



VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS INSTITUTAS  
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Bakalauro baigiamasis darbas

**Organizuojamų renginių automatinio parinkimo sistema**

Atliko:

Greta Budrauskaitė

parašas

Vadovas:

dr. Joana Katina

Vilnius  
2020

# Turinys

<b>Santrauka</b>	<b>3</b>
<b>Summary</b>	<b>4</b>
<b>Iyadas</b>	<b>5</b>
<b>1. Sukurtų panašių sprendimų analizė</b>	<b>7</b>
1.1. Kasvyksta.lt . . . . .	7
1.2. Mzirafos.lt . . . . .	7
1.3. Meetup.com . . . . .	8
1.4. Vilnius-events.lt . . . . .	8
1.5. Savaitgalis.lt . . . . .	9
1.6. Apibendrinimas . . . . .	9
<b>2. Sistemos modelis</b>	<b>11</b>
2.1. Sistemos funkcionalumas . . . . .	11
2.2. Sistemos naudojimo atvejai . . . . .	12
2.3. Duomenų bazės modelis . . . . .	13
2.4. Sistemos navigacijos modelis . . . . .	14
2.5. Rekomendacijų modelis . . . . .	15
<b>3. Sistemos praktinė realizacija</b>	<b>17</b>
3.1. Naudojamos technologijos . . . . .	17
3.2. Sistemos struktūra . . . . .	17
3.3. Naudotojo istorijos įgyvendinimas . . . . .	18
3.3.1. Automatinis renginių parinkimo įgyvendinimas . . . . .	18
3.3.2. Atributai . . . . .	19
3.3.3. Įvertinimas . . . . .	20
3.4. Renginių atvaizdavimas žemėlapyje . . . . .	21
3.5. Informacija apie renginius . . . . .	21
3.6. Komentarai po renginio . . . . .	22
3.7. Sistemos saugumas . . . . .	22
3.7.1. Naudotojo informacijos šifravimas . . . . .	22
3.7.2. Puslapių ir funkcijų apsauga . . . . .	22
3.7.3. Formų validacijos . . . . .	23
3.8. Sistemos testavimas . . . . .	23
<b>Išvados ir rekomendacijos</b>	<b>24</b>
<b>Priedas Nr. 1</b>	<b>25</b>
<b>Priedas Nr. 1</b>	<b>26</b>

## **Santrauka**

Šiame darbe yra aprašoma automatinė renginių parinkimo sistema, jos išsamus analizės bei realizavimo procesas. Darbo tikslas – sukurti renginių sistemą, kurioje pagal naudotojo pasirinktus kriterijus bus parenkami renginiai. Šiam tikslui įgyvendinti bus panaudoti dirbtinio intelekto metodai, kurie padės atrinkti labiausiai atitinkantį renginį naudotojui. Tikslo įgyvendinimui yra atsižvelgiama į jau esamas panašias sistemas, iš kurių buvo atrinkti privalumai ir trūkumai. Pagal šiuos kriterijus buvo suformuluoti pagrindiniai uždaviniai, sukurtas ir aprašytas detalus sistemos modelis. Pasiektas darbo rezultatas yra sukurta žiniatinklio programų sistema, gebanti atrinkti renginius registruotam naudotojui, pagal jo pasirinktus kriterijus. Sistemoje įgyvendintas modelis užtikrina šias funkcijas: naudotojas gali keisti savo parinktus kriterijus, gali vertinti sistemoje registruotus renginius, gali pamatyti kur vyksta renginiai, elektroniniu paštu gauti pranešimus apie pasižymėtą renginį.

# **Summary**

## **Automation Selection System of Organized Events**

This paper describes the automatic event selection system, its detailed analysis and implementation process. The aim of the thesis is to create an event system that will select events according to user-selected criteria. Artificial intelligence techniques will be used to accomplish this, which will help to select the most appropriate event for the user. Existing similar systems, from which advantages and disadvantages have been selected, are taken into account for the purpose. Based on these criteria, the main tasks were formulated and a detailed system model was developed and described. The result of the work is a web application system capable of selecting events for a registered user according to the criteria of his choice. The model implemented in the system provides the following functions: the user can change the criteria of his choice, can evaluate the events registered in the system, can see where the events are taking place, receive notifications about the marked event by e-mail.

## Ivadas

Šiuolaikinėje rinkoje siekiama parduoti vis daugiau produkcijos ir tapti patrauklesniu klientui. Todėl įmonės stengiasi pritraukti klientus tiek naujovėmis, tiek rinkoje populiariais metodais. Šiuo metu rinkoje yra populiarios rekomendacijų sistemos, kurios leidžia sukaupti kiek įmanoma daugiau informacijos apie šių sistemų naudotojus ir jiems pateikti labiausiai jų norus atitinkantį produktą ar paslaugą. Turėdami tokios informacijos, sistemos gali teikti įvairius pasiūlymus, kurie padėtų atnešti daugiau naudos pačioms kompanijoms, kurios būtent ir kuria tam tikrus produktus ar paslaugas.

Šiuo metu yra daug skirtingų rekomendacinių sistemų, kurių kiekviena turi savitus algoritmus ir kriterijus, pagal kuriuos stengiamasi nuspėti naudotojų pomėgius ir poreikius. Šių algoritmų naudojimo tikslis pradžia nėra žinoma, tačiau didelės kompanijos kaip „Netflix“, „Youtube“, „Facebook“ labai plačiai naudoja rekomendacines sistemas, siekdamos pritraukti kuo daugiau naudotojų bei palengvinti naudojimąsi programomis ar sumažinti skiriamą laiką naujovių paieškai. Pavyzdžiui, „Youtube“ sistema rekomenduoja naudotojui vaizdo įrašus. Sistema saugo visą naudotojo istoriją, pavyzdžiui, naudotojui peržiūrėjus vaizdo įrašą, sistema pagal peržiūrėtus vaizdo įrašus siūlo tokio pačio ar panašaus turinio įrašus ir taip bando nuspėti naudotojo pomėgius. Sistemos naudotojui yra patogiau, nes jis gali greitai rasti mėgstamus žiūrėti vaizdo įrašus. Kita žinoma sistema „Spotify“ veikia priešingai nei „Youtube“ sistema. „Spotify“ sistema nekreipia dėmesio į naudotojo istoriją, tačiau ji koncentruojasi į patikusių įrašų turinį. Sistema bando nuspėti, ko naudotojas norėtų klausytis pagal tai, kokio turinio ar žanro klausosi dabar. „Spotify“ nesiūlo tų pačių jau praeityje klausytų muzikinių takelių. Šiuo atveju visas dėmesys skiriamas į tam tikrą logiką, kuri ir dedikuoja algoritmui kaip reikia filtruoti duomenis. Tarkime, kreipiamas dėmesys į muzikos stilių, tad naudotojui siūloma panašaus stiliaus muzika, kurios jis dar nėra klausęs. Panašiu principu kitos kompanijos daro prielaidas ir siūlo savo klientams tai, kas atrodo jiems labiausiai patiktą.

Dėl poreikio apdoroti didelį informacijos kiekį yra sukurti automatinio mokymosi algoritmai, kurie padeda pasiekti maksimalų rezultatą per ganėtinai trumpą laiką. Vienas iš jų yra bendrasis filtravimas (angl. Collaborative filtering), o kitas turiniu pagrįstas filtravimas (angl. Content-based filtering). Abiejų algoritmų tikslas yra patenkinti naudotoją, pasiūlant maksimaliai atitinkantį produktą. [2] Pirmasis metodas koncentruojasi į naudotojo istorinius duomenis. Sistema saugo visus naudotojo veiksmus, juos analizuoja ir pagal tai pateikia pasiūlymus tolimesniems naudotojo veiksmams. Dėl šios priežasties dažnai tokios sistemos prašo, kad jų naudotojas pateiktų įvertinimą siūlomoms paslaugoms ar prekėms. Šis būdas padeda sukaupti daug informacijos apie sistemos naudotoją ir atlikti įvairias analizes, kurių išvadose ir nuspėjami naudotojo poreikiai bei pomėgiai. Tokie rezultatai padeda verslui ir jo plėtrai. Kitas būdas – turiniu pagrįstas filtravimas (angl. Content-based filtering) skiriasi nuo pirmojo. Šis algoritmas ieško panašumų duomenyse. Pavyzdžiui, jei naudotojas mėgsta „Jazz“ stiliaus muziką, tai jam turėtų pateikti kitus pasiūlymus, kurie yra susiję su „Jazz“ ir galbūt kiti žmonės įvertins, kad tai vienas geriausių pasiūlymų. Tad šis metodas nekreipia dėmesio į istoriją, tačiau orientuotas į turinį. Išanalizavus sukauptą informaciją, naudotojui yra pateikiami labiausiai atitinkantys produktai ir paslaugos. Vadinasi, abu metodai kaupia kiek įmanoma daugiau informacijos apie naudotoją ir analizuojant turimus duomenų šaltinius – siūlo paslaugas ir produktus.

Šiuo darbu norima parodyti, kaip galima patobulinti sistemas rekomendacijų algoritmais. Šiame darbe aprašoma kuriama sistema - rekomenduojamų renginių svetainė, kurioje bus pritaikyti

aukščiau išvardinti metodai siekiant sudominti lankytoją, kad jis dažniau naudotųsi sukurta sistema. Atsižvelgiama į tai, kad rekomendacijos panaudojimas suteiktų komfortą naudotojui, nereiktų ilgiau gaišti laiko ieškant naujų renginių. Šio darbo pagrindinis tikslas – sukurti rekomendacijų sistemą, kurioje suteikiama naudotojui galimybė greitai ir tikslingai peržiūrėti norimus renginius, kurie atitiks pagal tam tikrus kriterijus ir vertinimus.

Abu paminėti algoritmai turi privalumų bei trūkumų. Tačiau šiuolaikinės sistemos naudoja hibridines rekomendacines sistemas, įdiegia abudu šiuos algoritmus, norint gauti maksimalų rezultatą ir jį pateikti naudotojui. Dėl didelio susidomėjimo ir tokių sistemų paklausos – šis baigiamasis bakalauro darbas yra apie tokio tipo algoritmų integraciją į kuriamą sistemą. Siekiama išbandyti ir pritaikyti abu algoritmus kuriamai sistemai, sekti tam tikrus duomenis ir jais naudojantis pasiūlyti naudotojui aktualiausių renginių sąrašą.

Darbai įgyvendinti pasitelkiamos įvairios technologijos: Java programavimo kalba, „Spring“ karkasas, „Angular“ platforma bei MySQL duomenų bazė. Prieš kuriant sistemą buvo atlikta panašių sistemų analizė, kuri pateikiama šiame darbe. Pasiektas baigiamojo bakalauro darbo rezultatas – sukurta veikianti sistema, kurioje naudotojas gali užsiregistruoti, koreguoti savo profilį bei savo atributus, pagal kuriuos sistema ieško atitikmenų duomenyse, naršyti tarp renginių, juos įvertinti, atsisiųsti į savo elektroninį paštą. Taip pat naudotojas turi galimybę pamatyti, kokie sistemos atrinkti renginiai jam yra siūlomi pagal išsaugotus atributus ir renginių vertinimus.

Šio aiškinamojo rašto pirmajame skyriuje pateikiama panašių sistemų analizė, kurioje aprašyti palyginimai, kokios sistemos yra sukurtos renginių rekomendavimo tema. Atlikta analizė koncentruota į tai, ar sistemos turi rekomendacinio pobūdžio integruotų algoritmų bei aptariamoms pagrindinėms analizuojamų sistemų funkcijoms. Antrame skyriuje aprašomas teorinis sistemos modelis, kuriame pateikiamas kuriamos sistemos funkcionalumas, panaudojamų funkcijų diagramos, duomenų bazės struktūra, puslapių navigacijos struktūra bei rekomenduojamų modelių veikimai. Trečiame skyriuje pateikiamas visas svetainės įgyvendinimas. Paminimos pagrindinės technologijos, kuriomis remiantis sukurta sistema, kaip integruoti rekomendacinio pobūdžio algoritmai, koks svetainės saugumas, validacija bei aptiriamas testavimas. Galiausiai pateikiami ateities planai, kaip galima tobulinti svetainę bei padarytos išvados.

# 1. Sukurtų panašių sprendimų analizė

Atlikus panašių sistemų sprendimų paiešką internete rastos kelios sistemos, kurios naudojamos tiek Lietuvoje, tiek ir kitose šalyse. Jų pagrindinis tikslas leisti naudotojui matyti vykstančius renginius, kuriuose jis norėtų apsilankyti. Analizuojamos sistemos buvo pasirinktos neatsitiktinai, nes atkreiptas dėmesys į jų funkcionalumą bei pagrindinį sistemų produktą - organizuojamus renginius. Išskirti ir aprašyti sistemų privalumai bei trūkumai, esamos bei trūkstamos funkcijos. Baigiamajame darbe analizuojamos sistemos: kasvyksta.lt, mzirafos.lt, meetup.com, vilnius-events.lt, savaitgalis.lt sistemos. Žemiau pateikta analizė apie šias sistemas.

## 1.1. Kasvyksta.lt

Pirmoji analizuojama žiniatinklio sistema - Kasvyksta.lt. Šioje sistemoje galima rasti įvairiausių renginių, kurie gali būti mokami ir nemokami. Sistemoje naudotojas gali rasti papildomos informacijos, kuri nėra susijusi su renginiais: horoskopai, orų prognozė ir t. t. Naudotojas naršydamas renginių sistemoje gali pamatyti rekomenduojamų renginių sąrašą, tačiau tai nėra atrinkta individualiai naudotojui. Rekomenduojamas renginių sąrašas gali būti sudarytas tiek atsitiktine tvarka, tiek reklamos tikslais. Svetainėje prisijungęs naujas naudotojas gali ieškoti renginių pasitelkęs paieškos filtrą (pasirinkęs datą, vietą, kainą, renginio tipą ir kt.) bei pridėti savo organizuojamą renginį. Jei renginys yra mokamas ir naudotojas nori nusipirkti bilietą į jį dominantį organizuojamą renginį, tokiu atveju sistema nukreipia naudotoją į bilietų platinimo sistemą, kad naudotojas galėtų įsigyti bilietą. Šioje sistemoje naudotojas gali susikurti paskyrą, ją koreguoti, filtruoti renginius, skaityti detalią informaciją apie renginius bei matyti siūlomus renginius atsitiktine tvarka.



1 pav. Kasvyksta.lt

## 1.2. Mzirafos.lt

Mzirafos.lt – antroji analizuojama sistema, kurioje pateikiami tik organizuojamų renginių skelbimai, kurie vyksta trijuose Lietuvos miestuose: Vilniuje, Kaune ir Klaipėdoje. Naudotojas turi galimybę atsisiųsti programėlę į išmanųjį įrenginį arba naršyti internetinėje svetainėje. Šioje sistemoje yra galimybė filtruoti renginius pagal vietovę, pasirinkus vieną iš trijų variantų. Sistemoje naudotojas gali užregistruoti savo renginį, tačiau tokia sistemoje teikiama paslauga yra mokama. Programėlėje šalia kiekvieno renginio yra rodoma informacija apie susijusius renginius, kurie potencialiai aktualūs naudotojui ir yra sukurti pagal naudotojo turinio filtravimą. Taip pat, skaitant daugiau informacijos apie patį renginį, žemėlapyje galima pamatyti organizuojamo renginio vietą. Vienintelis trūkumas, jog naudotojas negali susikurti paskyros, todėl ši sistema tik reklamuoja renginius, pateikia nuorodas kur ieškoti detalesnės informacijos arba nukreipia į kitas svetaines, kurios yra oficialios renginių svetainės arba į bilietų platinimo sistemas kaip „bilietai.lt“ ar „tiketa.lt“.

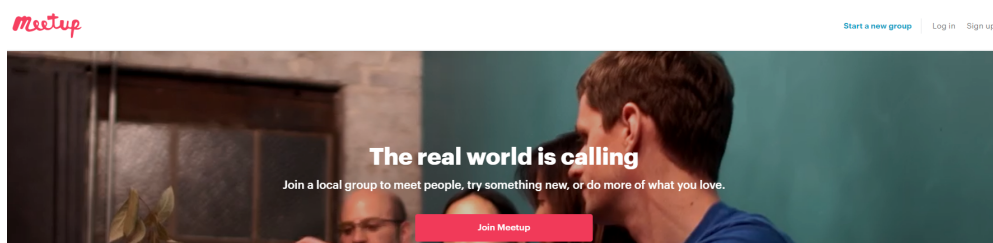
Mzirafos.lt sistema nesaugo daug informacijos apie naudotojus, todėl ji negali personalizuoti duomenų srauto ir siūlyti renginius atsižvelgiant į tai, kokiais renginiais naudotojas domėjosi praeityje. Tokiu principu veikianti sistema pasiekia gerą greitaveiką, tačiau neturi platesnio funkcionalumo.



2 pav. Mzirafos.lt

### 1.3. Meetup.com

Sekanti baigiamajame darbe analizuojama sistema yra Meetup.com. Ši sistema skatina pačius žmonės lokaliai organizuoti įvairius renginius ir burtis į socialines grupes pagal pomėgius bei tikslus. Norint naudotis šia sistema reikia užsiregistruoti, kitu atveju jokių veiksmų negalima atlikti. Registruotas naudotojas gali ieškoti renginių, kuriuose norėtų dalyvauti. Šioje sistemoje yra sukurta papildoma funkcija, kurią pasirinkęs naudotojas gali priklausyti kokiam nors grupei, pavyzdžiui kuri mėgsta „Jazz“ muziką. Tokiu atveju visiems grupės nariams siūlomi renginiai, kurie yra susiję su šiuo muzikos žanru. Taip pat galima pasirinkti bet kokią kitą kategoriją, pagal kurią sistema atrinks ir siūlys naudotojui dalyvauti pasirinktos kategorijos renginiuose. Analizuojamoje sistemoje yra susirašinėjimo su kitais naudotojais funkcija bei užklausų siuntimas į pasirinktą renginį. Šioje sistemoje taikomas turinio filtravimas, kuris leidžia naudotojui tikslingai ir greitai rasti, kokiam renginyje sudalyvauti.



3 pav. Meetup.com

### 1.4. Vilnius-events.lt

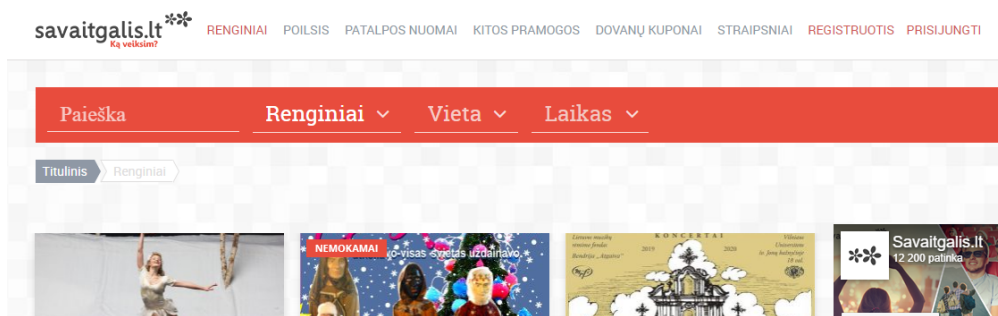
Ketvirtoji analizuojama žiniatinklio sistema - Vilnius-events.lt, kuri orientuota tik į vykstančius renginius Vilniuje. Šioje sistemoje nėra galimybės užregistruoti naudotojų, tačiau turi paiešką tarp renginių ir galimybę pranešti svetainės administracijai apie vykstantį renginį. Tokio atveju labiau orientuojamasi į tarpininkavimą tarp renginių organizatorių ir kliento. Naudotojas norėdamas perskaityti detalesnę informaciją apie renginį yra iš karto nukreipiamas į pirminę pasirinkto renginio sistemą bei įsigyti bilietus į organizuojamą renginį. Vilnius-events.lt sistemoje nėra rekomenduojamų renginių pateikimo sistemos naudotojui, kadangi sistemoje nėra galimybės užsiregistruoti naudotojui ir susinkti informacijos reikalingos rekomenduojamų renginių sąrašui pateikti.



4 pav. Vilnius-events.lt

## 1.5. Savaitgalis.lt

Paskutinė baigiamajame darbe analizuojama sistema - Savaitgalis.lt, kurioje galima peržiūrėti renginius vykstančius visoje Lietuvoje. Sistemoje renginiai išskirstomi pagal jų veiklos kategoriją. Taip pat naudotojas užregistruoti sistemoje bei naudotis paieškos filtru, taip surasdamas sau tinkamiausią organizuojamą renginį. Analizuojama sistema atlieka tarpininkavimo funkciją su oficialiomis renginių sistemomis. Tokiu atveju, norint sužinoti daugiau informacijos apie vykstantį renginį ar įsigyti bilietus, sistemos naudotoją nukreipia į kitas sistemas, kuriose naudotojas gali sužinoti reikiamą informaciją apie renginį. Šioje sistemoje naudotojas gali atlikti papildomas funkcijas: pirkti dovanų kuponus ar skaityti įvairius straipsnius. Analizuojamoje sistemoje Savaitgalis.lt taip pat nerasta integruotų jokių papildomų algoritmų, kurie galėtų naudotojui pateikti siūlomų renginių sąrašą pagal registruoto naudotojo pateiktus duomenis.



5 pav. Savaitgalis.lt

## 1.6. Apibendrinimas

Žemiau pateikta lentelė su pagrindinėmis funkcijomis, kurios bus naudojamos kuriamoje sistemoje. Lentelėje galima matyti kokios funkcijos yra naudojamos ir nenaudojamos aukščiau analizuotose sistemose. Atlikus pasirinktų sistemų analizę, galima teigti, kad automatinė rekomendacijų algoritmai nėra integruoti veikiančiose sistemose. Apibendrinimui žemiau pateikta lentelė vaizduoja, kokias pagrindines funkcijas turės kuriama sistema bei funkcijų palyginimas su analizuotomis sistemomis.

1 lentelė. Panašių sistemų analizė

Funkcija   Sistema	kasvyksta.lt	mzirafos.lt	meetup.com	vilnius-events.lt	Savaitgalis.lt
Automatinė renginių rekomendacija pagal įvertinimą.	-	-	-	-	-
Automatinė renginių rekomendacija pagal renginio atributą (angl. tag).	-	-	+	-	-
Galimybė žemėlapyje pamatyti vykstančius renginius.	+	+	+	-	-
Galimybė užsiregistruoti ir koreguoti savo profilį.	+	-	+	-	-
Gauti išsamią informaciją apie renginius į elektroninį paštą.	-	-	-	-	-
Renginių paieška.	+	+	+	+	+
Komentuoti renginį.	+	-	+	-	-

## 2. Sistemos modelis

Organizuojamų renginių sistemų analizės metu buvo išskirti pagrindiniai funkciniai reikalavimai naujai kuriamai organizuojamų renginių sistemai. Prieš kuriant rekomenduojamų renginių sistemą buvo sukurtas aiškus sistemos modelis, kurį apima sistemos naudojimo atvejų, navigacijos ir sekų diagramos bei duomenų bazės modelis.

### 2.1. Sistemos funkcionalumas

- Naudotojo registracija ir prisijungimas. Registracijos formą sudarys naudotojo asmeninių duomenų užpildymas: prisijungimo vardas, pilnas vardas, slaptažodis ir elektroninis paštas. Bus pridėtos papildomos formų validacijos, kad užtikrintų duomenų tikslumą. Užsiregistravus, naudotojas galės prisijungti prie sistemos pateikdamas registracijos metu įvestus duomenis – savo prisijungimo vardą ir slaptažodį. Suvedus teisingai duomenis, naudotojas bus nukreiptas į jo profilio puslapį, iš kurio galės nueiti į kitus sistemos puslapius.
- Profilio redagavimas. Kiekvienas prisijungęs naudotojas galės redaguoti savo profilio duomenis ir pakeisti registracijos formoje užpildytą informaciją: prisijungimo vardą, pilną vardą, elektroninį pašta ir t. t. Jei redagavimo metu bus klaidingai įvesti duomenis - iššoks pranešimas, kad kažkur įvyko klaida ir duomenų neišsaugos.
- Renginių peržiūrėjimas, ieškojimas. Kiekvienas naudotojas, nepriklausomai nuo to ar yra prisijungęs, galės atlikti renginių paiešką pasitelkęs datų filtravimą.
- Daugiau informacijos apie renginius. Kiekvienas prisijungęs naudotojas, galės spausti mygtuką esantį šalia renginio, kad sistema nukreiptų į kitą puslapį sužinoti daugiau informacijos. Tame puslapyje bus pateikiama detalesnė informacija ir, jei naudotojas nori įsigyti bilietus, nukreipiama į kitus šaltinius, kuriuose galima tai atlikti.
- Renginių atvaizdavimas žemėlapyje. Prisijungęs naudotojas turės galimybę matyti renginio lokaciją, kuri bus atvaizduojama detalios informacijos apie renginį puslapyje. Po renginio aprašymo bus matomas žemėlapis.
- Atributai. Prisijungęs naudotojas savo profilyje turės funkciją: pridėti atributą, redaguoti, naikinti. Šie atributai bus identifikuoti kiekvieno naudotojo ir padės sistemos algoritmams filtruoti renginius pagal pateiktus atributus. Administratorius turės papildomą funkciją, t. y. pridėti renginiui skirtus atributus, juos redaguoti, naikinti.
- Įvertinti renginį. Prisijungęs sistemos naudotojas galės įvertinti renginį skalėje nuo 1 iki 10 balų. Įvertinimo skalė bus pateikiama detalios informacijos apie renginį puslapyje po renginio pavadinimo.
- Rekomenduojami renginiai. Prisijungęs naudotojas turės dar vieną atskirą skiltį, kurioje dvejais metodais bus atvaizduojami renginiai. Matydamas renginius galės atlikti visus identiškus veiksmus kaip ir pradiniame renginių puslapyje: sužinoti daugiau informacijos arba gauti informaciją elektroniniu paštu.
- Informacija elektroniniu paštu. Tik prisijungę naudotojai turės galimybę atsisiųsti visą informaciją į savo el. pašta iš sistemos.
- Komentuoti renginį. Prisijungęs naudotojas galės komentuoti pasirinktą renginį.



6 pav. Sistemos naudojimo atvejai

## 2.2. Sistemos naudojimo atvejai

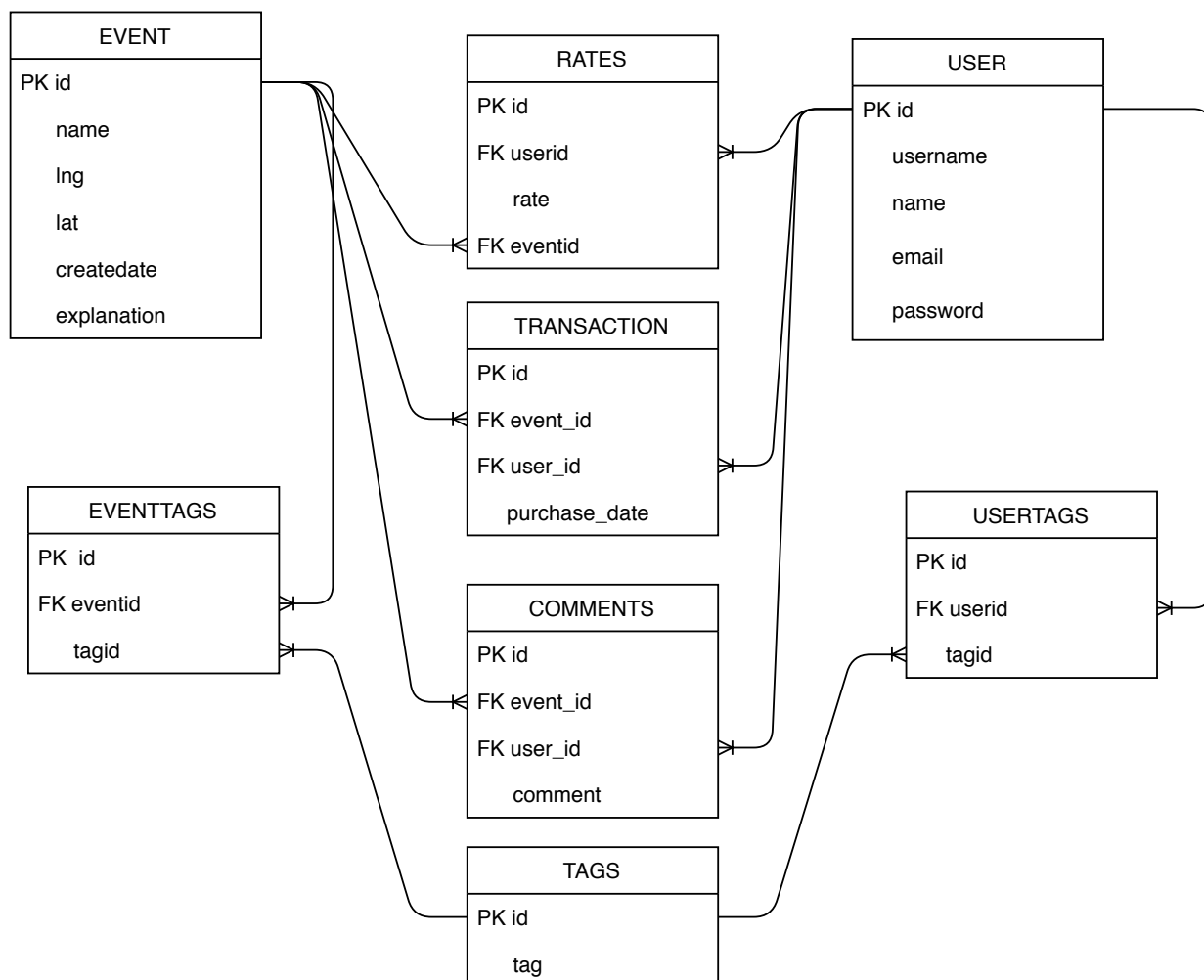
6 pav. galima matyti sistemoje esančius aktorius – neprisijungęs naudotojas, prisijungęs naudotojas, administratorius. Neprisijungęs naudotojas – galės tik ieškoti renginių ir užsiregistruoti. Prisijungęs naudotojas – galės matyti ir naršyti apie renginius, peržvelgti jų išsamią informaciją, papildomai galės matyti žemėlapyje atvaizduojamus renginius, komentuoti ar įvertinti renginius. Papildomas funkcionalumas – turės papildomą puslapį, kuriame bus jam siūlomi/rekomenduojami renginiai pagal jo įvestus atributus ar pagal jo paliktą įvertinimą. Taip pat galės valdyti savo profilį, atributus. Administratorius – turės teisę daryti viską, ką ir paprastas prisijungęs naudotojas, tačiau galės dar pridėti, redaguoti ar trinti renginius, jiems priskirti atributus, kuriuos taip galės redaguoti bei trinti. Taip pat galės sekti naudotojų susidomėjimą renginiais, kokie renginiai labiausiai domina lankytoją. Taip pat galės tvarkyti ir pačius naudotojus ir jų duomenis.

Kiekvienam naujam prisijungusiam naudotojui bus priskirta paprastojo naudotojo rolė. Ją keisti galės tik administratoriaus teises turintis naudotojas, kuris turės galimybę valdyti visus naudotojų duomenis. Sistemos naudotojų veiksmų apribojimas rolėmis padeda išvengti duomenų nutekėjimo,

t. y. didina sistemos saugumą.

## 2.3. Duomenų bazės modelis

Vios sistemos duomenų modelis pateikiamas 7 pav., kuris pateiktas žemiau. Sistemos duomenų bazės tikslas – saugoti naudotojų, renginių ir naudotojo atliktų veiksmų duomenis.



7 pav. Duomenų bazės modelis

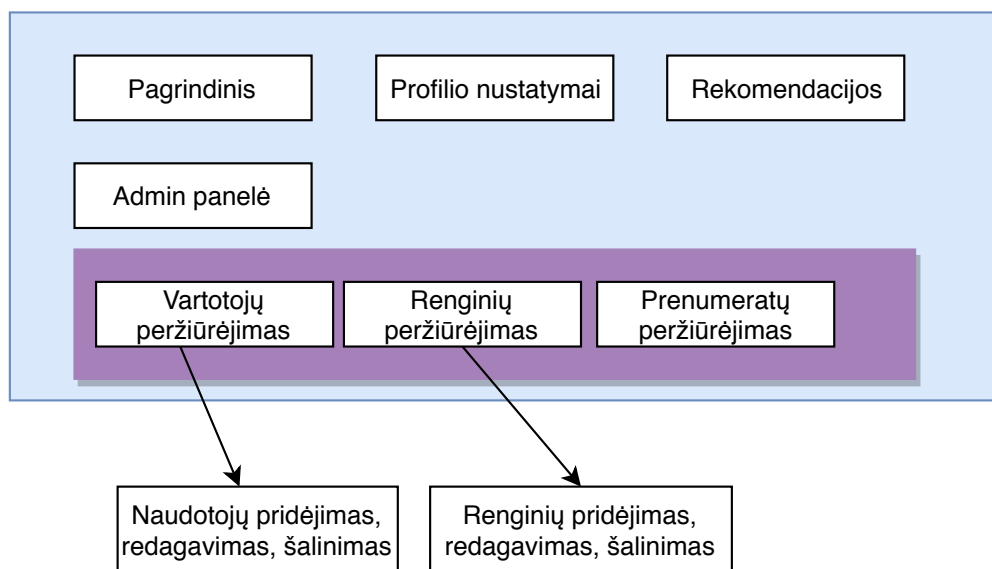
- naudotojas – lentelė, kurioje saugomi visi užsiregistravę naudotojai. Šioje lentelėje saugomas naudotojo pirminis raktas – identifikatorius, vardas, elektroninio pašto adresas, prisijungimo vardas, slaptažodis bei priskirta rolė sistemoje.;
- renginiai – lentelė, skirta renginių informacijos saugojimui. Lentelės įrašė yra pirminis raktas – identifikatorius, renginio pavadinimas, renginio vietovės nuorodos, papildoma informacija apie renginį;
- rates – lentelė, skirta naudotojų įvertinimui saugoti. Šioje lentelėje saugojami duomenys apie įvertinimus: identifikatorius, naudotojo identifikatorius, renginio identifikatorius bei įvertinimas;
- eventtags – lentelė, kurioje saugojama visa informacija apie renginio atributus. Joje saugojama identifikatorius, renginio identifikatorius ir atributo pavadinimas. Sistemoje gali būti

keletas atributų vienam renginiui;

- usertags – lentelė skirta saugoti naudotojo atributus. Joje yra identifikatorius, naudotojo identifikatorius bei atributo pavadinimas;
- transaction – lentelė skirta saugoti duomenis apie tai, kokius renginius naudotojai mėgsta ir apie kokius renginius jie nori daugiau sužinoti. Joje saugojamas identifikatorius, renginio identifikatorius, naudotojo identifikatorius bei data;
- tags – lentelė skirta saugoti atributus;
- comment – lentelė skirta saugoti prisijungusio naudotojo parašytus komentarus apie renginius. Kiekvienas renginys bus klasifikuojas pagal renginį.

## 2.4. Sistemos navigacijos modelis

Prieš sukuriant sistemą buvo suprojektuotas jos navigacijos modelis (8 pav.). Sistemos navigacijos modelis priklauso nuo akto, kuris naudojami sistema. Jeigu naudotojas nėra prisijungęs, jis gali matyti visų renginių sąrašą, prisijungimo ir registracijos galimybes. Prisijungęs naudotojas gali matyti visų renginių sąrašą, savo profilį, asmeniškai rekomenduojamų renginių sąrašą, galimybę atsijungti nuo sistemos. Daugiausiai teisių turintis naudotojas – administratorius. Jis mato viską, ką gali matyti prisijungęs naudotojas bei papildomai pasiekia sistemos administravimo aplinką, kurioje gali redaguoti naudotojus, juos šalinti, administruoti renginius ( pridėti naujus renginius, koreguoti jau egzistuojančius renginius), matyti koks yra susidomėjimas renginių bei koreguoti jiems priskirtus atributus.



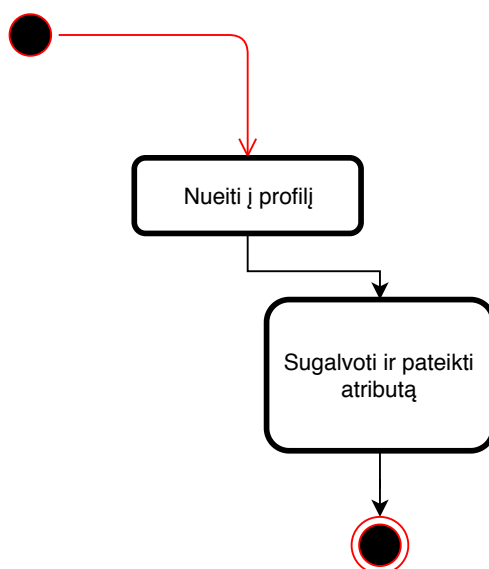
8 pav. Sistemos navigacijos modelis

8 pav. esantys laukai mėlyname fone yra naršymo juostoje. Kiti laukai pateikiami tik naudotojui atlikus tam tikrą veiksmą: redagavimą, trynimą, naujo objekto pridėjimą. Prisijungęs administratorius mato papildoma sekciją „Admin panelė“, kurioje pateikiamos papildomos funkcijos (funkcijos yra pavaizduotos kitos spalvos fone). Administratorius norėdamas grįžti į pradinį vaizdą, kokį mato prisijungęs naudotojas, turi spausti savo vardą – taip sugrįžta į pradinį vaizdą, kurį mato prisijungęs naudotojas.

## 2.5. Rekomendacijų modelis

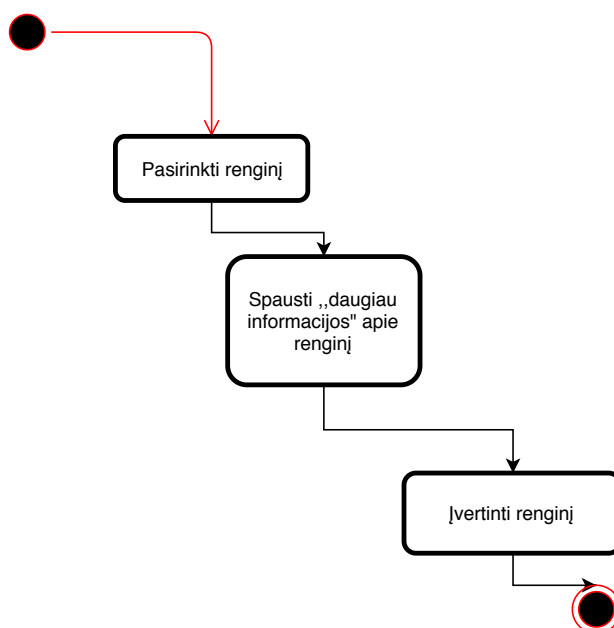
Rekomenduojamų svetainių modeliai atlieka vieną iš pagrindinių funkcijų visoje sistemoje – ne išimtis ir šioje kuriamoje svetainėje. Tokie modeliai grįsti įvairiais filtravimo algoritmais, kurie gali būti kombinuojami ar realizuoti po vieną. Plius tokia algoritmų kombinacija yra naudojama įvairiuose dirbtinio intelekto grįstose sistemose. Algoritmai padeda sukaupti daug informacijos ir naudojant tai galima apmokyti mašinas, kurios ateityje galėtų automatiškai generuoti kitus duomenis. Taip pat didelę įtaką daro filtravimo metodų įgyvendinimas sistemoje – yra galimybė įrašyti kiekvieno naudotojo veiksmus į duomenų bazę, pagal kuriuos vėliau galima nuspėti kitus galbūt netolimoje ateityje atliktinus naudotojo veiksmus. Dėl šios priežasties vienas pagrindinių ir svarbiausių dalykų kuriamoje sistemoje yra rekomendacijų pateikimo modelis. Jame kiekvienam naudotojui yra atskirai pateikiamas renginių sąrašas, kuris yra parenkamas pagal naudotojui aktualius ir geras rekomendacijas turinčius renginius. Rekomendacijų puslapis sudarytas iš dviejų dalių, kurių kiekvienoje yra po vieną renginių sąrašą. Vienas sąrašas susidaromas naudojant turinio pagrindo filtravimą, o kitas pagal naudotojų įvertinimus. Taip pat kaip ir pagrindiniame puslapyje, šį renginių sąrašą galima filtruoti pagal datas ar jų intervalus – tai dar labiau pagreitina renginio paiešką.

Pirmas filtras yra pagrįstas naudotojo įvestais atributais (angl. tags). Pirmiausiai naudotojas turi savo profilio puslapyje sugalvoti ir sukurti savo norimus atributus, pagal kuriuos sistemos algoritmai ieško atitikmenų ir pateikia juos naudotojui. Atributas yra universalus, t. y. atributas neprivalo būti tik renginio kategorija, žanru, stiliumi ar vieta. Pats atributas yra tekstinė reikšmė, todėl jis dar labiau palengvina renginių paiešką naudotojui, siekiant gauti tikslų būtent naudotojui aktualių renginių sąrašą. Kai naudotojas siveda savo atributus – sistema ieško renginių, kurie turėtų tokius pačius atributus. Jei atributai sutampa su renginiuose naudojamais atributais – šiuos renginius siūlo naudotojui. Žemiau pateikiama atributo sukūrimo veiklos diagrama (9 pav.). Toks paieškos būdas laikomas turiniu pagrįstos paieškos metodu. [1] Jei naudotojas nėra įvedęs jokių atributų, sistema neieško ir nepateikia naudotojui suasmenintų renginių sąrašo. Ieškant pagal atributus reikia papildomai atkreipti dėmesį į tai, kad naudotojui norima siūlyti renginius ne iš visų, o tik iš ateityje vykstančių ar šiuo metu vykstančių renginių.



9 pav. Atributo sukūrimo veiklos diagrama

Antras filtras yra pagrįstas reitingavimo principu iš istorinių duomenų. Tokiu atveju sistemos naudotojas anksčiau įvertinęs renginį gali tikėtis, kad panašaus turinio renginys bus automatiškai jam rekomenduojamas ateityje. Tik prisijungęs naudotojas gali pateikti renginio vertinimą. Taigi rekomendacijų puslapio antrąją dalį užima renginių sąrašas, atrinktas pagal naudotojo įvertinimus. Sistemoje yra nustatytas minimalus įvertinimo balas, pagal kurį traktuojama, kad renginys patiko naudotojui. Šis įvertis yra 6 balai. Jei įvertinimas yra lygus ar aukštesnis už 6 balus, galima teigti, kad sistemos naudotojui patinka tokio tipo renginys. Žemiau pateikiama renginio įvertinimo veiklos diagrama (pav. 10). [5] Galima teigti, kad šis siūlomas modelis yra kombinuotas iš dviejų metodų, nes kuriamoje sistemoje atsižvelgiama į naudotojo įvertinimą bei ieškoma panašių atributų turinčių renginių. Tai reiškia, kad šiam filtravimui yra svarbus ne tik renginio įvertinimas, bet ir jo turinio panašumas su kitais renginiais.



10 pav. Renginio įvertinimo veiklos diagrama

Taigi, abu algoritmai veikia vienas nuo kito nepriklausomai ir pateikia skirtingus renginių sąrašus. Tačiau algoritmai veikia tik tuo atveju jei naudotojas įveda jam aktualius atributus bei vertina renginius. Tokie algoritmai yra plačiai naudojami ir kitose sistemose. Jie teikia itin didelę naudą, jeigu norima suprasti sistemos naudotojų poreikius.

## 3. Sistemos praktinė realizacija

### 3.1. Naudojamos technologijos

Prieš sukuriant rekomenduojamų renginių sistemą, buvo pasirinktos technologijos, padėsiančios sukurti internetinę sistemą. Renkantis technologijas buvo atsižvelgta į tai, ar jos turi aiškia ir išsamia dokumentacija, kuri padeda susipažinti su technologijų funkcionalumu ir panaudojimo galimybėmis. Pasirinktos technologijos:

- Spring karkasas;
- Java programavimo kalba;
- MySQL duomenų bazių valdymo sistema;
- Angular platforma.

Visų pirma, sistemos sukūrimui pasirinkta Java programavimo kalba. Išnagrinėjus karkasus, kurie yra paremti šia programavimo kalba ir naudoja MVC (model-view-controller) modelį, buvo pasirinktas „Spring“ karkasas. MVC - architektūrinis šablonas, kuriamą sistemą išskirstantis į tris loginius sluoksnius: modelius, valdiklius ir vaizdus. Karkasas išsiskyrė savo išsamia ir lengvai suprantama dokumentacija ir siūlomomis funkcijomis, su kuriomis internetinių sistemų kūrimas yra daug patogesnis. Karkasas leidžia pasiekti duomenų bazėje esančius elementus naudojant objekto vaidmenų modeliavimą - JPA. Antra, veiksmams su duomenų baze naudojama MySQL duomenų bazių valdymo sistema, dirbanti SQL kalbos pagrindu. Iš duomenų bazės įrašai gaunami naudojant anksčiau minėto JPA užklausas. Sistemoje sukurta viena duomenų bazė, kuri aprašyta teoriniame modelyje. Trečia, pasirinkta naudoti platforma „Angular“ – HTML, CSS, ir JavaScript karkasas, kurio dizainas ir funkcionalumas palaikomas visose populiariausiose naršyklėse ir yra prisitaikantis prie skirtingų įrenginių ekranų dydžių. Naudojant šį karkasą, palengvinamas puslapio dizaino kūrimas, nes galima naudoti jau sukurtas CSS stilių klases ir JavaScript funkcijas, pateikiamas dokumentacijoje, o ne įgyvendinti jas pačiam. Kuriant sistemą panaudota ir papildoma Bootstrap biblioteka. Bootstrap biblioteka naudojama datos parinkimo kalendoriui, suteikiančiam platesnį funkcionalumą nei įprasta HTML datos įvestis. Naudojant šį kalendorių, galima pasirinkti kalendoriaus kalbą, vaizduojamą datos formatą, išjungti arba paryškinti norimas datas, pasirinkti, kokiame intervale aktyvuoti datos rinkimąsi. Kalendorius palaiko ir skirtingus atvaizdavimo tipus: gali būti įvesties laukas, kur naudotojas įrašo datą arba iššokantis langas su vaizduojamomis datomis. Taip pat vaizduojant renginius žemėlapyje remiamasi „Google MAPS“ teikiama atvaizdavimo galimybe.

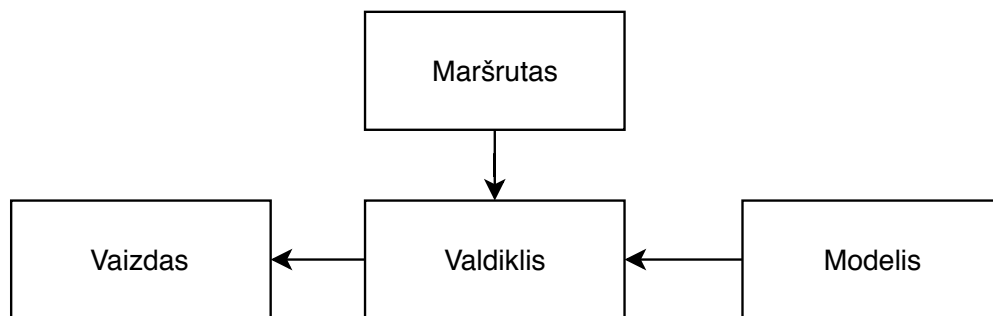
### 3.2. Sistemos struktūra

Automatinė rekomenduojamų renginių sistema kuriama Java „Spring“ karkasu, kuris sudarytas iš įvairių priklausomybių, tai yra įvairių bibliotekų, kurios yra atsakingos už tam tikrą funkcionalumą. Kaip ir minėta anksčiau, karkasas naudoja MVC (model-controller-view) modelį, jis yra išskaidomas į 3 pagrindines dalis:

- Modelius (angl. models) – naudojamus sąveikai su duomenų baze. Kiekvienas modelis atitinka skirtingą duomenų objektą. Pavyzdžiui, sistemoje yra lentelė „events“, kurioje saugojami visi reklamuojami renginiai ir detali informacija apie juos. Tad šiuo atveju modelyje taip pat bus tokie patys atitinkantys duomenys kaip duomenų bazėje esantys duomenų atributai: id, name, explantion, createdate, lng, lan. Dar modeliuose galima aprašyti ir ryšius,

kurie egzistuoja tarp tam tikrų lentelių. Šiuo atveju yra atskira lentelė „eventtags“, kurioje galima sukurti ryšį su „events“ lentele, taip palengvinant tam tikrų duomenų gavimą.

- Vaizdas – sistemoje vaizdai kuriami pasitelkiant „Angular“ technologiją. Ši technologija lengvai leidžia sukurti puslapius.
- Valdiklius – atsakingus už sistemos logikos valdymą. Valdiklis veikia kaip tarpininkas tarp vaizdų ir modelių. Jis reaguoja į naršyklės užklausas ir perduoda reikalingą informaciją iš modelio vaizdo komponentui. Valdikliuose aprašyti visi metodai, kurie yra susiję su sistemos duomenų bazėje esančių įrašų atvaizdavimu, pridėjimu, keitimu ar ištrynimu.



11 pav. Sistemos struktūra

Pagrindinių sistemos dalių sklandžiam veikimui reikalingi ir maršrutai (11 pav.). Naršyklei gavus užklausą vyksta tolimesnė eiga - nukreipiama į maršrutus, kuriuose nurodoma, koks valdiklis yra atsakingas už puslapį, į kurį kreipiamasi. Taigi, maršrutas nukreipia užklausą į tinkamą valdiklį, o pagal užklausą valdiklis gauna iš modelio reikalingą informaciją ir gražina ją į vaizdus. Taip susikuria atvaizdavimo ciklas, kurio pagalba naudotojas mato tik galutinį vaizdą - duomenų atvaizdavimą iš maršrutų.

### 3.3. Naudotojo istorijos įgyvendinimas

Kuriamoje sistemoje pagrindinis dėmesys skirtas naudotojui. Sistema suteikia galimybę naudotojui pateikti daugiau informacijos apie jo mėgstamus renginius, o sistema pasinaudojus papildoma informacija gali pateikti naudotojui aktualius pasiūlymus. Tad kuriamoje sistemoje būtent tai ir norima pateikti – kaupti kuo įmanoma daugiau duomenų, kad palengvinti naudotojo apsisprendimą dėl renginio pasirinkimo. Tokiu būdu ne tik užtikrinamas sistemos lankomumas, bet ir surenkama informacija apie tai, kuo naudotojai domisi ir kaip juos sudominti.

#### 3.3.1. Automatinis renginių parinkimo įgyvendinimas

Visas automatinis renginių parinkimas vyksta remiantis dirbtinio intelekto panaudojimo algoritmo galimybės kitose rekomenduojamų sistemose. Sukurtoje sistemoje galimi du būdai. Vienas būdas – kai naudotojas nurodo jam aktualius atributus. Šioje sistemoje užpildomas raktinių žodžių sąrašas, kuris gali atskleisti informaciją apie naudotojo pomėgius ir įpročius. Naudotojas, nurodydamas atributus, koncentruojasi į savo pomėgius. Kitu atveju, renginiai taip pat parenkami pagal pateiktus naudotojo vertinimus apie renginius. Žemiau pateikta detalesnė informacija apie sistemos įgyvendinimą.

### 3.3.2. Atributai

Vienas iš būdų – atributai, kuriuos naudotojas gali laisvai pasirinkti iš siūlomo raktinių žodžių sąrašo, pagal kuriuos galima filtruoti duomenis ir taip surasti geriausiai atitinkantį renginį. Renginiai taip pat turi priskirtus atributus. Šioje kuriamoje sistemoje naudotojas, nuėjęs į savo profilį, gali pastebėti, jog yra atskira lentelė, kurioje pateikiami naudotojo pridėti atributai ir yra galimybė ištrinti ar pridėti naują. Jeigu naudotojas nėra įvedęs nei vieno atributo, sistema tiesiog nepateikia jokių rekomenduojamų renginių. Tokiu atveju naudotojas turėtų nueiti į pagrindinį puslapį ir ieškoti renginių pačiam arba užpildyti atributus savo profilio puslapyje. Atributai, kuriuos užpildo naudotojas, yra lyginami su renginių atributais, kuriuos nurodo administratorius įkeldamas renginį į sistemą. Jei randamas sutapimas, galima daryti prielaidą, kad atitikmuo rastas – taip sukuriamas naujas sąrašas su naujai atrinktais renginiais. Toks sąrašas pateikiamas naudotojui ir atvaizduojamas įprastai kaip atvaizduojami pagrindiniame puslapyje.[4]

1 išeities kodas. Filtravimo pagal atributus kodas

```
1 public List<EventTag> findAllEventTags(Long userID) {
2     Date date1 = new Date();
3     List<UserTag> userTags = userTagRepository.findAllByUserId(
4         userID);
5     List<EventTag> eventTags = new ArrayList<>();
6     List<EventTag> eventsCurrent = new ArrayList<>();
7     for (UserTag userTag : userTags){
8         eventTags.addAll(eventTagRepository.findAllByTagId(
9             userTag.getTag().getId()));
10    }
11    for (EventTag eventTag : eventTags){
12        if (eventTag.getEvent().getCreateDate().compareTo(date1)
13            > 0) {
14            eventsCurrent.add(eventTag);
15        }
16    }
17    if (eventsCurrent.isEmpty()){
18        return null;
19    }
20    else {
21        return eventsCurrent.stream().distinct().collect(
22            Collectors.toList());
23    }
24 }
```

Kiekvieną kartą naudotojui pakoregavus atributų sąrašą – algoritmas perskaičiuoja rekomenduojamų renginių sąrašą. Taip užtikrinama, kad visada yra pateikiamas naujausias sąrašas. Rekomenduojamų renginių sąrašas taip pat perskaičiuojamas, jei įkeliamas naujas renginys. Svarbu paminėti, kad algoritmas atrinka tik tuos renginius, kurie vyks ateityje arba vyksta dabar. Tad prie viso algoritmo pridėtas dar papildomas duomenų filtravimas pagal datą - netavaizduojami jau praėję renginiai.

### 3.3.3. Įvertinimas

Kitas būdas – reitingavimas, kuris gali suteikti daugiau informacijos apie tai, ką naudotojas mėgsta ir pagal tai daryti išvadas, kas galėtų jam patikti ateityje. Kuriamoje sistemoje šis būdas įgyvendintas taip, kad prisijungęs naudotojas galėtų vertinti renginius, kurie naudotojui patiko. Naudotojui palikus įvertinimą, sistemos filtravimo algoritmas išfiltruoja visus renginius, kurie atitinka įvertinimą ir yra panašaus turinio. Algoritmas veikia tokiu principu: pirmiausiai surandami renginiai, kurių įvertinimas yra aukštas, tada pagal renginio egzistuojančius atributus surandami panašūs, kurie galėtų patikti naudotojui. Tačiau ši sąlyga galioja tik jau įvykusiems renginiams. Jei renginys dar nepasibaigė, bet naudotojas įvertino jį pakankamai gerai, tada sistema rekomenduos jam šį renginį.

2 išėities kodas. Filtravimas pagal pateiktą įvertinimą kodas

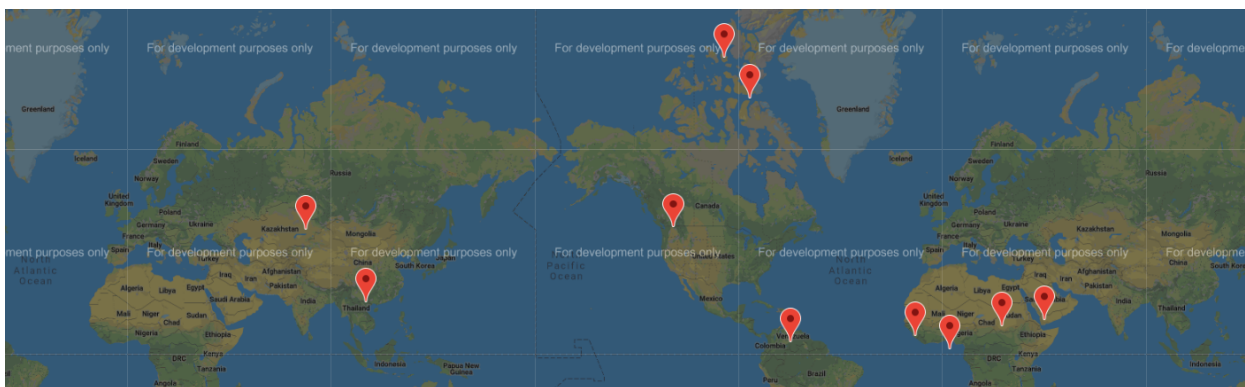
```
1 public List<Event> findById(long id) {
2     Date date1 = new Date();
3     List<Event> eventsCurrent = new ArrayList<>();
4     List<Rate> rates = ratingRepository.findAllByUserId(id);
5     List<Event> events = new ArrayList<>();
6     long maxRate = 0;
7     for (int i = 0; i < rates.size(); i++) {
8         for (int y = i+1; y < rates.size(); y++){
9             maxRate = rates.get(i).getId();
10            if ( rates.get(i).getEvent().getId().equals(rates
11                .get(y).getEvent().getId())){
12                if(maxRate < rates.get(y).getId()){
13                    maxRate = rates.get(y).getId();
14                    rates.remove(rates.get(i));
15                } else {
16                    rates.remove(rates.get(y));
17                }
18            }
19        }
20    }
21    for (Rate rate : rates){
22        if (rate.getId() == maxRate){
23            if(rate.getRate() >= 5) {
24                events.add(rate.getEvent());
25            }
26        }
27    }
28    for (Event event : events) {
29        if (event.getCreateDate().compareTo(date1) > 0) {
30            eventsCurrent.add(event);
31        }
32    }
33    return eventsCurre
34 }
```

Lygiai taip pat, kaip ir pirmasis algoritmas, šis algoritmas reaguoja į naudotojo paliekamus vertinimus ir iškart po to atnaujina rekomenduojamų renginių sąrašą. Svarbu paminėti, kad algoritmas

filtruoja tik tuos renginius, kurie vyks ateityje arba vyksta dabar. Taip užtikrinama, kad naudotojas gaus naujausių ir geriausiai atrinktų renginių sąrašą.

### 3.4. Renginių atvaizdavimas žemėlapyje

Žemėlapių modelis, sukurtas panaudojant „Google Maps“, pasiekiamas tik prisijungusiems naudotojams. Norint pamatyti žemėlapyje renginio vietą, reikia nueiti į detalią renginio informaciją, kuriuoje po renginio aprašymu pateikiamas žemėlapis su renginio vieta. Duomenų bazėje saugojamos renginių koordinatės: lng ir lan. Pagal saugojamas renginio koordinatas atvaizduojama jo vieta žemėlapyje. Taip pat administratoriaus aplinkoje yra galimybė šią informaciją koreguoti.



12 pav. Žemėlapių implementacijos pavyzdys

Toks atvaizdavimas yra labai populiarus ir naudingas visiems naudotojams, nes tai daug greitesnis būdas surasti norimo renginio lokaciją.

### 3.5. Informacija apie renginius

Kiekvienas prisijungęs naudotojas paspaudęs „Subscribe“ mygtuką gali gauti išsamią informaciją į savo elektroninį paštą. Tokiu būdu naudotojas turi reikiamą informaciją apie renginį. Taip pat šis metodas leis sistemos administratoriui atlikti analizę apie renginių populiarumą. Stebint susidomėjimą renginiais galima specializuotis į renginių kategoriją, kuri naudotojų vertinama palankiau. Visa tai gali administratorius rasti savo aplinkoje prie „Subscriptions“ žymeklio, kuris nuves į kitą puslapį, kuriame ir bus atvaizduojama visa reikalinga informacija.

3 išeities kodas. Informacijos gavimas el. paštu

```
1 public ResponseEntity <?> sendEmail(@RequestBody Transaction transaction)
  {
2     MimeMessage message = sender.createMimeMessage();
3     MimeMessageHelper helper = new MimeMessageHelper(message);
4     try {
5         helper.setTo(transaction.getUser().getEmail());
6         helper.setText(transaction.getEvent().getExplanation());
7         helper.setSubject(transaction.getEvent().getName());
8     } catch (MessagingException e) {
9         e.printStackTrace();
10    return new ResponseEntity <>(HttpStatus.CONFLICT);
```

```
11     }
12     sender . send ( message ) ;
13     return new ResponseEntity <>( HttpStatus . OK ) ;
14 }
```

Ši funkcija administratoriui padės stebėti aktyvų naudotojų domėjimąsi renginiais. Ši informacija yra naudinga sistemos administratoriui paslaugų analizės tikslais.

### 3.6. Komentarai po renginio

Prisijungęs naudotojas turi galimybę parašyti komentarą apie renginį. Komentarai apie renginį yra filtruojami ir yra pateikiami tik tie komentarai, kurie parašyti apie pasirinktą renginį. Tačiau naudotojas neturi galimybės koreguoti ar ištrinti komentaro. Visi komentarai saugojami duomenų bazės lentelėje. Įvedus naują komentarą sistema iš karto atsinaujina ir galima matyti naują komentarą. Komentarai yra anonimiški, kad nebūtų rodoma, kurie naudotojai juos parašė. Naujausias komentaras rodomas apačioje kitų komentarų.

### 3.7. Sistemos saugumas

Kuriant sistemą, kurioje naudotojas pateikia savo asmeninę informaciją, turi būti užtikrintas duomenų saugumas. Todėl sistemoje esančius duomenis pasiekia tik sistemos administratoriai. Naudotojui prisijungti prie sistemos pavyksta, jeigu jis įveda teisingą prisijungimo vardą ir slaptažodį. Kuo slaptažodis ilgesnis ir sudėtingesnis, tuo sudėtingiau yra atspėti kokį slaptažodį yra sugalvojęs naudotojas. Kiekvienam naujai prisiregistravusiam naudotojui yra suteikiama „user“ rolė, kuri užtikrina, jog naudotojas nematys administravimo funkcijų ir kitų naudotojų asmens duomenų.

#### 3.7.1. Naudotojo informacijos šifravimas

Siekiant apsaugoti prisijungusio naudotojo saugumą, „users“ lentelėje esantis slaptažodžio laukas yra užšifruojami pagal nurodytą raktą. Užšifravus lauką duomenų bazės lygmenyje galima užtikrinti dar didesnę naudotojo duomenų saugumą, nes kiti sistemos naudotojai administruojantys duomenų bazę negalės perskaityti naudotojo slaptažodžio, esančio lentelėje. Tai pagerina visos sistemos saugumą. Naudotojo informacija yra užšifruojama registracijos metu prieš visus duomenis įrašant į duomenų bazę.

#### 3.7.2. Puslapių ir funkcijų apsauga

Kuriant šią sistemą reikia užtikrinti, kad tam tikra naudotojų grupė galėtų matyti atitinkamus puslapius ir juose esančias funkcijas bei duomenis. Reikia suteikti roles, pagal kurias nustatomas sistemos funkcijų matomumas. Tik administratoriaus rolė gali pasiekti visas funkcijas sistemoje, išskyrus naudotoją, kuriam priskirta „user“ rolė. Sistemoje yra sukurtos šios rolės:

- administratoriaus – maršrutams, kuriuos gali pasiekti tik administratoriaus vaidmenį turintys naudotojai;
- prisijungusio naudotojo – maršrutams, kuriuos gali pasiekti prisijungę naudotojai. Kiekviena tarpinė funkcija priskiriama prie maršrutų.

Neprisijungusiems naudotojams, bandantiems pasiekti puslapius, kurie matomi tik prisijungusiems naudotojams, pateikiamas puslapis su klaidos pranešimu, kad tokio puslapio nėra ir pateikiama galimybė grįžti į pradinį puslapį. Ištrynimo patvirtinimo lange, apsaugant sistemą nuo netyčinio duomenų ištrynimo, paspaudus ant trynimo ikonos pateikiamas modalinis dialogo langas, kuriame reikia patvirtinti, ar tikrai norima pašalinti įrašą. Taip pat, jei norima išsaugoti naują renginį ar atnaujinti bet kokią informaciją - gražinami pranešimai su būseną, nurodančia, ar pavyko baigti veiksmą sėkmingai.

### **3.7.3. Formų validacijos**

Formų validacijos apsaugo duomenų bazę nuo netinkamų, klaidinančių, neaiškių duomenų. Siekiant to išvengti – naudojamos įvairios duomenų validacijos. Kiekviena sistemoje esanti forma yra tikrinama ar saugomi duomenys atitinka prasmę. Sistemos įvesties formos tikrinamos naudojant „Angular“ ir „Spring“ validacijos metodus. Jeigu bent vienas įvesties laukas yra netinkamai užpildytas – ekrane pasirodo pranešimas, ką būtent reikia koreguoti, norint užbaigti pradėtą veiksmą. Šiuo atveju, jei naudotojas blogai užpildo lauką, jo įrašas nėra išsaugojamas duomenų bazėje.

## **3.8. Sistemos testavimas**

Naujai kuriamą sistemą būtina tinkamai ištestuoti – tai sumažina klaidų tikimybę sistemą paleidus produkcinėje aplinkoje. Kol sistema yra kuriama, joje esančios klaidos yra lengviau ir greičiau ištaisomos nei sistemos naudojimo metu. Tokiu atveju, kai sistema yra pilnai naudojama, klaidos aptikimas užtrunka ženkliai ilgiau. Dėl šios priežasties sistema buvo nuolat testuojama jos kūrimo metu.

## Išvados ir rekomendacijos

Darbo galutinis rezultatas – sukurta automatinė rekomenduojamų renginių parinkimo sistema, kuri leidžia prisijungusiems naudotojams matyti jiems asmeniškai rekomenduojamus renginius, o administratoriams – valdyti sistemoje esančius įrašus ir stebėti naudotojų veiklą. Įvykdžius darbo tikslui pasiekti keliamus uždavinius padarytos išvados:

- Rekomenduojamų renginių pateikimas padeda greičiau rasti aktualius naudotojui renginius, palengvina informacijos apie renginį gavimą ir sumažina laiko trukmę skiriamą renginių paieškai;
- Kuriant sistemą svarbu atsižvelgti į naudotojo asmens duomenų saugojimą. Asmens duomenys turi būti pasiekiami tik tiems asmenims, kuriems yra suteikta prieiga arba rolė.
- Tokia sistema leidžia efektyviai reklamuoti renginius bei skatina domėjimąsi jais, kadangi naudotojai gali matyti renginių populiarumą.

Vis dėlto, darbo metu buvo sukurtas tik sistemos prototipas, kurį galima tobulinti ir vystyti toliau. sukurta sistema nėra galutinė ir ją patobulinti būtų galima:

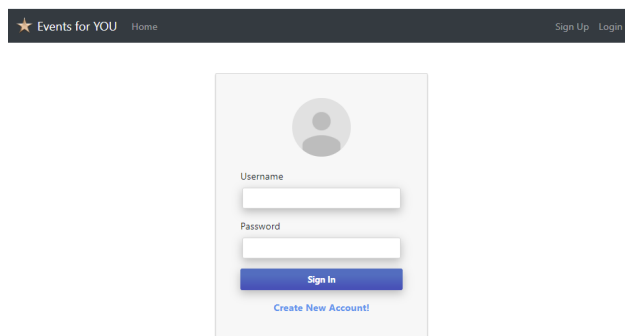
- Įgyvendinant didesnę renginių paieškos funkcionalumą.
- Sistemos duomenų bazėje galima saugoti specifinę renginių informaciją.

## Literatūra

- [1] Robin van Meteren and Maarten van Someren. Using Content-Based Filtering for Recommendation. NetlinQ Group, Gerard Brandtstraat 26-28, 1054 JK, Amsterdam.
- [2] Patrick Baudisch. Joining Collaborative and Content-based Filtering. Integrated Publication and Information Systems Institute IPSI.
- [3] Joseph A. Konstan, John Riedl, AI Borchers, and Jonathan L. Herlocker. Recommender Systems: A GroupLens Perspective. Dept. of Computer Science and Engineering University of Minnesota Minneapolis.
- [4] Dan R. Greening. Collaborative Filtering for Web Marketing Events. Chief Technology Officer, LikeMinds, Inc.
- [5] David M. Nichols. Implicit Rating and Filtering. In Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering, pp. 31-36. Budapest, Hungary(1997).
- [6] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh. Recommendation systems: Principles, methods and evaluation. (2015)

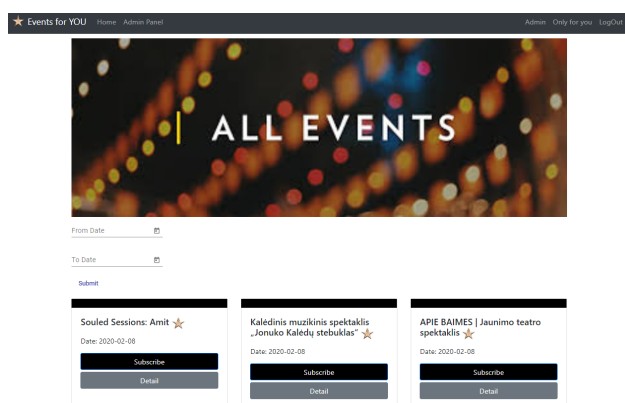
# Priedas Nr. 1

Sukurtos svetainės pavyzdžiai:



The screenshot shows a login form for 'Events for YOU'. At the top, there is a navigation bar with a star icon, 'Events for YOU', and 'Home'. On the right side of the bar are links for 'Sign Up' and 'Login'. The main content area features a light gray box with a circular profile icon placeholder. Below the icon are two input fields: 'Username' and 'Password'. A blue 'Sign In' button is positioned below the password field, and a link for 'Create New Account!' is located at the bottom of the box.

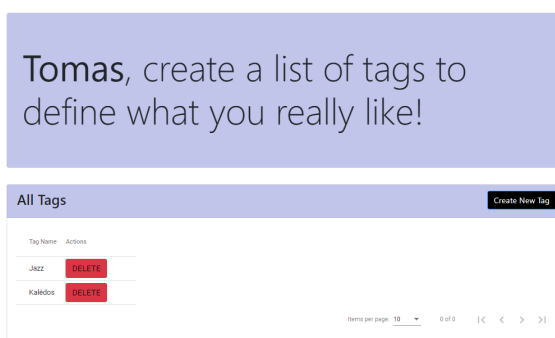
13 pav. Prisijungimas prie sistemos



The screenshot displays the 'ALL EVENTS' page. The top navigation bar includes 'Events for YOU', 'Home', 'Admin Panel', 'Admin', 'Data for you', and 'Logout'. The main header features a large banner with the text 'ALL EVENTS' over a background of colorful bokeh lights. Below the banner are two date filters: 'From Date' and 'To Date', each with a calendar icon. A 'Submit' button is located below the filters. The page shows three event cards. Each card includes the event title, a star icon, the date '2020-02-08', and buttons for 'Subscribe' and 'Detail'.

Event Title	Date	Buttons
Souled Sessions: Amit	2020-02-08	Subscribe, Detail
Kalėdinis muzikinis spektaklis „Jonuko Kalėdų stebuklas“	2020-02-08	Subscribe, Detail
APIE BAIMES   Jaunimo teatro spektaklis	2020-02-08	Subscribe, Detail

14 pav. Visų renginių atvaizdavimas pagrindiniame puslapyje

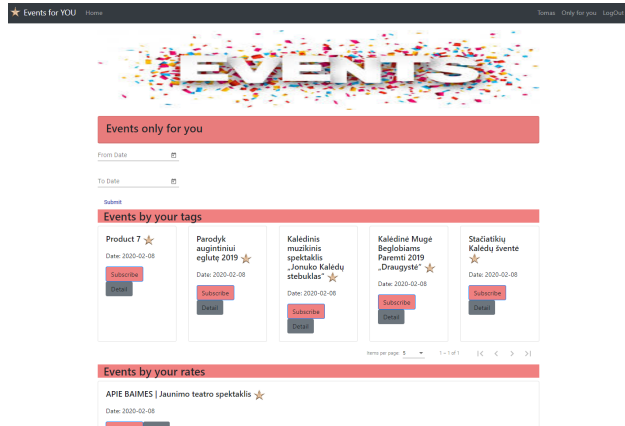


The screenshot shows a user profile page for 'Tomas'. A purple banner at the top contains the text: 'Tomas, create a list of tags to define what you really like!'. Below this is a section titled 'All Tags' with a 'Create New Tag' button. A table lists existing tags with their names and 'DELETE' buttons.

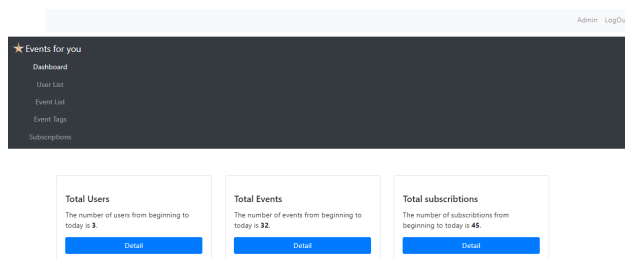
Tag Name	Actions
Jazz	DELETE
Kalėdos	DELETE

At the bottom of the table, there is a pagination control: 'Items per page: 10' and '0 of 0' with navigation arrows.

15 pav. Naudotojo suvedami atributai



16 pav. Sistemos atrinkti renginiai naudotojui



17 pav. Administratoriaus aplinka