

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Skrydžių bilietų paieškos sistema

Plane tickets search system

Laboratorinis darbas

Atliko: 2 kurso 4 grupės studentai
Olga Joana Šmitaitė
Martynas Talalas
Nikita Gedgaudas
Justas Lakštinis

Vilnius – 2018

TURINYS

IVADAS	2
1. REIKALAVIMAI	3
1.1. Funkciniai reikalavimai	3
1.1.1. Bendri sistemos funkciniai reikalavimai	3
1.1.2. Papildomi funkciniai reikalavimai.....	4
1.2. Nefunkciniai reikalavimai	4
1.2.1. Bendri sistemos nefunkciniai reikalavimai	4
1.2.2. Sistemos palaikymo ir veikimo reikalavimai	4
1.2.3. Reikšmių reikalavimai	5
2. STATINĖ PROGRAMŲ SISTEMOS STRUKTŪRA	6
2.1. Klasių diagrama	6
2.2. Reikalavimų - struktūrinio dalykinės srities modelio atsekamumo matrica.....	7
3. UŽDUOTYS	8
3.1. Užduočių aprašymai	9
3.2. Reikalavimų - užduočių atsekamumo matrica	17
4. TECHNINĖ ANALIZĖ	19
4.1. Komponentinė sandara.....	19
4.2. Diegimas.....	19
5. TESTAVIMO PLANAS IR SCENARIJAI	21
5.1. Skrydžių paieškos langas	21
5.2. Rezultatų filtravimas	23
5.3. Rezultatų rūšiavimas	24
5.4. Skrydžių paieškos rezultato informacijos peržiūra	25
5.5. Nukreipimas į bilietų platintojo puslapį.....	25
5.6. Būsimų skrydžių peržiūra	25
5.7. Būsimo skrydžio informacijos peržiūra	26
IŠVADOS IR REZULTATAI	27
PAKEITIMŲ SĄRAŠAS	28
ŠALTINIAI	29

Įvadas

Šis dokumentas papildo pirmajį „Skrydžių bilietų paieškos sistemos“ dokumentą. Antroje versijoje yra atliekama robastišumo analizė, preliminari projekto peržiūra ir apibrežiama techninė sistemos architektūra. Kaip ir pirmojoje dokumento versijoje, dokumente nurodomi sistemos reikalavimai, Statinė programų sistemos struktūra, užduotys ir, papildomai prisideda, techninė analizė. Reikalavimai išskirstyti į funkcinius ir nefunkcinius. Funkciniai reikalavimai nurodo pagrindines sistemos funkcijas, o nefunkciniai - nurodo kaip tas funkcijas sistema turi atliskti. Struktūrinės dalykinės srities modelis yra pateikiamas UML klasių diagramomis kartu su žodynais - sąrašu esybių su jų aprašymais. Užduočių skyrius aprašo sistemos atliekamas užduotis ir pasitelkus robastiškuo diagramomis parodyta kaip ir kokios sistemos dalys jas atliks. Techninės analizės skyriuje yra analizuojami techniniai sistemos aspektai - sistemos komponentai ir sistemos diegimas.

1. Reikalavimai

Šiame skyriuje išvardyti programų sistemos funkciniai ir nefunkciniai reikalavimai.

1.1. Funkciniai reikalavimai

1.1.1. Bendri sistemos funkciniai reikalavimai

FR1. Sistema turi atlikti skrydžių į vieną pusę paiešką pagal:

- išvykimo datos intervalą (nuo/iki),
- skrydžio kainos intervalą (nuo/iki),
- išvykimo ir atvykimo vietas,

FR2. Sistema turi atlikti skrydžių į dvi puses paiešką pagal:

- išvykimo ir atvykimo datų intervalus (nuo/iki),
- skrydžio kainos intervalą (nuo/iki),
- išvykimo ir atvykimo vietas,

FR3. Sistema turi turėti galimybę rūšiuoti ir/arba filtruoti paieškos rezultatus.

FR3.1. Sistema rūšiuoja skrydžius pagal:

- greičiausius,
- pigiausius,
- optimaliausius (žr. **NFR20**.).

FR3.2. Sistema filtruoja skrydžius pagal:

- aviakompanijas,
- persėdimų skaičių.

FR4. Pasirinkus skrydį sistema turi parodyti detalią skrydžio informaciją, kurioje yra:

- skrydžio išvykimo ir atvykimo datos ir laikai,
- skrydžio išvykimo ir atvykimo šalys, miestai ir oro uosto trumpiniai,
- skrydžio trukmė,
- skrydžio kaina.

FR5. Sistemoje turi būti galimybė vartotojui įsigyti skrydžio(-ų) bilietą, nukreipiant jį į atitinkamą aviakompanijos puslapį.

FR6. Sistema atitinkamame lange vartotojui turi parodyti būsimus skrydžius, į kuriuos jis įsigijo bilietus.

1.1.2. Papildomi funkciniai reikalavimai

FR7. Vartotojui norint įvesti datą, sistema turi pateikti mažą kalendorių arba leisti vartotojui sava-rankiškai įvesti datą.

FR8. Paieškos formoje nieko neįvedus arba įvedus netinkamai, sistema turi pateikti vartotojui pa-vyzdį, kaip turėtų būti suvesti duomenys.

1.2. Nefunkciniai reikalavimai

1.2.1. Bendri sistemos nefunkciniai reikalavimai

NFR1. Sistema turi būti lietuvių kalba.

NFR2. Programų sistema turi būti susieta su palaikomų aviakompanijų bilietų pirkimo svetainėmis.

NFR3. Programų sistema turi turėti prieigą prie aviakompanijų duomenų bazii, kuriose saugomi tvarkaraščiai, naudotojai, įvykiai ir užrašai.

NFR4. Programų sistema negali pažeisti Europos Žmogaus teisių ir pagrindinių laisvių apsaugos konvencijos

1.2.2. Sistemos palaikymo ir veikimo reikalavimai

NFR5. Sistema turi palaikyti tas operacines sistemas, kurios palaiko:

- Mozilla Firefox (nuo 58 versijos)
- Google Chrome (nuo 64 versijos)
- Microsoft Internet Explorer (nuo 11 versijos)
- Microsoft Edge (nuo 41 versijos)
- Apple Safari (nuo 11 versijos)

NFR6. Sistema turi palaikyti HTTPS standartą.

NFR7. Programų sistemos palaikymui yra būtina bent 426 x 240 (240p 16:9) ekrano rezoliucija ir ekrano dydis turi būti bent 4“.

NFR8. Įvykus sistemos sutrikimui, vartotojo darbo funkcionalumas turi būti atkurtas penkių minu-čių laikotarpyje.

NFR9. Didžiausia leistina programų sistemos apkrova yra 5000 vartotojų, prisijungusių vienu metu.

NFR10. Paieškos rezultatai pateikiami ne lėčiau kaip per 20 sekundžių.

NFR11. Programa turi būti įdiegta ir paleista serveryje, kuris veikia visomis metų dienomis 24 valan-das per parą.

NFR12. Sisteminės klaidos turi būti išsaugojamos klaidų žurnale, kuris turi būti tikrinamas bent kartą per vieną darbo dieną

NFR13. Praplėtus programų sistemos funkcionalumą būtina patikrinti atnaujinimus prieš leidžiant jais naudotis vartotojams

1.2.3. Reikšmių reikalavimai

NFR14. Dados įvedamos ir vaizduojamos DD/MM/YYYY formatu.

NFR15. Išvykimo data turi būti anksčiau arba sutapti su atvykimo data.

NFR16. Skrydžio kainos turi būti teigiamos ir rodomas eurais.

NFR17. Mažiausia skrydžio kaina turi būti mažesnė už didžiausią skrydžio kainą.

NFR18. Išvykimo ir atvykimo vietos negali sutapti.

NFR19. Vieta vaizduojama arba vedama formatu „<miestas> <oro uosto trijų raidžių trumpinys> <(nebūtina), šalis>“.

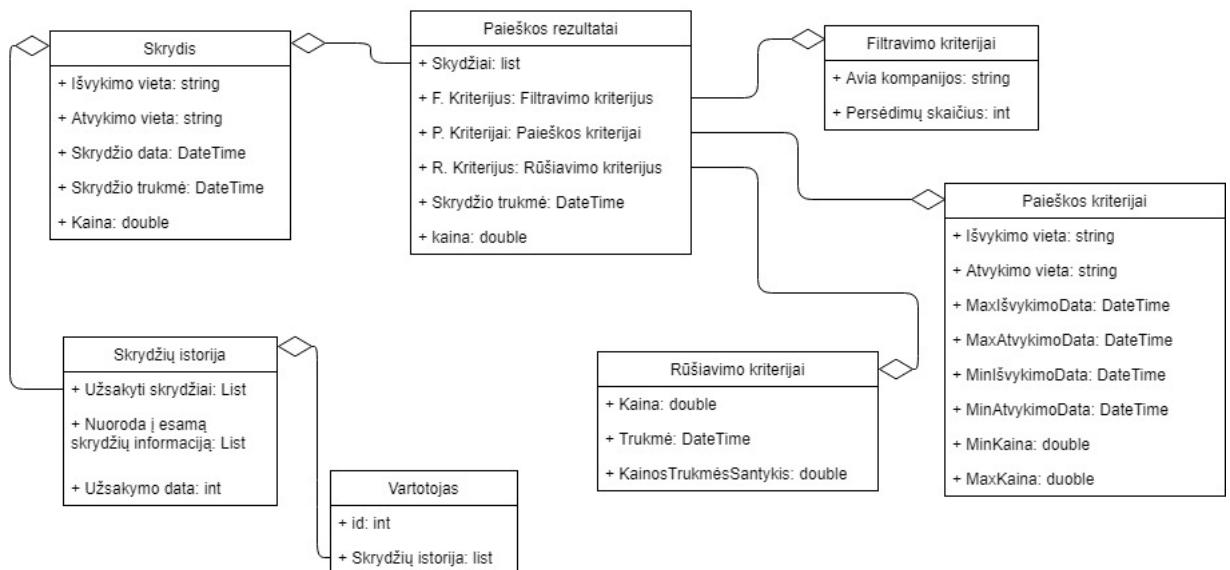
NFR20. Optimalesnis skrydis yra tas kurio kainos ir skrydžio trukmės santykis yra mažesnis.

2. Statinė programų sistemos struktūra

Šiame skyriuje pateikiamas struktūrinis nagrinėjamos dalykinės srities modelis. Modelis pateikiamas UML klasų diagramomis kartu su žodynu (žr. skyrius „Klasų diagrama“) - sąrašu esybių su jų aprašymais. Skyriaus pabaigoje pateikiamas reikalavimų - struktūrinio dalykinės srities modelio atsekamumo matrica (žr. skyrius „Reikalavimų - struktūrinio dalykinės srities modelio atsekamumo matrica“).

2.1. Klasų diagrama

Čia pateikiama ir aprašoma sistemos klasų diagrama ir pateikiamas jos žodynas



1 pav. Klasų diagrama

- E1. Paieškos rezultatai - aprašo skrydžių paieškos rezultatus pritaikius paieškos kriterijus.
- E2. Skrydis - detaliai aprašo skrydžio informaciją. Skrydžio informacija susideda iš išvykimo vienos, atvykimo vietas, skrydžio datos, skrydžio trukmės ir kainos.
- E3. Skrydžių istorija - aprašo vartotojo užsakyty, įvykusią bei būsimų skrydžių informaciją.
- E4. Filtravimo kriterijai - aprašo vartotojo nustatytus filtravimo kriterijus. Vartotojas gali pasirinkti norimas avia kompanijas ir pageidaujamų persėdimų skaičių
- E5. Paieškos kriterijai - aprašo vartotojo pasirinktus skrydžio bilietų paieškos kriterijus. Vartotojas gali nustatyti išvykimo vietą, atvikimo vietą, maksimalią ir minimalią išvykimo bei atvykimo datą, maksimalią bei minimalią kainą.
- E6. Rūšiavimo kriterijai - aprašo vartotojo pasirinktus skrydžių rūšiavimo kriterijus. Kriterijus sudaro kaina, skrydžio trukmė bei kainos ir trukmės santykis(optimalumas).
- E7. Vartotojas - asmuo naviguojantis skrydžių paieškos platformoje ir ieškantis reikiamų skrydžių.

2.2. Reikalavimų - struktūrinio dalykinės srities modelio atsekamumo matrica

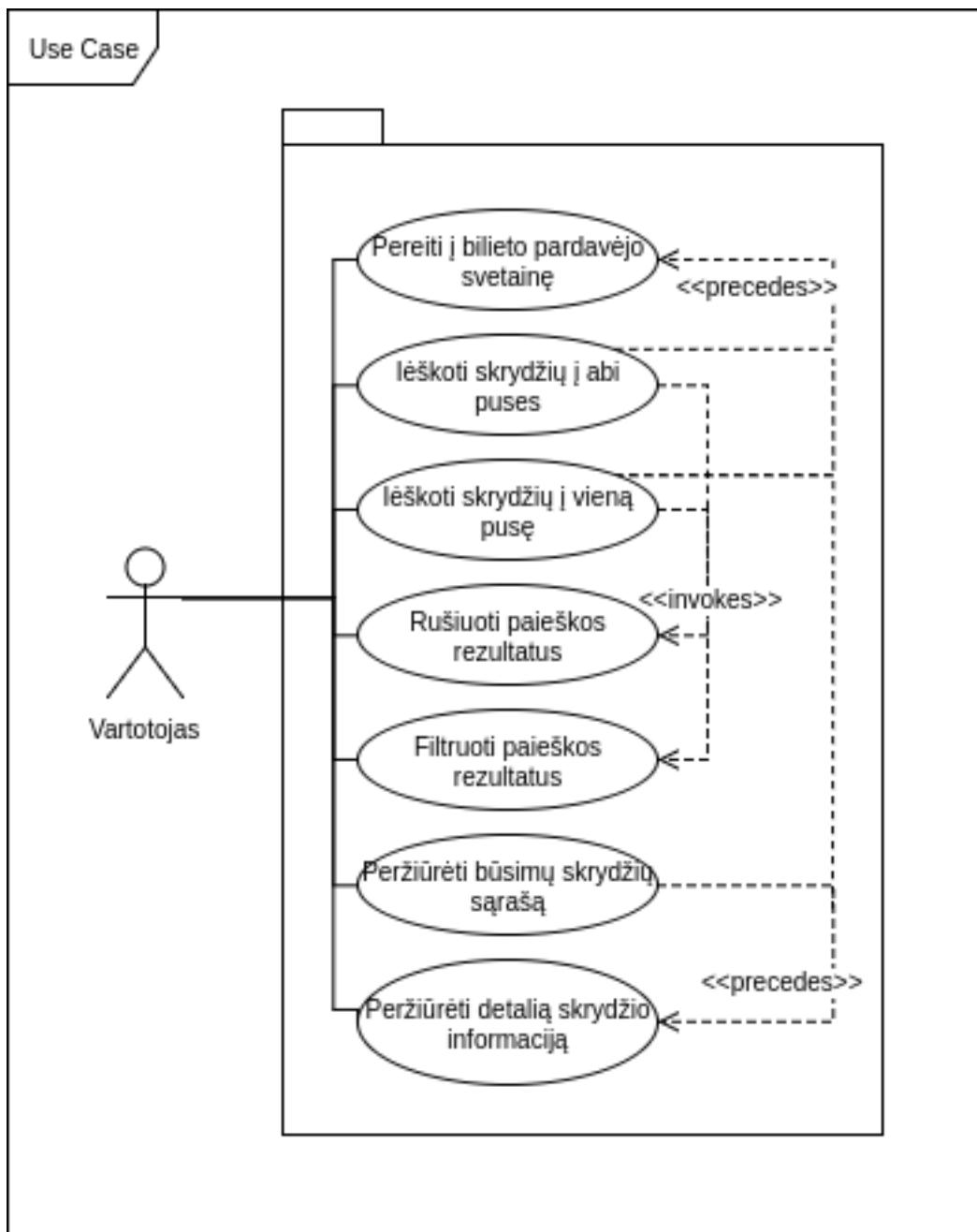
Šiame skyriuje yra pateikiama reikalavimų - struktūrinės dalykinės srities modelio atsekamumo matrica (žr. 1 lentelę). Matricoje pavaizduotos dalykinės srities modelio esybės bei sistemos funkciniai reikalavimai. Reikalavimų - struktūrinio dalykinės srities modelio atsekamumo matrica padeda susieti dalykinės srities modelio esybes su sistemos funkciniais(arba ne funkciniais) reikalavimais.

1 lentelė. Reikalavimų - struktūrinio dalykinės srities modelio atsekamumo matrica

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8
E1	X	X	X	X				
E2	X	X	X	X	X	X		
E3						X		
E4			X					
E5	X	X					X	X
E6			X					
E7	X	X	X	X	X	X	X	X

3. Užduotys

Šiame skyriuje pateikiamos sistemos atliekamos užduotys. Visos sistemos atliekamos užduotys yra atvaizduojamos užduočių diagramoje (žr. 2pav.). Atvaizdavus užduotis užduočių diagramoje, kiekviena užduotis yra detaliai aprašoma. Prie užduoties aprašymo yra pateikiami ir alternatyvūs scenarijai. Užduotims aprašyti taip pat yra naudojami ir kuriamos sistemos eskizai, leidžiantys geriau suprasti ir įsivaizduoti sistemos atliekamas užduotis. Skyriaus pabaigoje yra pateikama ir reikalavimų - užduočių atsekamumo matrica.



2 pav. Užduočių diagrama

3.1. Užduočių aprašymai

U1. Ieškoti skrydžių į vieną pusę

Vartotojas įveda išvykimo, atvykimo miestus. Pagal pageidavimą, įveda tinkamą išvykimo laiko intervalą, kainos intervalą. Tada vartotojas paspaudžia ant mygtuko „Ieškoti“ (žr. 16 pav.) ir yra nukreipiamas į paieškos rezultatų langą, kur jam, lentelės pavidalu rodomi surasti skrydžiai pagal vartojojo įvestus kriterijus (žr. 4 pav.).



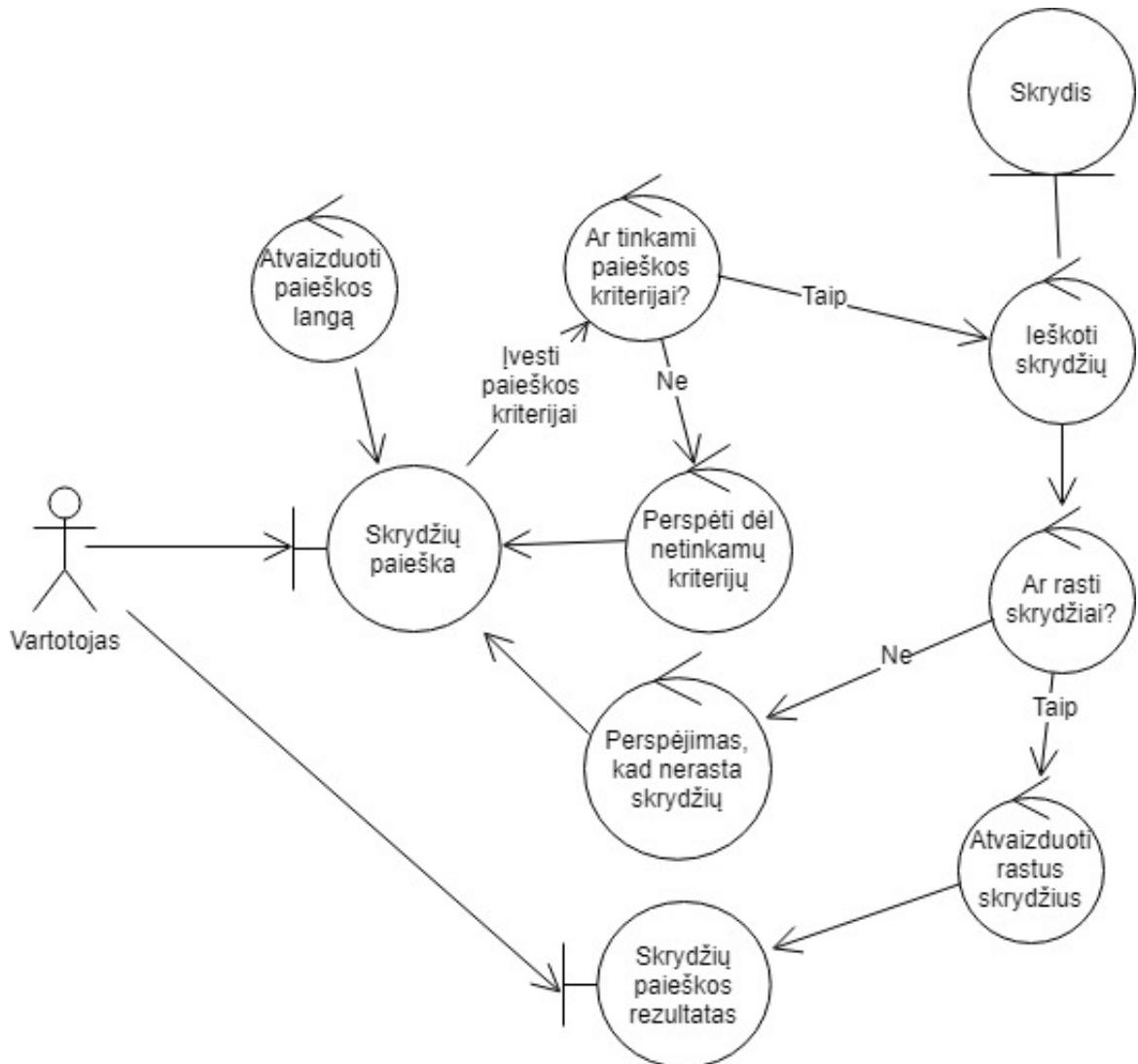
3 pav. Pagrindinis puslapis

Greičiausi		Pigiausi		Optimaliausi	
Išvykimas	Atvykimas	Pirmyn	Atgal	Kaina	Persėdimai
2018-01-01 00:00	2018-01-01 00:00			100	1
2018-01-01 00:00	2018-01-01 00:00			100	1
2018-01-01 00:00	2018-01-01 00:00			100	1
2018-01-01 00:00	2018-01-01 00:00			100	1
2018-01-01 00:00	2018-01-01 00:00			100	1
2018-01-01 00:00	2018-01-01 00:00			100	1
2018-01-01 00:00	2018-01-01 00:00			100	1
2018-01-01 00:00	2018-01-01 00:00			100	1
2018-01-01 00:00	2018-01-01 00:00			100	1
2018-01-01 00:00	2018-01-01 00:00			100	1

4 pav. Paieškos rezultatai

Alternatyvūs scenarijai:

- Jei vartotojas neįveida atvykimo ir/ar išvykimo miestą, jam paspaudus ant mygtuko „Ieškoti“ jis nėra nukreipiamas į rezultatų langą. Išvykimo ar atvykimo miesto įvedimo laukas su neįvesta reikšme (arba abu), paženklinami raudonai.



5 pav. Paieškos robastiškumo diagrama

U2. Ieškoti skrydžių į abi puses

Vartotojas įveda išvykimo, atvykimo miestus. Pažymi varnele, kad renkasi skrydį atgal. Pagal pageidavimą, įveda išvykimo ir atvykimo tinkamų laikų intervalus, kainos intervalą. Tada vartotojas paspaudžia ant mygtuko „Leškoti“ (žr. 6 pav.) ir yra nukreipiamas į paieškos rezultatų langą, kur jam, lentelės pavidalu rodomi surasti skrydžiai pagal vartojojo įvestus kriterijus (žr. 7 pav.).



6 pav. Pagrindinis puslapis

Greičiausiai		Pigiausiai		Optimaliausiai	
		Pirmyn	Atgal		
Persėdimai	<input type="checkbox"/> 1 <input type="checkbox"/> 2	Išvykimas	Atvykimas	Kaina	Persėdimai
		2018-01-01 00:00	2018-01-01 00:00	100	1
		2018-01-01 00:00	2018-01-01 00:00	100	1
		2018-01-01 00:00	2018-01-01 00:00	100	1
		2018-01-01 00:00	2018-01-01 00:00	100	1
		2018-01-01 00:00	2018-01-01 00:00	100	1
		2018-01-01 00:00	2018-01-01 00:00	100	1
		2018-01-01 00:00	2018-01-01 00:00	100	1
		2018-01-01 00:00	2018-01-01 00:00	100	1
Kompanija	<input type="checkbox"/> Lufthansa <input type="checkbox"/> American Airlines				

7 pav. Paieškos rezultatai

Alternatyvūs scenarijai:

- Jei vartotojas nejveida atvykimo ir/ar išvykimo miestą, jam paspaudus ant mygtuko „Ieškoti“ jis nėra nukreipiamas į rezultatų langą. Išvykimo ar atvykimo miesto įvedimo laukas su nejesta reikšme (arba abu), paženklinami raudonai.

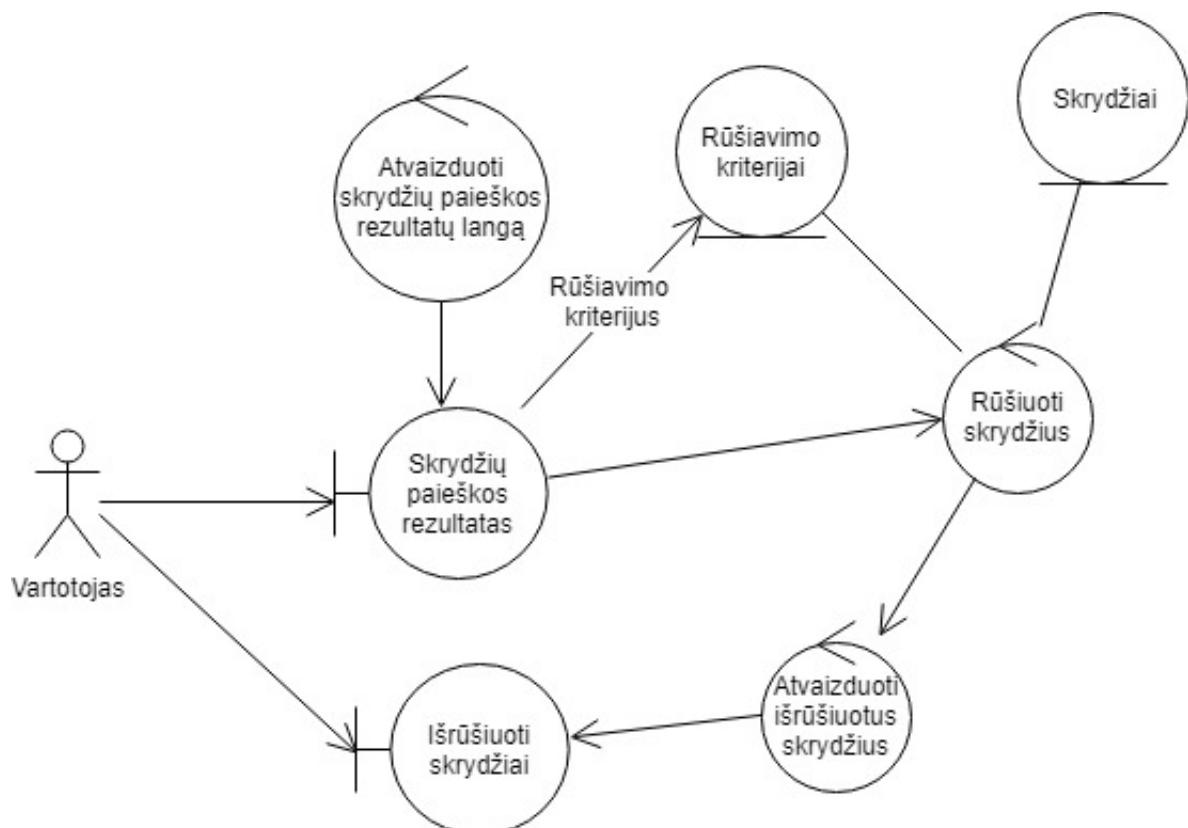
U3. Rušiuoti paieškos rezultatus

Paieškos rezultatų lange, vartotojas pasirenka vieną iš trijų rūšiavimo būdus atitinkančių mygtukų: „Greičiausias“, „Pigiausias“ arba „Optimalius“ (žr. 7pav., 4 pav.). Paspaudus ant

vieno iš jų, skrydžių paieškos rezultatai yra surūšiuojami pagal pasirinktą rušiavimo būdą.

Alternatyvūs scenarijai:

- Surušiavus rezultatus tam tikru pasirinktu būdu, vartotojas paspaudžia ant jau pasirinktą rūšiavimo būdą atitinkančio mygtuko. Rezultatai lieka surūšiuoti taip pat kaip ir prieš spaudžiant.



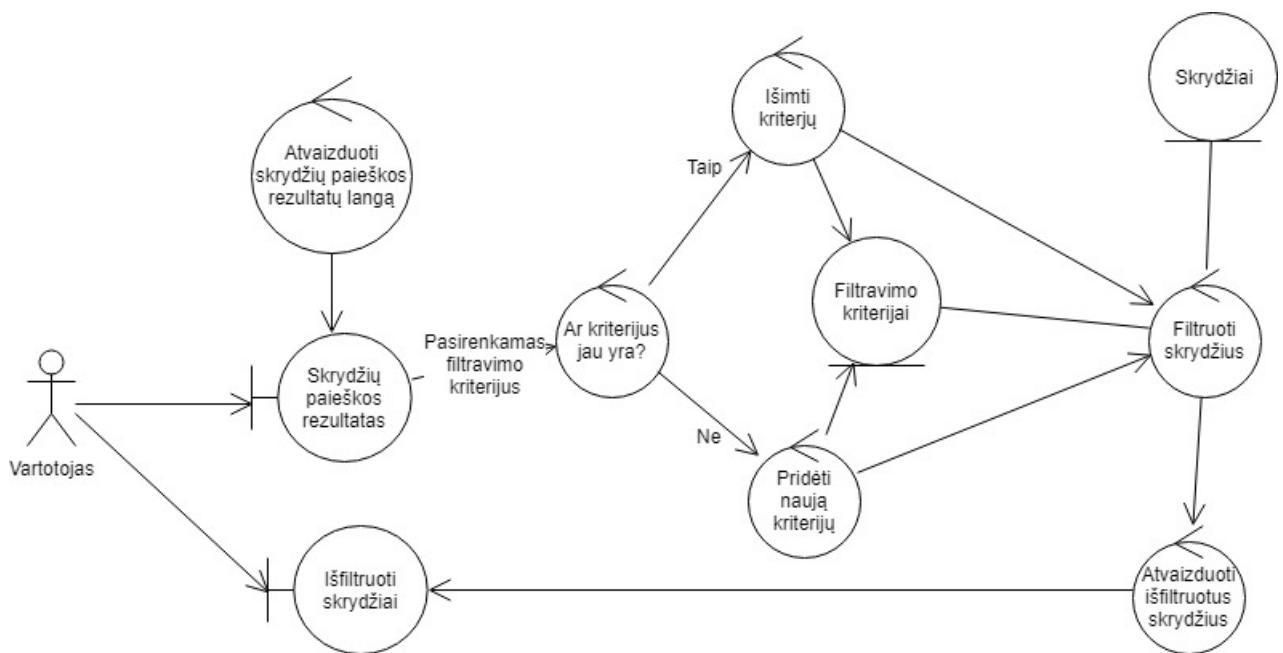
8 pav. Rušiavimo robastiškumo diagrama

U4. Filtruoti paieškos rezultatus

Paieškos rezultatų lange, vartotojas pažymi jį dominantų persėdimų skaičių ir skrydžių bendrovę (žr. 7pav., 4pav.). Vartotojui pažymėjus konkrečią reikšmę ar to, ar ano, paieškos rezultatai yra iš karto atnaujinami, vaizduojant tik tuos skrydžius, kurie tenkina filtrų reikšmes.

Alternatyvūs scenarijai:

- Vartotojui pakartotinai paspaudus ant jau prieš tai pažymėta filtro reikšmės, rezultatai iš karto atsinaujina, o konkretus pažymėtas filtras nuimamas. Kai visi filtri yra nuimti, rezultatai atsivaizduoja netaikant jiems jokio filtravimo.



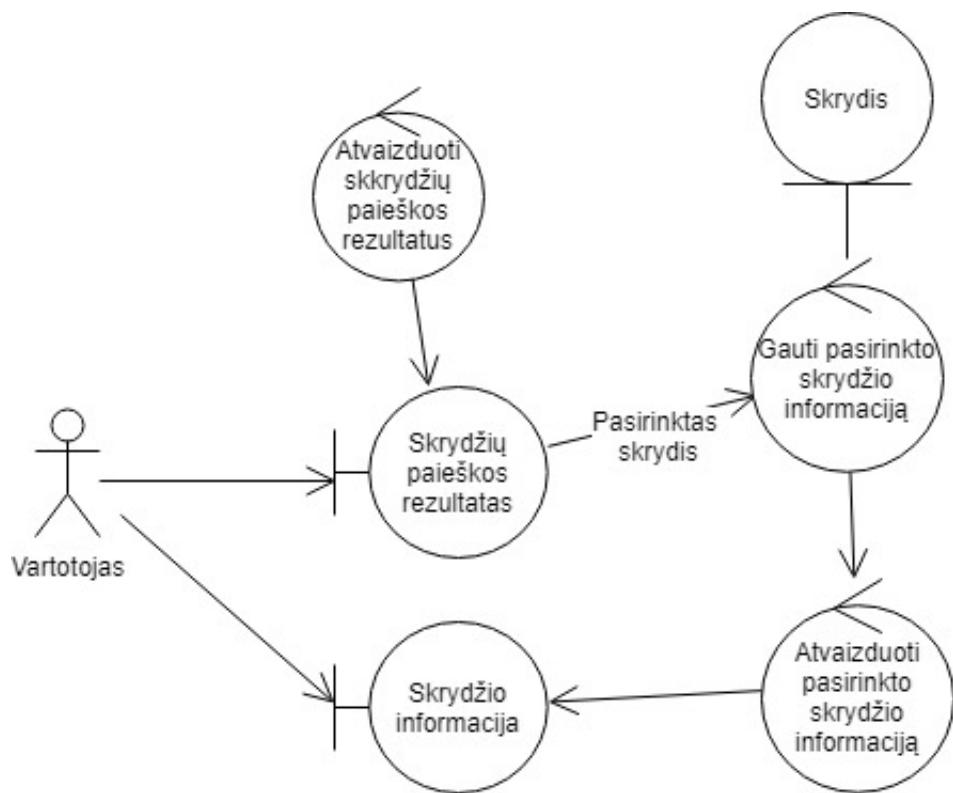
9 pav. Filtravimo robastiškumo diagramma

U5. Peržiūrėti detalią paieškos rezultato informaciją

Vartotojas atlieka skrydžių paiešką. Spaudžia ant pasirinkto skrydžio iš sąrašo ir yra atidaramos dialogas su detlia skrydžio informacija (žr. 10pav.).

Greičiausiai		Išvykimas		Optimaliausi	
Persėdimai	<input type="checkbox"/> 1 <input checked="" type="checkbox"/> 2	Išvykimas	Pirmas	Algal	
		2018-01-01 00:00	2018-01-01 00:00		Kaina
		2018-01-01 00:00	Atvykimas		Persėdimai
		2018-01-01 00:00	2018-01-01 00:00		100
			Berlynas, Schoenefeld (SXF), Vokietija		1
Kompanija	<input type="checkbox"/> Lufthansa <input checked="" type="checkbox"/> American Airlines	Vilnius (VNO), Lietuva	2018-01-01 00:00		100
		2018-01-01 00:00	2018-01-01 00:00		1
		2018-01-01 00:00	Skrydžio trukmė: 4val 25min		100
		2018-01-01 00:00	2018-01-01 00:00		1
		2018-01-01 00:00	Kaina: 100 eur		100
		2018-01-01 00:00	2018-01-01 00:00		1
		2018-01-01 00:00	Kompanija: Lufthansa		100
		2018-01-01 00:00	2018-01-01 00:00		1
		2018-01-01 00:00	Persėdimų skaičius: 1		100
		2018-01-01 00:00	2018-	PIRKTI	1
		2018-01-01 00:00	2018-01-01 00:00		100
		2018-01-01 00:00	2018-01-01 00:00		1
		2018-01-01 00:00	2018-01-01 00:00		100
		2018-01-01 00:00	2018-01-01 00:00		1

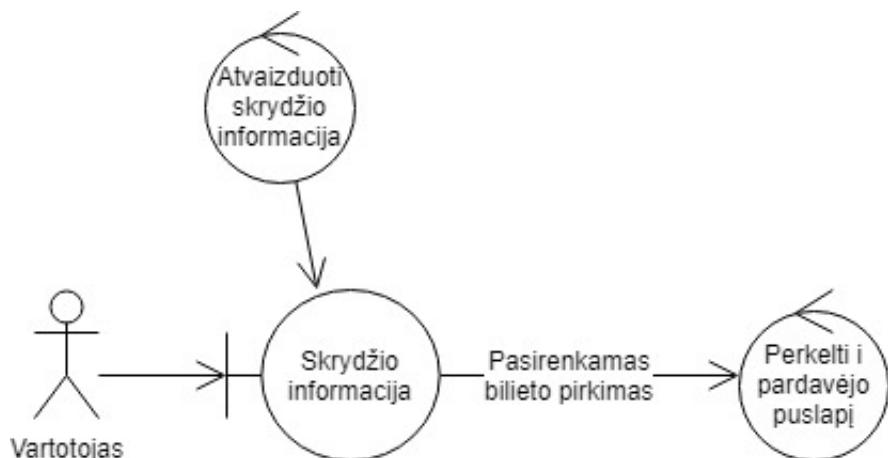
10 pav. Paieškos rezultato detalės



11 pav. Detalios skrydžio informacijos peržiūros robastiškumo diagrama

U6. Pereiti į bilieto pardavėjo svetainę

Vartotojas atlieka skrydžių paiešką. Spaudžia ant pasirinkto skrydžio. Atsidariusiame dialoge spaudžia mygtuką „Pirkti“ (žr. 10pav.) ir yra nukeliamas į bilieto pardavėjo puslap.



12 pav. Perėjimo į bilietu pardavėjo svetainę robastiškumo diagrama

U7. Peržiūrėti būsimų skrydžių sąrašą

Vartotojas spaudžia ant mygtuko „Būsimi skrydžiai“. Atidaromas skrydžių, į kuriuos vartotojas yra įsigijęs bilietus sąrašas (žr. 13pav.).

FLYONLINE

BŪSIMI SKRYDŽIAI

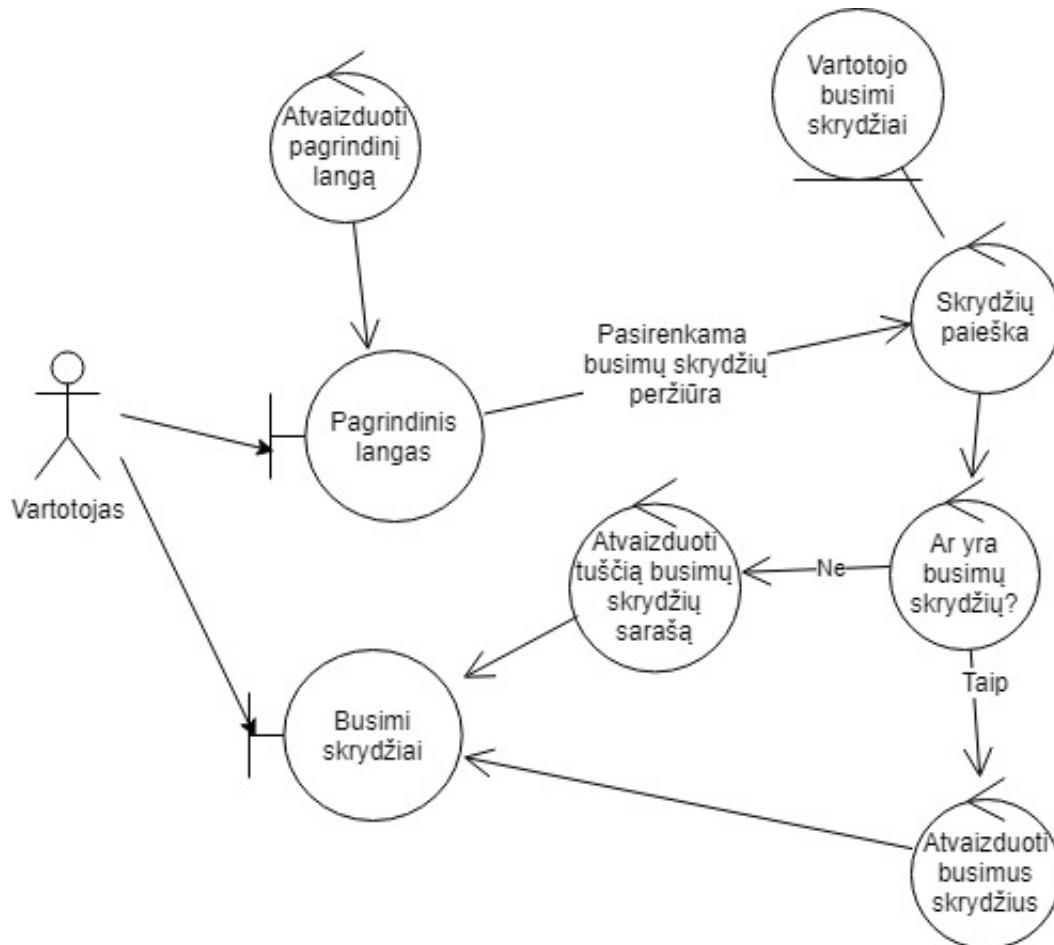
Skrydis	Išvykimas	Atvykimas
Vilnius - Berlynas	2018-01-01 00:00	2018-01-01 00:00
Vilnius - Berlynas	2018-01-01 00:00	2018-01-01 00:00
Vilnius - Berlynas	2018-01-01 00:00	2018-01-01 00:00
Vilnius - Berlynas	2018-01-01 00:00	2018-01-01 00:00
Vilnius - Berlynas	2018-01-01 00:00	2018-01-01 00:00



13 pav. Būsimų skrydžių sąrašas

Alternatyvūs scenarijai:

- Jei būsimų skrydžių sąrašas yra tuščias, sąrašo vietoje vartotojui parodoma žinutė „Sąrašas yra tuščias“.



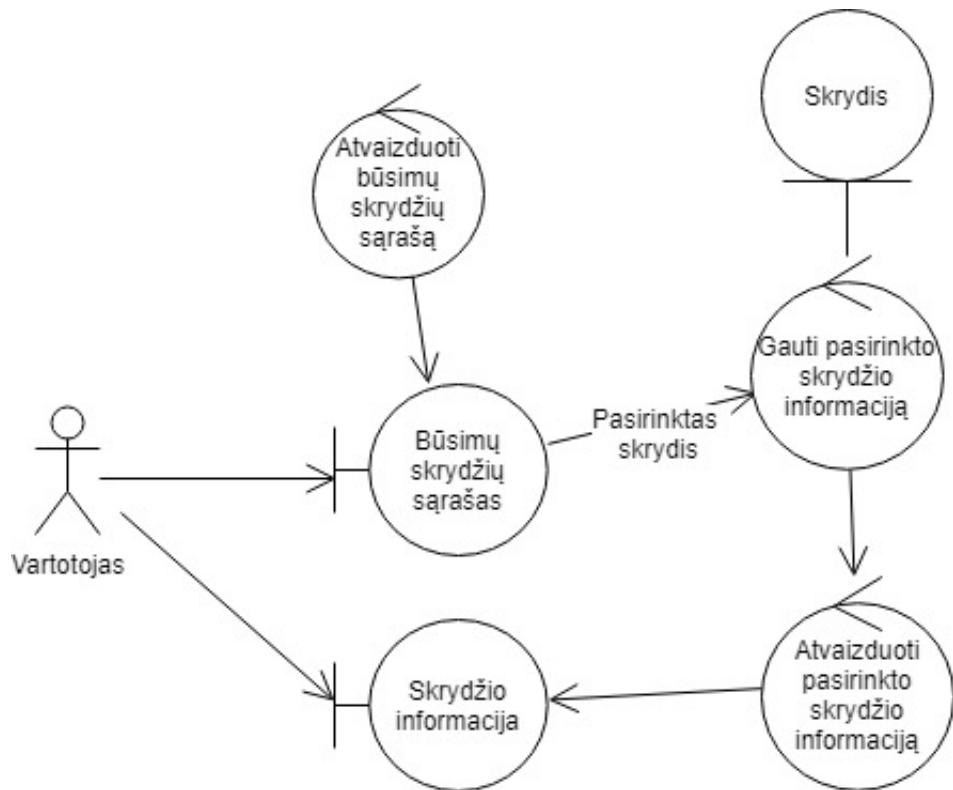
14 pav. Būsimų skrydžių peržiūros robastiškumo diagrama

U8. Peržiūrėti detalią būsimo skrydžio informaciją

Vartotojas atidaro būsimų skrydžių sąrašą. Spaudžia ant pasirinkto skrydžio iš sąrašo ir yra atidaromos dialogas su išsamia skrydžio informacija (žr. 15 pav.).

BŪSIMI SKRYDŽIAI			
Skrydis	Išvykimas	Atvykimas	X
Vilnius - Berlynas	2018-01-01 00:00	2018-01-01 00:00	
	Išvykimas:		
	Berlynas, Schoenefeld (SXF), Vokietija		
	2018-01-01 00:00		
Vilnius - Berlynas	2018-01-01 00:00	2018-01-01 00:00	
	Atvykimas:		
	Vilnius (VNO), Lietuva		
	2018-01-01 00:00		
Skrydžio trukmė: 4val 25min			

15 pav. Būsimo skrydžio informacija



16 pav. Detalios būsimo skrydžio informacijos robastiškumo diagrama

3.2. Reikalavimų - užduočių atsekamumo matrica

Šiame poskyryje pateikiama reikalavimų - užduočių atsekamumo matrica. Matricoje pavaizduotos užduotys(Use cases) bei reikalavimai. Ši matrica padeda atsekti kurios užduotys įgyvendina atitinkamus reikalavimus (žr. 2pav.).

2 lentelė. Reikalavimų - užduočių atsekamumo matrica

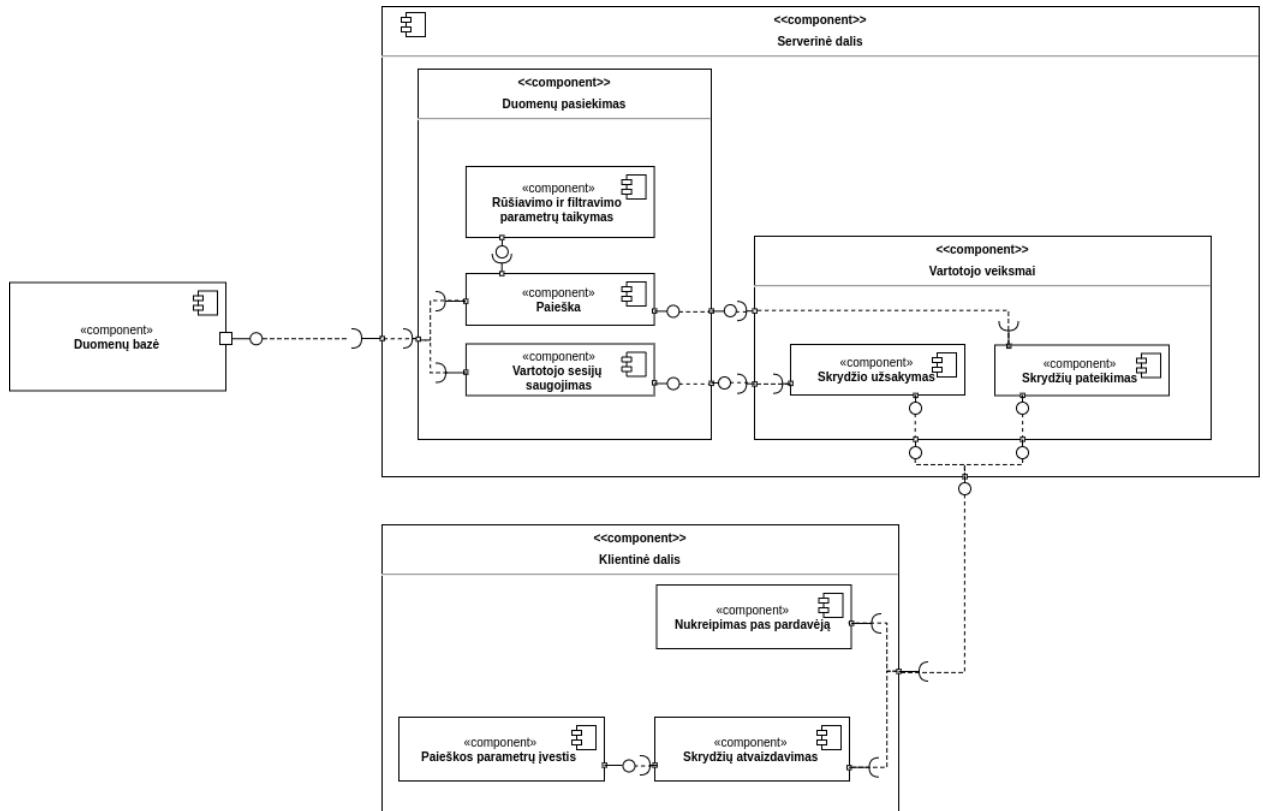
	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8
U1	X						X	X
U2		X					X	X
U3			X					
U4			X					
U5				X				
U6					X			
U7						X		
U8				X				

4. Techninė analizė

Šis skyrius aprašo siūlomą techninę architektūrą, sukurtą atsižvelgiant į pastabas, iškeltas atliekant robastiškumo analizę bei preliminarią projekto peržiūrą.

4.1. Komponentinė sandara

Čia pateikiama sistemos komponentinė sandara ir jos aprašymas

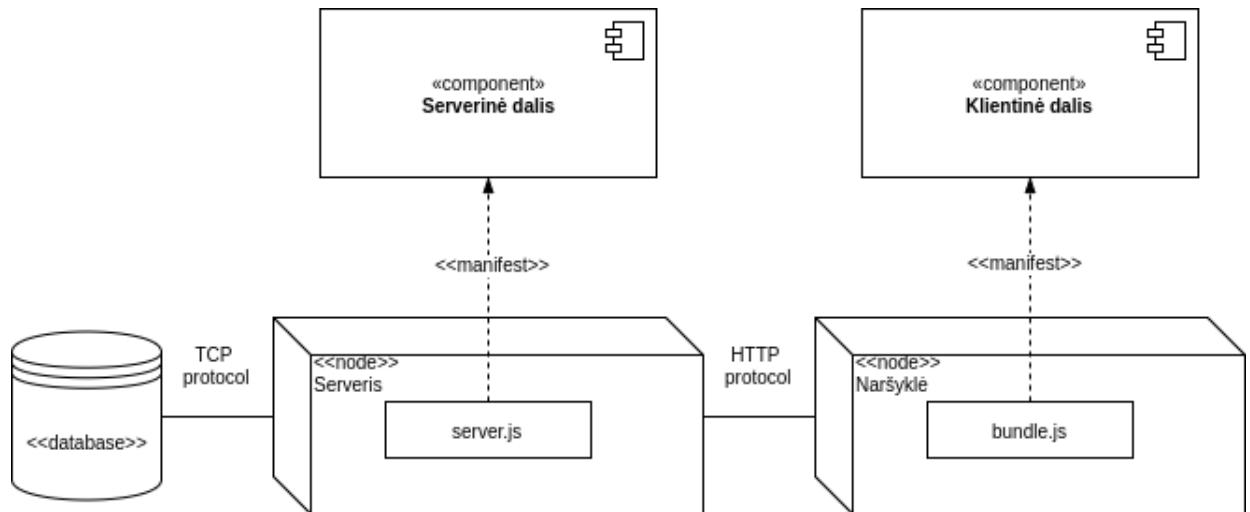


17 pav. Sistemos komponentinė sandara (UML Komponentų diagrama)

Visą sistemą galima suskaidyti į tris pagrindinius komponentus: Klientinę dalį, Serverinę dalį bei Duomenų bazę (žr. 17 pav.). Šių trijų komponentų rolės atitinka „3-tier“ architektūros sandarą - Atvaizdavimo sluoksnį, Verslo taisyklių sluoksnį bei Duomenų sluoksnį. Smulkesnis skaidymas į komponentus yra pagal funkcinius sistemos aspektus. Tai padaryta siekiant užtikrinti vykdomojo kodo testavimo ir bendrai palaikymo paprastumą, bei suteikia galimybę paprastesniams plečiamumui bei lankstensiam diegimui.

4.2. Diegimas

Čia aprašomas sistemos diegimas



18 pav. Sistemos diegimas (UML Diegimo diagramma, UML Manifestacijos diagrama)

Sistemos komponentų diegimas yra trivialus, pilnai dengiantis užsakovo dabartinius poreikius. Klasikinis kliento-serverio sistemos diegimas leidžia lengvai susieti įrangos kūrimo procesą su pastoviu atnaujinimui (Continuous Integration), be to, minimizuoja tikimybę sisteminių klaidų dėl tinklo ar diegimo kliūčių. Kylant poreikiui, šią sistemą lengva klasterizuoti, kas garantuoja stabilumą ir nebrangų palaikomumą (žr. 18pav.).

5. Testavimo planas ir scenarijai

5.1. Skrydžių paieškos langas

Reikalavimai testui: Atidarytas skrydžių paieškos langas.

Scenarijus:

1. Vartotojas suveda skirtingus išvykimo ir atvykimo miestus.
2. Suveda būsimą kelionės pirmyn datos intervalą („Data nuo“ yra ankstesnė arba lygi „Datai iki“).
3. Spaudžia „Ieškoti“.

Rezultatas: Atsidaro skrydžių paieškos rezultatų langas, kuriame rodomas paieškos kriterijus atitinkančiu skrydžių į vieną pusę sąrašas.

Scenarijus:

1. Vartotojas suveda tesingus paieškos kriterijus, pagal kuriuos rezultatų nėra.
2. Spaudžia „Ieškoti“.

Rezultatas: Parodomas pranešimas, kad rezultatų nėra.

Scenarijus:

1. Vartotojas pažymi „Atgal“.

Rezultatas: Kelionės atgal datos intervalo įvedimo laukai tampa redaguojami.

Scenarijus:

1. Vartotojas suveda skirtingus išvykimo ir atvykimo miestus.
2. Suveda būsimą kelionės pirmyn datos intervalą („Data nuo“ yra ankstesnė arba lygi „Datai iki“).
3. Pažymi „Atgal“.
4. Suveda būsimą kelionės atgal datos intervalą („Data nuo“ yra ankstesnė arba lygi „Datai iki“).
5. Spaudžia „Ieškoti“.

Rezultatas: Atsidaro skrydžių paieškos rezultatų langas, kuriame rodomas paieškos kriterijus atitinkančiu skrydžių į abi puses sąrašas.

Scenarijus:

1. Vartotojas suveda vienodus išvykimo ir atvykimo miestus.
2. Suveda būsimą kelionės pirmyn datos intervalą („Data nuo“ yra ankstesnė arba lygi „Datai iki“).
3. Spaudžia „Ieškoti“.

Rezultatas: Miestų įvedimo laukai paryškinami raudonai.

Scenarijus:

1. Vartotojas suveda išvykimo miestą.
2. Suveda būsimą kelionės pirmyn datos intervalą („Data nuo“ yra ankstesnė arba lygi „Datai iki“).
3. Spaudžia „Ieškoti“.

Rezultatas: Atvykimo miesto įvedimo laukas paryškinamas raudonai.

Scenarijus:

1. Vartotojas suveda atvykimo miestą.
2. Suveda būsimą kelionės pirmyn datos intervalą („Data nuo“ yra ankstesnė arba lygi „Datai iki“).
3. Spaudžia „Ieškoti“.

Rezultatas: Išvykimo miesto įvedimo laukas paryškinamas raudonai.

Scenarijus:

1. Vartotojas suveda būsimą kelionės pirmyn datos intervalą („Data nuo“ yra ankstesnė arba lygi „Datai iki“).
2. Spaudžia „Ieškoti“.

Rezultatas: Miestų įvedimo laukai paryškinami raudonai.

Scenarijus:

1. Vartotojas suveda skirtingus išvykimo ir atvykimo miestus.
2. Suveda būsimą kelionės pirmyn datos intervalą („Data nuo“ yra vėlesnė už „Datą iki“).
3. Spaudžia „Ieškoti“.

Rezultatas: Skrydžio pirmyn datos intervalo įvedimo laukai paryškinami raudonai.

Scenarijus:

1. Vartotojas suveda skirtingus išvykimo ir atvykimo miestus.
2. Spaudžia „Ieškoti“.

Rezultatas: Skrydžio pirmyn datos intervalo įvedimo laukai paryškinami raudonai.

Scenarijus:

1. Vartotojas suveda skirtingus išvykimo ir atvykimo miestus.
2. Suveda būsimą kelionės pirmyn datos intervalą („Data nuo“ yra ankstesnė arba lygi „Datai iki“).
3. Pažymi „Atgal“.
4. Suveda būsimą kelionės atgal datos intervalą („Data nuo“ yra vėlesnė už „Datai iki“).
5. Spaudžia „Ieškoti“.

Rezultatas: Skrydžio atgal datos intervalo įvedimo laukai paryškinami raudonai.

Scenarijus:

1. Vartotojas suveda skirtingus išvykimo ir atvykimo miestus.
2. Suveda būsimą kelionės pirmyn datos intervalą („Data nuo“ yra ankstesnė arba lygi „Datai iki“).
3. Pažymi „Atgal“.
4. Spaudžia „Ieškoti“.

Rezultatas: Skrydžio atgal datos intervalo įvedimo laukai paryškinami raudonai.

5.2. Rezultatų filtravimas

Reikalavimai testui: Atidarytas skrydžių paieškos rezultatų langas.

Scenarijus:

1. Vartotojas pasirenka filtruoti pagal persėdimų skaičių.
2. Spaudžia „Filtruoti“.

Rezultatas: Rodomi paieškos rezultatai tik su pasirinktu persėdimu skaičiumi.

Scenarijus:

1. Vartotojas pasirenka filtruoti pagal kompaniją.
2. Spaudžia „Filtruoti“.

Rezultatas: Rodomi paieškos rezultatai tik su pasirinkta kompanija.

Scenarijus:

1. Vartotojas pasirenka filtruoti pagal persėdimų skaičių.
2. Pasirenka filtruoti pagal kompaniją.
3. Spaudžia „Filtruoti“.

Rezultatas: Rodomi paieškos rezultatai tik su pasirinkta kompanija ir persėdimu skaičiumi.

Scenarijus:

1. Vartotojas pasirenka filtruoti pagal kriterijus, neatitinkančius nė vieno paieškos rezultato.
2. Spaudžia „Filtruoti“.

Rezultatas: Rodomas pranešimas, kad nėra paieškos rezultatų, atitinkančių pasirinktus kriterijus.

5.3. Rezultatų rūšiavimas

Reikalavimai testui: Atidarytas skrydžių paieškos rezultatų langas.

Scenarijus:

1. Vartotojas spaudžia mygtuką „Greičiausi“.

Rezultatas: Rodomas skrydžių paieškos rezultatų sarašas surušiuotas pagal kelionės trukmę nuo trumpiausios iki ilgiausios.

Scenarijus:

1. Vartotojas spaudžia mygtuką „Pigiausi“.

Rezultatas: Rodomas skrydžių paieškos rezultatų sarašas surušiuotas pagal kelionės kainą nuo mažiausios iki didžiausios.

Scenarijus:

1. Vartotojas spaudžia mygtuką „Optimaliausi“.

Rezultatas: Rodomas skrydžių paieškos rezultatų sarašas surušiuotas nuo optimaliausio iki mažiausiai optimalaus rezultato.

5.4. Skrydžių paieškos rezultato informacijos peržiūra

Reikalavimai testui: Atidarytas skrydžių paieškos rezultatų langas.

Scenarius:

1. Vartotojas spaudžia ant pasirinkto skrydžio.

Rezultatas: Atidaromas dialogas su išvykimo vieta ir laiku, atvykimo vieta ir laiku, skrydžio trukme, kaina, kompanija, persėdimu skaičiumi.

5.5. Nukreipimas į bilietų platintojo puslapį

Reikalavimai testui: Atidarytas skrydžio detalių dialogas.

Scenarius:

1. Vartotojas spaudžia mygtuką „Pirkti“.

Rezultatas: Vartotojas nukreipiama į bilieto pardavėjo puslapį.

5.6. Būsimų skrydžių peržiūra

Reikalavimai testui: Atidarytas skrydžių paieškos rezultatų langas, yra būsimų skrydžių.

Scenarius:

1. Vartotojas spaudžia mygtuką „Būsimi skrydžiai“.

Rezultatas: Atidaromas langas su būsimų skrydžių sąrašu.

Reikalavimai testui: Atidarytas skrydžių paieškos rezultatų langas, nėra būsimų skrydžių.

Scenarius:

1. Vartotojas spaudžia mygtuką „Būsimi skrydžiai“.

Rezultatas: Rodomas pranešimas, kad būsimų skrydžių nėra.

5.7. Būsimo skrydžio informacijos peržiūra

Reikalavimai testui: Atidarytas būsimų skrydžių langas.

Scenarijus:

1. Vartotojas spaudžia ant pasirinkto skrydžio.

Rezultatas: Atidaromas dialogas su išvykimo vieta ir laiku, atvykimo vieta ir laiku, skrydžio trukme.

Išvados ir rezultatai

Atlikus robastiškumo analizę, preliminarią projekto peržiūrą ir apibrežus techninę sistemos architektūrą (dalį ICONIX proceso žingsnių) pastebėjome, kad šioje darbo fazėje pirminiai sistemos reikalavimai atitinka sistemos numatomas užduotis, todėl programų sistemos reikalavimai nebuvo keisti.

Pakeitimų sąrašas

1. Skyrius „Struktūrinis dalykinės srities modelis“ perdarytas į „Statinė programų sistemos struktūra“,
2. Apibrėžta techninė sistemos analizė,
3. Užduočių aprašymuose pridėtos robustumo diagramos.

Šaltiniai

Analizuojant kuriamą sistemą, kuriant sistemos reikalavimus, apibrėžiant struktūrinį dalyki-nės sryties modelį bei aprašant sistemos atliekamas užduotis buvo remtasi šia literatūra:

1. Don Rosenberg ir Matt Stephens. *Use Case Driven Object Modeling with UML Theory and Practice*. Apress, 2007.