



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Gytis Lazauskas

**PROGRAMINIŲ PROJEKTŲ KOMANDOS FORMAVIMO IR VALDYMO
SISTEMA**

**SYSTEM FOR PROGRAMMING PROJECT TEAM BUILDING AND
MANAGEMENT**

Baigiamasis bakalauro darbas (projektas)

Programų inžinerijos studijų programa, valstybinis kodas 612I30003

Programų sistemų studijų kryptis

Vilnius, 2020

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU

Katedros vedėjas



(Parašas)

Darius Mažeika

(Vardas, pavardė)

2020-05-28

(Data)

Gytis Lazauskas

**PROGRAMINIŲ PROJEKTŲ KOMANDOS FORMAVIMO IR VALDYMO
SISTEMA**

**SYSTEM FOR PROGRAMMING PROJECT TEAM BUILDING AND
MANAGEMENT**

Baigiamasis bakalauro darbas (projektas)

Programų inžinerijos studijų programa, valstybinis kodas 612I30003

Programų sistemų studijų kryptis

Vadovas doc. dr. Simona Ramanauskaitė

(Pedag. vardas, vardas, pavardė)



(Parašas)

2020-05-23

(Data)

Vilnius, 2020

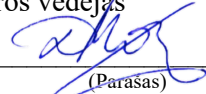
VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Programų sistemų studijų kryptis

Programų inžinerijos programa, valstybinis kodas 612I30003

TVIRTINU

Katedros vedėjas



(Parašas)

Darius Mažeika

(Vardas, pavardė)

2020-05-28

(Data)

**BAIGIAMOJO BAKALAURO DARBO
UŽDUOTIS**

2019-10-03 Nr. 1-5

Vilnius

Studentui (ei) GYČIUI LAZAUSKUI

(Vardas, pavardė)

Baigiamojo darbo tema: PROGRAMINIŲ PROJEKTŲ KOMANDOS FORMAVIMO IR VALDYMO SISTEMA

(angl. SYSTEM FOR PROGRAMMING PROJECT TEAM BUILDING AND MANAGEMENT)

patvirtinta 2019 m. lapkričio 15 d. dekanų potvarkiu Nr. 402fm

Baigiamojo darbo užbaigimo terminas 2020 m. birželio 3d.

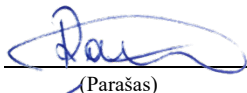
BAIGIAMOJO DARBO UŽDUOTIS:

Susipažinti su programinių projektų komandos subūrimui naudojamais metodais ir sistemomis. Parengti informacinės sistemos, kurioje būtų galima ieškoti bendraminčių ir komandos narių programinio projekto vystymui reikalavimų ir architektūros specifikacijas. Remiantis sudarytomis specifikacijomis, realizuoti sistemos prototipą, kuriame veiktų pagrindinės sistemai numatytos funkcijos. Atlikti sukurtos sistemos testavimą ir dokumentuoti jos veikimą.

Baigiamojo bakalauro darbo konsultantai:

(Pedag. vardas, vardas, pavardė)

Vadovė

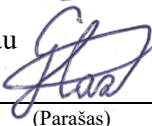


(Parašas)

doc. dr. Simona Ramanauskaitė

(Pedag. vardas, vardas, pavardė)

Užduotį gavau



(Parašas)

Gytis Lazauskas

(Vardas, pavardė)

2019-10-03

(Data)

Vilniaus Gedimino technikos universitetas

Fundamentinių mokslų fakultetas

Informacinių sistemų katedra

ISBN ISSN

Egz. sk.

Data-....-....

Programų inžinerijos studijų programos baigiamasis bakalauro darbas

Pavadinimas **Programinių projektų komandos formavimo ir valdymo sistema**

Autorius **Gytis Lazauskas**

Vadovas doc. dr. **Simona Ramanauskaitė**

Kalba



lietuvių



užsienio

Anotacija

Baigiamajame bakalauro darbe analizuojamas darbo paieškos bei komandos būrimo sistemos. Kuriami sistemos reikalavimai, architektūra. Realizuojama ir testuojama programinių projektų komandos formavimo sistema.

Šio darbo analitinėje dalyje yra analizuojamos sistemos, kurios padeda darbuotojams susirasti naujus darbo projektus arba projektų vadovams susirasti darbuotojų į savo komandas. Pateikiami šių sistemų privalumai ir trūkumai.

Praktinėje dalyje kuriama projektų komandos formavimo sistema atsižvelgiant į išanalizuotų sistemų funkcionalumą. Taip pat kuriami sistemos reikalavimai, aprašoma architektūra. Realizuojama sistema, aprašomos naudotos technologijos bei kiti įrankiai. Atliekamas sistemos testavimas. Pateikiamos darbo išvados.

Darbą sudaro: įvadas, komandos subūrimo procesai, technologijų reikalingų sistemai įgyvendinti apžvalga, sistemos projektavimas, sistemos testavimas ir išvados.

Darbo apimtis – 48 puslapiai teksto, 16 paveikslėlių, 32 lentelės, 15 šaltinių.

Prasminiai žodžiai: informacinė sistema, projektai, komandos formavimas, darbo sistema, darbo paieška.

Vilniaus Gedimino Technical University

Fundamental sciences faculty

Information systems department

ISBN ISSN

Copies No.

Date-....-....

Software engineering study programme bachelor thesis.

Title: **System for Programming Project Team Building and Management**

Author **Gytis Lazauskas**

Academic supervisor doc. dr. **Simona Ramanauskaitė**

Kalba



Lithuanian



Foreign (English)

Annotation

Bachelor thesis consists of analysis of job search and team building systems. Also, it consists of developed system requirements and architecture, working and testable programming project team building system.

In the analysis part, the systems which help workers find new projects to work on, or project managers to find new people for their team is analysed. The systems advantages and disadvantages are provided.

In practical part the system for project team building is created taking functional requirements into account. Also, the tools and technologies used for creating a system is described. System testing is performed. Conclusions are presented.

Work consists of introduction, processes of team building, review of technologies needed for system realisation, system design, system testing and conclusion.

Work volume – 48 pages of text, 16 pictures, 32 tables, 15 references.

Keywords: information system, projects, team building, work system, job search.

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Gytis Lazauskas, 20163552

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Programų inžinerija, PRIf-16/2

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)

SĄŽININGUMO DEKLARACIJA

2020 m. birželio 02 d.

Patvirtinu, kad mano baigiamasis projektas tema „Programinių projektų komandos formavimo ir valdymo sistema“ patvirtintas 2019 m. lapkričio 15 d. dekanų potvarkiu Nr. 402fm, yra savarankiškai parašytas. Šiame projekte pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Prenkant ir įvertinant medžiagą bei rengiant baigiamąjį projektą, mane konsultavo mokslininkai ir specialistai: docentas daktaras Simona Ramanauskaitė. Mano projekto vadovas docentas daktaras Simona Ramanauskaitė.

Kitų asmenų indėlio į parengtą baigiamąjį projektą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).



(Parašas)

Gytis Lazauskas
(Vardas ir pavardė)

SUTIKIMAS DĖL ASMENS DUOMENŲ NAUDOJIMO

2020-05-30

(Data)

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

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

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

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

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

Duomenų subjektas turi teisę bet kuriuo metu atšaukti savo sutikimą.

____Gytis Lazauskas

[Duomenų subjekto vardas, pavardė, parašas]

Duomenų valdytojo rekvizitai:
Vilniaus Gedimino technikos universitetas
Juridinio asmens kodas: 111950243
Adresas: Saulėtekio al. 11, LT-10223 Vilnius
Tel. (8 5) 274 5030
Faks. (8 5) 270 0112
El. paštas: vgtu@vgtu.lt

PVM mokėtojo kodas: LT119502413

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

Turinys

1. Įvadas	7
1.1. Tyrimo objektas	7
1.2. Darbo tikslas ir uždaviniai	7
1.3. Tyrimo naujumas bei praktinė jo vertė	7
2. Komandos subūrimas	8
2.1. Iššūkiai formuojant komandą	8
2.2. Komandos rolės	8
3. Analogiškų sistemų apžvalga	10
3.1. „LinkedIn“ apžvalga	10
3.2. „CVbankas“ apžvalga	12
3.3. „Fiverr“ apžvalga	13
3.4. Analogų analizės išvados	14
4. Internetinių sistemų kūrimo technologijų analizė	15
4.1. Front-end technologijų analizė	15
4.1.1. „Angular“	15
4.1.2. „React“	16
4.2. Back-end technologijų analize	17
4.2.1. „Spring Boot“ karkaso analizė	17
4.2.2. „Express“ karkaso analizė	18
4.3. Duomenų bazių technologijų analizė	18
4.3.1. „MySQL“ apžvalga	18
4.3.2. „MongoDB“ apžvalga	19
4.4. Išvados	19
5. Reikalavimų specifikacija	20
5.1. Naudotojų interfeisų reikalavimai	20
5.1.2. Naudotojo užduočių palyginimas	20
5.1.3. Užduočių formavimo kalbos reikalavimai	21
5.1.4. Interfeiso darnos ir standartizavimo reikalavimai	21
5.1.5. Pranešimų formulavimo reikalavimai	21
5.2. Funkciniai sistemos reikalavimai	21
5.2.1. Dalykiniai sistemos reikalavimai	21
5.2.2. Pagalbinės sistemos funkcijos	24
5.3. Nefunkciniai sistemos reikalavimai	24
5.3.1. Vidiniai Interfeiso reikalavimai	24
5.3.2. Vaizdavimo tikslumo reikalavimai	25
5.3.3. Diegimo reikalavimai	25
5.3.4. Kiti nefunkciniai reikalavimai	26
6. Informacinės sistemos projektavimas	27
6.1. Pasirinktos sistemų kūrimo technologijos	27
6.2. Dalykinės srities duomenų modeliavimas	27
6.3. Sistemos funkcijų projektavimas	28
6.3.1. Funkcijų pasiskirstymas pagal naudotojų vaidmenis ir vykdymo eiga	28
6.4. Atskirų funkcijų projektavimas	30
6.4.1. Sukurti patirtį	30
6.4.2. Sukurti projektą ir pridėti rolę	31
6.4.3. Kandidatuoti į rolę	32
6.4.4. Naudotojo profilio atvaizdavimas	34

7. Sistemos realizacija.....	36
7.1. Sistemos dizainas	36
7.2. Sistemos funkcionalumo programavimas	36
7.3. Įvedamų duomenų patikra.....	37
7.3.1. Prisijungimas ir registracija	37
7.3.2. Datų validavimas.	37
7.3.3. Naujos rolės įvedimo reikalavimai	38
8. Informacinės sistemos testavimas.....	40
8.1. Testavimo scenarijai	40
8.2. Testavimo atvejai	41
8.3. Testavimo rezultatai.....	45
9. Išvados	46
10. Literatūra.....	47

Lentelių sąrašas

5.1 lentelė. Naudotojo „Projekto vadovas“ funkcijų aprašymas	20
5.2 lentelė. Naudotojo „Darbuotojas“ funkcijų aprašymas	20
5.3 lentelė. Užduočių pasiskirstymas pagal vaidmenis	20
5.4 lentelė. UFK reikalavimai.....	21
5.5 lentelė. IDS reikalavimai	21
5.6 lentelė. PF reikalavimai	21
5.7 lentelė. DS reikalavimai	22
5.8 lentelė. Pagalbinės sistemos funkcijos.....	24
5.9 lentelė. Operacinės sistemos naudojimo reikalavimai.....	24
5.10 lentelė. Sąveikos su duomenų bazėmis reikalavimai.....	24
5.11 lentelė. Dokumentų mainų reikalavimai.....	24
5.12 lentelė. Darbo kompiuterių tinkluose reikalavimai	24
5.13 lentelė. Sąveikos su kitomis programomis reikalavimai	24
5.14 lentelė. Programavimo aplinkos reikalavimai	25
5.15 lentelė. Tikslumo reikalavimai	25
5.16 lentelė. Skaičiavimo tikslumo reikalavimai.....	25
5.17 lentelė. Patikimumo reikalavimai	25
5.18 lentelė. Robastiškumo reikalavimai.....	25
5.19 lentelė. Našumo reikalavimai	25
5.20 lentelė. Ruošinio reikalavimai	25
5.21 lentelė. Instaliavimo reikalavimai.....	26
5.22 lentelė. Pradinio duomenų bazių kaupimo reikalavimai	26
5.23 lentelė. Sistemos įsisavinamumo reikalavimai	26
5.24 lentelė. Aptarnavimo ir priežiūros reikalavimai	26
5.25 lentelė. Apsaugos reikalavimai.....	26
5.26 lentelė. Juridiniai reikalavimai	26
6.1 lentelė. Patirties sukūrimas	31
6.2 lentelė. Projekto ir rolės sukūrimas	32
6.3 lentelė. Kandidatavimas į projekto rolę	33
6.4 lentelė. Atsiliepimo apie projektą palikimas	35
8.1 lentelė. Testavimo scenarijai	40
8.2 lentelė. Testavimo atvejai	42

Paveikslėlių sąrašas

3.1 pav. Įgūdžio įvertinimas (Liwanag 2019).....	11
3.2 pav. „CV Bankas” darbo skelbimo kainoraštis	12
4.1 pav. „Angular“ formos (Groner 2017)	16
4.2 pav. Duomenų įvedimas į skirtingas duomenų bazes.....	19
6.1 pav. Sistemos duomenų modelio diagrama	27
6.2 pav. Rolės „Darbuotojas“ panaudos diagrama	29
6.3 pav. Rolės „Projekto vadovas“ panaudos diagrama	30
6.4 pav. Patirties sukūrimo sekų diagrama	31
6.5 pav. Projekto kūrimo sekų diagrama	32
6.6 pav. Kandidatavimo sekų diagrama.....	34
6.7 pav. Kandidatavimo būsenų diagrama.....	34
6.8 pav. Profilio atvaizdavimo sekų diagrama	35
7.1 pav. Sistemos logotipas	36
7.2 pav. Registracijos ir prisijungimo puslapis	37
7.3 pav. Datos pasirinkimo formatas	38
7.4 pav. Sėkmingai sukurtas projektas	39

Santrumpos

E2E – End-to-End – testavimo metodologija

DB – duomenų bazė

SQL – Structured Query Language – struktūrinių užklausų kalba

NoSQL – užklausų kalba kuri nėra struktūrizuota ir padalinta į lenteles

Crowdturfing – pasamdymas didelio kiekio žmonių atlikti smulkias veiklas socialiniuose puslapiuose

Front-end – kompiuterinės sistemos dalis atsakinga už naudotojo sąveiką

Back-end – kompiuterinės sistemos dalis atsakinga už duomenų manipuliavimą

HTML injekcija – HTML sąsajoje įvedama JavaScript kodo dalis

Smart-dumb – protingo tipo komponentai atlieka funkcijas, kvailo tipo tik atvaizduoja informaciją

Regex – simbolių seka, apibrėžianti sekos modelį

DOM – Dokumento objekto modelis

1. Įvadas

Sparčiai plečiantis informacinių technologijų sektoriui, vienas iš aktualiausių klausimų kylantis asmenims keičiantiems darbo pozicijas, yra darbdavio reputacija bei praeityje jo koordinuotų projektų sėkmė (Cable ir Turban 2003). IT sektorių darbuotojai, norintys pradėti dirbti mažose komandose dažnai susiduria su informacijos apie savo būsimus bendradarbius stoka. Naudojantis sistema, kurioje darbuotojai ir darbdaviai galėtų dalintis savo patirtimi bei atsiliepimais apie atliktus bendrus projektus, rizika pasirinkti darbo vietą nekvalifikuotų asmenų komandoje sumažėtų.

1.1. Tyrimo objektas

Projektų komandos formavimo sistema ir jos pritaikymas informacinių technologijų rinkoje.

1.2. Darbo tikslas ir uždaviniai

Tikslas – supaprastinti komandos subūrimą, skirtą informacinių sistemų projekto įgyvendinimui bei pagreitinti projekto ar darbuotojų paieškos procesus.

Uždaviniai:

1. Išanalizuoti ir palyginti esamas projektų komandos subūrimui ir darbo paieškoms naudojamas sistemas, nustatyti pagrindinius trūkumus.
2. Atsižvelgiant į sistemos analogus suformuoti sistemos funkcinius ir nefunkcinius reikalavimus.
3. Remiantis sistemos reikalavimais sukurti veikiančios sistemos prototipą, kuriame veiktų pagrindinės sistemos funkcijos.
4. Aprašyti sistemos testavimo scenarijus ir atvejus bei ištestuoti sistemą.

1.3. Tyrimo naujumas bei praktinė jo vertė

Naujumas – Nors internetinėje erdvėje yra daug sistemų siūlančių darbo paieškos funkcijas, tačiau sistema sukurta būtent mažo dydžio IT projektams, neturi daug populiarių analogų. Sistemos kūrimo metu į darbo paiešką technologijų srityje yra žiūrima kitu aspektu, atsižvelgiant į darbuotojų norą dirbti mažose komandose prie mažos ir vidutinės apimties projektų.

Praktinė vertė – darbe atlikta darbo paieškos sistemų analizė, tinkamiausių technologijų panaudojimas ir analizė kuriant informacinę sistemą pritaikytą komandos subūrimui, veikianti sistema, kuri gali būti naudojama komandos subūrimui bei atsiliepimų apie įvykdytus projektus rašymui.

2. Komandos subūrimas

2.1. Iššūkiai formuojant komandą

Komandos formavimas tai procesas, kurio metu yra surenkama grupė individų su skirtingomis patirtimis, darbo metodais ar išsilavinimais ir šios grupės pavertimas į efektyvų darbo vienetą (Wilemon, D. L. ir Thamhain, H. J. 1983). Norint sukurti sėkmingą komandą reikia suvokti pagrindinius komandos subūrimo metodus bei tikslą, kurio link judės komanda. Viena iš dažniausiai pasitaikančių konflikto priežasčių yra tikslaus rolių pasiskirstymo nebuvimas. Norint užtikrinti visų komandos narių efektyvų darbą komandoje, reikia priskirti atsakomybes kiekvienam komandos nariui, taip užtikrinant, jog visi procesai bus kieno nors prižiūrimi (Gander, F et al. 2018). Taip pat viena iš priežasčių neefektyvaus komandos darbo yra mažas motyvacijos kiekis. Motyvacijos trūkumo priežastis gali būti neįdomus projektas, o dažniau – neaiškus komandos tikslas, neegzistuojanti projekto vizija. Norint užtikrinti, jog darbuotojai būtų motyvuoti ir noriai imtųsi naujų darbų reikia privesti juos jaustis reikalingu komandoje bei tiksliai apibrėžti jų įtaką projekte. Žinant projekto tikslus darbuotojas lauks ateities užduočių, tiksliai žinodamas kaip prisideda prie projekto įgyvendinimo. Norint užtikrinti, jog naujas darbuotojas komandoje tiks ir bus motyvuotas reikia žinoti prie kokių projektų jis anksčiau dirbo ir kokie darbai jam yra įdomiausi. Dėl šios priežasties sistema, kuri pateikia visų potencialių darbuotojų patirtis ir atliktus projektus yra geras būdas projekto vadovui surasti žmogų kuris bus motyvuotas siekiant projekto įgyvendinimo.

2.2. Komandos rolės

Prieš kuriant sistemą skirtą komandos formavimui reikia tiksliai žinoti būtent kokie žmonės naudosis šia sistema ir kokių darbuotojų ar darbdavių jie ieškos. Dirbant su informacinėmis sistemomis pagrindinės rolės kartojasi beveik kiekvienoje komandoje, tačiau yra ir rečiau pasitaikančių vaidmenų, kuriuos komandos vadovas sukuria pagal savo nuožiūrą. Pagrindinės komandos rolės yra:

1. Projekto vadovas – tai žmogus patvirtinantis projekto planus ir strategijas. Atsakingas už darbuotojų įvedimą į komandą bei darbo pokalbius. Prioretizuoja projektus, stebi ar projektas vykdomas laiku ir ar telpama į biudžeto rėžius.
2. Projekto savininkas – ši rolė gali būti patikima ir projektų vadovui, tačiau plečiantis projektui šiuos du vaidmenis suvaldyti gali būti per sunku. Projekto savininkas ruošia planus, kurie tik vėliau būna įtraukti į sprintą, taip pat dėlioja abstraktesnį planą ir viziją projekto kūrimui ir palaikymui.
3. Verslo analitikas – rečiau pasitaikantis vaidmuo mažuose komandose, tačiau itin svarbi rolė projekto palaikymo stadijoje. Šis asmuo yra atsakingas už priimtinumą testus bei

reikalavimų kūrimą. Turint žmogų atsakingą už aiškių reikalavimų sukūrimą, programuotojams galima greičiau pradėti darbus.

4. Programuotojas – asmuo, kuris pagal sukurtus reikalavimus kuria ir vysto tam tikrą produktą programinės įrangos dalį. Šis žmogus yra reikalingas kiekviename projekte.
5. Testuotojas – asmuo komandoje, kuris atlieka automatinį arba rankinį testavimą. Ši rolė padeda sumažinti klaidų ir problemų skaičių produkcijos aplinkoje, taip pat padeda užtikrinti, jog atsiradus pakeitimams anksčiau veikusi sistemos dalis nėra sugadinta.

Šios rolės yra tik vienos iš pagrindinių, tačiau dažnu atveju komandose dirba ir duomenų bazių administratorius bei UX žmogus. Taigi, prieš pradedant kurti komandos formavimo sistemą reikia atsižvelgti, jog komandoje gali būti daug rolių ir sistemos naudotojai privalo turėti galimybę lengvai aprašyti esamas bei ieškomas pozicijas komandoje.

3. Analogiškų sistemų apžvalga

Norint surinkti komandą iš daug skirtingų rolių nėra taip paprasta. Būtent dėl šios priežasties egzistuoja internetinės sistemos padedančios surasit būtent tam tikros rolės darbuotojus, turinčių reikalaujama darbui patirtį. Analogiškų sistemų apžvalgos tikslas yra surinkti informacijos apie sistemas teikiančias darbo ar darbuotojų paieškos funkcionalumą bei išanalizuoti problemas kylančias naudojantis šiomis sistemomis organizuojant mažo arba vidutinio dydžio projektus.

Programinio projekto komandos susideda iš dviejų pagrindinių vaidmenų: programinės įrangos inžinierių ir vadovų. Vadovai ieško darbuotojų, kurie galėtų įgyvendinti sistemos reikalavimus ir sukurti veikiantį produktą, o inžinieriai ieško projektų, kuriuose galėtų dirbti ir tobulėti. Darbuotojų ir projektų paieška dažniausiai atliekama tam skirtuose internetiniuose puslapiuose, kuriuose prisiregistravus vartotojui yra siūlomi darbo skelbimai arba žmonės galintys tikti į ieškomą darbo poziciją. Taigi nors paieškos principas yra paprastas, tačiau daugelis analoginių sistemų jį įgyvendina skirtingai, todėl šių sistemų apžvalga yra labai svarbi.

3.1. „LinkedIn“ apžvalga

Ši sistema yra itin svarbi dėl savo išskirtinumo ir populiarumo, todėl jos apžvalga suteikia daug informacijos apie vartotojų įpročius naudojant darbo paieškos sistemas. „LinkedIn“ turi daugiau nei 575 milijonų vartotojų ir iš jų virš 260 milijonų yra aktyvūs bent kartą per mėnesį. Sistema išsiskiria iš kitų darbo paieškos sistemų dėl savo tipo – ji veikia kaip socialinis tinklapis. Šis veikimo principas pritraukia daug naujų vartotojų, aktyviai dalyvaujančių diskusijose ir platinančių savo asmeninę informaciją. Žmonės yra linkę dalintis savo informaciją socialiniuose tinkluose, nes tai jiems kelia džiaugsmą (Lin ir Lu 2011). Teoriškai, didžiulis informacijos kiekis turėtų padėti lengviau surasti tinkamiausią darbuotoją, tačiau tai turi įtakos informacijos reikšmingumui. Viena iš populiariausių „LinkedIn“ funkcijų yra kitų vartotojų rekomendavimas bei įgūdžių vertinimas (Caers, Ralf ir Castelyns 2011). Atsidarius profilį galima lengvai pridėti savo nuomonę apie tą žmogų kaip apie darbuotoją arba tiesiog įvertinti vieną iš jo gebėjimų (2.1 pav.).

The screenshot shows the LinkedIn 'Skills & Endorsements' section. A modal window titled 'Thanks for endorsing Jon' is open, asking for an endorsement for the skill 'Blogging'. The form includes a question 'How good is Jon at Blogging?' with three radio button options: 'Good', 'Very good', and 'Highly skilled' (which is selected). Below this is another question 'How do you know about Jon's Blogging skills?' with a dropdown menu showing 'Worked together directly on the same team c'. A note states: 'Your response is anonymous and will not be directly shared with Jon or other LinkedIn members. Learn more'. A blue 'Submit' button is at the bottom of the modal. In the background, the 'Blogging' skill card shows 39 endorsements, with three visible: Hannah Craig, Lizbeth (LB) Hernandez, and Amanda Ostrove.

3.1 pav. Įgūdžio įvertinimas (Liwana 2019)

Nors darbo paieškos sistema kaip socialinis tinklas pritraukia daug vartotojų, tačiau tai lemia ir didelę iškreiptos informacijos kiekį. „LinkedIn“ naudotojai skiria ir priima rekomendacijas neįsigilinę į žmonių įgūdžius (Rapanta ir Cantoni 2017). Rekomendavimas kitų žmonių dėl jų savybių tampa nebeaktualus, nes juo pradėta piktnaudžiauti. Sistemos vartotojai per daug negalvodami rekomenduoja savo pažįstamus ir draugus, apibūdindami jų įgūdžius kaip labai gerai išugdytus. Šis reiškinys verčia kitus sistemos naudotojus nepasikliauti šia funkcija, todėl ji tampa nebeaktualiai ir retai į ją yra atsižvelgiama ieškant komandos narių. Nors socialinio tinklo pobūdžio darbo paieškos sistema yra labai populiari, tačiau tai ateina su tam tikra kaina. Problema surasti patikimus komandos narius lieka neišspręsta, o rekomendacijų rašymo funkcija verčia susimastyti ar potencialus darbuotojas išties turi tiek daug įgūdžių, o gal tiesiog pažįstamų.

Kitas labai svarbus šios sistemos funkcionalumas yra paieška. Ši funkcija yra ypač svarbi turint daug aktyvių vartotojų ir ji vienintelė leidžia surasti aktualius darbo skelbimus ar darbuotojus greitai ir paprastai. Paieškos rezultatai kiekvienam vartotojui gali būti skirtingi ir jie priklauso nuo daugelio faktorių(). Turint per daug raktažodžių savo asmeninėje anketoje atsiranda galimybė niekad neatsidurti tarp paieškos rezultatų. Taip atsitinka todėl nes paieškos algoritmas tokio tipo anketas traktuoja kaip šlamštą. Tačiau tas pats gali nutikti ir turint per mažai informacijos apie save, todėl sistemos naudotojams visada turėtų būti aktualu atnaujinti informaciją apie save. Taip pat „LinkedIn“ paieška palaiko užklausų rašymą paieškos lauke (Bradbury 2011). Tai leidžia vartotojams ieškantiems labai specifinio kandidato susiaurinti jų paiešką ir leisti išsirinkti iš mažesnio kiekio, bet labiau tinkančio darbui kandidatų.

Taigi, „LinkedIn“ darbo paieškai tarp šimtų milijono žmonių gali pasiūlyti daug paieškos funkcijų, palengvinančių naudojimąsi sistema. Deja, funkcija turėjusi padėti sistemos naudotojams atrasti naujus kolegas pagal rekomendacijas yra visiškai neefektyvi dėl pačių vartotojų neteisingo naudojimosi sistema.

3.2. „CVbankas“ apžvalga

Internetinis darbo skelbimų puslapis Lietuvos rinkai „CVbankas“ egzistuoja jau daugiau nei 15 metų ir yra vienas populiariausių darbo paieškos puslapių šalyje. Priešingai nei „LinkedIn“ puslapis neturi rekomendacijų apie darbdavį funkcijos ir iš išvaizdos primena skelbimų portalą. Pati sistema dalinama į dvi dalis: darbo paieškos ir darbuotojų paieškos. Darbo paieškai reikalaujama susikurti savo anketą ir įkelti CV, darbuotojų paieškos dalis turi atskirą registracijos formą anketai ir pačio skelbimo sukūrimui. Darbo skelbimo sukūrimas kainuoja ir už papildomą sumą gali būti iškeltas aukščiau nei kiti skelbimai sąraše (2.2 pav.).

1 SKELBIMAS	1 VIP SKELBIMAS	3 SKELBIMAI
Jei ieškote 1 darbuotojo	Jei ieškote 1 darbuotojo ir norite rinktis iš daugiau kandidatų	Jei ieškote 2-3 darbuotojų
129 Eur 165-Eur -27%	238 Eur 274-Eur -13%	249 Eur 300-Eur -20%
1 darbo skelbimas	1 darbo skelbimas + VIP išskėlimas	3 darbo skelbimai
Paslauga galios 30 dienų	Paslauga galios 30 dienų	Paslauga galios 30 dienų
Matysite atsiųstų CV kontaktus	Matysite atsiųstų CV kontaktus	Matysite atsiųstų CV kontaktus
Papildomi pasirinkimai	Skelbimo išskėlimas	
CV duomenų bazė	CV duomenų bazė	
50 CV kontaktų 89 € daugiau	- peržiūros neribojamos (paslėptais kontaktais); - paslauga leidžia atverti 50 CV kontaktų; - kontaktai atveriami paspaudus Rodyti kontaktus.	

3.2 pav. „CV Bankas“ darbo skelbimo kainoraštis

Viena iš problemų naudojantis šią sistemą yra rekomendacijų apie darbuotojus ir darbdavius nebuvimas. Norint ieškoti asmens komandai tarp daugelio gyvenimo aprašymų reikia pasikliauti, jog sistemos naudotojai nemeluoja savo aprašymuose. Taip pat ieškant darbo mažoje įmonėje informacijos kiekis apie ją priklauso nuo skelbimą įdėjusio asmens, todėl norint surasti papildomos informacijos, jos reikia ieškoti kitose internetiniuose puslapiuose.

Paieškos sistema atlieka savo funkciją, tačiau rezultatai priklauso nuo to ar asmuo įkėlęs skelbimą sumokėjo papildomai už paslaugą jo pasiūlymą iškelti aukščiau. Toks sistemos veikimo principas atstumia

naujus vartotojus norinčius susirasti pirmą darbą neinvestuojant pinigų į jo ieškojimą. Dar didesnė problema iškyla ieškant būtent žmonių IT komandai. Ieškant darbuotojų negalima pasirinkti su kokio tipo technologijomis jie turėtų mokėti dirbi, galimas tik vienintelis pasirinkimas iš sąrašo– Informacinės technologijos. Bandant ieškoti technologijos paieškos lauke, rezultatai neatitinka įvestos reikšmės ir yra siūloma rinktis iš tų pačių, skelbimų sąrašui priklausančių, asmenų ar įmonių.

Taigi, nors „CVbankas“ ir yra vienas iš populiariausių darbo skelbimų puslapių Lietuvos rinkai, naudojimasis juo stengiantis sudaryti mažo arba vidutinio dydžio komandą IT projektui yra neefektyvus. Paieškos sistema neturi funkcijų susiaurinti paieškos rezultatų, o rekomendacijų nebuvimas prailgina paieškos procesą pridėdamas papildomą žingsnį informacijos ieškojimui kituose internetiniuose puslapiuose.

3.3. „Fiverr“ apžvalga

Šios sistemos biznio modelis yra visiškai kitoks nei prieš tai apžvelgtų sistemų. Į sistemą registruojasi kūrėjai ir už iš anksto numatytą sumą siūlo savo paslaugas, tokias kaip: logotipų nupiešimas, internetinio puslapio dizaino sukūrimas ir kt. Šio tinklapio veikimo principas panašesnis į turgaus, kuriame apsilankęs projekto vadovas gali išsirinkti skirtingas projekto dalis iš skirtingų asmenų ir visas jas sujungti į vieną. Toks pardavimo būdas dažnai pritraukia žmones su fiksuotu ir mažu biudžetu, ieškančių kaip būtų galima patobulinti kurią nors savo projekto dalį.

Vienas iš didžiausių šios svetainės privalumu yra reitingavimo funkciją. Užsakovui gavus atliktą darbą jis gali įvertinti darbuotoją skalėje nuo 1 iki 5, taip leisdamas kitiems užsakovams žinoti apie darbo kokybę. Ši funkcija veikia teisingai ir neklaidina žmonių, nes įvertinimą gali rašyti tik užsakovai sumokėję už paslaugas, taip apsisaugoma nuo neteisingo vertinimo.

Deja, ši sistema dėl savo lengvai užsakomų darbų funkcijos susilaukė ne tik asmenų siekiančių užsidirbti ir mažos apimties projektų, bet ir nelegalia veikla užsiimančių pardavėjų. Svetainėje yra perpildyta Crowdturfing pasiūlymų tokiems socialiniams tinklapiams kaip „Facebook“, „Instagram“ ir „Twitter“ (Lee, Webb, ir Ge 2014) Už kelis dolerius sistemoje galima pasisamdyti žmonių, kurie socialinio tinklo profilius pripildys sekėjų. „Fiverr“ administratoriai šių darbų neblokuoja, todėl tinklapis tampa apkrautas nelegalius darbus siūlančiais asmenimis.

Nors sistema yra patogi naudotis ir turi didelį srautą užsakovų bei darbuotojų, tačiau jos modelis nėra pritaikytas komandų kūrimui. Atskiros projekto dalis gali būti atliktos skirtingų darbuotojų siūlančių savo paslaugas „Fiverr“ sistemoje, tačiau nėra žmonių, kurie sujungtų visą sukurta funkcionalumą į vieną

veikiantį projektą. Dėl šios priežasties sistema nėra tinkama projektų kūrimui, tačiau puikiai tinka pakeisti kelis dalis jau egzistuojančioje informacinėje sistemoje.

3.4. Analogų analizės išvados

Atsižvelgus į analogų sistemų pagrindinius privalumus ir trūkumus galima teigti, jog svarbiausios funkcijos komandos subūrimo svetainei yra: paieška leidžianti tiksliai išfiltruoti rezultatus, rekomendacijų ir atsiliiepimų rašymas, kuris yra leidžiamas tik buvusiems ir esamiems bendradarbiams.

4. Internetinių sistemų kūrimo technologijų analizė

Kuriant IS turi būti naudojamos naujausios technologijos, taip užtikrinant lengvesnį sistemos palaikymą ateityje. IS kūrimas yra padalinta į tris dalis: Front-end, Back-end ir duomenų bazę.

4.1. Front-end technologijų analizė

Pagrindinis šiuolaikinių internetinių puslapių privalumas yra vieno puslapio aplikacijos palaikymas. Naudojant šią technologiją svetainė su vartotoju sąveikauja dinamiškai perrašydama dabartinį puslapį, o ne įkeldama iš serverio visus naujus puslapius, taip sukurdamą vartotojui geresnę naudojimosi sistema patirtį. Šį funkcionalumą palaiko „Angular“ karkasas bei JavaScript biblioteka „React“, todėl prieš pradėdant kurti sistemą reikalinga šių produktų analizė.

4.1.1. „Angular“

Šis produktas laikomas karkasu, nes savyje turi daug funkcionalumo ir jo kūrėjai yra aiškiai apibrėžę kaip sistema turėtų būti struktūrizuota. Dėl šių priežasčių programuotojams nereikia gaišti laiko gilinantis į kitas alternatyvias bibliotekas, kurias jie galėtų pritaikyti savo projekte. Taip pat kadangi pati struktūra yra nustatyta kūrėjų, kitų žmonių kodas yra lengviau suprantamas ir aiškesnis.

Komponentai yra pagrindinė karkaso dalis iš kurių yra kuriamos sistemos, šie blokai yra atpažįstami su `@Component` dekoratoriumi (Schmiedehausen 2018). Patys komponentai savyje turi visą logiką, kurią naudodami keičia ir atnaujiną modelis. Modeliai atsakingi tik už puslapio atvaizdavimą ir susideda iš HTML ir CSS klasių. Toks sistemos dalinimas padeda suprasti kuri sistemos dalis atsakinga už funkcionalumą, o kuri tik už atvaizdavimą stilių.

Vienos iš svarbiausių funkcijų „Angular“ karkase yra navigacija tarp skirtingų komponentų ir informacijos dalijimasis su jais. Naudojant šias funkcijas galima saugiai siųsti vartotojų duomenis iš vieno komponento į kitą ir to nereikia daryti per URL įvedimo lauką. Komponentų dalijimasis duomenimis gali būti *smart-dumb* tipo, taip užtikrinant, jog tik už duomenų atvaizdavimą ir siuntimą atsakingas komponentas negalės informacijos pakeisti ar iškreipti. Taip pat turint tokią komponentų struktūrą galima lengvai pernaudoti funkcionalumą daugelyje komponentų.

„Angular“ karkase taip pat egzistuoja ir formų kūrimas. Ši funkcija yra itin svarbi komandos subūrimo projektui, kuriame formos yra registracijoje, prisijungime ir projekto registracijoje. Turint tokį funkcionalumą projekto kūrimo laikas sutrumpėja ir sistema išlieka apsaugotas nuo HTML *injection* atakų. „Angular“ formų kūrimas leidžia naudoti *regex* išraišką ir paversti laukus privalomais ar priskirti jiems tipą (2.3 pav.).

The screenshot shows a web browser window with the address bar displaying 'https://ng-forms-fields-validation.firebaseio.com/validateSubmit'. The page title is 'Angular Forms Fields Validation'. The form contains the following fields:

- Name ***: A text input field with a red border and a red 'x' icon. Below it, the text 'Please inform your name' is displayed.
- Email ***: A text input field with a red border and a red 'x' icon. Below it, the text 'Please inform your email' is displayed.
- Address ***: A text input field with a red border and a red 'x' icon. Below it, the text 'Please inform your address' is displayed.
- Address 2**: A text input field with a white border.
- Zip Code ***: A text input field with a red border and a red 'x' icon. Below it, the text 'Please inform your zipCode' is displayed.
- City ***: A text input field with a red border and a red 'x' icon. Below it, the text 'Please inform your city' is displayed.
- State / Province / Region ***: A text input field with a red border and a red 'x' icon. Below it, the text 'Please inform your state' is displayed.
- Country ***: A text input field with a red border and a red 'x' icon. Below it, the text 'Please inform your country' is displayed.

At the bottom of the form, there are two buttons: 'Submit' (blue) and 'Reset' (gray).

4.1 pav. „Angular“ formos (Groner 2017)

Didžioji dalis šio karkaso yra sukurta „Google“ todėl ateityje galima tikėtis, jog šis projektas nebus apleistas ir jį pasieks kasmetiniai atnaujinimai. Taigi „Angular“ yra tinkamas naudoti komandos subūrimo projekte dėl tokių funkcijų kaip formų kūrimas, aiški struktūra, navigacija tarp komponentų bei dažni karkaso atnaujinimai.

4.1.2. „React“

Ši biblioteka buvo sukurta „Facebook“ įmonės, 2013 metais ir išleista kaip atviro kodo. Dabar ji yra naudojama daugelyje projektų, tokių kaip „WhatsApp“ ir „Instagram“. Biblioteka gali būti naudojama ne tik internetinių puslapių, bet ir mobiliųjų aplikacijų kūrime (Wohlgethan 2018).

Vieni iš pagrindinių priežasčių, kodėl „React“ yra populiariausia biblioteka naudojama internetinių svetainių kūrime yra greitis. Puslapiai naudojantys šią biblioteką yra greitesni nei kitomis bibliotekomis ar karkasais sukurti puslapiai. Taip yra dėl virtualios DOM kopijos. Naršant po puslapį DOM nėra atnaujinamas, tačiau virtuali jo kopija yra. Kai virtuali kopija pilnai atsinaujinama ji pakeičia originalų DOM. Dėl šio funkcionalumo puslapio atnaujinimas vyksta žymiai greičiau.

„React“ taip pat yra išdalinta į komponentus ir kiekvienas komponentas atsako už modelį arba modelio dalį. Todėl kaip ir „Angular“ karkasą naudojančiose sistemose, duomenys yra perduodami paprastai ir saugiai.

Mobiliųjų aplikacijų kūrimo srityje „React“ yra labiau pažengus nei „Angular“. Naudojant „React Native“ funkcionalumą, aplikacijas kurti yra paprasčiau ir dauguma komponentų gali būti pernaudota internetiniuose puslapiuose. Taip pat nereikia kurti atskirų sistemų Android ir iOS operacinėms sistemoms, tas pats projektas gali veikti ant abiejų tipų telefonų.

Taigi, nors „React“ biblioteka sukurtos sistemos veikia greičiau ir yra lengviau sukuriami jų mobiliųjų aplikacijų variantai, tačiau „Angular“ siūlo paprastesnę formą diegimą, palaikantį daugiau įvedimo apribojimų.

4.2. Back-end technologijų analize

Norint jog duomenys esantys duomenų bazėje būtų tinkamai gaunami, pakeisti, ištrinti ar pridėti reikalinga sistemos dalis atsakingą už duomenų manipuliaciją. Tam yra skirta Back-end dalis ir vieni jos populiariausių karkasų yra Java „Spring Boot“ ir NodeJs „Express“ (Goel 2020). Vienas nuo kito jie labai skiriasi, turi savų pliusų ir minusų ir yra naudojami skirtingose programavimo kalbose.

4.2.1. „Spring Boot“ karkaso analize

Karakasas buvo sukurtas 2003 metais ir yra plačiai naudojamas iki šiol. Jo populiarumo priežastis, buvo staigus Java kalbos populiarėjimas. Taip ilgai esant vienam iš populiariausių karkasų pasaulyje, „Spring Boot“ sukaupė didžiulį kiekį funkcionalumo, todėl pats karkasas yra didelis ir sunkiai išmokstamas.

Aplikacijos naudojančios „Spring Security“ karkasą yra saugesnės nei „Express“, tačiau jos užima žymiai daugiau kodo eilučių. Vienas iš šio karkaso saugumo privalumu yra lengvas sistemos saugumo praplėtimas, norint, jog projektas atitiktų iškeltus reikalavimus (Cervantes et al. 2019). Taip pat naudojant „Spring Security“ yra galimybė užkoduoti ir verifikuoti slaptažodžius.

Vienas iš didžiausių „Spring“ karkaso privalomų yra platus spektras papildomų sistemų iš to pačio tiekėjo. Turint projektą naudojanti šį karkasą, galima lengvai įdiegti debesų duomenų funkcionalumą, saugumą, mobiliųjų aplikacijų palaikymą ir daug kitokių priedų¹.

Nors šis karkasas turi daug ką pasiūlyti, tačiau pasileisti jį užtrunka ilgai, jis veikia lėčiau ir pritaikant jį komandos subūrimo projekte bus neišnaudotos visos jo galimybės.

¹ <https://spring.io/projects>

4.2.2. „Express“ karkaso analizė

Šiuo metu Node.js yra populiari technologija JavaScript kalboje. Tai platforma sukurta veikti greitai ir prisitaikyti prie kintančio projekto dydžio (Bangare et al. 2016). „Express“ Karkasas yra greitas ir užima mažai vietos, nes buvo sukurtas naudojant V8 JavaScript variklį.

Vienas iš pagrindinių šio karkaso pliusų yra paprastas paleidimas ir lengvas programavimas, nereikalaujantis daug kodo. Taip pat jis nereikalauja daug kompiuterio resursų, todėl gali būti lengvai keičiamas net ir tuo pačiu metu pasileidus „Front-end“ projekto dalį. Prijungimas prie duomenų bazės reikalauja mažai pastangų, tam reikalingos tik kelios kodo eilutės.

Dėl savo populiarumo „Express“ turi daug papildomų bibliotekų, kurios pagerina karkaso veiklą ir leidžia neperrašinėti kitų žmonių kodo, o tiesiog jį panaudoti.

Taigi, komandos sukūrimo projektui šis karkasas yra tinkamesnis dėl savo greitumo ir paprastumo. Projektui plečiantis „Express“ plėsis kartu ir tai įvykdyti nebus taip sunku, kaip naudojant didesnės apimties karkasus su daug skirtingo funkcionalumo.

4.3. Duomenų bazių technologijų analizė

Kiekvienam projektui reikalinga duomenų bazė. Jos yra dalinamos į du tipus: reliacinės ir nereliacinės duomenų bazės. Reliacinės duomenų bazės pagrindinis elementas yra reliacinė lentelė. Tokia lentelė turi iš anksti nustatytą stulpelių skaičių ir pavadinimą. Kiekviename stulpelyje yra naujas duomenų bazės įrašas. Nereliacinėse duomenų bazėse duomenys saugomi objektuose, raktų-reikšmių saugyklose.

Vienos iš populiariausių duomenų bazių yra „MySQL“ ir „MongoDB“ todėl jų analizė yra svarbi ir būtina prieš pradėdant kurti projektą.

4.3.1. „MySQL“ apžvalga

Ši duomenų bazė yra atviro kodo, pirmą kartą išleista 1995 metais ir iki šiol palaikoma įmonės „MySQL AB“ (Lambert 2017). Šia sistema naudojasi tokio didelės įmonės kaip: „WordPress“, „FaceBook“ ir „Twitter“. Ši duomenų bazė yra tokia populiari dėl saugumo ypatybių ir reto išsijungimo kiekio.

„MySQL“ naudoja struktūrizuotą užklausų kalbą SQL todėl duomenis galima keisti užklausomis „SELECT“, „UPDATE“, „INSERT“, „DELETE“. Visa informacija yra saugoma skirtinguose lentelėse, kuriuose, tačiau jos gali būti sujungtos. Prieš norint įkelti duomenis į duomenų bazę, jiems turi būti sukurta atskira lentelė su aprašytais tipais. Dėl šios priežasties kūrimo ir išleidimo procesai gali užtrukti, nes norint pakeisti lentelės laukus tam reikia laiko.

Nors reliacinės duomenų bazės schema yra lengviau suprantama ir gyvuoja jau daugelį metų, tačiau ilgas lentelių kūrimo procesas sudaro problemų norint sukurti projektą per trumpą laiko tarpą.

4.3.2. „MongoDB“ apžvalga

Viena geriausių šios duomenų bazės savybių yra apribojimų nebuvimas kuriant schemas dizainą. „MongoDB“ sukuria dokumentus neturinčius schemų, kuriuose galima talpinti bet kokio tipo informaciją (Lambert 2017). Tai gali sukelti duomenų neatitikimo problemų, tačiau jų galima išvengti panaudojus apribojimus kituose projekto vietose. Nors šioje duomenų bazėje ir nėra naudojama SQL, tačiau duomenų įvedimo procesas yra panašus (2.4 pav.).

MySQL	MongoDB
INSERT	
<pre>INSERT INTO account (`A/c number`, `first name`, `last name`) VALUES ('12345746352', 'Mark', 'Jacobs');</pre>	<pre>db.account.insert({ A/c number: "12345746352", first name: "Mark", last name: "Jacobs" });</pre>
UPDATE	
<pre>UPDATE account SET contact number = 9426227364 WHERE A/c number = '12345746352'</pre>	<pre>db.account.update({ A/c number: '12345746352' }, { \$set: {contact number: 9426227364 } });</pre>
DELETE	
<pre>DELETE FROM account WHERE e-mail address = 'jv1994@gmail.com';</pre>	<pre>db.account.remove({ "E-mail address": "jv1994@gmail.com" });</pre>

4.2 pav. Duomenų įvedimas į skirtingas duomenų bazines

Pagrindinis „MongoDB“ privalumas prieš SQL tipo DB yra greitis. Galimybė ieškoti informacijos didžiuliuose nestruktūrizuotose failuose leidžia „MongoDB“ veikti žymiai greičiau nei reliacinių tipų duomenų bazėms.

Taigi atsižvelgus į abiejų tipų duomenų bazines, „MongoDB“ yra geresnė dėl savo greičio ir greitesnio kūrimo ir išleidimo proceso.

4.4. Išvados

Atlikus vieną iš populiariausių programavimo technologijų ir karkasų analizę buvo pasirinktos tinkamiausios technologijos šio projekto įgyvendinimui bei palaikymui.

5. Reikalavimų specifikacija

5.1. Naudotojų interfeisų reikalavimai

Programinių projektų komandos formavimo ir valdymo sistemoje egzistuoja dvi naudotojų rolės:

- Projekto vadovas
- Darbuotojas

Žemiau esančiose lentelėse yra pateikiamos funkcijos, kurias gali atlikti naudotojas turintis projekto vadovo ar darbuotojo rolę.

5.1 lentelė. Naudotojo „Projekto vadovas“ funkcijų aprašymas

Rolė:	Projekto vadovas
Funkcijos:	Prisijungti prie sistemos, peržiūrėti darbuotojų profilius, atnaujinti ir pridėti asmeninius duomenis, pridėti įrašus apie įgytas patirtis ir atliktus darbus, peržiūrėti projektus, sukurti ir redaguoti projektus, sukurti ir redaguoti roles priklausančias sukurtiems projektams. Išrinkti darbuotoją iš kandidatų sąrašo tam tikrai rolei.

5.2 lentelė. Naudotojo „Darbuotojas“ funkcijų aprašymas

Rolė:	Darbuotojas
Funkcijos:	Prisijungti prie sistemos, peržiūrėti darbuotojų profilius, peržiūrėti projektus, atnaujinti ir pridėti asmeninius duomenis, pridėti įrašus apie įgytas patirtis ir atliktus darbus, pateikti paraišką į tam tikrą rolę, palikti atsiliepimą apie rolę prie kurios buvo arba yra dirbama.

5.1.2. Naudotojo užduočių palyginimas

Abi rolės turi skirtingas užduotis, kurias gali atlikti. Žemiau esančioje lentelėje yra nurodytos užduotys ir veiksmai – kurti (C), skaityti (R), keisti (U), ištrinti (D), kuriuos atlikti gali asmuo priskirtas vienai iš rolių.

5.3 lentelė. Užduočių pasiskirstymas pagal vaidmenis

Nr.	Užduoties pavadinimas	Projekto vadovas	Darbuotojas
1.	Prisijungimas prie sistemos	C	C
2.	Darbuotojų profilio peržiūra	R	R
3.	Asmeninių duomenų keitimas	UD	UD
4.	Asmeninių duomenų pridėjimas	C	C
5.	Projektų peržiūra	R	R
6.	Įrašų apie atliktus darbus kūrimas	C	C
7.	Projekto sukūrimas	C	-
8.	Projekto redagavimas	UD	-
9.	Rolės sukūrimas	C	-
10.	Rolės redagavimas	UD	-
11.	Aplikavimas į darbo vietą	-	C
12.	Atsiliepimo apie darbo vietą sukūrimas	-	C
13.	Atsiliepimo apie darbo vietą pakeitimas	-	UD

Nr.	Užduoties pavadinimas	Projekto vadovas	Darbuotojas
14.	Kandidato priskyrimas rolei	C	-
15.	Rolės peržiūra	R	R
16.	Projekto vadovo patirčių peržiūra	R	R
17.	Kandidatų sąrašo peržiūra	R	-

5.1.3. Užduočių formavimo kalbos reikalavimai

5.4 lentelė. UFK reikalavimai

Reikalavimo Nr.	Aprašymas
UFKR-01	Vartotojo sąsaja turi būti pateikta lietuvių kalba.

5.1.4. Interfeiso darnos ir standartizavimo reikalavimai

5.5 lentelė. IDS reikalavimai

Reikalavimo Nr.	Aprašymas
IDSR-01	Sistema turi turėti grafinę vartotojo sąsają.
IDSR-02	Vartotojo sąsaja turi būti pasiekama per naršyklę.
IDSR-03	Vartotojo sąsajos elementai turi būti atvaizduoti naudojant naršyklę per mobilųjį įrenginį.

5.1.5. Pranešimų formulavimo reikalavimai

5.6 lentelė. PF reikalavimai

Reikalavimo Nr.	Aprašymas
PFR-01	Sistema turi atvaizduoti informacinį pranešimą, kai funkciją kurią yra bandoma atlikti yra tuo metu negalima.
PFR-02	Sistema turi atvaizduoti klaidos pranešimą, kai vartotojas bando atlikti veiksmą kuris nėra galimas. Pranešimas atvaizduojamas kai bandoma jungtis su netinkamu slaptažodžiu arba susikurti anketą jau su egzistuojančiu elektroniniu paštu.
PFR-03	Sistema turi atvaizduoti įspėjamąjį pranešimą, kai yra suvedama neteisingas informacijos tipas. Pranešimas atvaizduojamas kai bandoma įvesti tik skaičius į vartotojo vardo lauką.

5.2. Funkciniai sistemos reikalavimai

5.2.1. Dalykiniai sistemos reikalavimai

5.7 lentelė. DS reikalavimai

Reikalavimo Nr.	Reikalavimas	Įeiga	Rezultatas	Skaičiavimai
DSR-01	Sistema turi leisti naudotojui užsiregistruoti.	Naudotojo vardas, elektroninis paštas, slaptažodis, rolė.	Užregistruotas naujas naudotojas.	Sukuriamas naudotojo ID, užkoduojamas slaptažodis, tikrinama ar nėra jau užregistruotas naudotojas su tokiu pačiu elektroniniu paštu.
DSR-02	Sistema turi leisti naudotojui prisijungti.	Naudotojo elektroninis paštas, slaptažodis.	Naudotojas prijungtas prie sistemos.	Tikrinama ar egzistuoja naudotojas, tikrinama ar slaptažodis teisingas.
DSR-03	Sistema turi leisti naudotojui atsijungti.	Nėra.	Naudotojas atjungtas iš sistemos.	Ištrinamas naudojo žetonas.
DSR-04	Sistema turi leisti naudotojui registruojantis pasirinkti rolę.	Rolė.	Užregistruotas naudotojas su pasirinkta role.	Nėra.
DSR-05	Sistema turi leisti naudotojui atnaujinti savo duomenis	Naudotojo vardas, elektroninis paštas.	Naudotojo duomenys atnaujinti.	Verifikuojami duomenys.
DSR-06	Sistema turi leisti naudotojui pridėti papildomą informaciją apie save	Karjeros aprašymas, gyvenama vieta, darbo sritis, gimimo data.	Pridėti papildomi naudotojo duomenys.	Verifikuojami duomenys.
DSR-07	Sistema turi leisti naudotojui pridėti darbo ar mokymosi patirtį.	Patirties pavadinimas, patirties pradžia, patirties pabaiga, aprašymas, rolė.	Pridėta nauja patirtis.	Verifikuojami duomenys.
DSR-08	Sistema turi leisti naudotojui atnaujinti arba ištrinti darbo ar mokslo patirtis.	Patirties pavadinimas, patirties pradžia, patirties pabaiga, aprašymas, rolė.	Atnaujinta naudotojo patirtis.	Verifikuojami duomenys.
DSR-09	Sistema turi leisti naudotojui su role „Projekto vadovas“ pridėti naują projektą.	Projekto pavadinimas, pradžios data, pabaigos data, aprašymas.	Pridėtas naujas projektas.	Sukuriamas projekto ID.

Reikalavimo Nr.	Reikalavimas	Įeiga	Rezultatas	Skaičiavimai
DSR-10	Sistema turi leisti naudotojui su role „Projekto vadovas“ redaguoti sukurtą projektą arba jį ištrinti.	Projekto pavadinimas, pradžios data, pabaigos data, aprašymas.	Projektas atnaujintas.	Verifikuojami duomenys.
DSR-11	Sistema turi leisti naudotojui su role „Projekto vadovas“ pridėti naujas roles prie egzistuojančio projekto.	Rolės pavadinimas, aprašymas, darbo tipas.	Pridėta rolė prie egzistuojančio projekto.	Sukuriamas rolės ID, rolė priskiriama projektui.
DSR-12	Sistema turi leisti naudotojui su role „Projekto vadovas“ redaguoti arba ištrinti sukurtas projekto roles.	Rolės pavadinimas, aprašymas, darbo tipas.	Atnaujinta rolė prie egzistuojančio projekto.	Verifikuojami duomenys.
DSR-13	Sistema turi leisti naudotojui peržiūrėti visus egzistuojančius projektus.	Nėra.	Atvaizduojami visi projektai.	Nėra.
DSR-14	Sistema turi leisti naudotojui su role „Darbuotojas“ kandidatuoti į projekto rolę.	Nėra.	Pridedama kandidatūra į rolę.	Nėra.
DSR-15	Sistema turi leisti naudotojui su role „Projekto vadovas“ peržiūrėti kandidatus į jo sukurtas projekto roles.	Nėra.	Atvaizduojamas kandidatų sąrašas.	Nėra.
DSR-16	Sistema turi leisti naudotojui su role „Projekto vadovas“ pasirinkti kandidatą į jo sukurtą projekto rolę.	Nėra.	Pridedamas darbuotojas prie rolės.	Nėra.
DSR-17	Sistema turi leisti naudotojui peržiūrėti visus naudotojų profilius.	Nėra.	Atvaizduojamas pasirinktas profilis.	Nėra.
DSR-18	Sistema turi leisti naudotojui ieškoti kitų naudotojų pagal jų vardus.	Naudotojo vardas.	Atvaizduojamas ieškomo naudotojo profilis.	Nėra.

Reikalavimo Nr.	Reikalavimas	Įeiga	Rezultatas	Skačiavimai
DSR-19	Sistema turi leisti naudotojui ieškoti projektų pagal projektų pavadinimus.	Projekto pavadinimas.	Atvaizduojamas ieškomas projektas.	Nėra.
DSR-20	Sistema turi leisti naudotojui ieškoti rolių pagal jų tipus.	Rolės pavadinimas.	Atvaizduojamas projektas su ieškoma role.	Nėra.

5.2.2. Pagalbinės sistemos funkcijos

5.8 lentelė. Pagalbinės sistemos funkcijos

Reikalavimo Nr.	Reikalavimas
PSF-01	Sistema turi rodyti klaidos pranešimą.
PSF-02	Sistema turi rodyti įvesties validacijos pranešimą

5.3. Nefunkciniai sistemos reikalavimai

5.3.1. Vidiniai Interfeiso reikalavimai

5.9 lentelė. Operacinės sistemos naudojimo reikalavimai

Reikalavimo Nr.	Reikalavimas
OSNR-01	Sistema privalo veikti visuose operacinėse sistemose, kuriose yra įdiegta internetinė naršyklė palaikanti HTML5.

5.10 lentelė. Sąveikos su duomenų bazėmis reikalavimai

Reikalavimo Nr.	Reikalavimas
SSDBR-01	Sistema turi naudotis MongoDB duomenų bazę.
SSDBR-02	Duomenų bazėje turi būti saugomi visi duomenys, kurie yra įvedami vartotojui naudojantis sistema.
SSDBR-03	DBVS turi užtikrinti kelių vartotojų sukūrimą su skirtingomis teisėmis.

5.11 lentelė. Dokumentų mainų reikalavimai

Reikalavimo Nr.	Reikalavimas
DMR-01	Sistemos dalys turi dalintis duomenimis JSON formatu.
DMR-02	Sistema statistinę informaciją turi pateikti CSV formatu.

5.12 lentelė. Darbo kompiuterių tinkluose reikalavimai

Reikalavimo Nr.	Reikalavimas
DKTR-01	Naudotojų įrenginiuose reikia turėti naršyklę palaikančią HTML5 ir interneto ryšį.
DKTR-02	Sistema turi palaikyti HTTP protokolą.

5.13 lentelė. Sąveikos su kitomis programomis reikalavimai

Reikalavimo Nr.	Reikalavimas
SSKPR-01	Sistema turi palaikyti elektroninio pašto adresų atidarymą per įdiegtas elektroninio pašto programas.

5.14 lentelė. Programavimo aplinkos reikalavimai

Reikalavimo Nr.	Reikalavimas
PAR-01	Sistema turi palaikyti elektroninio pašto adresų atidarymą per įdiegtas elektroninio pašto programas.
PAR-02	Sistema turi būti kuriama naudojant Typescript kalbą.
PAR-03	Sistema turi būti kuriama naudojant Angular ir Express karkasus.
PAR-04	Sistema turi būti kuriama naudojant Visual Studio Code programavimo aplinką.

5.3.2. Vaizdavimo tikslumo reikalavimai

5.15 lentelė. Tikslumo reikalavimai

Reikalavimo Nr.	Reikalavimas
TR-01	Data rodoma formatu metai-mėnesis-diena
TR-02	Sistema turi būti kuriama naudojant Typescript kalbą.
TR-03	Sistema turi būti kuriama naudojant Angular ir Express karkasus.
TR-04	Sistema turi būti kuriama naudojant Visual Studio Code programavimo aplinką.

5.16 lentelė. Skaičiavimo tikslumo reikalavimai

Reikalavimo Nr.	Reikalavimas
STR-01	Sistema turi atvaizduoti projekto trukmę vienos dienos tikslumu.
STR-02	Sistema turi pateikti patirties įgijimo laikotarpį mėnesių tikslumu.
STR-03	Naudotojo amžius skaičiuojamas metų tikslumu.

5.17 lentelė. Patikimumo reikalavimai

Reikalavimo Nr.	Reikalavimas
PR-01	Kritiniai sistemos trikdžiai turi būti pašalinti per mažiau nei 2 valandas.
PR-02	Ne kritiniai sistemos trikdžiai turi būti pašalinti per mažiau nei 4 valandas.

5.18 lentelė. Robastiškumo reikalavimai

Reikalavimo Nr.	Reikalavimas
ROR-01	Sistema turi turėti atsargines kopijas atskirame serveryje ir atnaujinti jas ne rečiau nei kas mėnesį.
ROR-02	Duomenų bazės atsarginės kopijos turi būti atnaujinamos ne rečiau nei kas 2 savaites.

5.19 lentelė. Našumo reikalavimai

Reikalavimo Nr.	Reikalavimas
NR-01	Sistema turi atlikti registracijos patvirtinimą ne ilgiau nei per 4 sekundes.
NR-02	Sistema turi atlikti darbuotojų ar projektų paiešką ne ilgiau nei per 5 sekundes.

5.3.3. Diegimo reikalavimai

Sistema veiks kaip internetinis puslapis, norint užtikrinti, jog visi vartotojai naudotųsi ta pačia versija bei sumažinant galimybę vartotojui neteisingai įsirašyti programinę įrangą.

5.20 lentelė. Ruošinio reikalavimai

Reikalavimo Nr.	Reikalavimas
RU-01	Sistema turi būti pateikta „GitHub“ platformoje, pagrindiniame repozitorijos skyriuje.

5.21 lentelė. Instaliavimo reikalavimai

Reikalavimo Nr.	Reikalavimas
IR-01	Sistema turi būti įdiegiama skirtingomis stadijomis: 1) duomenų bazė; 2) back-end; 3) front-end;

5.22 lentelė. Pradinio duomenų bazių kaupimo reikalavimai

Reikalavimo Nr.	Reikalavimas
PDBKR-01	Produkcijos Duomenų bazėje turi egzistuoti vienas vartotojas su administratoriaus role, kuris galėtų, reikalui esant, redaguoti ar ištrinti įrašus.

5.23 lentelė. Sistemos įsisavinamumo reikalavimai

Reikalavimo Nr.	Reikalavimas
SJR-01	Sistemos instaliaciją ir atnaujinimo procesai turi būti dokumentuoti lietuvių kalba.

5.3.4. Kiti nefunkciniai reikalavimai

5.24 lentelė. Aptarnavimo ir priežiūros reikalavimai

Reikalavimo Nr.	Reikalavimas
AIPR-01	Po pirmojo sistemos paleidimo, sistema turi būti palaikoma sistemos kūrėjų bent pusę metų.
AIPR-02	Sistemos atnaujinimai turi vykti tarp 12-8 valandų, ne dažniau nei kartą per 2 savaites.
AIPR-03	Naujo funkcionalumo kūrimas yra vykdomas pasirašius naują sutartį už sutartinę kainą.

5.25 lentelė. Apsaugos reikalavimai

Reikalavimo Nr.	Reikalavimas
APR-01	Naudotojui nesinaudojant sistema daugiau nei 15 minučių, tačiau būnant prisijungus, sesija yra automatiškai uždaroma.
APR-02	Vartotojo slaptažodis turi būti užšifruojamas prieš saugant į duomenų bazę.
APR-03	Sistema turi būti apsaugota nuo HTML kode paliktų vietų JavaScript funkcijų įvedimui.
APR-04	Naudotojo duomenys gali būti keičiami tik pačio naudotojo.
APR-05	Slaptažodis privalo būti ne trumpesnis nei 7 simbolių.

5.26 lentelė. Juridiniai reikalavimai

Reikalavimo Nr.	Reikalavimas
JU-01	Sistema turi būti kuriama naudojant galiojančias licencijas.
JU-02	Sistema turi nepažeisti Lietuvos Respublikos asmens duomenų apsaugos įstatymo Nr. I-1374

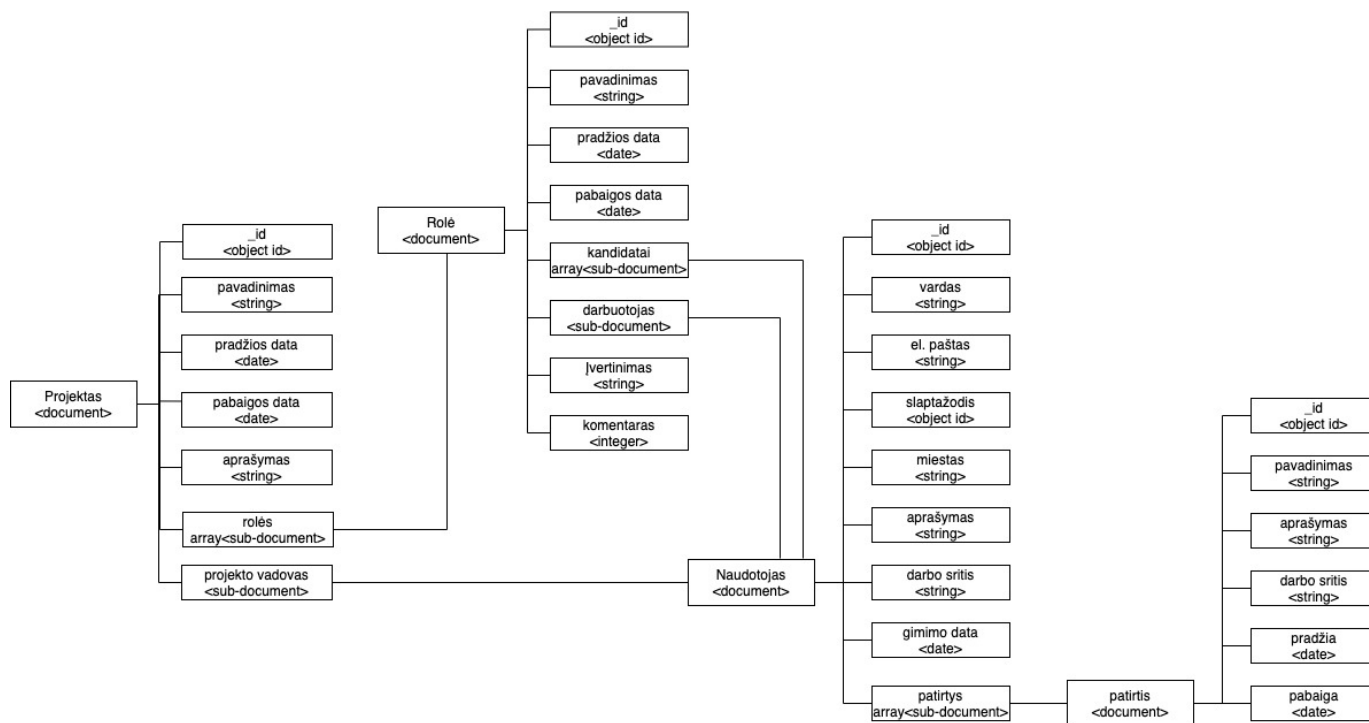
6. Informacinės sistemos projektavimas

6.1. Pasirinktos sistemų kūrimo technologijos

Sistemos kūrimui buvo pasirinktos populiarios ir plačiai naudojamos technologijos. Dėl šios priežasties informacijos kiekis apie jas internete yra platus ir tai padeda lengviau pasirinkti teisingus programavimo sprendimus. Sistemos realizacijai pasirinkta *JavaScript* kalba. Šios kalbos privalumas yra platus jos panaudojimo spektras- su ja galima programuoti tiek front-end tiek back-end. Taip pat norint labiau struktūrizuoti kodą parašyta su *JavaScript* buvo panaudotas plėtinys *TypeScript*. Plėtinys yra naudojamas norint užtikrinti didesnę kodo skaitomumą bei sumažinti klaidų skaičių. Front-end sistemos daliai naudojama karkaso *Angular* 8 versija, stilius pasirinktas „Workwise“ iš themeForest internetinės parduotuvės ir perdarytas pagal specifinius reikalavimus. Back-end daliai naudojamas *NodeJS* variklis bei *Mongoose* paketas, kuris kodą konvertuoja į NoSQL užklausas. Sistemos programavimas atliekamas *Visual Studio Code* programavimo aplinkoje, kuri užima mažiau atminties ir veikia greičiau nei jos konkurentai.

6.2. Dalykinės srities duomenų modeliavimas

Sistemoje naudojama NoSQL duomenų bazė turinti penkias kolekcijas: Naudotojas, Projektas, Patirtis, Rolė ir Įvertinimas. Ryšiai tarp šių kolekcijų yra pavaizduoti žemiau esančioje diagramoje (žr. 6.1 pav.).



6.1 pav. Sistemos duomenų modelio diagrama

Pagrindinės lentelės, sukurtos remiantis funkciniais reikalavimais, yra:

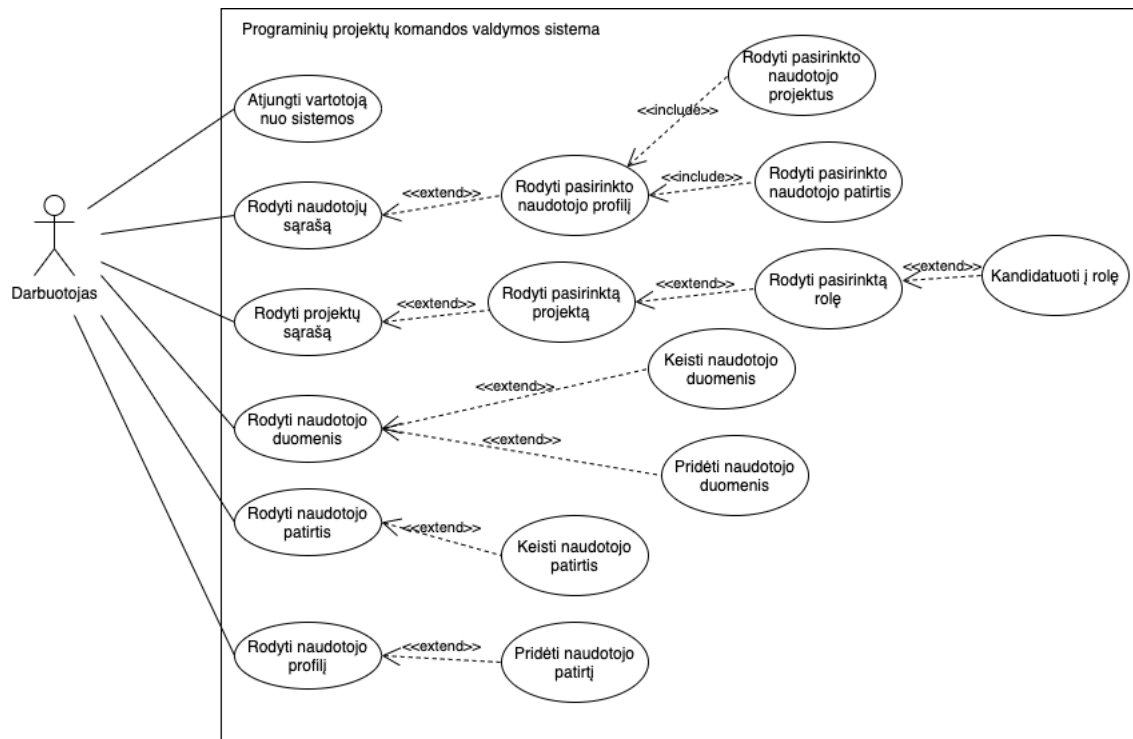
1. Naudotojas – registruotas asmuo turinti savo anketą bei turintis galimybę ją papildyti išsamesne informacija apie save.
2. Patirtis – naudotojo įvedamas įrašas apie anksčiau turėtas darbo patirtis, projektus arba atliktus mokymus.
3. Projektas – naudotojo turinčio rolę „Projekto vadovas“ sukurtas įrašas su intencija susirasti komandą darbui prie aprašyto projekto. Projektas yra aprašomas, jam skiriamas pavadinimas bei laiko terminas.
4. Rolė – įrašas kuris priklauso projektui ir vienas be projekto egzistuoti negali. Jį sukuria naudotojas savo projekto puslapyje ir skiria jai pavadinimą, aprašymą bei tipą.
5. Įvertinimas – įrašas įvedamas naudotojo kuris yra priskirtas kaip darbuotojas egzistuojančiai rolei. Įvertinimas skirtas aprašyti kaip sekėsi dirbti komandoje užimant tam tikrą rolę. Įvertinimas rašomas pasibaigus projektui.

6.3. Sistemos funkcijų projektavimas

6.3.1. Funkcijų pasiskirstymas pagal naudotojų vaidmenis ir vykdymo eiga

Naudotojas priklausantis rolei „Darbuotojas“

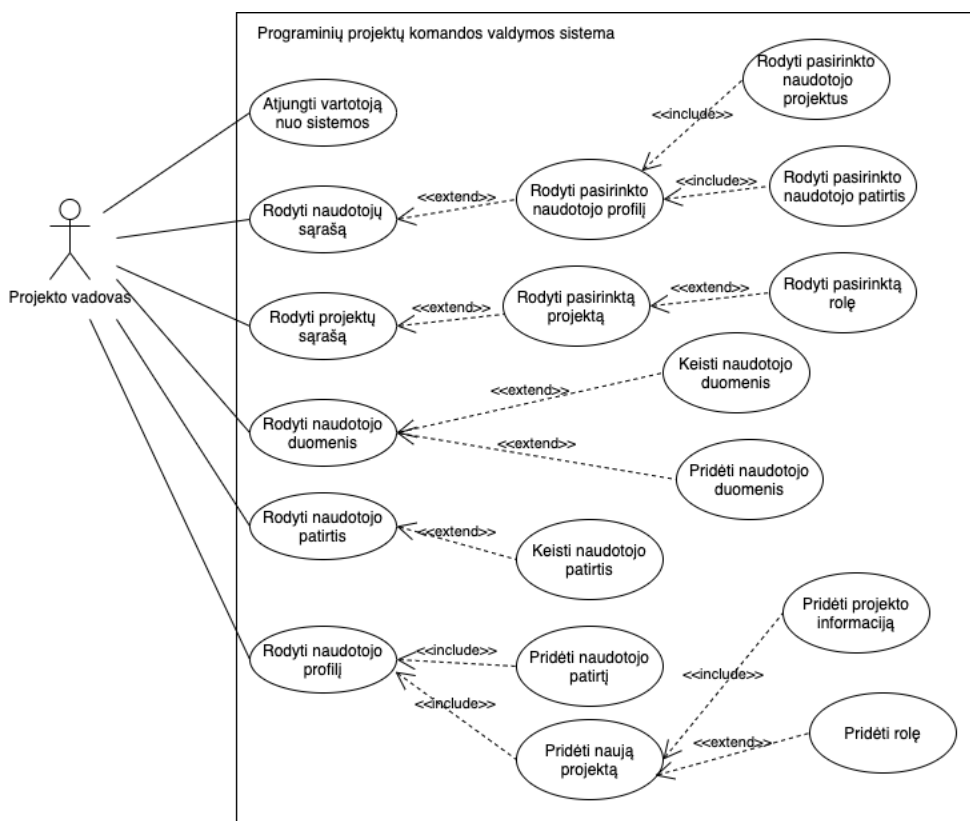
Naudotojo kuris yra prisiregistravęs su role „Darbuotojas“, galimos užduotys yra pateiktos panaudos atvejų diagramoje (žr. 6.2 pav.).



6.2 pav. Rolės „Darbuotojas“ panaudos diagrama

Naudotojas priklausantis rolei „Projekto vadovas“

Naudotojo kuris yra prisiregistravęs su role „Projekto vadovas“, galimos užduotys yra pateiktos panaudos atvejų diagramoje (žr. 6.3 pav.).



6.3 pav. Rolės „Projekto vadovas“ panaudos diagrama

6.4. Atskirų funkcijų projektavimas

Atskirų funkcijų projektavimas yra svarbi sistemos architektūros dalis, nes ji padeda suvokti kokias mažesnes funkcijas reikia atlikti norint įgyvendinti tam tikrą funkcionalumą.

6.4.1. Sukurti patirtį

Sukurti naują patirtį gali naudotojas turintis bet kokią rolę. Patirtį galima sukurti iš asmeninio profilio puslapio. Įvedant naują patirtį būtina nurodyti:

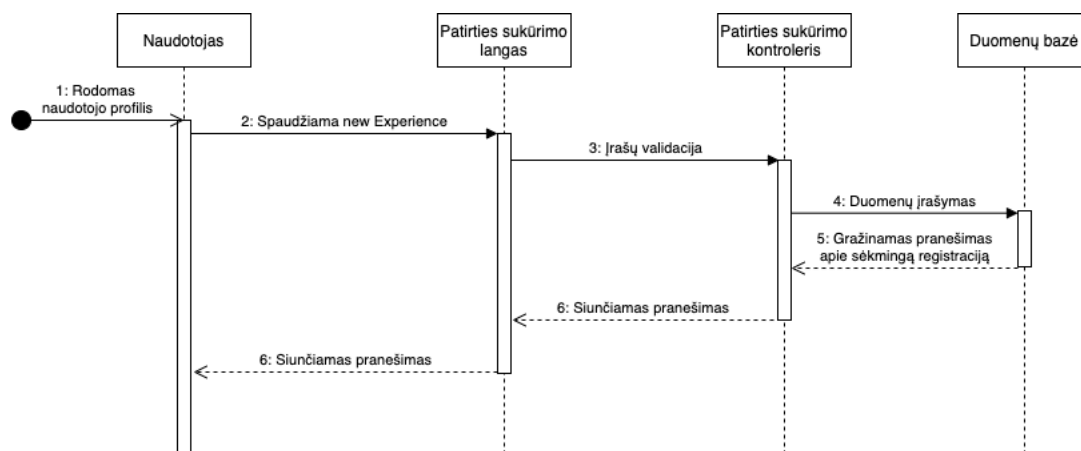
- Pavadinimą
- Aprašymą
- Pradžios ir pabaigos datas
- Patirties tipą

Pavadinimas privalo būti trumpesnis nei 30 simbolių, Įvedant pradžios datą negalima įvesti datos vėlesnės nei einamoji diena. Pridėta patirtis atvaizduojama profilio puslapyje po skirtuku Patirtys. Norint redaguoti pasirinktą patirtį reikia pasirinkti redagavimo funkciją, kuri atidaro patirties formą su užpildytais pasirinktos patirties laukais kuriuos galima keisti. Pakeitus laukus ir paspaudus *save* mygtuką turi atsirasti

pranešimas jog patirtis buvo sėkmingai atnaujinta. Prie *save* mygtuko taip pat yra pasirinkimas ištrinti patirtį. Paspaudus šį mygtuką atsiranda pranešimas apie sėkmingai ištrintą patirtį.

6.1 lentelė. Patirties sukūrimas

užduoties numeris	1
užduoties pavadinimas	Sukurti naują patirtį.
siekiamas tikslas	Sukurtas naujas patirties įrašas duomenų bazėje.
užduoties vykdymą inicijuojantis įvykis (triggeris)	Atvaizduojama patirties registracijos forma.
"prieš" sąlygos	Naudotojas yra atsidaręs savo profilio puslapį.
sėkmingos baigties "po" sąlygos	Naudotojui atvaizduojamas pranešimas apie sukurtą patirtį.
nesėkmingos baigties sąlygos	Registracijos laukai yra užpildyti netinkamai, neatitinka validacijos reikalavimų.
laiko sąnaudos užduočiai vykdyti	10 minučių užpildyti patirties kūrimo laukus sistemoje.
Po užduotys:	Naudotojo patirties registracijos patvirtinimas; Naujos patirties įrašymas į duomenų bazę; Naujos patirties atvaizdavimas profilio puslapyje.



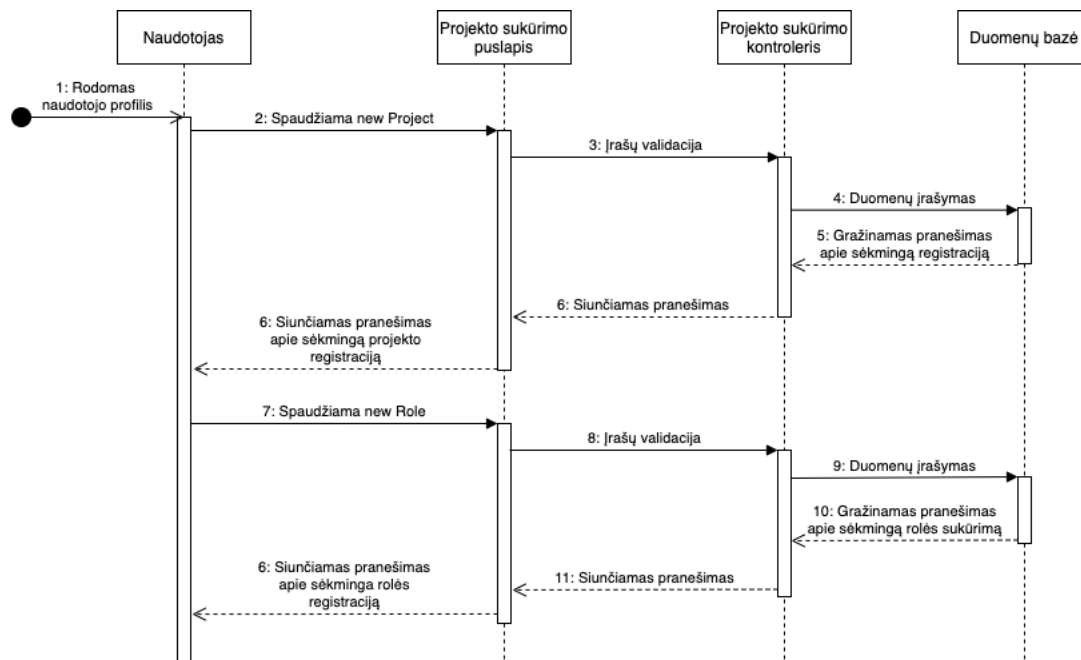
6.4 pav. Patirties sukūrimo sekų diagrama

6.4.2. Sukurti projektą ir pridėti rolę

Projektą sukurti gali tik naudotojas priklausantis rolei „Projekto vadovas“. Norint sukurti naują projektą reikia būti atsidarius savo profilio puslapį ir pasirinkti mygtuką „Naujas Projektas“. Projekto kūrimo formoje visi laukai yra privalomi ir pabaigos data negali būti anksčiau nei dabartinė diena. Sukūrus projektą jį galima keisti ir ištrinti, taip pat atsiranda funkcija pridėti naują rolę. Sėkmingai sukūrus projektą atvaizduojamas pranešimas indukuojantis, jog projektas buvo sukurtas. Pasirinkus pridėti rolę atvaizduojama rolės formą, kurioje įvedama informacija apie ją. Išsaugojus naują rolę atsiranda galimybė ja keisti ir ištrinti, taip pat jos pavadinimas yra atvaizduojamas rolių sąrašė.

6.2 lentelė. Projekto ir rolės sukūrimas

užduoties numeris	2
užduoties pavadinimas	Sukurti naują projektą ir pridėti rolę
siekiamas tikslas	Sukurta naujas projekto įrašas duomenų bazėje; Sukurta naujas rolės įrašas duomenų bazėje.
užduoties vykdymą inicijuojantis įvykis (triggeris)	Atvaizduojama projekto registracijos forma.
"prieš" sąlygos	Naudotojas yra atsidaręs savo profilio puslapį.
sėkmingos baigties "po" sąlygos	Naudotojui atvaizduojamas pranešimas apie sukurtą projektą; Naudotojui atvaizduojamas pranešimas apie sukurtą rolę; Nauja rolė atvaizduojama rolių sąrašė.
nesėkmingos baigties sąlygos	Registracijos laukai yra užpildyti netinkamai, neatitinka validacijos reikalavimų.
laiko sąnaudos užduočiai vykdyti	10 minučių užpildyti projekto kūrimo laukus sistemoje; 5 minutės užpildyti vienos rolės laukus.
Po užduotys:	Naudotojo projekto registracijos patvirtinimas; Naujo projekto įrašymas į duomenų bazę; Naujos rolės įrašymas į duomenų bazę; Naujo projekto atvaizdavimas profilio puslapyje; Naujos funkcijos rolės sukūrimui atvaizdavimas projekto puslapyje.



6.5 pav. Projekto kūrimo sekų diagrama

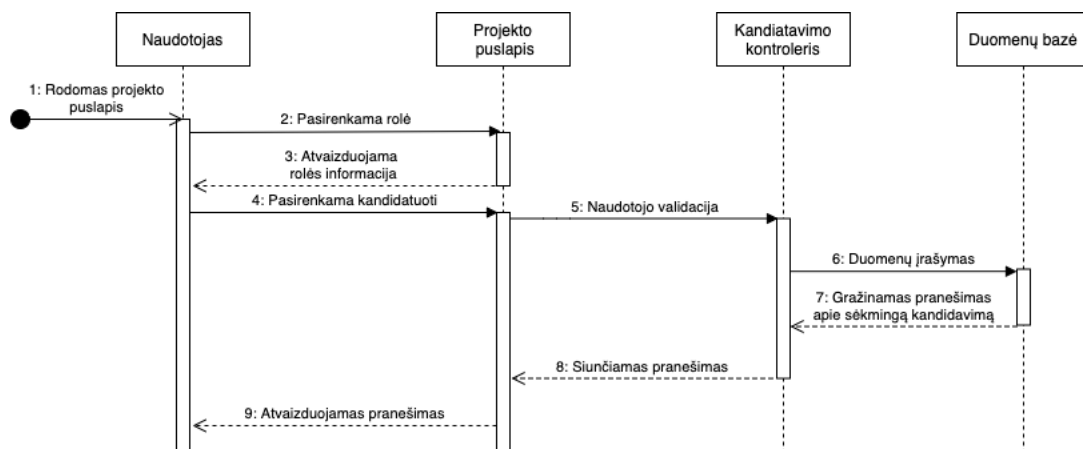
6.4.3. Kandidatuoti į rolę

Norint kandidatuoti į bet kurią projekto rolę reikia būti užsiregistravus kaip darbuotojui. Suradus norimą projektą per paiešką apie pagrindiniame puslapyje ir paspaudus ant jo gali pamatyti projekto

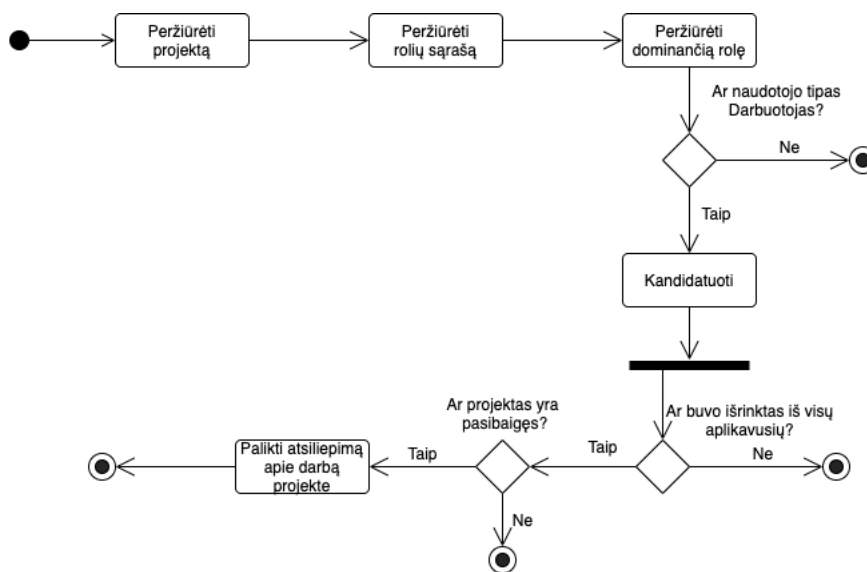
aprašymą ir visas jam priklausomas roles. Pasirinkus vieną iš rolių galima pamatyti jos aprašymą bei darbo tipą. Paspaudus ant mygtuko „Aplikuoti“ naudotojas yra pridedamas į kandidatų sąrašą kurį mato tik to projekto vadovas. Po sėkmingo naudotojo įrašymo į kandidatų sąrašą, žinutė apie sėkmingai įvykdytą veiksmą yra atvaizduojama.

6.3 lentelė. Kandidatavimas į projekto rolę

užduoties numeris	3
užduoties pavadinimas	Kandidatuoti į projekto rolę.
siekiamas tikslas	Kandidatų sąrašas duomenų bazės rolės įrašė papildomas naudotoju.
užduoties vykdymą inicijuojantis įvykis (triggeris)	Atvaizduojama informacija apie projektą.
"prieš" sąlygos	Naudotojas yra atsidaręs projekto puslapį.
sėkmingos baigties "po" sąlygos	Naudotojui atvaizduojamas pranešimas apie sėkmingą pridėjimą prie kandidatų sąrašo.
nesėkmingos baigties sąlygos	Registracijos laukai yra užpildyti netinkamai, neatitinka validacijos reikalavimų.
laiko sąnaudos užduočiai vykdyti	10 minučių užpildyti projekto kūrimo laukus sistemoje.
Po užduotys:	Kandidato įrašymas į duomenų bazę; Kandidato atvaizdavimas kandidatų sąrašė.



6.6 pav. Kandidavimo sekų diagrama



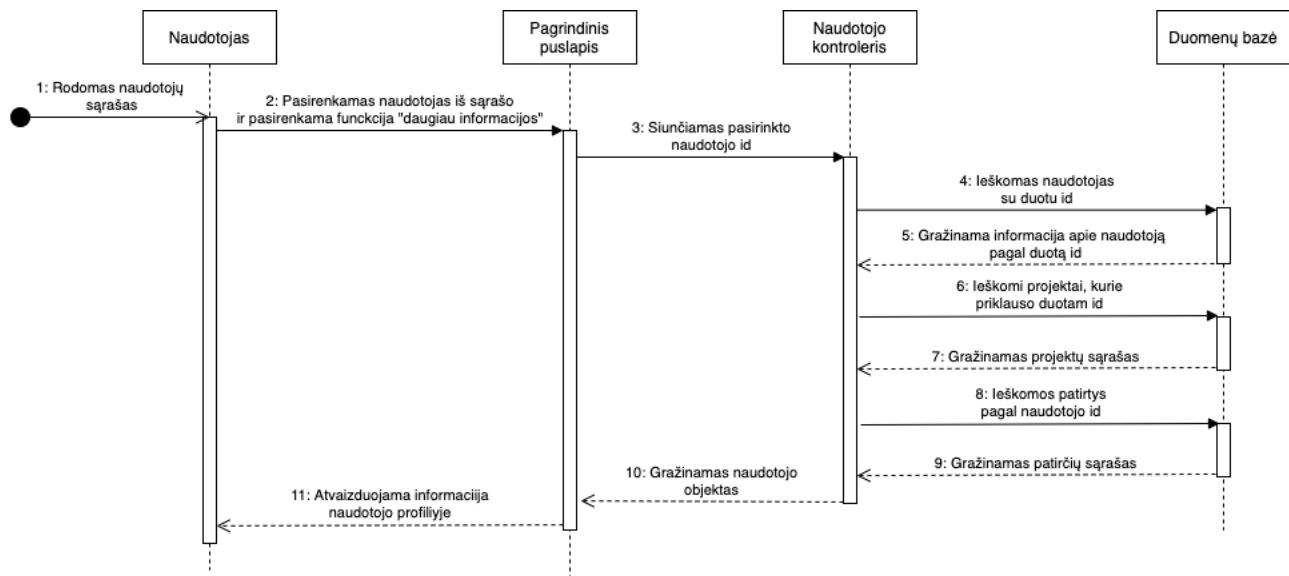
6.7 pav. Kandidavimo būsenų diagrama

6.4.4. Naudotojo profilio atvaizdavimas

Visi naudotojai gali matyti visų naudotojų profilius. Pagrindiniame puslapyje žvalgantis po naudotojų sąrašą ir susidomėjęs vienu iš naudotojų galima pasirinkti funkciją „Daugiau informacijos“, kuri atidaro naudotojo profilį. Naudotojo profilyje matoma pagrindinė jo informacija: duomenys, darbo ar mokslo patirtys bei sukurti projektai arba projektai kuriuose naudotojas dalyvauja.

6.4 lentelė. Atsiliepimo apie projektą palikimas

užduoties numeris	4
užduoties pavadinimas	Naudotojo profilio atvaizdavimas
siekiamas tikslas	Atvaizduotas naudotojo profilis
užduoties vykdymą inicijuojantis įvykis (trigeris)	Naudotojo pagrindinė informacija atvaizduota pagrindiniame puslapyje, naudotojų sąraše.
"prieš" sąlygos	Naudotojas pamatęs jį dominantį naudotoją sąraše, pasirenka funkcija „Daugiau Informacijos“
sėkmingos baigties “po” sąlygos	Naudotojo profilis su visom naudotojo patirtimis bei projektais yra atvaizduojamas.
nesėkmingos baigties sąlygos	Naudotojo profilio informacija nėra užkrauta, nes naudotojas buvo ištrintas, o puslapis nebuvo perkrautas.
laiko sąnaudos užduočiai vykdyti	5 sekundės profilio informacijai užkrauti.
Po užduotys:	Atvaizduota pasirinkto naudotojo informacija.



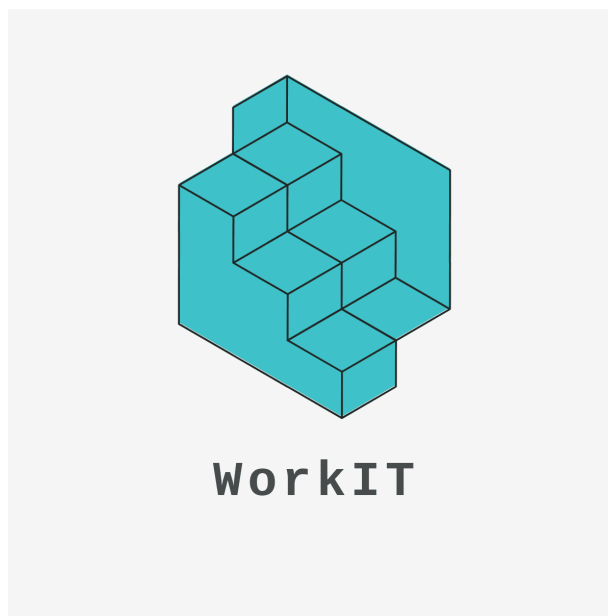
6.8 pav. Profilio atvaizdavimo sekų diagrama

7. Sistemos realizacija

7.1. Sistemos dizainas

Sistemos dizainui buvo pasirinkta naudoti jau kitų asmenų sukurtą dizainą ir jį perdaryti pagal funkcinio reikalavimus. Dizainas buvo panaudotas iš internetinės dizaino puslapių parduotuvės www.themeforest.net, dizaino autorius – „Gambolthemes“. Pagrindinis programinių projektų komandos formavimo ir valdymo sistemos dizaino privalumas yra mažas atskirų puslapių skaičius. Norint sukurti projektą nereikia eiti per tris skirtingus etapus, taip pat norint pridėti patirtį tai galima atlikti atsidarius modalinį puslapį, kuris užima mažą dalį ekrano ir nereikalauja puslapio perkrovimo.

Sistemos logotipui buvo naudojama nemokama internetinė logotipų generavimo sistema – „hatchful“. Logotipo kūrimas yra ilgas procesas ir galimybė jį tiesiog susigeneruoti labai paspartina projekto kūrimą. Generuojant logotipą buvo stengiamasi naudoti panašią į tinklapio spalvą, tačiau ne identišką, nes tokiu atveju logotipas susilies su kitais sistemos komponentais.



7.1 pav. Sistemos logotipas

Visame tinklapyje yra naudojamos piktogramos geriau apibūdinančios mygtukų prasmes bei padedančios naudotis sistema greičiau ir patogiau.

7.2. Sistemos funkcionalumo programavimas

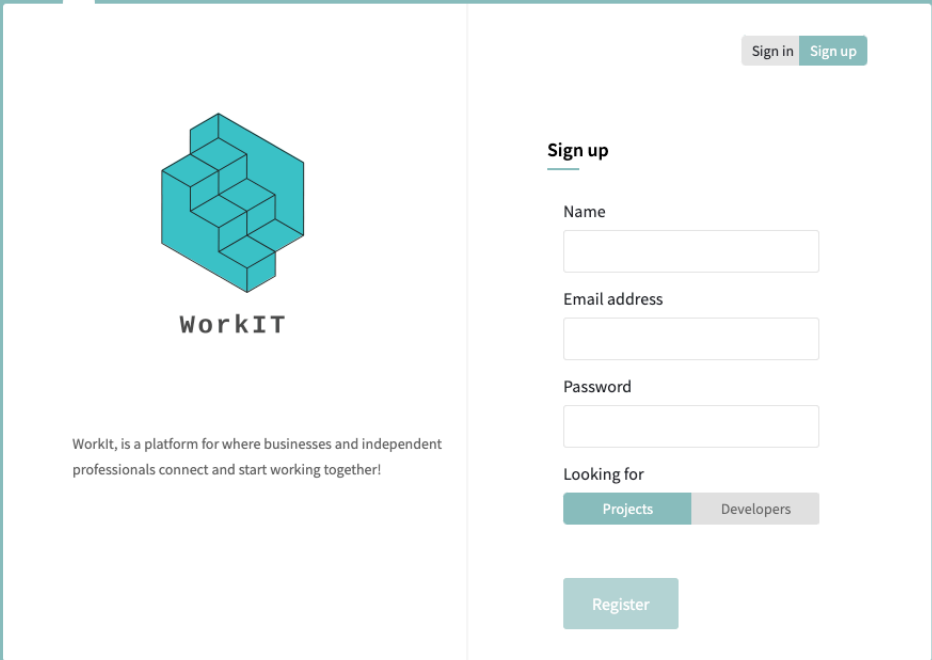
Sistemos pagrindiniam funkcionalumui buvo naudojamas *TypeScript* programavimo kalba ir *Angular* karkasas. Šios technologijos padėjo sistemos dalis sumažinti į mažesnius komponentus ir leisti juos pernaudoti. Naudojant atskirus failus sistemos servisams kurti padėjo išvengti kodo dubliavimosi. Taip pat atskiri failai skirti tik aprašyti puslapio stilių padėjo atskirti sistemos funkcijas ir skaičiavimus nuo dizaino

elementų. *Angular* karkase jau esančios formų validacijos sutrumpino programavimo procesą kuriant daugiau nei šešias skirtingas forma. Aplankalas skirtas sistemos modeliams leido per panaudoti visus duomenų modelius, todėl galima buvo užtikrinti, jog bet kokiame komponente gaunant duomenis iš duomenų bazės ar iš įvestos formos, jie yra teisingų tipų.

7.3. Įvedamų duomenų patikra

7.3.1. Prisijungimas ir registracija

Prisijungimas ir registracija vyksta tame pačiame puslapyje, tačiau yra padalintas į du skirtingus skirtukus. Norint užsiregistruoti yra reikalaujama įvesti prisijungimo vardą, kuris negali būti ilgesnis nei 20 simbolių, elektroninį paštą, kuriam yra skirta speciali validacija – elektroninis paštas negali būti toks pats koks jau egzistuoja sistemoje ir privalo atitikti reguliariąjį reiškinį – `[a-z0-9._%+-]+@[a-z0-9.-]+.[a-z]{2,4}$`. Taip pat norint prisiregistruoti reikalingas slaptažodis bei tipas – darbuotojas ar projekto vadovas. Neatitikus visų šių reikalavimų registracijos mygtukas yra nepaspaudžiamas.

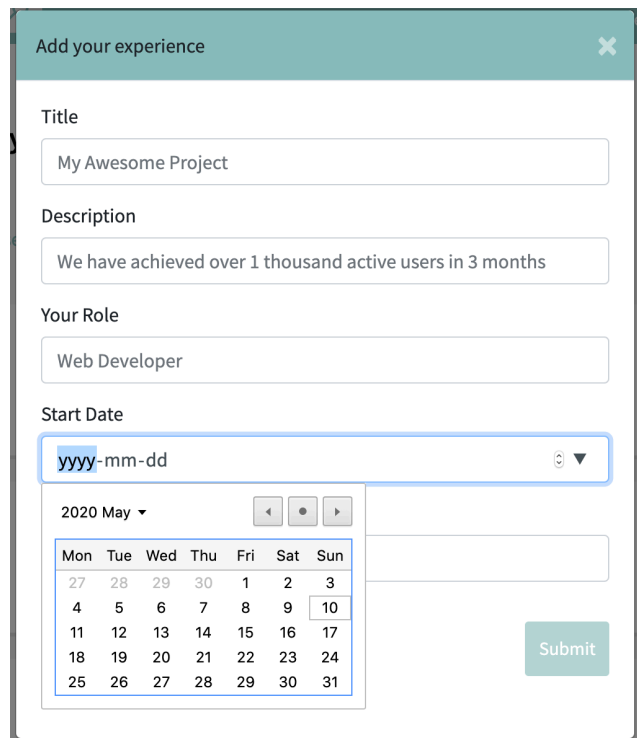


7.2 pav. Registracijos ir prisijungimo puslapis

7.3.2. Datų validavimas.

Datos yra svarbi kiekvieno projekto dalis. Norint sukurti projektą reikia bent preliminariai žinoti kada prasidės programinės įrangos kūrimas ir kada bus pirmas jos paleidimas į viešumą. Taip pat datos yra

svarbios ieškant darbuotojų, kurių darbo trukmė tam tikroje įmonėje gali turėti didžiulės įtakos renkant tarp kandidatų. Norint užtikrinti, jog naudotojai atsitiktinai arba neatsitiktinai neįvestu sistemai nesuprantamu datų visi datų įvedimo laukai yra nustatyti priimti datą, kurios formatas yra Metai-Mėnuo-Diena.



The screenshot shows a web form titled "Add your experience" with a close button (X) in the top right corner. The form contains four input fields: "Title" (with the text "My Awesome Project"), "Description" (with the text "We have achieved over 1 thousand active users in 3 months"), "Your Role" (with the text "Web Developer"), and "Start Date". The "Start Date" field is highlighted with a blue border and contains the placeholder text "yyyy-mm-dd". Below the "Start Date" field, a calendar for May 2020 is displayed. The calendar has a header with "2020 May" and navigation arrows. The days of the week are listed in the first row: Mon, Tue, Wed, Thu, Fri, Sat, Sun. The dates are listed in the subsequent rows. The date "10" is highlighted in the calendar. A "Submit" button is located at the bottom right of the form.

Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

7.3 pav. Datos pasirinkimo formatas

Taip pat stengiantis išvengti neadekvačių datų įvedimo buvo panaudotos minimalios ir maksimalios datos įvedimui. Pavyzdžiui pridedant naują darbo patirtį ji negali būti vėlesnė nei ši diena, todėl šiuo atveju maksimali diena yra šiandien. Kuriant projektą jo pabaigos diena negali būti jau praeityje, nes tai reikštu, jog projektas yra pasibaigęs jam dar net neprasidėjus, todėl tokiu atveju minimali data yra šiandien.

7.3.3. Naujos rolės įvedimo reikalavimai

Rolės sukūrimo, kuri nepriklauso jokiame projektui būti negali, todėl norint užtikrinti, jog vartotojas pirmiausia sukurs projektą, o tik vėliau pridės rolę, mygtukas, kuris atidaro rolės forma yra paslėptas iki kol vartotojas sukuria projektą. Užpildžius visus projekto informacijos laukus ir pasirodžius informacinei žinutei apie sėkmingai sukurta projektą, kairėje pusėje, atsiranda mygtukas siūlantis pridėti naują rolę (žr. 7.4 pav.). Tokiu būdu naudotojas gali lengviau sekti instrukcijas ir jis nėra apkraunamas didžiuliu kiekiu formų, kurios tikrai nemotyvuoja registruoti naują projektą.

Status

Main info

New Role

Project Registration

Title

Darbo skelbimai

Description

Kuriamas naujas projektas skirtas darbuotojams lengviau susirasti darbus pagal projektus.

Start Date

2019-09-01

Expected End Date

2020-06-01

Submitted Successfully! Press 'New Role' to add team role

Delete

Update

Save & Quit

7.4 pav. Sėkmingai sukurtas projektas

8. Informacinės sistemos testavimas

Šis skyrius skirtas testavimo atvejams bei kelių iš jų atvejų aprašymui, taip pat jų pritaikymas automatiniams testams. Visi duomenys kurie buvo naudojami testavime yra pavyzdinei ir sukurti testavimo tikslams.

8.1. Testavimo scenarijai

Testavimo scenarijai yra sukurti remiantis jau sukurtais reikalavimais ir jie yra skirti patikrinti ar sukurta sistema atitinka iškeltus reikalavimus. Žemiau pateiktoje lentelėje yra pateikti testavimo scenarijai su scenarijaus numerius, reikalavimo numeriu, aprašymo ir testo scenarijaus svarba. Svarba gali būti skirtingu lygių ir yra dalinamas:

- Kritinis – reikalavimas turi įtakos pagrindiniam sistemos funkcionalumui todėl privalo būti atsakingai ištestuotas įvairiais būdais;
- Aukštas – reikalavimas yra susijęs su pagrindinėmis sistemos funkcijomis ir jo neįgyvendinimas turėtų įtakos bazinėms užduotims atlikti;
- Vidutinis – reikalavimas užtikrina, jog sistema bus patogiau naudotis, jeigu jis bus įgyvendintas;
- Mažas – reikalavimas aprašantis mažą dalį funkcionalumo, kurio pagrindinis tikslas yra pagerinti naudotojo patirtį naudojantis sistema, tačiau jo nebuvimas nesukeltu problemų atlikti bet kokias užduotis.

8.1 lentelė. Testavimo scenarijai

Scenarijaus numeris	Reikalavimo numeris	Aprašymas	Svarba
TS-01	IDSR-02	Ištestuoti, ar vartotojo sąsaja yra pasiekama per naršyklę.	kritinė
TS-02	IDSR-03	Ištestuoti, ar vartotojo sąsajos elementais yra atvaizduojami per naršyklę mobiliajame įrenginyje.	aukšta
TS-03	PFR-01	Ištestuoti, ar yra atvaizduojamas pranešimas kai nėra galimybės prisijungti prie duomenų kontrolierio.	aukšta
TS-04	PFR-02	Ištestuoti, ar sistema atvaizduoja pranešimą registruojantis, kai jau yra sukurtas profilis su tokiu pačiu elektroniniu paštu.	aukšta
TS-05	PFR-03	Ištestuoti, ar sistema atvaizduoja klaidos pranešimą, kai yra bandoma įvesti skaičius į vartotojo vardo lauką.	vidutinė
TS-06	DSR-01	Ištestuoti, ar įvedus visus reikiamus laukus registracijos formoje naudotojas yra perkeliamas į pagrindinį puslapį ir jo profilis yra sukuriamas.	kritinė
TS-07	DSR-02	Ištestuoti, ar turint naudotojo anketą galima prie sistemos prisijungti.	kritinė
TS-08	DSR-03	Ištestuoti, ar esant prisijungus prie sistemos galima atsijungti.	vidutinė
TS-09	DSR-04	Ištestuoti, ar registruojantis vartotojas gali pasirinkti ir rolę ir ar informaciją sistemoje yra atvaizduojama pagal rolės tipą.	kritinė
TS-10	DSR-05	Ištestuoti, ar pakeitus naudotojo duomenys, šie duomenys yra atnaujinami.	aukšta

Scenarijaus numeris	Reikalavimo numeris	Aprašymas	Svarba
TS-11	DSR-06	Ištestuoti, ar naudotojui pridėjus duomenis apie save jie yra atvaizduojami jo profilyje.	aukšta
TS-12	DSR-07	Ištestuoti, ar naudotojui pridėjus naują patirtį ji yra atvaizduojama naudotojo profilyje.	aukšta
TS-13	DSR-08	Ištestuoti, ar naudotojas gali redaguoti savo sukurtas patirtis.	aukšta
TS-14	DSR-09	Ištestuoti, ar projekto vadovas gali sukurti projektą.	kritinė
TS-15	DSR-10	Ištestuoti, ar projekto vadovas gali redaguoti projektą.	aukšta
TS-16	DSR-11	Ištestuoti, ar projekto vadovas prie egzistuojančio projekto gali pridėti roles.	aukšta
TS-17	DSR-12	Ištestuoti, ar projekto vadovas sukūręs projektą su rolėmis gali jas redaguoti.	vidutinė
TS-18	DSR-13	Ištestuoti, ar naudotojas gali peržiūrėti visus egzistuojančius projektus.	kritinė
TS-19	DSR-14	Ištestuoti, ar darbuotojas gali kandidatuoti į projekto rolę.	aukšta
TS-20	DSR-15	Ištestuoti, ar projekto vadovas gali peržiūrėti visus kandidatus į jo sukurtą rolę.	aukšta
TS-21	DSR-16	Ištestuoti, ar projekto vadovas gali pasirinkti darbuotoją iš kandidatų sąrašo.	aukšta
TS-22	DSR-17	Ištestuoti, ar naudotojas gali peržiūrėti kitų naudotojų profilius.	kritinė
TS-23	DSR-18	Ištestuoti, ar naudotojas gali vykdyti paieška pagal naudotojų vardus.	vidutinė
TS-24	DSR-19	Ištestuoti, ar naudotojas gali vykdyti paieška pagal projektų pavadinimus.	vidutinė
TS-25	DSR-20	Ištestuoti, ar naudotojas gali vykdyti paieška pagal rolių tipus.	vidutinė
TS-26	OSNR-01	Ištestuoti, ar sistema veikia visose naršyklėse palaikančiose HTML5.	kritinė
TS-27	DMR-01	Ištestuoti, ar sistema siunčiami duomenys yra JSON formato.	aukšta
TS-28	DMR-02	Ištestuoti, ar duomenų bazės duomenys yra pasiekiam CSV formatu.	maža
TS-29	TR-01	Ištestuoti, ar data pateikiama formatu „metai-mėnesis-diena“.	maža
TS-30	APR-01	Ištestuoti, ar naudotojas yra išregistruotas po 15 minučių nesinaudojant sistema.	maža
TS-31	APR-02	Ištestuoti, ar naudotojų slaptažodžiai duomenų bazėje yra saugomi užšifruoti.	kritinė
TS-32	APR-04	Ištestuoti, ar naudotojas gali keisti tik savo sukurtus duomenis ir patirtis.	kritinė
TS-32	APR-05	Ištestuoti, ar naudotojo įvedamas slaptažodis yra ilgesnis nei 7 simbolių	vidutinė

8.2. Testavimo atvejai

Testavimo atvejai yra parengti keliems testavimo scenarijams. Atvejai pateikti žemiau esančioje lentelėje su testavimo atvejo numeriu,, aprašymu kas turi būti atlikta prieš vykdant testą, pačio testavimo eiga išvardinta žingsniais, laukiamu rezultatu bei atsakymu (taip arba ne) ar rezultatas buvo gautas tenkinamas.

8.2 lentelė. Testavimo atvejai

Testavimo atvejo numeris	Testavimo scenarijus	Kas turi būti padaryta prieš vykdant testą	Testavimo eiga	Laukiamas rezultatas	Rezultatas
TA-01	TS-03 Ištestuoti, ar yra atvaizduojamas pranešimas kai nėra galimybės prisijungti prie duomenų kontrolierio.	Atidaryta naudotojo registracijos forma. Išjungtas serveris komunikuojantis su duomenų baze.	1. Įvedamas naudotojo vardas, elektroninis paštas, slaptažodis; 2. Spaudžiamas mygtukas pateikti registracijos formą.	Atvaizduojamas pranešimas, jog įvyko klaida, prašome pamėginti vėliau.	taip
TA-02	TS-04 Ištestuoti, ar sistema atvaizduoja pranešimą registruojantis, kai jau yra sukurtas profilis su tokiu pačiu elektroniniu paštu.	Sukuriamas naudotojas. Atsijungiama iš jo paskyros. Atidaroma registracijos forma.	1. Įvedamas naudotojo vardas, elektroninis paštas, kuris buvo įvestas prieš tai, slaptažodis; 2. Spaudžiamas mygtukas pateikti registracijos formą.	Atvaizduojamas pranešimas, jog naudotojas su tokiu elektroniniu paštu jau egzistuoja sistemoje.	taip
TA-03	TS-05 Ištestuoti, ar sistema atvaizduoja klaidos pranešimą, kai yra bandoma įvesti skaičius į vartotojo vardo lauką.	Atidaryta naudotojo registracijos forma.	1. Įvedamas naudotojo vardas su skaitmenimis.	Atvaizduojamas pranešimas, jog vartotojo vardas gali būti sudarytas tik iš raidžių.	taip
TA-04	TS-06 Ištestuoti, ar įvedus visus reikiamus laukus registracijos formoje naudotojas yra perkeliamas į pagrindinį puslapį ir jo profilis yra sukuriamas.	Atidaryta naudotojo registracijos forma.	1. Įvedamas naudotojo vardas, elektroninis paštas, slaptažodis; 2. Spaudžiamas mygtukas pateikti registracijos formą.	Atvaizduojamas pagrindinis puslapis.	taip
TA-05	TS-07 Ištestuoti, ar turint naudotojo anketą galima prie sistemos prisijungti.	Sukurta naudotojo anketa. Atidaryta naudotojo prisijungimo forma.	1. Įvedamas naudotojo elektroninis paštas ir slaptažodis; 2. Spaudžiamas mygtukas prisijungimui.	Atvaizduojamas pagrindiniais puslapis.	taip
TA-06	TS-08 Ištestuoti, ar esant prisijungus prie sistemos galima atsijungti.	Prisijungta prie naudotojo paskyros	1. Spaudžiamas mygtukas atsijungimui.	Atvaizduojamas prisijungimo puslapis.	taip

Testavimo atvejo numeris	Testavimo scenarijus	Kas turi būti padaryta prieš vykdant testą	Testavimo eiga	Laukiamas rezultatas	Rezultatas
TA-07	TS-10 Ištestuoti, ar pakeitus naudotojo duomenys, šie duomenys yra atnaujinami.	Prisijungta prie naudotojo paskyros, atidarytas naudotojo profilis.	<ol style="list-style-type: none"> 1. Spaudžiamas mygtukas duomenų pakeitimui; 2. Įvedamas naujas naudotojo vardas; 3. Spaudžiamas mygtukas duomenų išsaugojimui. 	Atvaizduojamas naudotojo profilis su pakeistais duomenimis.	taip
TA-08	TS-11 Ištestuoti, ar naudotojui pridėjus duomenis apie save jie yra atvaizduojami jo profilyje.	Prisijungta prie naudotojo paskyros, atidarytas naudotojo profilis.	<ol style="list-style-type: none"> 1. Spaudžiamas mygtukas duomenų pakeitimui; 2. Įvedamas aprašymas apie naudotoją; 3. Spaudžiamas mygtukas duomenų išsaugojimui. 	Atvaizduojamas naudotojo profilis su pridėtais duomenimis.	taip
TA-09	TS-12 Ištestuoti, ar naudotojui pridėjus naują patirtį ji yra atvaizduojama naudotojo profilyje.	Prisijungta prie naudotojo paskyros, atidarytas naudotojo profilis.	<ol style="list-style-type: none"> 1. Spaudžiamas mygtukas naujos patirties pridėjimui; 2. Teisingai Įvedami visi laukai; 3. Spaudžiamas mygtukas duomenų išsaugojimui. 	Atvaizduojamas naudotojo profilis su jo nauja patirtimi, patirčių sąraše.	taip
TA-10	TS-13 Ištestuoti, ar naudotojas gali redaguoti savo sukurtas patirtis.	Prisijungta prie naudotojo paskyros, atidarytas naudotojo profilis. Sukurta patirtis.	<ol style="list-style-type: none"> 1. Spaudžiamas redagavimo mygtukas esantis šalia patirties, patirčių sąraše; 2. Įvedama nauja informacija; 3. Spaudžiamas mygtukas duomenų išsaugojimui. 	Atvaizduojamas naudotojo profilis su jo pakeista informacija apie naudotojo patirtį.	taip
TA-11	TS-14 Ištestuoti, ar projekto vadovas gali sukurti projektą.	Prisijungta prie naudotojo priklausančio rolei „Projekto vadovas“. Atidarytas naudotojo profilis.	<ol style="list-style-type: none"> 1. Spaudžiamas mygtukas skirtas projekto sukūrimui; 2. Projekto kūrimo formoje įvedami visi laukai kurių yra reikalaujama; 3. Spaudžiamas mygtukas duomenų užsaugojimui. 	Atvaizduojamas profilio kūrimo puslapis su informaciniu pranešimu apie sėkmingai sukurtą projektą.	taip

Testavimo atvejo numeris	Testavimo scenarijus	Kas turi būti padaryta prieš vykdant testą	Testavimo eiga	Laukiamas rezultatas	Rezultatas
TA-12	TS-15 Ištestuoti, ar projekto vadovas gali redaguoti projektą.	Sukurtas projektas, naudotojas yra prisijungęs prie sistemos ir atsidaręs projekto redagavimo puslapį.	<ol style="list-style-type: none"> Įvedama nauja informacija į laukus, kuriuose yra įvesta sena projekto informacija; Spaudžiamas mygtukas duomenų pakeitimui. 	Atvaizduojamas profilio kūrimo puslapis su informaciniu pranešimu apie sėkmingai redaguota projektą.	taip
TA-13	TS-16 Ištestuoti, ar projekto vadovas prie egzistuojančio projekto gali pridėti roles.	Sukurtas projektas, jo savininkas yra atsidaręs projekto redagavimo puslapį.	<ol style="list-style-type: none"> Spaudžiamas mygtukas naujos rolės sukūrimui; Rolės formoje užpildomi laukai apie rolės pavadinimą, aprašymą ir darbo tipą; Spaudžiamas mygtukas rolės sukūrimui. 	Atvaizduojamas rolės puslapis su informaciniu pranešimu apie sėkmingai sukurtą rolę. Rolių sąrašas atsiranda naujos rolės pavadinimas.	taip
TA-14	TS-19 Ištestuoti, ar darbuotojas gali kandidatuoti į projekto rolę.	Prie sistemos prisijungęs naudotojas su role „Darbuotojas“. Yra sukurtas projektas su bent viena role. Naudotojas yra pagrindiniame puslapyje.	<ol style="list-style-type: none"> Pagrindiniame puslapyje iš sąrašo yra išsirenkamas projektas; Paspaudžiamas mygtukas „Daugiau Informacijos“ prie projekto; Atsidariusiame projekte pasirenkama rolė iš sąrašo; Paspaudžiama ant rolės; Paspaudžiamas mygtukas kandidatavimui į rolę. 	Atvaizduojamas pranešimas, jog naudotojas buvo sėkmingai pridėtas į kandidatų sąrašą.	taip

Testavimo atvejo numeris	Testavimo scenarijus	Kas turi būti padaryta prieš vykdant testą	Testavimo eiga	Laukiamas rezultatas	Rezultatas
TA-15	TS-21 Ištestuoti, ar projekto vadovas gali pasirinkti darbuotoją iš kandidatų sąrašo.	Prie sistemos yra prisijungęs naudotojas su role „Projekto vadovas“. Naudotojas yra sukūręs projektą ir jame yra bent viena rolė. Į rolę yra kandidatavęs bent vienas naudotojas. Naudotojas yra to projekto redagavimo puslapyje.	<ol style="list-style-type: none"> 1. Naudotojas atsidaro rolės formą paspaudęs ant rolės pavadinimo, rolių sąrašę; 2. Naudotojas paspaudžia mygtuką „Peržiūrėti kandidatus“. 3. Modaliniame puslapyje pasirenkamas naudotojas iš sąrašo; 4. Prie naudotojo pasirenkamas mygtukas „Pasirinkti“, kuris priima darbuotoją į rolę. 	Atvaizduojamas pranešimas, jog naudotojas buvo sėkmingai priskirtas rolei ir atvaizduojamas naudotojo rolės puslapyje	taip
TA-16	TS-30 Ištestuoti, ar naudotojas yra išregistruotas po 15 minučių nesinaudojant sistema.	Prie sistemos prisijungęs naudotojas. Atidarytas pagrindinis puslapis.	<ol style="list-style-type: none"> 1. 15 minučių nieko nespausti. 	Atvaizduojamas prisijungimo puslapis.	taip

8.3. Testavimo rezultatai

Sistemos testavimui buvo sukurti 32 testavimo scenarijai (10 kritinio, 13 aukšto, 7 vidutinio, 2 mažo svarbumo) ir 16 scenarijų. Visi testavimo scenarijai buvo išbandyti ir realizuoti.

9. Išvados

1. Išanalizavus tris internetines sistemas, skirtas darbuotojų paieškai, norint sukurti informacinių technologijų projektus ar projektų paieškai ieškant darbo, buvo nustatyti pagrindiniai sistemų trūkumai bei silpnosios savybės. Pagrindinis trūkumas egzistuojantis visose trejose sistemose yra nepakankamas informacijos kiekis apie naudotojų anksčiau įgyvendintus projektus.
2. Suformavus sistemos funkcinius ir nefunkcinius reikalavimus buvo apibrėžtos esminės sistemos funkcijos. Realizavus reikalavimus buvo užtikrinta, jog sistema atlieka pagrindines savo funkcijas bei yra patogi naudotis vartotojui.
3. Remiantis aprašytais sistemos reikalavimais įgyvendinta sistemos realizacija, panaudotas šiuolaikinis dizainas ir formų validacija. Taigi, sukurta sistema, lyginant su jos konkurentais, užtikrina, jog projektus ir gyvenimo patirtis bus galima įvesti paprasčiau, kandidatuoti į projekto roles bus lengviau bei naudojimasis sistema bus efektyvesnis ir patogesnis.
4. Ištestavus sistema galima teigti, jog sistema veikia tinkamai ir yra paruošta naudojimui.



10. Literatūra

- Bangare, Sunil, S. L. Bangare, S. Gupta, M. Dalal, and A. Inamdar. 2016. *Using Node.Js to Build High Speed and Scalable Backend Database Server Fabric Fault Detection View Project Brain Tumor Detection from MRI View Project Using Node.Js to Build High Speed and Scalable Backend Database Server*.
- Bradbury, Danny. 2011. "Data Mining with LinkedIn." *Computer Fraud and Security* 2011(10):5–8.
- Cable, Daniel M., and Daniel B. Turban. 2003. "The Value of Organizational Reputation in the Recruitment Context: A Brand-Equity Perspective." *Journal of Applied Social Psychology* 33(11):2244–66.
- Caers, Ralf, and Vanessa Castelyns. 2011. "LinkedIn and Facebook in Belgium." *Social Science Computer Review* 29(4):437–48.
- Cervantes, Humberto, Rick Kazman, Jungwoo Ryoo, Junsung Cho, Geumhwan Cho, Hyoungshick Kim, and Jina Kang. 2019. "Data-Driven Selection of Security Application Frameworks During Architectural Design." in *Proceedings of the 52nd Hawaii International Conference on System Sciences*. Hawaii International Conference on System Sciences.
- Lee, Kyumin, Steve Webb, and Hancheng Ge. 2014. *The Dark Side of Micro-Task Marketplaces: Characterizing Fiverr and Automatically Detecting Crowdturfing*.
- Leona Lambert. 2017. *Mongo DB For You! CreateSpace Independent Publishing Platform*, North Charleston, SC, USA.
- Lin, Kuan Yu, and Hsi Peng Lu. 2011. "Why People Use Social Networking Sites: An Empirical Study Integrating Network Externalities and Motivation Theory." *Computers in Human Behavior* 27(3):1152–61.
- Gander, F., Ruch, W., Platt, T., Hofmann, J., & Elmer, T. (2018). Current and ideal team roles: Relationships to job satisfaction and calling. *Translational Issues in Psychological Science*, 4(3), 277–289.
- Rapanta, Chrysi, and Lorenzo Cantoni. 2017. "The LinkedIn Endorsement Game: Why and How Professionals Attribute Skills to Others." *Business and Professional Communication Quarterly* 80(4):443–59.
- Schmiedehausen, Kim. 2018. *Single Page Application Architecture with Angular*.

- Wilemon, D. L. & Thamhain, H. J. (1983). Team building in project management. *Project Management Quarterly*, 14(2), 73–81.
- Wohlgethan, Eric. 2018. *Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js*.