VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis bakalauro darbas

Dizaino konkursų informacinė sistema

Atliko: 4 kurso, 1 grupės studentas
Irmantas Liepis (parašas)

Darbo vadovė:
dr. Joana Katina

Vilnius
2017

Turinys

Anotacija

Dizaino darbų kūrimo poreikis atsiranda siekiant įtvirtinti savo verslą ar kuriant konkrečią reklaminę kampaniją. Šio darbo tikslas – sukurti tokią sistemą, kurioje užsakovai galėtų paskelbti dizaino konkursus, o laisvai dirbantys dizaineriai galėtų patalpinti savo siūlomus darbus. Šiam tikslui pasiekti buvo įvertintos esančios alternatyvos ir panašaus pobūdžio tinklalapiai, nustatyti konkretūs reikalavimai bei svarstomi įvairūs architektūriniai sprendimai. Galutinis darbo rezultatas – sukurta funkcionuojanti sistema, gebanti registruoti užsakovus ir dizainerius, leidžianti užsakovams paskelbti dizaino konkursus, vertinti įkeltus darbus ir išrinkti nugalėtoją, o dizaineriams leidžianti dalyvauti konkursuose bei įkelti savo darbus, būti įvertintiems.

Summary

The main goal of this project was to create a system in which designers are able to publish design contests and freelance designers have an opportunity to participate in them by uploading their work. In order to achieve this goal, similar systems were analyzed, a list of requirements was established and various architectural options were taken into consideration. The end result is a functional website that allows for registration of clients and freelance designers, ability to rate and comment on the uploaded designs, as well as choosing a winner.

Įvadas

Šiais laikais mūsų gyvenimas sunkiai įsivaizduojamas be paslaugų, kurias suteikia informacinės sistemos. Informacinių sistemų yra įvairių: vienos jų skirtos konkrečioms verslo sritims, kitos skirtos tam tikrų organizacijų veiklai palaikyti. Šiame darbe aprašoma informacinė sistema „Dizaino konkursai“ pritaikyta lietuviškoms įmonėms, kurios sieka pozicionuoti savo verslą ar nori kurti reklaminę kampaniją.

Priklausomai nuo išsikeltų tikslų ir turimo biudžeto, dizaino darbams įmonės įprastai kreipiasi į reklamos agentūrą arba renkasi laisvai samdomą dizainerį. Viena iš problemų kreipiantis į reklamos agentūrą yra ta, kad tinkamam dizainui reikia didelio biudžeto, o kreipiantis į laisvai samdomą dizainerį, dizaino kokybė gali neatitikti įmonės reikalavimų. Pagrindinis sistemos „Dizaino konkursai“ tikslas – pateikti alternatyvą, kuri padėtų išspręsti minėtas problemas ir būtų lengvai prieinama.

Šio **darbo tikslas** yra sukurti sistemą, kurioje užsakovai galėtų skelbti dizaino konkursus, o laisvai dirbantys dizaineriai galėtų talpinti savo siūlomus darbus. Pagrindinės projekto metu iškilusios užduotys buvo išanalizuoti analogiškas sistemas, suprojektuoti teorinį sistemos modelį, sumodeliuoti duomenų bazės schemą, taip pat išsiaiškinti tam tikras architektūrines priemones bei programiškai įgyvendinti sistemos numatytąsias naudotojų funkcijas.

1. Panašių sistemų analizė

Ieškodamas panašių sistemų internete, buvo gan sunku atrasti panašaus pobūdžio sistemų. Mano ieškotos sistemos turėjo užsiimti laisvai samdomų dizainerių registracija, konkursų skelbimo funkcionalumu, bei galimybe užsakovui rinktis iš keleto dizainerių darbų. Panašaus pobūdžio puslapių daugiausiai radau forumuose. Vienas jų – **phpfusion-lt.com/dizaino-konkursai,f39**, kuriame konkursą paskelbti ir peržiūrėti gali bet kas, bei nėra minimalaus konkurso prizo.

Panaši sistema yra **uzdarbis.lt/f110/dizaino-konkursai**. Šiame forume nustatytas minimalus prizas, laikotarpis, galimybė pratęsti konkursų galiojimo laiką, tačiau galima paminėti keletą trūkumų: įkeliamus darbus gali peržiūrėti net neregistruoti naudotojai, įskaitant ir pačio konkurso laikotarpiu, mažas dizaino kategorijų skaičius.

Panašiausia į „Dizaino konkursų“ sistema yra tinklalapis **logokonkursas.lt**. Ši sistema registruotiems naudotojams leidžia skelbti dizaino konkursus, o konkursuose dalyvauti gali tik registruoti dizaineriai. Taip pat pritaikytos tokios funkcijos kaip minimalus prizas, fiksuotas laikotarpis. Tačiau yra keletas pastebimų trūkumų – mažas kategorijų pasirinkimas, o taip pat paskelbiamo konkurso laikotarpis nėra lankstus, konkurso paskelbimas yra apmokestintas (15% nuo prizinės sumos).

Paskutinė panaši sistema yra užsienietiškas tinklalapis **99designs.com**. Tai plačiai išvystyta dizaino konkursų sistema. Skelbiant konkursą galima labai detaliai išdėstyti reikalavimus ir panašių darbų pavyzdžius. Ši sistema pasižymi profesionalia išvaizda, animacijomis, dėmesiu detalėms. Tačiau sistema nėra pritaikyta Lietuvos rinkai, užsakovai turi mokėti mokestį, kuris priklauso nuo užsakymo pobūdžio. Taip pat nėra apmokestinamas vienas dizaino konkursas, o yra perkamas paslaugų paketas, pavyzdžiui, norint paskelbti dizaino konkursą, reikia nusipirkti paketą už $559, kurį įsigijus galima skelbti iki 15 tinklalapių dizaino konkursų. Jei užsakovas nori paskelbti dar ir logotipo konkursą, tai jam reikės įsigyti atskirą paketą logotipo konkursams, kurio minimali kaina yra $299. Į šias kainas įeina atlygis dizaineriams, kurie nugalės konkursuose, tačiau minimali paslaugų kaina yra per didelė Lietuvos naudotojams.

Minėtų sistemų palyginimas pavaizduotas lentelėje 1.

Sistema / Funkcija	phpfusion	uzdarbis.lt	logokonkursas.lt	99designs.com	Kuriama sistema
Logotipo konkurso minimali kaina	40 €	50 €	45 €	299 $	40 €
Konkurso paskelbimo forma	Nėra	Nėra	Yra	Yra	Yra
Dizainų peržiūros ribojamas	Nėra	Nėra	Yra	Yra	Yra
Privatūs konkursai	Nėra	Nėra	Yra	Yra	Yra
Dizainų įvertinimo metodai	Komentarai	Komentarai	Komentarai, 5 žvaigždučių sistema	Komentarai, 5 žvaigždučių sistema	Komentarai, 5 žvaigždučių sistema
Kategorijų skaičius	3	5	11	88	48
Konkursų paieška	Nėra	Nėra	Nėra	Nėra	Yra

1 lentelė. Sistemų palyginimas

2. Teorinis sistemos modelis

2.1. Bendras sistemos modelis

Šioje dalyje aprašomas sistemos veikimas ir jos reikalavimai.

2.1.1. Sistemos apžvalga

Siekiant pašalinti minėtų sistemų trūkumus, buvo sukurta paprastesnė, patogesnė ir ekonomiškesnė sistema „Dizaino konkursai“. Ši sistema leidžia užsakovams paskelbti konkursą norimam dizainui gauti, detaliai nurodant jo reikalavimus. Konkurso laikotarpiu dizaineriai gali įkelti jų sukurtus dizainus, o užsakovas gali įvertinti dizainerių įkeltus dizainus bei paprašyti pakoreguoti jau įkeltus dizainus. Kai konkurso laikas pasibaigia, užsakovas gali išsirinkti geriausiai tinkantį dizainą ir, sumokėjus dizaineriui, gauti visas teises tam dizainui naudoti. Ši sistema veikia kaip tarpininkas tarp užsakovo ir laisvai samdomų dizainerių.

2.1.2. Sistemos reikalavimai

Galutiniame darbo variante buvo įgyvendinti šie išsikelti funkciniai reikalavimai:

- Naudotojų registracija ir prisijungimas
- Skirtingų naudotojų tipų sąveika su sistema
- Individualios formos pritaikytos kiekvienai dizaino konkurso kategorijai
- Užsakovai gali patalpinti naują dizaino konkursą
- Užsakovai gali vertinti įkeltus dizainus
- Užsakovai gali rašyti komentarus jų paskelbtuose konkursuose
- Dizaineriai gali įkelti savo dizainus
- Dizaineriai gali pakeisti savo įkeltus dizainus
- Dizaineriai gali rašyti komentarus jų paskelbtuose darbuose
- Neregistruoti naudotojai gali peržiūrėti informaciją apie viešai paskelbtus konkursus

- Administratorius gali patvirtinti ar atmesti konkursus
- Administratorius gali pratęsti konkurso galiojimo trukmę

Taip pat sistemai buvo išsikelti šie nefunkciniai reikalavimai:

- Sistema naudotis turi būti paprasta ir intuityvu
- Sistemos naudotojo sąsaja turi būti lietuvių kalba
- Sistema turėtų sklandžiai veikti visose moderniose naršyklėse

Užsakovai gali užregistruoti dviejų tipų konkursus:

- Viešus – konkurso informaciją ir įkeltus dizainus gali peržiūrėti visi naudotojai, įskaitant neprisiregistravusius naudotojus.
- Privačius – konkurso įkeltus dizainus, komentarus ir įvertinimus gali matyti tik registruoti dizaineriai.

2.2. Sistemos naudotojai

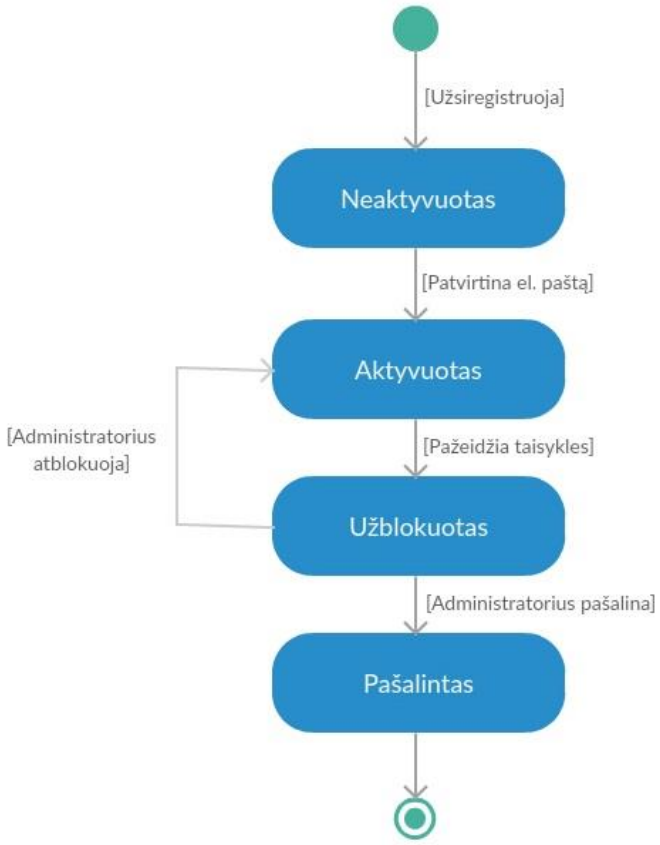
2.2.1. Naudotojų tipai

Dizaino konkursų sistemoje išskiriami naudotojų tipai:

- Svečiai – tai neregistruoti naudotojai. Jie gali peržiūrėti vykstančius ir pasibaigusius konkursus, peržiūrėti geriausių darbų galeriją.
- Nepatvirtinti naudotojai – tai naudotojai, kurie užsiregistravo, bet dar nepatvirtino savo el. pašto paskyros. Šio tipo naudotojas turi identišką funkcionalumą kaip svečias, bet patvirtinus paskyrą jis iškart įgauna užsakovo ar dizainerio statusą. Tai prevencinė priemonė nuo spamo botų.
- Užsakovai – tai patvirtinti užsakovai. Jie gali paskelbti dizaino konkursą, įvertinti įkeltus dizainus, išrinkti tinkamiausią dizainą.
- Dizaineriai – tai patvirtinti dizaineriai. Jie gali dalyvauti dizaino konkursuose, įkelti savo dizainus ir juos pakeisti, peržiūrėti įvertinimus ir komentarus.
- Administratorius – tai asmuo, atsakingas už sistemos palaikymą. Jis gali patvirtinti ar atmesti netinkamus konkursus, pratęsti konkurso galiojimo laikotarpį, tvarkyti registruotus naudotojus.

2.2.2. Naudotojų būsenos

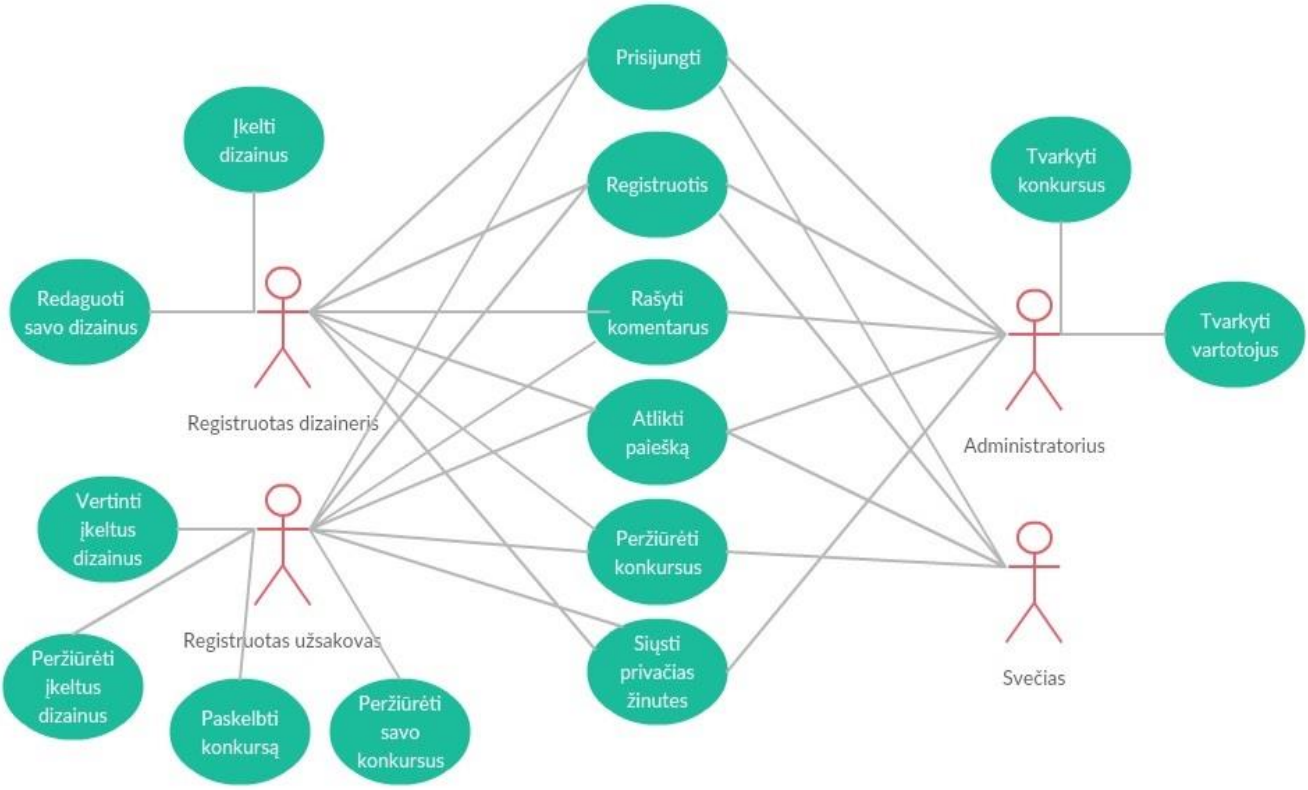
Šioje sistemoje naudotojams yra priskiriama viena iš 4 būsenų: neaktyvuotas, aktyvuotas, užblokuotas, pašalintas. Tik ką prisiregistravęs naudotojas laikomas neaktyvuotu. Jam išsiunčiamas el. laiškas su patvirtinimo nuoroda, kurią atidarius naudotojo būsena pakeičiama į aktyvuotą. Jei naudotojas pažeidžia sistemoje nurodytas taisykles ar viršija leistinų prisijungimo bandymų skaičių, jo būsena pakeičiama į užblokuotą atitinkamai administratoriaus sprendimu ar automatiškai. Administratoriui nusprendus, kad naudotoją galima atblokuoti, jo būsena pakeičiama į aktyvuotą. Kitu atveju, naudotojo būsena pakeičiama į pašalintą. Šią būsenų kaitą pavaizduota 2 pav.



2 pav. Naudotojų būsenos kaitos diagrama

2.2.3. Naudotojų funkcijos

Naudotojų naudojimosi funkcijos pavaizduotos 3 pav.



3 pav. Sistemos naudotojų naudojimo atvejų diagrama

Toliau detaliai aprašomi 3 pav. pavaizduoti panaudojimo atvejai.

1. PANAUDOJIMO ATVEJIS: Registruotis
Naudotojas/Aktorius: Visi naudotojai **Aprašas:** Norint paskelbti konkursus ar juose dalyvauti, naudotojas privalo užsiregistruoti sistemoje. Naudotojas atidaro registracijos formą ir ją užpildo. Jei forma užpildoma tinkamai, naudotojas išsaugomas sistemoje. **Sąlyga „prieš“:** Naudotojas nori užsiregistruoti sistemoje. **Sužadinimo sąlyga:** Naudotojas atidaro registracijos formos langą ir užpildo registracijos formą. **Sąlyga „po“:** Naudotojas išsaugomas duomenų bazėje, naudotojui išsiunčiamas elektroninis laiškas su patvirtinimo nuoroda.

1. Panaudojimo atvejo aprašas

2. PANAUDOJIMO ATVEJIS: Prisijungti
Naudotojas/Aktorius: Visi naudotojai **Aprašas:** Norint pilnavertiškai naudotis sistemos funkcionalumu, naudotojas privalo būti prisijungęs. Atidarius prisijungimo formą ir ją užpildžius, naudotojas prijungiamas į sistemą. **Sąlyga „prieš“:** Naudotojas turi būti prisiregistravęs, duomenų bazėje yra naudotojo įrašas. **Sužadinimo sąlyga:** Naudotojas nori prisijungti prie sistemos, kad galėtų naudotis daugiau sistemos funkcijų. **Sąlyga „po“:** Naudotojas gali naudotis atitinkamomis sistemos funkcijomis, naudotojo prisijungimas saugomas naršyklės sesijos saugykloje.

2. Panaudojimo atvejo aprašas

3. PANAUDOJIMO ATVEJIS: Patvirtinti paskyrą
Naudotojas/Aktorius: Nepatvirtintas naudotojas **Aprašas:** Norint naudotis registruoto naudotojo funkcionalumu, naudotojas privalo patvirtinti savo el. pašto adresą. Kai naudotojas užsiregistruoja sistemoje, jam išsiunčiama patvirtinimo nuoroda, kurią atidarius naudotojo paskyra patvirtinama. **Sąlyga „prieš“:** Naudotojas turi būti užsiregistravęs sistemoje, naudotojas nori patvirtinti savo paskyrą. **Sužadinimo sąlyga:** Naudotojas atidaro nuorodą, kuri jam buvo nusiųsta į el paštą užsiregistravus sistemoje. **Sąlyga „po“:** Naudotojo paskyra atnaujinama duomenų bazėje, naudotojas gali naudotis visomis jam suteiktomis sistemos funkcijomis.

3. Panaudojimo atvejo aprašas

4. PANAUDOJIMO ATVEJIS: Peržiūrėti konkursus
Naudotojas/Aktorius: Visi naudotojai **Aprašas:** Naudotojas atidaro konkursų skiltį ir gali peržiūrėti vykstančius ir pasibaigusius konkursus. **Sąlyga „prieš“:** Naudotojas nori peržiūrėti vykstančius konkursus. **Sužadinimo sąlyga:** Naudotojas atidaro konkursų skiltį. **Sąlyga „po“:** Naudotojas mato visų konkursų sąrašą ir gali peržiūrėti jų informaciją.

4. Panaudojimo atvejo aprašas

5. PANAUDOJIMO ATVEJIS: Ieškoti konkurso
Naudotojas/Aktorius: Visi naudotojai **Aprašas:** Naudotojas į paieškos lauką įveda dalį konkurso pavadinimo ir sistema filtruoja esamus konkursus pagal paieškos frazę. **Sąlyga „prieš“:** Naudotojas nori atrasti konkretų konkursą. **Sužadinimo sąlyga:** Naudotojas įveda paieškos frazę į paieškos laukelį. **Sąlyga „po“:** Sistema atnaujina konkursų sąrašą, kad jis rodytų tik konkursus atitinkančius paieškos frazę.

5. Panaudojimo atvejo aprašas

6. PANAUDOJIMO ATVEJIS: Rašyti komentarus
Naudotojas/Aktorius: Administratorius, dizaineris, užsakovas **Aprašas:** Naudotojas prisijungia prie sistemos ir atidaręs konkurso informacijos puslapį įveda komentarą į pateiktą lauką. Komentaras pridedamas ir išsaugomas duomenų bazėje. **Sąlyga „prieš“:** Naudotojas nori parašyti komentarą. Naudotojas turi būti prisijungęs prie sistemos, jo paskyra turi būti patvirtinta. **Sužadinimo sąlyga:** Naudotojas įveda komentarą į numatytą tekstinį lauką. **Sąlyga „po“:** Komentaras pridedamas prie konkurso ir išsaugomas duomenų bazėje.

6. Panaudojimo atvejo aprašas

7. PANAUDOJIMO ATVEJIS: Siųsti privačias žinutes
Naudotojas/Aktorius: Administratorius, dizaineris, užsakovas **Aprašas:** Patvirtintas naudotojas gali komunikuoti su kitais dizaineriais, užsakovais ar administratoriumi atidaręs savo pašto paskyrą sistemoje, nurodęs gavėjo slapyvardį ir irašęs žinutę. Išsiųsta žinutė išssaugoma duomenų bazėje, o žinutės gavėjas informuojamas el. paštu. **Sąlyga „prieš“:** Naudotojas nori parašyti žinutę kitam naudotojui. Naudotojas turi būti prisijungęs prie sistemos, jo paskyra turi būti patvirtinta. **Sužadinimo sąlyga:** Naudotojas užpildo privačios žinutės siuntimo formą ir ją išsiunčia. **Sąlyga „po“:** Žinutė išsaugoma duomenų bazėje, o gavėjas apie naują žinutę informuojamas el. paštu.

7. Panaudojimo atvejo aprašas

8. PANAUDOJIMO ATVEJIS: Paskelbti konkursą
Naudotojas/Aktorius: Užsakovas **Aprašas:** Užsakovas užpildo norimo dizaino paskelbimo formą ir taip paskelbiamas naujas dizaino konkursas. **Sąlyga „prieš“:** Naudotojas nori paskelbti naują dizaino konkursą. Naudotojas turi būti patvirtintas užsakovas. **Sužadinimo sąlyga:** Užsakovas pasirenka dizaino kategoriją ir užpildo jos formą. **Sąlyga „po“:** Konkursas išsaugomas duomenų bazėje. Kad jis taptų matomas kitiems, konkursą privalo patvirtinti administratorius.

8. Panaudojimo atvejo aprašas

9. PANAUDOJIMO ATVEJIS: Peržiūrėti įkeltus dizainus
Naudotojas/Aktorius: Užsakovas **Aprašas:** Kai dizaineriai įkelia naujus dizainus, juos peržiūrėti gali tik užsakovas. **Sąlyga „prieš“:** Naudotojas nori peržiūrėti įkeltus dizainus. Naudotojas turi būti prisijungęs kaip patvirtintas užsakovas, konkursas yra paskelbtas ir nėra užbaigtas. **Sužadinimo sąlyga:** Užsakovas atidaro savo paskelbto konkurso puslapį. **Sąlyga „po“:** Rodomi visi įkelti dizainai.

9. Panaudojimo atvejo aprašas

10. PANAUDOJIMO ATVEJIS: Vertinti įkeltus dizainus
Naudotojas/Aktorius: Užsakovas **Aprašas:** Įkeltus dizainus užsakovas gali vertinti nuo 1 iki 5 žvaigždučių arba palikti komentarą dizaino autoriui. **Sąlyga „prieš“:** Naudotojas nori įvertinti ar pakomentuoti įkeltą dizainą. Naudotojas turi būti prisijungęs kaip patvirtintas užsakovas, konkursas yra paskelbtas ir nėra užbaigtas. **Sužadinimo sąlyga:** Užsakovas įvertina dizainą arba jį pakomentuoja. **Sąlyga „po“:** Dizaino reitingas ar komentaras išsaugomas duomenų bazėje. Pagal šiuos įvertinimus dizaineris gali modifikuoti esamą dizainą ar įkelti kitą.

10. Panaudojimo atvejo aprašas

11. PANAUDOJIMO ATVEJIS: Peržiūrėti savo konkursus
Naudotojas/Aktorius: Užsakovas **Aprašas:** Užsakovas gali pamatyti savo paskelbtų konkursų sąrašą, ir taip lengviau pasiekti konkretaus konkurso puslapį. **Sąlyga „prieš“:** Užsakovas nori peržiūrėti savo paskelbtų konkursus. Naudotojas turi būti prisijungęs kaip patvirtintas užsakovas. **Sužadinimo sąlyga:** Užsakovas atidaro savo paskelbtų konkursų puslapį. **Sąlyga „po“:** Filtruojami tam užsakovui priklausantys konkursai, ant jo paspaudus nukreipiama į to konkurso puslapį.

11. Panaudojimo atvejo aprašas

12. PANAUDOJIMO ATVEJIS: Įkelti dizainus
Naudotojas/Aktorius: Dizaineris **Aprašas:** Konkurso laikotarpiu dizaineris gali įkelti vieną ar kelis dizainus. Dizaineris pasirenka dizaino failus iš savo kompiuterio. Šie dizainai išsaugomi serverio failų sistemoje ir dizainų nuorodos išsaugomos duomenų bazėje. **Sąlyga „prieš“:** Naudotojas nori įkelti savo sukurtą dizainą. Naudotojas turi būti prisijungęs kaip patvirtintas dizaineris, konkursas yra paskelbtas ir nėra užbaigtas. **Sužadinimo sąlyga:** Dizaineris paspaudžia failų įkelimo mygtuką ir pasirenka tinkamus dizainų failus. **Sąlyga „po“:** Dizainai įkeliami į serverio failų sistemą ir jų nuorodos išsaugomos duomenų bazėje.

12. Panaudojimo atvejo aprašas

13. PANAUDOJIMO ATVEJIS: Redaguoti savo dizainus
Naudotojas/Aktorius: Dizaineris **Aprašas:** Vykstant konkursui dizaineris gali pakeisti jo įkelta dizainą kitu. Dizaineris pasirenka dizaino failą iš savo kompiuterio. Šis dizainas ir jo nuoroda išsaugomi duomenų bazėje. **Sąlyga „prieš“:** Naudotojas nori pakeisti jau įkeltą dizainą. Naudotojas turi būti prisijungęs kaip patvirtintas dizaineris, konkursas yra paskelbtas ir nėra užbaigtas.

Sužadinimo sąlyga: Dizaineris paspaudžia failų įkėlimo mytguką ir pasirenka tinkamą dizaino failą. **Sąlyga „po“:** Dizainas įkeliamas į serverio failų sistemą ir jo nuoroda pakeičiama duomenų bazėje.

13. Panaudojimo atvejo aprašas

14. PANAUDOJIMO ATVEJIS: Tvarkyti konkursus
Naudotojas/Aktorius: Administratorius **Aprašas:** Paskelbus konkursą, jį privalo patvirtinti administratorius, o taip pat konkursą galima pratęsti. Tai padarius, konkurso statusas atitinkamai pakeičiamas ir išsaugomas duomenų bazėje. **Sąlyga „prieš“:** Administratorius nori pakeisti konkurso būseną. Naudotojas turi būti administratorius. **Sužadinimo sąlyga:** Patikrinęs konkurso informaciją, administratorius patvirtina, atmeta ar pratęsia konkursą paspausdamas atitinkamą mygtuką. **Sąlyga „po“:** Konkurso statusas pakeičiamas į „aktyvų”, „atmestą” ar „pratęstą”, ir išsaugomas duomenų bazėje.

14. Panaudojimo atvejo aprašas

15. PANAUDOJIMO ATVEJIS: Tvarkyti naudotojus
Naudotojas/Aktorius: Administratorius **Aprašas:** Naudotojui pažeidus sistemos taisykles, administratorius gali juos užblokuoti laikinai arba pašalinti visam laikui. Tai padarius naudotojo būseną pakeičiama atitinkamai į “užblokuotą” ar “pašalintą”. **Sąlyga „prieš“:** Administratorius nori pakeisti naudotojo būseną. Naudotojas pažeidžia sistemos naudojimosi taisykles. **Sužadinimo sąlyga:** Administratorius atidaro naudotojo profilį valdymo panelėje ir patvirtina naudotojo būsenos pakeitimą. **Sąlyga „po“:** Naudotojo būsena pakeičiama ir išsaugoma duomenų bazėje.

15. Panaudojimo atvejo aprašas

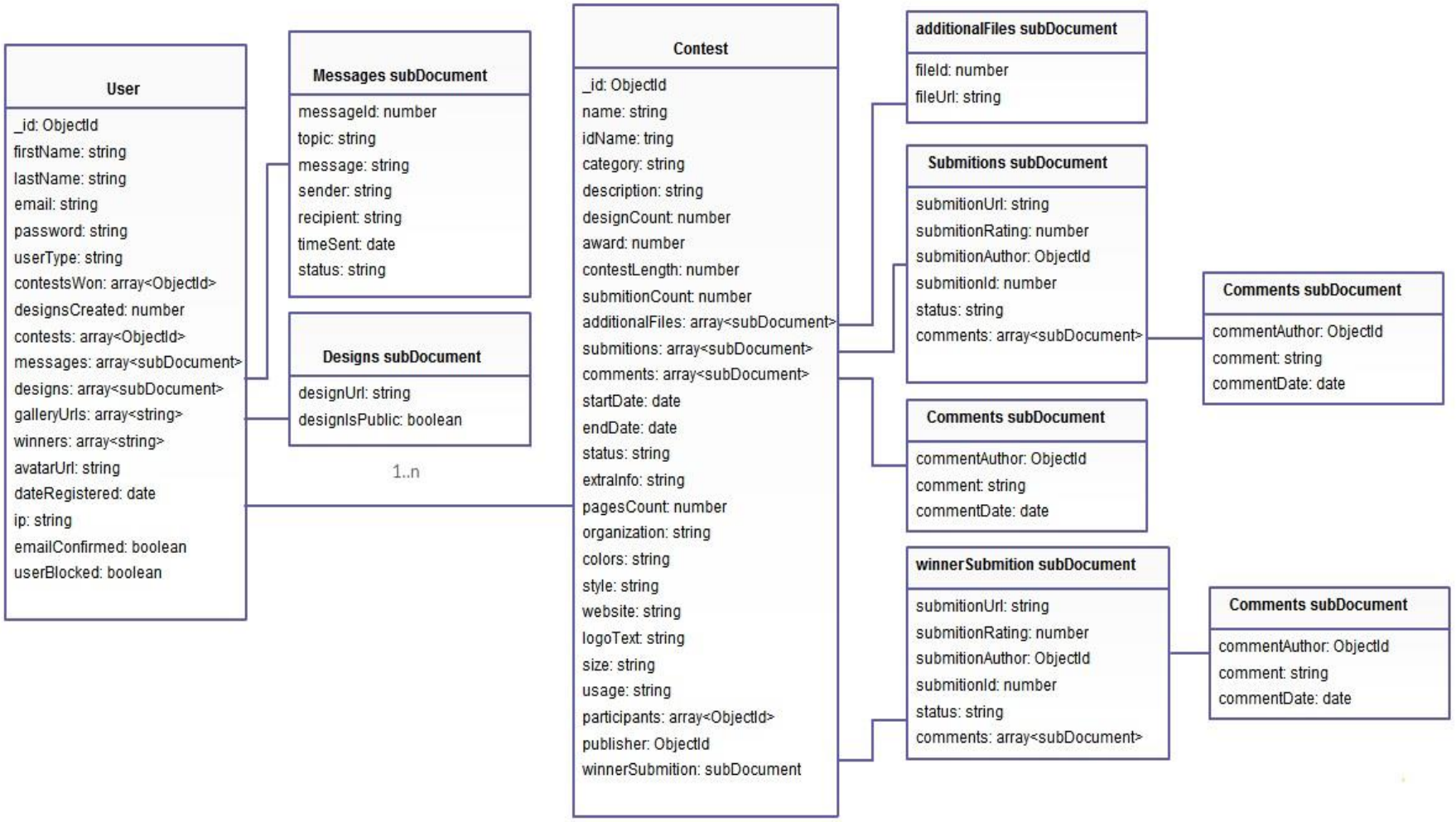
2.3. Duomenų bazės modelis

„Dizaino konkursų“ sistemos duomenys saugomi nereliacinėje NoSQL duomenų bazėje. Tai dokumentais orientuotas duomenų modelis, o šie yra saugomi kolekcijose. „Dizaino konkursų“ sistemoje duomenys saugomi dviejose kolekcijose: *User* ir *Contest*. Ryšys tarp *User* ir *Contest* kolekcijų yra **vienas su daug**. Tai reiškia kad vienas naudotojas gali sukurti daug konkursų, bet vienas konkursas tegali priklausyti vienam naudotojui (užsakovui). Kiekvienam naudotojo ar konkurso dokumentui automatiškai priskiriama unikali lauko *_id* reikšmė, pagal kurią bus daromos užklausos į duomenų bazę.

Kolekcijoje *User* saugomi dokumentai susiję su naudotoju: prisijungimo duomenys, paskelbti konkursai ir dizainų informacija. Laukuose *firstName*, *lastName*, *email*, *password*, *userType* – saugoma iš registracijos formos laukų surinkta informacija. Lauke *password* saugomas *bcrypt* slaptažodžio santraukos funkcija (angl. *hash function*) užšifruotas slaptažodis.[3] Jei naudotojas prisiregistruoja kaip dizaineris, jam inicializuojami papildomi *contestWon* ir *designsCreated* laukai. Laukas *emailConfirmed* skirtas išsaugoti *boolean* reikšmei, kuri nurodo ar naudotojo paskyra yra patvirtinta (*true)* ar ne (*false)*. Naudotojui užsiregistravus, jo IP adresas išsaugomas lauke *ip* tam, kad jeigu naudotojas bus užblokuotas, šis IP adresas bus įtraukiamas į draudžiamų IP sąrašą. Naudotojo privačios žinutės saugomos *messages* subdokumente, į kurį įeina tokia informacija kaip siuntėjas, tema, žinutė ir siuntimo laikas.

Kolekcijoje *Contest* saugomi dokumentai, kuriuose saugoma informacija apie paskelbtus konkursus ir jų specifinę informaciją. Kadangi dauguma kategorijų turi formas su unikaliais laukais, nemaža dalis kolekcijos laukų lieka nepanaudoti. Kiekviena forma turi 5 privalomus laukus: *name* – konkurso pavadinimas, *category* – konkurso kategorija, *description* – konkurso aprašymas, *award* – prizas laimėjus konkursą, *contestLength* – konkurso trukmė dienomis. Subdokumentas *additionalFiles* skirtas saugoti informacijai apie pagalbinius failus, pridėtus prie konkurso. Subdokumente *submitions* saugoma informacija apie įkeltus dizainus: dizaino URL, pagal kurį dizainai įkeliami iš serverio failų sistemos, dizaino reitingas bei komentarai. Taip pat išsaugoma informaciją apie konkurso pradžią, pabaigą ir statusą.

Visa sistemos duomenų bazės diagrama pavaizduota 4 pav.



4 pav. Sistemos duomenų bazės diagrama

3. Praktinė dalis

Šioje dalyje aprašomos pasirinktos technologijos ir praktinio sistemos įgyvendinimo architektūriniai sprendimai.

3.1. Technologijos

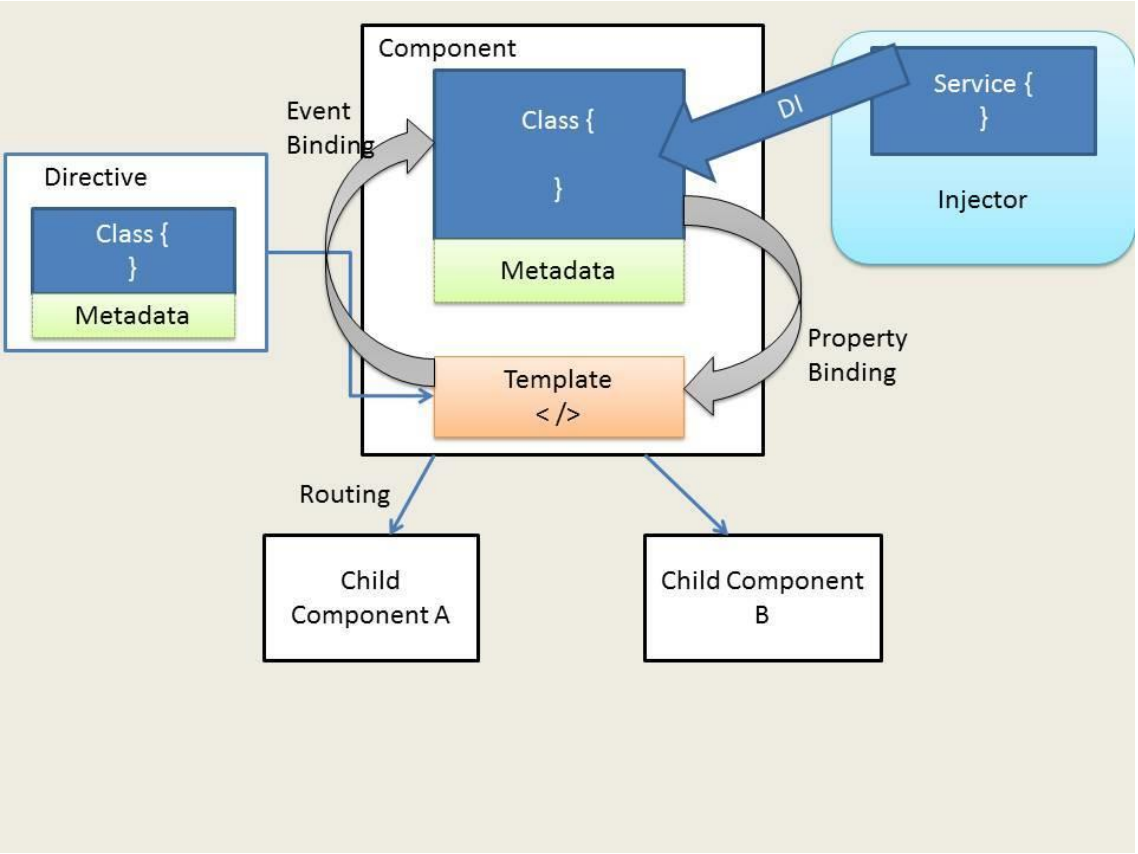
„Dizaino konkursų“ projektas sukurtas remiantis *MEAN stack (MongoDB, Express, AngularJS, Node.js)* technologijomis. Serverio pusės programavimui pasirinkau Node.js aplinką. Šį pasirinkimą lėmė šio karkaso populiarumas, paprastumas, ankstesnė darbo patirtis ir šiam projektui tinkama architektūra. Naudotojo sąsajos (*front-end*) realizavimui pasirinkau AngularJS 2 karkasą. Šį karkasą pasirinkau dėl jo unikalios architektūros, laisvai pasirenkamų duomenų struktūrų, bei noro tobulinti įgūdžius su šiuo perspektyviu karkasu. Pagrindinių naudotų technologijų sąrašas nurodytas žemiau:

- AngularJS 2 – JavaScript karkasas naudotojo sąsajos kūrimui
- Typescript programavimo kalba
- Bootstrap – HTML, CSS karkasas naudotojo sąsajos kūrimui
- Node.js platforma
- MongoDB duomenų bazė
- Mongoose tvarkyklė (*driver*) darbui su MongoDB
- Express tinklinių paslaugų programavimo karkasas
- NPM – Node.js paketų valdiklis
- System.js – tai universalus dinaminis modulių krovimo įrankis
- JavaScript užduočių paleidimo įrankis Gulp

3.2. Bendra sistemos architektūra

Ši sistema priskiriama *Single Page Application* tipui. Tai reiškia, kad visas reikiamas kodas (HTML, CSS, JavaScript užkraunamas vienu įkėlimu. Tokiu būdu aplikacijos gyvavimo laikotarpiu puslapis neperkraunamas, o papildomi duomenys užkraunami tik po atitinkamų naudotojo veiksmų.

AngularJS 2 naudoja unikalią komponentais pagrįstą architektūrą. Komponentai pasižymi tuo, kad juose galima aprašyti tik su tam tikru funkcionalumu susijusį kodą, o tai reiškia, kad komponentai yra sujungti atlaidžiai ir juos lengva testuoti. Tokio komponento pavyzdys, parodantis sąveikas (angl. *interactions*) su aplinkiniais elementais, pavaizduotas 5 pav.

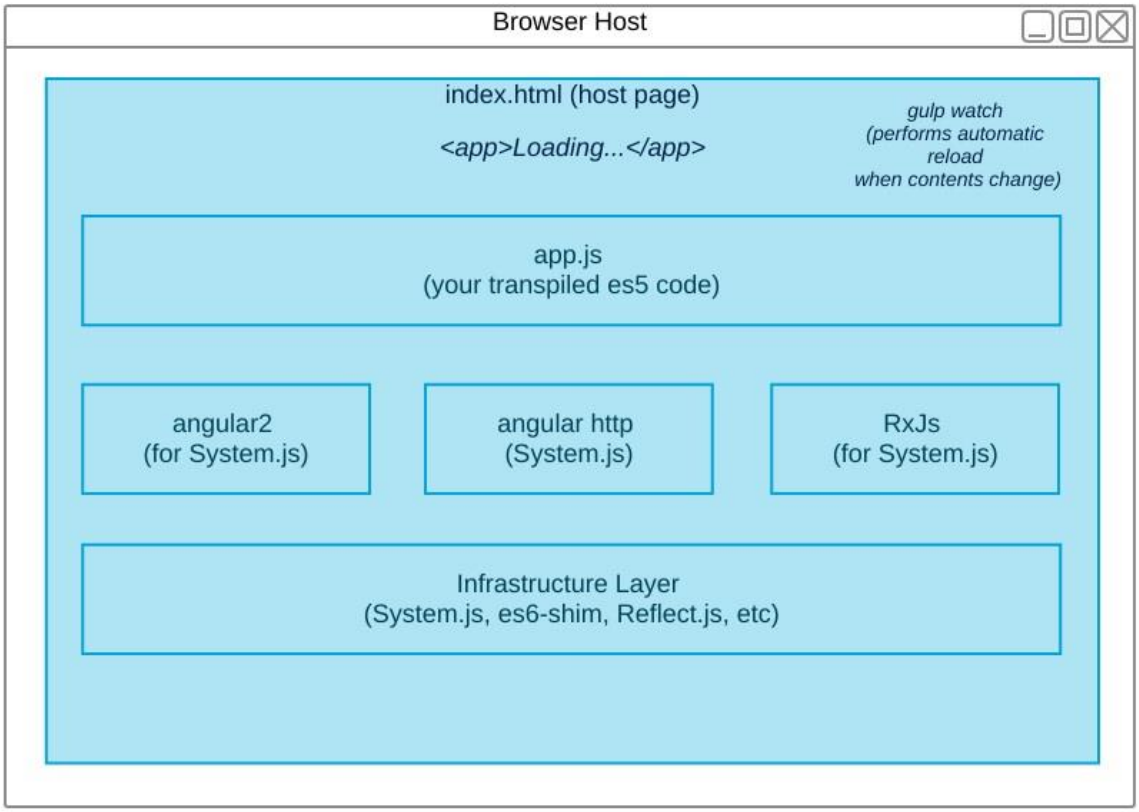


5 pav. AngularJS 2 komponento struktūros diagrama

AngularJS 2 architektūra leidžia labai lankstų duomenų stuktūros modelių pasirinkimą. Šiam projektui aš pasirinkau komponentams duomenis įkelti iš servisų, kurie gauna savo duomenis iš HTTP užklausų į mano sukurtą *web API*, o šį galima pasiekti adresu */api/v1/*. Servisai yra įkeliami į komponentus injektoriaus pagalba *dependency injection* principu. Tai reiškia, kad šie servisai yra *singleton* tipo ir kiekvienas jį naudojantis komponentas gauna naują serviso kopiją [1].

Servisų metodai naudoja *Observable* objektą duomenims persiųsti iš API į servisą ir iš serviso į komponentą. Obervable objektas reprezentuoja kolekciją, į kurią galima įkelti duomenis. Šis objektas leidžia kitiems objektams prisiregistruoti prie *Observable* objekto ir kai šie duomenys (iš API) apdorojami, jie persiunčiami į komponentus, kuriems tų duomenų reikia. Šis mechanizmas dar žinomas kaip *observer pattern* [2].

Kaip ši aplikacija užkraunama naršyklėje pavaizduota 6 pav.

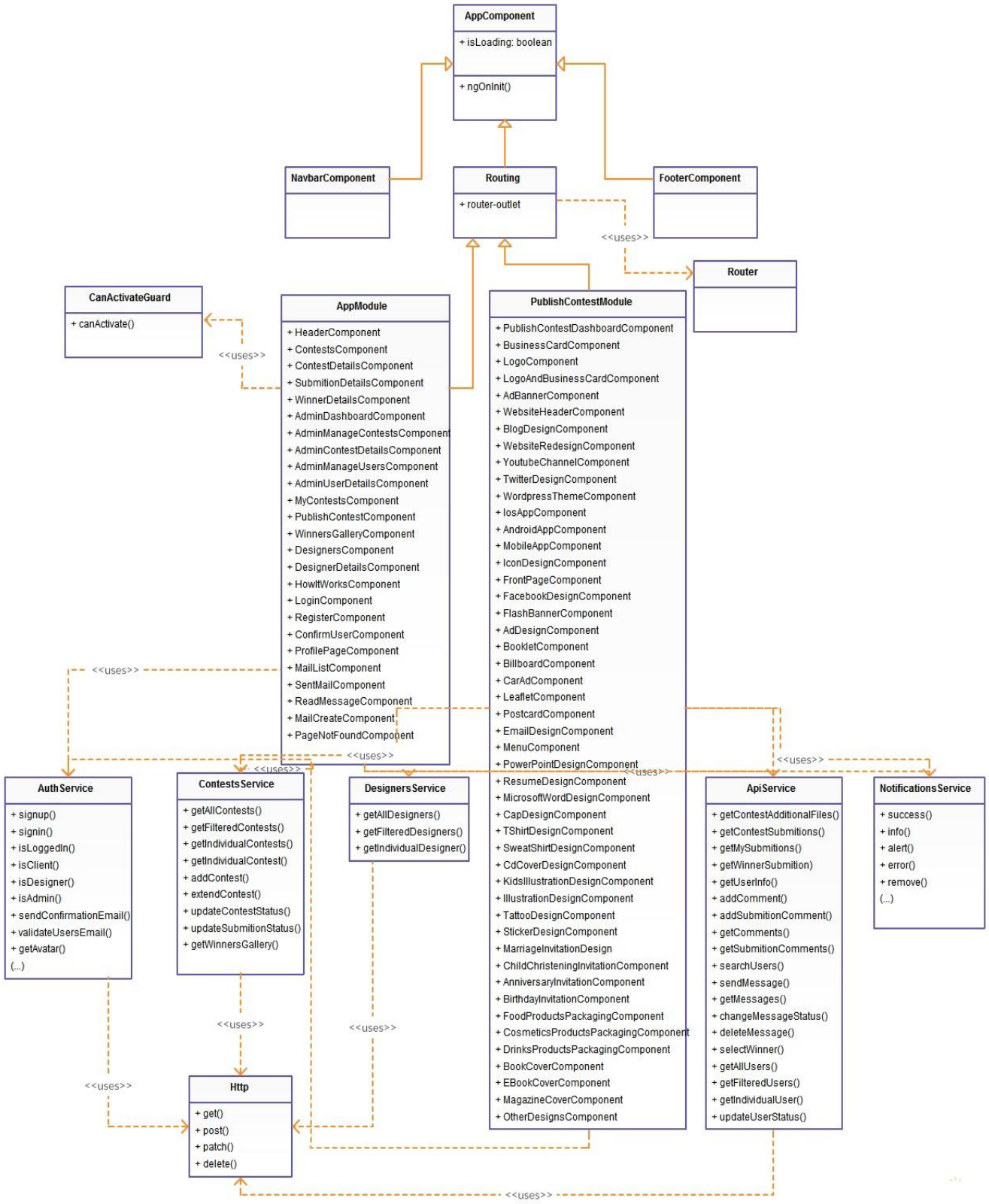


6 pav. Aplikacijos užkrovimo naršyklėje struktūra

Kaip matome diagramoje, aplikacijoje naudojama RxJs (*Reactive extensions for javaScript)* rinkinys, leidžiantis servisuose ateinančius duomenis modifikuoti į *Observable* objektą naudojant tokius operatorius kaip „map“ ar „filter“. System.js nurodo aplikacijai kur reikia ieškoti atitinkamų modulių ir komponentų. Kadangi komponentai buvo programuojami su TypeScript programavimo kalba, šiuos failus reikėjo sukompiliuoti į ES5 sintaksę, kurią pripažįsta visos modernios naršyklės. Tai buvo įgyvendinta naudojant Gulp užduotį „build:production“, kuri perrašė TypeScript failus į ES5 sintaksę, tada juos minifikavo, ir galiausiai suspaudė viską į vieną failą, pavadintą „app.js“, kuris ir naudojamas *index.html* faile paleidžiant sistemą naršyklėje.

„Dizaino konkursų“ sistemoje naudojami 2 moduliai: *AppModule* ir *PublishContestModule*. Moduliai leidžia deleguoti tam tikrų komponentų grupę į atskirus modulius, kurių nebūtina užkrauti iki kol naudotojui reikia pasinaudoti šio modulio komponentais. Modulyje aprašomi naudojami komponentai, trečių šalių bibliotekos ir servisai [4]. Abu moduliai naudoja bendrus servisus duomenims ir pranešimams gauti, o šie servisai naudoja HTTP modulį, skirtą darbui su HTTP užklausomis. Modulyje *AppModule* naudojama didžioji dalis komponentų, susijusių su prisijungimu, konkursų informacija, privačiomis žinutėmis ir t.t. Modulis *PublishContestModule* aktyvuojamas, kai naudotojas nori paskelbti naują dizaino konkursą. Šiame modulyje naudojami komponentai, kuriuose nurodytas kategorijų sąrašas ir kiekvienos kategorijos formos.

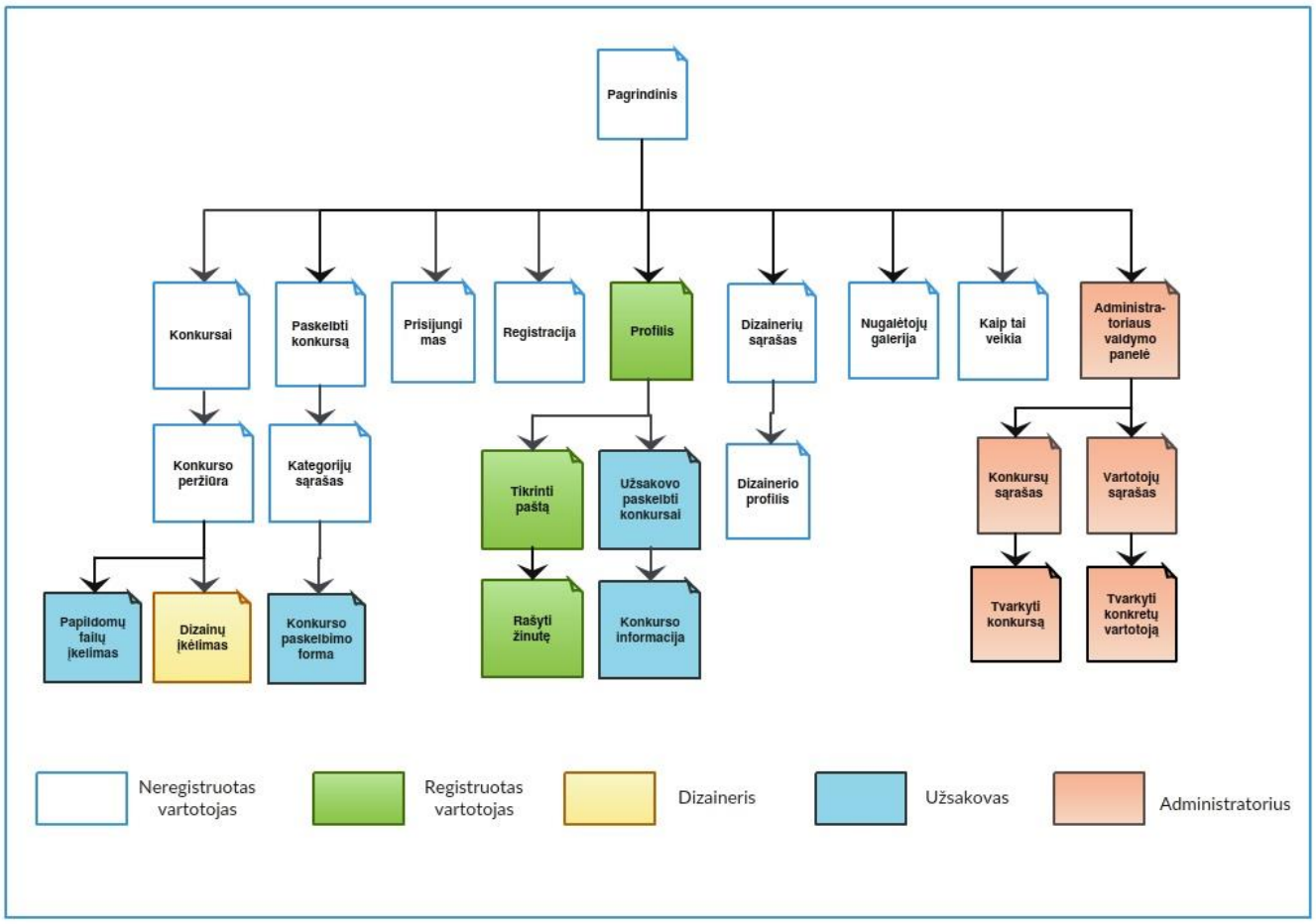
Aukštame lygyje paleidus aplikaciją yra naudojamas komponentas *AppComponent*, kuris sudarytas iš 3 komponentų: *NavbarComponent, router-outlet* ir *FooterComponent*. Tai *SPA* principas, kur kiekviename puslapyje rodomas *NavbarComponent* viršuje ir *FooterComponent* apačioje, o vietoj *router-outlet* dinamiškai užkraunamas vis kitas komponentas priklausomai nuo naudotojo navigacijos veiksmų. Verta paminėti, kad kai kurie komponentai iš *AppModule* yra apsaugomi naudojant *CanActivateGuard* klasę, kuri patikrina naudotojo būseną, prieš leidžiant pasiekti norimą puslapį. Tai itin aktualu administratoriaus valdymo ar privačių žinučių archyvui pasiekti. Naudojamų modulių, komponentų ir servisų sąryšiai pavaizduoti 7 pav.



7 pav. Sistemos naudojamų komponentų sąryšių diagrama

3.3. Grafinės sąsajos struktūra

Sistemos grafinę naudotojo sąsają sudaro puslapiai, parašyti su HTML kodu ir AngularJS 2 struktūrinėmis direktyvomis (angl. *Structural directives*). Šios direktyvos leidžia dinamiškai užkrauti duomenis, priklausomai nuo naudotojo būsenos ir tipo. Pavyzdžiui, neprisijungęs naudotojas nemato kitų sistemos funkcijų, kurias jis matytų prisijungęs prie sistemos. Taip pat, jeigu naudotojas prisijungia nepatvirtinęs savo el. pašto paskyros, jis apie tai įspėjamas ir didžioji dalis funkcionalumo jam uždraudžiama. Visa svetainės navigacijos schema pavaizduota 8 pav.



8 pav. Sistemos navigacijos schema

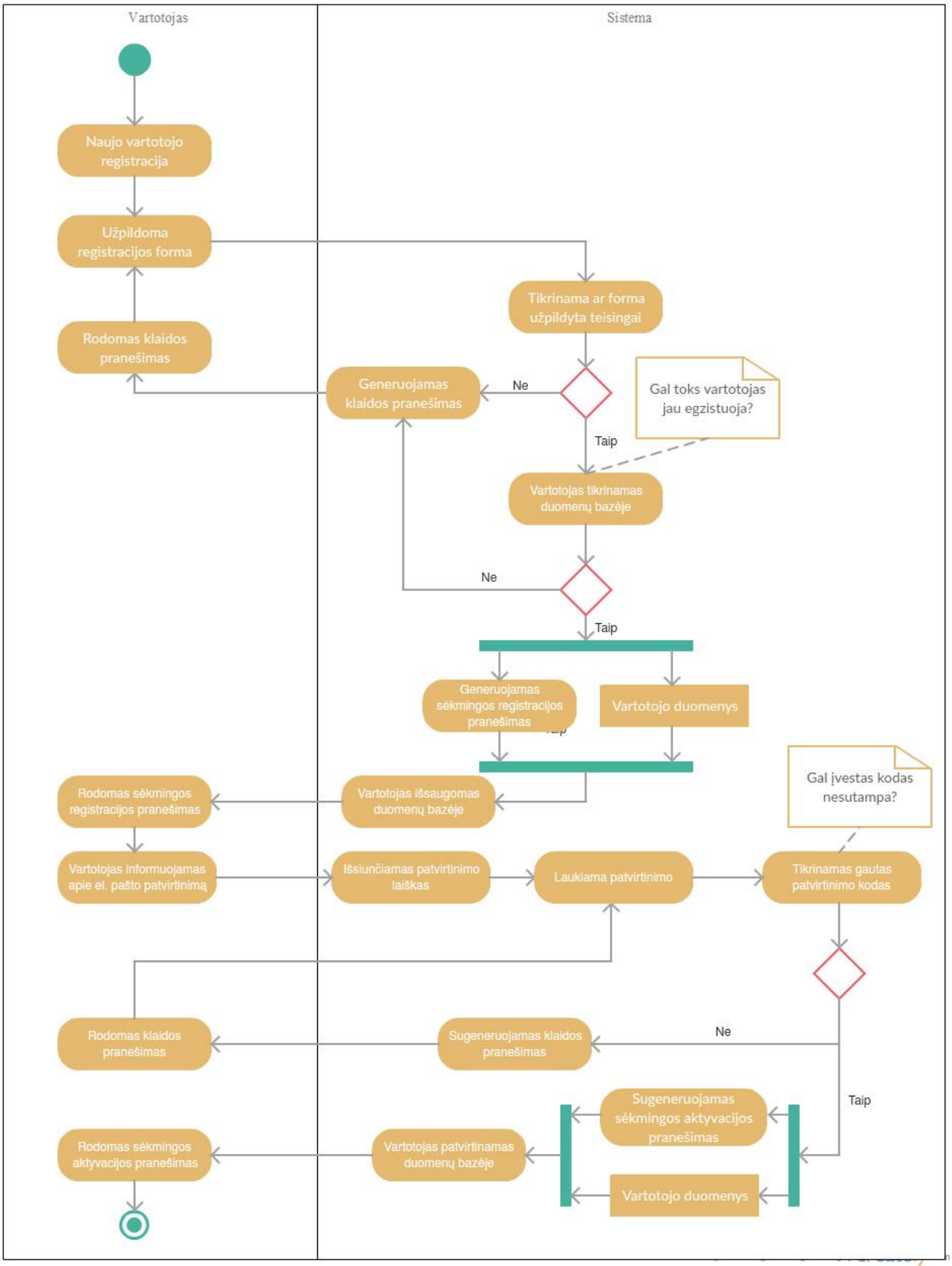
3.4. Pagrindinės sistemos funkcijos

Šioje dalyje aprašomos pagrindinės realizuotos sistemos funkcijos ir jų reprezentacijos UML diagramomis.

3.4.1. Registracija ir prisijungimas

Registracija sistemoje sudaryta iš dviejų dalių: formos užpildymo ir el. pašto paskyros patvirtinimo. Formos laukų tikrinimas vyksta **RegisterComponent** komponente su įprastais ir parašytais pritaikytais tikrintojais (angl. *validators*) slaptažodžio ir el. pašto laukams. Formoje įvestų duomenų neleidžiama išsaugoti tol, kol nėra tinkamai užpildomi visi privalomi laukai. Sėkmingai užpildžius formą, iškviečiamas **AuthService** klasės **signup()** metodas, kuris išsiunčia *POST* užklausą į minėtą *web API* ir, jei neįvyksta klaida, naudotojas išsaugomas duomenų bazėje ir apie tai informuojamas integruotu *toast* pranešimu. Naudotojui užsiregistravus, jam išsiunčiamas el. laiškas su nuoroda patvirtinti savo paskyrą. Tinkamai atidarius nuorodą, paskyra yra aktyvuojama ir naudotojas gauna prieigą prie jam priklausomo sistemos funkcionalumo.

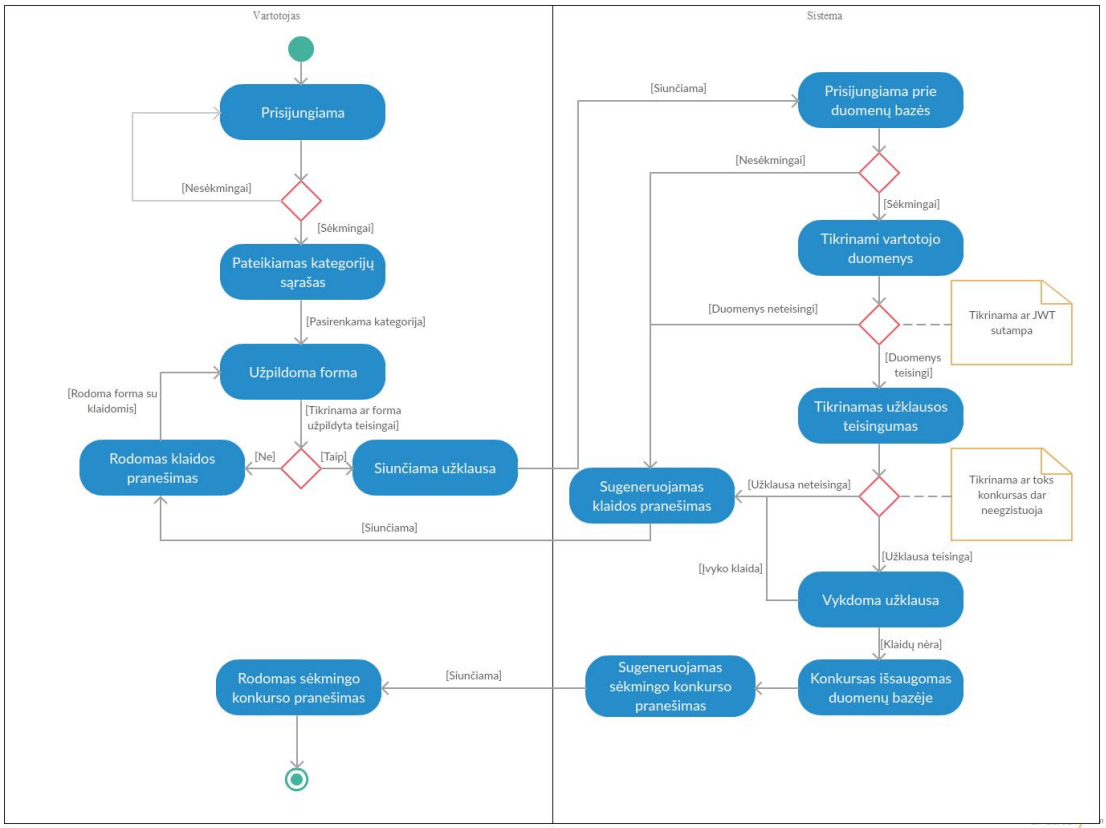
Naudotojui prisijungiant, jis turi įvesti savo slapyvardį ir slaptažodį. Jei naudotojui nepavyksta prisijungti daugiau kartų nei nustatyta sistemos konfigūracijoje, naudotojas saugumo sumetimais yra laikinai užblokuojamas. Naudotojas gali būti atblokuojamas susisiekus su administratoriumi. Tokiu atveju administratorius iš valdymo gali pakeisti naudotojo būseną į aktyvią iškviesdamas **AuthService** klasės **changeUserStatus()** metodą. Registracijos procesą aiškiau parodo 9 pav.



9 pav. Naudotojo registracijos diagrama

3.4.2. Dizaino konkurso paskelbimas

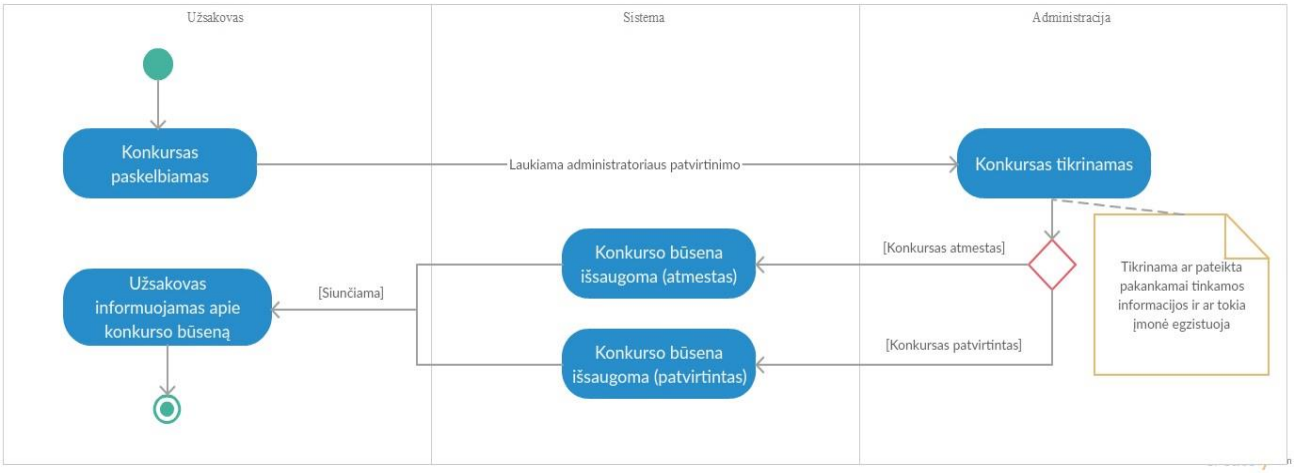
Naują dizaino konkursą gali paskelbti tik patvirtinti užsakovai. Šiam sėkmingai prisijungus ir pasirinkus dizaino kategoriją, jam išmetama tos kategorijos forma. Tinkamai užpildžius formą, iškviečiamas **ContestsService** klasės **addContest()** metodas, kuris išsiunčia *POST* užklausą į Node.js *web API*. Prieš pridedant konkursą į duomenų bazę, yra atliekami keli duomenų ir naudotojo, siunčiančio šią užklausą, patikrinimai. Kadangi konkursą paskelbti gali tik patvirtinti užsakovai, norint apsisaugoti nuo netinkamų užklausų įdiegtas apsaugos mechanizmas, žinomas kaip **JWT** (*JSON Web Token)*. Tai toks prieigos raktas, kuris priskiriamas naudotojui prisijungus prie sistemos ir išsaugomas naršyklės *SessionStorage* talpykloje. Šis prieigos raktas pridedamas prie užklausos adreso ir atkoduojamas naudojant *JsonWebToken* modulio **decode()** metodą, patvirtindamas naudotojo tapatybę. Jei nėra klaidų, konkursas išsaugomas duomenų bazėje ir naudotojas apie tai informuojamas. Šis procesas detaliau pavaizduotas 10 pav.



10 pav. Dizaino konkurso paskelbimo diagrama

3.4.3. Konkurso patvirtinimas

Užsakovui paskelbus konkursą, jis turi būti patvirtintas administratoriaus, kad konkursą galėtų matyti kiti sistemos naudotojai. Šis procesas pavaizduotas 11 pav. Administratorius apie paskelbtą naują konkursą informuojamas el. paštu ir sistemos privačia žinute. Administratorius valdymo panelėje gali patvirtinti konkursą arba jį atmesti dėl netinkamai pateiktos informacijos. Programiškai tai įvykdoma **AdminContestDetailsComponent** klasėje iškvietus **validateContest()** metodą, kuris kreipiasi į *web API* ir pakeičia konkurso statusą į aktyvų ar atmestą, priklausomai nuo administratoriaus pasirinkimo.



11 pav. Konkurso patvirtinimo diagrama

3.4.4. Dizainų įkėlimas

Dizainus įkelti į aktyvų konkursą gali tik patvirtinti dizaineriai. Pasirinkti galima vieną arba kelis failus. Pasirinkus „įkelti“, iškviečiamas **makeFileRequest()** metodas, kurio segmentas parodytas 12 pav. Šioje funkcijoje aprašomi *formData* ir *XMLHttpRequest* objektai. *FormData* objektas leidžia į jį pridėti *key-value* poras, ir šis objektas pridedamas siunčiant *XMLHttpRequest* užklausą. Šiuo atveju naudojamas **append()** metodas, kuris priima pridedamų dizainų failų informaciją ir prideda ją prie *formData* objekto, o tada išsiunčiamas apdoroti į Node.js serverį. Šis serveris naudoja *multer* modulį, kuris padeda apdoroti ir išsaugoti failus pasirinktoje direktorijoje [5]. Prieš išsaugant failus, failai yra patikrinami ir ne paveikslėlių tipo failai yra atmetami.

```
makeFileRequest(url: string, files: Array<File>, fileType: string) {
    return new Promise((resolve, reject) => {
        var formData: any = new FormData();
        var xhr = new XMLHttpRequest();
        for(var i = 0; i < files.length; i++) {
            formData.append(fileType, files[i], files[i].name);
        }
        xhr.upload.addEventListener("progress", (evt) => this.calculateUploadProgress(evt), false);
        xhr.onreadystatechange = function () {
            if (xhr.readyState == 4) {
                if (xhr.status == 200) {
                    console.log(xhr.response);
                    resolve(JSON.parse(xhr.response));
                } else {
                    console.log(xhr.response);
                    reject(xhr.response);
                }
            }
        }
        xhr.onerror = function(e) {
            console.log('Klaida įkeliant failus');
            console.log(e);
        };
        xhr.open("POST", url, true);
        xhr.send(formData);
    });
}
```

12 pav. Dizainų įkelimo kodo segmentas

3.5. Testavimas

Testavimas buvo atliktas dviem metodais: statine kodo analize ir esminių funkcijų testavimu rankiniu būdu.

3.5.1. Statinė kodo analizė

Šiam projektui sukurti buvo naudojamas *Visual Studio Code* kodo redagavimo įrankis. Kad kodas būtų tvarkingas ir be klaidų, buvo naudojami keli papildiniai ir konfigūraciniai failai. Vienas iš šių papildinių – *TSLint*, kuris patikrina *TypeScript* kalbos kodo kokybę ir praneša apie šiuos netikslumus ar klaidas. Taip pat naudojamas panašaus pobūdžio papildinys – *JSHint*, kuris naudojamas patikrinti visiems *JavaScript* failams dėl jų kokybės, ypač Node.js failams. HTML kodo kokybę užtikrinama statiniu HTML kodo analizės įrankiu – *HTMLHint*. Galiausiai yra naudojami *TypeScript* konfigūraciniai failai, skirti *Visual Studio Code* redaktoriui, kurie praneša apie netinkamą *TypeScript* kodo sintaksę.

3.5.2. Esminių funkcijų testavimas

Testavimo metu siekiama parodyti, kad programa daro tai, ko iš jos tikimasi, bei surasti klaidas anksčiau, negu programą pradeda naudoti galutiniai jos naudotojai. Dėl šių priežasčių, esminių sistemos funkcijų testavimas buvo nuolat atliekamas rankiniu būdu. Taip pat sistemą daviau išbandyti keliems pažįstamiems ir pagal atsiliepimus šią sistemą tobulinau.

Išvados ir rekomendacijos

Bakalaurinio darbo metu buvo išanalizuotos panašios sistemos, išsikelti reikalavimai kuriamai sistemai, suprojektuotas sistemos modelis ir programiškai įgyvendintos esminės sistemos funkcijos. Galutiniame darbo variante buvo sukurta sistema, leidžianti užsakovams paskelbti dizaino konkursus, o laisvai samdomiems dizaineriams – juose dalyvauti, ir būti įvertintiems. Taip pat sistemoje leidžiama bendrauti su kitais naudotojais, rašyti komentarus, peržiūrėti dizainerių statistikas ir redaguoti įkeltus dizainus.

Rekomendacijos

- Vienas iš galimų šios sistemos tobulinimo pasiūlymų – pridėti atskirą kategorijų puslapį, kuriame būtų detaliau aprašyta kiekviena kategorija, su kitų dizainų ir konkursų pavyzdžiais. Tai leistų užsakovams peržvelgti panašius konkursus ir taip palengvintų jų darbą skelbiant konkurso reikalavimus.
- Taip pat šią sistemą būtų galima praplėsti įdiegiant diskusijų forumą. Nors dabartinė sistema leidžia naudotojams siųsti privačias žinutes ir rašyti komentarus, atskira vieta diskusijoms praplėstų sistemos galimybes ir būtų patrauklesnė naudotojams.
- Kitas pasiūlymas būtų patobulinti esamą sistemos grafinę sąsają pritaikant modernaus dizaino principus ir animacijas. Tokiu būdu būtų galima į sistemą pritraukti daugiau dizainerių ir suteikti sistemai profesionalesnį įvaizdį.
- Galiausiai, sistemą būtų galima tobulinti įdiegus internetinės bankininkystės metodus, kurie leistų užsakovams tiesiogiai sumokėti dizaineriams už jų darbus.

Literatūra

1. Ari Lerner, Felipe Coury, Nate Murray, Carlos Taborda. Ng2-book 2. The Complete Book on AngularJS 2. Fullstack.io, 2016.
2. E. Gamma, R. Helm, R. Johnson and J. Vlissides. Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.
3. Bcrypt. Nuoroda: https://en.wikipedia.org/wiki/Bcrypt. Žiūrėta: 2017-01-02.
4. AngularJS 2 dokumentacija. Nuoroda: https://angular.io/docs/ts/latest/. Žiūrėta: 2017-01-02.
5. Multer. Nuoroda: https://github.com/expressjs/multer. Žiūrėta: 2017-01-02.