**Profesionālās izglītības kompetences centrs**

**„Rīgas Valsts tehnikums”**

Izglītības programma: Programmēšana

**KVALIFIKĀCIJAS DARBS**

**Internacionāls mežsaimniecības portāls uzņēmēju un privātpersonu komunikācijas veicināšanai**

Paskaidrojošais raksts 68 lpp.

Audzēknis: Aleksejs Zubovs

Vadītājs: Normunds Pauders

Normu kontrole: Olga Sabanska

**Rīga 2021**

Saturs

[Ievads 4](#_Toc74218210)

[1. Uzdevumu nostādne 5](#_Toc74218211)

[2. Prasību specifikācija 6](#_Toc74218212)

[2.1. Ieejas informācijas apraksts 6](#_Toc74218213)

[2.1.1. Ieejas informācijas apraksts 6](#_Toc74218214)

[2.1.2. Izejas informācijas apraksts 7](#_Toc74218215)

[2.2. Funkcionālās prasības 8](#_Toc74218216)

[2.3. Nefunkcionālās prasības 10](#_Toc74218217)

[3. Uzdevumu izvēles risināšanas platformas 11](#_Toc74218218)

[4. Programmatūras produkta modelēšana un projektēšana 13](#_Toc74218219)

[4.1. Sistēmas struktūras modelis 13](#_Toc74218220)

[4.1.1. Sistēmas arhitektūra 13](#_Toc74218221)

[4.1.2 Sistēmas ER modelis 14](#_Toc74218222)

[4.2. Datu plūsmju modelis 16](#_Toc74218223)

[4.2.1. Reģistrācijas process 16](#_Toc74218224)

[4.2.2. Lietotāja ieejas process 17](#_Toc74218225)

[4.2.3. Profila rediģēšanas process 19](#_Toc74218226)

[4.2.4. E-pasta apstiprinājums 22](#_Toc74218227)

[4.2.5. Paroles nomainīšana 24](#_Toc74218228)

[4.2.6. Preču meklēšana 25](#_Toc74218229)

[4.2.7. Preču izveidošana 27](#_Toc74218230)

[4.2.8. Noliktavu izveidošana 29](#_Toc74218231)

[4.2.9. Produkta pieveduma noliktavā pievienošana 31](#_Toc74218232)

[4.2.10. Piegādes maršruta pievienošana 33](#_Toc74218233)

[4.2.11. Sūdzības iesniegšana 35](#_Toc74218234)

[4.2.12. Vēstules iesniegšana 37](#_Toc74218235)

[4.2.13. Konta frizēšana 38](#_Toc74218236)

[4.2.14. Konta bloķēšana 38](#_Toc74218237)

[5. Datu struktūras apraksts 40](#_Toc74218238)

[6. Lietotāja ceļvedis 52](#_Toc74218239)

[Pielikumi 56](#_Toc74218240)

# Ievads

Šī kvalifikācijas darba mērķis ir izveidot ērtu tiešsaistes veikalu, kurā tiek savākts visvairāk preču no visdažādākajām kategorijām. Šis veikals ir tirdzniecības platforma ar bezmaksas piekļuvi. Katram sistēmā reģistrētam lietotājam ir tiesības pieteikties, lai kļūtu par pārdevēju.

Šis veikals nodrošina ērtu preču meklēšanu. Tas tiek panākts, pateicoties daudzajiem katalogā uzstādītajiem filtriem. Katram produktam ir savs tagu komplekts, kas ļauj lietotājam atrast tieši to, kas viņam nepieciešams. Meklēšanas algoritmi pārbaudīs atbilstību gan produktu nosaukumam, gan to aprakstam.

Veikals sniedz iespēju ērti kļūt ar pārdevēju. Lietotājam pārdevēja lomā ir ļoti ērti pievienot, modificēt un dzēst produktus no datu bāzes. Tomēr visas ar produktiem saistītās izmaiņas datu bāzē tiek saglabātas. Tas ļauj atklāt krāpnieciskas shēmas. Ja tiks atklātas krāpnieciskas shēmas, pārdevēja konts nekavējoties tiks iesaldēts un bloķēts. Tomēr, pat neskatoties uz bloķēšanu, visi sūtītie sūtījumi tiks piegādāti adresātiem.

Par lietojumprogrammas lietošanas noteikumu pārkāpumiem moderatoriem jāuzrauga. Moderatoram nav tiesību bloķēt kontu, taču viņš to var iesaldēt.

# Uzdevumu nostādne

Šī kvalifikācijas darba mērķis ir izveidot ērtu tirdzniecības platformu. Katram lietotājam ir iespēja reģistrēties bez maksas. Savukārt katram reģistrētam lietotājam ir iespēja veikt neierobežotu pirkumu skaitu. Ir arī iespēja pieteikties, lai kļūtu par pārdevēju. Pēc pirkuma veikšanas ir iespēja iesniegt sūdzību pret pārdevēju. Šajā gadījumā tiek atvērts strīds, kura iznākumu nosaka moderators. Strīda laikā, par kuru notiek domstarpības, puses sarunājas sarunā, lai pierādītu savu lietu. Nepareizas pārdevēja apsūdzības gadījumā lietotājam briesmas nedraud. Savukārt pārdevējiem var būt pienākums maksāt atmaksu.

Pārdevēja lietotājiem ir ļoti ērti kontrolēt savus produktus.

Interneta veikala sistēma lietotājiem nodrošina šādas lomas: lietotājs, pārdevējs, piegādātājs, moderators, administrators, īpašnieks.

Kopumā sistēmai ir jābūt šādai funkcionalitātei:

* jauna lietotāja reģistrācija;
* piesakieties sistēmā;
* profila rediģēšana reģistrētiem un jau ielogotajiem lietotājiem;
* skatīt produktus katalogā;
* preces pievienošana grozam;
* preču pasūtīšana no groza vai tieši no kataloga;
* pieteikšanās kļūt par pārdevēju;
* preču un noliktavu pārvaldība lietotājiem ar pārdevēja lomu.

# Prasību specifikācija

## 2.1. Ieejas informācijas apraksts

### 2.1.1. Ieejas informācijas apraksts

* Informācija par lietotāja reģistrācijas formu, kas sastāv no:
  + lietotājvārda, kas ir unikāla simbolu virkne, ne īsāka par 6 simboliem;
  + e-pasta, kas ir unikāla simbolu virkne;
  + paroles, kas ir simbolu virkne, ne īsāka par 6 simboliem;
  + paroles apstiprināšanas, kas ir simbolu virkne.
* Informācija par lietotāja ieejas formu, kas sastāv no:
  + lietotājvārda vai e-pasta, simbolu virkne;
  + paroles, kas ir simbolu virkne.
* Informācija par lietotāja profila formu, kas sastāv no:
  + lietotājvārda, kas ir simbolu virkne;
  + vārda, kas ir simbolu virkne;
  + uzvārda, kas ir simbolu virkne;
  + e-pasta, kas ir unikāla simbolu virkne;
  + tālruna numura, kas ir simbolu virkne;
  + valsts, kas ir simbolu virkne, attiecās uz tabulu ar valstīm;
  + pilsētas, kas ir simbolu virkne, attiecās uz tabulu ar pilsētām;
  + adreses, kas ir simbolu virkne.
* Informācija par kategoriju datubāzē, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + ikonas, kas ir simbolu virkne (ceļš uz ikonu).
* Informācija par subkategoriju datubāzē, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + kategorijas, kas ir simbolu virkne, attiecās uz tabulu ar kategorijām;
  + ikonas, kas ir simbolu virkne (ceļš uz ikonu).
* Informācija par valūtu datubāzē, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + saīsināta apzīmējuma, kas ir simbolu virkne;
  + simbola apzīmējuma, kas ir simbolu virkne;
  + cenas, kas ir vērtība ar peldējošo komatu.
* Informācija par valsts datubāzē, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + saīsināta garuma, kas ir simbolu virkne;
  + valūtas, ko izmanto šī valsts, kas ir simbolu virkne, attiecās uz tabulu ar valūtām.
* Informācija par pilsētām datubāzē, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + valsts, kas ir simbolu virkne, attiecās uz tabulu ar valstīm.
* Informācija par ražotājiem datubāzē, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + mājaslapas adreses, kas ir simbolu virkne;
  + ikonas, kas ir simbolu virkne (ceļš uz ikonu).
* Informācija par produktu, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + kategorijas, kas ir simbolu virkne;
  + subkategorijas, kas ir simbolu virkne;
  + pārdevēja lietotājvārda, kas ir simbolu virkne;
  + produkta bildes, kas ir simbolu virkne (ceļš uz bildi).
* Informācija par produkta variantu, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + produkta, kas ir simbolu virkne, attiecās uz tabulu ar produktiem;
  + valūtas, kas ir simbolu virkne, attiecās uz tabulu ar valūtām;
  + cenas, kas ir vērtība ar peldējošo komatu;
  + bildes, kas ir simbolu virkne (ceļš uz bildi).
* Informācija par noliktavu, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + pilsētas, kas ir simbolu virkne, attiecās uz tabulu ar pilsētām;
  + pārdevēja lietotājvārda, kas ir simbolu virkne.
* Informācija par pievedumu, kas sastāv no:
  + produkta varianta, kas ir simbolu virkne, attiecās uz tabulu ar produkta variantiem;
  + noliktavas, kas ir simbolu virkne, attiecās uz tabulu ar noliktavām;
  + daudzuma, kas ir vesels skaitlis.

### 2.1.2. Izejas informācijas apraksts

* Informācija par lietotāja profilu, kas sastāv no:
  + lietotājvārda, kas ir simbolu virkne;
  + vārda, kas ir simbolu virkne;
  + uzvārda, kas ir simbolu virkne;
  + e-pasta, kas ir simbolu virkne;
  + tālruna numura, kas ir simbolu virkne;
  + valsts, kas ir simbolu virkne;
  + pilsētas, kas ir simbolu virkne;
  + adreses, kas ir simbolu virkne.
* Informācija par kategoriju, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + ikonas, kas ir simbolu virkne (ceļš uz ikonu).
* Informācija par subkategoriju, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + kategorijas, kas ir simbolu virkne, attiecās uz tabulu ar kategorijām;
  + ikonas, kas ir simbolu virkne (ceļš uz ikonu).
* Informācija par valūtu, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + saīsināta apzīmējuma, kas ir simbolu virkne;
  + simbola apzīmējuma, kas ir simbolu virkne;
  + cenas, kas ir vērtība ar peldējošo komatu.
* Informācija par valsts, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + saīsināta garuma, kas ir simbolu virkne;
  + valūtas, ko izmanto šī valsts, kas ir simbolu virkne, attiecās uz tabulu ar valūtām.
* Informācija par pilsētu, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + valsts, kas ir simbolu virkne, attiecās uz tabulu ar valstīm.
* Informācija par ražotāju, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + mājaslapas adreses, kas ir simbolu virkne;
  + ikonas, kas ir simbolu virkne (ceļš uz ikonu).
* Informācija par produktu, kas sastāv no:
  + nosaukuma, kas ir simbolu virkne;
  + kategorijas, kas ir simbolu virkne;
  + subkategorijas, kas ir simbolu virkne;
  + pārdevēja lietotājvārda, kas ir simbolu virkne;
  + produkta bildes, kas ir simbolu virkne (ceļš uz bildi).

## Funkcionālās prasības

* Lietotāja reģistrācija:
  + obligātie lauki reģistrācijas procesā. Pārbaudes katra lauka pareizumam;
  + e-pasta apstiprināšanas sistēma. Pēc lietotāja reģistrēšanas ir iespējams apstiprinājuma e-pastu. Lietotājam ir jābūt ierobežotai funkcionalitātei ar neapstiprinātu e-pastu. Lietotāji ar neapstiprinātu e-pastu nevar kļūt par pārdevēju vai piegādes personu;
  + paroles šifrēšana.
* Lietotāja profila iespējas:
  + apskatīt un mainīt profila datus. Daži lauka dati ir jāparbaud server;
  + nosūtīt e-pasta apstiprinājuma pieprasījumu pa e-pastu;
  + nosūtīt pieprasījumu kļūt par pārdevēju;
  + nosūtīt pieprasījumu kļūt par piegādes vīrieti;
  + ievietot vai mainīt profila bildi.
* Lietotāja ieeja:
  + lietotājam jāspēj pieteikties, izmantojot savu lietotājvārdu un vai e-pastu;
  + ja lietotājam neizdodas pieteikties, jāparāda kļūda;
  + pēc 5 neveiksmīgām pieteikšanās reizēm lietotājs ir bloķēts. Pagaidu bloķēšana ilgst pusstundu;
* Lietotāja iespējas:
  + apskatīt katalogu produktu sarakstu;
  + norādīt valūtu, kurā tiks norādītas visas cenas;
  + diltrēt un kārtot produktus katalogā gan pēc cenas, gan pēc kategorijas;
  + pievienot produktus grozam;
  + veikt pasūtījumus produktiem gan no kataloga, gan no groza;
  + pasūtot preci, izvēlieties piegādes maršrutu;
  + pēc pasūtījuma iesniegt sūdzību par produktu.
* Īpašnieka iespējas:
  + lietotājam ar īpašnieka lomu vajadzētu būt iespējai rediģēt datus savā datu bāzē no tīmekļa lietojumprogrammas.
* Pārdevēja iespējas:
  + izveidot modificēt un dzēst noliktavas;
  + izveidot modificēt un dzēst produktus un to variantus;
  + izveidot pārskatus par produktu piegādēm uz noliktavām.
* Piegādātāja iespējas:
  + pievienot, rediģēt un dzēst piegādes maršrutus;
  + pieņemt un noraidīt produktu transportēšanas pieprasījumus.
* Maksājumi ar Paypal:
  + paypal API implementēšana lapā;
  + droša maksājuma id un šifra koda glabāšana datubāzē;
  + statusa glabāšana datubāzē, lai viegli var izsekot, kas ir samaksājis un kas nav.

## Nefunkcionālās prasības

* Prasības procesam:
  + datu bāzi jāglabā Microsoft SQL Server;
  + front-end un back-end ir jāuzraksta C Sharp valodā asp.net Core platformā;
  + izstrādē jāizmanto BOOTSTRAP.
* Prasības produktam:
  + lietotāja mijiedarbībai ar sistēmu jābūt angļu valodā.
* Prasības lietotāja interfeisam:
  + lietojumprogrammas saskarnei jābūt vieglai un neradot acu nogurumu;
  + saskarnei jābūt intuitīvai pat personai, kas nav pazīstama ar šo projektu;
  + saskarnei jāspēj pielāgoties ierīces izmēram, no kura tiek palaista lietojumprogramma.

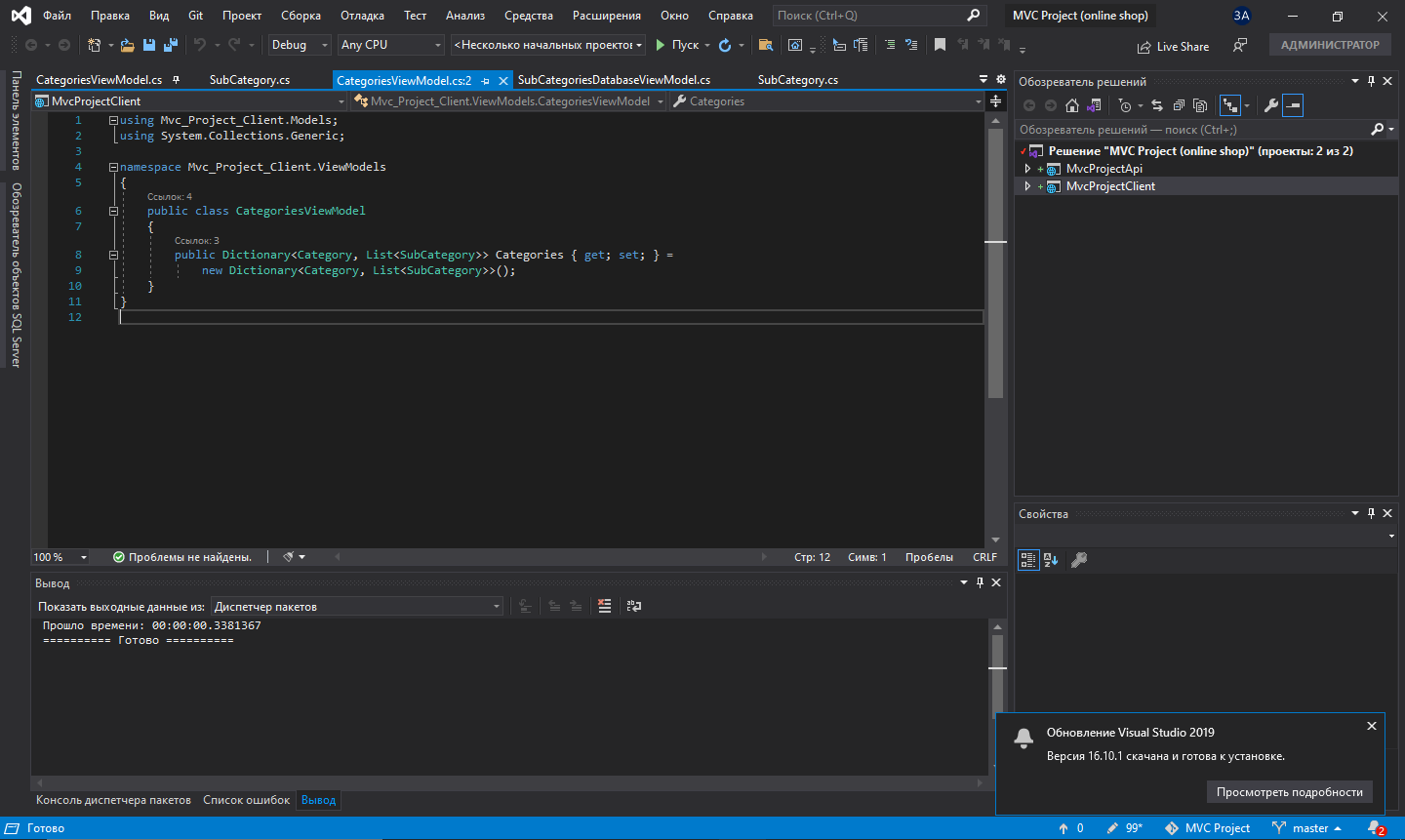
# Uzdevumu izvēles risināšanas platformas

Es izvēlējos ASP.NET Core kā galveno attīstības vidi savam kvalifikācijas projektam. Šis lēmums ir saistīts ar faktu, ka šobrīd Microsoft aktīvi atbalsta tehnoloģiju.

ASP.NET Core ir bezmaksas un atvērta koda tīmekļa sistēma un ASP.NET pēctece, ko izstrādājusi Microsoft. Tas ir modulārs ietvars, kas darbojas gan pilnā .NET Framework, operētājsistēmā Windows, gan starpplatformu .NET Core. Tomēr ASP.NET Core 3. versija darbojas tikai ar .NET Framework .NET Framework atbalstu.

Sistēma ir pilnīga pārrakstīšana, kas apvieno iepriekš atsevišķos ASP.NET MVC un ASP.NET Web API vienā programmēšanas modelī.

Neskatoties uz to, ka tā ir jauna sistēma, kas veidota uz jauna tīmekļa kaudzes, tai ir augsta koncepcijas saderības pakāpe ar ASP.NET. ASP.NET Core lietojumprogrammas atbalsta versiju versijas, kurās dažādas lietojumprogrammas, kas darbojas vienā mašīnā, var atlasīt dažādas ASP.NET Core versijas. Tas nav iespējams ar iepriekšējām ASP.NET versijām.

Veidojot tīmekļa lietojumprogrammu asp.net Core, ir izvēle starp vairākiem dizaina modeļiem. savā tīmekļa lietojumprogrammā es izvēlējos dizaina modeli ar nosaukumu MVC. Šis saīsinājums apzīmē modeis-skats-kontrolieris. No lietotāja izejošos datus kontrolieris saņem modeļa formā, pēc tam tas apstrādā šos datus un pēc tam atdod veidlapu lietotājam. 

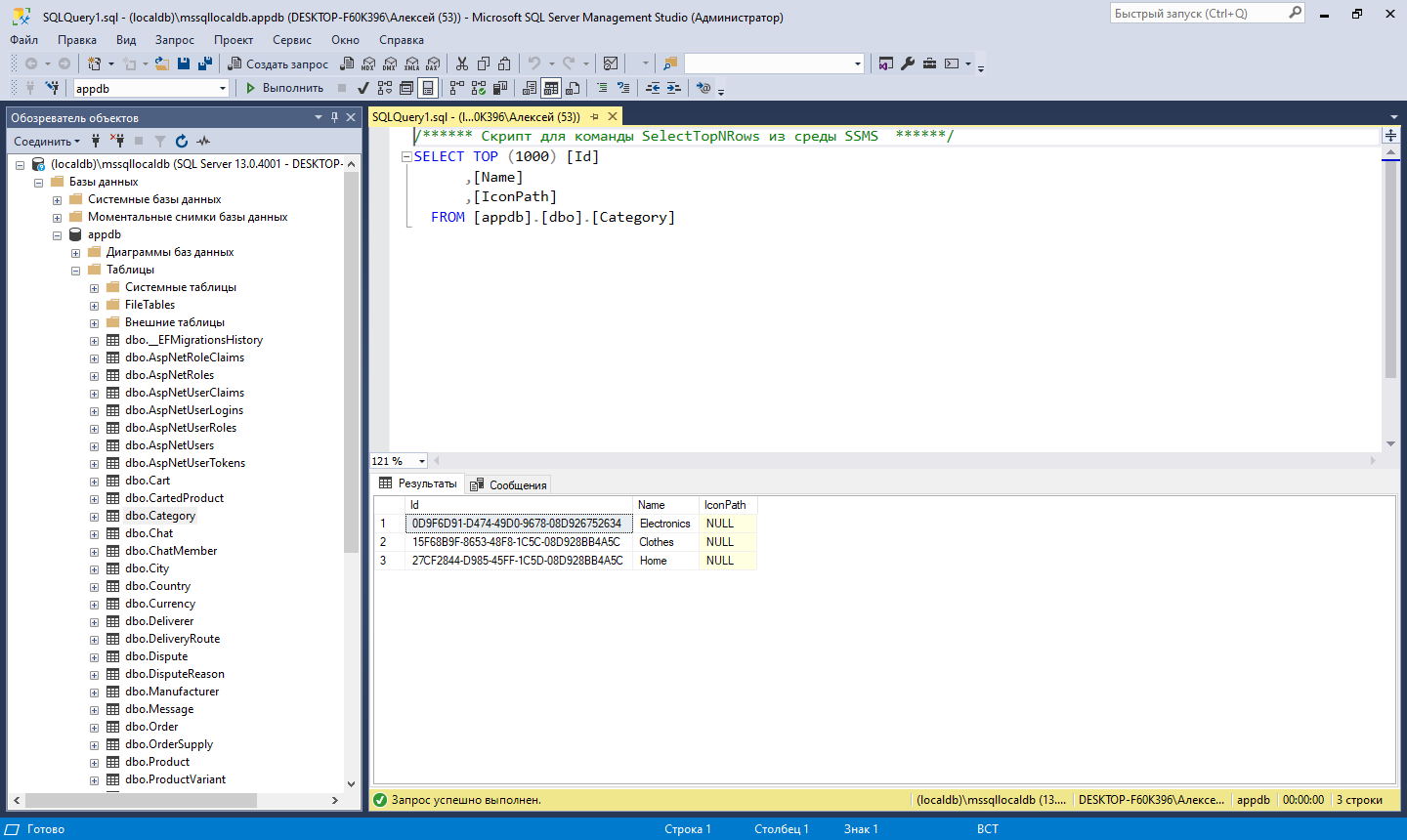
Attēls 3.1 Visual Studio ASP.NET Core lietotne

Mana datu bāze tiek glabāta Microsoft SQL Server. lai pārvaldītu datu bāzi, kuru izmantoju Microsoft SQL Server Management Studio.

SQL Server Management Studio (SSMS) ir programmatūras lietojumprogramma, kas pirmo reizi tika palaista kopā ar Microsoft SQL Server 2005 un tiek izmantota visu Microsoft SQL Server komponentu konfigurēšanai, pārvaldībai un administrēšanai. Tas ir Enterprise Manager pēctecis SQL 2000 vai agrāk. Rīks ietver gan skriptu redaktorus, gan grafiskos rīkus, kas darbojas ar servera objektiem un funkcijām.

SSMS centrālā iezīme ir Object Explorer, kas ļauj lietotājam pārlūkot, atlasīt un rīkoties ar jebkuru no serverī esošajiem objektiem. Tas arī piegādāja atsevišķu Express izdevumu, kuru varēja brīvi lejupielādēt, tomēr jaunākās SSMS versijas pilnībā spēj izveidot savienojumu un pārvaldīt jebkuru SQL Server Express instanci. Microsoft arī integrēja atpakaļejošu saderību ar vecākām SQL Server versijām, tādējādi ļaujot jaunākai SSMS versijai izveidot savienojumu ar vecākām SQL Server gadījumu versijām. Tas tiek piegādāts arī kopā ar Microsoft SQL Server Express 2012, vai arī lietotāji to var lejupielādēt atsevišķi.

Sākot ar 11. versiju, lietojumprogrammas pamatā bija Visual Studio 2010 čaula, lietotāja interfeisam izmantojot WPF. 18. un jaunākas versijas ir balstītas uz Visual Studio 2017 izolēto apvalku.

2015. gada jūnijā Microsoft paziņoja par savu nodomu atbrīvot nākotnes SSMS versijas neatkarīgi no SQL Server datu bāzes dzinēju izlaidumiem.

Attēls 3.2 Microsoft SSMS

Lielākajā daļā gadījumu datu bāzes korekcija notika no Visual Studio, bet dažreiz, kad datu bāzes atjaunināšana prasīja nepamatoti lielas pūles, tā tika labota no Microsoft SQL Server Management Studio.

# Programmatūras produkta modelēšana un projektēšana

## 4.1. Sistēmas struktūras modelis

### 4.1.1. Sistēmas arhitektūra

Kopumā veikala sistēma sastāvēs no 6 daļām:

1. lietotāja daļa;
2. pārdevēja daļa;
3. piegādātāja daļa;
4. moderatora daļa;
5. administratora daļa;
6. īpašnieka daļa.

***Lietotāja daļai*** ir reģistrācijas un pieteikšanas iespēja. Pēc reģistrēšanās un pieteikšanās sistēmā lietotājam ir iespēja rediģēt savus profila datus. Lietotājs var pārlūkot produktu katalogu, nepiesakoties, bet, lai pievienotu produktu grozam vai pasūtītu produktu, joprojām ir jāpiesakās. Pēc preču pasūtīšanas un samaksas par to lietotājam ir iespēja sākt strīdu. Pēc strīda beigām pārdevējam var būt pienākums samaksāt klientam visas preces izmaksas. Ja klients kādu laiku no pārdevēja nesaņem atmaksu, viņš var iesniegt pārdevējam sūdzību. Lietotājs ir zemākais šī veikala lomu hierarhijas līmenis, tāpēc visām pārējām lomām ir tādas pašas atļaujas kā parastajiem lietotājiem.

***Pārdevēja daļā*** ir jūsu preču pārvaldības funkcijas. Pārdevējs var pievienot, dzēst un mainīt savus produktus. Turklāt visas datu bāzes izmaiņas vai dzēsumi tiek saglabāti, lai nepieļautu krāpnieciskas shēmas no pārdevēja puses.

***Piegādātāja daļā*** ir funkcionalitāte preču piegādes maršrutu kontrolei. Pārvadātājs var pievienot, dzēst un mainīt piegādes maršrutus un to parametrus. Turklāt visas datu bāzes izmaiņas vai dzēsumi tiek saglabāti, lai nepieļautu krāpnieciskas shēmas no piegādātāja puses. Pārvadātājam nekavējoties jāatstāj informācija par paku piegādi. Pārvadātājam ir tiesības izņemt sūtījumu, ja tā izmēri un svars neatbilst pārdevēja deklarētajiem.

***Moderatoru daļā*** ir funkcionalitāte darbam ar klientiem, pārdevējiem un piegādātājiem. Moderatoram ir iespēja nodot lietotāju pārdevēja vai piegādātāja lomai. Moderatoram ir iespēja redzēt visus pašreizējos strīdus starp klientu un pārdevēju. Moderators var izbeigt šos strīdus un uzlikt pārdevējam pienākumu samaksāt atmaksu. Moderators arī redz visas lietotāju sūdzības par pārdevējiem. Tas var iesaldēt pārdevējus un viņu preces, rezultātā saldētas preces un šo preču pārdevēji katalogā netiek rādīti.

***Administratora daļā*** ir funkcionalitāte pārdevēju vai piegādātāju lietotāju īslaicīgai vai mūžīgai bloķēšanai. Administrators var reģistrēt vai noņemt moderatorus.

***Īpašnieka daļā*** ir funkcionalitāte datu bāzes pilnīgai kontrolei. Īpašnieks var reģistrēt un noņemt administratorus. Ja īpašnieku nav, sistēma automātiski izveido profilu ar īpašnieka lomu. Tas ir ļoti noderīgi, pirmo reizi palaižot lietojumprogrammu. Dati, kas izmantoti, izveidojot jaunu sistēmas īpašnieku, tiek ņemti no šīs lietojumprogrammas iestatījumiem. Ja nepieciešams, varat tos mainīt.

### 4.1.2 Sistēmas ER modelis

Datu bāzes projektēšanā datu kopu un saišu starp tām attēlošanai tika lietota realitāšusaišu diagramma, kas sastāv no divu veidu objektiem – entītēm (loģiskais objekts) un relācijām (attiecības).

Datu bāze sastāv no šādām tabulām:

* **“AspNetUsers”** - tabula, kurā glabājas informācija par lietotājiem. Tas arī saglabā informāciju no lietotāja profila;
* **“Vendor”** - tabula, kurā tiek saglabāti visi lietotāji ar pārdevēja lomu;
* **“Deliverer”** - tabula, kurā tiek glabāta informācija par lietotājiem ar piegādātāja lomu;
* **“Country”** - tabula ar valstu sarakstu. Mainot valsti lietotāja profilā, šis lietotājs mainīs šai valstij raksturīgo valūtu;
* **“City”** - tabula pievienota valstu tabulai. Tajā ir uzskaitītas visas pilsētas no katras valsts;
* **“Currency”** - tabula ar valūtu sarakstu un to izmaksām. Katras valūtas vērtība tiek mērīta ASV dolāros;
* **“Product”** - tabula, kurā uzskaitīti visi katra pārdevēja produkti;
* **“ProductVariant”** - produktu opciju tabula. Lietotāji pasūta tieši produktu variantu. Katra produkta varianta cenu var saistīt ar jebkuru valūtu;
* **“Order”** - tabula ar visiem pasūtījumiem. Tā kā viens un tas pats pārdevēja produkts var atrasties dažādās valstīs, šī tabula ir saistīta ar produktu krājumiem noliktavās;
* **“OrderSupply”** - starp tabula starp preču krājumiem noliktavās un pasūtījumiem. Nepieciešams gadījumiem, kad pārdevējam vienā pusē ir uzreiz vairākas noliktavas.
* **“DeliveryRoute”** - tabula ar preču piegādes maršrutiem;
* **“Waybills”** - tabula ar rēķiniem par pasūtījumu, piemēram, par piegādes izmaksām vai par komisiju;
* **”Category”** - lai filtrētu katalogā, ir nepieciešama tabula ar kategorijām, kurās var būt produkti;
* **“SubCategory”** - tabula selektīvākai produktu filtrēšanai;
* **“Manufacturer”** - ražotāja informācijas tabula. Izmanto, lai filtrētu produktus katalogā;
* **“Storage”** - tabula, kurā glabājas informācija par visām noliktavām. Pārdevējs var reģistrēt noliktavu, lai tajā norādītu preču pieejamību;
* **“Stock”** - ar noliktavas tabulu saistīta tabula, kas parāda katra produkta pieejamību jebkurā noliktavā;
* **“Dispute”** - strīdu informācijas tabula. Saistīts ar pasūtījumu tabulu un visu tērzēšanas tabulu;
* **“DisputeReason”** - tabula ar informāciju par visiem strīda iemesliem;
* **“Chat”** - tērzēšanas informācijas tabula;
* **“ChatMember”** - tabula ar informāciju par katru tērzēšanas dalībnieku;
* **“Message”** - tabula ar informāciju par katra tērzēšanas dalībnieka ziņojumiem.

Datu bāzes relācijas parada kā savstarpēji ir savienotas divas vai vairākas entītijas:

* starp valsts un pilsētu tabulām ir viens pret daudziem attiecība, jo vienā valstī ir daudz pilsētu, bet viena pilsēta pieder vienai valstij;
* starp pilsētu un lietotāju tabulām ir viens pret daudziem attiecība, jo vienā pilsētā ir daudz lietotāju, bet viens lietotājs pieder vienai pilsētai;
* starp valsts un valūtu ir viens pret daudziem attiecība, jo vienai valūtai ir daudz valsts, bet viena valstī pieder viena valūta;
* starp lietotāju un pārdevēju ir viens pret viens attiecība, jo vienam lietotājam ir viens vai nulle pārdevēju, un vienam pārdevējam ir viens lietotājs;
* starp pārdevējam un produktam ir viens pret daudziem attiecība, jo vienam pārdevējam ir daudz produktu, bet vienam produktam pieder viens pārdevējs;
* starp apakškategorijai un produktam ir viens pret daudziem attiecība, jo vienai apakškategorijai ir daudz produktu, bet vienm produktam pieder viena apakškategorija;
* starp kategorijai un apakškategorijai ir viens pret daudziem attiecība, jo vienai kategorijai ir daudz apakškategoriju, bet vienai apakškategorijai pieder viena kategorija;
* starp ražotājam un produktam ir viens pret daudziem attiecība, jo vienam ražotājam ir daudz produktu, bet vienam produktam pieder viens ražotājs;
* starp pārdevējam un noliktavai ir viens pret daudziem attiecība, jo vienam pārdevējam ir daudz noliktavu, bet vienai noliktavai pieder viens pārdevējs;
* starp produktam un produkta variantam ir viens pret daudziem attiecība, jo vienam produktam ir daudz produkta variantu, bet vienam produkta variantam pieder viens produkts;
* starp noliktavai un piepildījumam ir viens pret daudziem attiecība, jo vienai noliktavai ir daudz piepildījumu, bet piepildījumam pieder viena noliktava;
* starp produkta variantam un piepildījumam ir viens pret daudziem attiecība, jo vienam produkta variantam ir daudz piepildījumu, bet vienam piepildījumam pieder viens produkta variants;
* starp lietotāju un piegādātājam ir viens pret viens attiecība, jo vienam lietotājam ir viens vai nulle piegādātāju, un vienam piegādātājam ir viens lietotājs;
* starp piegādātājam un piegādes maršrutam ir viens pret daudziem attiecība, jo vienam piegādātājam ir daudz piegādes maršrutu, bet vienam piegādes maršrutam pieder viens piegādātājs;
* starp piegādes maršrutam un pasūtījumam ir viens pret daudziem attiecība, jo vienam piegādes maršrutam ir daudz pasūtījumu, bet vienam pasūtījumam pieder viens piegādes maršruts;
* starp piepildījumam un pasūtījumu piepildījumam ir viens pret daudziem attiecība, jo vienam piepildījumam ir daudz pasūtījumu piepildījumu, bet vienam pasūtījumu piepildījumam pieder viens piepildījums;
* starp pasūtījumu piepildījumam un pasūtījumam ir viens pret daudziem attiecība, jo vienam pasūtījumu piepildījumam ir daudz pasūtījumu, bet vienam pasūtījumam pieder viens pasūtījumu piepildījums;
* starp pilsētai un pasūtījumam ir viens pret daudziem attiecība, jo vienai pilsētai ir daudz pasūtījumu, bet vienam pasūtījumam pieder viena pilsēta;
* starp lietotājam un pasūtījumam ir viens pret daudziem attiecība, jo vienam lietotājam ir daudz pasūtījumu, bet vienam pasūtījumam pieder viens lietotājs;
* starp pasūtījumam un strīdai ir viens pret daudziem attiecība, jo vienam pasūtījumam ir daudz strīdu, bet vieni strīdai pieder viens pasūtījums;
* starp strīdas iemeslai un strīdai ir viens pret daudziem attiecība, jo vienam strīdas iemeslam ir daudz strīdu, bet vieni strīdai pieder viens strīdas iemesls;
* starp strīdai un čata dalībniekam ir viens pret daudziem attiecība, jo vienai strīdai ir daudz čata dalībnieku, bet vienam čata dalībniekam pieder viena strīda;
* starp lietotājam un čata dalībniekam ir viens pret daudziem attiecība, jo vienam lietotājam ir daudz čata dalībnieku, bet vienam čata dalībniekam pieder viens lietotājs;
* starp čatam un čata dalībniekam ir viens pret daudziem attiecība, jo vienam čatam ir daudz čata dalībnieku, bet vienam čata dalībniekam pieder viens čats;
* starp čata dalībniekam un vēstulei ir viens pret daudziem attiecība, jo vienam čata dalībniekam ir daudz vēstuļu, bet vienai vēstulei pieder viens čata dalībnieks;

## Datu plūsmju modelis

### Reģistrācijas process

Šajā lietojumprogrammā reģistrācijas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts. Šajā gadījumā veiksmīgas reģistrācijas gadījumā lietotājs tiks novirzīts uz galveno lapu.

Šajā gadījumā pirmā daļa pārbauda datu klātbūtni laukā ar lietotājvārdu, izmantojot e-pasta paroli un paroles apstiprinājumu. Tad, pēc veiksmīgas pārbaudes, tiek pārbaudīti šo vērtību garumi. Pēdējā pārbaude, kas šeit ir pārbaudīta klientā, ir paroles un paroles apstiprinājuma vienlīdzības pārbaude.

Pēc veiksmīgas visu pārbaudes pabeigšanas klientā šie dati tiek nosūtīti uz serveri, izmantojot http pieprasījumu. Serverī datu pārbaude notiek līdzīgi: tiek pārbaudīta datu esamība un garums, un pēc tam tiek pārbaudīts, vai parole atbilst paroles apstiprinājumam. Jebkādu kļūdu gadījumā atbilde tiek atgriezta lietotājam ar statusa kodu un ar ziņojumu un kļūdu sarakstu.

Tad klientam tiek nosūtīta atbilde ar statusa kodu. Klients pārbauda servera atbildes rezultātu un, atkarībā no atbildei pievienotā statusa koda, dodas uz nākamo lapu vai parāda kļūdu.

### Lietotāja ieejas process

Attēls . Reģistrācijas process

Šajā lietojumprogrammā lietotāja ieejas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

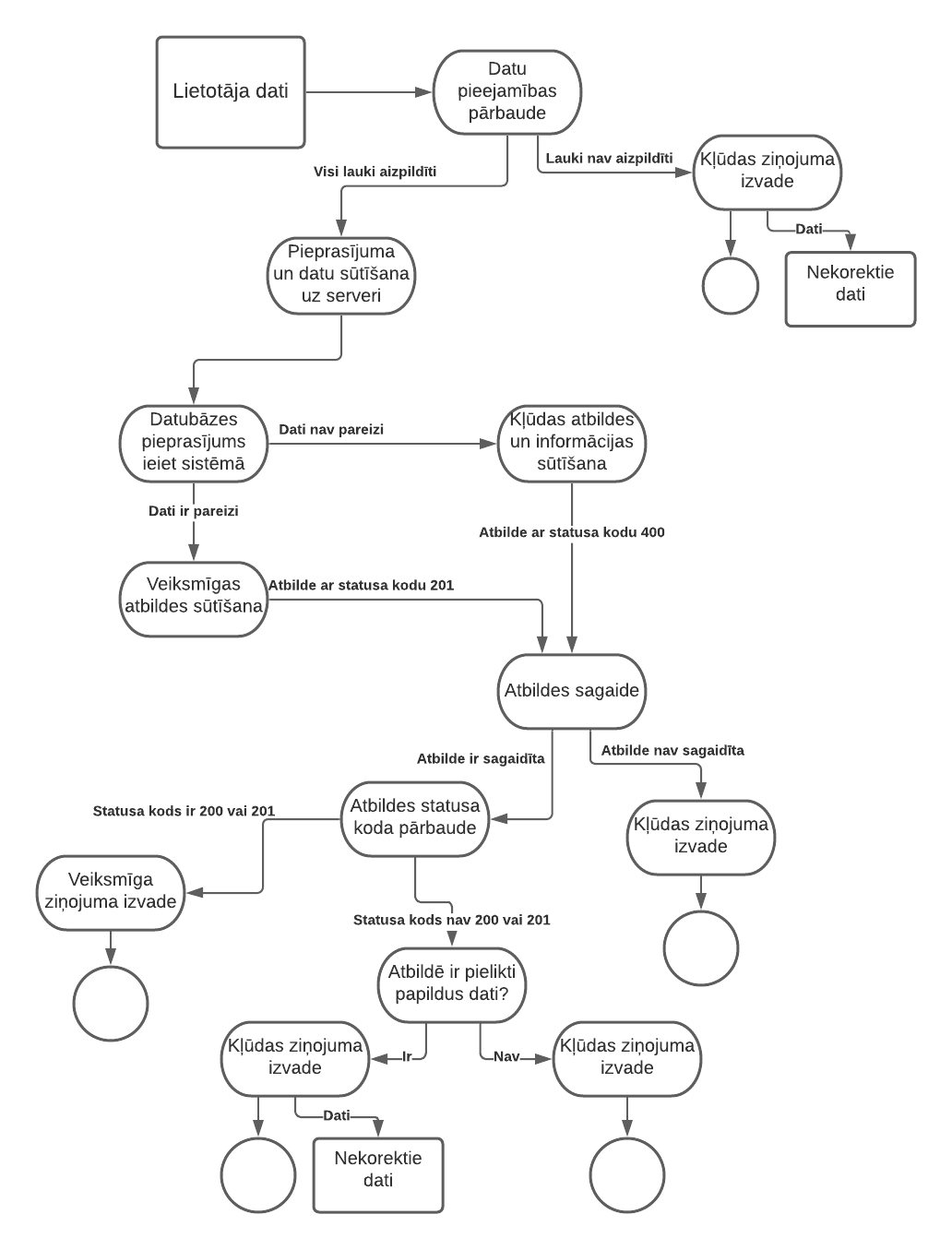
Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts. Šajā gadījumā veiksmīgas reģistrācijas gadījumā lietotājs tiks novirzīts uz galveno lapu.

Šajā gadījumā vispirms tiek parsēti dati no lietotāja, proti, lietotājvārds un parole. Pirmkārt, klientā tiek pārbaudīts, vai šie dati eksistē.

Pēc veiksmīgas visu pārbaudes pabeigšanas klientā šie dati tiek nosūtīti uz serveri, izmantojot http pieprasījumu. Serverī datu pārbaude notiek līdzīgi: tiek pārbaudīta datu esamība. Pēc tam lietotājs tiek meklēts ar lietotājvārdu, kas atbilst ievadītajiem datiem. Ja lietotājs nav atrasts, tas līdzīgi meklēs lietotājus ar atbilstošu e-pastu. Turklāt, ja lietotājs joprojām tiek atrasts, sistēma mēģinās pieteikties, izmantojot lietotāja ievadīto paroli. Pēc šīs funkcijas izpildes atbilde ar rezultātu tiek atgriezta klientam. Jebkādu kļūdu gadījumā atbilde tiek atgriezta lietotājam ar statusa kodu un ar ziņojumu un kļūdu sarakstu.

Tad klientam tiek nosūtīta atbilde ar statusa kodu. Klients pārbauda servera atbildes rezultātu un, atkarībā no atbildei pievienotā statusa koda, dodas uz nākamo lapu vai parāda kļūdu.



Attēls . Lietotāja ieejas process

### Profila rediģēšanas process

Šajā lietojumprogrammā lietotāja ieejas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

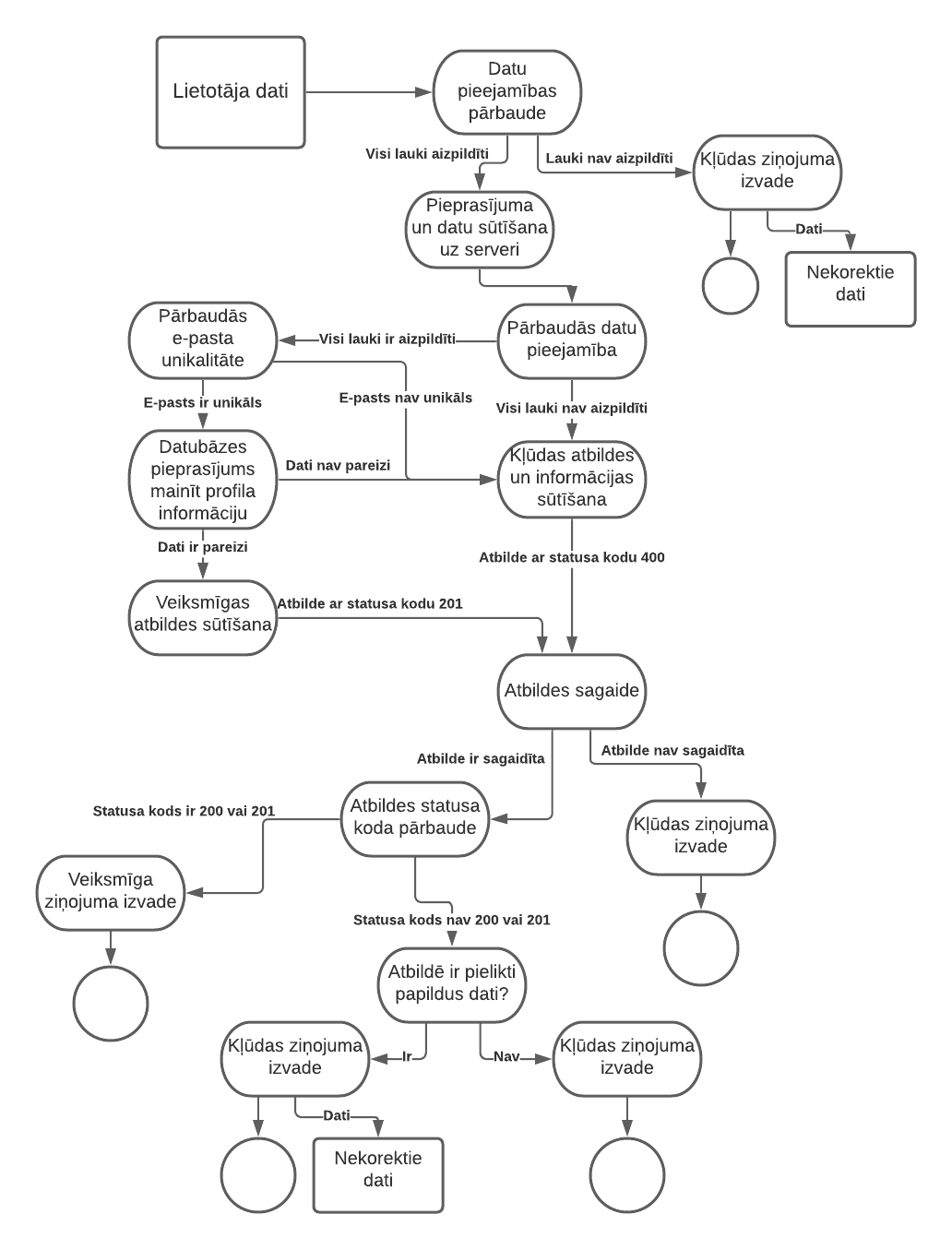
Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.

Šajā gadījumā vispirms visi dati tiek savākti no lietotāja un pēc tam tiek pārbaudīti klientā. Proti, tiek pārbaudīta e-pasta klātbūtne. Ja dati, kas saņemti no veidlapas, tiek pārbaudīti, tad tie tiek pārsūtīti uz serveri, izmantojot http pieprasījumu.

Serverī datu pārbaude notiek līdzīgi: tiek pārbaudīta datu esamība. Tad tiek pārbaudīts ievadītā e-pasta formāts un tā unikalitāte sistēmā. Ja pārbaude ir veiksmīga, dati tiek ierakstīti datu bāzē, un pēc tam lietotājs atgriež atbildi ar statusa kodu. Jebkādu kļūdu gadījumā atbilde tiek atgriezta lietotājam ar statusa kodu un ar ziņojumu un kļūdu sarakstu.

Tad klientam tiek nosūtīta atbilde ar statusa kodu. Klients pārbauda servera atbildes rezultātu un, atkarībā no atbildei pievienotā statusa koda, dodas uz nākamo lapu vai parāda kļūdu.



Attēls . Profila maiņas process

### E-pasta apstiprinājums

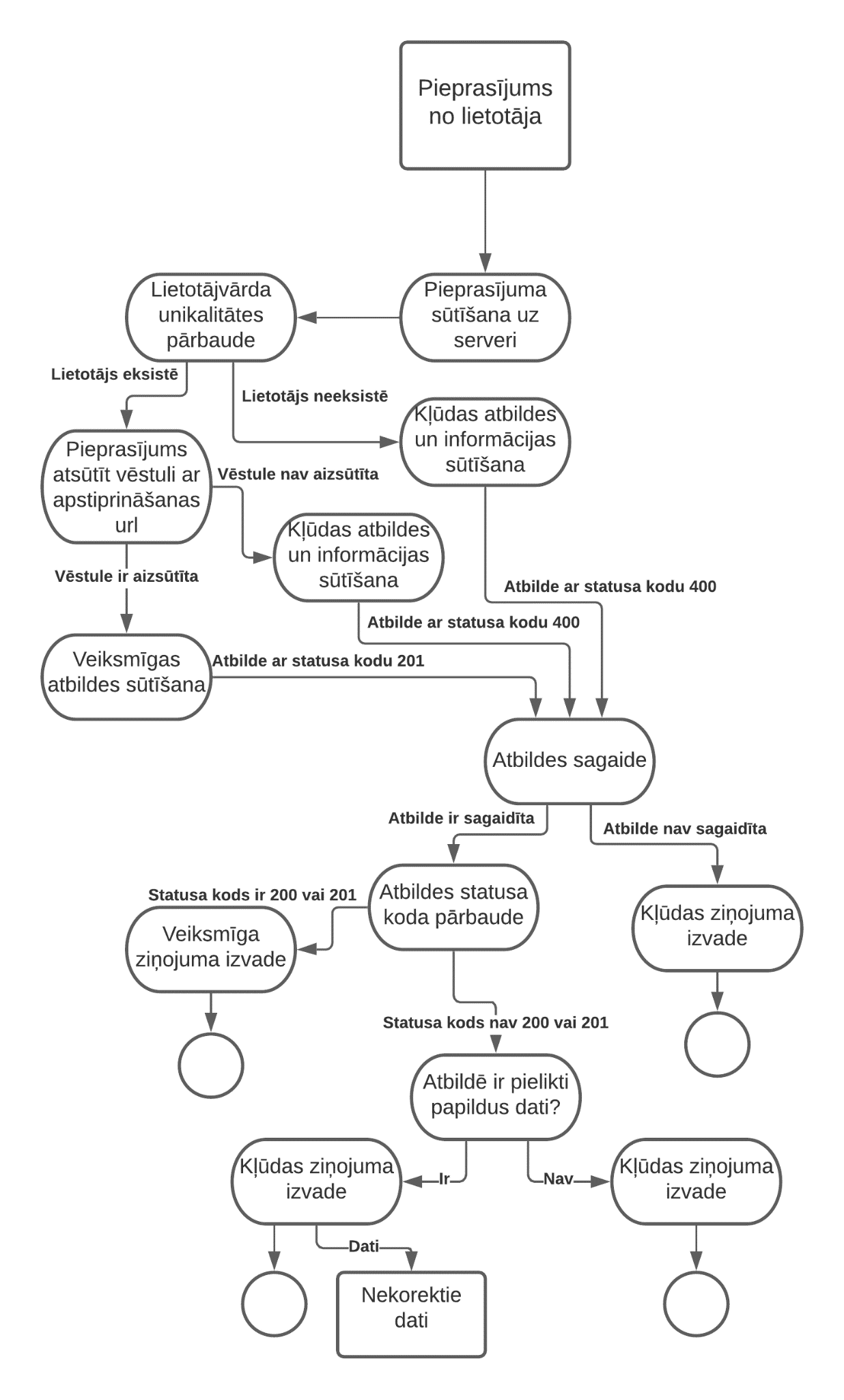
E-pasta apstiprināšanas process nedaudz atšķiras no citiem procesiem, jo ​​tas nepārraida datus, un arī šis process sastāv no diviem apakšprocesiem. Pirmā apakšprocesa būtība ir nosūtīt lietotājam e-pastu ar apstiprinājuma saiti. Otrās tiesas tiesnesis ir pārbaude apstiprinājuma saitei, uz kuru lietotājs ir pārslēdzies. Attiecīgi paša pasta apstiprinājums notiek tieši otrajā procesā.

Abi apakšprocessi apakšprocess ir līdzīgs ar pārēiem processiem.

Pirmajā daļā uz serveri tiek nosūtīts pieprasījums

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.



Attēls 4.4 E-pasta apstiprinājums

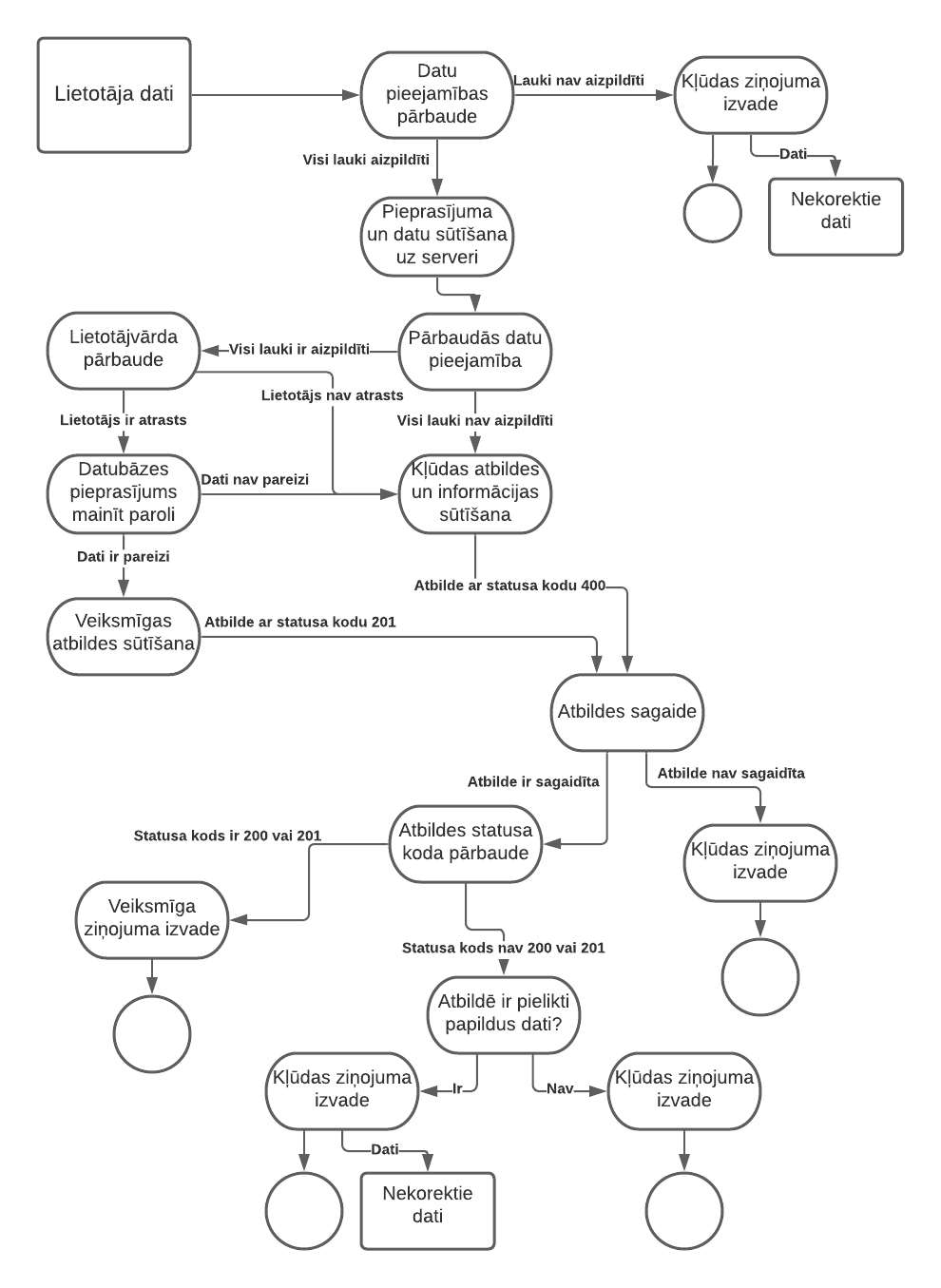
### Paroles nomainīšana

Šajā lietojumprogrammā lietotāja paroles mainīšanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.



Attēls . Paroles maiņas process

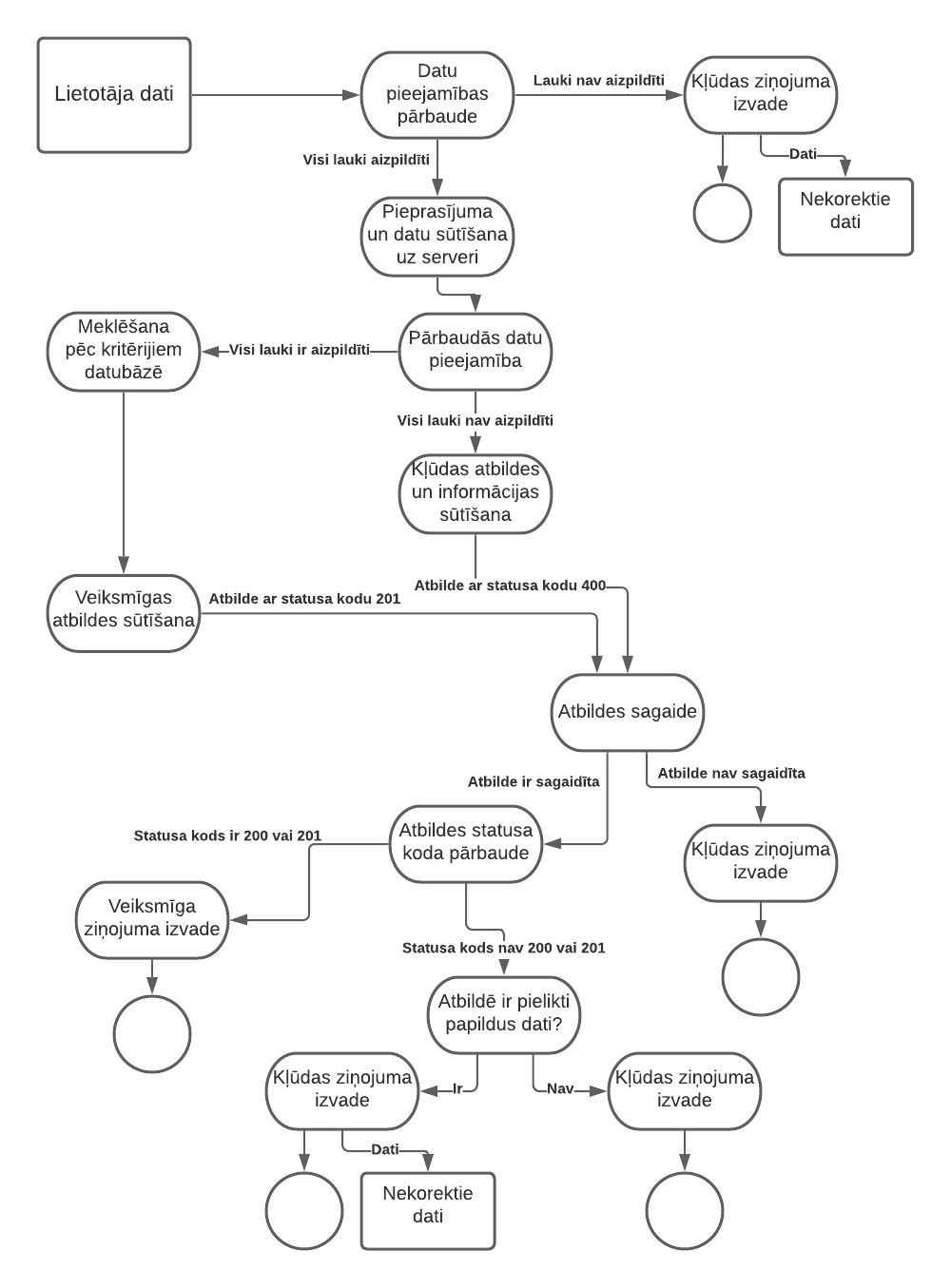
### Preču meklēšana

Šajā lietojumprogrammā preču meklēšanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.



Attēls . Preču meklēšanas process

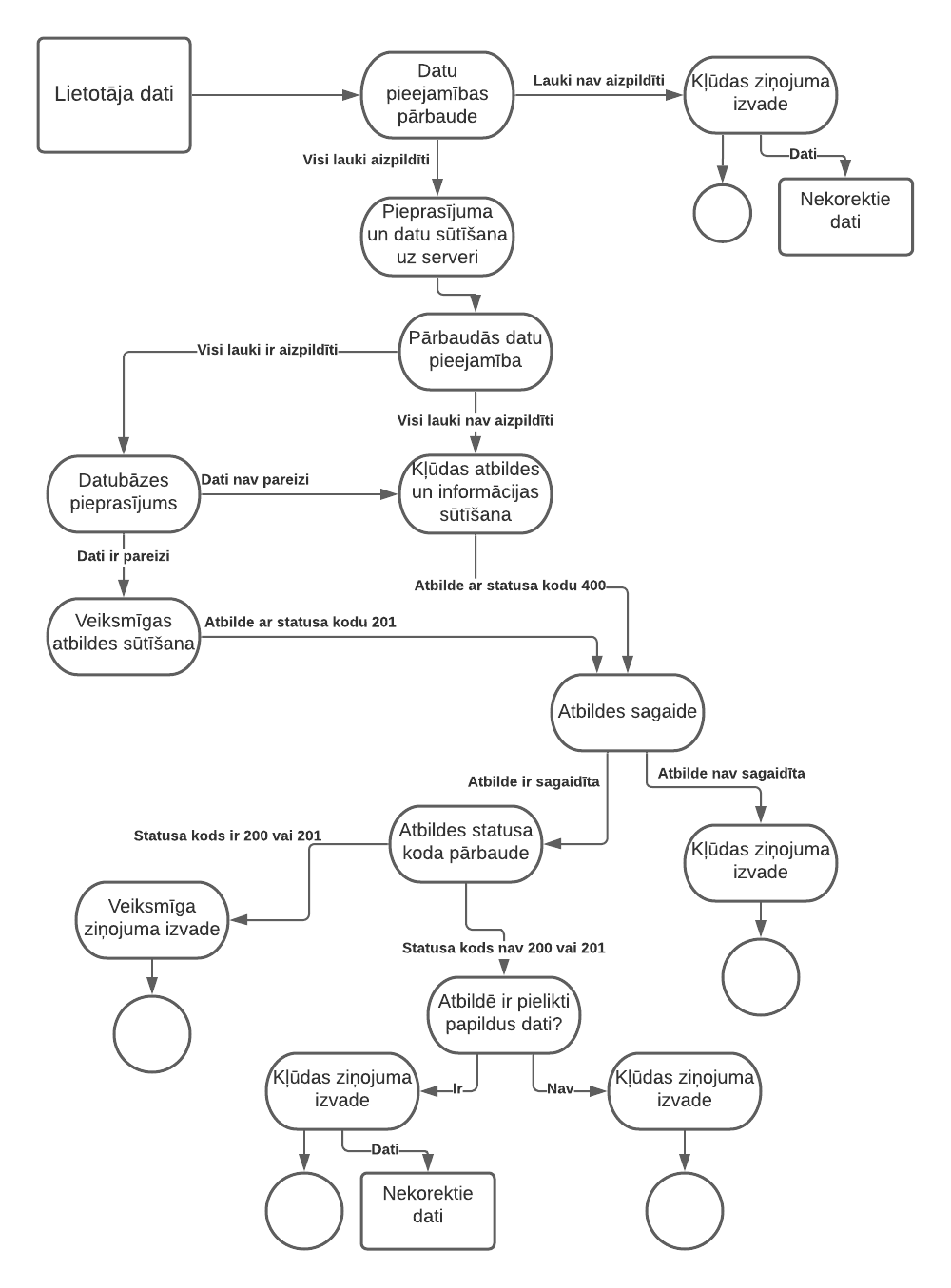
### Preču izveidošana

Šajā lietojumprogrammā preču ievietošanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.



Attēls . Preču izveidošanas process

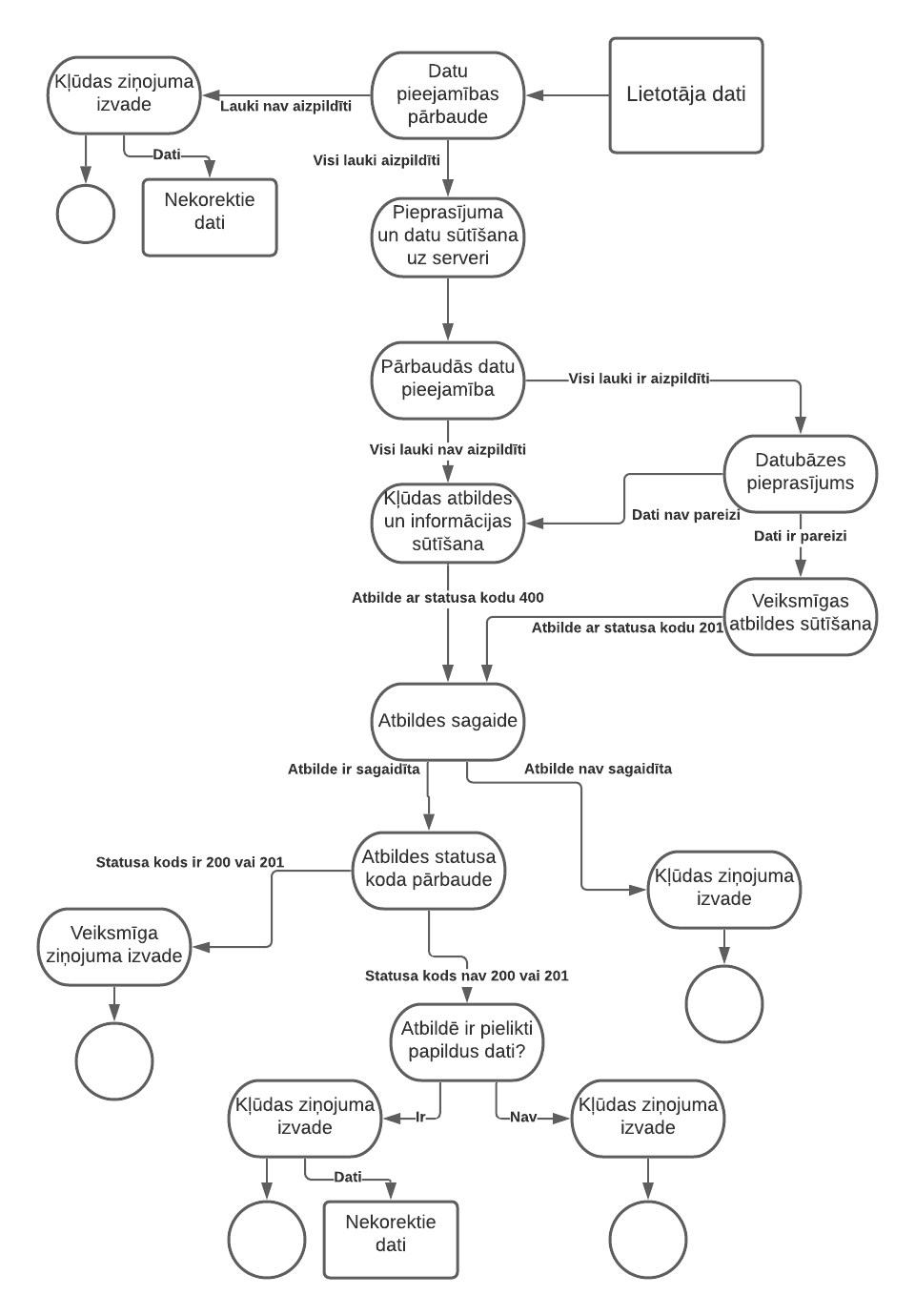
### Noliktavu izveidošana

Šajā lietojumprogrammā noliktavu ievietošanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.



Attēls . Noliktavu izveidošanas process

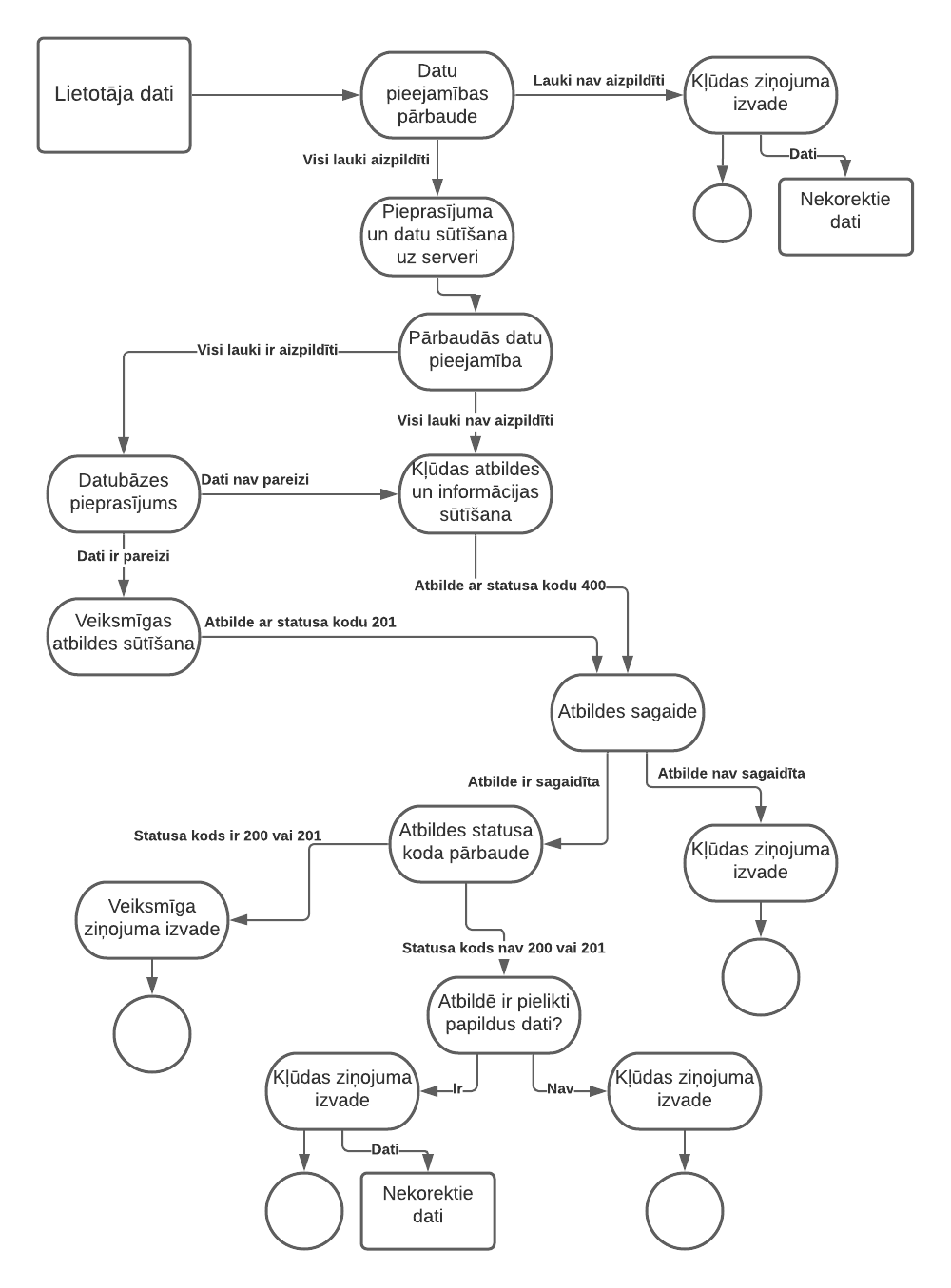
### Produkta pieveduma noliktavā pievienošana

Šajā lietojumprogrammā produkta pieveduma noliktavā pievienošanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.



Attēls 4.9 Produkta pieveduma noliktavā pievienošanas process

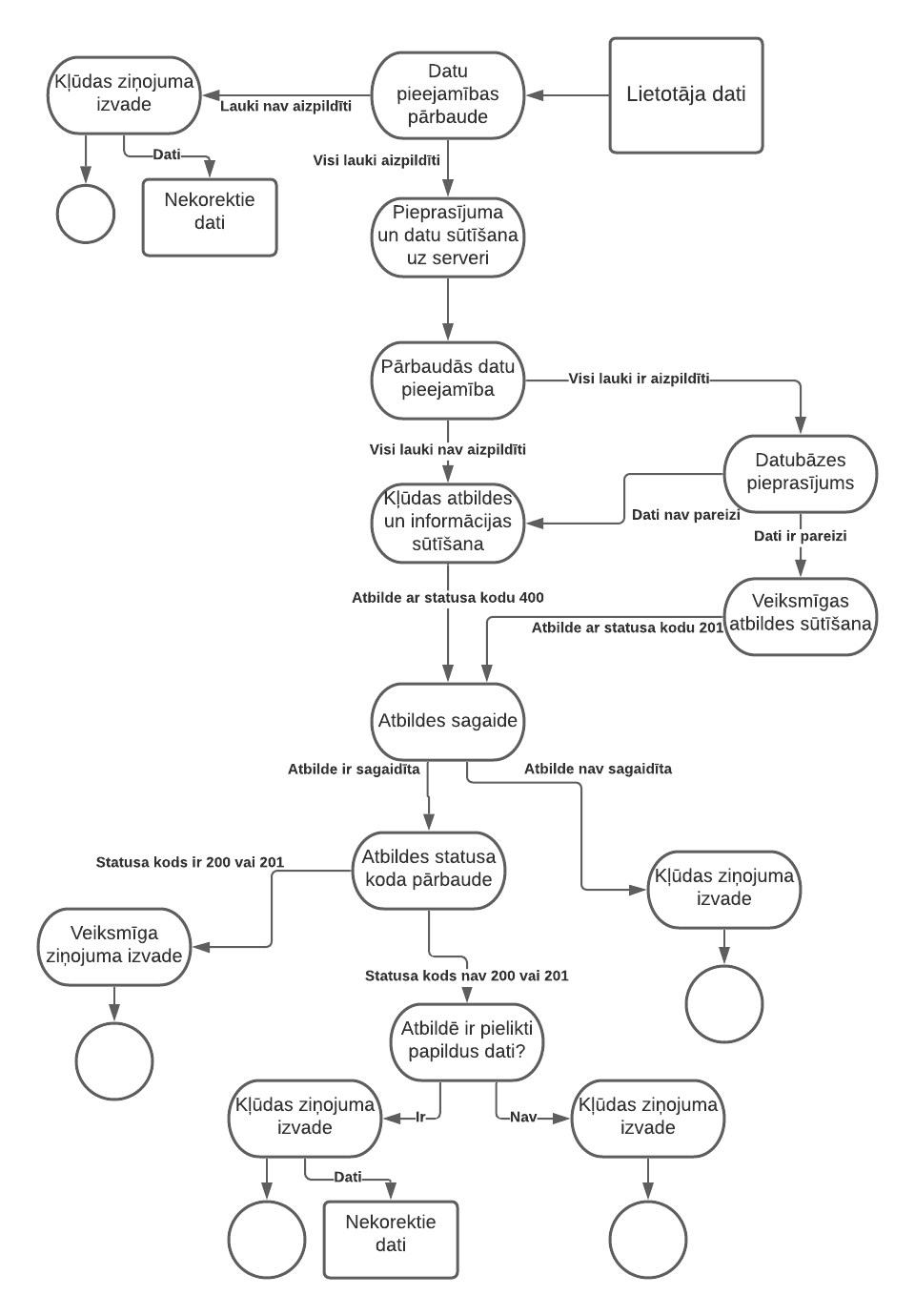
### Piegādes maršruta pievienošana

Šajā lietojumprogrammā piegādes pievienošanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.



Attēls . Piegādes maršruta pievienošanas process

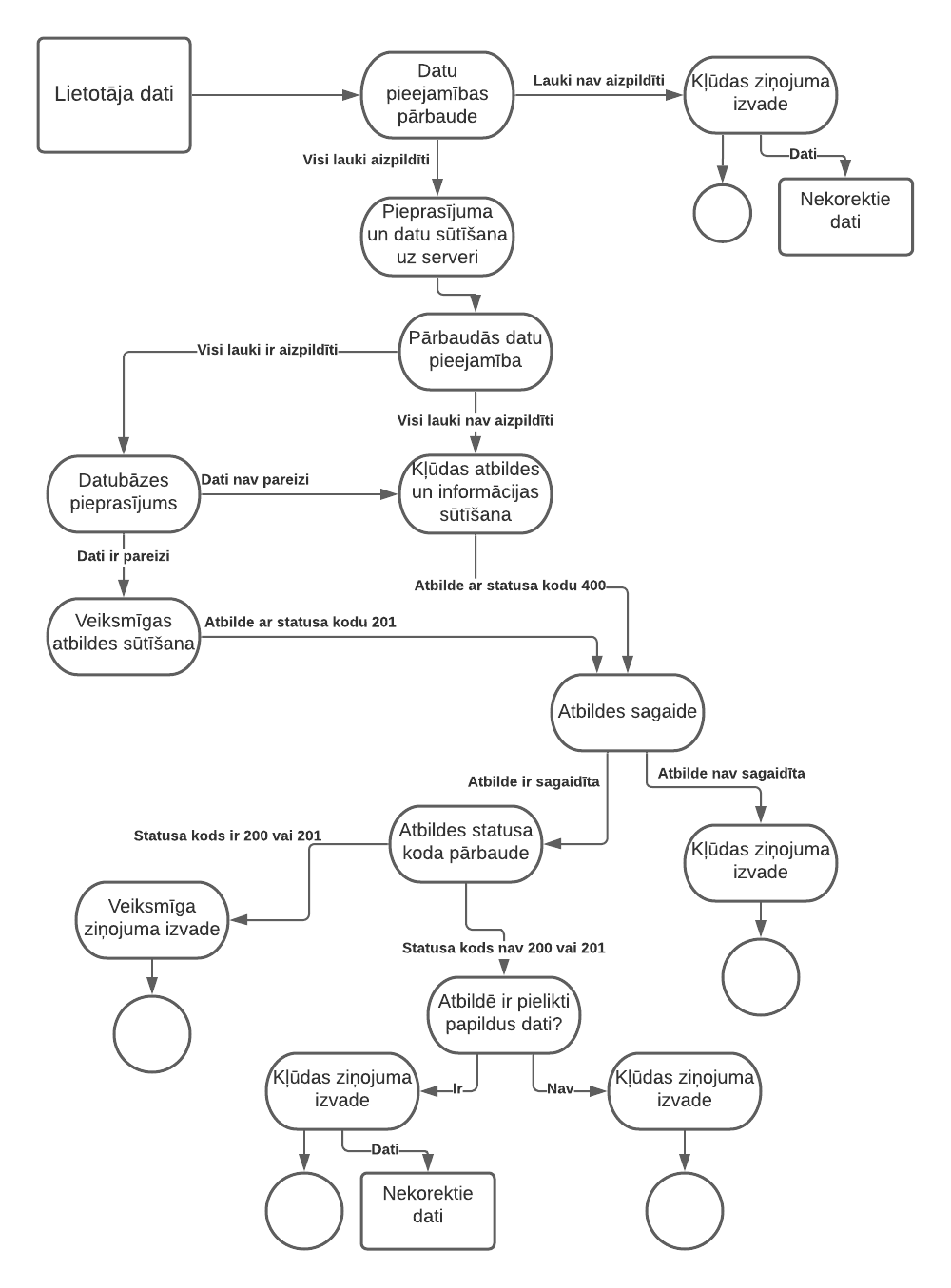
### Sūdzības iesniegšana

Šajā lietojumprogrammā sūdzības iesniegšanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.



Attēls . Sūdzības iesniegšanas process

### Vēstules iesniegšana

Šajā lietojumprogrammā vēstules iesniegšanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.

### Konta frizēšana

Šajā lietojumprogrammā konta frizēšanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.

### Konta bloķēšana

Šajā lietojumprogrammā konta bloķēšanas process, tāpat kā jebkurš cits process, sastāv no 3 daļām.

Pirmā daļa sastāv no datu apkopošanas un provizoriskas validācijas. Pēc datu apkopošanas visi ievadītie dati tiek pārbaudīti klientā, vai tajā nav kļūdu. Tas tiek darīts, lai ietaupītu trafiku un servera resursus. Galu galā, ja dati klientam neiztur verifikāciju, tie netiks nosūtīti uz serveri.

Otro daļu veido servera puses datu pārbaude. Pēc datu saņemšanas no servera tas sāk automātisku datu pārbaudi vai tā saukto modeļa pārbaudi. Ja šīs automātiskās pārbaudes laikā rodas kļūda, serveris atgriež atbildi ar kļūdu sarakstu ar statusu 400. Ja automātiskā datu pārbaude tika veiksmīgi pabeigta, serveris palaiž izstrādātāja rakstīto skriptu. Skripts var saturēt izstrādātāja rakstītas pārbaudes. Ja dati iztur šīs pārbaudes, tie tiek ierakstīti datu bāzē. Vai ir veiksmīgi rakstīts datu bāzē, tad klientam tiek atgriezta atbilde ar statusu 200 vai 201. Atbildē ar statusu 200 vai 201 var būt objekts, ko nosūtījis klients.

Trešo daļu veido klienta atbildes pārbaude no servera. Ja atbildei ir veiksmīgs statusa kods, process parasti beidzas, dodoties uz nākamo lapu. Pretējā gadījumā lietotājs paliks pašreizējā lapā, kurā tiks parādīts kļūdu saraksts.

# Datu struktūras apraksts

Datu bāze sastāv no 31 tabulas. Tajā tiek glabāta informācija par lietotājiem, precēm, pasūtījumiem, pārdevējiem, piegādātājiem, u.t.t. Šī datu bāze tika veidota saskaņā ar normalizācijas noteikumiem, tāpēc tajā nav datu masīvu un datu dublēšanās. Datu bāzes Er-diagramma ir pielikumos. Tā kā šajā datu bāzē ir pat 31 tabula, Er diagramma ir sadalīta vairākās pielikumos. Er-diagramma ir otrajā, trešajā, ceturtajā un piektajā pielikumā.

Tabulā **“AspNetUsers”** tiek glabāta informācija, kas nepieciešama lietotāja piekļuves sistēmā. Tas arī saglabā lietotāja konta paroli. Drošības nolūkos lietotāja paroles lauks ir šifrēts. Atgriežot datus no šīs tabulas, paroles lauks netiek atgriezts. Tabulā tiek saglabāta arī visa lietotāja profilā norādītā informācija. Datu izmaiņu vēsture nav saglabāta tabulā.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula . "AspNetUsers" tabulas struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Nosaukums | Tips | Garums | Piezīmes |
| 1 | Id | nvarchar | 450 | Primārā atslēga |
| 2 | FirstName | nvarchar | MAX | Vārds |
| 3 | LastName | nvarchar | MAX | Uzvārds |
| 4 | CityId | uniqueidentifier | - | Ārēja atslēga (pilsēta) |
| 5 | AddressLine1 | nvarchar | MAX | Adrese |
| 6 | AddressLine2 | nvarchar | MAX | Adrese |
| 7 | PostalIndex | nvarchar | MAX | Pasta indekss |
| 8 | UserName | nvarchar | 256 | Lietotājvārds |
| 9 | NormalizedUserName | nvarchar | 256 | Lietotājvārds |
| 10 | Email | nvarchar | 256 | E-pasts |
| 11 | NormalizedEmail | nvarchar | 256 | E-pasts |
| 12 | EmailConfirmed | bit | - | Vai e-pasts ir apstiprināts |
| 13 | PasswordHash | nvarchar | MAX | Parole |
| 14 | SecurityStamp | nvarchar | MAX | AspNetIdentity lauks |
| 15 | ConcurencyStamp | nvarchar | MAX | AspNetIdentity lauks |
| 16 | PhoneNumber | nvarchar | MAX | Tālruna numurs |
| 17 | PhoneNumberConfirmed | bit | - | Vai tālruna numurs ir apstiprināts |
| 18 | TwoFactorEnabled | bit | - | Vai ir ieslēgta 2 faktoru aizsardz. |
| 19 | LockoutEnd | datetimeoffset | 7 | Bloķēšanas periods |
| 20 | LockoutEnabled | bit | - | Vai ir bloķēts |
| 21 | AccessFailedCount | int | - | Neveiksmu ieeju skaits. |

Tabula ar nosaukumu **“Cart”** glabā visu lietotāju iepirkumu grozus. Šī tabula ir saite starp lietotāju tabulu un tabulu ar priekšmetiem grozos.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula . "Cart" tabulas struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Username | nvarchar | 450 | Ārēja atslēga (lietotājs) |

Tabula ar nosaukumu **“CartedProduct”** tiek izmantota, lai datus par produktiem saglabātu lietotāja grozos. Ieraksti no šīs tabulas nerezervē produktu, tāpēc dati no šīs tabulas nav saistīti ar datiem par preču pieejamību noliktavā. Šajā tabulā ir visi nepieciešamie dati, lai ar vienu klikšķi jūs varētu veikt vairākus pasūtījumus vienlaikus. Pēc pasūtījuma veikšanas katrs ieraksts no šīs tabulas tiks pārvietots uz tabulu pēc pasūtījumiem.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"CartedProduct" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | CartId | uniqueidentifier | - | Ārēja atslēga (grozs) |
| 3 | ProductVariantId | uniqueidentifier | - | Ārēja atslēga (produkta variants) |
| 4 | DeliveryRouteId | uniqueidentifier | - | Ārēja atslēga (piegāde) |
| 5 | amount | int | - | Daudzums |

Tabulā ar **“Category”** nosaukumu tiek glabāta informācija par visām šī veikala kategorijām, kas pieejama produktiem. Lietotāji, meklējot produktu katalogā, filtrā var norādīt kategoriju, lai sašaurinātu meklēšanas loku. Pasūtot preci, lietotājam tiks piedāvāti līdzīgi produkti ar līdzīgām kategorijām. Galvenajā lapā pirmajā rindā tiks parādīti produkti ar kategoriju, kuru lietotāji visbiežāk pasūtīja.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Category" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Name | nvarchar | MAX | Kategorijas nosaukums |
| 3 | IconPath | nvarchar | MAX | Ikona |

Tabulā ar nosaukumu **“Chat”** glabājas tikai tērzēšanas ID. Lai gan tabulā netiek glabāti citi dati kā primārā atslēga, tā ir ārkārtīgi nepieciešama tērzēšanas glabāšanai. Tērzēšanas nosaukums netiek glabāts tabulā ar tīriem, lai katrs lietotājs varētu pārdēvēt tērzēšanu, kā vēlas.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Chat" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |

Tērzēšanas dalībnieku glabāšanai izmanto tabulu ar nosaukumu *“****ChatMember”****.* Dotajā tabulā tiek saglabāts arī *nosaukums*, ko lietotājs ievietojis tērzēšanai*.* Tādējādi to pašu tērzēšanu katram lietotājam var izsaukt atšķirīgi.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula 5.6

**"ChatMember" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Username | nvarchar | 450 | Ārēja atslēga (lietotājs) |
| 3 | ChatId | uniqueidentifier | - | Ārēja atslēga (chat) |
| 4 | ChatName | nvarchar | MAX | Chata nosaukums |

Tabulā ar nosaukumu***“City”*** tiek saglabāti pilsētu nosaukumi*.* Šī tabula ir pievienota tabulai ar valstīm. Tas ļauj izvairīties no lietotāja ievadītas neeksistējošas pilsētas nosaukuma.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula 5.7

**"City" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Name | Nvarchar |  | Pilsētas nosaukums |
| 3 | CountryId | uniqueidentifier | - | Ārēja atslēga (valsts) |

Tabulā ar ***“Country”*** nosaukumu tiek saglabāts valsts nosaukums, saīsinātais valsts nosaukums un šajā valstī izmantotā valuta. Pēc tam, kad lietotājs profilā parādīs savu dzīvesvietas valsti, visām kataloga precēm būs cena tādā valūtā, kāda tagad ir šajā valstī.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula 5.8

**"Country" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Name | nvarchar | MAX | Valsts nosaukums |
| 3 | ShortName | nvarchar | MAX | Valsts apzīmējums |
| 4 | DefaultCurrencyId | uniqueidentifier | - | Ārēja atslēga (valūta) |

Tabulā zem **“Currency”** nosaukuma glabājas valūtas nosaukums, saīsinātais nosaukums, zīme un valūtas cena. Visu datubāzē esošo valūtu vērtība ir saistīta ar ASV dolāra vērtību.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Currency" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Name | nvarchar | MAX | Valūtas nosaukums |
| 3 | ShortName | nvarchar | MAX | Valūtas īss nosaukums |
| 4 | Symbol | nvarchar | MAX | Valūtas apzīmējums |
| 5 | Cost | float |  | Valūtas cena |

Tabulā zem **“Deliverer”** nosaukuma glabājas informācija par piegādātājiem. Šita tabula ir nepieciešama, lai tabulu saitēs bija saites tikai ar eksistējošiem piegādātājiem.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula 5.10

**"Deliverer" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Username | nvarchar | 450 | Ārēja atslēga (lietotājs) |

Tabulā zem **“DeliveryRoute”** nosaukuma glabājas informācija par piegādes maršrutiem starp valstīm. Maršrutā arī ir ievietota informācija par tranportēta produkta maksimāli iespējamiem izmēriem un masu. Arī glabājas cena par transportēšanu.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"DeliveryRoute" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | DelivererUsername | nvarchar | 450 | Ārēja atslēga (piegādātājs) |
| 3 | CountryFromId | uniqueidentifier | - | Ārēja atslēga (valsts) |
| 4 | CountryToId | uniqueidentifier | - | Ārēja atslēga (valsts) |
| 5 | CurrencyId | uniqueidentifier | - | Ārēja atslēga (valūta) |
| 6 | Cost | float | - | Cena |
| 7 | AvgDeliveryTime | int | - | Piegādes laiks |
| 8 | MaxMass | float | - | Maksimālā masa |
| 9 | MaxLength | float | - | Maksimālais garums |
| 10 | MaxSideSum | float | - | Maksimālie izmēri |

Tabulā zem **“Dispute”** nosaukuma glabājas informācija par piegādes strīdām.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Dispute" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | OrderId | uniqueidentifier | - | Ārēja atslēga (pasūtījums) |
| 3 | ChatId | uniqueidentifier | - | Ārēja atslēga (čats) |
| 4 | DisputeReasonId | uniqueidentifier | - | Ārēja atslēga (iemesls) |
| 5 | Details | nvarchar | MAX | Strīdas detaļas |

Tabulā zem **“DisputeReason”** nosaukuma glabājas informācija par strides iemesliem.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"DisputeReason" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Name | nvarchar | MAX | Iemesla nosaukums |

Tabulā zem **“Manufacturer”** nosaukuma glabājas informācija par ražotājiem.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Manufacturer tabulas struktūra"**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Name | nvarchar | MAX | Nosaukums |
| 3 | ManufacturerUrl | nvarchar | MAX | Mājaslapa |
| 4 | IconPath | nvarchar | MAX | Ikona |

Tabulā zem **“Message”** nosaukuma glabājas informācija par vēstulēm.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Message" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Text | nvarchar | MAX | Teksts |
| 3 | DateTime | datetime | - | Ievietošanas datums |
| 4 | ChatMemberId | uniqueidentifier | - | Ārēja atslēga (čata dalībnieks) |

Tabulā zem **“Order”** nosaukuma glabājas informācija par pasūtījumiem.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Order" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | CityId | uniqueidentifier | - | Ārēja atslēga (pilsēta) |
| 3 | AddressLine1 | nvarchar | MAX | Piegādes adrese |
| 4 | AddressLine2 | uniqueidentifier | - | Piegādes adrese |
| 5 | DeliveryRouteId | uniqueidentifier | - | Ārēja atslēga (piegāde) |
| 6 | CurrencyId | uniqueidentifier | - | Ārēja atslēga (valūta) |

Tabulā zem **“OrderSupply”** nosaukuma glabājas informācija par piegādes maršrutiem starp valstīm.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"OrderSupply" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | StockId | uniqueidentifier | - | Ārēja atslēga (krājumi) |
| 3 | OrderId | uniqueidentifier | - | Ārēja atslēga (pasūtījums) |
| 4 | Amount | int | - | Daudzums |

Tabulā zem **“Product”** nosaukuma glabājas informācija par precēm.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Product" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | SubCategoryId | uniqueidentifier | - | Ārēja atslēga (sub kategorija) |
| 3 | ManufacturerId | uniqueidentifier | - | Ārēja atslēga (ražotājs) |
| 4 | Name | nvarchar | - | Nosaukums |
| 5 | VendorUserName | nvarchar | 450 | Ārēja atslēga (pārdevējs) |
| 6 | Tags | nvarchar | - | Piezīmes |

Tabulā zem **“ProductVariant”** nosaukuma glabājas informācija par preces variantiem.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"ProductVariant" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | ProductId | uniqueidentifier | - | Ārēja atslēga (produkts) |
| 3 | CurrencyId | uniqueidentifier | - | Ārēja atslēga (valūta) |
| 4 | Cost | float | - | Cena |
| 5 | Name | nvarchar | MAX | Nosaukums |

Tabulā zem **“Stock”** nosaukuma glabājas informācija par produktu piegādes noliktavos.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Stock" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | StorageId | uniqueidentifier | - | Ārēja atslēga (noliktava) |
| 3 | ProductVariantId | uniqueidentifier | - | Ārēja atslēga (produkta variants) |
| 4 | Quantity | int | - | Daudzums |

Tabulā zem **“Storage”** nosaukuma glabājas informācija par noliktavām, kur glabājas produkti.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Storage" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | VendorUsername | nvarchar | 450 | Ārēja atslēga (pārdevējs) |
| 3 | Name | nvarchar | MAX | Nosaukums |
| 4 | CityId | uniqueidentifier | - | Ārēja atslēga (pilsēta) |

Tabulā zem **“SubCategory”** nosaukuma glabājas informācija par apakškategorijām.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"SubCategory" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | CategoryId | uniqueidentifier | - | Ārēja atslēga (kategorija) |
| 3 | Name | nvarchar | MAX | Nosaukums |
| 4 | IconPath | nvarchar | MAX | Ikona |

Tabulā zem **“Supply”** nosaukuma glabājas informācija par produktu piegādes noliktavos.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Supply" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | ProductVariantId | uniqueidentifier | - | Ārēja atslēga (produkta variants) |
| 3 | StorageId | uniqueidentifier | - | Ārēja atslēga (noliktava) |
| 4 | Amount | int | - | Daudzums |
| 5 | DateTime | datetime | - | Izveidošanas datums |

Tabulā zem **“Vendor”** nosaukuma glabājas informācija par pārdevējiem.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Vendor" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Username | nvarchar | 450 | Primārā atslēga |
| 2 | CompanyName | nvarchar | MAX | Kompānijas nosaukums |

Tabulā zem **“Waybill”** nosaukuma glabājas informācija par pavadzīmēm.

Šajā tabulā ir galvenā atslēga Guid formātā. Tas nozīmē, ka šajā laukā tiek saglabāta unikāla rakstzīmju secību vērtība. Tas nodrošina, ka dati šajā laukā nekad netiks atkārtoti un ka lietotājs nesaņems kļūdas ne unikālas primārās atslēgas vērtības dēļ.

Tabula .

**"Waybill" tabulas struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Garums** | **Piezīmes** |
| 1 | Id | uniqueidentifier | - | Primārā atslēga |
| 2 | Name | nvarchar | MAX | Nosaukums |
| 3 | OrderId | uniqueidentifier | - | Ārēja atslēga (noliktava) |
| 4 | currencyId | uniqueidentifier | - | Ārēja atslēga (noliktava) |
| 5 | Cost | float | - | Cena |

# Lietotāja ceļvedis

Tas ir ceļvedis manas lietotnes izmantošanai.

Katram jaunajam lietotājam, kurš izmanto šo lietojumprogrammu, ir ierobežotas iespējas. Lietotājs, kurš nav pieteicies sistēmā, nekādā veidā nevar mijiedarboties ar datu bāzē esošajiem datiem, bet var tos tikai apskatīt. Proti, viņš katalogā var apskatīt produktus un to variantus. Arī nereģistrēts vai nepierakstījies lietotājs, meklējot produktus, var izmantot kataloga meklēšanas filtrus.

Lai reģistrētos sistēmā, augšējā labajā stūrī noklikšķiniet uz reģistrācijas pogas. Pēc tam lietotājs tiek novirzīts uz lapu ar reģistrācijas veidlapu, kas viņam jāaizpilda. Ir jāaizpilda visi lauki reģistrācijas veidlapā. pēc veiksmīgas reģistrācijas lietotājs tiks vilkts uz galveno lapu.

Lai pieteiktos, jums jānoklikšķina uz pieteikšanās pogas augšējā labajā stūrī. Jūs tiksiet novirzīts uz lapu ar pieteikšanās veidlapu. Tam ir divi lauki un abi lauki, kas jāaizpilda, lai iekļūtu sistēmā. Augšējā laukā varat ievadīt ne tikai savu lietotājvārdu, bet arī e-pastu. Pēc veiksmīgas pieteikšanās jūs tiksiet novirzīts uz galveno lapu.

Lai mainītu profila datus, jums jāiet uz lapu ar savu profilu. Šai lapai var piekļūt tikai pieteicies lietotājs, tāpēc jums jāpārliecinās, vai esat pieteicies. Pēc tam, kad esat pārliecinājies, ka esat pieteicies, jums jānoklikšķina uz pogas Profils augšējā labajā stūrī. Jūs tiksiet novirzīts uz lapu ar veidlapu ar personisko informāciju, lauki, kuros tiks aizpildītas vērtības, kas tiek saglabātas datu bāzē par jums. Lai mainītu datus, vienkārši mainiet vērtību laukos. Visas izmaiņas laukos tiks automātiski nosūtītas serverim, izmantojot ajax pieprasījumus. Lai atgrieztos, noklikšķiniet uz projekta nosaukuma ekrāna augšējā kreisajā stūrī vai noklikšķiniet uz pogas Atpakaļ apakšā zem veidlapas.

Ja esat šīs lietojumprogrammas īpašnieks un man ir piekļuve kontam, kas atrodas īpašnieka lomā, jums ir piekļuve datu bāzei. Ja esat pieteicies sistēmas mu-mu, izmantojot kontu, kas atrodas tālvadības pults īpašnieka lomā, augšējā labajā stūrī redzēsit pogu datu bāze. Noklikšķinot uz šīs pogas, jūs tiksiet novirzīts uz lapu ar tabulu sarakstu no datu bāzes. Katrs šī saraksta elements Šī ir saite, kas ved uz lapu ar tabulu un šajā tabulā aizpildītiem datiem. No turienes jūs varat pievienot, rediģēt un dzēst datus no datu bāzes.

Ja esat pārdevējs, tad augšējā labajā stūrī, noklikšķinot uz tā, tiks parādīta pārdevēja poga, uz kuras jūs tiksiet novirzīts uz lapu preču, noliktavu un piederumu pārvaldībai. No turienes jūs varat izveidot modificēt un dzēst datus.

Nobeigums

Kvalifikācijas darbā noteiktais mērķis nav sasniegts.

Mērķis netika sasniegts, jo es sev uzstādīju pārāk augstu latiņu. Protams, es varētu izdomāt vietni ar piecām tabulām datu bāzē un pāris dienu laikā to izdarīt kādā PHP, bet es neesmu tāds cilvēks, kurš meklētu vienkāršus veidus. Tā vietā es izvēlējos izstrādāt sarežģītu tīmekļa lietojumprogrammu vidē, par kuru pat nekad nebiju dzirdējis. Iepriekš minēto faktoru dēļ, tā kā nebija pietiekami daudz laika un pūļu pastāvīgu lietu dēļ, kas mani novērsa no darba, es nevarēju panākt šo projektu pienācīgā līmenī.

Attiecībā uz tirdzniecības vērtību šim projektam nav nekāda labuma. Pat pilnīgi gatavā formā šis veikals nedod pilnīgi neko jaunu, bet tikai kopē konkurentu rīcībā esošo. Otrkārt, tirgū izveidoto konkurentu skaits ir milzīgs, un šī projekta liktenis ir nogrimt starp visiem šiem konkurentiem.

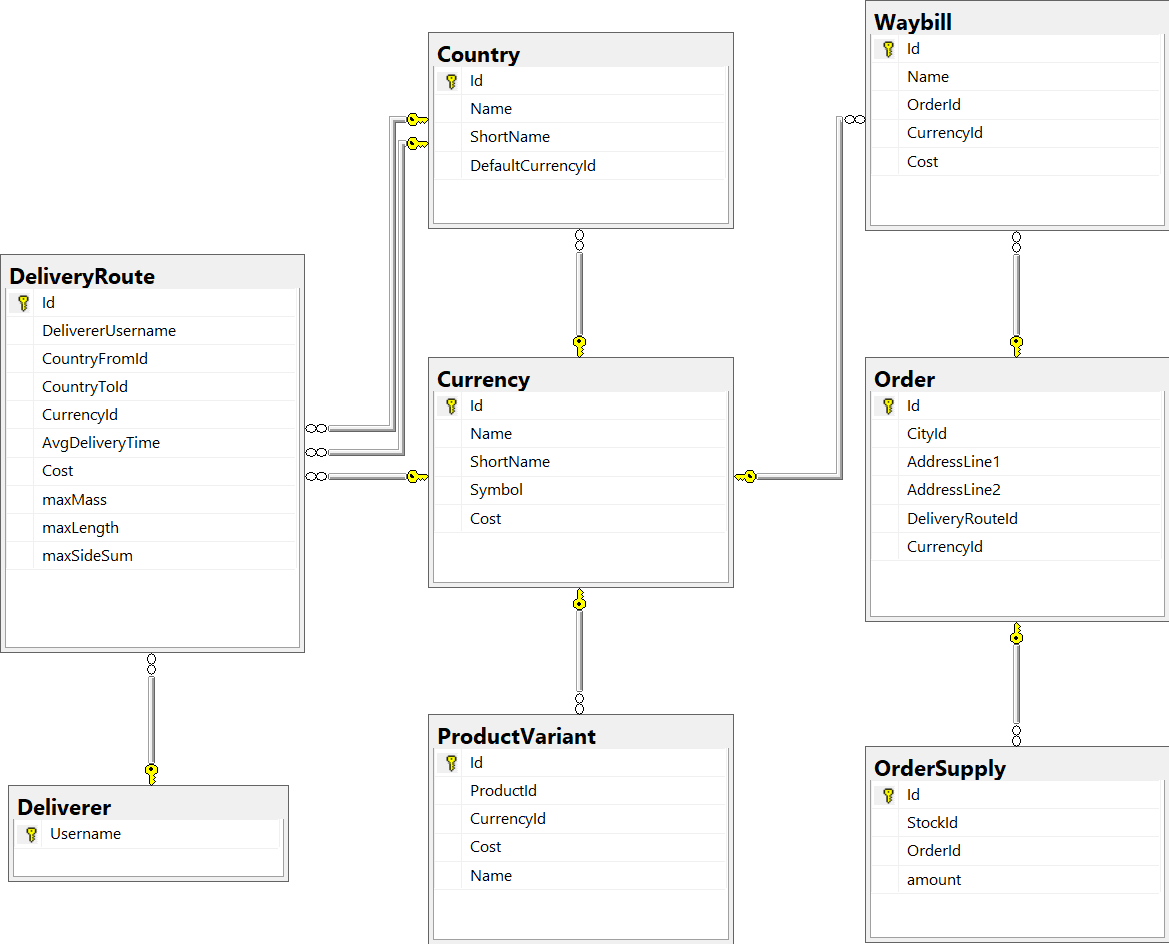
Kāpēc es projektam izvēlējos tieši šo tēmu? Interneta veikals nav tik vienkārši izdarāms. Projekts ar šo tēmu ir milzīgs lauks, lai eksperimentētu ar spēli ar kodu. Milzīgā darba apjoma dēļ jūs sākat domāt, ko un kur var sagriezt un kā saspiest kodu, nepasliktinot tā kvalitāti. Es uzskatu, ka šie apstākļi ir vispiemērotākie, lai apgūtu programmēšanu pilnīgi jaunā, iepriekš nepazīstamā platformā.

Informācijas avoti

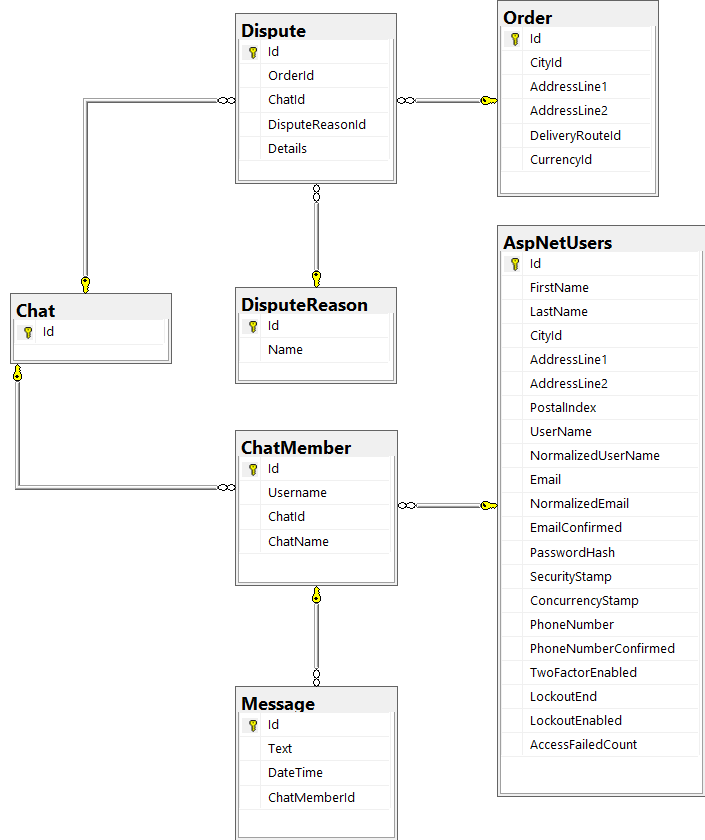
* <https://metanit.com/sharp/aspnet5/>
* <https://www.c-sharpcorner.com/article/upload-and-display-image-in-asp-net-core-3-1/>
* <https://docs.microsoft.com/en-us/aspnet/core/mvc/views/view-components?view=aspnetcore-5.0>
* <https://stackoverflow.com/questions/36529424/ajax-with-viewcomponent>
* <https://www.pluralsight.com/product/skills?utm_term=&aid=7010a000002LZ5aAAG&promo=&utm_source=branded&utm_medium=digital_paid_search_google&utm_campaign=XYZ_EMEA_Brand_E&utm_content=&cq_cmp=1620628967&gclid=Cj0KCQjwzYGGBhCTARIsAHdMTQxZ5nReYqSKEErX06Tzt_hoJdkp3X88Qd139057gsWuqC8O99dmE3waAi6tEALw_wcB>

# Pielikumi

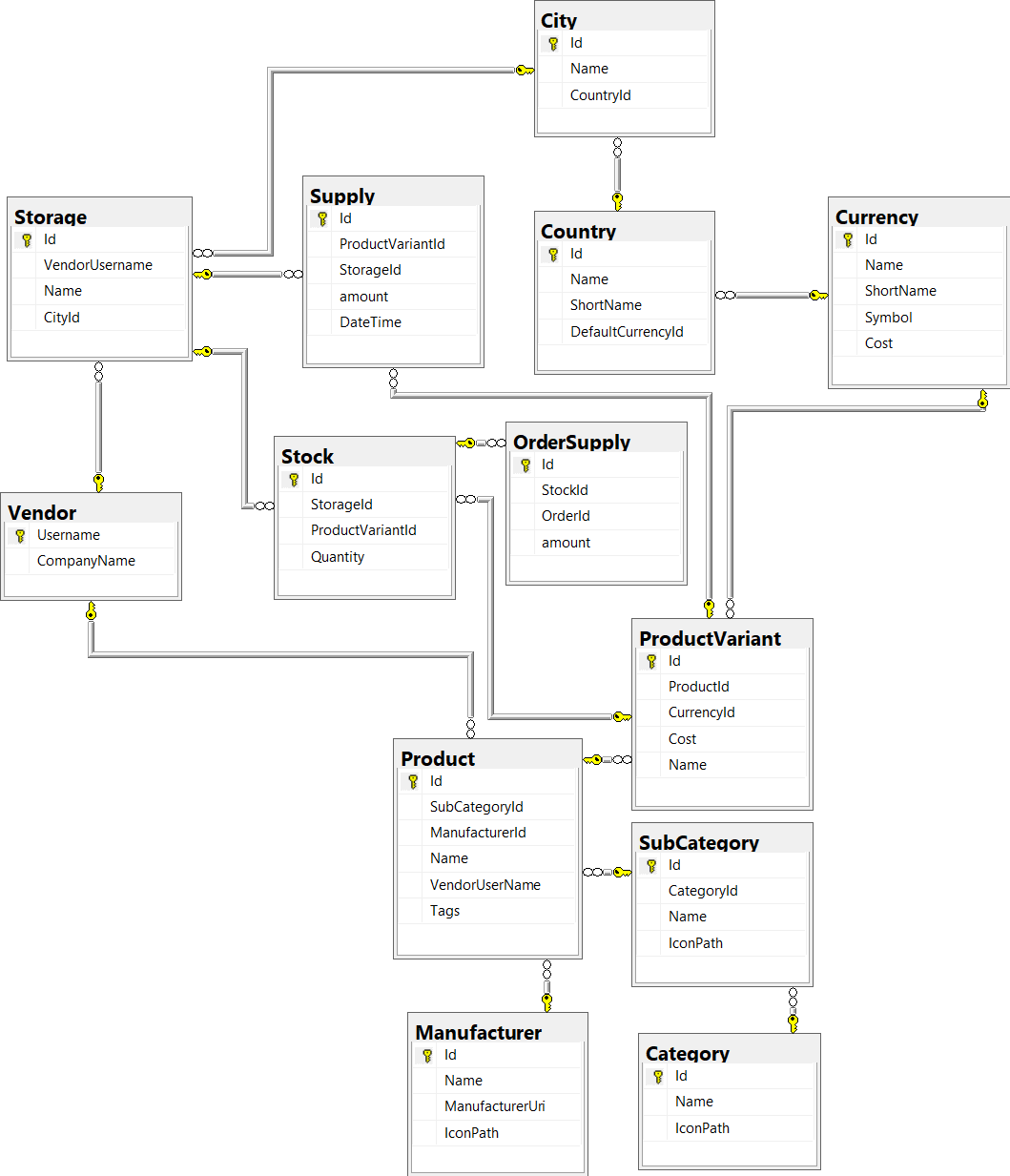
Pielikums 1

**ERD daļa ar pasūtījumiem**

Pielikums 2

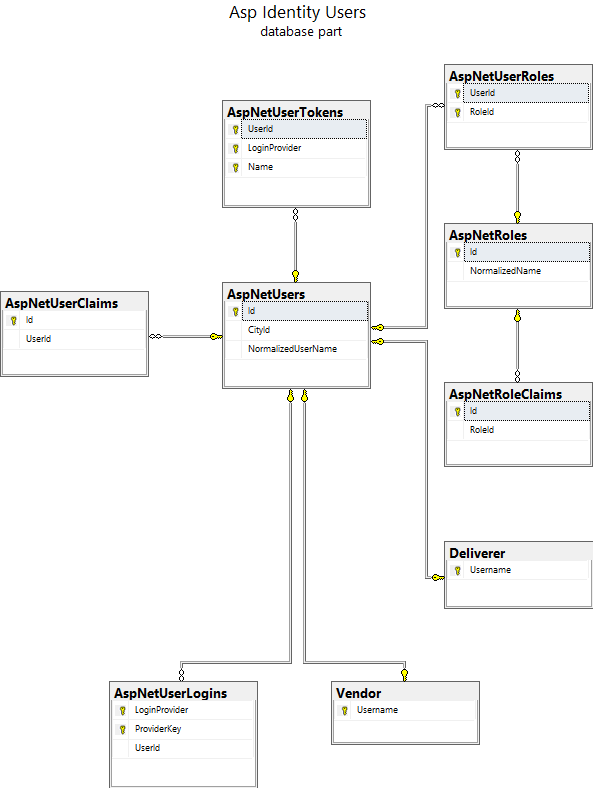
**ERD strīdu daļa**

Pielikums 3

**ERD pasūtījumu daļa**

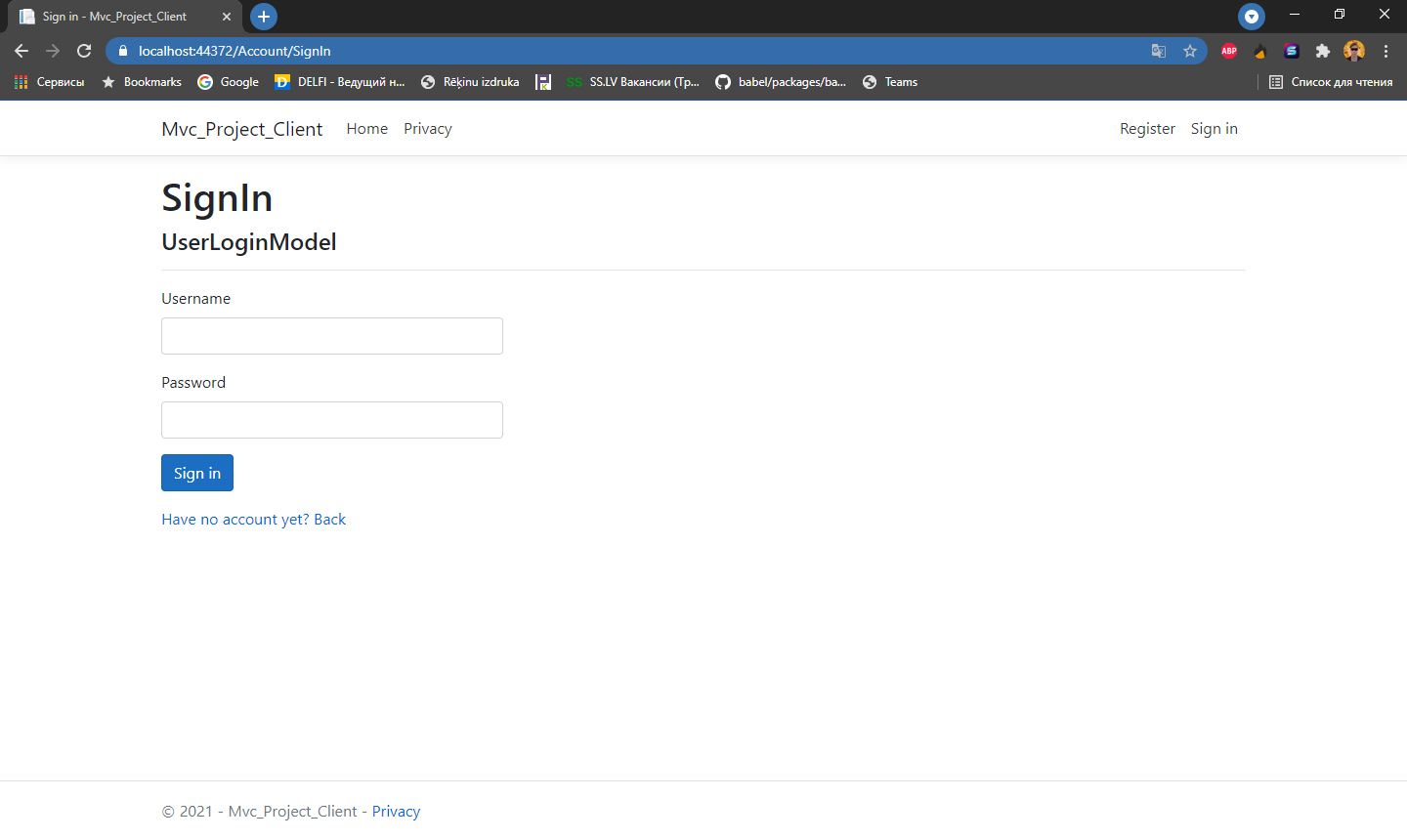
Pielikums 4

**ERD daļa ar lietotājiem**

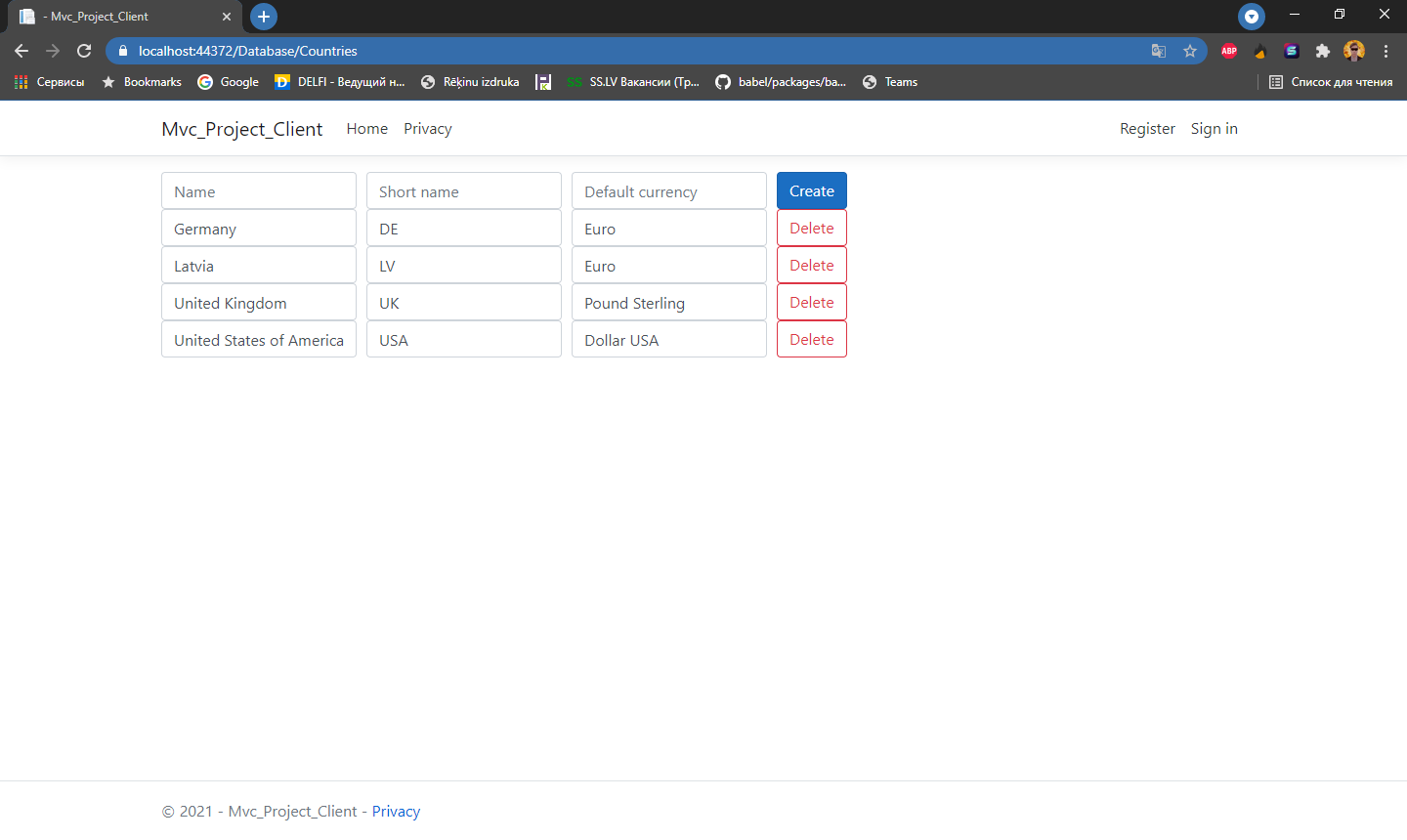


Pielikums 5

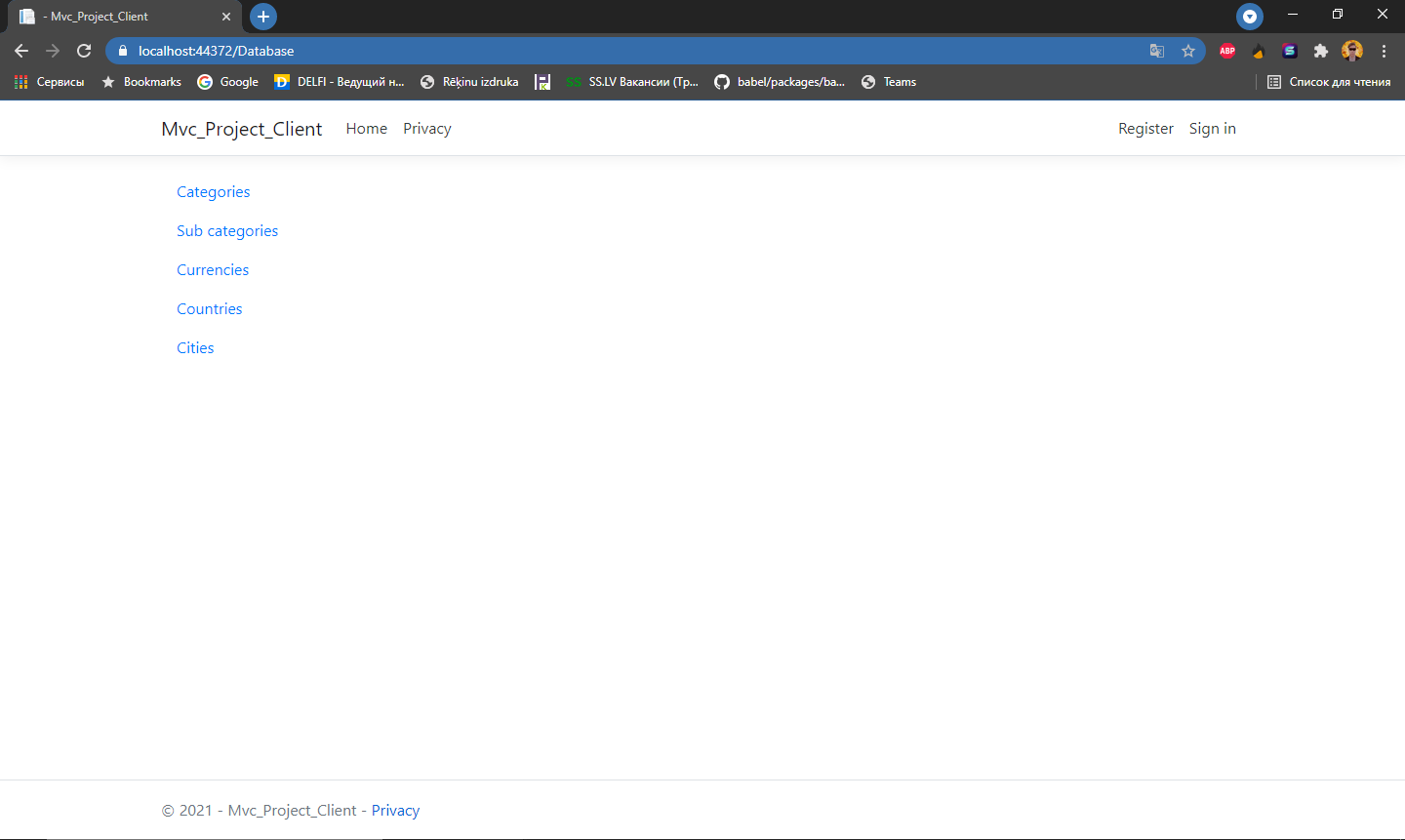
**Ieeja**



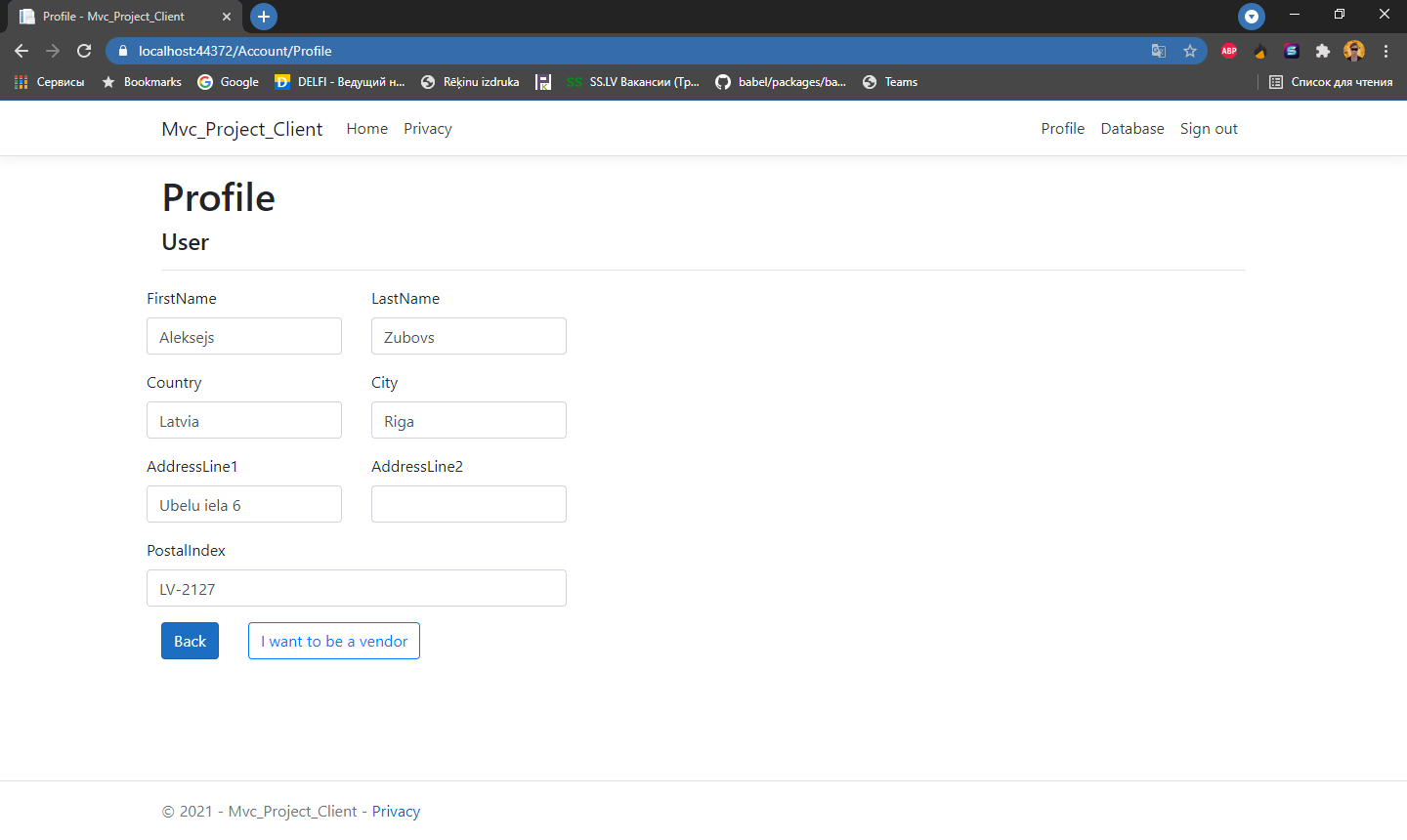
Pielikums 6

**Datubāze**

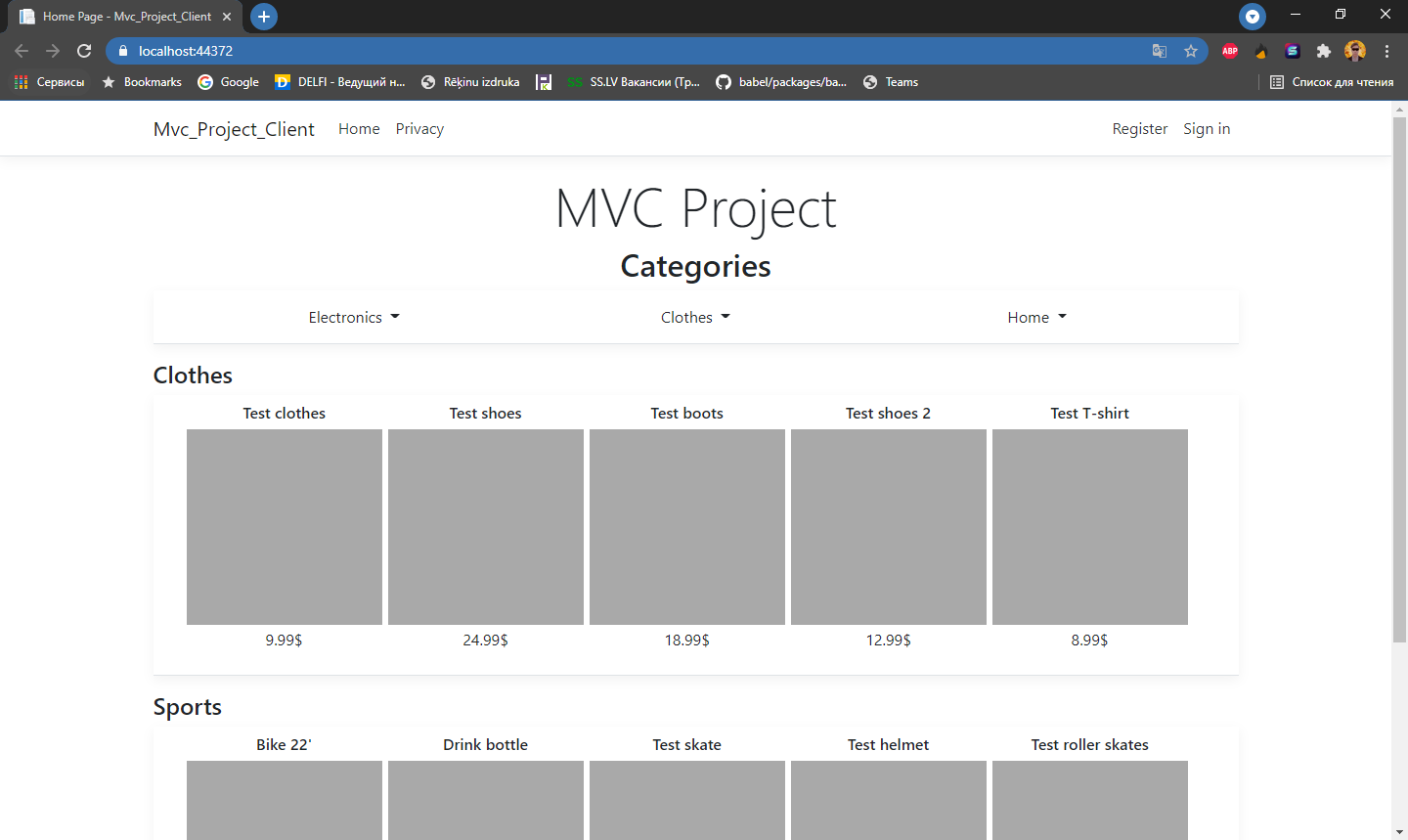
Pielikums 7

**Datubāze**

Pielikums 8

**Profils**

Pielikums 9

**Galvenā lapa**

Pielikums 10

**Reģistrācija**