/information/olc/COCOMO.html>
5. Akadēmisko terminu datubāze [tiešsaiste] – [skatīts 31.05.2015.]  
Pieejams: <http://termini.lza.lv/>
6. Microsoft Developer Network [tiešsaiste] – [skatīts 31.05.2015.]  
Pieejams: <https://msdn.microsoft.com/>
7. Stackoverflow [tiešsaiste] - [skatīts 31.05.2015.]  
Pieejams: <http://stackoverflow.com/>

Kvalifikācijas darbs „Preču cenu uzraudzības 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: Oskars Kalniņš \_\_\_\_\_ .05.2015.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: prof. Guntis Arnicāns \_\_\_\_\_ .05.2015.

Recenzents: M.dat. Pēteris Krastiņš

Darbs iesniegts 01.06.2015.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: Darja Solodovņikova \_\_\_\_\_

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

\_\_\_\_.06.2015. prot. Nr. \_\_\_\_\_

Komisijas sekretārs(-e): \_\_\_\_\_