

VILNIAUS UNIVERSITATAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

**FoodKept**  
**PIRMASIS LABORATORINIS DARBAS**  
**Programų sistemų inžinerija II**

Darbą atliko: Programų sistemų II kurso studentai  
Wasted komanda

Vilnius,  
2022

## **Santrauka**

Semestro ilgio kursui mūsų komandai buvo duota užduotis perimti kolegų domeno (arba aplikacijos) kūrimą ir tobulinimą. Pagrindinė užduoties dalis - išmokti ir suprasti programų kūrimo principus, juos realizuoti puslapio kūrime, bei modeliuoti programų sistemų kūrimo procesus.

Šiam laboratoriniui darbui mums buvo duotos užduotys:

1. Nustatyti, išanalizuoti ir dokumentuoti vartotojų reikmes.
2. Nustatyti verslo ir sistemos ribas.
3. Aprašyti koncepcinio ir procesų modelius (Modeliams naudoti BPMN ir UML modeliavimo kalbas).
4. Nustatyti, argumentuoti ir validuoti reikalavimus.

Numatomas rezultatas yra dokumentas aprašantis verslo modelį, vartotojų reikmes ir domeno modelį, kartu su reikalavimais.

# Turinys

1.Kontekstas.....	5
1.1. Žodynas .....	5
1.2. Valdymo principai .....	5
1.2.1. Misija.....	5
1.2.2. Vizija .....	5
1.2.3. Vertybės.....	5
1.3. Išorinė analizė.....	5
1.3.1. Verslo modelis.....	6
1.3.2. Klientai .....	6
1.3.3. Verslo ribos .....	7
1.3.4. Pajamų šaltinis.....	7
1.3.5. Išlaidos.....	7
1.4. SSGG Analizė.....	8
1.4.1. Stiprybės .....	8
1.4.2. Silpnybės .....	8
1.4.3. Galimybės.....	8
1.4.4. Grėsmės .....	8
1.4.5. SSGG rezultatai .....	9
1.5. Porterio jėgos .....	9
1.5.1. Konkurencija šakoje .....	9
1.5.2. Pakaitų gresmė.....	9
1.5.3. Vartotojai.....	10
1.5.4. Tiekėjai .....	10

1.5.5. Potencialūs konkurentai.....	10
2.Suinteresuoti asmenys.....	10
2.1. Įmonės savininkai .....	10
2.2. Pardavėjai .....	11
2.3. Pirkėjai.....	11
2.4. Programinės įrangos inžinieriai, kuriantys aplikaciją.....	11
2.5. Išvados .....	11
3.Verslo modelis .....	12
3.1. Dalykinės srities modeliai .....	12
3.1.1. Pradinis modelis.....	12
3.1.2. Galutinis modelis .....	13
4.Reikalavimai .....	14
4.1.1. UML Panaudos atvėjų diagrama.....	14
4.2. Pradiniai reikalavimai.....	15
4.3. Funkciniai reikalavimai .....	15
4.4. Nefunkciniai reikalavimai .....	20
5.Validacija .....	21
5.1. Reikalavimų atsekamumo lentelė.....	21

# **1. Kontekstas**

## **1.1. Žodynas**

Pardavėjas – registruota maisto pardavimo įstaiga arba restoranas puslapyje parduodantys maisto produktus.

Pirkėjas – registruotas fizinis asmuo, galintis peržvelgti pardavėjų pasiūlymus ir įsigyti jų siūlomų maisto prekių.

Vartotojas (klientas) - pardavėjas, pirkėjas, ar kitas asmuo/įstaiga, naudojanti internetinės aplikacijos paslaugas.

## **1.2. Valdymo principai**

Mūsų nuomone norint, jog įmonė būtų sėkminga, ji turi ne tik rūpintis tuo, ką daro, bet ir žinoti, kodėl daro tai, ką daro. Toliau yra pateikiami mūsų pagrindiniai valdymo principai.

### **1.2.1. Misija**

Misija – sumažinti maisto švaistymo problemą.

### **1.2.2. Vizija**

Mūsų vizija – sujungti galutinį vartotoją su restoranų tinklais.

### **1.2.3. Vertybės**

Vertybės, kurių mes siekiame:

- 1. Tvarumas*
- 2. Bendradarbiavimas*
- 3. Sveika gyvensena*

## **1.3. Išorinė analizė**

*FoodKept* tai paslauga jungianti vietinius pardavėjus su pirkėjais. Paslauga šiuo metu yra pradinėje stadijoje, teikiama per svetainę, o galutinis produktas greičiausiai būtų ir mobilioji

programėlė. Šios paslaugos idėja atsirado iš esamų pasaulinių problemų, poreikio, tačiau tai yra ir akademinė užduotis.

### **1.3.1. Verslo modelis**

Kaip dauguma e-prekybos paslaugas teikiančių verslų, mūsų verslas pasirinko B2B2C (*Business to business to customer*) modelį.

### **1.3.2. Klientai**

Mūsų verslo modelis turi dvi atskiras klientų bazes – pardavėjus ir pirkėjus. Kiekvienai klientų bazei reikalingas skirtingas požiūris, kad būtų išlaikomos ir gaunamos pajamos, be vieno iš jų verslas nefunkcionuos.

#### **1.3.2.1. Pardavėjai**

Pagrindinis verslo pajamų šaltinis - pardavėjai, kadangi puslapis ne tik populiarintų jų paslaugas, tačiau ir atvestų naujų klientų, kurie jau naudojami platformos paslaugomis. Tai, žinoma, priklauso nuo pardavėjų sugebėjimo pritraukti naujų klientų, kas nebus įmanoma su maža prekių pasiūla. Tai ir būtų pagrindinė kliūtis verslui tapti pelningam.

#### **1.3.2.2. Pirkėjai**

Verslas iš pirkėjų tiesioginių pajamų negautų, nes pačiam paslaugos naudojimui nebūtų taikomas joks mokestis. Kadangi pirkėjai sudaro didžiąją dalį mūsų klientų, juos pritraukti yra būtina. Tai pasiektume padarydami mūsų programėlę visiškai nemokamą, kuo paprastesnę ir lengvai naudojamą.

#### **1.3.2.3. Partneriai**

Pagrindiniai verslo partneriai yra restoranai ir maisto įstaigos. Tai - mūsų verslo varomoji jėga. Kiti potencialūs partneriai yra reklamos įmonės bei maisto išvežiojimo kompanijos.

### **1.3.3. Verslo ribos**

Kadangi mūsų verslas yra paslaugų sektoriaus dalis, FoodKept internetinė aplikacijos pagrindinė funkcija – suteikti pardavėjams ir pirkėjams platformą, kurioje galima būtų parduoti ir pirkti maisto produktus. Dėl šių priežasčių verslas negali garantuoti produktų, kuriuos kelia ir administruoja pardavėjai, kokybės. Anaiptol, šios problemos potencialus sprendimas galėtų būti administratoriui priskiriamas įgaliojimas peržiūrėti ir trinti pardavėjų produktus. Tam prireiktų komandos, kuri būtų atsakinga už pardavėjo keliamų produktų kokybės patikrą.

### **1.3.4. Pajamų šaltinis**

Dabartinis puslapy variantas neturi galimybės generuoti pajamų, tačiau sugalvojome kelis būdus, kaip būtų imanoma tai pasiekti.

#### **1.3.4.1. Tiesioginis**

Generuoti pajamas iš pirkėjų būtų per sudėtinga, kadangi aplikacijos sėkmė priklauso nuo didelio klientų skaičiaus, todėl tai gali atbaidyti daug potencialių aplikacijos naudotojų. Dėl šios priežasties geriausias pajamų šaltinis būtų paslaugų apmokestinimas pardavėjams. Galimi būdai apmokestinti paslaugas:

1. Pasirenkamo laikotarpio (mėnesio/pusmečio/metų) prenumerata
2. Vienkartinis paslaugų mokestis
3. Papildomo matomumo mokestis

Kai kurie būdai apmokestinti paslaugas pardavėjams prieštarauja vienas kitam, todėl iš karto jų visų implementuoti nebutų imanoma, reikėtų pasirinkti geriausią.

#### **1.3.4.2. Netiesioginis**

Nors už aplikacijos paslaugas pirkėjai tiesiogiai mokėti neturės, netiesiogiai gauti pajamų iš šios vartotojų pusės yra imanoma puslapyje parduodant vietas reklamoms.

### **1.3.5. Išlaidos**

Pradžioje pagrindinės išlaidos būtų patikimos internetinės aplikacijos kūrimas. Jei aplikaciją kursime (tobulinsime) mes, ši dalis nekainuotų. Po aplikacijos sukūrimo ir paleidimo atsirastų kitų išlaidų:

1. Infrastruktūra (hostingas)
2. Aplikacijos taisymas
3. Klientų aptarnavimas
4. Marketingas

## **1.4. SSGG Analizė**

Atlikome savo produkto ir organizacijos SSGG analizę, kad įvertintume esamą situaciją.

### **1.4.1. Stiprybės**

1. Didelis potencialas, nes paslauga galėtų veikti visame pasaulyje (su atitinkamu kalbos pritaikymu).
2. Verslas nereikalauja jokios fizinės vietos.
3. Paprasta vartotojo sąsaja ir mobiliems pritaikytas dizainas.
4. Privaloma autentifikacija.

### **1.4.2. Silpnybės**

1. Paslaugos efektyvumas priklauso nuo didelės vartotojų bazės.
2. Šiuo metu neįdiegtas pajamų gavimo būdas.
3. Jokio personalizavimo.
4. Silpnas neteisėtų produktų reguliavimas.
5. Nestabili svetainė.
6. Vartotojai negali palyginti panašių/skirtingų pardavėjų produktų.

### **1.4.3. Galimybės**

1. Auganti rinka.
2. Augantis e-parduotuvių populiarumas.
3. Auganti tendencija pirkti paskutinius ir/ar baigiančius galioti produktus.
4. Automatinių klientų aptarnavimo konsultantų kūrimas.
5. Nėra tiesioginių konkurentų.

### **1.4.4. Grėsmės**

1. Įstatymai, dėl pandemijos, ribojantys pardavėjus.



2. Panašaus pobūdžio aplikacijos.
3. Padidėjęs duomenų nutekinimų skaičius.
4. Kompiuterinio raštingumo trūkumas.

#### **1.4.5. SSGG rezultatai**

Mūsų paslauga turi pastebimų privalumų, tačiau grėsmės ir trūkumai turi būti pripažinti ir mažinami, o galimybės išnaudotos tobulinant savo verslą. Pavyzdžiui, silpnybė, kad paslauga veiksminga tik esant didelei vartotojų bazei, galėtų būti pašalinta pritraukiant nepopuliarius restoranus ar maisto tiekimo įstaigas, pakviečiant įmones nemokamai naudotis platforma ir suteikiant reklamą. Įvardintos galimybės kaip auganti e-parduotuvių ir likusių produktų pirkimo tendencija gali atnešti greitą aplikacijos populiarumą. Viena iš stiprybių – autentifikavimas – yra privalomas, taip būtų pašalinta netikrų pardavėjų grėsmė. Paskutinioji grėsmė gali būti išsprendžiama įdiegiant mobiliąją aplikaciją, kuri palengvintų naudojimo procesą.

### **1.5. Porterio jėgos**

Norėdami geriau suprasti įmonės padėtį ir konkurencingumą rinkoje, atlikome 5 Porterių jėgų analizę.

#### **1.5.1. Konkurencija šakoje**

Naujai duotojo tipo internetinei aplikacijai rinkoje konkurenciją sukelia jau egzistuojančių restoranams/maisto įstagoms paslaugas tiekiančių verslų populiarumas. Kadangi tam, kad taptų konkurencingas, panašaus tipo verslas privalo sukurti plačią klientų bazę, mastu prilygstančią rinkoje lyderiaujantiems verslams, manome, jog konkurentams nugalėti prireikia didelių tiek laiko, tiek materialių investicijų. Taigi, naujiems verslams gana sudėtinga patekti į rinką.

#### **1.5.2. Pakaitų grėsmė**

Nors įmonės veiklai pakaito nėra, bet pačioje veikloje naudojamiems funkcionalumams pakaitų grėsmė reali. Rinkoje yra verslų, kurie galėtų potencialiai modifikuoti savo veiklą ir pildyti ją maisto pardavimui su nuolaidomis pritaikyti. Keli tokių verslų yra Wolt bei BoltFood. Kadangi pastarųjų verslų naudotojų bazė yra plati, būtų sunku išlaikyti FoodKept įmonės naudotojų lojalumą. Todėl pakaitų grėsmė yra gana aukšta.

### **1.5.3. Vartotojai**

Potenciali vartotojų bazė yra labai didelė – tiek pirkėjų, tiek pardavėjų. Yra labai didelis skaičius pardavėjų bei pirkėjų, todėl nei vienas vartotojas nėra kritinė mūsų aplikacijos veikimo dalis. Taip pat nėra jokių tiesioginių konkurentų, besistengiančių sustabdyti maisto švaistymą. Tačiau, kaip jau minėjome prieš tai, pakaitų grėsmė yra gan didelė, todėl svarbu patenkinti vartotojus.

### **1.5.4. Tiekėjai**

Mūsų paslaugai reikalingi minimalūs resursai. Vienintelis reikalavimas mūsų paslaugos veikimui IT sferoje yra infrastruktūra (hostingas) ir jos gebėjimas padengti išlaidas, nes jos keitimo kaina yra minimali. Tiekėjo galia yra labai maža.

### **1.5.5. Potencialūs konkurentai**

Šiuo metu, mūsų supratimu, tiesioginių konkurentų Lietuvos rinkoje nėra. Nors mums pavyko rasti panašiai veikiančių įmonių (pvz., *'Bolt Food, 'Wolt'*), tačiau nei viena iš jų nesuteikia paskutinės minutės produktų pirkimo galimybės. Tačiau, nors vietoje neturime tiesioginės konkurencijos, konkuruoti užsienio rinkoje būtų sunku atskirti savo produktą nuo konkurentų. Apskritai konkurencija yra maža.

## **2. Suinteresuoti asmenys**

Pagrindinis mūsų tikslas yra patenkinti kiek įmanoma daugiau suinteresuotų asmenų grupių. Norint tai padaryti yra būtina išanalizuoti kiekvienos suinteresuotų asmenų grupės reikmes ir norus, todėl sukūrėme sarašą, kuriame aprašome kiekvienos grupės reikmes.

### **2.1. Įmonės savininkai**

1. Stabilios pajamos.
2. Naudotojų pasitenkinimas.
3. Lengvas sistemos valdymas.
4. Pelningų pokyčių implementacija.
5. Konkurencingumo rinkoje išlaikymas.

6. Naujų naudotojų pritraukimas.

## **2.2. Pardavėjai**

1. Galimybė reklamuoti savo veiklą.
2. Galimybė pritraukti pirkėjus išskirtinės kokybės produktais.
3. Įvaizdžio kūrimas remiantis pirkėjų atsiliepimais (ši internetinė aplikacija yra gera vieta pirkėjų atsiliepimams ir įvertinimams rinkti).
4. Pirkėjų informavimas apie nuolaidas bei naujus maisto produktus.

## **2.3. Pirkėjai**

1. Galimybė rasti artimiausius pardavėjus.
2. Galimybė pasirinkti tinkamiausią variantą atsižvelgiant į kainas ir kitų vartotojų įvertinimus.
3. Informacija apie kainas, aptarnavimą, darbo laiką ir t.t.
4. Išreikšti nuomonę apie savo patirtį.
5. Tikra (nefiltruota) informacija.

## **2.4. Programinės įrangos inžinieriai, kuriantys aplikaciją**

1. Reikalavimų supratimas.
2. Apmokėjimas už patobulinimų įgyvendinimą.
3. Logiški terminai.
4. Konstruktyvūs atsiliepimai.

## **2.5. Išvados**

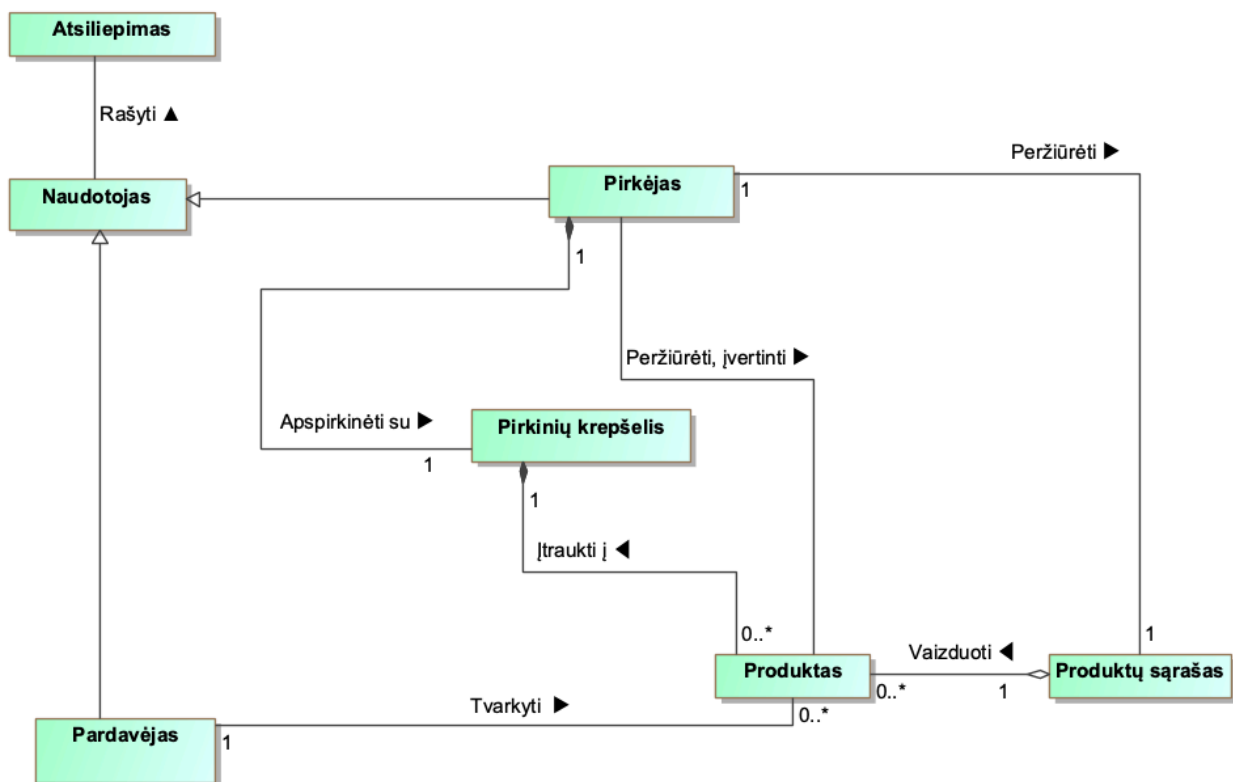
Norėdami plėstis, turime atsižvelgti į visų suinteresuotų asmenų grupių norus ir reikalavimus, bei stengtis kuo daugiau jų įgyvendinti. Svarbu išlaikyti nefunkcinius reikalavimus (programos veikimas, saugumas, pastovumas), tačiau tuo pačiu stengtis implementuoti naujas idėjas, kad patenkintume vartotojus. Taip pat svarbu paminėti, jog tvarkingi ir aiškūs reikalavimai padeda siekti šių tikslų.

### 3. Verslo modelis

#### 3.1. Dalykinės srities modeliai

Šiame poskyryje vaizduojami pradinis bei galutinis dalykinės srities modeliai, atspindintys pagrindines programos klases prieš implementuojant funkcionalumo papildymą, bei po jo.

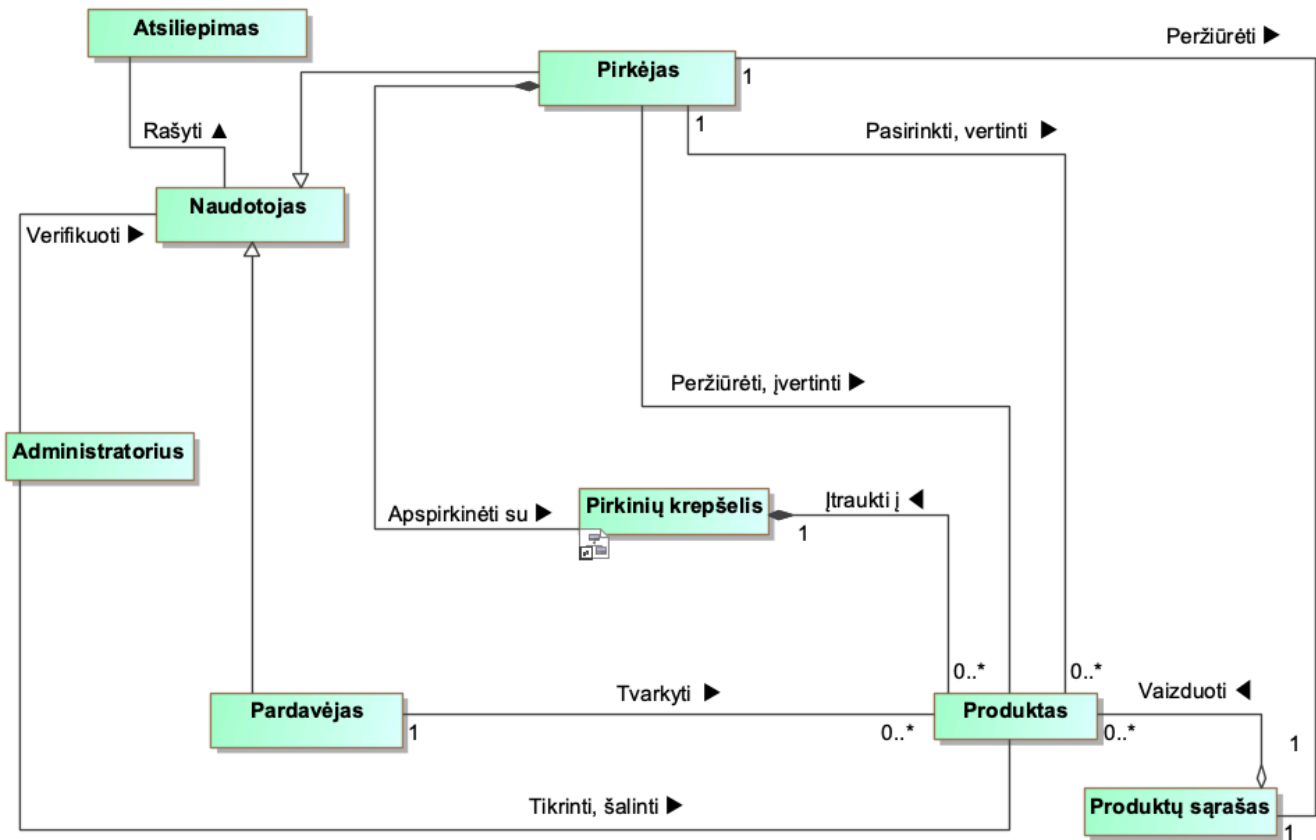
##### 3.1.1. Pradinis modelis



pav. 1 Pirminė UML diagrama

1 pav. vaizduojamas FoodKept internetinės aplikacijos dalykinės srities modelis prieš planuojamus pakeitimus. Modelio įgyvendinimui panaudota UML klasių diagrama, į kurią įtrauktos pagrindinės domeno esybės. Aplikacijos pagrindiniai naudotojai yra pirkėjai bei pardavėjai. Pastarieji tvarko svetainės produktų sąraše (parduotuvėje) esančius produktus – priskiria arba keičia jų nuotrauką, kainas, kiekius bei šalina juos. Šiuos kiekvieno restorano administruojamus produktus bendrame produktų sąraše gali įvertinti, peržiūrėti bei į savo pirkinių krepšelį pridėti prie aplikacijos prisijungę pirkėjai. Paskutinė aplikacijoje implementuota funkcija – atsiliepimai, kuriuos apie aplikaciją gali rašyti tiek naudotojai, tiek restorantai.

### 3.1.2. Galutinis modelis



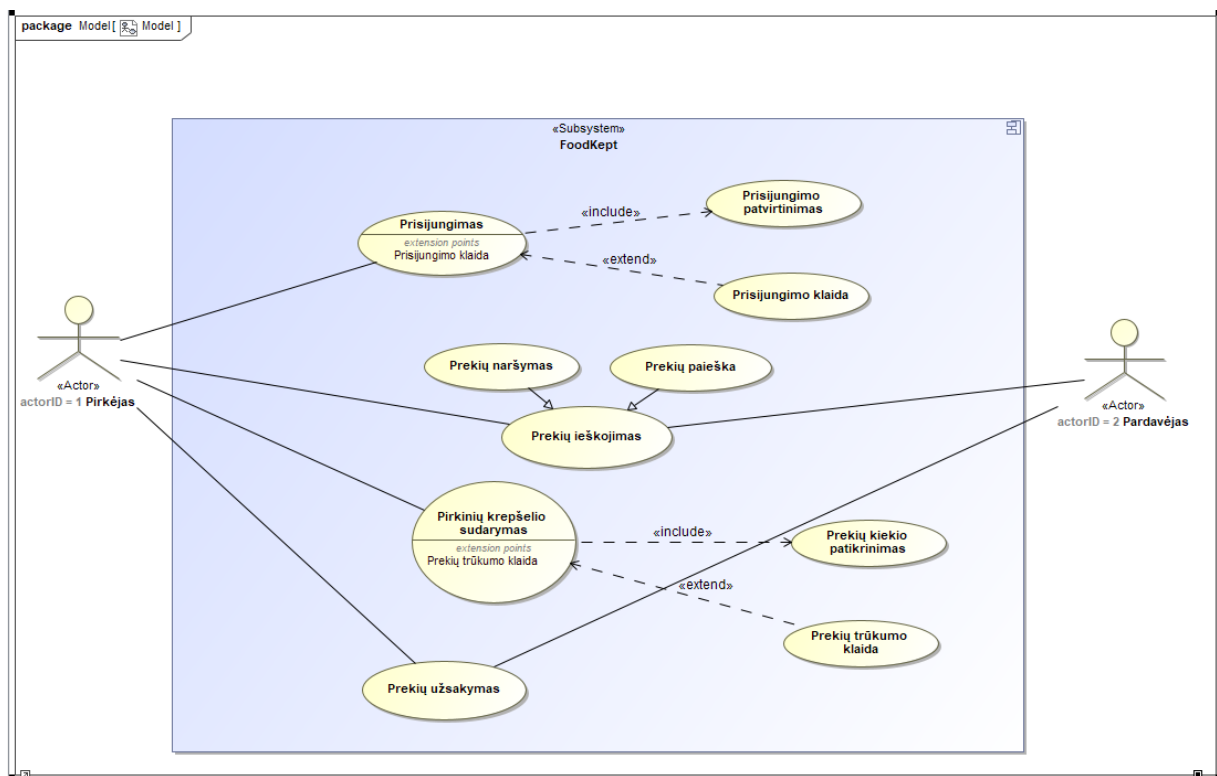
pav. 2 Galutinis UML modelis

2 pav. vaizduojamas internetinės aplikacijos modelis įvykdžius užsibrėžtus pakeitimus. Kadangi pagrindinis tikslas – sukurti aplikacijų programavimo sąsają (API), kuri leistų pardavėjams patogiau užregistruoti savo parduodamas prekes ar žaliavas, klasių modelis pasikeitė nestipriai. Vaizduojant antrinius išsikeltus reikalavimus modelis buvo papildytas administratoriumi. Ši klasė pridėta produktų bei naudotojų valdymui (verifikavimui, tikrinimui bei šalinimui).

## 4. Reikalavimai

### 4.1.1. UML Panaudos atvėjų diagrama

3 pav. pateiktas, aplikacijos veikimo principas naudojant panaudos atvėjų diagramą. Aplikacijos pakeitimai praplečia, prekių paieškos sistemą pirkėjams. Iliustracijoje, galite matyti pasirinkimą ieškoti arba naršyti prekes, tai palengvina naudotojo prekių ieškojimą.



pav. 3 Panaudos atvėjų UML diagrama

## **4.2. Pradiniai reikalavimai**

1. Sukurti aplikacijų programavimo sąsają (API), per kurią pardavėjai galėtų patogiau užregistruoti savo parduodamas prekes ar žaliavas.
2. Pridėti papildomų funkcijų pirkėjams: prekių paieška, pirkimų istorija, atsiliepimas apie produktą, detalesnė prekių esamos vietos informacija.
3. Pardavėjams turi būti galima pridėti naujas prekes, įkelti įvairaus formato paveikslėlius, redaguoti jau įkeltas prekes, ištrinti jas.
4. Sukurti administratoriaus rolę, galinčią redaguoti restoranų ir pirkėjų paskyras, skelbimus.

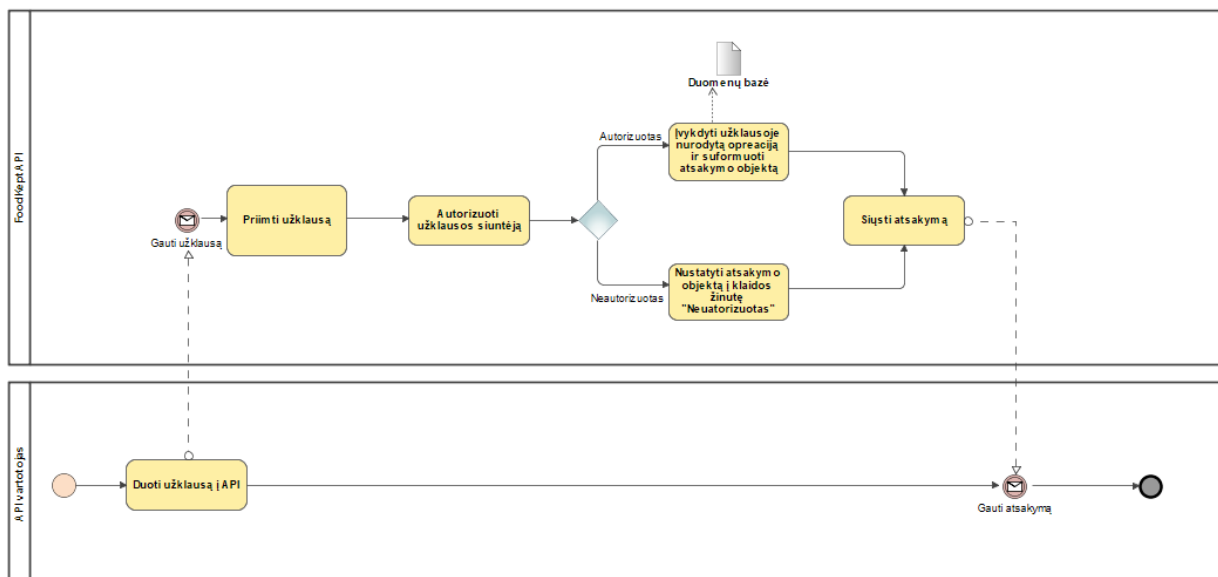
## **4.3. Funkciniai reikalavimai**

1. Pardavėjai turėtų galėti:
  - 1.1. Pridėti didelį skaičių parduodamų produktų
  - 1.2. Pakeisti parduodamo produkto aprašymą/nuotrauką/kiekį
  - 1.3. Ištrinti parduodamą produktą
  - 1.4. Norima tvarka išdėstyti parduodamus produktus
  - 1.5. Išsaugoti ir redaguoti informaciją apie produktą
2. Pridedant produktą turi būti nurodyta:
  - 2.1. Produkto pavadinimas
  - 2.2. Kaina
  - 2.3. Nuotrauka
  - 2.4. Aprašymas (pasirenkamai)
  - 2.5. Sudėtis (pasirenkamai)
3. Pirkėjai turi galėti:
  - 3.1. Pranešti apie netinkamą produktą
  - 3.2. Pranešti apie netinkamą skelbimą
  - 3.3. Pranešti apie netinkamą atsakymą
  - 3.4. Išreikšti savo nuomonę (feedback)

4. Administratorius turi galėti:
  - 4.1. Ištrinti netinkamą skelbimą
  - 4.2. Ištrinti netinkamo restorano paskyrą
  - 4.3. Ištrinti pirkėjo paskyrą
5. Produktų įkėlimo sistema, turėtų priimti bet kokio tipo paveikslėlio formatą.
6. Kiekvienas naujas produktas turi turėti unikalų produkto numerį, kurį pirkėjas galėtų išsaugoti savo užsakymų istorijoje.
7. Pirkėjas privalo turėti galimybę paieškos laukelyje atlikti paiešką pagal parduodamo produkto atstumą, produkto tipą ir kainą.
8. Pirkėjai privalo turėti galimybę teksto langelyje prie produkto/skelbimo/atsakymo pranešti apie netinkamą.
9. Pirkėjas privalo turėti galimybę žemėlapyje matyti parduodamo produkto vietą.

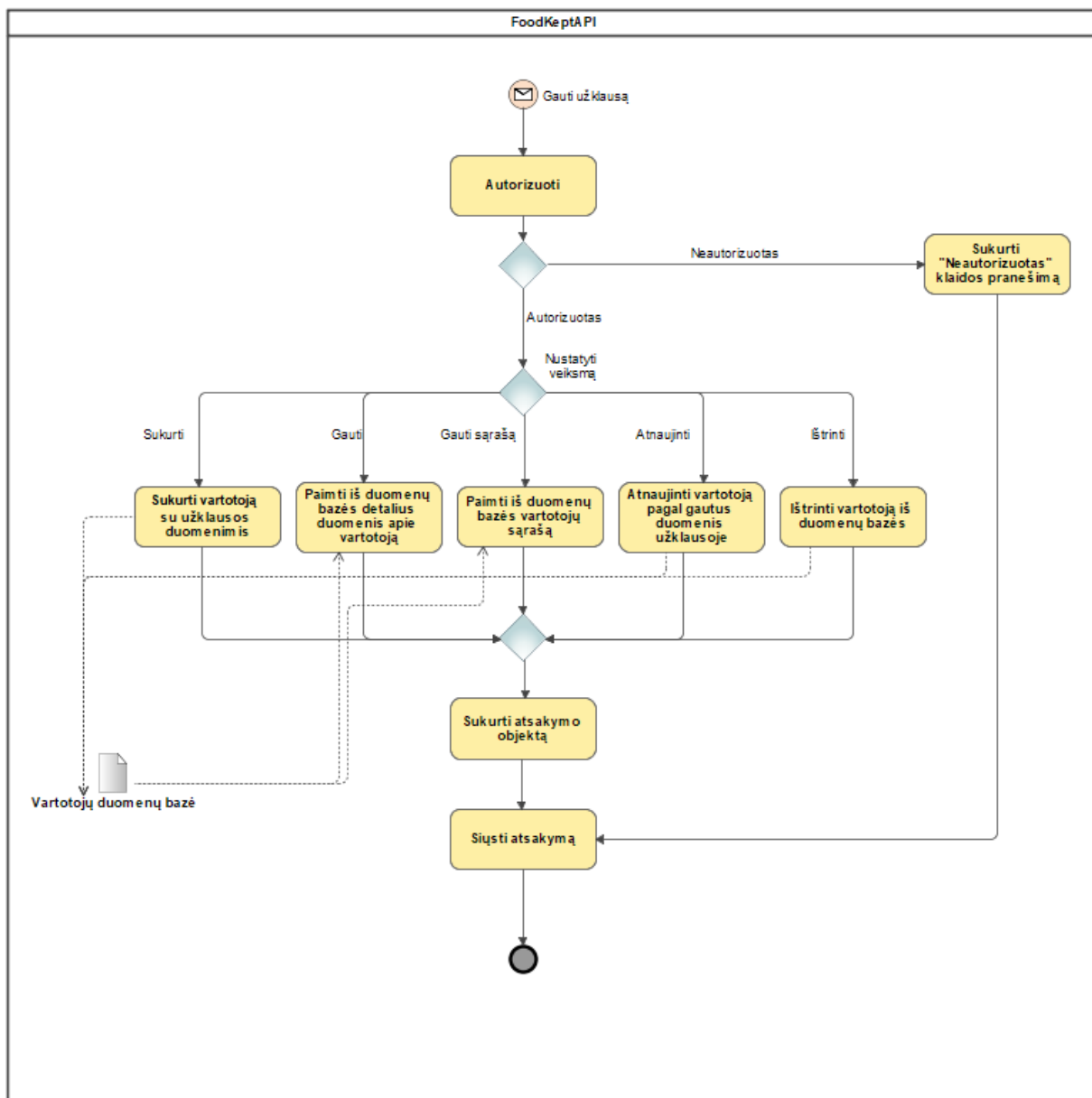
Išanalizavę reikalavimus ir verslo procesą, sukūrėme kelias BPMN diagramas, kurios atspindi bendrą verslo procesą, (4 pav.), taip pat mažesnius svarbius procesus detaliau (5 ir 6 pav.).





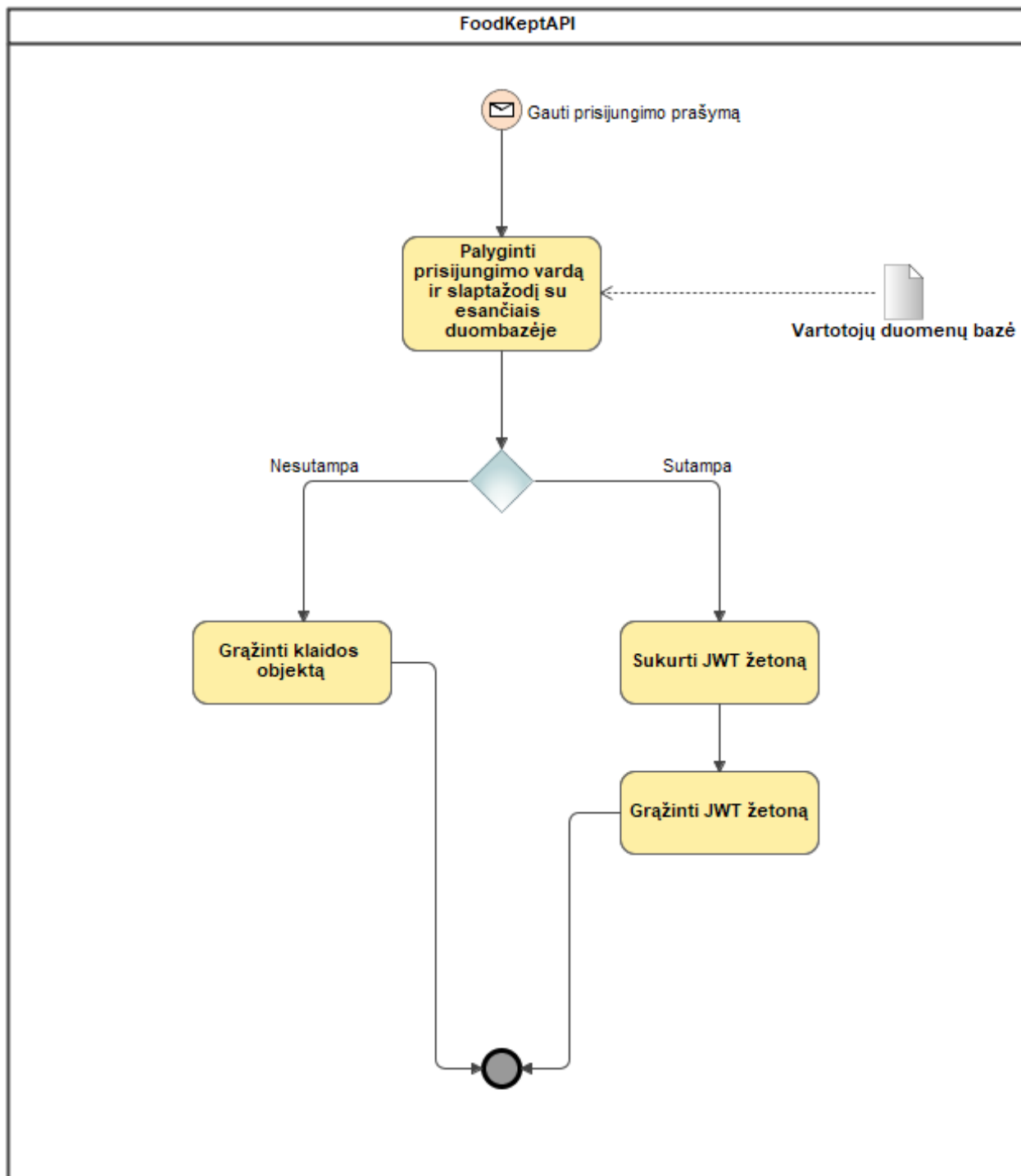
*pav. 4 API užklausų/atsakymų veikimo BPMN diagrama*

Pav. 4 vaizduojama API užklausų veikimo diagrama. Visą procesą pradeda vartotojas, šiuo atveju restoranas. Sistema gavusi užklausą pirmiausia ją autorizuoja, tuomet užklausoje nurodytas operacijas bei suformuoja atsakymo objektą ir jį išsiunčia atgal vartotojui.



pav. 5 Vartotojų valdymo BPMN diagrama

Pav 5. vaizduojama administratoriaus vartotojų valdymo BPMN diagrama. Autorizuotas administratorius gali vykdyti 5 veiksmus: Sukurti vartotoją, gauti informaciją apie vartotoją, gauti vartotojų sąrašą, atnaujinti vartotojo duomenis bei ištrinti vartotoją. Šioje diagramoje vartotojas yra ir restoranai, ir pirkėjai. Atlikti pakeitimai yra saugomi į duomenų bazę, grąžinamas objektas yra arba duomenys iš duomenų bazės arba redaguoto/ištrinto vartotojo detali informacija.



*pav. 6 Restorano prisijungimo per API diagrama*

Pav. 6 vaizduojama prisijungimo per API diagrama. Autorizacija kitose užklausose vyksta tikrinant JWT žetono galiojimą bei jame užkoduotą informaciją, todėl sėkmingai įvykdytą prisijungimo užklausa vartotojui sukuria žetoną.

#### **4.4. Nefunkciniai reikalavimai**

1. API turi būti pasiekiamas iš interneto.
2. API neveikimo laikas negali viršyti 3% viso laiko per mėnesį.
3. Vartotojų slaptažodžiai duombazėje turi būti užšifruoti
4. Duomenų bazė negali būti tiesiogiai pasiekama iš interneto
5. Visi galimi API veiksmai turi būti aprašyti pagal OpenAPI specifikaciją (<https://swagger.io/specification/>)
6. Sistema turi būti atspari SQL injekcijoms
7. API autorizacija turi būti įgyvendinta JWT žetonų pagalba, kuris galioja 1440 sekundes.

## 5. Validacija

### 5.1. Reikalavimų atsekamumo lentelė

Kad patikrinti, ar mūsų suformuluoti reikalavimai atitinka neformalius pradinius reikalavimus, sukūrėme atsekamumo lentelę (lentelė 1.).

	FR	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	2.5	3.1	3.2	3.3	3.4	4.1	4.2	4.3	5	6	7	8	9
Reikalavimai																							
Sukurti API		X	X	X	X	X	X	X	X	X	X								X				
Sukurti administratoriaus rolę																X	X	X					
Pridėti funkcijų pardavėjams		X	X	X	X	X	X	X	X	X	X												
Pridėti funkcijų pirkėjams												X	X	X	X					X	X	X	X

*lentelė 1. Reikalavimų atsekamumo lentelė*