



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Lukas Lukoševičius

EL. KOMERCIJOS REKOMENDACIJŲ SISTEMA GRAFŲ DUOMENŲ BAZĖJE
E-COMMERCE RECOMMENDATION SYSTEM BASED ON GRAPH
DATABASE

Baigiamasis bakalauro darbas

Programų inžinerijos studijų programa, valstybinis kodas 6121BX023
Programų sistemų studijų kryptis

Vilnius, 2023

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Lukas Lukoševičius

EL. KOMERCIJOS REKOMENDACIJŲ SISTEMA GRAFŲ DUOMENŲ BAZĖJE
E-COMMERCE RECOMMENDATION SYSTEM BASED ON GRAPH
DATABASE

Baigiamasis bakalauro darbas

Programų inžinerijos studijų programa, valstybinis kodas 6121BX023
Programų sistemų studijų kryptis

Vadovas

lektorė Jelena Stankevič

Konsultantas

-

Konsultantas

-

Vilnius, 2023

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Studijų kryptis: Programų sistemos

Studijų programa: Programų inžinerija , valstybinis kodas 6121BX023

Specializacija: Programų inžinerija

TVIRTINU

Katedros vedėjas

Nikolaj Goranin

2023-05-26

BAKALAURO BAIGIAMOJO DARBO (PROJEKTO) UŽDUOTIS

Nr. PRIfs19/5-4009

Vilnius

Studentas (-ė): Lukas Lukoševičius

Baigiamojo darbo (projekto) tema: El. komercijos rekomendacijų sistema grafų duomenų bazėje

Baigiamojo darbo (projekto) užbaigimo terminas pagal numatytą studijų kalendorinį grafiką.

BAIGIAMOJO DARBO (PROJEKTO) UŽDUOTIS:

Tikslas: Sukurti el. komercijos rekomendacijų sistemą grafų duomenų bazėje, siekiant įvertinti grafų duomenų bazių galimybes.

Uždaviniai:

1. Atlikti el. komercijos rekomendacijų sistemų, jų funkcijų, algoritmų apžvalgą.
2. Atlikti grafų duomenų bazių valdymo sistemų analizę.
3. Suformuluoti funkcinius ir nefunkcinius reikalavimus el. komercijos rekomendacijų sistemai.
4. Atlikti el. komercijos rekomendacijų sistemos projektavimą ir programavimą.
5. Atlikti sukurtos sistemos testavimą.

Planuojamas rezultatas: El. rekomendacijų sistema.

Vadovas Jelena Stankevič

Vilniaus Gedimino technikos universitetas	ISBN	ISSN
Fundamentinių mokslų fakultetas	Egz. sk.	
Informacinių sistemų katedra	Data*.....*	

Pirmosios pakopos studijų Programų inžinerijos programos bakalauro baigiamasis darbas 3	
Pavadinimas	El. komercijos rekomendacijų sistema grafų duomenų bazėje
Autorius	Lukas Lukoševičius
Vadovas	Jelena Stankevič

	Kalba: lietuvių
--	------------------------

Anotacija
<p>Baigiamojo darbo pagrindinis tikslas - sukurti el. komercijos rekomendacijų sistemą grafų duomenų bazėje, siekiant įvertinti grafų duomenų bazių galimybes. Šis darbas yra aktualus, nes šiais laikais nemenkas kiekis žmonių aktyviai perka produktus, siūlomus rekomendacijų sistemų, be to, taikomi grafų duomenų bazių sprendimai siekiant inovatyvumo ir optimalumo. Darbas atliktas kelias etapas. Iš pradžių atlikta teminė literatūros ir technologijų analizė. Tada atliktas sistemos projektavimas ir programavimas. Pabaigoje atliktas rankinis testavimas. Literatūra buvo metodiškai pritaikyta perskaitant ir aprašant aktualius teiginius. Sistema įgyvendinta pritaikant informacinės technologijas: „java“ programavimo kalbą, „Spring-Boot“ karkasą, „Thymeleaf“ šablono, „Neo4j“ grafų duomenų bazių valdymo sistemą. Įvertintos ir pritaikytos grafų duomenų bazių galimybės. Darbo rezultatas yra sklandžiai veikianti, grafų duomenų bazę paremta el. komercijos rekomendacijų sistema. Pagrindinės išvados yra padaryta el. komercijų rekomendacijų sistemų, funkcinių ir algoritmų apžvalga ir technologijų baigiamojo darbo problemų sprendimui pasirinkimas. Panašių technologijų palyginimas. Funkcinių ir nefuncinių reikalavimų sukūrimas siekiant kokybiškos ir detalios sistemos. Pritaikyta išmokta praktika sistemos projektavimui ir programavimui. Atlikus testavimą patikrinta ar sistema veiks be trikdžių.</p> <p>Darbą sudaro 7 dalys - įvadas, literatūros analizė, technologijų analizė, reikalavimų specifikacija, programų sistemos projektas, testavimas ir išvados.</p> <p>Darbo apimtis - 69 p. teksto be priedų, 33 paveikslėliai, 34 lentelės, 22 literatūros šaltiniai.</p>

Prasminiai žodžiai: Grafų duomenų bazių valdymo sistema, rekomendacijų sistema, el. komercija.

Vilnius Gediminas Technical University	ISBN	ISSN
Faculty of Fundamental Sciences	Copies No.	
Department of Information Systems	Date	

Bachelor Degree Studies Software Engineering study programme Bachelor Graduation Thesis 3	
Title	E-commerce Recommendation System Based on Graph Database
Author	Lukas Lukoševičius
Academic supervisor	Jelena Stankevič

	Thesis language: Lithuanian
--	-----------------------------

Annotation
<p>The main goal of the final thesis is to create an electronic commerce recommendation system based on a graph database in order to evaluate the capabilities of graph databases. This work is relevant because a significant number of people actively purchase products recommended by recommendation systems, and graph database solutions are applied to achieve innovation and optimum. The work consists of several stages. Firstly, a thematic analysis of literature and technologies was conducted. Then system design and programming were performed. Finally, manual testing was carried out. Literature was methodically applied by reviewing and describing relevant statements. The system was implemented using the following information technologies: the „Java“ programming language, the „Spring-Boot“ framework, the „Thymeleaf“ template, and the „Neo4j“ graph database management system. The capabilities of graph databases were assessed and adapted. The result of the work is a smoothly functioning electronic commerce recommendation system based on a graph database. The main conclusions include an overview of electronic commerce recommendation systems, functionalities, and algorithms, as well as the selection of technologies to solve the issues addressed in the final thesis. A comparison of similar technologies was made, and functional and non-functional requirements were created to ensure a high-quality and detailed system. The acquired practices were applied to system design and programming. Through testing, it was verified that the system functions without issues.</p> <p>The thesis consists of seven parts: introduction, literature analysis, technology analysis, requirements specification, system design, testing, and conclusions.</p> <p>The scope of the work includes 69 pages of text excluding annexes, 33 figures, 34 tables, and 22 literature sources.</p>

Keywords: Graph database management system, recommendation system, e. commerce.
--

SUTIKIMAS DĖL ASMENS DUOMENŲ NAUDOJIMO

2023-05-23

(Data)

Šiuo sutikimu aš, [\[Lukas Lukoševičius\]](#) (toliau – Duomenų subjektas) sutinku, kad Vilniaus Gedimino technikos universitetas, juridinio asmens kodas 111950243, adresas Saulėtekio al. 11, LT-10223 Vilnius (toliau – Duomenų valdytojas), tvarkytų mano asmens duomenis kitų studentų mokymosi tikslu. T. y. tvarkytų (*pažymėkite tinkamą*):

- ☒ vardą, pavardę, bakalauro baigiamąjį darbą;
- ☐ bakalauro baigiamąjį darbą, nenurodant vardo, pavardės;
- ☐ vardą, pavardę, magistro baigiamąjį darbą;
- ☐ magistro baigiamąjį darbą nenurodant vardo, pavardės.

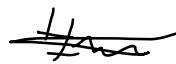
Šiuo tikslu tvarkomų asmens duomenų Duomenų valdytojas neperduos jokiems tretiesiems asmenims, studentams su baigiamaisiais darbais bus leidžiama susipažinti vidinėje informacinėje sistemoje. Duomenų subjekto asmens duomenys šiuo tikslu bus naudojami ne ilgiau nei 5 metai.

Šiuo sutikimu Duomenų subjektas patvirtina, kad yra supažindintas su šiomis teisėmis:

- Susipažinti su savo duomenimis ir kaip jie yra tvarkomi (teisė susipažinti);
- Reikalauti ištaisyti arba, atsižvelgiant į asmens duomenų tvarkymo tikslus papildyti asmens neišsamius asmens duomenis (teisė ištaisyti);
- Savo duomenis sunaikinti arba sustabdyti savo duomenų tvarkymo veiksmus (išskyrus saugojimą) (teisė sunaikinti ir teisė „būti pamirštam“);
- Reikalauti, kad asmens duomenų valdytojas apribotų asmens duomenų tvarkymą (teisė apriboti);
- Teisė į duomenų perkėlimą (teisė perkelti);
- Nesutikti, kad būtų tvarkomi asmens duomenys, kai šie duomenys tvarkomi ar ketinami tvarkyti kitais tikslais;
- Pateikti skundą Valstybinei duomenų apsaugos inspekcijai;

Duomenų subjektas turi teisę bet kuriuo metu atšaukti savo sutikimą.

Lukas Lukoševičius



[Duomenų subjekto vardas, pavardė, parašas]

Duomenų valdytojo rekvizitai:

Vilniaus Gedimino technikos universitetas

Juridinio asmens kodas: 111950243

Adresas: Saulėtekio al. 11, LT-10223 Vilnius

Tel. (8 5) 274 5030

Faks. (8 5) 270 0112

El. paštas: vgtu@vgtu.lt

PVM mokėtojo kodas: LT119502413

Duomenų apsaugos pareigūno tel. (8 5) 251 2191, el. paštas: dap@vgtu.lt

Turinys

TURINYS	7
PAVEIKSLĖLIŲ SĄRAŠAS	9
LENTELIŲ SĄRAŠAS	10
TERMINAI	11
IVADAS.....	8
1. LITERATŪROS ANALIZĖ	9
1.1. Rekomendacijų sistemos.....	9
1.2. Rekomendacijų sistemų implementavimas el. komercijoje	10
1.3. Rekomendacijų sistemų taikymo pavyzdžiai	11
1.3.1. Amazon.com	11
1.3.2. CDNOW	12
1.3.3. eBay	14
1.3.4. Rekomendacijų sistemų pritaikymo apibendrinimas	14
1.4. Grafaus paremtas rekomendacijų veikimas	14
1.5. Literatūros analizės apibendrinimas.....	15
2. TECHNOLOGIJŲ ANALIZĖ	17
2.1. Technologijų pasirinkimas.....	17
2.2. Neo4j grafų duomenų bazių valdymo sistema	17
2.2.1. Neo4j duomenų bazių valdymo sistemoje taikomi algoritmai.....	19
2.2.2. Apibendrinimas apie algoritmus	20
2.3. Microsoft Azure Cosmos DB duomenų bazių valdymo sistema	20
2.3.1. Microsoft Azure Cosmos DB apibrėžimas	20
2.3.2. Cosmos DB nagrinėjamos problemos.....	22
2.3.3. Papildomos ypatybės	23
2.4. DBVS palyginimas	23
2.5. Dalykinės srities aprašymas	25
2.6. Technologijų pasirinkimas.....	28
3. REIKALAVIMŲ SPECIFIKACIJA	29
3.1. Reikalavimų svarba.....	29
3.2. Funkciniai reikalavimai.....	29
3.3. Nefunkciniai reikalavimai	30
3.4. Sistemos reikalavimų apibendrinimas	31
4. PROGRAMŲ SISTEMOS PROJEKTAS	33
4.1. Projektavimo planas.....	33
4.2. Pagrindinės sistemos užduotys.....	33
4.2.1. Registracijos užduotis	35
4.2.2. Prisijungimo užduotis	37
4.2.3. Prekių pirkimo užduotis	39
4.2.4. Įvertinti prekę užduotis	41
4.2.5. Paskyros duomenų keitimo užduotis.....	43
4.2.6. Rekomendacijų valdymo užduotis	45
4.3. Pagrindinių sistemos užduočių apibendrinimas	47
4.4. Klasų struktūra	48

4.5. Duomenų bazės modelis	49
4.6. Sistemos architektūra	51
4.7. Sistemos įgyvendinimas.....	52
4.8. Programų sistemos projekto apibendrinimas	57
5. TESTAVIMAS	58
5.1. Testavimo eiga	58
5.2. Testavimo apibendrinimas	66
IŠVADOS	67
LITERATŪROS SĄRAŠAS	68
PRIEDAI.....	70
PRIEDAS 1	70
PRIEDAS 2	71
PRIEDAS 3	72
PRIEDAS 4.....	75
Įdėti prekę į krepšelį seka.....	75
Pašalinti iš krepšelio seka	76
Atrinkti prekes seka	77
Keisti slaptažodį seka.....	78
Keisti lytį seka.....	79
Keisti gyvenamąją vietą seka.....	80
Keisti gimimo metus seka	81
Apžvelgi populiariausias prekes seka	82
Apžvelgti rekomenduojamas prekes pagal paskyros duomenis seka.....	83
Apžvelgti rekomenduojamas prekes pagal profilį seka	84
Apžvelgti pirmo prisijungimo rekomendacijas seka.....	85
PRIEDAS 5	86
PRIEDAS 6.....	88

Paveikslėlių sąrašas

1 pav. Rekomendacijų paieška paremta grafais (DataStax, 2013).....	15
2 pav. Vartotojų profilio pavyzdys (DataStax, 2013)	15
3 pav. Vaizdusis paveikslėlis	25
4 pav. Užduočių diagrama	34
5 pav. Sistemos modelio klasių diagrama	49
6 pav. Konceptinis duomenų bazės modelis	50
7 pav. Neo4j sukurta duomenų vizualizacija	51
8 pav. Komponentų diagrama	52
9 pav. Netinkamas duomenų įvedimas registracijos lauke	53
10 pav. Netinkamas duomenų įvedimas keičiant vartotojo vardą	53
11 pav. Pradinis svetainės puslapis	54
12 pav. Neprisijungusio vartotojo pirkinių krepšelis	54
13 pav. Prisijungusio vartotojo rekomendacijos pagal profilį	55
14 pav. Perkamiausių prekių katalogas	55
15 pav. Personalizuota rekomendacija prisijungusiam vartotojui	55
16 pav. Prisijungusio vartotojo pirktų prekių katalogas	56
17 pav. Papildyta personalizuota rekomendacija	56
18 pav. Prisijungusio vartotojo visų vartotojų sąrašas	57
19 pav. Pirmasis serveris ir jo paleidimas	71
20 pav. Antrasis serveris ir jo paleidimas	71
21 pav. Krepšelių išsaugojimas grafų duomenų bazėje (Webber, 2021).....	72
22 pav. Paieškos į plotį algoritmo pavyzdys (Chao, 2018).....	73
23 pav. Įdėti prekę į krepšelį seka.....	75
24 pav. Pašalinti iš krepšelio seka.....	76
25 pav. Atrinkti prekes seka.....	77
26 pav. Keisti slaptažodį seka	78
27 pav. Keisti lytį seka	79
28 pav. Keisti gyvenamąją vietą seka	80
29 pav. Keisti gimimo metus seka	81
30 pav. Apžvelgti populiariausias prekes seka	82
31 pav. Apžvelgti rekomendacijas pagal paskyros duomenis seka	83
32 pav. Apžvelgti rekomendacijas pagal paskyros profilį seka	84
33 pav. Apžvelgti pirmo prisijungimo rekomendacijas seka	85

Lentelių sąrašas

1 lentelė. Būdai kuriais rekomendacijos sistemos patobulina el. komercijos svetaines (Deepjyoti & Mala, 2022)	10
2 lentelė. Neo4j ypatybės (Hoff, 2009).....	17
3 lentelė. Neo4j papildomas ypatybės (Neo4j, 2023)	18
4 lentelė. Microsoft Azure Cosmos DB ypatybės (DB-Engines, 2023)	21
5 lentelė. Neo4j ir Microsoft Azure Cosmos DB palyginimas	24
6 lentelė. Programavimo kalbų palaikymas	24
7 lentelė. Esamos būsenos vaizdžio paveikslėlio aprašymas.....	26
8 lentelė. Funkciniai reikalavimai	29
9 lentelė. Nefunkciniai reikalavimai	30
10 lentelė. Sistemos vartotojų istorijų paaiškinimai	34
11 lentelė. Registruotis sekų diagramos paaiškinimas.....	36
12 lentelė. Prisijungti sekos diagramos paaiškinimas	38
13 lentelė. Prekių pirkimas sekų diagramos paaiškinimas.....	40
14 lentelė. Įvertinti prekę sekų diagramos paaiškinimas	42
15 lentelė. Keisti prisijungimo vardą sekų diagramos paaiškinimas	44
16 lentelė. Apžvelgti rekomendacijas pagal kelių profilių duomenis sekų diagramos paaiškinimas	46
17 lentelė. Naudotojų užduočių pasiskirstymas pagal vaidmenis.....	47
18 lentelė. CRUD matrica	48
19 lentelė. Testavimo scenarijų aprašymas	58
20 lentelė. Testavimo scenarijų atvejai	60
21 lentelė. Įdėti prekę į krepšelį sekų diagramos paaiškinimas	75
22 lentelė. Pašalinti prekę iš krepšelio sekų diagramos paaiškinimas	76
23 lentelė. Atrinkti prekes sekų diagramos paaiškinimas	77
24 lentelė. Keisti slaptažodį sekų diagramos paaiškinimas	78
25 lentelė. Keisti lytį sekų diagramos paaiškinimas	79
26 lentelė. Keisti gyvenamąją vietą sekų diagramos paaiškinimas	80
27 lentelė. Keisti gimimo metus sekų diagramos paaiškinimas.....	81
28 lentelė. Apžvelgti populiariausias prekes sekų diagramos paaiškinimas.....	82
29 lentelė. Apžvelgti rekomendacijas pagal paskyros duomenis sekų diagramos paaiškinimas.....	83
30 lentelė. Apžvelgti rekomenduojamas pagal paskyros profilį sekų diagramos paaiškinimas	84
31 lentelė. Apžvelgti pirmo prisijungimo rekomendacijas sekų diagramos paaiškinimas	85
32 lentelė. Sistemos reikalavimų matrica	86
33 lentelė. Reikalavimų matrica komponentams	86
34 lentelė. Reikalavimų ryšio matrica.....	87

Terminai

API (angl. Application Programing Interface) – programėlių programavimo sąsaja

HTTP (angl. Hypertext Transfer Protocol) – hiperteksto perkėlimo protokolas

IT (angl. Information Technologies) – informacinės technologijos

CRUD (angl. Create Read Update Delete) – kurti, skaityti, atnaujinti ir ištrinti

SQL (angl. Structured Query Language) – struktūrizuotų paieškų kalba

IoT (angl. Internet of Things) – interneto dalykai

IP (angl. Internet Protocol address)– interneto protokolo adresas

DBMS (angl. Database Management System)– duomenų bazių valdymo sistema

GDPR – (angl. General Data Protection Regulation) bendros informacijos apsaugos reguliacija

Ivadas

Darbo aktualumas. Šis darbas yra skirtas nagrinėti el. komercijos rekomendacijų sistemas, kurios yra paremtos grafų duomenų bazėmis, apžvelgti įvairius pavyzdžius ir pabandyti sukurti analoginę sistemą. Šis darbas yra aktualus, nes įvairios komercija užsiimančios kompanijos naudoja ne tik reliacines duomenų bazines, bet ir grafais paremtas. Grafų duomenų bazės nuo reliacinių duomenų bazių išsiskiria savo veikimo principais, algoritmais ir pritaikymu. Rekomendacijų sistemos šiais laikais yra labai aktualus reiškinys, kuris liečia beveik kiekvieną asmenį gyvenantį modernioje visuomenėje. Dažniausiai rekomendacijų sistemos yra pritaikomos el. komercijoje, tačiau tokios sistemos yra pritaikytos ir kitose srityse tokiose kaip muzikos versle ar internetinėse reklamose.

Darbo problema – rekomendacijų sistemos taikymas grafų duomenų bazėje, siekiant optimizuoti veikimą ir funkcionalumą.

Darbo objektas – rekomendacijų sistema.

Darbo tikslas – sukurti el. komercijos rekomendacijų sistemą grafų duomenų bazėje, siekiant įvertinti grafų duomenų bazių galimybes.

Darbo uždaviniai:

1. Atlikti el. komercijos rekomendacijų sistemų, jų funkcijų, algoritmų apžvalgą.
2. Atlikti grafų duomenų bazių valdymo sistemų analizę.
3. Suformuluoti funkcinis ir nefunkcinis reikalavimus el. komercijos rekomendacijų sistemai.
4. Atlikti el. komercijos rekomendacijų sistemos projektavimą ir programavimą.
5. Atlikti sukurtos sistemos testavimą.

Tyrimo metodai. Paieškos internete, išteklių paieška skaitmeninėse bibliotekose, panašių sistemų lyginimas, argumentavimas, mokslinių darbų skaitymas ir pasirėmimas.

1. Literatūros analizė

1.1. Rekomendacijų sistemos

Rekomendacijų sistemos el. komercijoje šiuolaikinėje visuomenėje yra labai plačiai pritaikytos. Šios sistemos buvo pradėtos kurti ir buvo pradėtos taikyti prieš daugiau negu 20 metų. Per šį tarpą rekomendacijų sistemos pasikeitė nuo naujovės, kurią naudodavo tik keleta el. komercijos svetainių iki rimto versle pritaikomo įrankio. Šis įrankis iš esmės keičia el. komercijos pasaulį. Dauguma didžiausių internetinių puslapių jau naudoja rekomendacijų sistemas, siekiant padėti jų klientams rasti produktus, kuriuos jie nori įsigyti. Rekomendacijų sistemos mokosi iš kliento ir rekomenduoja produktus. Nagrinėjamajame šaltinyje yra pateikiami paaiškinimai kaip el. komercijos verslai pakelia pardavimus pasitelkus rekomendacijų sistemas (Schafer, Konstan, & Riedl, 1999). Keleta nagrinėjamų internetinių verslų naudoja daugiau negu vieną rekomendacijų sistemą. Pagal pavyzdžius yra sukuriami rekomendacijų sistemų klasifikacija. Į klasifikacijas įtraukiamos grafinės vartotojų sąsajos, kuriomis yra naudojasi vartotojai. Naujos rekomendacijų sistemoms el. komercijos srityje programuojamos pasiremiant idėjomis, naujomis technologijomis (Schafer, Konstan, & Riedl, 1999).

Rekomendacijų sistema yra informacijos sistemos poklasis, teikiantis pasiūlymus dėl elementų, kurie labiausiai tinka konkrečiam vartotojui. Rekomendacijų sistemos yra efektingas įrankis skaitmeninės informacijos filtravimui. Šios sistemos tapo plačiai naudojamos, dėl nuolatos besikeičiančių kompiuterių naudotojų įpročių, personalizacijos tendencijų ir vis platesnio prieinamumo prie interneto. Nors rekomendacijų sistemos pasižymi tiksliais rekomendacijomis, jos kenčia dėl visokių ribojimų ir iššūkių kaip lėtas paleidimas, sudėtingas pritaikymas ir panašiai. Dėl egzistuojančių įvairių būdų pritaikyti sistemą, pasirinkimas tampa kompleksiniu darbu, kai reikia sukurti rekomendacijų sistemą orientuotą į praktišką pritaikymą. Papildomai kiekvienas sistemos pritaikymo būdas turi savo ypatybių, privalumų ir trūkumų, iškelia skirtingus klausimus į kuriuos reikėtų atkreipti dėmesį. Rekomendacijų sistemos dažnai yra taikomos knygų, filmų, produktų rekomendacijoms, nukreiptoms į konkrečius vartotojus (Deepjyoti & Mala, 2022).

Siekiant tikslingai pritaikyti didelius informacijos kiekius nagrinėjamoje knygoje Masinis pritaikymas (Pine, 1993), Joe Pine argumentuoja, jog kompanijos turėtų pasikeisti nuo seno masinės produkcijos pasaulio, kuriame „standartizuoti produktai, homogeninės parduotuvės ir ilgas produktų gyvavimas ir kūrimo ciklai yra pagrindinė taisyklė“ į naują pasaulį, kuriame „įvairovė ir pritaikymas pakeičia standartizuotus produktus“. Pajudėjimas link el. komercijos suteikė galimybę kompanijoms savo klientams suteikti daugiau ir protingesnių pasirinkimų. Tačiau, prieš išsiplečiant į naują pritaikymo lygmenį, verslai turi padidinti informacijos kiekį, kurį vartotojai turi peržiūrėti. Peržiūrėjus didesnį

informacijos kiekį vartotojai gali pasirinkti iš didesnio prekių sąrašo, kurie yra pritaikyti vartotojo poreikiams. Vienas iš informacijos pertekliaus sprendimų yra rekomendacijų sistemos panaudojimas. Šių sistemų dėka vartotojų apsipirkimo patirtis tampa labiau personalizuota.

Rekomendacijų sistemos yra tapusios svarbiu el. komercijos įrankiu. Šis įrankis mokosi iš klientų, siekiant rekomenduoti geresnius produktus savo vartotojams. Šios sistemos yra plačiai naudojamos ir yra efektingas būdas informacijos filtravimui. El. komercijos pasaulis šiais laikais yra pasikeitęs nuo masinės produkcijos pasaulio iki smarkiai į vartotoją nukreipto ir personalizuoto pasaulio.

1.2. Rekomendacijų sistemų implementavimas el. komercijoje

Rekomendacijų sistemos yra naudojamos el. komercijos internetinėse svetainėse, siekiant taikliau pasiūlyti produktus vartotojams. Produktai gali būti rekomenduojami pagal parduodamiausias prekes tinklalapyje, remiantis vartotojų demografija, arba pagal ankstesnę vartotojų pirkimo elgseną, siekiant numatyti tolimesnę vartotojų pirkimo elgseną. Bendrai šios technikos yra personalizacijos dalis internetiniame tinklalapyje, nes jos leidžia tinklalapį adaptuoti pagal kiekvieną vartotoją. Rekomendacijos sistemos automatizuoja personalizaciją internete, įgalinant individualią personalizaciją kiekvienam vartotojui. Tokios apimties personalizacija yra vienas būdas realizuoti Pine's idėją internete. Pine tikriausiai sutiktų su Jeff Bezos, Amazon.com™ internetinės svetainės įkūrėju, kai jis pasakė „If I have 2 million customers on the Web, I should have 2 million stores on the Web.“. Išvertus lietuviškai ši sakinių jis nuskambėtų panašiai kaip „Jeigu mes turime 2 milijonus vartotojų internete, aš turėčiau turėti 2 milijonus parduotuvių internete“. Rekomendacijų svetainės labiausiai patobulina el. komerciją trimis būdais (Schafer, Konstan, & Riedl, 1999).

1 lentelė. Būdai kuriais rekomendacijos sistemos patobulina el. komercijos svetaines (Deepjyoti & Mala, 2022)

Būdas	Paaiškinimas
Prekių naršyklės nukreiptos į pirkėjus	Internetinių svetainių lankytojai dažnai atmetinai peržvelgia tinklalapį, siūlomus produktus ir nieko neperka. Tvarkingos rekomendacijų sistemos gali padėti pirkėjams rasti produktus, kuriuos jie nori įsigyti
Papildomas pardavimas	Rekomendacijų sistemos gali patobulinti papildomą pardavimą, protingai pasiūlydama papildomų produktų, kuriuos vartotojas gali įsigyti. Jeigu rekomendacijų sistemos yra geros, per laiko tarpsnį vidutinis užsakymo krepšelis turėtų padidėti. Pavyzdžiui, tinklalapis gali rekomenduoti papildomus produktus išsiregistravimo procese, pagal produktus, kurie jau yra pirkimo krepšelyje

Lojalumas	<p>Pasaulyje, kuriame tinklalapių konkurentai yra tik vienu arba dvejais paspaudimais vienas nuo kito, vartotojų lojalumo įgavimas yra būtina verslo strategija (Schafer, Konstan, & Riedl, 1999). Rekomendacijų sistemos propaguoja lojalumą, sukuriant pridėtinės vertės santykį tarp tinklalapio ir kliento. Tinklalapiai investuoja į išmokimą apie savo klientus, naudoja rekomendacijų sistemas, siekiant operatyvizuoti išmokimą ir reprezentuoja skirtingus interfeisus, kurie sutampa su klientų poreikiais. Klientai susimoka šiems tinklalapiams, grįžtant į tinklalapius, kurie atitinka jų poreikius labiausiai. Kuo daugiau vartotojas naudoja rekomendacijų svetainę – mokina rekomendacijų sistemą ko jie nori – tuo labiau jie tampa lojalūs tai svetainei. „Net jeigu konkurentas sukurtų sistemą su tokiu pačiu pajėgumu, klientas turėtų praleisti didžiulį kiekį energijos ir laiko mokinant konkurentą ką kompanija jau žino“ (Pine, 1993)</p>
-----------	--

Nagrinėjamajame šaltinyje el. komercijoje taikomos rekomendacijų sistemos aiškinamos penkias būdais. Pirma, yra aptariama keleta rekomendacijų sistemų pavyzdžių, pritaikytų El. komercijoje. Antra, yra nagrinėjama kaip kiekvienas pavyzdys naudoja rekomendacijų sistemas siekiant patobulinti internetinės svetainės pajamas. Trečia, yra apibūdinamas rekomendacijų sistemų išmaketavimas. Ketvirta, yra ištiriamos pastangos reikalingos vartotojams rasti naujas rekomendacijas. Penkta, yra apibūdinama keleta pasiūlymų naujoms rekomendacijų sistemų aplikacijoms.

Rekomendacijų sistemos yra implementuojamos internetinėse svetainėse, siekiant padidinti pardavimų kiekį. Produktai yra rekomenduojami pagal vartotoją. Atsižvelgiama į vartotojo elgseną. El. komercijos svetainėse modernios rekomendacijų sistemos patobulina veikimą.

1.3. Rekomendacijų sistemų taikymo pavyzdžiai

Sekančiame skyriuje yra aptariami trys El. komercijos verslai, kurie pritaiko vieną arba daugiau rekomendacijų sistemų variacijų, kurios yra panaudojamos internetiniuose tinklalapiuose. Kiekvienam tinklalapiui ir variacijai yra suteikiamas trumpas ypatybių apibūdinimas sistemoje.

1.3.1. Amazon.com

Pirkėjai, kurie jau pirkė (Customers who Bought): dauguma el. komercijos tinklalapių tokių kaip Amazon.com yra struktūrizuota su savo informacijos puslapiu kiekvienai knygai, suteikiamos detalės apie pirkinio informaciją. Pirkėjai, kurie jau pirkė ypatybė yra randama kiekviename knygos informacijos puslapyje. Iš esmės tai yra du skirtingi rekomendacijų sąrašai. Pirmas sąrašas rekomenduoja knygas, pirktas klientų, kurie nusipirkė pasirinktą knygą. Antras sąrašas rekomenduoja autorius, kurių knygos yra dažnai

pirkto klientų, kurie nusipirko kūrinį parašytą pasirinktos knygos autoriaus (Schafer, Konstan, & Riedl, 1999).

Akys (Eyes): Akys ypatybė informuoja klientus per elektroninį pašta apie naujas prekes pridėtas į Amazon.com katalogą. Klientai įveda savo užklausas pagal autorių, pavadinimą, temą, ISBN, arba publikacijos datos informaciją. Klientai gali panaudoti paprastas ir labiau kompleksiškas Boolean tipo (AND/OR) paieškas (Schafer, Konstan, & Riedl, 1999).

Dažnai pirkto kartu (Frequently Bought Together): Sukuriamos geresnės produktų rekomendacijos ir geresnė vartotojų patirtis. Papildomų produktų rekomendacija yra būdas padaryti apsipirkimo patirtį lengvesne ir paprastesne, rekomenduojant perkamą produktą papildančius produktus. Pavyzdžiui, perkant telefoną kartu rekomenduojama įsigyti ir telefono dėkliuką (Ayala, 2021).

Knygų atitikmenų ieškotojas (Book Matcher): Knygų atitikmenų ieškojimo ypatybė suteikia galimybę klientams duoti atsakomąjį ryšį apie knygas, kurias jie perskaitė. Klientai įvertina knygas, kurias jie perskaitė 5-taškų skalėje nuo „nekenčiau“ iki „labai patiko“. Įvertinus keleta knygų, klientai gali paprašyti gauti rekomendacijas knygų, kurios jiems galimai patiks. Šiuo tarpsniu pustuzinis neįvertintų tekstų yra atvaizduojami, kurie koreliuoja su vartotojų indikuotais skoniais. Atsakomasis ryšys šioms rekomendacijoms yra suteikiamos pagal „įvertink šias knygas“ ypatybę, kur klientai gali indikuoti įvertimą vienai arba daugiau rekomenduotų knygų (Schafer, Konstan, & Riedl, 1999).

Klientų komentarai (Customer Comments): klientų komentarai ypatybė leidžia klientams gauti teksto rekomendacijas pagal kitų klientų rekomendacijas. Įsikūręs kiekvienos knygos informacijos puslapyje yra sąrašas apie 1-5 žvaigždučių įvertinimus ir parašytus komentarus įvestus klientų, kurie yra perskaitę svarstomą knygą ir pateikę atsiliepimą. Klientai gali apsvarstyti kitų klientų komentarus (Svensson & Lyon, 2017). Vartotojų komentarai padeda klientams sužinoti daugiau apie produktą ir padeda nuspręsti ar tai jiems tinkamas produktas. Komentarai turėtų suteikti vartotojams geresnę įžvalgą apie produktą iš kitų vartotojų pusės (Amazon, 2023).

1.3.2. CDNOW

Albumų patarėjas (Album Advisor): albumų patarėjo ypatybė CDNOW™ (www.cdnw.com) veikia dvejais skirtingais režimais. Vieno albumo režime klientai suranda informacijos puslapį pasirinktam albumui. Sistema rekomenduoja 10 kitų albumų susijusių su pasirinktu albumu. Keleto artistų režime klientai įveda ne daugiau negu 3 skirtingus artistas. Po to, sistema rekomenduoja 10 albumų susijusių su pasirinktu artistu (Schafer, Konstan, & Riedl, 1999).

Mano CDNOW (My CDNOW): mano CDNOW leidžia klientams parengti jų pačių muzikos parduotuvę, paremtą albumais ir artistais, kurie jiems patinka. Klientai parenka albumus, kurios jie jau turi nusipirkę ir kurie artistas yra jų mėgstamiausi. Pirkiniai iš CDNOW yra įvesti automatiškai į „jau turiu“

sąrašą. Nors „jau turiu“ sąraše esančių pirminių įvertinimai iš pradžių yra gero įvertinimo indikacija, klientai gali sugrįžti atgal ir pakeisti savo įvertinimą į „jau turiu ir patinka“ ir „jau turiu bet nepatinka“. Kai klientai užklausia rekomendacijų, sistema nuspėja 6 albumus, kuriuos vartotojai mėgs pagal tai ką jie jau turi nusipirkę (Schafer, Konstan, & Riedl, 1999).

1.3.3. eBay

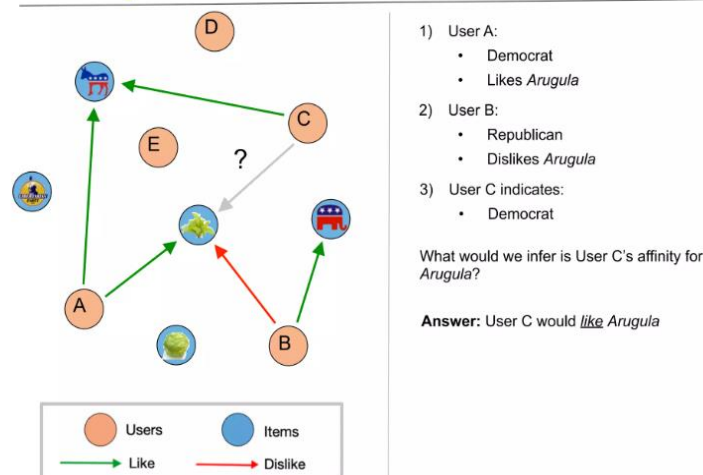
Atsakomojo ryšio profilis (Feedback Profile): Atsakomojo ryšio profilio ypatybė, kuri yra pritaikoma eBay.com svetainėje. Ji leidžia pirkėjams ir pardavėjams papildyti kitų klientų atsakomojo ryšio profilius su kuriais buvo atlikti pardavimo veiksmai. Atsakomąjį ryšį sudaro patenkinamumo įvertinimas (patenkintas/neutralus/nepatenkintas) ir konkretus įvertinimas apie kitą klientą. Atsakomasis ryšys yra panaudojamas siekiant sukurti rekomendacijų sistemą pirkėjams, kurie gali peržiūrėti pardavėjų profilius. Šį profilį sudaro lentelė, kurioje yra sunumeruoti sukurti įvertinimai per pastarąsias 7 dienas, pastarąjį mėnesį ir pastaruosius 6 mėnesius. Kartu yra pateikiamas apibendrinimas (pvz. 867 pozityvūs įvertinimai nuo 776 skirtingų klientų). Pateikus užklausa, klientai gali naršyti individualius įvertinimus ir pardavėjų komentarus (eBay, 2023).

1.3.4. Rekomendacijų sistemų pritaikymo apibendrinimas

Buvo aprašyta kaip trys el. komercijos verslai pritaiko vieną arba daugiau rekomendacijų sistemų variacijų. Šios variacijos yra pritaikomos internetiniuose puslapiuose. Variacijos išsireiškia elektroninių laiškų pritaikymu, ryšio tarp pardavėjo ir kliento tvirtinimu, įvairių profilių kūrimu. Galima teigti, jog yra įvairiausių būdų pritaikyti rekomencijų sistemas.

1.4. Grafais paremtas rekomendacijų veikimas

Kompanija e-Bay savo vartotojams rekomendacijas kuria remiantis grafais. Vartotojas ir produktai arba dalykai su kuriais vartotojai sąveikauja, yra apibrėžiami mazgais. Mazgai yra jungiami ryšiais, kurie nusako ar vartotojui patinka ar nepatinka kitas mazgas (DataStax, 2013). Rekomendacija yra surandama lyginant ką mėgsta kiti vartotojai. Jeigu vartotojai A ir C mėgsta republikonų žurnalą, o vartotojas A mėgsta gražgarstę, tai tikriausiai vartotojui C irgi patiks gražgarstė, todėl gražgarstė rekomenduojama vartotojui C. Yra ir vartotojas B, kuris nemėgsta gražgarstės, tačiau į jo nuomonę neatsižvelgiama, nes šiam vartotojui nepatinka republikonų žurnalas. Šios rekomendacijos yra apskaičiuojamas matematiškai ryšiams priskiriant įvairias skaitines reikšmes.



1 pav. Rekomendacijų paieška paremta grafais (DataStax, 2013)

Dažnai yra kuriami vartotojų profiliai siekiant nustatyti kas vartotojui patinka ir nepatinka. Vėliau šie profiliai yra lyginami tarp vartotojų siekiant surasti rekomendacijas surandant bendrai mėgstamus dalykus (DataStax, 2013).



2 pav. Vartotojų profilio pavyzdys (DataStax, 2013)

Kompanija e-Bay savo vartotojų profilių kūrimui rekomendacijų sistemoje taiko Linux operacine sistema operuojančias mašinas, Cassandra DBVS, 8GB JVM (Java virtualios mašinos) dinaminės atminties alokacijas, dydžio pakopų tankinimo strategiją (DataStax, 2013).

Rekomendacijos surandamos atsižvelgiant į vartotojų sukurtą skonio profilį. Šis profilis yra lyginamas su kitais vartotojas, tokiu būdu surandamos rekomendacijos.

1.5. Literatūros analizės apibendrinimas

Atlikus literatūros analizę galima geriau suprasti baigiamojo darbo kontekstą. Pasidomėjus ir perskaičius aktualų tekstą apie rekomendacijų pritaikymą el. komercijoje išugdomas gilesnis supratimas

kaip veikia el. komercija ir rekomendacijos. Šiuolaikinės el. komercijos parduotuvės sėkmingai taiko rekomendacijų sistemas ir grafais paremtas duomenų bazines. Pateikiami sprendimai, kuriuos galima pritaikyti savo sistemos kūrimui. Literatūros ir straipsnių šia tema yra pakankamai daug, norint suprasti kaip veikia rekomendacijos el. komercijoje. Taip pat randama naudingų išvalgų apie grafų duomenų bazių taikymą. Rekomendacijų sistemoms yra labai svarbi sąveika tarp vartotojo ir sistemos. Literatūros analizė yra labai svarbus procesas siekiant suprasti baigiamo darbo tikslus ir problemas, kurias siekiama išspręsti.

2. Technologijų analizė

2.1. Technologijų pasirinkimas

Baigiamojo darbo iškeltoms problemoms išspręsti buvo svarstomos skirtingos technologijos. Didžiausias dėmesys kreipiamas į duomenų bazių valdymo sistemas. Pasirinktos ir palygintos dvi populiariausios grafus palaikančios DBVS. Pagal įvairius kriterijus palyginus DBVS, šiuo atveju Neo4j ir Microsoft Azure Cosmos DB, buvo pasirinkta projektą projektuoti ir sukurti pasitelkiant Neo4j siūlomą grafais paremtą duomenų bazių sprendimą. Taip pat šiame skyriuje yra ir dalykinės srities aprašymas skirtas numatyti koks turėtų būti taikomų technologijų kontekstas. Programavimo technologijų pasirinkimo analizė nebuvo atlikta, nes buvo nuspręsta taikyti Java siūlomas technologijas.

2.2. Neo4j grafų duomenų bazių valdymo sistema

Dabartiniame IT sektoriuje yra naudojama daug skirtingų duomenų bazių valdymo sistemų. Didžiąją dalį duomenų bazių valdymo sistemų terpės užima reliacinės duomenų bazių valdymo sistemos, tačiau savo vietą turi ir ne reliacinės DBVS. Tokią sistemą kūrėjai apibūdina kaip ACID-atitinkančią (atomiškumas (angl. atomicity), nuoseklumas (angl. consistency), izoliacija (angl. isolation), tvarumas (angl. durability)) transakcinę duomenų bazę su gimtuoju grafų palaikymu ir apdorojimu. Neo4j yra sukurtas Java pagrindu ir yra pasiekiamas su programine įranga parašyta kitomis programavimo kalbomis. Duomenų bazės užklausos vykdomos pasitelkiant Cypher užklausų kalbą. Sąveikos vyksta per transakcinį Http galutinį tašką arba per dvejetainį Bolt protokolą (Hoff, 2009). Sistemoje viskas yra išsaugoma briaunos (angl. edge), mazgo (angl. node), atributo (angl. attribute) formomis. Kiekvienas mazgas ir briauna gali turėti bet kokią kiekį atributų. Mazgai ir briaunos gali turėti pavadinimus. Pavadinimai gali būti naudojami siekiant susiaurinti paieškas. Kiekvienas mazgas atitinka kokią nors esybę (asmenį, vietą, dalyką, kategoriją) arba kokią nors kitokią dalį informacijos ir kiekvienas ryšys arba dar vadinamas santykis apibūdina kaip du mazgai yra tarpusavyje susiję. Ši struktūra leidžia sukurti modelius įvairiems scenarijams nuo kelių sistemos, įrenginių tinklo iki populiacijos medicininių istorijų ir bet ko kito panašaus ką galima apibūdinti įvairias ryšiais. Siekiant išnagrinėti Neo4j yra išskiriamos ypatybės.

2 lenletė. Neo4j ypatybės (Hoff, 2009)

Pavadinimas	Aprašas
Apibūdinimas	Pasižyminti keičiamumu, ACID-palaikanti grafų duomenų bazė su puikaus efektyvumo paskirstyto klasterio architektūra, prieinama savarankiškai ir debesijoje

Reitingavimas	Iš daugumos šiuolaikinių duomenų bazių valdymo sistemų yra įvertinta 20-toje vietoje, o iš grafų duomenų bazių 1 vietoje
Licenzijos tipas	Atviro kodo
Tik debesijos palaikymas	Ne
DBaaS (duomenų bazes kaip serviso) pasiūlymai	Neo4j Aura: Neo4j's pilnai valdoma debesų paslauga: nėra administratoriaus, visada su grafais paremta duomenų bazė skirta debesų kūrėjams
Implementacijos kalbos	Java, Scala
Serveriai operuojantys Neo4j	Linux, OS X, Solaris, Windows
Duomenų schema	Be schemos ir neprivalomos schemos
Spausdinimas	Taip
Antriniai indeksai	Taip
SQL	Ne
APIs ir kitokie prieigos metodai	Bolt protokolas, Cypher užklausų kalba, Java API, Neo4j-OGM, RESTful HTTP API, Spring Data Neo4j, TinkerPop 3
Palaikomos programavimo kalbos	.Net, Clojure, Elixir, Go, Groovy, Haskell, Java, JavaScript, Perl, PHP, Python, Ruby, Scala
Serverio pusės skriptai	Taip
Trigeriai	Taip
Skaidymo metodai	Taip pasitelkiant Neo4j Fabric
Replikacijos metodai	Prižastinis klasteriavimas pasitelkiant Raft protokolą
MapReduce metodai	Ne
Tvarumo koncepcija	Priežastiniai ir galutinio nuoseklumo koncepcijos.
Išoriniai raktai	Taip
Transakcijų koncepcija	ACID
Lygiagretumas	Taip
Patvarumas	Taip
Vartotojo koncepcija	Vartotojai, rolės ir leidimai. Prijungiamas autentifikavimas su palaikomais standartais (LDAP, Active, Directory, Kerberos)

3 lentelė. Neo4j papildomas ypatybės (Neo4j, 2023)

Specifinės charakteristikos	Neo4j pritaiko grafų technologiją, kuri buvo ištestuota ir pritaikta puikiam efektyvumui ir dideliam masteliui. Neo4j esanti puikios
------------------------------------	--

	efektyvumo padalinto klasterio architektūra leidžia klientams naudotis sudėtingiausiais OLTP (Online Transaction Processing) procesais, kitaip tariant tinkliniu transakcijų procesavimu ir duomenų mokslo darbo krūviais, tuo pačiu metu išsaugant ACID principus ir duomenų vientisumą. Su Neo4j, klientai įgauna pasirinkimo laisvę dislokuoti savarankiškai paskelbtas, hibridines arba tarpdebesines platformas
Pagrindiniai klientai	Virš 800 naudotojų komercijos srityje ir virš 4300 startuolių naudoja Neo4j. Flagmaniniai klientai yra eBay, Walmart, Cisco, Citibank, ING, UBS, HP, CenturyLink, Telenor, TomTom, Telia, Comcast, Scripps Interactive Network, The National Geographic Society, Airbus, Orange, AT&T, Verizon, Dhs, US Army, Pitney Bowes, Microsoft ir daug panašių

Ši duomenų bazių valdymo sistema yra lyderiaujanti rinkoje, dėl savo funkcijų, pritaikytų technologijų, patikimo ir prieinamumo.

2.2.1. Neo4j duomenų bazių valdymo sistemoje taikomi algoritmai

Yra pritaikoma keleta protingų algoritmų siekiant Neo4j DBVS veikimą padaryti spartesnį ir optimalesnį. Pagrindiniai paieškos algoritmai būtų gylio (angl. depth-first) ir pločio (angl. breadth-first). Paieškos algoritmai pritaiko Dijkstra's algoritmą ir A* algoritmą. Darbo papildinyje (PRIEDAS 3, PRIEDAS 1) yra plačiau nagrinėjama kaip šie algoritmai yra pritaikomi paieškose.

Paiešką į gylį paieškos algoritmas įvykdo savo funkciją keliaudamas nuo pradžios mazgo iki kokios nors pabaigos mazgo, pasiekus pabaigą procesas yra kartojamas kitoku keliu iš to paties pradinio mazgo iki kol paieška yra sėkmingai pabaigiama. Paieškos į gylį algoritmas yra puikus pasirinkimas, kai yra siekiama atrasti diskrečias informacijos dalis. Tokie algoritmai yra geras strateginis pasirinkimas bendriems keliavimais grafais (Chao, 2018).

Paieškos į plotį paieškos algoritmai vykdo paieškas tiriant grafą sluoksniais. Šios paieškos pradedamos su mazgais vieno lygmens gylio atstumu nuo pradinio mazgo, toliau seka antro lygmens gylio mazgai, tada trečio gylio iki kol visas grafas bus praeitas.

Apibendrinant šie algoritmai paspartina Neo4j DBVS veikimą. Duomenims rasti paieška į gylį ir į plotį yra pritaikoma kitaip negu kitose DBVS.

2.2.2. Apibendrinimas apie algoritmus

Algoritmai taikomi DBVS yra skirti optimizuoti ir patobulinti veikimą, siekiant DBVS padaryti pranašesnę už kitas. Grafų paieškos algoritmai padeda naudotojui ieškoti duomenų pačiu efektyviausiu įmanomu būdu. Bet „labiausiai efektyvus“ priklauso nuo ieškomų rezultatų. Paieška į plotį nėra pats efektyviausias paieškos būdas, jeigu rezultatai labiau atitinka paieškos į gylį užklausas ir atvirkščiai (Chao, 2018).

2.3. Microsoft Azure Cosmos DB duomenų bazių valdymo sistema

Toliau nagrinėjama kita DBVS, kuri kai kuriais aspektais panaši į Neo4j. Apžvelgiamos savybės, pranašumai kitų sistemų atžvilgiu. Kompanija Microsoft siekia padėti geriau valdyti duomenis su tokiais servisas kaip Azure Cosmos DB, multi-modeline duomenų bazių paslauga, kuri yra skaidriai panaudoja ir replikuoja vartotojo duomenis skirtingose duomenų laikymo vietose, kad ir kur vartotojas būtų. Microsoft Azure Cosmos DB galima klasifikuoti kaip PaaS, tai reiškia platformą kaip servisą (Microsoft, Common Azure Cosmos DB use cases, 2019). Šią DBVS taiko dideli verslai.

2.3.1. Microsoft Azure Cosmos DB apibrėžimas

Nagrinėjant DBVS siekiant suprasti kuo ji išsiskiria, apibendrinamos įvairios savybės. Cosmos DB yra duomenų bazių paslauga, kuri yra paskirstyta globaliai. Ši paslauga leidžia vartotojui valdyti duomenis net jeigu jie yra saugomi duomenų centruose, kurie yra padalinti po visa pasaulį. Taip pat ši paslauga leidžia protingai paskirstyti kompiuterinius resursus.

Šis servisas palaiko keleta duomenų modelių pasitelkiant vieną galinį galą (angl. Backend). Tai reiškia, jog Cosmos DB galima pritaikyti dokumentų, raktinių verčių, reliacinių ir grafų modeliams. Cosmos DB daugiau ar mažiau yra NoSQL duomenų bazė, kadangi jai nereikia pasikliauti jokiais schemomis. Tačiau, nors ir yra panaudojama užklausų kalba panaši į SQL ir gali nesunkiai palaikyti ACID tipo transakcijas, kai kurie žmonės Cosmos DB klasifikuoja kaip NewSQL tipo duomenų bazę. Kas išskiria Cosmos DB nuo kitų NewSQL tipo duomenų bazių yra tai, jog nėra reliacinio duomenų modelio (Microsoft, Common Azure Cosmos DB use cases, 2019).

Microsoft Azure Cosmos DB yra sukurta ir palaikoma pagal GDPR standartus. Šie standartai yra skirti tam, jog kompanijos, kurios renka ir apdoroja asmeninius duomenis atsižvelgę į privatumo ir duomenų apsaugos principus (Microsoft, Azure Cosmos DB and GDPR, 2018). Ši DBVS pritaiko 4 būdus duomenų bazių ir jose esančios informacijos apsaugojimui.

1. Interneto protokolo adreso ugniasienė (IP Firewall).
2. Autentifikacija ir autorizacija (Authentication and authorization).

3. Vartotojų leidimai (User permission).

4. Geofencing'as (Geofencing).

Šiuo metu organizacijos kreipia didelį dėmesį į duomenis. Dėl to duomenų strategijoms yra teikiamas prioritetas vietoje kitų verslo strategijų. Daugumai organizacijų vertė, kurią teikia duomenys, turi didelę disruptinę galią. Kūrėjai transformuoja organizacijas pasitelkiant kitos kartos aplikacijas, varomas duomenų, tačiau duomenų dydžio, transportavimo greičio ir plečiamumo problemos numato didelius iššūkius. Organizacijoms siekiant grakščiai palaikyti naujus panaudojamumo atvejus arba kapitalizuoti naujas verslo galimybes, greitesnis, geresnis ir labiau patikimas priėjimas prie duomenų yra kritiškai svarbus. Šiem tikslam išspręsti yra kuriamas palaikomumas su įvairiausiomis naujomis aplikacijomis tokioms kaip MongoDB, įvairūs API ir kitokie agregatai (Leone, 2021). Siekiant išskirti aptariamą DBVS nuo kitų yra apžvelgiamos ypatybės.

4 lentelė. Microsoft Azure Cosmos DB ypatybės (DB-Engines, 2023)

Pavadinimas	Microsoft Azure Cosmos DB
Apibūdinimas	Globaliai paskirstyta, horizontaliai plečiama, multi-modelinė duomenų bazių paslauga
Pagrindinis duomenų bazės modelis	Dokumentų išsaugojimas Grafų DBVS Raktinių-verčių išsaugojimas Plačių stulpelių išsaugojimas
Licenzijos tipas	Komercinė licenzija
Tik debesijos palaikymas	Taip
Duomenų schema	Be schemos
Spausdinimas	Taip
Antriniai indeksai	Taip
SQL	SQL-tipo paieškų kalba
APIs ir kitokie prieigos metodai	DocumentDB API Graph API (Gremlin) MongoDB API RESTful HTTP API Table API
Palaikomos programavimo kalbos	.NET C# Java

	JavaScript JavaScript (Node.js) MongoDB client drivers written for various programming languages Python
Serverio pusės skriptai	JavaScript
Trigeriai	JavaScript
Skaidymo metodai	Sharding (skaldymas)
Replikacijos metodai	Taip
MapReduce metodai	Su Hadoop integracija
Išoriniai raktai	Taip
Transakcijų koncepcija	Multi-dalykinės ACID transakcijos su momentinių versijų izoliacija kažkurioje particijoje
Lygiagretumas	Taip
Patvarumas	Taip

Ši duomenų bazių valdymo sistema išsiskiria savo ypatybių gausumu, naujų technologijų pritaikymu ir integravimu stambiaus masto versluose.

2.3.2. *Cosmos DB nagrinėjamos problemos*

Tradiciškai prireiktų daug laiko ir sunkaus darbo siekiant sukurti panašią globaliai paskirstytą duomenų bazę, kurią vartotojas pats įgalintų savo duomenų centruose, pasitelkiant savo ryšius ir kitus resursus. Vykdomo procesas savyje dauguma atvejų nuvytų dauguma kompanijų net pagalvojus apie paruošimą naudotis Cosmos DB, dauguma svarsto, jog nebūtų verta investicijos. Kas nutinka yra tai, jog dauguma kompanijų verčiau pasirenka nesinaudoti Cosmos DB.

Debesų kompiuterijai ir platformai kaip servisui įžengus į rinką, pasidarė lengviau rasti globaliai paskirstytas duomenų bazes siekiamam tikslui. Dabar net įmanoma vienam asmeniui tik keliais paspaudimais sukurti architektūrą ir valdyti duomenų bazes su tik keliais pelės paspaudimais. Azure Cosmos DB perkelia tai į kitą lygmenį, suteikiant duomenų bazių sistemą, kurią yra įmanoma plėsti atsižvelgiant į poreikius.

Azure Cosmos DB turi nemažai skirtingų panaudojimo atvejų. Panaudos atvejai, kuriuos būtų galima išskirti būtų pritaikymas IoT ir telematikoje, perpardavinėjime ir marketinge, žaidimuose, tinklo ir mobiliosiose aplikacijose. IoT panaudos atvejai dažniausiai dalinasi atvejais kaip įsisavinamai, procesavimas ir duomenų išsaugojimas. Azure Cosmos DB yra plačiai pritaikytas Microsoft'o el. komercijos platformose tokiose kaip Windows Store ir Xbox Live. Žaidimuose duomenų bazių lygmuo yra

kritiškai svarbus komponentas. Ši DBVS yra dažnai pritaikoma tinklo ir mobiliuose aplikacijose ir yra puikiai tinkama socialinių interakcijų modeliavimui, integracijai su trečios šalies servisais (Microsoft, Common Azure Cosmos DB use cases, 2019).

Bendrai, Cosmos DB suteikia galimybę bizniui sukurti lanksčią duomenų bazę, kuri leistų jiems pasiekti savo verslo poreikius. Bet Cosmos DB yra ypatingai naudinga kompanijoms, kurios ieško duomenų bazių sistemos, kuri yra plečiama ir globaliai paskirstoma. Globaliai paskirstyta reiškia tai, jog resursai yra paskirstyti ir padalinti horizontaliai kiekviename pasaulio regione, kartu ir replikuota skirtinguose geografinėse vietovėse. Tai reiškia, jog delsos laikas yra minimalus ir kad vartotojai turi greitesnę, malonesnę patirtį (Microsoft, Common Azure Cosmos DB use cases, 2019).

2.3.3. Papildomos ypatybės

Vartotojai gali išnaudoti ir tokius privalumus kaip:

- **Paslauga parengta naudojimui:** tai suteikia produktą, kuris yra paremtas Azure ir gali būti automatiškai replikuotas duomenų centruose, kurie yra paskirstyti po visa pasaulį.
- **Multi-API:** kadangi duomenys yra indeksuojami automatiškai, vartotojai gali pasiekti juos pasitelkiant bet kokią savo pasirinktą API. Jei gali matyti savo duomenis pasitelkiant SQL, Gremlin, JavaScript, Azure Table Storage ir MongoDB.
- **Keletas nuoseklumo lygmenų:** CosmosDB naudoja penkis skirtingus nuoseklumo lygmenis: ribojamus senumo, stiprumo, sesijos, galutinumo ir nuoseklus-priešdėlio.
- **Delsos laikas:** labai žemas delsos laikas yra praktiškai garantuotas esant mažesniai nei 10 milisekundžių rašymo laikui ir esant mažesniai nei 15 milisekundžių rašymo laikui.

2.4. DBVS palyginimas

Šiame darbe nagrinėjama DBVS Microsoft Azure Cosmos DB gali būti klasifikuojama kaip NoSQL duomenų bazė kaip paslauga, o Neo4j grupuojama kaip grafų duomenų bazė. DBVS yra palyginamos siekiant numatyti, kuri būtų geriau tinkama numatytam tikslui. Neo4j išsiskiria savo nemokamumu ir atviru kodu, Microsoft Azure Cosmos savo efektyvumu (TrustRadius, 2016). Pasiekti duomenis, norint juos panaudoti, didžioji dauguma kompanijų, naudojančių DBVS pritaiko SQL užklausų kalbą, tačiau duomenims pasiekti naudojant Neo4j DBVS yra reikalaujama naudotis Cypher užklausų kalba (Hashmi, 2021). Debesyje esančią informaciją yra lengviau pasiekti su Microsoft Azure Cosmos DB, negu su Neo4j, nes pritaikomi skirtingi CRUD operacijoms realizuoti reikalingi mechanizmai. Siekiant apsaugoti vartotojų duomenis ir užkirsti kelią galimiems įsilaužimams Microsoft Azure Cosmos DB pritaiko daug daugiau

įvairių mechanizmų (Hashmi, 2021). Alternatyvos šioms DBVS yra ArrangoDB, Redis, OrientDB, MongoDB, ApacheCassandra, Microsoft Office Access (SaaSHub, 2022).

5 lentelė. Neo4j ir Microsoft Azure Cosmos DB palyginimas

Sistema	Neo4j	Microsoft Azure Cosmos DB
Pradinė kaina	Nemokama	\$65
Reitingas	8.4/10	5.0/10
Debesų palaikymas	Taip	Taip
Big Data kategorija	Taip	Ne
NoSQL duomenų bazių kategorija	Ne	Taip
ACID transakcijų palaikymas	Taip	Taip
Skaidymo metodas	Taip, pasitelkiant Neo4j Fabric	Taip, pasitelkiant Sharding (skaldymas)
Replikacijos metodai	Taip	Taip
SQL	Ne	Taip
Spausdinimas	Taip	Taip
Antriniai indeksai	Taip	Taip
MapReduce metodai	Ne	Taip
Trigeriai	Taip	Taip, JavaScript
Tik debesų palaikymas	Ne	Taip
Lygiagretumas	Taip	Taip

6 lentelė. Programavimo kalbų palaikymas

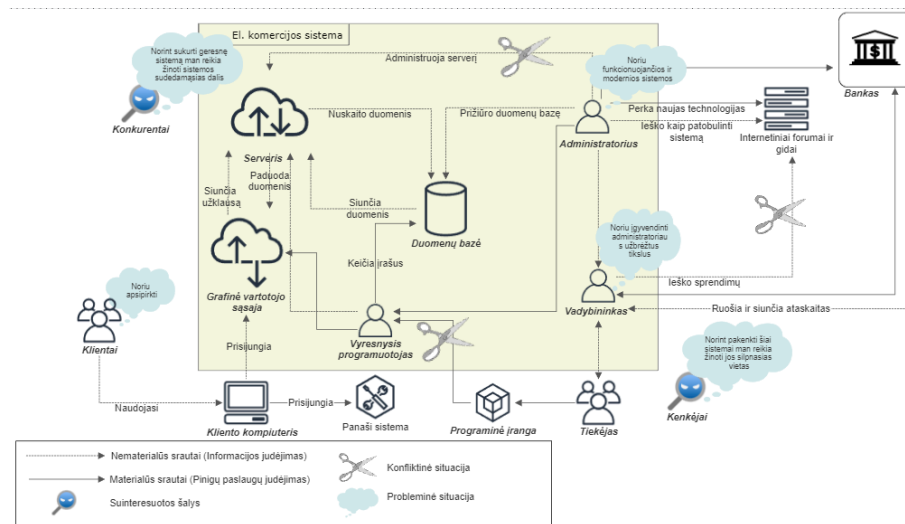
Sistema	Neo4j	Microsoft Azure Cosmos DB
Java	Taip	Taip
.Net	Taip	Taip
C#	Ne	Taip
JavaScript	Taip	Ne
Python	Taip	Taip
Clojure	Ne	Taip
PHP	Taip	Ne
Rudy	Taip	Ne
Scala	Taip	Ne

Neo4j palaiko daugiau programavimo kalbų (Neo4j, 2023), negu Microsoft Azure Cosmos DB (Kaushik, 2021).

Grafais paremtos DBVS yra gali būti pranašesnės už reliacines DBVS, nes teisingai pritaikius grafais paremtas DBVS galima padidinti pajamas, sukurti didesnę įsitraukimą, panaikinti nereikalingas rizikas. Sukuriamas ir geresnis veikimas. Pagerinami realaus laiko pajėgumai, sukuriamas gebėjimas pasinaudoto naujausiais transakciniais duomenimis, propoguojamas lankstumas inkorporuoti naujus duomenų šaltinius (Svensson & Lyon, 2017).

2.5. Dalykinės srities aprašymas

Dalykinės srities aprašymas yra skirtas iš sistemos projektuotojo perspektyvos apibūdinti kaip turėtų veikti projektuojama sistema. Kuriamo projekto dalykinė sritis yra el. komercija. Siekiama platesnio požiūrio atsižvelgiant į grėsmes, konkurentus, galimybes, rizikas ir panašiai. Požiūris formuluojamas patyrinėjus panašius praktiškai pritaikytus projektus. Baigiamojo darbo tikslo pasiekimui buvo patyrinėtos žinomos ir kokybiškai veikiančios el. komercijos svetainės. Grėsmės dažniausiai kyla iš išorinių veiksmų. Tai gali būti šnipinėjantys asmenys ar kokie nors kitokie kenkėjai. Konkurentai gali pritaikyti sėkmingas sistemos dalis be leidimo. Rizikos kyla iš patirties trūkumo. Dalykinę sritį pavyksta gražiai apibūdinti pasitelkus vaizdžiuosius paveikslėlius. Šių paveikslėlių dėka galima suprasti projektuotojaus numatomą sistemos vietą ir funkciją savo aplinkoje. Platesnis požiūris sukuriama įvertinus įvairius aspektus, kurie tiesiogiai arba netiesiogiai liečia ir įtakoja sistemą.



3 pav. Vaizdžius paveikslėlis

Vaizdžius paveikslėlis apibūdina sistemą ir jos aplinką. Projektuojamos sistemos viduje yra labai svarbūs serveriai. Pirmasis serveris įgalina Neo4j duomenų bazę, o kitas serveris iš šios duomenų bazės paima duomenis ir juos atvaizduoja grafinėje sąsajoje. Duomenis gražiai pavaizduoti yra siekiama

internetinėje svetainėje. Sistemoje dalyvauja įvairūs aktoriai. Galima išvardinti administratorių, kuris įjungia ir prižiūro sistemą, vyresnįjį programuotoją, kuris stengiasi įgyvendinti administratoriaus numatytas idėjas, vadybininkas, kuris bendrauja su tiekėjais. Aktoriai turi skirtingas atsakomybes ir prieigas. Aplinka nėra labai komplikauta. Svarbūs forumai ir gidai, kurių pagalba ieškoma būdų atnaujinti sistemą. Svarbus bankas, iš kurio imamos lėšos skirtos naujos programinės įrangos įsigyjimui. Klientai iš išorės pasiekia sistemos svetainę pasinaudodami savo techninę įrangą. Aplinkoje yra ir neigiamų šalių. Šiuo atveju kenkėjai ir konkurentai. Vaizdžiojo paveikslėlio pagalba sistema apibūdinama atsižvelgiant į išorinius ir vidinius veiksnus.

7 lentelė. Esamos būsenos vaizdžiojo paveikslėlio aprašymas

Schemos objektas	Objekto paskirtis (funkcijos)	Ryšiai su kitais objektais	Problemų
Administratorius	Prižiūro sistemą, ieško kaip pritaikyti naujas technologijas ir perka produktus siekiant patobulinti esamą sistemą	<i>Įeinantys ryšiai</i> Analizuoja konkurentų elgseną Valstybės įstatymai <i>Išeinantys ryšiai</i> Nurodymai darbuotojams Dokumentų pildymas	Nori funkcionuojančios ir modernio sistemos
Vadybininkas	Bendradarbiauja su tiekėjais, ruošia ir siunčia ataskaitas bankui	<i>Įeinantys ryšiai</i> Gaunama informacija iš bendradarbių Gauna pinigus, kuriuos skiria administratorius <i>Išeinantys ryšiai</i> Sudarius sutartį, užpildomi atitinkami dokumentai Konsultacijos su tiekėjais	Nori įgyvendinti administratoriaus užbrėžtus tikslus
Tiekėjas	Kuria naują programinę įrangą	<i>Įeinantys ryšiai</i> Vadybininko reikalavimai <i>Išeinantys ryšiai</i> Programinė įranga ir jos palaikymas	
Vyresnysis programuotojas	Tobulina ir prižiūro serverį, keičia ir	<i>Įeinantys ryšiai</i>	

	pritaiko duomenų bazės įrašus. Modifikuoja grafinę vartotojo sąsają	Reikalavimai naujos programinės įrangos kūrimui Sistemos duomenys ir pritaikomos technologijos <i>Išeinantys ryšiai</i> Programinės įrangos palaikymas. Naujų funkcijų įgyvendinimas	
Klientai	Pasinaudoja kompiuteriu ir sistemos grafine vartoto sąsaja siekiant apsipirkti	<i>Įeinantys ryšiai</i> Intencija kažką įsigyti	Nori apsipirkti
Programinė įranga	Pagerina sistemos veikimą		
Duomenų bazė	Išsaugo duomenis		
Internetiniai forumai ir gidai	Žiniatinklis, kuris padeda ieškoti sprendimų į kilusias problemas		
Bankas	Saugo ir išduoda pinigus		
Serveris	Valdo duomenų srautą tarp duomenų bazės ir grafinės vartotojo sąsajos		
Grafinė vartotojo sąsaja	Tarpininkauja tarp serverio ir kliento kompiuterio		
Kliento kompiuteris	Prisijungia prie sistemos		
Panaši sistema	Siūlo panašius sprendimus į		

	klientams kilusias problemas		
Kenkėjai	Žlugdo sistemos veikimą	<i>Įeinantys ryšiai</i> Stebi sistemos būseną	Norint pakenti sistemai siekia sužinoti jos silpnas vietas
Konkurentai	Kuria ir valdo panašias sistemas	<i>Įeinantys ryšiai</i> Stebi sistemos būseną	Norint sukurti geresnę sistemą jiems reikia žinoti priešo sistemos sudedamąsias dalis

Esamos būsenos vaizdžio paveikslėlio aprašymas yra skirtas geriau numatyti kokią sistemą yra siekiama sukurti. Ryšiai su kitais objektais skirti apibūdinti sąveikas vykstančias tarp skirtingų sistemos dalių. Aprašomos sistemos objektų problemos. Objektų aprašymas išplečia objektus. Siekiamą sistemą lengviau sukurti žinant kokie objektai sąveikauja tarpusavyje, kas yra sistemos ribose ir kas yra už sistemos ribų.

2.6. Technologijų pasirinkimas

Apibendrinant šį skyrių galima teigti, jog buvo atlikta svarbi technologijų analizė. Duomenų bazių valdymo sistemos pasirinkimas. DBVS yra kritinis veiksnys reikalingas išspręsti iškeltas problemas. Palyginus dvi populiarias DBVS buvo pasirinkta Neo4j DBVS, nes funkcionalumu ir suprantamumu atrodė pranašesnė už Microsoft Azure Cosmos DB. Taip pat ir apibūdinta sistemos dalykinė sritis, kurioje pritaikyta pasirinkta DBVS. Programavimo technologijų pasirinkimas nebuvo išsamiai svarstomas.

3. Reikalavimų specifikacija

3.1. Reikalavimų svarba

Reikalavimų specifikacija yra svarbus poreikis projektui. Reikalavimai padeda nuspręsti kokias sistemos dalis reikia plėtoti. Sistemos veikimas yra pagrįstas išpildytais reikalavimais. Reikalavimai išskiriami į funkcinis ir nefunkcinis. Toliau šiame skyriuje aprašyti reikalavimai. Sistemos projektavimo skyriuje yra bandoma surasti sprendimą iškeltiems uždaviniams, siekiama tenkinti iškeltus reikalavimus. Funkciniai reikalavimai labai svarbūs siekiant sukurti veikiančios sistemos pagrindą. Nefunkciniai reikalavimai yra labai svarbūs siekiant apibrėžti kraštutinius atvejus. Funkcinius ir nefunkcinius reikalavimus galima sugrupuoti pagal tai koks aktorius arba sistemos dalis.

3.2. Funkciniai reikalavimai

Funkciniai reikalavimai yra skirti apibūdinti sistemos elgseną koku nors atveju. Funkciniai reikalavimai turi savo numerį ir aprašymą. Reikalavimai formuojami apžvelgus praktikoje pritaikytas elektroninės komercijos sistemas. Reikalavimus galima skirstyti į grupes, pagal reikalavimuose esančius panašumus.

8 lentelė. Funkciniai reikalavimai

Reikalavimo nr.	Aprašymas	Užduoties nr.	Svarba
FR-0101	Vartotojas turi galėti sukurti naują paskyrą, įvedant prisijungimo vardą, slaptažodį, el. paštą ir pasirenkant papildomą informaciją	1	Aukšta
FR-0102	Vartotojas turi galėti prisijungti prie savo paskyros įvedant prisijungimo vardą ir slaptažodį	2	Aukšta
FR-0103	Vartotojas turi galėti rūšiuoti prekes pagal pasirinktą kategoriją	3	Aukšta
FR-0104	Vartotojas turi galėti įdėti pasirinktą prekę į krepšelį	3	Aukšta
FR-0105	Vartotojas turi galėti pašalinti pasirinktą prekę iš krepšelio	3	Aukšta
FR-0201	Sistema turėtų leisti prisijungusiam vartotojui įvertinti pirktas prekes	4	Maža
FR-0202	Sistema turėtų leisti vartotojui pirkti prekę arba prekes	3	Aukšta
FR-0203	Prisijungęs vartotojas turi galėti pasikeisti savo prisijungimo vardą, slaptažodį, gimimo metus, amžių ir gyvenamąją vietą	5	Maža
FR-0301	Rekomenduojamos populiariausios prekės yra rekomenduojamos pagal pardavimų kiekį	6	Vidutinė

FR-0302	Personalizuotos rekomendacijos kuriamos pagal tai ką registruotas vartotojas pirkęs anksčiau	6	Aukšta
FR-0303	Papildomos personalizuotos rekomendacijos kuriamos pagal tai ką registruotas ir susieti kiti registruoti vartotojai yra pirkę anksčiau	6	Aukšta
FR-0304	Pagal profilį personalizuotos rekomendacijos yra kuriamos pagal vartotojo profilio informaciją	6	Vidutinė
FR-0305	Pirmą kartą prisijungusiam vartotojui rekomenduojamos numatytos prekės	6	Vidutinė
FR-0306	Rekomenduojamos 3 skirtingos personalizuotos prekės	6	Aukšta

Su vartotoju susiję reikalavimai suformuluoja pradžią paprastam sistemos veikimui. Aprašomi reikalavimai, kuriuos pritaiko beveik visos internetinės svetainės. Į prisijungusį vartotoją atkreipti reikalavimai išplečia sistemos vartotojo galimybes. Prie sistemos prijungtas vartotojas turi daugiau ką daryti su preke. Prisijungęs vartotojas turėtų gauti personalizuotas rekomendacijas. Aprašyti funkciniai reikalavimai sukuria kontūrą kaip turėtų veikti pilnai įgyvendinta sistema. Apibūdinamas sistemos veikimas.

3.3. Nefunkciniai reikalavimai

Norint suprojektuoti kokybiškai veikiančią sistemą yra kuriami nefunkciniai reikalavimai. Kiekvienas nefunkcinis reikalavimas turi savo numerį ir aprašą. Teisingai suformuluoti nefunkciniai reikalavimai padeda apibūdinti kraštutinius atvejus. Šie reikalavimai išryškina svarbias smulkias detales. Šios detalės užtikrina sistemos veikimą be trikdžių. Skirtingiems vartotojams sistema tampa suprantamesne. Nefunkciniai reikalavimai taip pat skirti siekiant teisingai pritaikyti technologijas.

9 lentelė. Nefunkciniai reikalavimai

Reikalavimo nr.	Aprašymas	Svarba
NFR-0101	Vartotojui puslapiai turėtų užsikrauti greičiau negu per 5 sekundes, esant vidutiniškai apkrovai sistemoje	Aukšta
NFR-0102	Sistema turėtų pakrauti puslapius be klaidų 97% kartų per mėnesį	Vidutinė
NFR-0103	Programos kodas, kuriame apskaičiuojami duomenys turėtų bent jau 90% būti aprašytas Java kalba.	Aukšta
NFR-0201	Sistemos sukurta internetinė svetainė turėtų būti pasiekama 98% kartų per mėnesį naudojant Chrome, Firefox ir Edge interneto naršykles	Maža

NFR-0202	Sistemos vartotojo grafinė sąsaja turi būti įgyvendinta kaip internetinė svetainė, turi turėti bent jau 2 skirtingas sąsajas atitinkamai pagal vartotojo tipą ir privilegijas	Vidutinė
NFR-0203	Esant sistemos grafinės vartotojo sąsajos trikdžiams vartotojas turėtų galėti sąveikauti su preke 6/10 kartų	Aukšta
NFR-0204	Vartotojo tinklalapio sąsaja turėtų būti pateikta taisyklinga lietuvių kalba neįskaitant prekių pavadinimų	Aukšta
NFR-0205	Vartotojui įvedus netinkamus duomenis jam turėtų būti atvaizduojamas paaiškinamasis tekstas su įspėjimu	Vidutinė
NFR-0205	Vartotojo sąsaja turėtų veikti sklandžiai esant 100 vartotojų apkrovai	Vidutinė
NFR-0301	Siekiant neapkrauti sistemos, duomenų bazė turėtų gauti apie 1000 užklausų per valandą	Aukšta
NFR-0302	Vartotojui sąveikaujant su sistema duomenys skaitomi ir įrašomi į duomenų bazę tik per Spring-Boot serverį	Vidutinė
NFR-0303	Duomenų bazė turėtų būti nepasiekiamo ne dažniau negu 2 dienas per mėnesį	Vidutinė
NFR-0304	Duomenų bazę turėtų turėti bent vieną atsarginę kopiją	Vidutinė

Nefunkciniai reikalavimai atsižvelgia į platformas ir būdus pasiekti sistemą. Šios platformos ir būdai pasiekti sistema turi savo ribų. Į šias ribas reikia atsižvelgti siekiant sukurti prieinamesnę ir geriau integruotą sistemą. Reikalavimai su laiko intervalais, yra kuriami siekiant atkreipti dėmesį, jog kuriama sistema siekia pastovumo. Su duomenų baze kuriami nefunkciniai reikalavimai yra būtini, norit sukurti optimalesnę sistemą. Su rekomendacijomis susiję nefunkciniai reikalavimai yra reikalingi, nes jie padeda išplėsti jau esančius rekomendacijų funkcinius reikalavimus. Sąsajos nefunkciniai reikalavimai užtikrina, jog vartotojas greitai susiorientuos svetainės puslapiuose, nepasiklys tarp perteklinės informacijos ir greitai pasieks savo tikslą. Paprasta sistemos sąsaja leidžia greitai ir efektyviai išbandyti funkcinius reikalavimus. Norint turėti kokybišką sistemą yra reikalingi nefunkciniai reikalavimai. Šie reikalavimai apibūdina kraštutinius sistemos atvejus.

3.4. Sistemos reikalavimų apibendrinimas

Sistemos reikalavimai skirstomi į funkcinius ir nefunkcinius. Papildomi reikalavimai, reikalavimų ryšių matrica ir bazinių reikalavimų revizija yra (PRIEDAS 5). Sistema projektuojama pagal funkcinius reikalavimus, tačiau atsižvelgiama ir į nefunkcinius reikalavimus. Svarbiausi reikalavimai aprašo vartotojo

sąveikaujančio su sistema elgseną. Iš reikalavimų nuspėjamas visas sistemos veikimas. Sistema suprojektuojama gerai, jeigu tenkinami reikalavimai.

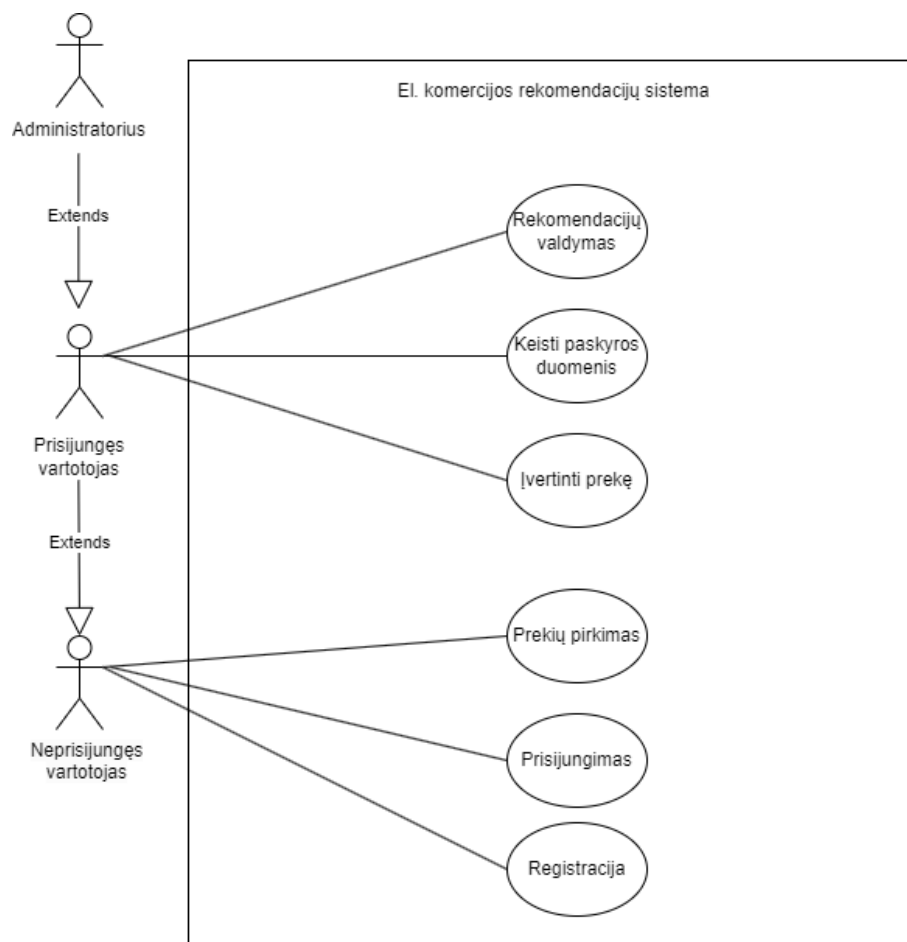
4. Programų sistemos projektas

4.1. Projektavimo planas

Šiame skyriuje yra aprašomas sistemos projektas. Aiškinama kokia architektura yra pasitelkta. Tada parodomos ir paaiškinamos klasių bei duomenų bazės diagramos. Duomenų bazės modelis sukurtas atsižvelgiant į duomenis, kurių gali prireikti aprašytiems sistemos funkcionalumams įgyvendinti. Svarbi dalis šiame skyriuje yra užduočių apipavidalinimas. Užduotys kuriamos pagal funkcinis reikalavimus. Užduočių diagramos yra papildytos sekų diagramomis. Suprojektuojamos svarbios sistemos užduotys, pagal kurias paremtas sistemos veikimas. Šio skyriaus pabaigoje yra aptariamas sistemos įgyvendinimas.

4.2. Pagrindinės sistemos užduotys

Užduočių UML diagramos yra skirtos suprojektuoti ir vizualizuoti sistemos įgyvendinimui reikalingas funkcijas. Užduotys detalizuojamos smulkesnėse diagramose. Šioms užduotims išplėtoti sukuriama sekų diagramos. Pagrindinės sistemos užduotys yra reikalingos siekiant įgyvendinti numatytus sistemos funkcinis reikalavimus. Vizualiai atvaizduoti reikalavimai leidžia susidaryti aiškesnį supratimą apie sistemos veikimą. Diagramos leidžia suprasti kaip susiję arba nesusiję sistemos aktoriai, reikalavimai. Sistemos užduotys suformuotos pagal funkcinis reikalavimus, kuriuos sėkmingai išpildo šiuo metu pasaulyje populiarios el. komercijos svetainės. Pagal užduotis sistema atlieka įvairius scenarijus. Scenarijams apibūdinti yra braižomos sekų diagramos. Paaiškinama kokios dalys sąveiką tarpusavyje. Sekų diagramos kaip ir užduočių diagramos atitinka funkcinis reikalavimus. Sekų diagramos tarpusavyje gali būti panašios.



4 pav. Užduočių diagrama

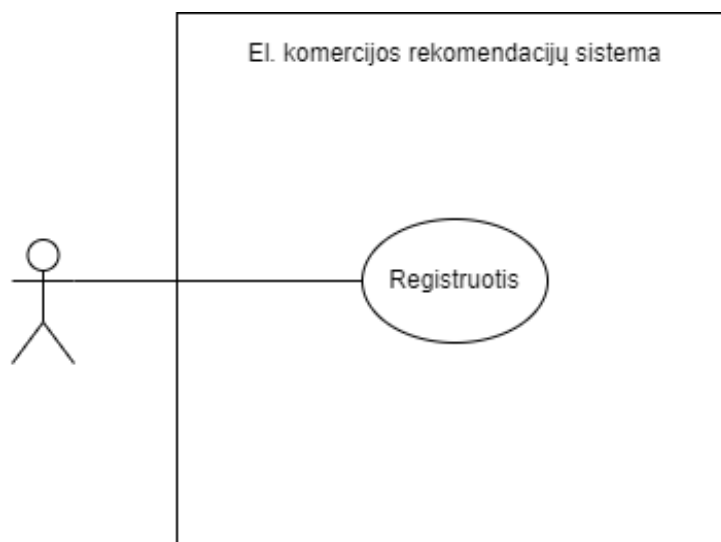
Iš bendros sistemos užduočių diagramos išvedamos smulkesnės sistemos užduotys, be kurių įgyvendinimo sistema neišspręstų vartotojo problemų. Šioms užduotims nubraižomos sekų diagramos, taip detalizuojamos užduotys. Bendra sistemos užduočių diagrama leidžia suprasti, jog sistemos veikimą galima patogiai suskirstyti. Patogiai apsibrėžus užduotis yra efektyviau ir greičiau projektuojama, o vėliau ir programuojama sistema.

10 lentelė. Sistemos vartotojų istorijų paaiškinimai

Nr.	Pavadinimas	Aprašymas
1.	Registracija	Kaip neprisijungęs vartotojas, noriu registruotis, kad galėčiau pasiekti papildomas svetainės funkcijas
2.	Prisijungimas	Kaip neprisijungęs vartotojas, noriu prisijungti, kad galėčiau pasiekti savo paskyrą
3.	Prekių valdymas	Kaip neprisijungęs vartotojas, noriu valdyti prekes, kad galėčiau įdėti prekę į krepšelį, pašalinti prekę iš krepšelio, pirkti prekes ir atrinkti prekes

4.	Įvertinti prekę	Kaip prisijungęs vartotojas, noriu įvertinti prekę, kad palikčiau savo atsiliepimą
5.	Keisti paskyros duomenis	Kaip prisijungęs vartotojas, noriu keisti paskyros duomenis, kad galėčiau keisti slaptažodį, prisijungimo vardą, gimimo metus, lytį ir gyvenamąją vietą
6.	Rekomendacijų valdymas	Kaip prisijungęs vartotojas, noriu valdyti rekomendacijas, kad galėčiau apžvelgti populiariausias prekes, apžvelgti rekomendacijas pagal paskyros duomenis, apžvelgti rekomenduojamas prekes pagal profilį, apžvelgti rekomendacijas pagal kelių paskyrų duomenis ir apžvelgti pirmą kartą prisijungusiam vartotojui sukurtas rekomendacijas

4.2.1. Registracijos užduotis



22 pav. Registruotis užduotis

Užduotis: Registruotis užduotis.

Pagrindinis aktorius: Vartotojas.

Apimtis: El. komercijos rekomendacijų sistema.

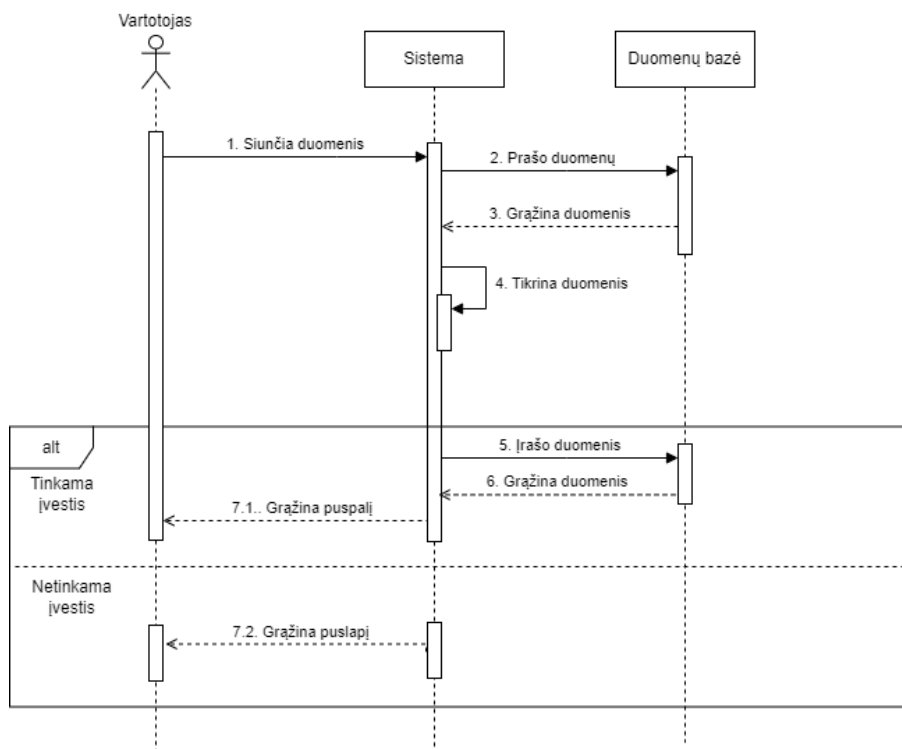
Aprašymas: Vartotojas gali registruotis prie sistemos.

Trigeris: Vartotojas pasirenka atlikti veiksmą su sistema.

Veiksmų seka:

- 1.Siunčiami duomenys sistemai.
- 2.Sistema siunčia užklausą į duomenų bazę ir gauna atsaką.
- 3.Sistema patikrina duomenis.
- 4.Sistema siunčia užklausą į duomenų bazę užrašyti duomenis ir gauna atsaką.

5.Sistema grąžina atsaką vartotojui.



23 pav. Registruotis užduoties seka

Sekos pavadinimas: Registruotis užduoties seka.

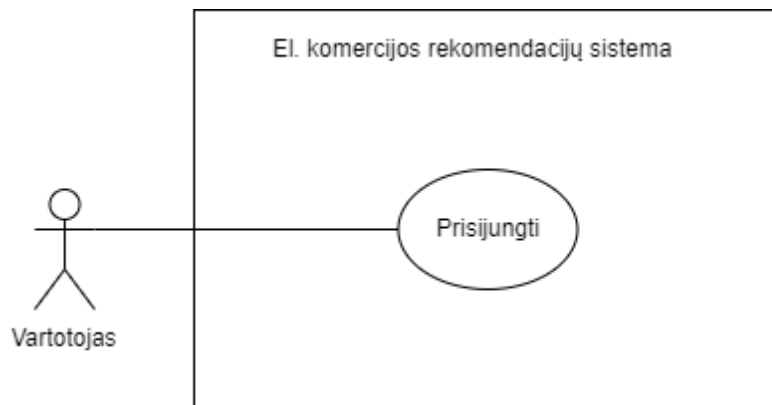
Sąveikos objektai: Vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0101

11 lentelė. Registruotis sekų diagramos paaiškinimas

užduoties numeris	1
užduoties pavadinimas	Registruotis užduotis
siekiamas tikslas	Registruotis prie sistemos
užduoties vykdymą inicijuojantis įvykis (trigeris)	Neprireistravęs vartotojas pasirenka atlikti veiksmą su sistema
užduoties prioritetas	1-as (aukštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė, grafinė vartotojo sąsaja
"prieš" sąlygos	Sėkmingai pasiekta internetinė svetainė
sėkmingos baigties "po" sąlygos	Sėkmingai pasiekta internetinės svetainės dalis skirta prisijungusiam vartotojui
nesėkmingos baigties "po" sąlygos	Nepasiekta internetinės svetainės dalis skirta prisijungusiam vartotojui

4.2.2. Prisijungimo užduotis



22 pav. Prisijungti užduotis

Užduotis: Prisijungti užduotis.

Pagrindinis aktorius: Vartotojas.

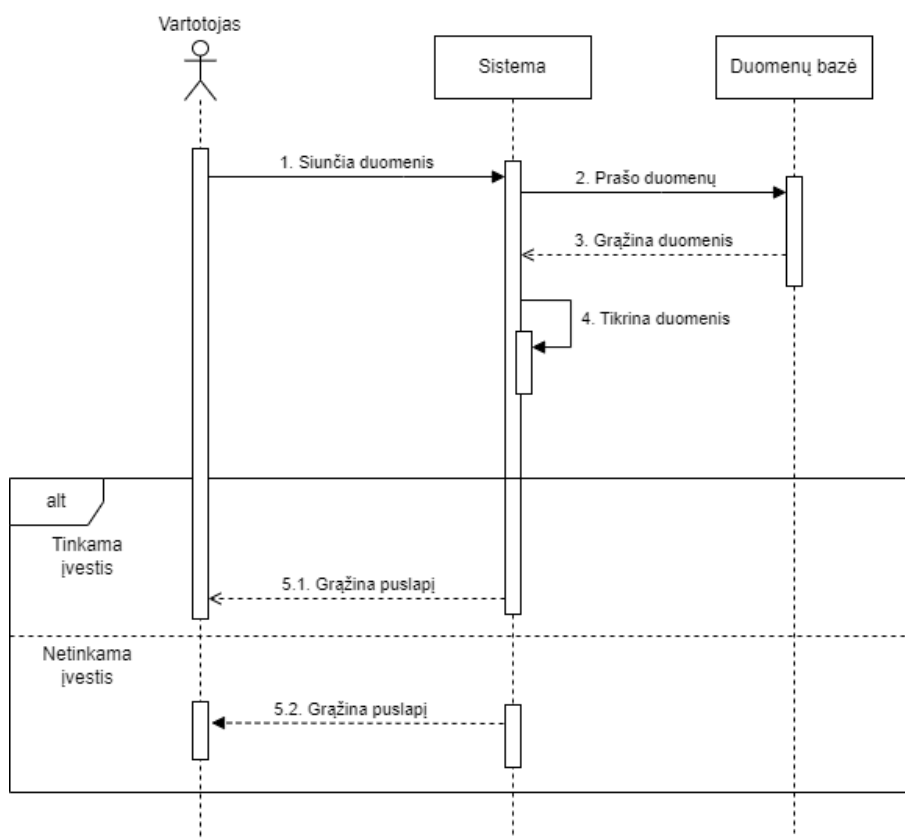
Apimtis: El. komercijos rekomendacijų sistema.

Aprašymas: Vartotojas gali prisijungti prie sistemos.

Trigeris: Vartotojas pasirenka atlikti veiksmą su paskyra.

Veiksmų seka:

- 1.Siunčiami duomenys sistemai.
- 2.Sistema siunčia užklausą į duomenų bazę ir gauna atsaką.
- 3.Sistema patikrina duomenis.
- 4.Sistema siunčia užklausą į duomenų bazę užrašyti duomenis ir gauna atsaką.
- 5.Sistema grąžina atsaką vartotojui.



23 pav. Prisijungti seka

Sekos pavadinimas: Prisijungti seka.

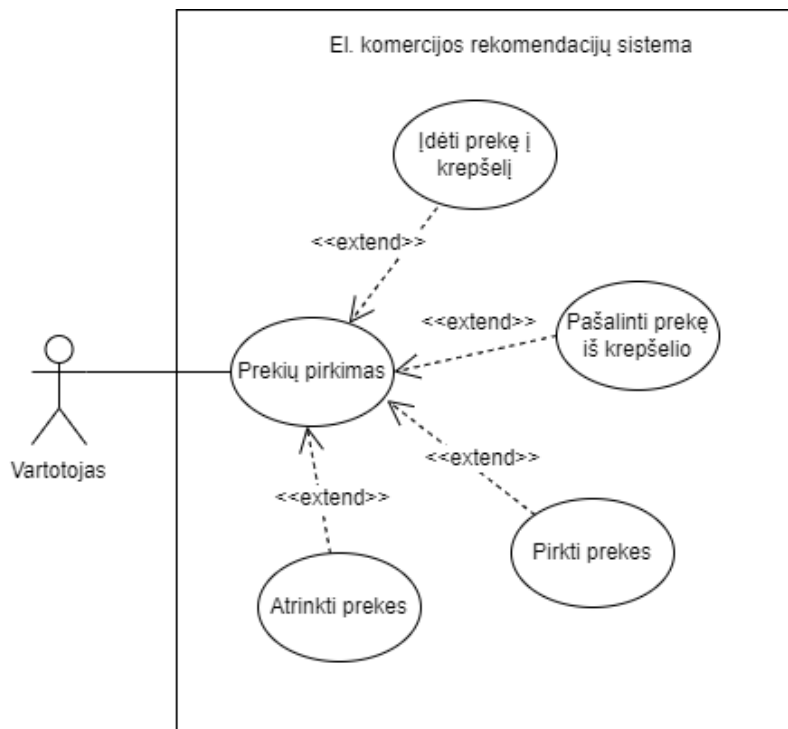
Sąveikos objektai: Vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0102.

12 lentelė. Prisijungti sekos diagramos paaiškinimas

užduoties numeris	2
užduoties pavadinimas	Prisijungti
siekiamas tikslas	Prisijungti prie sistemos
užduoties vykdymą inicijuojantis įvykis (trigeris)	Vartotojas pasirenka atlikti veiksmą su paskyrą
užduoties prioritetas	1-as (auštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė, grafinė vartotojo sąsaja
"prieš" sąlygos	Sėkmingai pasiekta internetinė svetainė
sėkmingos baigties "po" sąlygos	Sėkminga pasiekta internetinės svetainės dalis skirta prisijungusiam vartotojui
nesėkmingos baigties "po" sąlygos	Nepasiekta internetinės svetainės dalis skirta prisijungusiam vartotojui

4.2.3. Prekių pirkimo užduotis



22 pav. Prekių pirkimas užduotis

Užduotis: Prekių pirkimas.

Pagrindinis aktorius: Vartotojas.

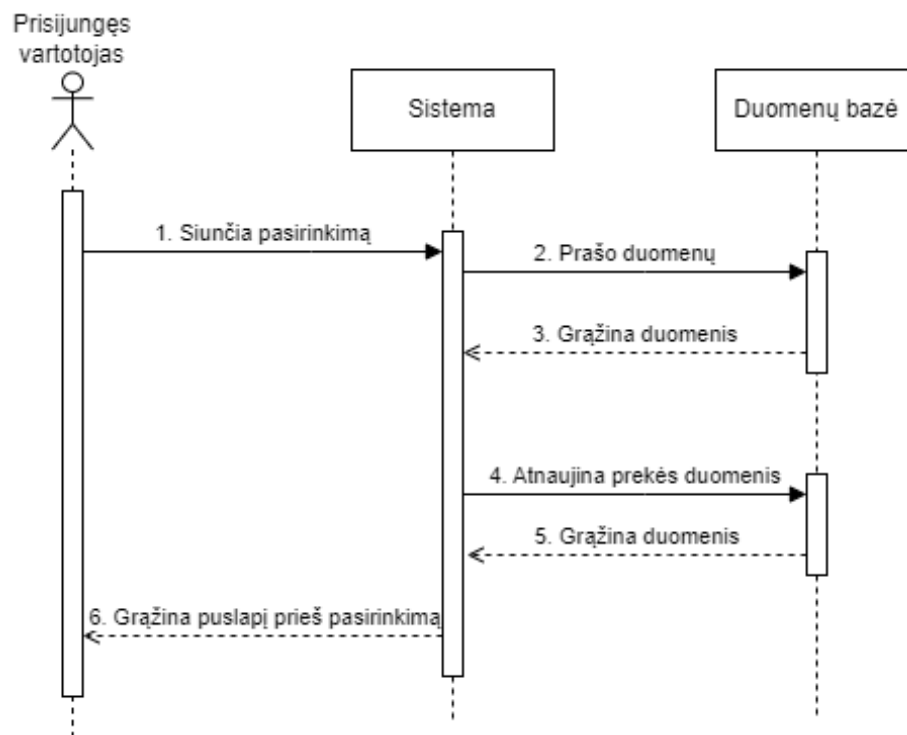
Apimtis: El. komercijos rekomendacijų sistema.

Aprašymas: Vartotojas gali įdėti prekę į krepšelį, pašalinti prekę iš krepšelio, pirkti prekes ir atrinkti prekes.

Trigeris: Vartotojas pasirenka atlikti veiksmą su produktu.

Veiksmų seka:

- 1.Siunčiami duomenys sistemai.
- 2.Sistema siunčia užklausą iš duomenų bazę ir gauna atsaką.
- 3.Sistema atrenka duomenis. Jeigu yra poreikis užrašo duomenis į duomenų bazę ir gauna atsaką.
- 4.Sistema grąžina atsaką vartotojui.



23 pav. Prekių pirkimas seka

Sekos pavadinimas: Prekių pirkimas.

Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

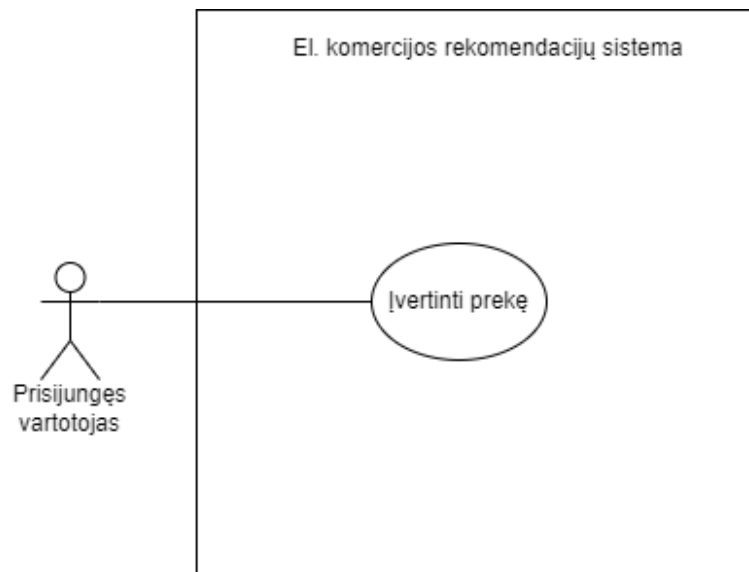
Įrėmintas funkcinis reikalavimas: FR-0202.

13 lentelė. Prekių pirkimas sekų diagramos paaiškinimas

užduoties numeris	3
užduoties pavadinimas	Prekių pirkimas
siekiamas tikslas	Pirkti prekes esančias krepšelyje
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka pirkti prekes
užduoties prioritetas	1-as (aukštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė, grafinė vartotojo sąsaja
"prieš" sąlygos	Sėkmingai pasiekta internetinės svetainės puslapis atvaizduojantis prekių krepšelį
sėkmingos baigties "po" sąlygos	Atnaujinama informacija ir vartotojas lieka tame pačiame puslapyje
nesėkmingos baigties "po" sąlygos	Neatnaujinama informacija ir vartotojas lieka tame pačiame puslapyje

Kitos panašios sekų diagramos yra aprašytos prieduose (PRIEDAS 4).

4.2.4. Įvertinti prekę užduotis



22 pav. Įvertinti prekę užduotis

Užduotis: Įvertinti prekę.

Pagrindinis aktorius: Neprisijungęs vartotojas.

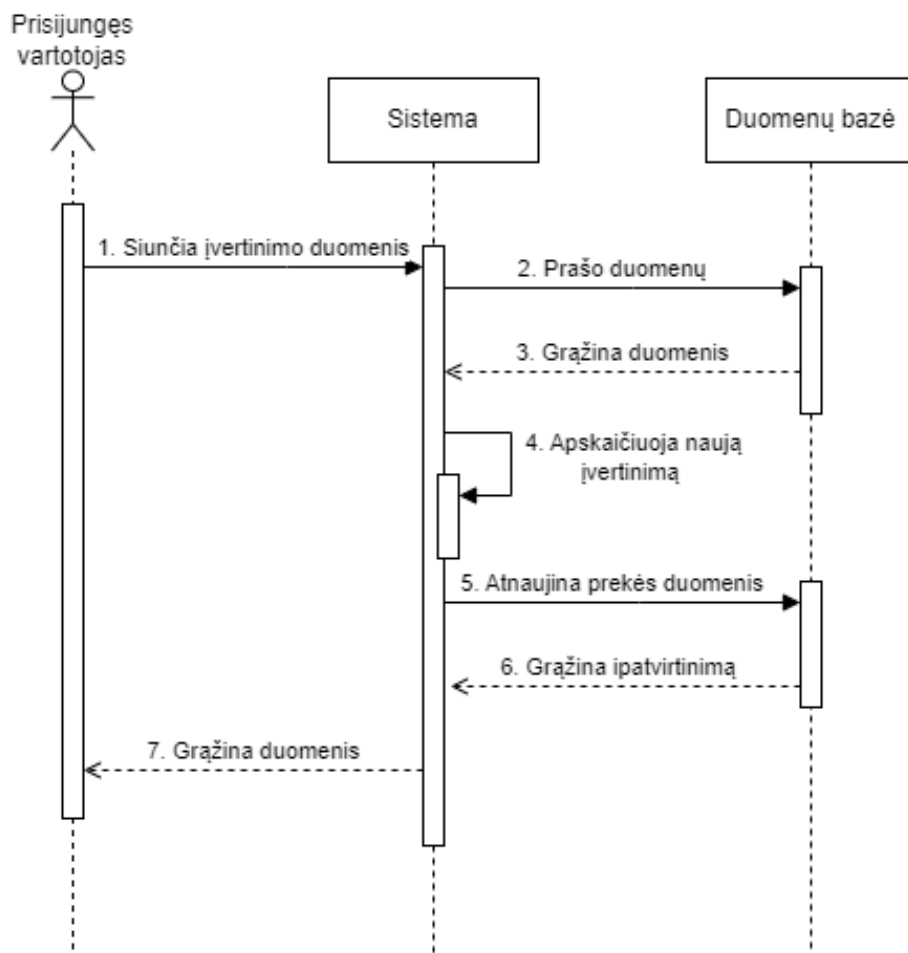
Apimtis: El. komercijos rekomendacijų sistema.

Aprašymas: Prisijungęs vartotojas gali įvertinti prekę, kurią jis yra nusipirkęs anksčiau.

Trigeris: Prisijungęs vartotojas pasirenka veiksmą įvertinti prekę.

Veiksmų seka:

- 1.Siunčiami duomenys sistemai.
- 2.Sistema gauna duomenis iš duomenų bazės.
- 3.Sistema apskaičiuoja duomenis.
- 4.Sistema atnaujina duomenų bazės duomenis.
- 5.Gražinamas atsakas prisijungusiam vartotojui.



23 pav. Įvertinti prekę seka

Sekos pavadinimas: Įvertinti prekę.

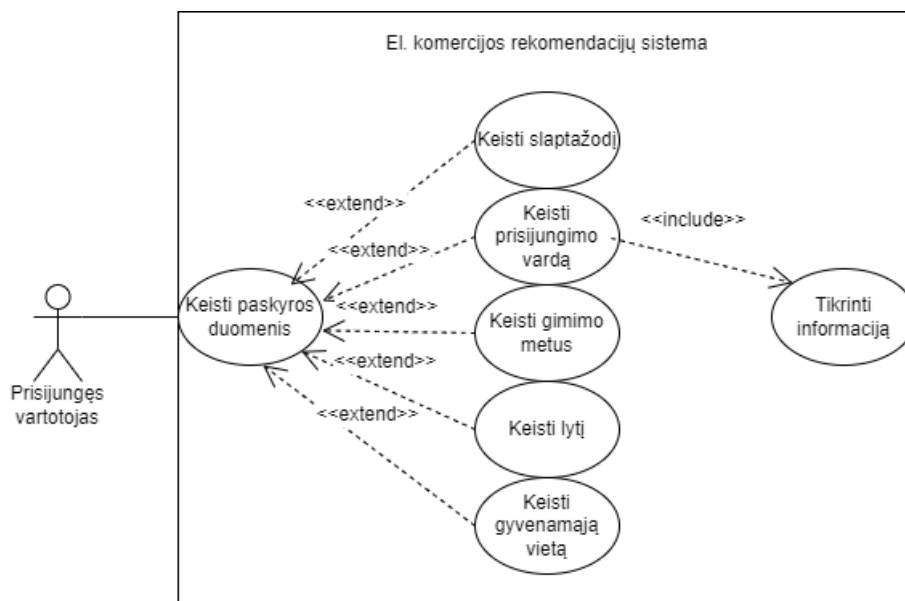
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0201.

14 lentelė. Įvertinti prekę sekų diagramos paaiškinimas

užduoties numeris	4
užduoties pavadinimas	Įvertinti prekę
siekiamas tikslas	Įvertinti prekę, kurią prisijungęs vartotojas yra pirkęs anksčiau
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka įvertinti prekę
užduoties prioritetas	3-as (žemas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Prisijungęs vartotojas
antriniai aktoriai	Sistema, duomenų bazė, grafinė vartotojo sąsaja
"prieš" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis atvaizduojantis pirktas prekes
sėkmingos baigties "po" sąlygos	Sėkmingai atnaujintas internetinės svetainės puslapis atvaizduojantis pirktas prekes
nesėkmingos baigties "po" sąlygos	Nesėkmingai atnaujintas internetinės svetainės puslapis skirtas prisijungusiam vartotojui

4.2.5. Paskyros duomenų keitimo užduotis



22 pav. Keisti paskyros duomenis užduotis

Užduotis: Keisti paskyros duomenis.

Pagrindinis aktorius: Prisijungęs vartotojas.

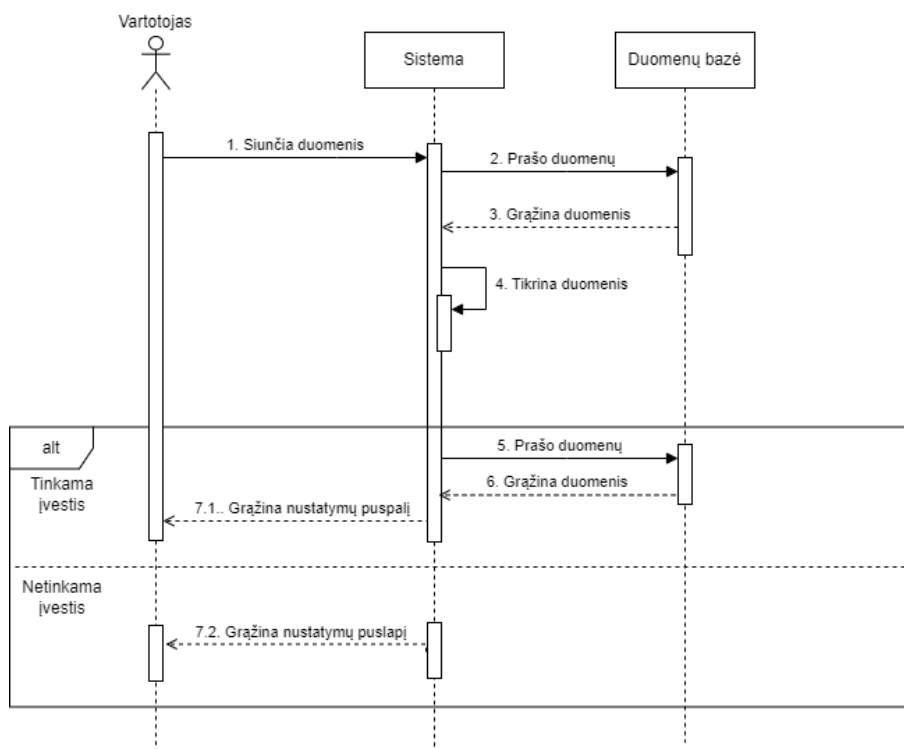
Apimtis: El. komercijos rekomendacijų sistema.

Aprašymas: Prisijungęs vartotojas gali keisti slaptažodį, keisti prisijungimo vardą, keisti gimimo metus, keisti lytį ir keisti gyvenamąją vietą.

Trigeris: Prisijungęs vartotojas pasirenka veiksmą su paskyra.

Veiksmų seka:

- 1.Siunčiami duomenys sistemai.
- 2.Jeigu reikia patikrinti duomenis sistema gauna duomenis iš duomenų bazės ir sistemos viduje juos patikrina.
- 3.Jeigu nereikia tikrinti duomenų arba duomenys buvo patikrinti ir atitiko reikalavimus, duomenys atnaujinami arba įrašomi duomenų bazėje.
- 4.Siunčiamas atsakas prisijungusiam vartotojui.



23 pav. Keisti prisijungimo vardą seka

Sekos pavadinimas: Keisti prisijungimo vardą.

Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

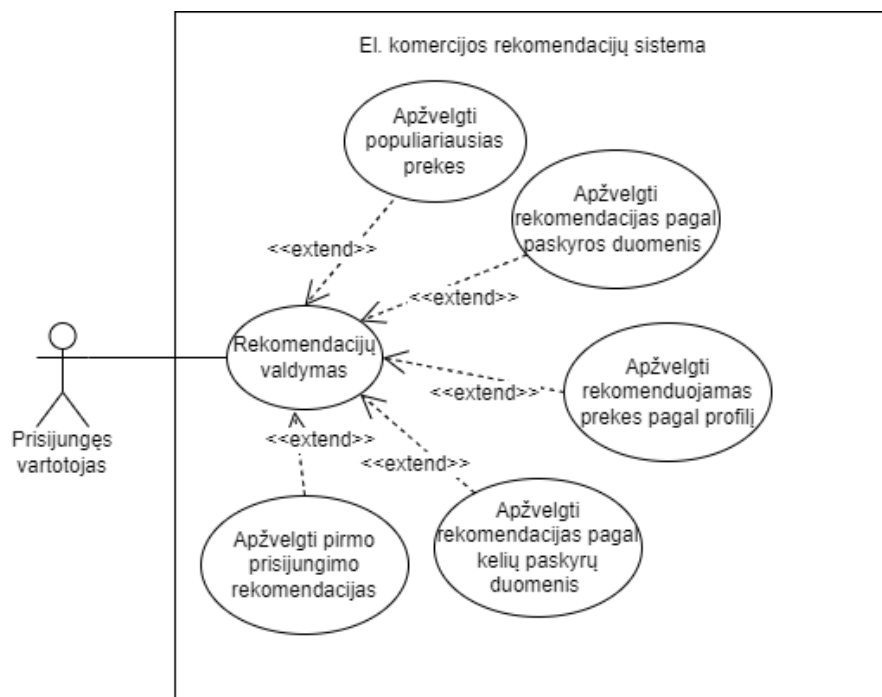
Įrėmintas funkcinis reikalavimas: FR-0203.

15 lentelė. Keisti prisijungimo vardą sekų diagramos paaiškinimas

užduoties numeris	5
užduoties pavadinimas	Keisti prisijungimo vardą
siekiamas tikslas	Keisti paskyros prisijungimo
užduoties vykdymą inicijuojantis įvykis (triggeris)	Prisijungęs vartotojas pasirenka keisti paskyros duomenis
užduoties prioritetas	3-as (žemas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Prisijungęs vartotojas
antriniai aktoriai	Sistema, duomenų bazė, grafinė vartotojo sąsaja
"prieš" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis skirtas keisti paskyros duomenis
sėkmingos baigties "po" sąlygos	Sėkmingai atnaujintas internetinės svetainės puslapis skirtas keisti paskyros duomenis
nesėkmingos baigties "po" sąlygos	Nesėkmingai atnaujintas internetinės svetainės puslapis skirtas keisti paskyros duomenis

Kitos panašios sekų diagramos yra aprašytos prieduose (PRIEDAS 4).

4.2.6. Rekomendacijų valdymo užduotis



22 pav. Rekomendacijų valdymas užduotis

Užduotis: Rekomendacijų valdymo užduotis.

Pagrindinis aktorius: Prisijungęs vartotojas.

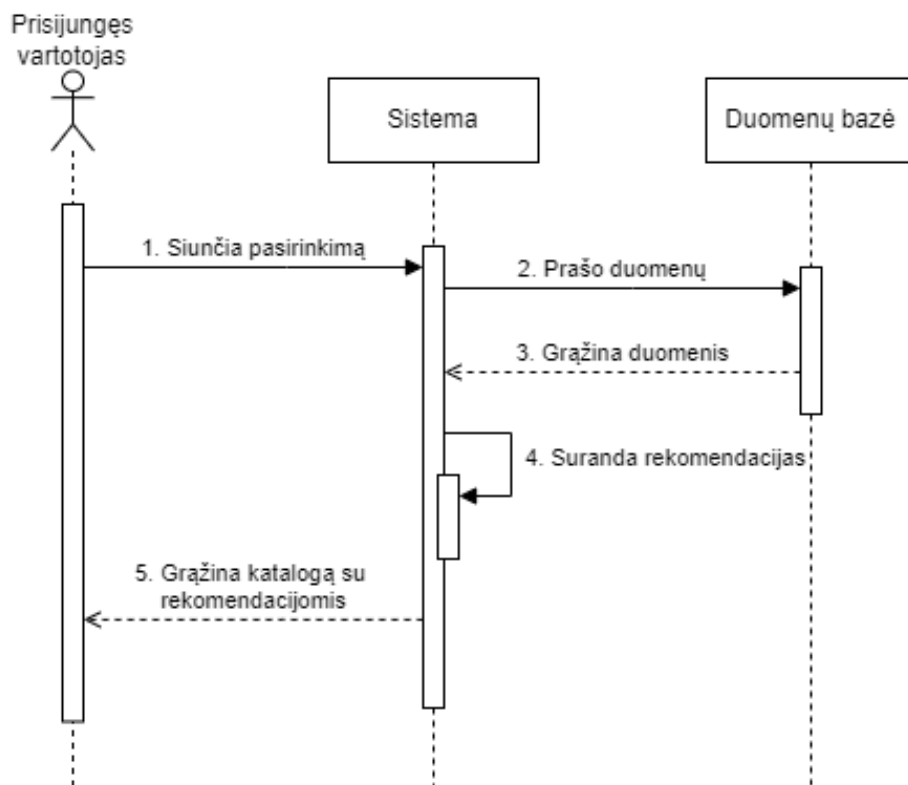
Apimtis: El. komercijos rekomendacijų sistema.

Aprašymas: Prisijungęs vartotojas gali apžvelgti populiariausias prekes, apžvelgti rekomendacijas pagal paskyros duomenis, apžvelgti rekomenduojamas prekes pagal profilį ir apžvelgti rekomendacijas pagal kelių paskyrų duomenis.

Trigeris: Prisijungęs vartotojas pasirenką veiksmą su rekomendacijomis.

Veiksmų seka:

- 1.Siunčiami duomenys sistemai.
- 2.Sistema gauna duomenis iš duomenų bazės.
- 3.Sistema apskaičiuoja ir atrenka duomenis.
- 4.Sistema grąžina atsaką prisijungusiam vartotojui.



23 pav. Apžvelgti rekomendacijas pagal kelių profilių duomenis seka

Sekos pavadinimas: Apžvelgti rekomendacijas pagal kelių profilių duomenis.

Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0303

16 lentelė. Apžvelgti rekomendacijas pagal kelių profilių duomenis sekų diagramos paaiškinimas

užduoties numeris	6
užduoties pavadinimas	Apžvelgti rekomendacijas pagal kelių profilių duomenis seka
siekiamas tikslas	Apžvelgti rekomendacijas pagal kelių paskyrų duomenis.
užduoties vykdymą inicijuojantis įvykis (trigeris)	Prisijungęs vartotojas pasirenka apžvelgti rekomendacijas pagal kelių profilių duomenis
užduoties prioritetas	1-as (aukštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Prisijungęs vartotojas
antriniai aktoriai	Sistema, duomenų bazė, grafinė vartotojo sąsaja
"prieš" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis turinti nuorodas į rekomendacijų puslapius
sėkmingos baigties "po" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas
nesėkmingos baigties "po" sąlygos	Nesėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas

Kitos panašios sekų diagramos yra aprašytos prieduose (PRIEDAS 4).

4.3. Pagrindinių sistemos užduočių apibendrinimas

Išskaidžius bendrą sistemos užduočių diagramą į mažesnes dalis galima išvelgti sistemos veikimo detales. Žinant šias detales yra lengva rasti sistemos trūkumus. Geros užduotys atsako į sistemos funkcinis reikalavimus.

Skirtingų sistemos vartotojų užduotis ir privilegijas galima paskirstyti pagal vaidmenis. Sistemoje dalyvauja paprasti vartotojai ir registruoti vartotojai. Išskirsčius vartotojus pagal vaidmenis galima suprojektuoti išmanesnę sistemą, primenančią šiuo metu el. komercijoje pritaikytas sistemas.

17 lentelė. Naudotojų užduočių pasiskirstymas pagal vaidmenis

Nr.	Užduoties pavadinimas	Registruotas vartotojas, Administratorius	Neregistruotas vartotojas
1.	Užsiregistruoti sistemoje	-	+
2.	Prisijungti prie sistemos	+	-
3.	Atsijungti nuo sistemos	+	-
4.	Peržiūrėti vartotojo profilį	+	-
5.	Keisti vartotojo profilio duomenis	+	-
6.	Filtruoti prekių katalogą pagal kategoriją	+	+
7.	Peržiūrėti perkamiausias prekes	+	-
8.	Gauti rekomendacijas pagal profilį	+	-
9.	Gauti personalizuotas rekomendacijas	+	-
10.	Peržiūrėti pirkinių krepšelį	+	+
11.	Peržiūrėti pirktas prekes	+	-
12.	Įvertinti pirktas prekes	+	-
13.	Peržiūrėti vartotojų sąrašą	+	-
14.	Pridėti vartotoją į asmeninį sąrašą	+	-

Naudotojų pasiskirstymo lentelė yra naudinga siekiant pastebėti skirtumus tarp skirtingo tipo aktorių. Sistema, kuri turi limitus pagal aktoriaus tipą labiau primena praktiškai pritaikytas el. komercijos sistemas.

CRUD lentelėje galima matyti skirtingo tipo vartotojų teises vykdančias užduotis. Ši lentelė yra skirta suprasti kaip skirtingo tipo aktoriai sąveikauja su duomenimis, kuriuos pritaiko sistema. Lentelės žymėjimas: C – kurti, R – skaityti, U – keisti, D – trinti/naikinti.

18 lentelė. CRUD matrica

Užduoties Nr.	Vartotojo rolė		Registruotas vartotojas, Administratorius	Neregistruotas vartotojas
	Duomenys			
D1	Naudotojo duomenis		RUD	C
D2	Produktų duomenis		RU	R

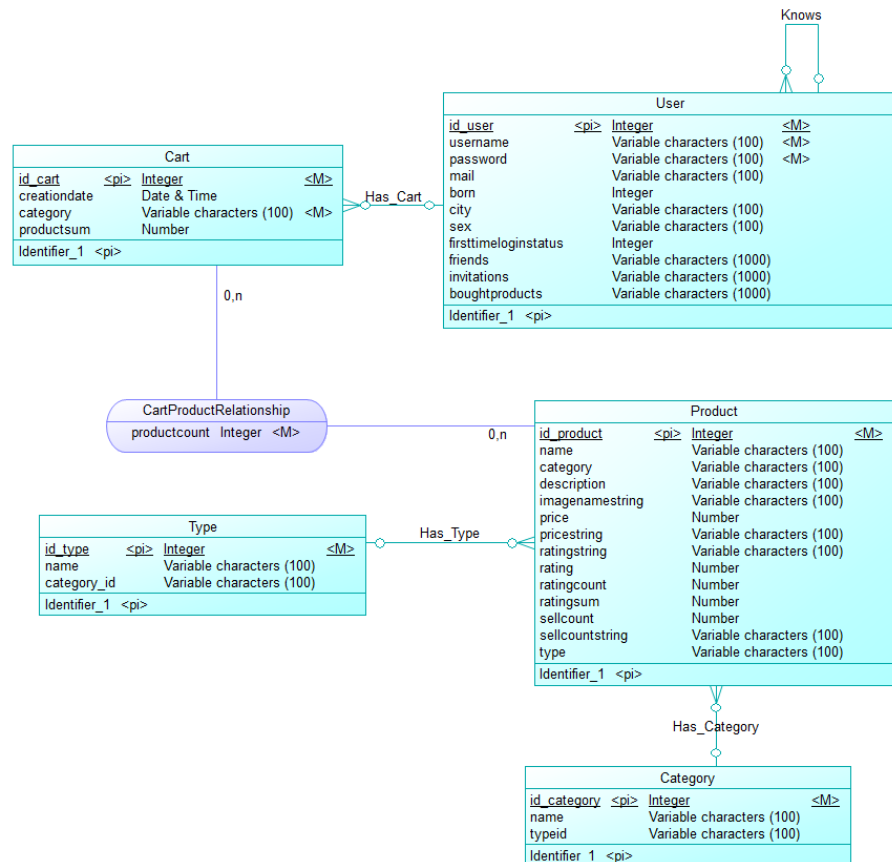
Papildomos lentelės apibūdinančios skirtingo tipo vartotojų įgaliojimus ir užduočių paskirstymą padeda geriau suprasti kuo skiriasi sistemoje esantys dalyviai. Skirtingiems vartotojams pasiekiamos skirtingos užduotys, be to, yra ir skirtingas poveikis duomenų bazei.

Užduočių diagramų pagalba lengviau numatyti kokią logiką reikės suprogramuoti sėkmingam reikalavimų įgyvendinimui. Aiškiau suprantamos pagrindinės sistemos užduotys. Užduočių diagramos sukuria papildomą sluoksnį tarp sistemos įgyvendinimo ir idėjos. Šio sluoksnio pagalba programuotojui lengviau suprasti kokie aktoriai ir kaip sąveikauja su sistema. Žinant sąveiką, lengviau kurti patvaresnę ir konkretesnę sistemą. CRUD ir naudotojų užduočių paskirstymo pagal vaidmenis lentelės panaudotos įvertinti skirtingus sistemos dalyvius. Sekų diagramos kaip ir užduočių diagramos yra puiki priemonė. Ši priemonė padeda sukurti dinamiškesnę užduoties supratimą. Šios priemonės pagalba numatoma elgsena. Sekų diagramos neblogai dekomponuoja sistemą. Sistemos dekomponavimas padeda geriau suprasti kaip veikia atskiros sistemos dalys. Kaip ir su užduočių diagramomis taip ir su sekų diagramomis yra siekiama atsakyti į iškeltus funkcinius reikalavimus. Taigi, pagrindinės sistemos užduotys yra labai patogu apibūdinti užduočių ir sekų diagramomis.

4.4. Klasių struktūra

Pagrindinė klasių diagrama yra skirta suprasti pagrindinius sistemos objektus, kaip jie sąveikauja ir papildo vienas kitą. Klasių diagramoje nubraižytos klasės, kurios atspindi projekto duomenų bazės modelį. Tai yra esybės turinčios svarbius atributus, reikalingus sistemos tikslų įgyvendinimui.

cart), tačiau pirminių vežimėlis gali priklausyti tik vienam vartotojui (angl. user). Pritaikyta asocijacija skirta išsaugoti prekių kiekius. Asocijacija padeda suprasti kiek tos pačios prekės vienetų yra vežimėlyje. Tipo ir kategorijos esybės yra reikalingos sistemos apskaičiavimams atlikti.



6 pav. Konceptinis duomenų bazės modelis

Šis modelis reikalingas norint turėti pagrindą. Su turimu pagrindu galima toliau plėtoti el. komercijos sistemą. Esių pakanka norint išpildyti jau esančius funkcinis reikalavimus, tačiau nėra pasiekama optimalumo ir funkcijų gausumo bei protingo išskaidymo. Nėra numatoma daug kraštutinių atvejų. Plačiai praktikoje naudojami el. komercijos koncepciniai modeliai yra daug labiau komplikuoti.

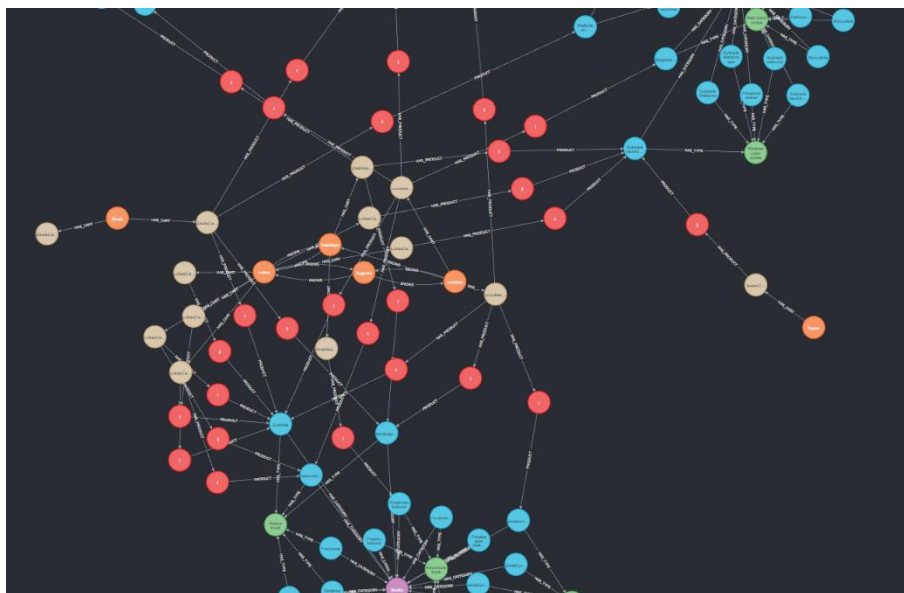
Koncepciniame duomenų bazės modelyje yra 6 esybės:

- *User* – sistemoje užsiregistruoto vartotojo esybė, kurioje bus saugomi tokie atributai kaip: vartotojo vardas, el. paštas, slaptažodis, amžius, gyvenamasis miestas, lytis. Šioje esybėje saugojami atributai reikalingi įgyvendinti pagrindinį sistemos veikimą. Išplėstiniam sistemos veikimui reikalingi tokie atributai: pirkti produktai, pirmo prisijungimo statusas, draugai, pakvietimai.
- *Product* – sistemoje įrašyto produkto esybė. Saugomi atributai reikalingi prekės apibūdinimui. Šioje esybėje saugomi tokie pagrindo atributai kaip: kategorija, aprašymas, paveikslėlio nuorodos tekstas, vardas, kaina, kainos tekstas, įvertinimas, įvertinimo tekstas,

tipas. Taip pat yra ir atributų reikalingų išplėstiniam sistemos veikimui: įvertinimų kiekis, pardavimų kiekis, įvertinimų suma.

- *Cart* – esybė ryšiais apibūdinanti kokias prekes krepšelyje turėjo vartotojas. Atributai naudojami išskirti krepšelį nuo kitų krepšelių.
- *CartItem* – tarpinė esybė, kurios pagalba sužinoti kiek tos pačios prekės vartotojas pirko.
- *Category* – esybė apibūdinanti kokiai kategorijai priklauso prekė. Labai daug atributų joje nėra, bet ryšių pagalba susiejama su produktais.
- *Type* – esybė apibūdinanti kokiam tipui priklauso prekė. Daug atributų neturi tačiau ryšiais susiejama su produktais.

Neo4j grafų duomenų bazėje šios esybės yra gražiai pavaizduojamos mazgų pavidalu. Mazgai susieti ryšiais siekiant įgyvendinti sistemos duomenų modelį.

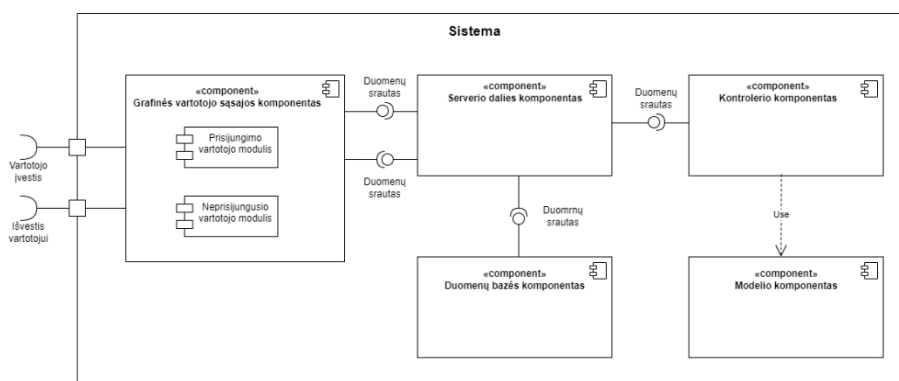


7 pav. Neo4j sukurta duomenų vizualizacija

Šioje duomenų vizualizacijoje atvaizduojami pagrindiniai duomenys. Svarbiausi duomenys yra vartotojai ir pirkiniai krepšeliai. Atvaizduojami oranžiniais ir šviesiai rudos spalvos burbuliukais.

4.6. Sistemos architektūra

Projektuojama sistema yra sudaryta iš daug dalių. Šias dalis galima pavadinti komponentas. Komponentai detalizuojami pasitelkiant modulius.



8 pav. Komponentų diagrama

Sistemos architektūros centre yra kontrolerio komponentas. Šis komponentas atsakingas už sistemos veikimą, logiką ir kitų komponentų integraciją. Kontroleris naudoja modelio ir grafinės vartotojo sąsajos komponentus. Pagal modelio komponentą nustatoma kokia forma naudojami duomenys, o su grafinės vartotojo sąsajos komponento nustatoma kokios struktūros puslapiai turėtų būti gražinami galutiniam vartotojui. Kontrolerio komponentas sąveikauja su serverio dalies komponentu. Šis serveris yra Apache Tomcat. Aktyvuoja duomenų srautus tarp internetinės naršyklės ir sistemos. Serverio komponentas sąveikauja su duomenų bazės komponentu.

Sistemos architektūra yra panaudojama tiksliau apibūdinti sistemos vidinę struktūrą. Bandyta pritaikyti MVC (angl. Model-View-Controller) projektavimo architektūrą.

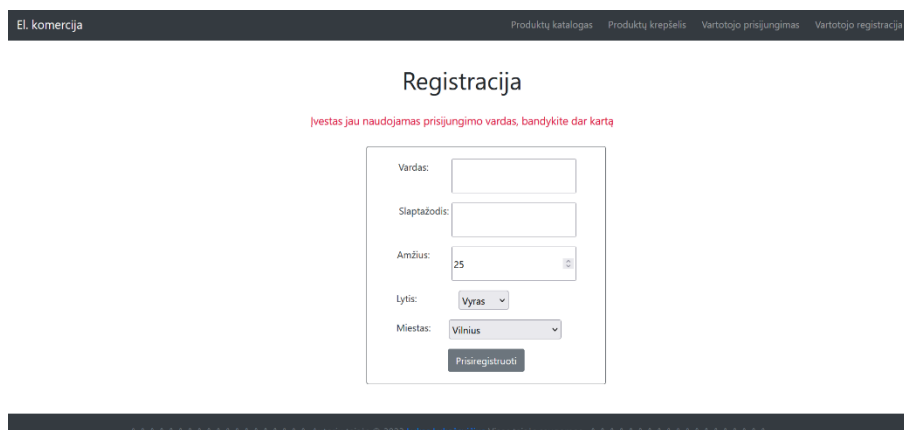
4.7. Sistemos įgyvendinimas

Suprogramuotoje sistemoje buvo įgyvendinti funkciniai ir nefunkciniai reikalavimai. Šiame skyriuje pateikiamas sistemos realizavimas ir pavyzdžiai iš realizuotos el. komercijos rekomendacijų sistemos, kuri duomenų skaitymui ir rašymui pasitelkia grafų duomenų bazę.

Sistemos kūrimui buvo pritaikyta IntelliJ integruota programavimo aplinka skirta kurti projektus Java programavimo kalba ir Neo4j grafų duomenų bazė. Siekiant įgyvendinti visus funkcionalumus buvo naudojami papildiniai. Svarbiausi papildiniai panaudoti sistemos funkcijų įgyvendinimui buvo Thymeleaf Java html šablonai duomenų įterpimui į internetinius puslapius ir Spring-Boot Java papildinys serverio ir sąsajos su duomenų baze kūrimui. Pirmiausiai buvo sukurtas neprisijungusio vartotojo funkcionalumas, tada registracija, kiek vėliau funkcionalumas susijęs su produktais ir galiausiai buvo kuriama protinga sąsaja su duomenų baze. Eigoje buvo kuriamas personalizuotos rekomendacijos. Iš pradžių pagal vartotojo registracijos profilį, kiek vėliau pagal tai ką vartotojas pirko anksčiau. Naudotojas duomenis apie prekes gauna iš serverio, kuris duomenis gauna iš duomenų bazės naudojant užklausas Cypher kalba. Funkcijos sugeneruoja užklausas duomenų bazei pagal Spring-Boot šabloną. Šios užklausos reikalingos duomenų skaitymui ir rašymui į duomenų bazę.

Duomenų mainai įgyvendinami per du serverius. Vienas serveris įgalioja duomenų bazę, o kitas sukuria kelią duomenis keliauti tarp sistemos, duomenų bazės ir grafinės vartotojo sąsajos, kuri kuriamos sistemos atveju yra internetinė svetainė. Serverių ekrano nuotraukos yra prieduose (PRIEDAS 2).

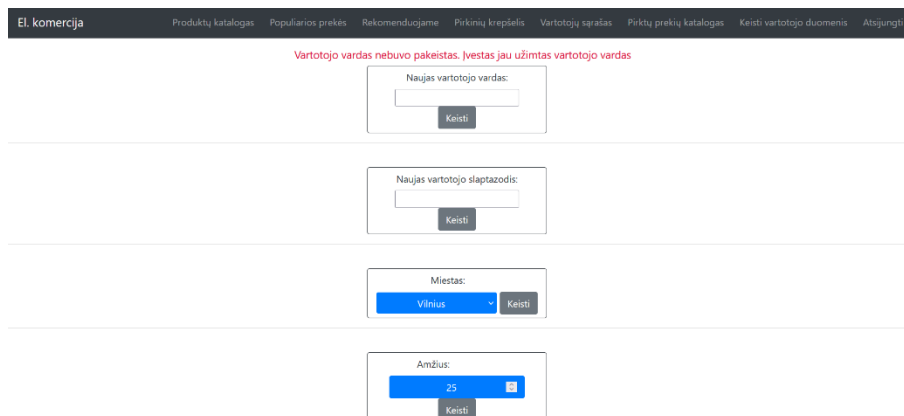
Siekiant, jog vartotojo įvesti duomenys nesukeltų nenumatytų sistemoje trykių ir atitiktų keliamus reikalavimus, įvesti duomenys yra tikrinami. Dauguma profilio duomenų: miestas, amžius, lytis yra parenkami iš sąrašo, kuris sudarytas pagal numanomą vartotojų demografiją. Naujai registruojamo vartotojo įvesti duomenys yra patikrinami Java LoginController klasėje ir jeigu nėra klaidų, duomenys perduodami serveriui, kuris juos nukreipia ir įrašo į duomenų bazę. Registracija ir vartotojo duomenų keitimo puslapiai turi numatytas klaidų žinutes, kurios parodomos, kai vartotojas netinkamai suveda duomenis.



9 pav. Netinkamas duomenų įvedimas registracijos lauke

Įvedus netinkamus duomenis registracijos formoje vartotojui pateikiamas pranešimas kur padaryta klaida, jog galėtų bandyti prisiregistruoti iš naujo.

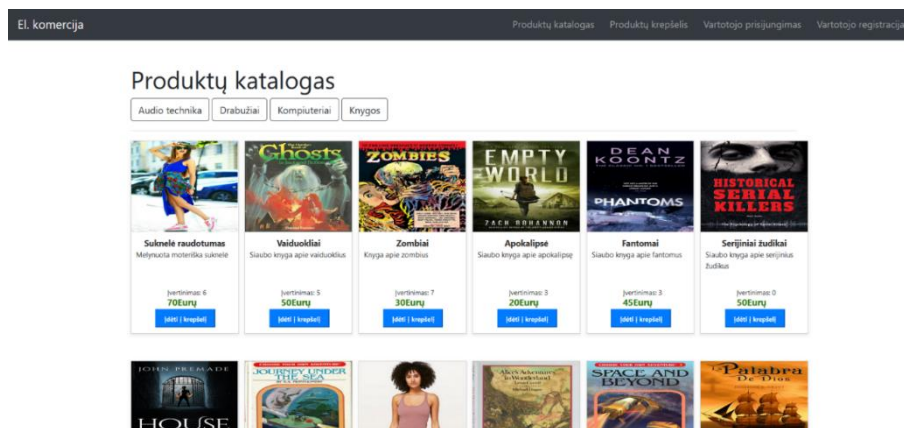
Vartotojo profilio duomenų keitimo formoje taip pat yra numatyti pranešimai netinkamos įvesties atveju.



10 pav. Netinkamas duomenų įvedimas keičiant vartotojo vardą

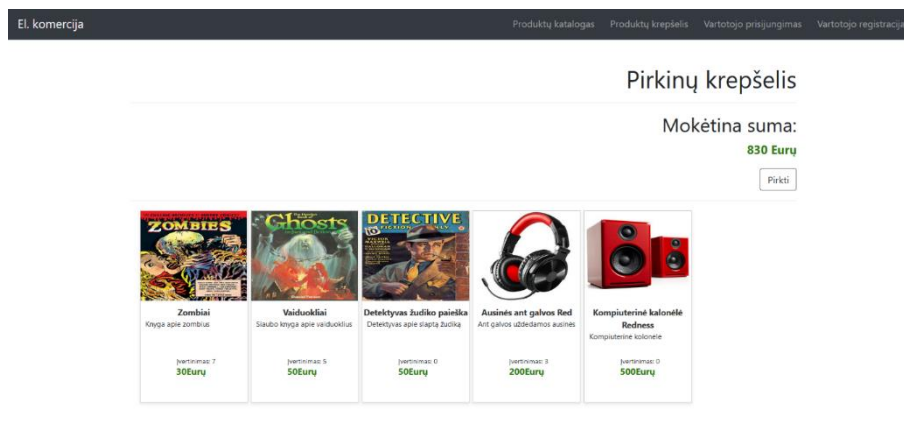
Matyti produktų katalogą, dėti prekes į krepšelį ir pirkti prekes įdėtas į krepšelį gali visi vartotojai, prisijungusieji ir neprisijungusieji. Tačiau nemaža dalimi sistemos funkcijų gali pasinaudoti tik prisiregistravę ir prisijungę vartotojai. Pirma apibūdinamas neprisijungusiam vartotojui pasiekiamas

pagrindinis funkcionalumas, nes jis gerai parodo el. komercijos svetainės funkcionavimo pagrindą. Neprisijungęs vartotojas pradiniam puslapyje mato prekių katalogą.



11 pav. Pradinis svetainės puslapis

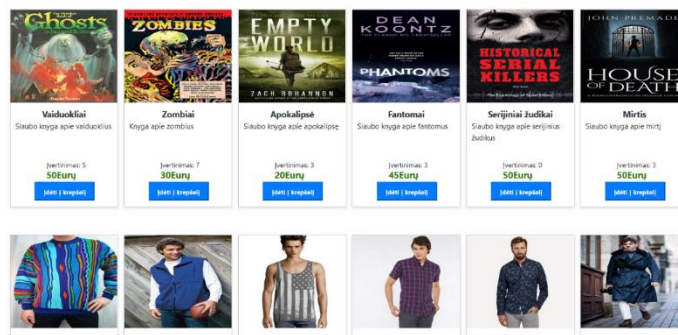
Iš pradinio puslapio neprisijungęs vartotojas gali filtruoti ir rasti prekes pagal kategoriją. Taip pat gali įdėti prekes į krepšelį. Pirkinių krepšelio puslapyje galima pamatyti apskaičiuotą visų perkamų produktų sumą, apžvelgti perkamus produktus ir juos įsigyti paspaudus pirkti mygtuką.



12 pav. Neprisijungusio vartotojo pirkinių krepšelis

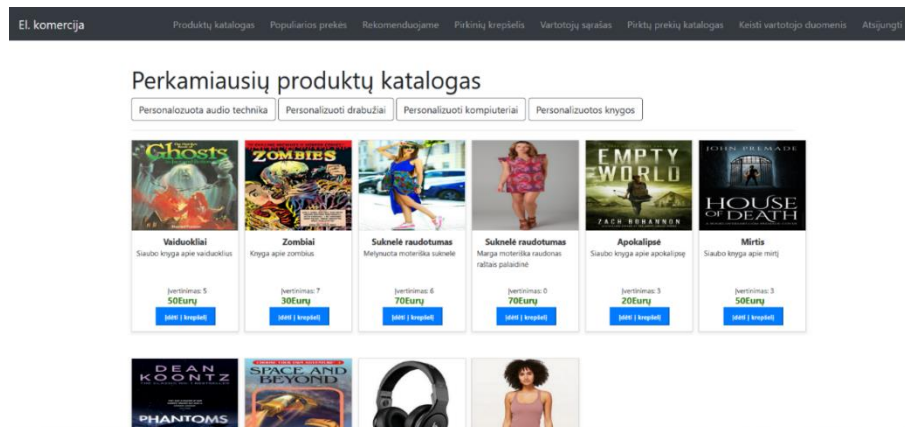
Taip pat neprisijungęs vartotojas gali pasiekti ir registracijos puslapį, kuriame teisingai užpildžius duomenis yra sukuriamą naują galiojanti paskyra, su kuria galima išnaudoti kitas sistemos galimybes. Prisijungęs vartotojas gali pasiekti katalogą, kuriame yra sugeneruojamos produktų rekomendacijos pagal prisijungusio vartotojo profilį. Neprisijungęs vartotojas gali prisijungti prie sistemos, jeigu jau turi susikūręs sau paskyrą.

Rekomendacijos šiai dienai

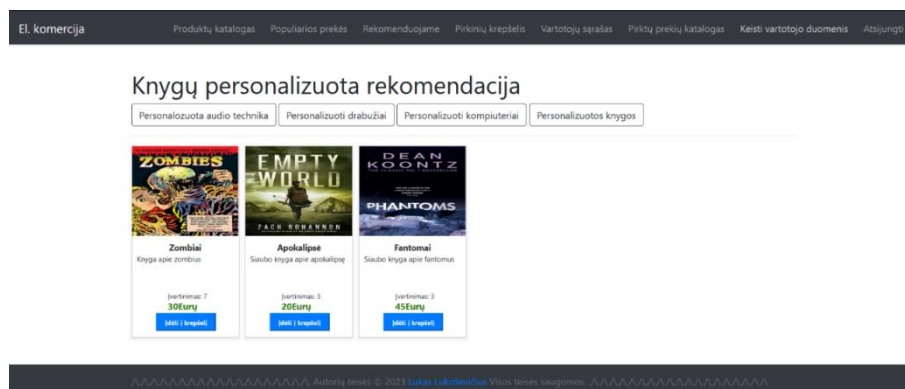


13 pav. Prisijungusio vartoto rekomendacijos pagal profilį

Prisijungęs vartotojas gali rasti ir perkamiausias prekes. Perkamiausių prekių kataloge taip pat yra ir kitaip personalizuotos prekių rekomendacijos. Šios rekomendacijos sudaromos pagal tai ką vartotojas yra pirkęs anksčiau. Jeigu vartotojas yra anksčiau pirkęs siaubo žanro knygą, jam sistema parekomenduos siaubo žanro knygas.

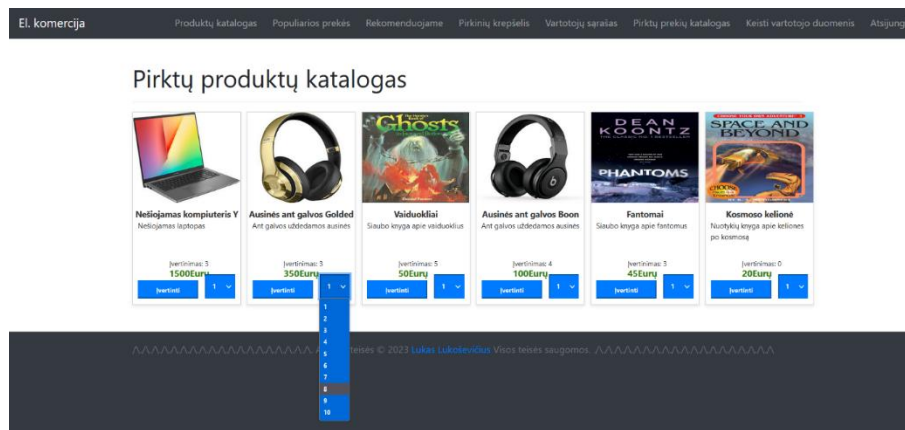


14 pav. Perkamiausių prekių katalogas



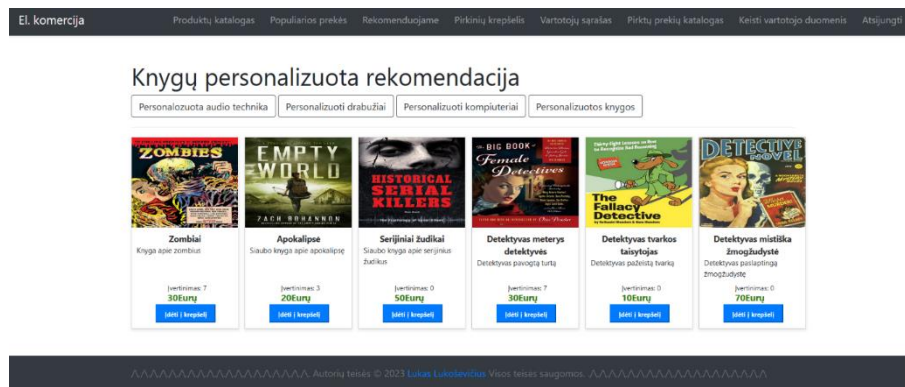
15 pav. Personalizuota rekomendacija prisijungusiam vartotojui

Prisijungęs vartotojas taip pat gali rasti ir pirktų prekių katalogą. Šiame kataloge yra parodoma kokias prekes prisijungęs vartotojas yra pirkęs anksčiau ir iš šio puslapio galima įvertinti pirktas prekes.



16 pav. Prisijungusio vartotojo pirktų prekių katalogas

Prekių įvertinimai yra išsaugomi į duomenų bazę. Prisijungęs vartotojas taip pat turi galimybę sąveikauti su kitais prisiregistravusiais vartotojas. Yra galimybė pasikviesti vartotojus į savo pakviestų vartotojų sąrašą. Yra kitokio tipo personalizuotos rekomendacijos, kurios atsižvelgia į prisijungusio vartotojo pirkimų istoriją ir kitų vartotojų pirkimo istoriją. Pagal panašumus randamos rekomendacijos.



17 pav. Papildyta personalizuota rekomendacija

Galima pastebėti, jog prisijungęs vartotojas gauna detektyvinių knygų rekomendaciją. Taip yra todėl, nes tokio tipo knygą yra pirkęs vartotojas, kuris irgi mėgsta knygas. Ši rekomendacija randama pagal numanomą vartotojo pomėgių profilį. Kiekvienas prisiregistravęs ir prisijungęs vartotojas turi prieigą prie registruotų vartotojų sąrašo. Iš šio sąrašo galima pridėti vartotoją į savo asmeninį sąrašą. Atskleidžiama informacija tik vartotojo vardas.

4.8. Programų sistemos projekto apibendrinimas

57

5. Testavimas

5.1. Testavimo eiga

Šiame skyriuje pateikiami testavimo scenarijai, kelių iš jų testavimo atvejai bei sistemos testavimo rezultatai. Testavimo scenarijai yra skirti patikrinti, ar sukurta sistema atitinka jai keliamus reikalavimus. Kiekvienas pateiktas testavimo atvejis turi unikalų numerį, reikalavimo numerį ir svarbumo būseną. Pasirinktas aprašomasis testavimo būdas. Tai yra rankinis testavimo būdas. Šis būdas buvo pasirinktas dėl savo paprastumo, nesunkaus pritaikymo ir suprantamumo. Testavimas atliekamas suvedant įvairius duomenis arba vykdant numatytas užduotis, tuo pačiu metu stebint kaip reaguoja sistema, duomenų bazė ir grafinė sąsaja. Svarbumo būsenos:

- **Kritinis** – kai testuojamas reikalavimas turi įtakos visos sistemos veikimui ir privalo būti tenkinamas.
- **Aukštas** – kai testuojamas reikalavimas yra susijęs su pagrindiniu funkcionalumu ir privalo būti tenkinamas.
- **Vidutinis** – kai testuojamas reikalavimas yra reikalingas užtikrinti, kad būtų patogų naudotis sistema, todėl scenarijus turi būti tenkinamas.
- **Mažas** – kai testuojamas reikalavimas apibrėžia duomenų pateikimo formatą, stilių ar kitą funkciją, kuri neturi įtakos sistemos veikimui ar pagrindiniam funkcionalumui.

19 lentelė. Testavimo scenarijų aprašymas

Scenarijaus numeris	Reikalavimo numeris	Aprašymas	Svarba
TS-01	NFR-0201	Patikrinti ar naudotojo sąsaja pasiekama per Chrome, Firefox ir Edge naršykles	Kritinė
TS-02	NFR-0303	Patikrinti ar duomenų bazėje saugomi duomenys, kurie įvedami arba keičiami, naudojantis svetainę	Kritinė
TS-03	FR-0101	Patikrinti ar sistema leidžia prisiregistruoti prie sistemos.	Aukšta
TS-04	FR-0102	Patikrinti ar sistema leidžia vartotojui prisijungti	Aukšta

TS-05	NFR-0205	Patikrinti ar sistema leidžia prisijungusiam vartotojui atsijungti ir pasiekti sąsają skirtą neprisijungusiam vartotojui	Aukšta
TS-06	FR-0203	Patikrinti ar sistema leidžia prisijungusiam vartotojui pakeisti prisijungimo vardą	Vidutinė
TS-07	FR-0301	Patikrinti ar populiarių prekių puslapyje rodomos prekės, kurios buvo parduotos daugiausiai kartų	Vidutinė
TS-08	FR-0104	Patikrinti ar vartotojas gali įdėti prekes į pirkinių krepšelį	Vidutinė
TS-09	FR-0105	Patikrinti ar vartotojas gali pašalinti prekę iš pirkinių krepšelio	Aukšta
TS-10	FR-0304	Patikrinti ar prisijungęs vartotojas mato rekomendacijas pagal registracijos profilį.	Aukšta
TS-11	FR-0103	Patikrinti ar vartotojas gali matyti pasirinktos kategorijos prekes	Vidutinė
TS-12	FR-0202	Patikrinti ar sistema leidžia vartotojui nusipirkti prekes	Aukštas
TS-14	FR-0201	Patikrinti ar sistema leidžia prisijungusiam vartotojo įvertinti anksčiau pirktą prekę	Maža
TS-15	FR-0305	Patikrinti ar naujai prisiregistravęs ir pirmą kartą prisijungęs vartotojas mato kitokį produktų sąrašą	Vidutinis
TS-16	FR-0302	Patikrinti ar prisijungęs vartotojas mato rekomendacijas pagal tai ką jis yra pirkęs anksčiau	Vidutinis
TS-17	NFR-0205	Patikrinti ar naujai prisiregistruojančiam vartotojui parodomas klaidos pranešimas, jeigu įvesti netinkami duomenys	Vidutinis

TS-18	NFR-0101	Patikrinti ar vartotojo įvesti duomenys iš registracijos puslapio yra išsaugomi duomenų bazėje	Didelis
TS-19	NFR-0202	Patikrinti ar vartotojas internetinėje svetainėje gali sėkmingai panaudoti visas naršymo juostos nuorodas	Vidutinis

Turint aprašytus testavimo scenarijus ir jų svarbą bei priskyrimą kokius funkcinis reikalavimus jie patikrina galima sudaryti testavimo atvejus. Šie testavimo atvejai yra testavimo scenarijų užpildymas ir aprašymas kaip atsakė sistema. Kiekvienas testavimo atvejis turi savo testavimo atvejo numerį, prieš sąlygą, testavimo eigą, tikėtiną rezultatą ir statusą ar rezultatas buvo pasiektas.

20 lentelė. Testavimo scenarijų atvejai

TS Nr. Tikimo kriterijus				
TA Nr.	Prieš sąlyga	Testavimo eiga	Tikėtinas rezultatas	Rezultatas
TS-01. Patikrinti ar naudotojo sąsaja pateikiama per naršyklę Į tinklalapio Chrome arba Firefox naršyklės paieškos juostą įvedus svetainės URL nuorodą pakraunama svetainė				
TA-01	Paleista interneto naršyklė	1. Pasiekama svetainė suvedus localhost:8080 į naršyklės paieškos juostą	Nukreipiama į svetainės namų puslapį	Taip
TS-02. Patikrinti ar duomenų bazėje saugomi duomenys, kurie įvedami arba keičiami, naudojantis svetaine Patikrinus duomenų bazę turėtų būti naujas egzistuojantis įrašas arba pakeistas įrašas				
TA-02	Paleista internetinė svetainė ir atidarytas registracijos puslapis	1. Suvedami duomenys į vardo, slaptažodžio ir el. pašto laukelius. Pasirenkama kita aktuali informacija	Duomenų bazėje randamas naujas įrašas sukūrus naują paskyrą ir matomi pakeitimai kai	Taip

		2. Paspaudžiamas mygtukas registruotis	paskyros duomenys pakeisti	
		3. Duomenų bazėje pasiekiamas mazgas, kurio pavadinimas yra naujai prisijungusio vartotojo vardas		
<p>TS-03. Patikrinti ar sistema leidžia prisiregistruoti prie sistemos</p> <p>Teisingai suvedus duomenis vartotojas nukreipiamas į naują namų puslapį arba gauna klaidos pranešimą</p>				
TA-03	Paleista internetinė naršyklė ir atidarytas registracijos puslapis	1. Suvedami duomenys į vardo, slaptažodžio ir el. pašto laukelius. Pasirenkama kita aktuali informacija 2. Paspaudžiamas mygtukas registruotis	Teisingos įvesties atveju vartotojas yra nukreipiamas namų puslapį skirtą prisiregistravusiam vartotojui	Taip
TA-04	Paleista internetinė naršyklė ir atidarytas registracijos puslapis	1. Suvedami duomenys į vardo, slaptažodžio ir el. pašto laukelius. Pasirenkama kita aktuali informacija 2. Paspaudžiamas mygtukas registruotis	Neteisingos įvesties atveju vartotojui yra parodomas pranešimas paaiškinantis kur įvesti netinkami duomenys	Taip
<p>TS-04. Patikrinti ar sistema leidžia vartotojui prisijungti</p> <p>Teisingai suvedus duomenis vartotojas nukreipiamas į naują namų puslapį arba gauna klaidos pranešimą</p>				
TA-05	Paleista internetinė naršyklė ir atidarytas prisijungimo puslapis	1. Suvedami duomenys į vardo ir slaptažodžio laukelius	Sėkmingo prisijungimo atveju vartotojas turėtų	Taip

		2. Paspaudžiamas mygtukas prisijungti	būti nukreiptas į namų puslapį	
TA-06	Paleista internetinė naršyklė ir atidarytas prisijungimo langas	1. Suvedami duomenys į vardo ir slaptažodžio laukelius 2. Paspaudžiamas mygtukas prisijungti	Nesėkmingo prisijungimo atveju vartotojas turėtų likti prisijungimo puslapyje ir turėtų būti parodytas pranešimas, jog neteisingi prisijungimo duomenys	Taip
<p>TS-05 Patikrinti ar prisijungęs vartotojas gali atsijungti nuo savo paskyros ir naudotis sąsaja skirta neprisijungusiam vartotojui</p> <p>Paspaudus atsijungimo nuorodą vartotojui užkraunama sąsaja skirta neprisijungusiam vartotojui</p>				
TA-07	Paleista naršyklė ir atidarytas prisijungimo puslapis	1. Vartotojas suveda prisijungimo duomenis 2. Paspaudžia mygtuką prisijungti 3. Meniu juostoje skirtoje prisijungusiam vartotojui paspaudžiama nuoroda atsijungti	Vartotojas prisijungia prie savo paskyros ir iš sąsajos skirtos prisijungusiam vartotojui, paspaudus nuorodą atsijungti, išeina į sąsają skirtą neprisijungusiam vartotojui	Taip
<p>TS-06 Patikrinti ar sistema leidžia prisijungusiam vartotojui pakeisti prisijungimo vardą</p> <p>Tikrinamas ir keičiamas vartotojo vardas, parodomas pranešimas</p>				
TA-08	Paleista naršyklė, vartotojas	1. Vartotojas įveda naują prisijungimo vardą	Vartotojui parodomas	Taip

	prisijungęs prie savo paskyros ir įjungęs paskyros duomenų keitimo puslapį. Įvedamas tinkamas naujas vardas	2. Paspaudžiamas mygtukas keisti 3. Duomenų bazėje patikrinama ar pasikeitė įrašas priskirtas vartotojui, kuris pakeitė savo vardą	pranešimas, jog sėkmingai pakeistas prisijungimo vardas. Duomenų bazėje pakeistas įrašas	
TA-09	Paleista naršyklė, vartotojas prisijungęs prie savo paskyros ir įjungęs paskyros duomenų keitimo puslapį. Įvedamas netinkamas naujas vardas	1. Vartotojas įveda naują prisijungimo vardą 2. Paspaudžiamas mygtukas keisti 3. Įrašas duomenų bazėje priskirtas prisijungusiam vartotojui nepasikeičia	Vartotojui parodomas pranešimas jog vardas nepakeistas, duomenų bazėje joks įrašas neatnaujintas	Taip
<p>TS-07 Patikrinti ar populiarių prekių puslapyje rodomos prekės, kurios buvo parduotos daugiausiai kartų</p> <p>Svetainės skiltyje su populiariomis prekėmis parodomos daugiausiai kartų parduotos prekės</p>				
TA-11	Paleista naršyklė, vartotojas prisijungęs prie savo paskyros	1. Vartotojas paspaudžia nuorodą esančią meniu juostoje pavadinimu Populiarios prekės 2. Užkraunamas puslapis, kuriame yra 10 prekių 3. Duomenų bazėje patikrinama ar keleta atvaizduotų prekių buvo parduota daug kartų 4. Lyginamas atvaizduotų prekių pardavimų kiekis ir	Duomenų bazėje palygintas prekių pardavimų kiekis. Pagrįsta, jog atvaizduotos daugiausiai kartų parduotos prekės	Taip

		neatvaizduotų prekių pardavimų kiekis		
<p>TS-08 Patikrinti ar prisijungęs vartotojas gali įdėti prekes į pirkinių krepšelį</p> <p>Vartotojas paspaudęs mygtuką įdėti prekę į krepšelį turėtų ją matyti pirkinių krepšelyje</p>				
TA-11	Paleista naršyklė ir vartotojas atidaręs bet kuri prekių puslapį	<ol style="list-style-type: none"> 1. Pasirenkama bet kuri prekė 2. Paspaudžiamas mygtukas įdėti į krepšelį 3. Meniu juostoje pasirenkama nuoroda pirkinių krepšelis 4. Patikrinama ar atvaizduota pasirinkta prekė 5. Java sistemos kode užprogramuoti ciklą kuris atspauzdina prekių krepšelyje esantį prekių sąrašą paspaudus nuorodą į pirkinių krepšelio puslapį 	Pirkinių krepšelio puslapyje parodoma prekė įdėta į pirkinių krepšelį. Java programos konsolėje išvedamas tekstas su prekių pavadinimais	Taip
<p>TS-09 Patikrinti ar vartotojas gali pašalinti prekę iš pirkinių krepšelio</p> <p>Pirkinių krepšelio puslapyje paspaudus mygtuką pašalinti prekę turėtų būti pašalinama prekė iš sąrašo ir atnaujinamas puslapis</p>				
TA-12	Paleista naršyklė ir atidarytas pirkinių krepšelio puslapis. Į pirkinių krepšelį įdėta bent viena prekė	<ol style="list-style-type: none"> 1. Pasirenkama prekė kuri bus šalinama iš pirkinių krepšelio 2. Paspaudžiamas mygtukas pašalinti 	Stebima ar po mygtuko paspaudimo atnaujintame puslapyje nėra atvaizduota pasirinktos prekės	Taip

<p>TS-10 Patikrinti ar prisijungęs vartotojas mato rekomendacijas pagal registracijos profilį</p> <p>Rekomendacijos pagal profilį kuriamos pagal vartotojo amžių, lytį ir gyvenamąją vietą</p>				
TA-13	Paleista naršyklė, prisijungta prie paskyros ir atidarytas rekomendacijų puslapis	1. Paspaudžiama nuoroda Rekomenduojame ir naršymos juostos	Vartotojas mato rekomendacijas pagal profilį	Taip
<p>TS-11 Patikrinti ar vartotojas gali matyti pasirinktos kategorijos prekes</p> <p>Paspaudus mygtuką su kategorijos pavadinimu produktų kataloge atvaizduojamos prekės iš tikėtinios kategorijos</p>				
TA-14	Paleista internetinė naršyklė	1. Vartotojas paspaudžia mygtuką su kategorijos pavadinimu bet kuriame prekių kataloge	Atvaizduojamos tik prekės, kurios yra tos pačios kategorijos	Taip
<p>TS-16 Patikrinti ar prisijungęs vartotojas mato rekomendacijas pagal tai ką jis yra pirkęs anksčiau</p> <p>Vartotojas, kuris pirkęs bent vieną prekę praeityje turėtų matyti rekomenduojamas prekes. Šios prekės randamos pagal bendrus bruožus</p>				
TA-15	Paleista interneto naršyklė, vartotojas prisijungęs prie savo paskyros	1. Įdedama prekė į pirkinių krepšelį 2. Pirkinių krepšelio puslapyje paspaudžiamas mygtukas pirkti 3. Nekeliaujama į rekomendacijų puslapį 4. Patikrinama ar rekomenduojama prekė pagal panašumą su 2 punkte nupirktą preke	Rekomendacijų puslapyje atvaizduojama prekė pagal panašumą su neseniai pirktą preke	Taip

Sistemos testavimui buvo sukurta 19 skirtingų scenarijų. Sistemos funkcinį ir nefunkcinį reikalavimų testavimui buvo nagrinėjama 12 scenarijų ir jiems buvo parašyta 15 testavimo atvejų. Visi ištestuoti atvejai buvo įgyvendinti sėkmingai.

5.2. Testavimo apibendrinimas

Sistemos išbandymas darbinėje aplinkoje pavyko sklandžiai. Šie rankiniai testai yra paprastas ir greitas būdas ištestuoti ir sukurti įrašus apie testavimo atlikimą. Sukuriamas šioks toks vieno asmens patentas, jog sistema veiks be didesnių klaidų. Šiuos testus yra lengva replikuoti ir pritaikyti. Testavimo eiga gali būti įvairi, todėl tokio pobūdžio testus gali kurti keli asmenys, vėliau juos palyginti. Vietoj konkrečių sistemos kodo eilučių testuojamas funkcionalumas. Keliuose testiniuose atvejuose teko kodą papildyti papildomom eilutėmis siekiant įsitikinti, jog gaunamas ar negaunamas tikėtinas rezultatas. Iš viso buvo sukurta 19 skirtingų scenarijų, kurie skirti ištestuoti ar pritaikyti funkciniai ir nefunkciniai reikalavimai. Iš 19 sukurtų scenarijų 12 iš jų buvo pasirinkti ir jiems iš viso buvo parašyta 15 testavimo atvejų. Visi ištestuoti testavimo atvejai buvo sėkmingai. Tokiu būdu buvo įrodyta, jog sistema atitinka iškeltus reikalavimus.

Išvados

1. Atlikus el. komercijos rekomendacijų sistemų, funkcijų ir algoritmų apžvalgą paaiškėjo, kad jose yra skirtingai taikomi panašūs sprendimai, todėl sistemos realizavimui pasirinkta grafų duomenų bazė, siekiama atkartoti funkcijas ir algoritmus rekomendacijų sudarymui. Nustatyta, jog grafų duomenų bazėje esančias mazgų ir ryšių struktūros optimaliai pritaikomos naudingos informacijos radimui. Turint žinių apie rekomendacijų sistemas, funkcijas ir algoritmus galima teigti, jog nėra vieno teisingo algoritmo ar sistemos, todėl pasirinkta atkartoti suprantamiausiai atrodantį sprendimą.
2. Išanalizavus skirtingas to paties tipo grafų duomenų bazių valdymo sistemas, pastebėta jog šiuo metu populiariausios yra Neo4j ir Microsoft Cosmos DB, iš to seka, jog bus lyginamos šios DBVS siekiant surasti sistemai labiau tinkantį duomenų bazės sprendimą. Padaryta išvada, kad Neo4j DBVS yra labiau orientuota į grafus. Išvada būtų tokia, jog sistemą sukurti ir atsakyti į baigiamojo darbo temą galima paprasčiau ir konkrečiau pasitelkiant Neo4j grafų duomenų bazių valdymo sistemą.
3. Suformulavus funkcinis ir nefunkcinis reikalavimus el. komercijos rekomendacijų sistemai buvo sukurti pagrindiniai sistemos tikslai, pagal kuriuos numatoma kaip bus suprojektuota ir vėliau veiks sistema. Gauta, jog be funkcinų reikalavimų neina sukurti sistemos, kuri atsakytų į baigiamojo darbo temą tikslingai, o be nefunkcinų reikalavimų detalai. Funkcinių reikalavimų pagalba logiškai sukuriamas rekomendacijų sistema. Nefunkcinių reikalavimų pagalba detalizuojama sistema. Reikalavimai yra sistemos pamatai be kurių neįmonama turėti tikslumo ir detališkumo.
4. Atlikus el. komercijos rekomendacijų sistemos projektavimą ir programavimą, gauta, jog be UML diagramų ir jų aprašymo yra sudėtinga sukurti sistemos kontekstą reikalingą rekomendacijų sistemos veikimui. Be Java programavimo kalba teikiamos logikos ir sprendimų neina veiksmingai pritakyti skirtingų komponentų. Nuspręsta braižyti UML užduočių, sekų, klasių ir komponentų diagramas. Duomenų bazė projektuojama pagal UML klasių diagramą. Pasirinkti Java technologiniai sprendimai Spring-Boot ir Thymeleaf. Pritaikyti Apache Tomcat ir Neo4j Community serveriai. Atlikus projektavimą ir programavimą yra sukurta el. komercijos rekomendacijų sistema pagal UML diagramas pasitelkiant Java technologijas.
5. Atlikus sukurtos sistemos testavimą patikrinta ar sistema veiks sklandžiai. Rankiniu būdu kuriami testai, iš kurių sprendžiama ar yra laikomasi sistemos funkcinų ir nefunkcinių reikalavimų. Testavimo rezultatai buvo teigiami, tai yra, pagal reikalavimus sukurti testinių scenarijų atvejai buvo patikrinti ir gauti teigiami rezultatai. Galima teigti, jog be sistemos testavimo negalima turėti kokio nors patento, kad sistema veikia sklandžiai ir veiks sklandžiai ateityje.

Literatūros sąrašas

1. Amazon. (2023). Customer Reviews. Tiriamasis straipsnis <https://www.sellersnap.io/amazon-frequently-bought-together/>
2. Ayala, J. (2021). What is Amazon's Frequently Bought Together and How Does it Work. Tiriamasis straipsnis <https://www.sellersnap.io/amazon-frequently-bought-together/>
3. Chao, J. (2018). Graph Databases for Beginners: Graph Search Algorithm Basics. Tiriamasis straipsnis <https://go.neo4j.com/rs/710-RRC-335/images/Graph-Databases-for-Beginners-EN-US.pdf>
4. DataStax, A. (2013). Graph Based Recommendation Systems at eBay. Tiriamasis straipsnis <https://www.slideshare.net/planetcassandra/e-bay-nyc>
5. DB-Engines. (2023). Microsoft Azure Cosmos DB System Properties. Tiriamasis straipsnis <https://db-engines.com/en/system/Microsoft+Azure+SQL+Database>
6. Deepjyoti, R., & Mala, D. (2022). A systematic review and research perspective on recommender systems. J Big Data. Tiriamasis straipsnis <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00592-5>
7. eBay. (2023). Feedback profiles. Tiriamasis straipsnis <https://www.ebay.com/help/account/changing-account-settings/feedback-profiles?id=4204>
8. G2. (2018). Compare Azure Cosmos DB and Neo4j. Tiriamasis straipsnis <https://www.g2.com/compare/azure-cosmos-db-vs-neo4j-graph-database>
9. Hashmi, A. (2021). Graph Databases: Neo4j AuraDB versus Azure Cosmos DB. Tiriamasis straipsnis https://medium.com/@adnan_hashmi/graph-databases-neo4j-auradb-versus-azure-cosmos-db-65860f66d4
10. Hoff, T. (2009). Neo4j - a Graph Database the Kicks ButtoX. Possibility Outpost. Tiriamasis straipsnis <http://highscalability.com/neo4j-graph-database-kicks-buttox>
11. Kaushik, N. (2021). Microsoft adds more programming language support to Azure SDK. Tiriamasis straipsnis <https://content.techgig.com/technology/microsoft-adds-more-programming-language-support-to-azure-sdk/articleshow/80755079.cms>
12. Leone, M. (2021). Supporting Next-generation Applications at Scale with Microsoft Azure Cosmos DB. Tiriamasis straipsnis <https://learn.microsoft.com/en-us/azure/cosmos-db/introduction>
13. Marrancone, L. (2020). How to create recommendation engine in neo4j. Tiriamasis straipsnis <https://medium.com/larus-team/how-to-create-recommendation-engine-in-neo4j-7963e635c730>

14. Microsoft. (2018). Azure Cosmos DB and GDPR. Specifikacija <https://learn.microsoft.com/en-us/compliance/regulatory/gdpr>
15. Microsoft. (2019). Common Azure Cosmos DB use cases. Tiriamasis straipsnis <https://learn.microsoft.com/en-us/azure/cosmos-db/use-cases>
16. Neo4j. (2023). Neo4j Drivers. Specifikacija <https://neo4j.com/download-center/>
17. Pine, J. (1993). Mass Customization. Knyga
18. SaaSHub. (2022). DB, neo4j VS Azure Cosmos. Tiriamasis straipsnis <https://db-engines.com/en/system/Microsoft+Azure+Cosmos+DB%3BNeo4j>
19. Schafer, J., Konstan, J., & Riedl, J. (1999). Recommender Systems in E-Commerce. Tiriamasis darbas <https://dl.acm.org/doi/pdf/10.1145/336992.337035>
20. Svensson, A., & Lyon, W. (2017). How to Design Retail Recommendation Engines with Neo4j. Tiriamasis straipsnis <https://www.slideshare.net/neo4j/how-to-design-retail-recommendation-engines-with-neo4j>
21. TrustRadius. (2016). Azure Cosmos DB vs. Neo4j. Tiriamasis straipsnis <https://www.trustradius.com/compare-products/azure-cosmos-db-vs-neo4j>
22. Webber, J. (2021). Powering Real-Time Recommendations. Tiriamasis straipsnis https://go.neo4j.com/rs/710-RRC-335/images/Neo4j_WP_Recommendations_EN_BUS.pdf?_ga=2.216021586.597232194.1522833081-1282627512.1522833081

Priedai

PRIEDAS 1

A* algoritmas yra Dijkstra's algoritmo patobulinta versija sudedant kelis Dijkstra's algoritmo elementus su elementais iš geriausias-pirmas paieškos.

Tariamasi kaip „A-žvaigždė“, A* yra paremtas įžvalgomis, jog kai kurios paieškos yra informuotos, kas leidžia priimti geresnius sprendimus ieškant, kuriuos kelius reikėtų priimti keliaujant per grafą. Kaip ir Dijkstra's algoritmas, A* gali apieškoti didelį grafo plotą, bet kaip geriausias-pirmas paieška, A* naudoja euristiką padedančią paieškoje.

Papildomai, nors Dijkstra's algoritmas teikia pirmenybę apieškoti mazgus, kurie yra arti pradinio mazgo, geriausias-pirmas paieška teikia pirmenybę mazgams, kurie yra arčiau galutinio mazgo. A* balansuoja tarp dviejų metodikų, siekiant užtikrinti, jog kiekviename lygmenyje pasirenkamas mazgas su trumpiausiu keliu.

Sistemos įgyvendinimui reikalingi du serveriai. Vienas iš jų yra Neo4j Community server, reikalingas įgalinti duomenų bazę, o kitas yra Apache Tomcat server, kuris paleidžiamas Java Spring-Boot programos pagalba. Antrasis serveris yra reikalingas duomenų mainams tarp duomenų bazės, sistemos ir grafinės vartotojo sąsajos.

```

C:\Users\luluk>cd C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\bin
C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\bin>neo4j console
Directories in use:
home:         C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0
config:       C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\conf
logs:        C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\logs
plugins:     C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\plugins
import:      C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\import
data:        C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\data
certificates: C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\certificates
licenses:    C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\licenses
run:         C:\Users\luluk\Desktop\universiteto\Programavimo\neo4j-community-5.6.0\run
Starting Neo4j.
WARNING! You are using an unsupported Java runtime.
* Please use Oracle(R) Java(TM) 17, OpenJDK(TM) 17 to run Neo4j.
* Please see https://neo4j.com/docs/ for Neo4j installation instructions.
2023-05-08 08:10:06.543+0000 INFO Starting...
2023-05-08 08:10:07.841+0000 INFO This instance is ServerId{9fec874c} (9fec874c-caaa-4a02-ba6f-595a7ea5dacc)
2023-05-08 08:10:10.262+0000 INFO ===== Neo4j 5.6.0 =====
2023-05-08 08:10:14.232+0000 INFO Bolt enabled on localhost:7687.
2023-05-08 08:10:16.194+0000 INFO Remote interface available at http://localhost:7474/
2023-05-08 08:10:16.201+0000 INFO id: 0E6337CB0CEA2EFD2701D1ACD8E4F22DBC9C64610CDEBAE7611146418C748B
2023-05-08 08:10:16.203+0000 INFO name: system
2023-05-08 08:10:16.203+0000 INFO creationDate: 2023-04-12T18:16:43.838Z
2023-05-08 08:10:16.205+0000 INFO Started.

```

19 pav. Pirmasis serveris ir jo paleidimas

```

C:\Program Files\Java\jdk-17\bin\java.exe ...
...
:: Spring Boot ::
(v3.0.5)

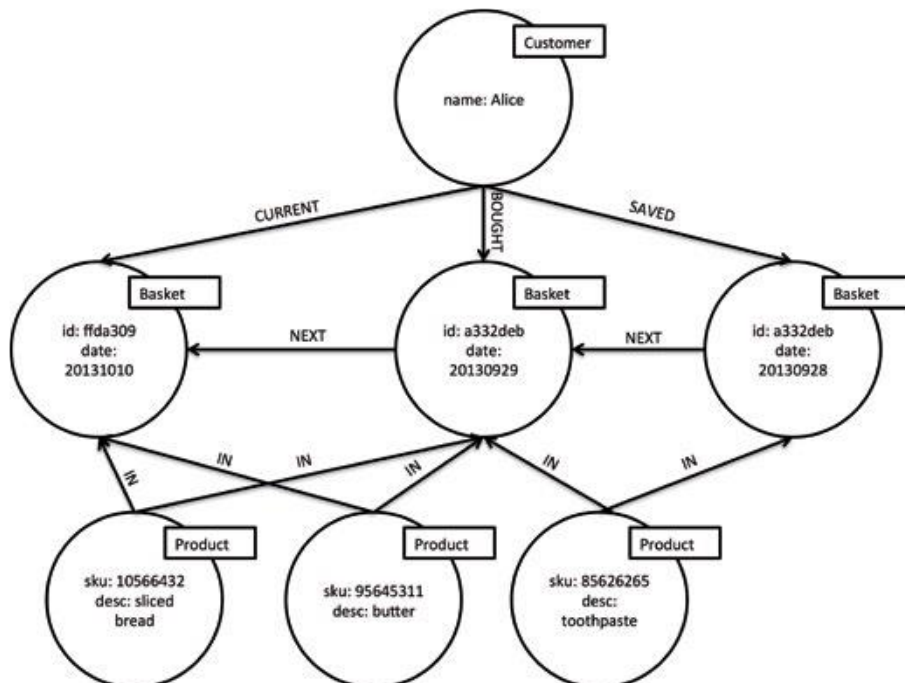
2023-05-07T20:39:35.053+03:00 INFO 17440 --- [main] app.webserver.ecommerce.Server : Starting Server using Java 17.0.6 with PID
17440 (C:\Users\luluk\Desktop\universiteto\Programavimo\Best One So far MVC works\E-Commerce system (Demo 60)\target\classes started by luluk in
C:\Users\luluk\Desktop\universiteto\Programavimo\Best One So far MVC works\E-Commerce system (Demo 60))
2023-05-07T20:39:35.060+03:00 INFO 17440 --- [main] app.webserver.ecommerce.Server : No active profile set, falling back to 1
default profile: "default"
2023-05-07T20:39:36.070+03:00 INFO 17440 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data Neo4j repositories
in DEFAULT mode.
2023-05-07T20:39:36.256+03:00 INFO 17440 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in
179 ms. Found 0 Neo4j repository interfaces.
2023-05-07T20:39:36.265+03:00 INFO 17440 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data Neo4j repositories
in DEFAULT mode.
2023-05-07T20:39:36.304+03:00 INFO 17440 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in
38 ms. Found 3 Neo4j repository interfaces.
2023-05-07T20:39:36.879+03:00 INFO 17440 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-05-07T20:39:36.890+03:00 INFO 17440 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-05-07T20:39:36.891+03:00 INFO 17440 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1
.7]
2023-05-07T20:39:37.001+03:00 INFO 17440 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded
WebApplicationContext
2023-05-07T20:39:37.001+03:00 INFO 17440 --- [main] org.springframework.web.context.support.AnnotationConfigWebApplicationContext : Root WebApplicationContext: initialization

```

20 pav. Antrasis serveris ir jo paleidimas

Be serverių internetinis puslapis neveiktų. Nepavyktų pasiekti sistemos teikiamo funkcionalumo ir nebūtų sprendžiamos iškeltos baigiamojo darbo problemos.

Schemas ir maketai yra kuriami siekiant paaiškinti skirtingų sistemos dalių veikimą. Pardavimo schemas pavyzdys pasitelkiant Neo4j grafų duomenų bazę siekia paaiškinti kaip sistemos vartotojas sąveikauja su sistema. Tiksliau kaip kliento pirkimo krepšelis yra išsaugojamas grafų duomenų bazėje.



21 pav. Krepšelių išsaugojimas grafų duomenų bazėje (Webber, 2021).

Kliento prekės pirkimą galima apibūdinti grafo iliustracija. Grafo iliustracija parodo kaip susieti paprasti pirkinių krepšelių sąrašai, sujungti su **NEXT** santykiu, yra panaudojami siekiant sukurti kliento pirkinių sąrašą. Šis ryšys nurodo kaip keičiasi pirkinio sąrašas kintant laikui. Pažvelgus į šį grafą galime įžvelgti, jog klientas apsilankė tris kartus. Atidėjo savo pirkimą vėlesniam laikui. Šis atidėjimas yra apibrėžiamas su **SAVED** santykiu tarp kliento ir krepšelio mazgų. Galiausiai klientas nusipirko vieną krepšelį. Šis veiksmas nurodomas **BOUGHT** santykiu tarp kliento ir krepšelio mazgo. Šiuo metu klientas surenka krepšelį, šis rinkimas yra atvaizduojamas su **CURRENT** santykiu, kuris nukreipia į aktyvų krepšelį, kuris yra susieto sąrašo priekyje.

Yra svarbu suprasti, jog atvaizduota ne schema arba ER-diagrama bet reprezentuojama aktuali vieno kliento duomenų struktūra. Pritaikytas realus kliento grafas būtų milžiniškas. Per daug didelis, jog būtų galima nagrinėti kaip pavyzdį, tačiau demonstruotų tokio pačio tipo struktūrą.

Grafų formoje yra lengva išsiaiškinti klientų elgesį. Jie tapo potencialiu nauju klientu, bet nenusipirko dantų pastos ir grįžę kitą dieną nusipirko dantų pasta, duotos ir sviesto. Galiausiai klientas nusprendė nusipirkti duonos ir dantų pastos. Čia galime įžvelgti atsikartojantį elgesio modelį, kurį pritaikius galima patobulinti kliento aptarnavimą.

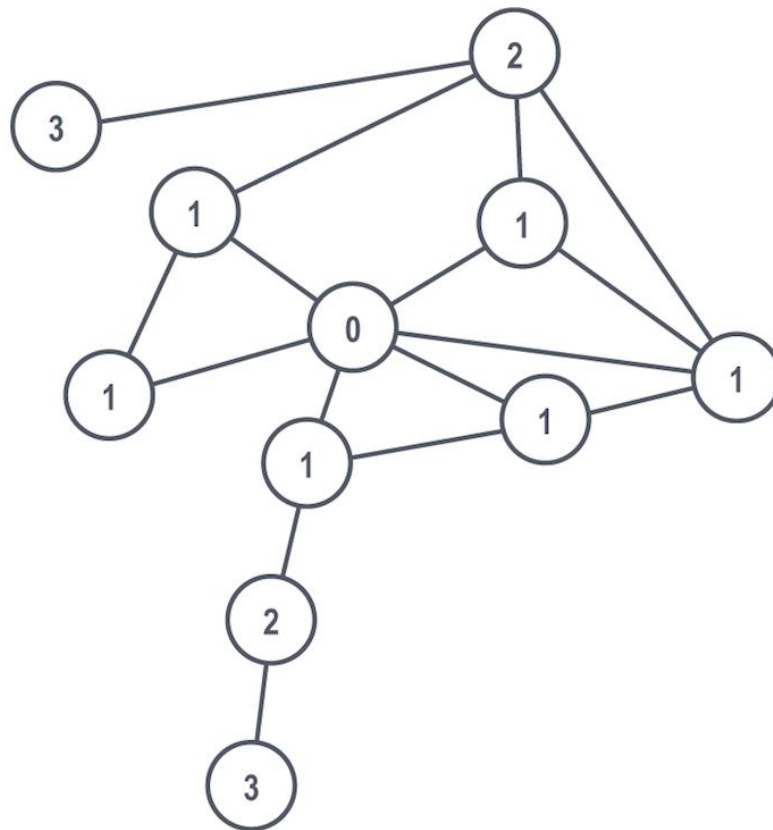
Turint klientų grafą ir praeities produktus, kuriuos jie pirkė, galime pagalvoti apie rekomendacijas, kuriuos galėtų įtakoti jų tolimesnį pirkimo elgesį. Viena iš paprasčiausių rekomendacijų yra pateikti parduotuvėje esančių populiariausių prekių sąrašą. Šią rekomendaciją aprašyti yra paprasta pasitelkiant Cypher duomenų bazių užklausų kalbą (Webber, 2021).

Paiešką į gylį paieškos algoritmas atlieka savo veiklą keliaudamas nuo pradžios mazgo iki kokios nors pabaigos mazgo, pasiekus pabaigą procesas yra kartojamas kitoku keliu iš to paties pradinio mazgo iki kol paieška yra sėkmingai pabaigiama. Apskritai paieškos į gylį algoritmas yra geras pasirinkimas, kai yra siekiama atrasti diskrečias informacijos dalis. Tokie algoritmai yra geras strateginis pasirinkimas bendriems keliavimais grafais (Chao, 2018).

Pats klasiškiausias arba paprasčiausio lygmens paieškos į gylį algoritmas yra vadinamas uniformuota paieška. Šios paieškos metu algoritmas ieško kelią iki kol yra pasiekiamas grafo galas, tada keliaujama į pradžios mazgą ir bandoma surasti kitą mazgą.

Priešingai, dirbant su semantiškai turtingomis grafų duomenų bazėmis atsiranda informuotos paieškos. Ši paieška atlieka ankstyvų paieškos nutraukimą, jeigu nerandama jokių mazgų su suderintais išeinamaisiais ryšiais. Informuotos paieškos turi trumpesnį vykdymo laiką.

Paieškos į plotį paieškos algoritmai vykdo paieškas tiriant grafą vienu sluoksniu po kito. Šios paieškos pradedamos su mazgais vieno lygmens gylio atstumu nuo pradinio mazgo, toliau seka antro lygmens gylio mazgai, tada trečio gylio iki kol visas grafas bus praeitas.

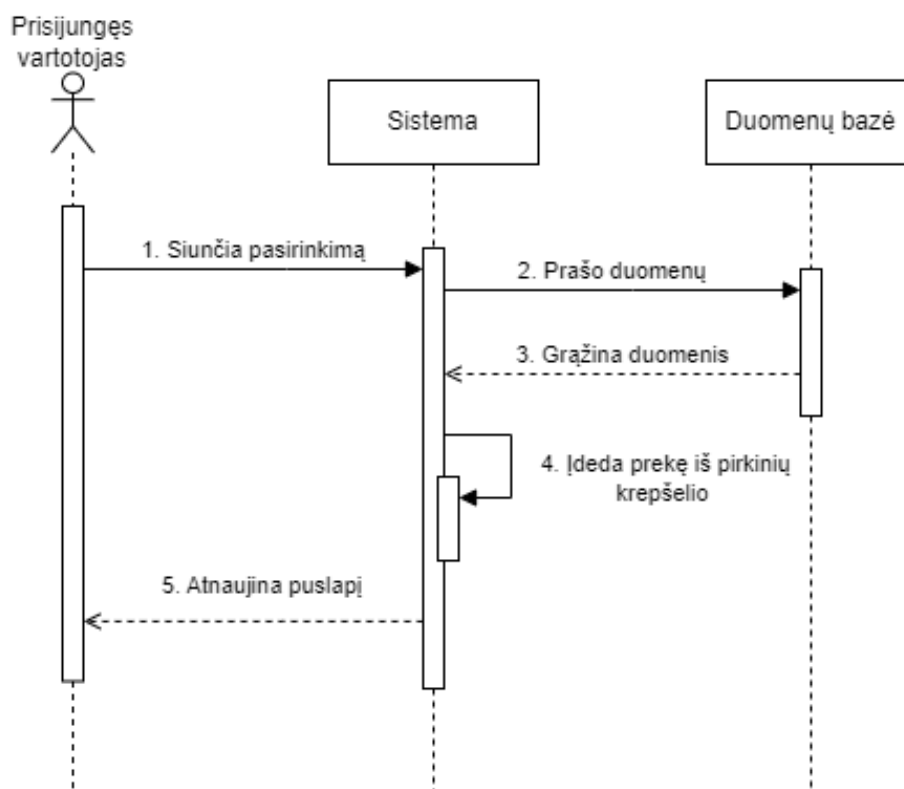


22 pav. Paieškos į plotį algoritmo pavyzdys (Chao, 2018)

Paieškos į gylį paieškos algoritmas yra pradedamas pasirinktame mazge, šiuo atveju pasirinktas mazgas yra pažymimas skaičiumi 0. Toliau keliaujama iki kito mazgų, kurie yra tik vieno žingsnio atstumu, naujai pasiekti mazgai žymimi skaičiumi 1. Mazgai, kurie yra pasiekiami per du žingsnius pažymimi skaičiumi 2 ir taip toliau (Chao, 2018).

Su Neo4j galima sukurti įvairius rekomendacijų sistemas panaudojant algoritmus. Rekomendacijų algoritmus galima suskirstyti į turiniu paremtą filtravimą ir bendradarbiavimu paremtą filtravimą. Turiniu paremtas filtravimas rekomenduoja dalykus, kurie yra panašūs į tuos, kuriuos vartotojas peržiūro, įvertino gerai arba nusipirko anksčiau. Bendradarbiavimu paremtas filtravimas pritaiko preferencijas, įvertinimus ir tinkle esančių kitų vartotojų veiksmus siekiant rekomenduoti dalykus (Leone, 2021).

Įdėti prekę į krepšelį seka



23 pav. Įdėti prekę į krepšelį seka

Sekos pavadinimas: Įdėti prekę į krepšelį seka.

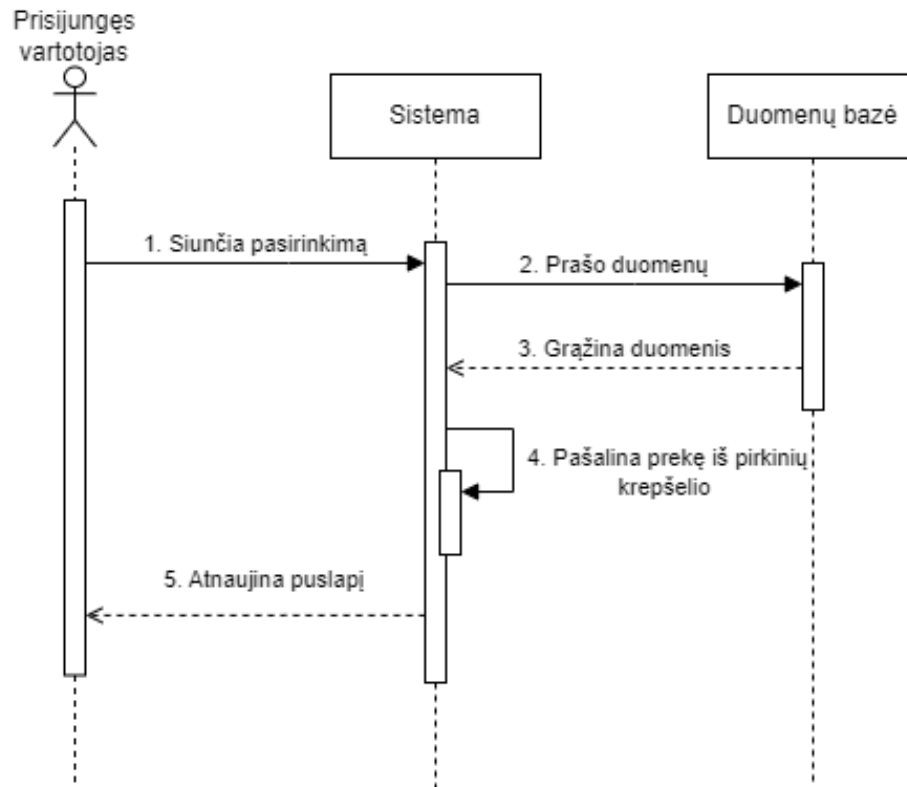
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0104.

21 lentelė. Įdėti prekę į krepšelį sekų diagramos paaiškinimas

užduoties numeris	7
užduoties pavadinimas	Įdėti prekę
siekiamas tikslas	Įdėti prekę į krepšelį
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka įdėti prekę į krepšelį
užduoties prioritetas	1-as (aukštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiekta internetinės svetainės puslapis su produktais
sėkmingos baigties "po" sąlygos	Atnaujinama informacija ir vartotojas lieka tame pačiame puslapyje
nesėkmingos baigties "po" sąlygos	Neatnaujinama informacija ir vartotojas lieka tame pačiame puslapyje

Pašalinti iš krepšelio seka



24 pav. Pašalinti iš krepšelio seka

Sekos pavadinimas: Pašalinti iš krepšelio seka.

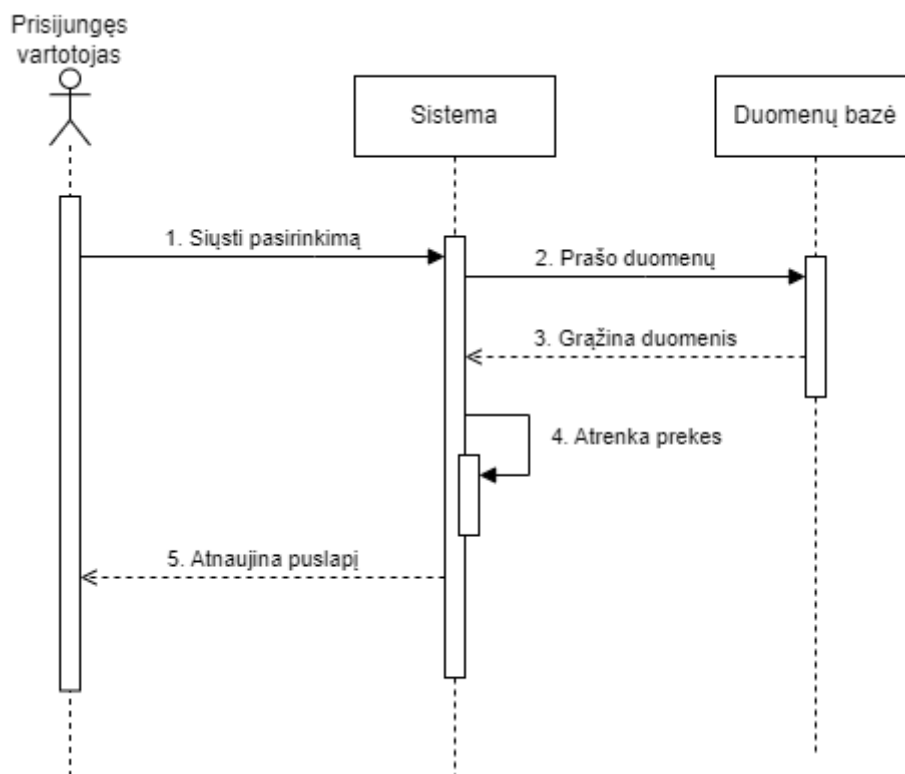
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0105.

22 lentelė. Pašalinti prekę iš krepšelio sekų diagramos paaiškinimas

užduoties numeris	8
užduoties pavadinimas	Pašalinti prekę
siekiamas tikslas	Pašalinti prekę iš krepšelio
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka pašalinti prekę iš krepšelio
užduoties prioritetas	1-as (aukštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiekta internetinės svetainės puslapis su produktais
sėkmingos baigties "po" sąlygos	Atnaujinama informacija ir vartotojas lieka tame pačiame puslapyje
nesėkmingos baigties "po" sąlygos	Neatnaujinama informacija ir vartotojas lieka tame pačiame puslapyje

Atrinkti prekes seka



25 pav. Atrinkti prekes seka

Sekos pavadinimas: Atrinkti prekes seka.

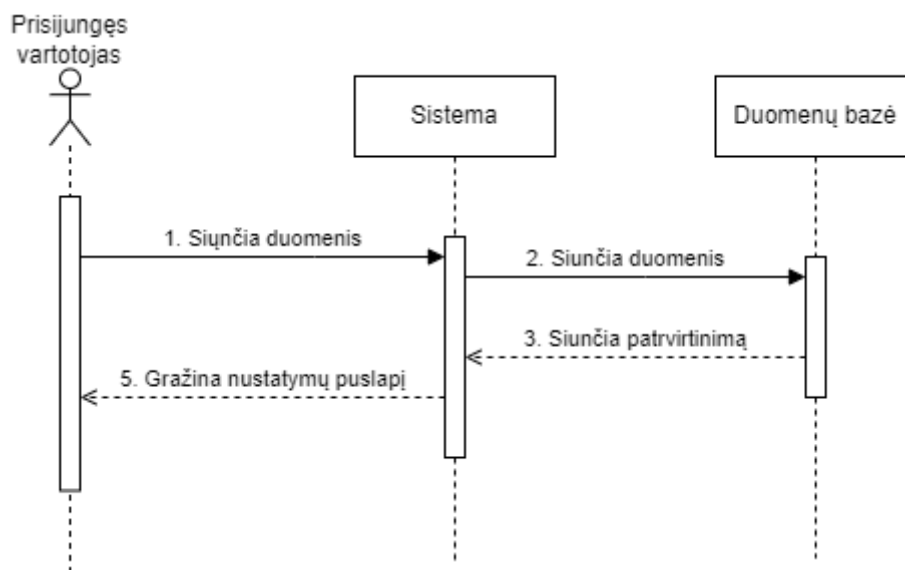
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0103.

23 lentelė. Atrinkti prekes sekų diagramos paaiškinimas

užduoties numeris	9
užduoties pavadinimas	Atrinkti prekes
siekiamas tikslas	Atrinkti prekes iš prekių katalogo
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka atrinkti prekes
užduoties prioritetas	2-as (vidutinis)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiekta internetinės svetainės puslapis su prekėmis
sėkmingos baigties "po" sąlygos	Atnaujinama informacija ir vartotojas lieka tame pačiame puslapyje
nesėkmingos baigties "po" sąlygos	Neatnaujinama informacija ir vartotojas lieka tame pačiame puslapyje

Keisti slaptažodį seka



26 pav. Keisti slaptažodį seka

Sekos pavadinimas: Keisti slaptažodį seka.

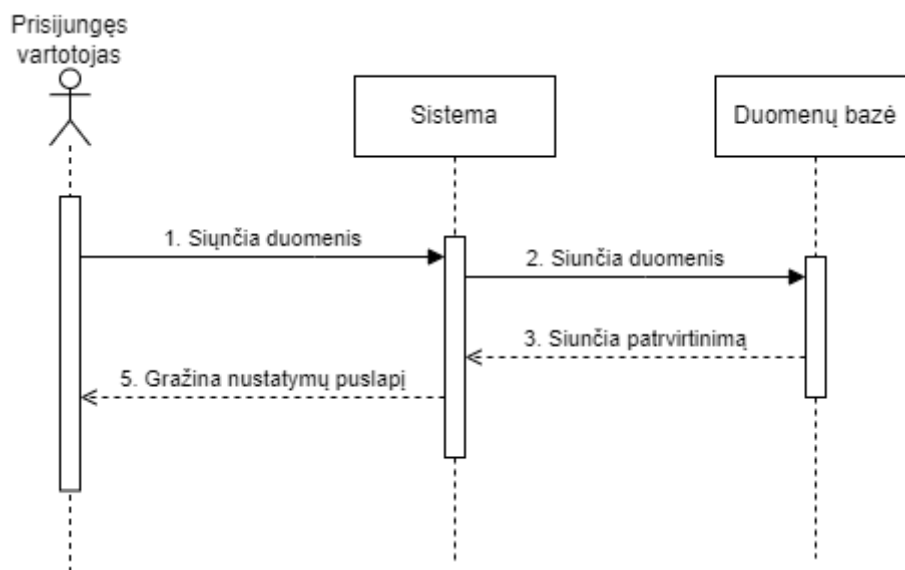
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0203.

24 lentelė. Keisti slaptažodį sekų diagramos paaiškinimas

užduoties numeris	10
užduoties pavadinimas	Keisti slaptažodį
siekiamas tikslas	Keisti vartotojo paskyros slaptažodį
užduoties vykdymą inicijuojantis įvykis (trigeris)	Vartotojas pasirenka keisti paskyros duomenis
užduoties prioritetas	3-as (žemas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiekta internetinės svetainės puslapis su nustatytais
sėkmingos baigties "po" sąlygos	Atnaujinama informacija ir vartotojas lieka tame pačiame puslapyje
nesėkmingos baigties "po" sąlygos	Neatnaujinama informacija ir vartotojas lieka tame pačiame puslapyje

Keisti lytį seką



27 pav. Keisti lytį seką

Sekos pavadinimas: Keisti lytį seką.

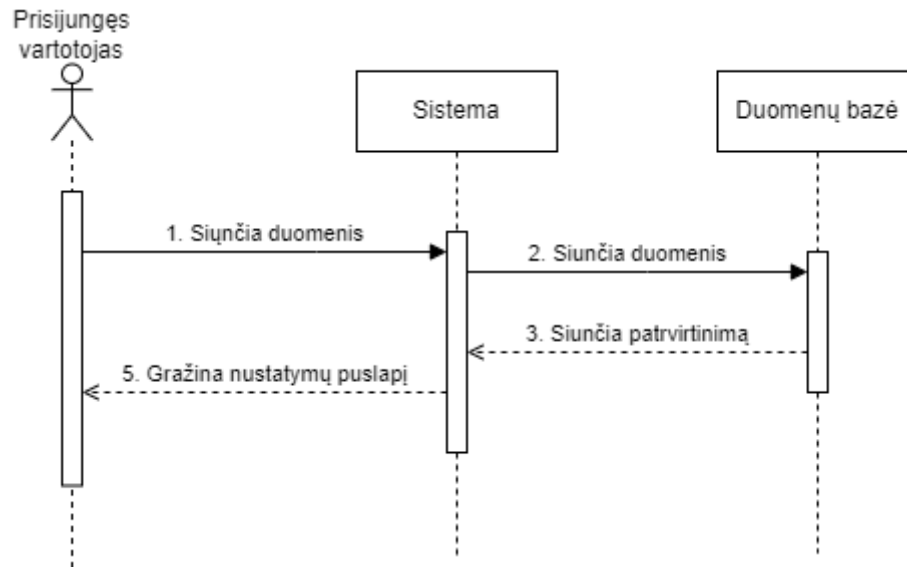
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0203.

25 lentelė. Keisti lytį sekų diagramos paaiškinimas

užduoties numeris	11
užduoties pavadinimas	Keisti lytį
siekiamas tikslas	Keisti vartotojo paskyros lytį
užduoties vykdymą inicijuojantis įvykis (trigeris)	Vartotojas pasirenka keisti paskyros duomenis
užduoties prioritetas	3-as (žemas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiekta internetinės svetainės puslapis su nustatymais
sėkmingos baigties "po" sąlygos	Atnaujinama informacija ir vartotojas lieka tame pačiame puslapyje
nesėkmingos baigties "po" sąlygos	Neatnaujinama informacija ir vartotojas lieka tame pačiame puslapyje

Keisti gyvenamąją vietą seka



28 pav. Keisti gyvenamąją vietą seka

Sekos pavadinimas: Keisti gyvenamąją vietą seka.

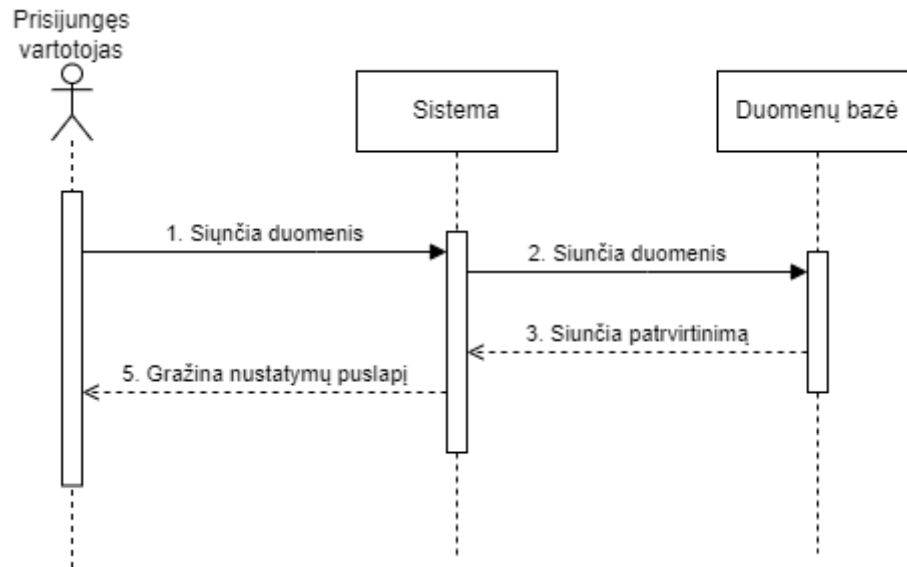
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0203.

26 lentelė. Keisti gyvenamąją vietą sekų diagramos paaiškinimas

užduoties numeris	12
užduoties pavadinimas	Keisti gyvenamąją vietą
siekiamas tikslas	Keisti paskyros gyvenamąją vietą
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka keisti paskyros duomenis
užduoties prioritetas	3-as (žemas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiekta internetinės svetainės puslapis su nustatymais
sėkmingos baigties "po" sąlygos	Atnaujinama informacija ir vartotojas lieka tame pačiame puslapyje
nesėkmingos baigties "po" sąlygos	Neatnaujinama informacija ir vartotojas lieka tame pačiame puslapyje

Keisti gimimo metus seka



29 pav. Keisti gimimo metus seka

Sekos pavadinimas: Keisti gimimo metus seka.

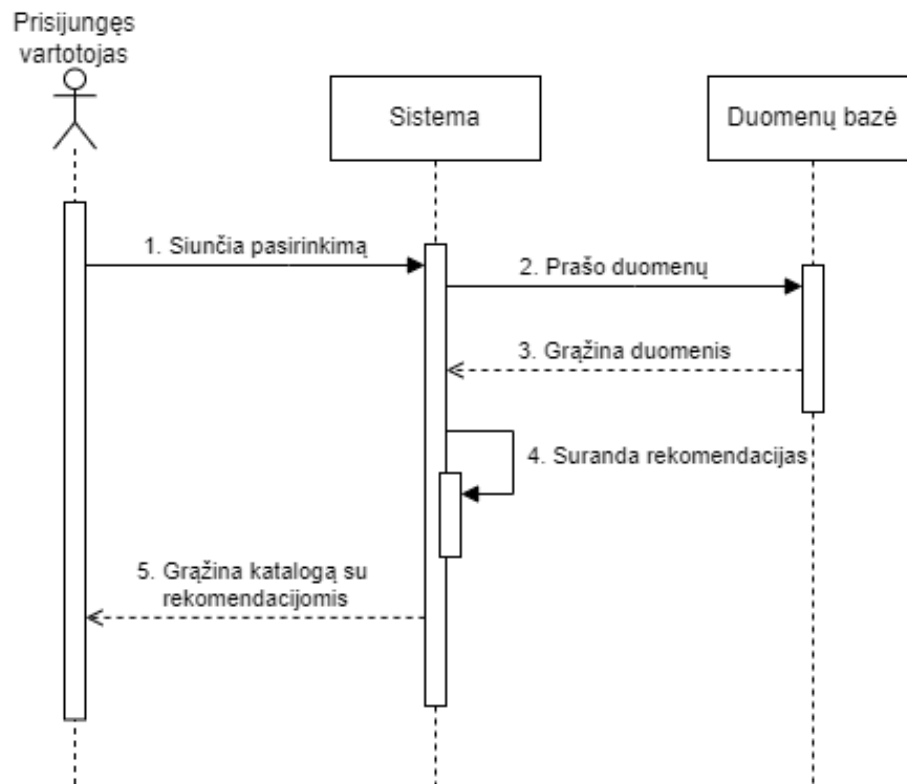
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0203.

27 lentelė. Keisti gimimo metus sekų diagramos paaiškinimas

užduoties numeris	13
užduoties pavadinimas	Keisti gimimo metus
siekiamas tikslas	Keisti paskyros gimimo metus
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka keisti paskyros duomenis
užduoties prioritetas	3-as (žemas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiekta internetinės svetainės puslapis su nustatymais
sėkmingos baigties "po" sąlygos	Atnaujinama informacija ir vartotojas lieka tame pačiame puslapyje
nesėkmingos baigties "po" sąlygos	Neatnaujinama informacija ir vartotojas lieka tame pačiame puslapyje

Apžvelgi populiariausias prekes seka



30 pav. Apžvelgti populiariausias prekes seka

Sekos pavadinimas: Apžvelgti populiariausias prekes seka.

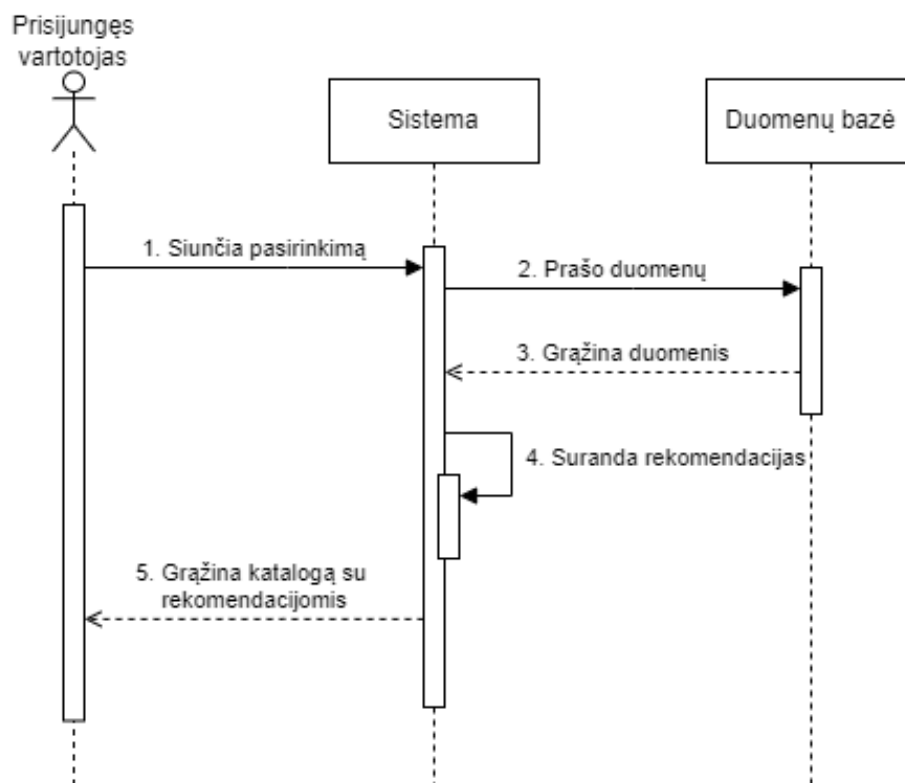
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0203.

28 lentelė. Apžvelgti populiariausias prekes sekų diagramos paaiškinimas

užduoties numeris	14
užduoties pavadinimas	Apžvelgti populiariausias prekes
siekiamas tikslas	Apžvelgti populiariausias prekes
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka apžvelgti rekomendacijas
užduoties prioritetas	1-as (aukštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis turinti nuorodas į rekomendacijų puslapius
sėkmingos baigties "po" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas
nesėkmingos baigties "po" sąlygos	Nesėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas

Apžvelgti rekomenduojamas prekes pagal paskyros duomenis seka



31 pav. Apžvelgti rekomendacijas pagal paskyros duomenis seka

Sekos pavadinimas: Apžvelgti rekomendacijas pagal paskyros duomenis.

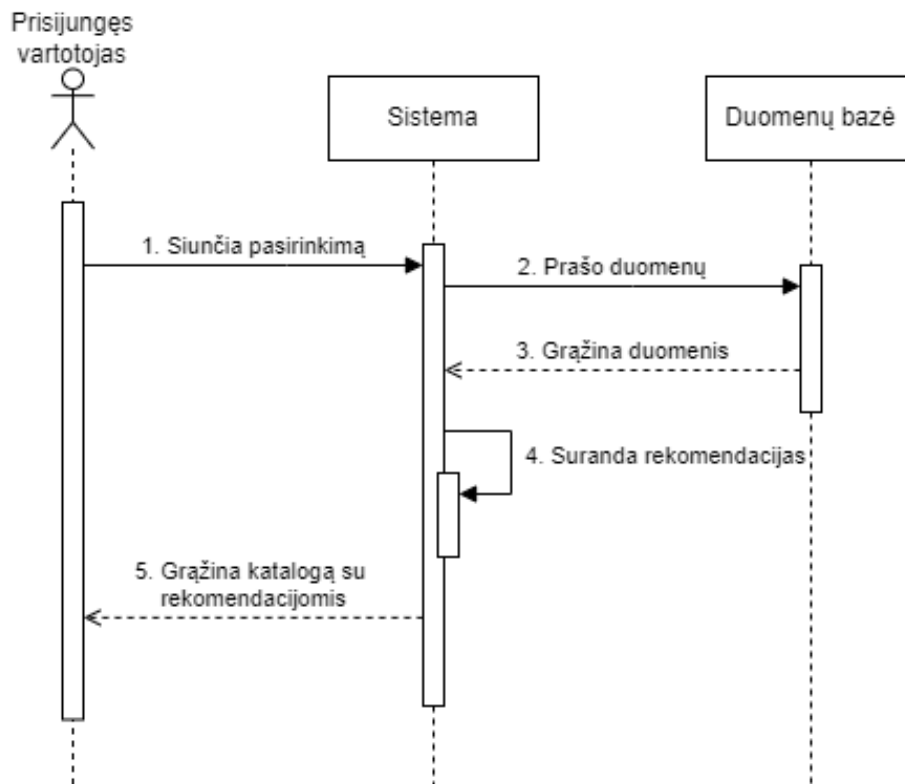
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0203.

29 lentelė. Apžvelgti rekomendacijas pagal paskyros duomenis sekų diagramos paaiškinimas

užduoties numeris	15
užduoties pavadinimas	Apžvelgti rekomendacijas pagal paskyros duomenis
siekiamas tikslas	Apžvelgti rekomendacijas pagal paskyros duomenis
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka apžvelgti rekomendacijas
užduoties prioritetas	1-as (aukštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis turinti nuorodas į rekomendacijų puslapius
sėkmingos baigties "po" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas
nesėkmingos baigties "po" sąlygos	Nesėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas

Apžvelgti rekomenduojamas prekes pagal profilį seka



32 pav. Apžvelgti rekomendacijas pagal paskyros profilį seka

Sekos pavadinimas: Apžvelgti rekomendacijas pagal paskyros profilį seka.

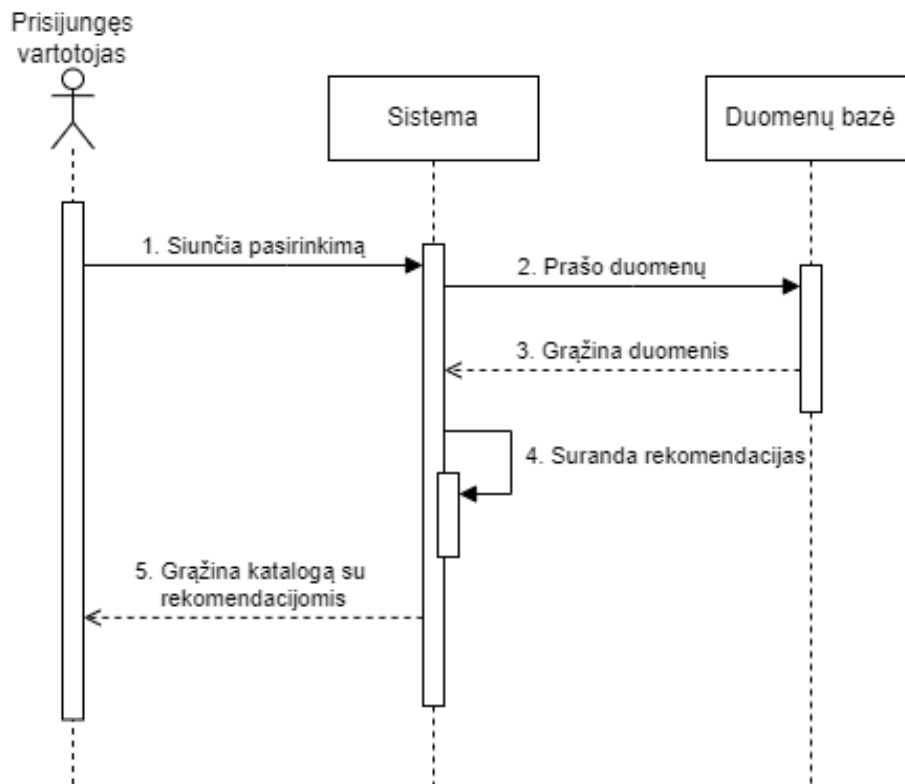
Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0203.

30 lentelė. Apžvelgti rekomenduojamas pagal paskyros profilį sekų diagramos paaiškinimas

užduoties numeris	16
užduoties pavadinimas	Apžvelgti rekomendacijas pagal paskyros duomenis
siekiamas tikslas	Apžvelgti rekomendacijas pagal paskyros duomenis
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasirenka apžvelgti rekomendacijas
užduoties prioritetas	1-as (aukštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis turinti nuorodas į rekomendacijų puslapius
sėkmingos baigties "po" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas
nesėkmingos baigties "po" sąlygos	Nesėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas

Apžvelgti pirmo prisijungimo rekomendacijas seka



33 pav. Apžvelgti pirmo prisijungimo rekomendacijas seka

Sekos pavadinimas: Apžvelgti pirmo prisijungimo rekomendacijas.

Sąveikos objektai: Prisijungęs vartotojas, sistema, duomenų bazė.

Įrėmintas funkcinis reikalavimas: FR-0203.

31 lentelė. Apžvelgti pirmo prisijungimo rekomendacijas sekų diagramos paaiškinimas

užduoties numeris	17
užduoties pavadinimas	Apžvelgti pirmo prisijungimo rekomendacijas
siekiamas tikslas	Apžvelgti pirmo prisijungimo rekomendacijas
užduoties vykdymą inicijuojantis įvykis (triggeris)	Vartotojas pasireinka apžvelgti rekomendacijas
užduoties prioritetas	1-as (aukštas)
užduoties vykdymo sritis	El. komercijos rekomendacijų sistema
užduoties lygmuo	Sumarinis tikslas
pirminis aktorius	Vartotojas
antriniai aktoriai	Sistema, duomenų bazė
"prieš" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis turinti nuorodas į rekomendacijų puslapius
sėkmingos baigties "po" sąlygos	Sėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas
nesėkmingos baigties "po" sąlygos	Nesėkmingai pasiektas internetinės svetainės puslapis skirtas atvaizduoti rekomendacijas

32 lentelė. Sistemos reikalavimų matrica

Nr.	Aprašymas
SR-01	Sistema turi leisti susikurti naują paskyrą
SR-02	Sistema turi leisti prisijungti prie savo paskyros
SR-03	Sistema turi leisti pasikeisti paskyros duomenis
SR-04	Sistema turi leisti pirkti įdėti prekę į krepšelį
SR-05	Sistema turi leisti pašalinti prekę iš krepšelio
SR-06	Sistema turi leisti pirkti prekes esančias krepšelyje
SR-07	Sistema turi leisti įvertinti pirktą prekę
SR-08	Sistema turi leisti peržiūrėti populiarių prekių katalogą
SR-09	Sistema turi leisti filtruoti prekes pagal kategorijas
SR-10	Sistema turi leisti peržiūrėti prekes pagal paskyros profilį
SR-11	Sistema turi leisti peržiūrėti prekes pagal kelių paskyrų profilius
SR-12	Sistema turi leisti peržiūrėti prekes pagal paskyros duomenis

33 lentelė. Reikalavimų matrica komponentams

Nr.	Aprašymas
SDK-01	Serverio dalies komponentas turėtų gauti duomenis iš internetinės naršyklės
SDK-02	Serverio dalies komponentas turėtų perduoti duomenis internetinei naršyklei
SDK-03	Serverio dalies komponentas turėtų gauti duomenis iš Java posistemo
SDK-04	Serverio dalies komponentas turėtų užtikrinti, jog duomenys nebus prarasti
KK-01	Kontrolierio komponentas turėtų rasti prekes pagal vartotojų profilį
KK-02	Kontrolierio komponentas turėtų rasti populiariausias prekes
KK-03	Kontrolierio komponentas turėtų patikrinti iškeltus reikalavimus duomenims
KK-04	Kontrolierio komponentas turėtų kurti Cypher užklausų kalbos užklausas Neo4j duomenų bazei
KK-05	Kontrolierio komponentas turėtų gauti modelius reikalingus saveikos tarp duomenų kūrimui
KK-06	Kontrolierio komponentas turėtų gauti html puslapius ir užpildyti juos duomenimis, kuriuose pritaikytas Thymeleaf papildinys
MK-01	Modelio komponentas turėtų turėti duomenų struktūras ir metodus
MK-02	Modelio komponente esančiose klasės kintamieji atsakingi už ryšių kūrimą grafų duomenų bazėje turi turėti numatytą anotaciją
MK-03	Modelio komponente turi būti numatomos visi duomenų bazėje naudojami mazgai ir ryšiai

GVSK-01	Grafinės vartotojo sąsajos komponentas atsakingas už internetinės svetainės puslapių dizainą
GVSK-02	Grafinėje vartotojo sąsajos komponente aprašyti html puslapiai turėtų kreiptis į metodus esančias kontrolerio komponente
GVSK-03	Grafinė vartotojo sąsajos komponente aprašyti html puslapiai turėtų palaikyti produktų informacijos iš duomenų bazės įterpimą
GVSK-04	Grafinė vartotojo sąsajos komponente aprašyti html puslapiai turėtų palaikyti vartotojų informacijos iš duomenų bazės įterpimą
DBSK-01	Duomenų bazės serverio komponentas įgalina lokalią duomenų bazę

34 lentelė. Reikalavimų ryšio matrica

Reikalavimas	Grafinės vartotojo sąsajos komponentas	Duomenų bazės serverio komponentas	Serverio dalies komponentas	Modelio komponentas	Kontrolerio komponentas	Duomenų bazės serverio komponentas
SR-01	X	X	X	X	X	
SR-02	X	X	X	X	X	
SR-03	X	X	X	X	X	
SR-04	X	X	X	X	X	
SR-05	X		X		X	
SR-06	X	X	X		X	
SR-07	X	X	X	X	X	
SR-08	X	X	X	X	X	
SR-09	X	X	X	X	X	
SR-10	X	X	X	X	X	
SR-11	X	X	X	X	X	
SR-12	X	X	X	X	X	
SDK-01	X		X			
SDK-02	X		X			
SDK-03	X	X	X	X	X	
SDK-04	X		X		X	
KK-01	X	X	X	X	X	
KK-02	X	X	X	X	X	

KK-03				X	X	
KK-04					X	
KK-05				X	X	
KK-06	X				X	
MK-01				X	X	
MK-02				X	X	
MK-03				X	X	
GVSK-01	X				X	
GVSK-02	X				X	
GVSK-03	X				X	
GVSK-04	X				X	
DBSK-01			X			X

PRIEDAS 6

GitLab nuoroda į projekto repozitoriją saugančią projekto programinį kodą:
<https://studgit.vilniustech.lt/20184920/e-commerce>