



Kauno technologijos universitetas

Informatikos fakultetas

Laiko planavimo sistema

Baigiamasis bakalauro studijų projektas

Mantvydas Razulevičius

Projekto autorius

doc. Sigitas Drąsutis

Vadovas

Kaunas, 2020



Kauno technologijos universitetas

Informatikos fakultetas

Laiko planavimo sistema

Baigiamasis bakalauro studijų projektas

Programų sistemos (612I30002)

Mantvydas Razulevičius

Projekto autorius

doc. Sigitas Drąsutis

Vadovas

**Pareigų sutrumpinimas Vardenis
Pavardenis**

Recenzentas / Recenzentė

Kaunas, 2020



Kauno technologijos universitetas

Informatikos fakultetas

Mantvydas Razulevičius

Laiko planavimo sistema

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Mantvydo Razulevičius, baigiamasis projektas tema „Laiko planavimo sistema“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Razulevičius, Mantvydas. Laiko planavimo sistema. Bakalauro studijų baigiamasis projektas / vadovas doc. Sigitas Drąsutis; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Programų sistemos.

Reikšminiai žodžiai: API, laiko planavimas, internetinis puslapis,

Kaunas, 2020. 50 p.

Santrauka

Darbe pristatoma laiko planavimo sistema. Įvade pateikiama rinkos analizė, kuri parodo produkto aktualumą. Taip pat aptariami rinkoje esantys konkurentai, jų pranašumai bei trūkumai. Projektavimo dalyje pateikiami reikalavimai sistemai, bei techninei įrangai. Plačiai aprašoma ir būsimos sistemos analizė. Testavimo dalyje pristatomas testavimo planas, aprašomas jo išpildymas. Dokumentacijos dalyje pateikiamas paprastas naudotojo vadovas, padedantis naujam sistemos vartotojui greičiau įgyti žinių reikalingų darbui su sistema. Darbo pabaigoje pateikiami rezultatai ir išvados.

Razulevičius, Mantvydas. Time Scheduling System. Bachelor's Final Degree Project / supervisor assoc. prof. Sigitas Drąsutis; Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer Sciences, Software Systems.

Keywords: API, time planning, web application.

Kaunas, 2020. 50 p.

Summary

A time scheduling system is presented in the thesis. Introduction section shows relevance of the problem. Also this part has comparison table with other systems. Modelling section includes system and technical requirements. It also describes analysis of system. Testing section includes testing plan description and its execution results. Documentation section has simple user guide which helps to get basic knowledge about system and learn to use it. At the end of the paper results and conclusions of the thesis are presented.

Turinys

Lentelių sąrašas.....	8
Paveikslų sąrašas.....	9
Santrumpų ir terminų sąrašas.....	10
Įvadas	11
1. Analizė	12
1.1. Techninis pasiūlymas	12
1.1.1. Sistemos apibrėžimas	12
1.1.2. Bendras veiklos tikslas	12
1.1.3. Sistemos pagrįstumas	12
1.1.4. Konkurencija rinkoje	12
1.1.5. Prototipai ir pagalbinė informacija.....	14
1.1.6. Ištekliai, reikalingi sistemai sukurti	14
1.2. Galimybių analizė	14
1.2.1. Techninės galimybės	14
1.2.2. Vartotojų pasiruošimo analizė	14
2. Projektas.....	15
2.1. Reikalavimų specifikacija.....	15
2.1.1. Komer cinė specifikacija	15
2.1.2. Sistemos funkcijos.....	15
2.1.3. Vartotojo są sąjos specifikacija.....	22
2.1.4. Realizacijai keliami reikalavimai	22
2.1.5. Techninė specifikacija	22
2.2. Projektavimo metodai.....	22
2.2.1. Projektavimo valdymas ir eiga.....	22
2.2.2. Projektavimo technologija	23
2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos	24
2.3. Sistemos projektas.....	24
2.3.1. Statinis sistemos vaizdas	24
2.3.2. Dinaminis sistemos vaizdas	28
3. Testavimas	35
3.1. Testavimo planas.....	35
3.2. Testavimo kriterijai	35
3.3. Komponentų testavimas	35
3.3.1. Automatinis testavimas	35
3.3.2. Rankinis API testavimas.....	37
3.4. Statinė kodo analizė	39
3.5. Vartotojo są sąjos testavimas	40
4. Dokumentacija naudotojui	43
4.1. Apibendrintas sistemos galimybių aprašymas.....	43
4.2. Vartotojo vadovas	43
Rezultatai ir išvados.....	49
Literatūros sąrašas.....	50

Lentelių sąrašas

1.1 lentelė. Konkurentų apžvalga	13
3.1 lentelė. Naujo darbo sukūrimas	37
3.2 lentelė. Visų darbų paėmimas	37
3.3 lentelė. Vieno darbo paėmimas	38
3.4 lentelė. Darbo redagavimas	38
3.5 lentelė. Darbo šalinimas	38
3.6 lentelė. Vartotojo sąsajos testavimas	40

Paveikslų sąrašas

2.1 pav. Sistemos panaudojimo atvejų diagrama	16
2.2 pav. Autentifikavimo posistemė	17
2.3 pav. Darbų posistemė	18
2.4 pav. Nustatymų posistemė	19
2.5 pav. Pakvietimų posistemė	20
2.6 pav. Pranešimų posistemė	21
2.7 pav. Tvarkaraščio posistemė	21
2.8 pav. Iteracinis projektavimo modelis	23
2.9 pav. Sistemos išdėstymo diagrama	25
2.10 pav. Sistemos komponentų diagrama	25
2.11 pav. Sistemos paketų diagrama	26
2.12 pav. Darbų posistemės klasių diagrama	27
2.13 pav. Duomenų bazės schema	28
2.14 pav. Registracijos sekų diagrama	29
2.15 pav. Prisijungimo sekų diagrama	30
2.16 pav. Visų darbų peržiūros sekų diagrama	31
2.17 pav. Darbo sukūrimo sekų diagrama	31
2.18 pav. Darbo redagavimo sekų diagrama	32
2.19 pav. Darbo šalinimo sekų diagrama	33
2.20 pav. Darbo informacijos peržiūros sekų diagrama	33
2.21 pav. Darbų sąrašo atnaujinimo sekų diagrama	34
3.1 pav. Vienetų testų kodo padengimas	36
3.2 pav. Visi atlikti vienetų testai	37
3.3 pav. Swagger grafinė vartotojo sąsaja	39
3.4 pav. Resharper aptiktos klaidos	39
3.5 pav. Perspėjimai dėl kodo kokybės, prieš peržiūrą	40
3.6 pav. Perspėjimai dėl kodo kokybės, po peržiūros	40
4.1 pav. Neprisijungusio vartotojo meniu juosta	43
4.2 pav. Registracijos forma	43
4.3 pav. Prisijungimo prie sistemos forma	44
4.4 pav. Pagrindinis meniu po prisijungimo	44
4.5 pav. Pasirinktos dienos darbų sąrašas	44
4.6 pav. Bendras darbų sąrašas	45
4.7 pav. Naujo darbo pridėjimo bei redagavimo formos	45
4.8 pav. Pakvietimų atlikti darbą puslapis	46
4.9 pav. Darbo informacijos peržiūros puslapis	46
4.10 pav. Pakvietimų sąrašas	46
4.11 pav. Pranešimų sąrašas	47
4.12 pav. Darbo pradžios ir pabaigos laikų nustatymas	47
4.13 pav. Tvarkaraščio atnaujinimo mygtukas	47
4.14 pav. Elementų kiekio puslapyje nustatymas	48
4.15 pav. Paskyros šalinimas	48

Santrumpų ir terminų sąrašas

- API (angl. *Application Programming Interface*) – programos sąsaja, leidžianti pasiekti duomenis bei funkcionalumą.
- Autorizacija – asmens prieigos teisių nustatymo procesas.
- DBVS – duomenų bazių valdymo sistema.
- HTTP (angl. *HyperText Transfer Protocol*) – protokolas, skirtas keistis informacija pasauliniame tinkle.
- JSON (angl. *JavaScript Object Notation*) – duomenų saugojimo ir perdavimo formatas.
- SQL (angl. *Structured Query Language*) – standartinė reliacinių duomenų bazių valdymo kalba.
- UML (angl. *Unified Modeling Language*) – grafinė modeliavimo kalba, skirta objektinių sistemų atvaizdavimui ir dokumentavimui.

Ivadas

Modernioje visuomenėje dėl vis didėjančio gyvenimo tempo tampa vis sunkiau planuoti ir paskirstyti darbus. Atlikti paskirtas užduotis, neatsilikti nuo darbo grafiko padeda išankstinis planavimas. Be tokio planavimo neretai atidedami svarbūs, net būtini darbai, o vėlavimas juos atlikti žmogui sukelia stresą, kenkiantį jo fizinei ir psichologinei būklei, o įmonėms - grėsmę patirti nuostolius.

Sistemos tikslas išspręsti problemas, susijusias su laiko planavimu, paversti jį patogesniu, taip taupant vartotojo laiką. Norint padėti išspręsti šią problemą bus sukurta programinė įranga, kuri būtų patogi vartotojui, bei neperkrauta nereikalingo funkcionalumo. Tai bus pasiekta atlikus šios uždavinius:

1. Atlikti rinkoje esančių sistemų analizę ir padaryti išvadas.
2. Suprojektuoti būsimą sistemą, pasinaudojant analizės metu gauta informacija.
3. Remiantis projektu, realizuoti sistemą.
4. Parengti sukurtos sistemos testavimo planą.
5. Dokumentuoti sukurtą sistemą.

Darbas struktūrą sudaro šios pagrindinės dalys:

1. Analizė

Apžvelgiama į problemą, kuri bus išspręsta sukūrus sistemą. Apžvelgiami rinkoje jau esantys konkurentai, bandoma ieškoti kuriamos sistemos pranašumų.

2. Projektavimas

Šioje dalyje pristatomas projektas, jam iškelti reikalavimai, projektavimo metodika bei kūrimui naudotos technologijos.

3. Testavimas

Šioje dalyje matomas parengtas testavimo planas, pristatomas jo atlikimas ir gauti rezultatai.

4. Dokumentacija

Šioje dalyje pateikiamas vartotojo vadovas.

Sistemos apimtis:

- Bendras kodo eilučių kiekis : ~5700
- Testavimui ~1100
- Back end ~2700
- Front end ~2000
- Puslapių skaičius : 9
- API metodų skaičius: 27

1. Analizė

Šiame skyriuje bus pateikta su darbo problematika susijusios informacijos analizė.

1.1. Techninis pasiūlymas

1.1.1. Sistemos apibrėžimas

Laiko planavimo sistema – tai interneto svetainė padedanti suskurti dienotvarkę ir efektyviai paskirstyti laiką.

1.1.2. Bendras veiklos tikslas

Padėti lengviau susiplanuoti, paskirstyti darbus tiek žmonėms, tiek ir įmonėms. Taip pat vienas iš svarbiausių sistemai iškeltų tikslų yra paprastumas, kad kiekvienas sugebėtų greitai išmokti naudotis sistema.

1.1.3. Sistemos pagrįstumas

Žmonės visada skuba ir laiko taupymas jiems padeda nuveikti daugiau. Tai pagrindinė priežastis, kodėl kuriama sistema yra reikalinga. Nesvarbu, ar esi paprastas vartotojas, kuriam patogiau jau ryte turėti veiklų tvarkaraštį, nes tai padaro jo dieną produktyvesnę, ar žmogus atsakingas už kitų žmonių atliktus darbus, ši sistema sumažins tau problemų planuojant grafiką. Sudėtingas skaičiavimo algoritmas atsižvelgs į tavo nustatytus prioritetus ir taip iškels svarbiausius darbus į priekį.

1.1.4. Konkurencija rinkoje

Ši sritis yra stipriai konkurencinga, kadangi jau yra nemažai šia veikla užsiimančių organizacijų. Bet daugumos sistemų funkcionalumas skirtingas. Toliau bus pateikta keletas pavyzdžių, aptarta po keletą kiekvienos programos funkcijų bei pažymėtos išskirtinės savybės.

- **Appointy**

1. Darbo valandų nustatymas[3].
2. Pasirenkami laiko intervalai[3].
3. Skirtingi tvarkaraščių tipai, turintys skirtingą funkcionalumą[3].

Palaikomas Google, tai leidžia pritraukti daugiau klientų socialiniuose tinkluose[3].

- **Caledly**

1. Integracijos su skirtingomis programomis[4].
2. Susitikimų organizavimas[4].
3. Nustatymai, leidžiantys apsaugoti nuo paskutinės minutės renginių ir panašių nepageidaujamų dalykų[4].
4. Tvarkaraščiai pritaikomi tiek individualiam, tiek komandiniam darbui[4].

Sistema skirta įvairių tvarkaraščių sudarymui[4].

- **Fantastical 2**

1. Keli skirtingi kalendoriai[5].
2. Pranešimų siuntimas[5].
3. Veiklų ir užduočių pridėjimas, šalinimas[5].
4. Pakvietimai[5].

Renginių, darbų organizavimo sistema. Iš pirmo žvilgsnio sistema nesudėtinga, tačiau turi labai daug perteklinio funkcionalumo.

Išskirtinis dalykas, kurį galima būtų panaudoti kuriamoje sistemoje yra susitikimų planavimas[5]. Kviečiami žmonės patvirtina, kad dalyvaus. Tuomet visiems išrenkamas bendras, tinkamas darbo laikas[5].

- **Sling**

1. Darbo pamainų sudarymas[6].
2. Savaitinių darbo grafikų sudarymas[6].
3. Darbuotojų grafikų susikirtimai (konfliktų sprendimas)[6].
4. Šabloninių tvarkaraščių sudarymas, su numatytu valandų ar pamainų kiekiu[6].
5. Darbuotojai grafikus gali rinktis patys[6].

Tai pati panašiausia sistema į mano kuriamą. Pagrindinis tikslas – darbuotojų tvarkaraščio sudarymas[6].

1.1 lentelė. Konkurentų apžvalga

Lyginimo kriterijai	Appointy	Caledly	Fantastical 2	Sling	Mano kuriama sistema
Kaina	0-79.99\$ per mėnesį	0-12\$ per mėnesį	0-4.99\$ per mėnesį	0-4\$ per mėnesį	Nemokama
Naudojimas	Asmeninimas / komandai	Asmeninimas / komandai	Asmeninimas / komandai	Asmeninimas / komandai	Asmeninimas/komanda i
Darbo valandų nustatymas	Taip	Ne	Taip	Ne	Taip
Sinchronizacija su kitomis programomis	Taip	Taip	Taip	Taip	Ne
Darbuotojų žymėjimas spalvomis	Ne	Taip	Taip	Taip	Ne
Vartotojų kiekis	Priklauso nuo kainos, galima pildyti už papildomą mokestį	Priklauso nuo kalendorių skaičiaus ir kainos	Nenurodyta	Priklausomai nuo perkamo plano	Nėra apribojimų
Priminimai	Ne	Ne	Taip	Ne	Taip
Pakvietimai	Iš dalies taip	Ne	Taip	Ne	Taip
Funkcionalumo kiekis	Specializuotas	Specializuotas	Specializuotas, platus	Specializuotas	Specializuotas, taikoma konkrečioms vartotojams, siauras

Buvo pasirinkti kriterijai, kurie atspindi numatomam sistemos vartotojui reikalingą funkcionalumą. Iš šių kriterijų matome, jog kuriama nauja sistema turi ir privalumų prieš jau esančias rinkoje. Pagrindinis jų, nereikalingo funkcionalumo nebuvimas, kaip daugumoje minėtų sistemų.

1.1.5. Prototipai ir pagalbinių informacija

Produktas buvo kuriamas nuo pradžių, jokiais prototipais naudojamas nebuvo.

1.1.6. Išteklių, reikalingi sistemai sukurti

Sistemos kūrimas užtruks apie 2 mėnesius. Dar 1 mėnėsis bus skirtas dokumentacijos rengimui, į kurią įėjo naujos ir esamų sistemų analizė, projektavimas, testavimo plano, bei vartotojų gido parengimas.

Sistema bus kuriama vieno programuotojo, kurio darbo turėtų pakakti atliekant visas užduotis. Visi darbai kartu užtruks apie 300 darbo valandų.

1.2. Galimybių analizė

1.2.1. Techninės galimybės

Sistemai realizuoti techninių kliūčių nėra. .Net Core technologija yra labai plačiai naudojama visame pasaulyje. Tai pat Angular karkasas yra vienas iš populiariausių ir dažniausiai naudojamų Javascript karkasų pasaulyje.

Vienintelė didesnė problema itin sudėtingas algoritmas skirtas darbų išdėstymui pagal laiką. Norint jį patobulinti yra tikimybė jo kodo eilučių skaičius išaugtų kelis kartus ir jo sudėtingumas padidėtų daug kartų. Todėl pasirinkta supaprastinta, tačiau funkcionuojanti jo versija.

Sukurti pilnai veikiančią sistemą per tokį trumpą laiko tarpą yra gan sudėtinga užduotis. Todėl tai tik būsimos sistemos prototipas.

1.2.2. Vartotojų pasiruošimo analizė

Vartotojas gali būti tiek individualus asmuo, tiek ir įmonė. Vartotojų parengimas sistemos naudojimui neturėtų būti sudėtingas. Sistemoje dažniausiai tenka užpildyti įvedimo formas. Tai nėra sudėtinga, kadangi visada pateikiami paaiškinimai kokius duomenis reikia įvesti. Taip pat stengiamasi, jog visi paaiškinimai būtų nesudėtingi ir vartotojui būtų greitai suprantami. Taigi būtinas tik bendras vartotojo pasiruošimas naudotis kompiuteriu, naršykle bei skaityti sistemos siūlomus paaiškinimus.

2. Projektas

Šiame skyriuje aprašoma detali sistemos specifikacija.

2.1. Reikalavimų specifikacija

2.1.1. Komercinė specifikacija

Projektas sukurtas specialiai baigiamajam bakalauriniam darbui. Projekto užsakovas ir vadovas doc. Sigitas Drąsutis. Visus darbus, tiek programavimą, projektavimą, dokumentavimą, tiek ir rinkos analizę atliko studentas Mantvydas Razulevičius.

Biudžetas nėra planuojamas, kadangi tai ne komercinis projektas. Sistemos naudotojai būtų tiek paprasti žmonės visame pasaulyje (reikia mokėti anglų kalbą, nes pradinėje versijoje jų daugiau nėra), tiek ir įmonės.

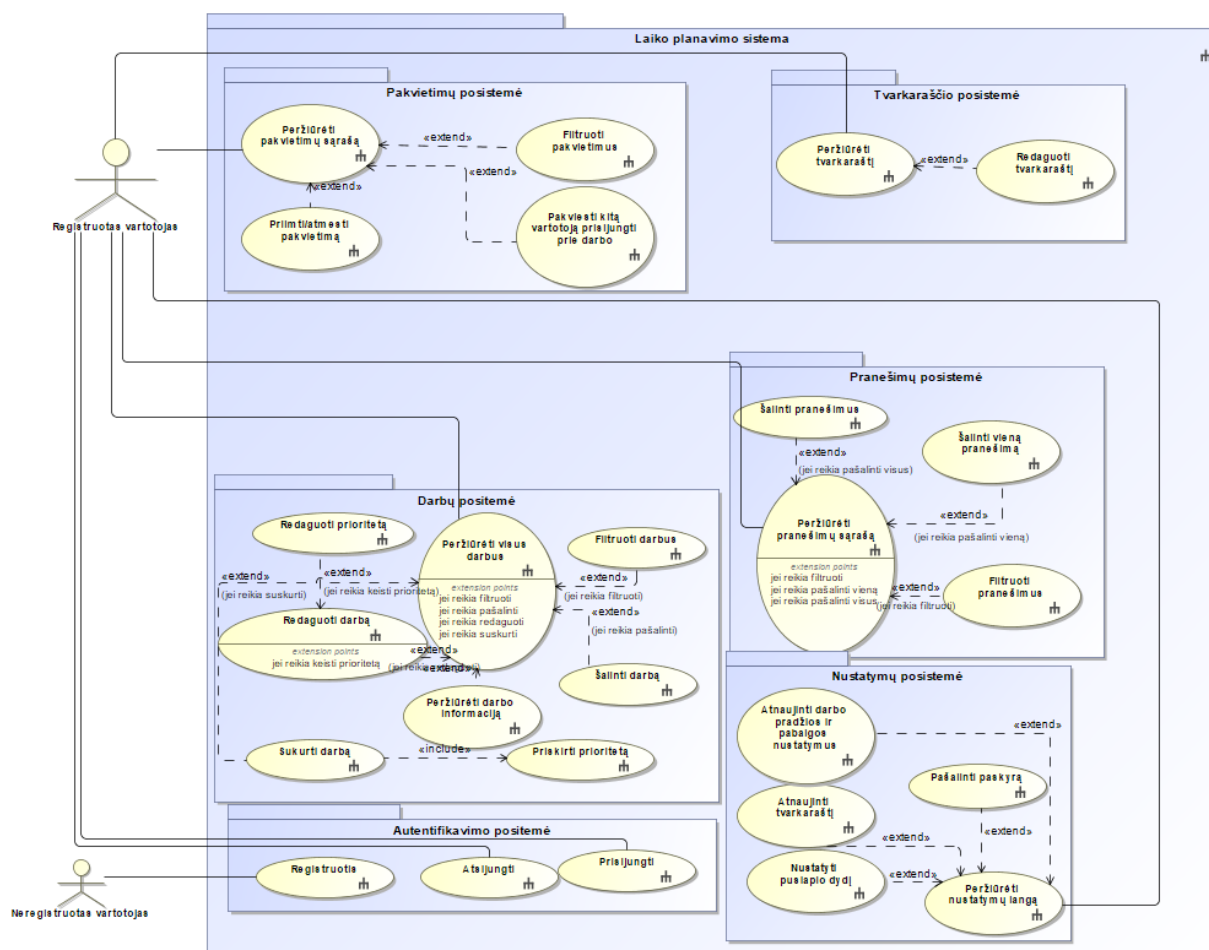
Trukmė pilnam paruošimui (programavimui ir dokumentacijai) apie 3 mėnesius. Pradžia – 2020 sausio 26d. Tiksliai pabaigos data nėra numatyta, tačiau gegužės pradžioje projektas jau turi būti paruoštas pristatymui.

2.1.2. Sistemos funkcijos

Sistemos reikalavimai išskirstomi į šias pagrindines posistemas :

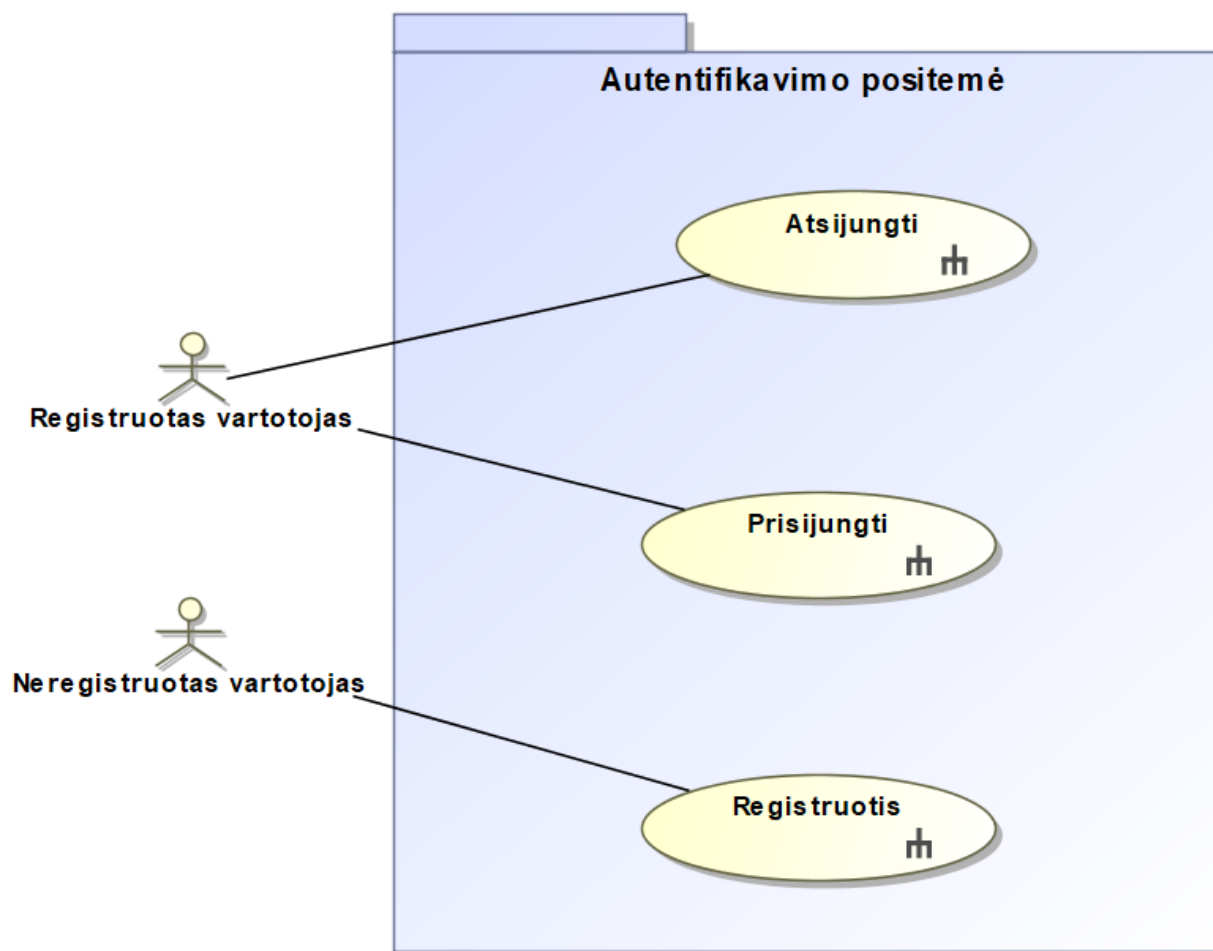
1. Pakvietimų posistemė
2. Autentifikavimo posistemė
3. Tvarkaraščio posistemė
4. Pranešimų posistemė
5. Nustatymų posistemė

Toliau bus pateikta visi funkciniai reikalavimai, bet kiekvienos posistemės analizė bus pateikta atskirai.



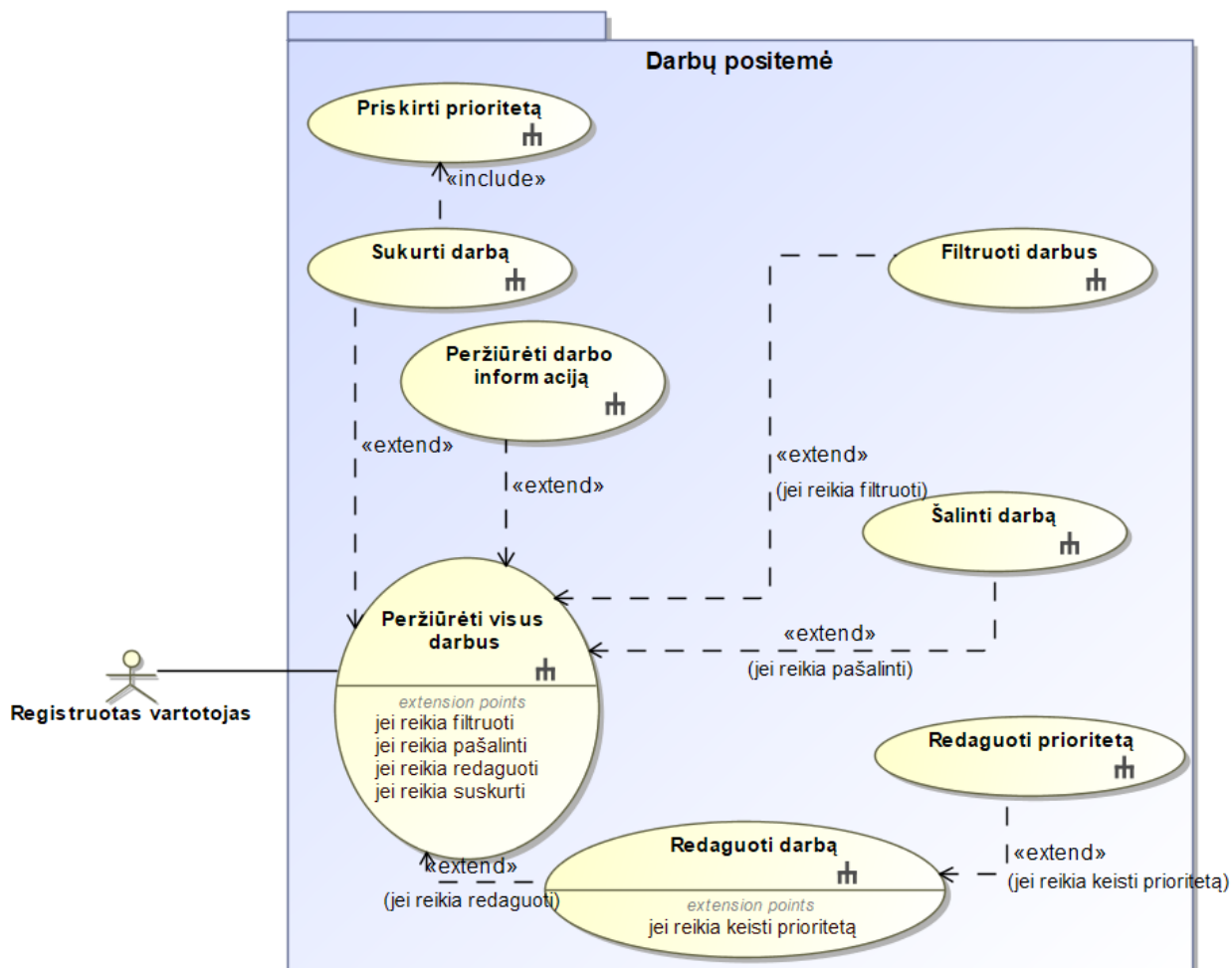
2.1 pav. Sistemos panaudojimo atvejų diagrama

Pradžioje pateikiama diagrama (Žr. 2.1 pav.), kuri parodo sistemos atliekamas funkcijas. Diagramoje aiškiai matyti visos sistemos veiklai būtinos dalys, bendras jų funkcionalumas. Visos posistemės turi susijusio funkcionalumo, pavyzdžiui pranešimų posistemė yra naudojama daugumoje kitų sistemos dalių.



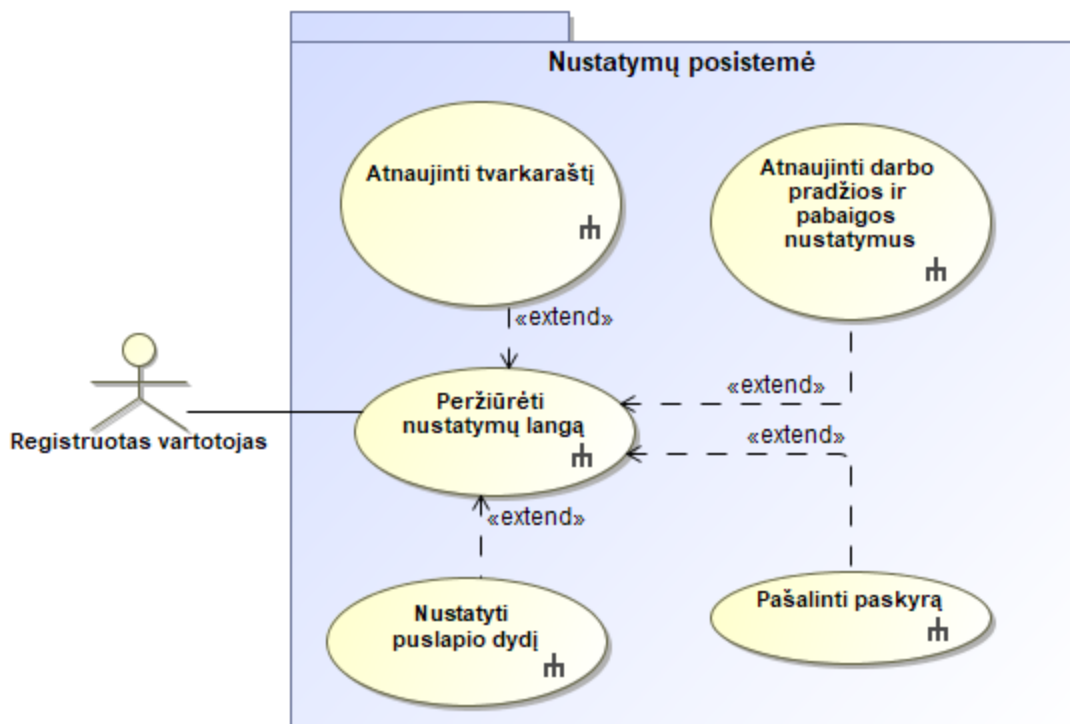
2.2 pav. Autentifikavimo posistemė

Autentifikavimo posistemė (Žr. 2.2 pav.) – viena svarbiausių visoje sistemoje. Ji užtikrina, kad vartotojas matytų tik savo informaciją. Tai vartotojas gali matyti įvykdęs prisijungimą prie sistemos. Kita svarbi funkcija registracija. Jos forma panaši į daugelį šiuo metu internete egzistuojančių registracijos formų. Taip pat svarbu paminėti, jog po atsijungimo vartotojas negali pasiekti savo informacijos iki tol kol vėl prisijungs.



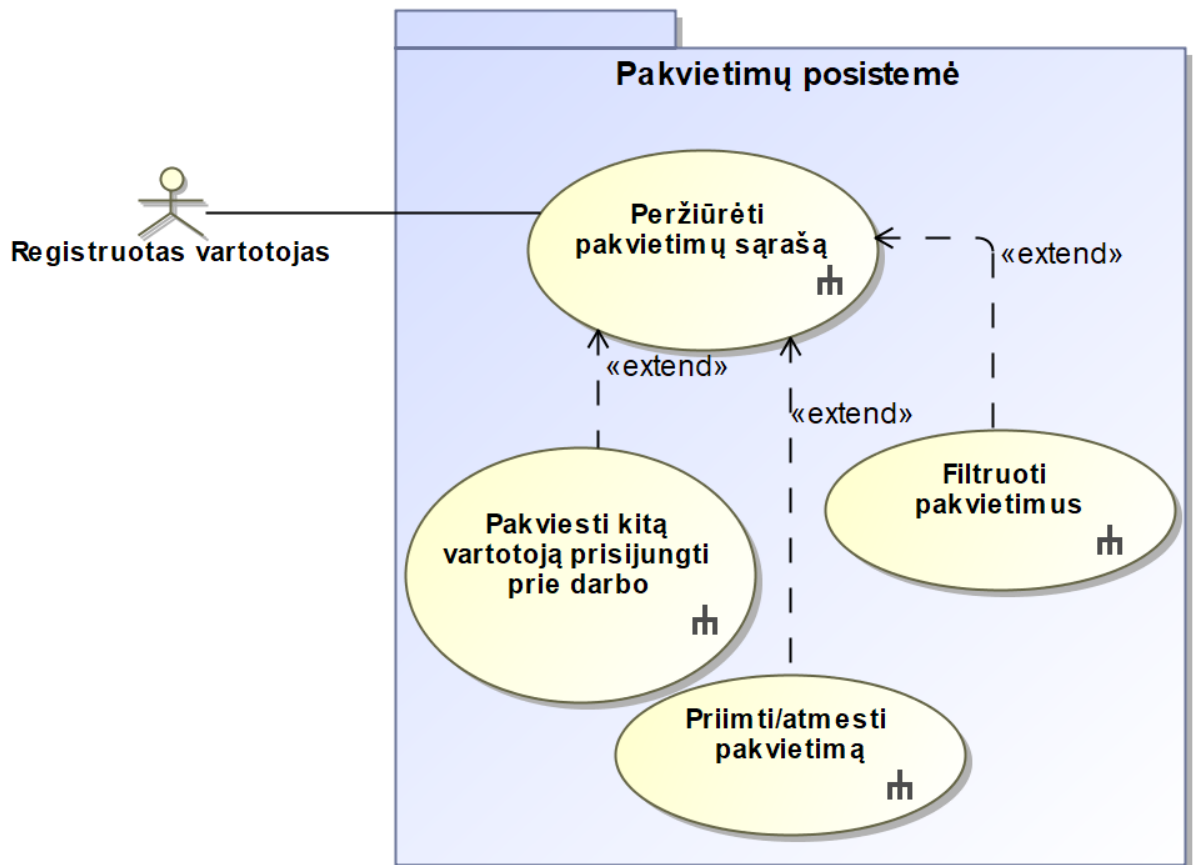
2.3 pav. Darbų posistemė

Darbų posistemė (Žr. 2.3 pav.) – tai dėl ko buvo kuriamas šis projektas. Darbų sąrašas, naujų užduočių pridėjimas ir šalinimas. Kurdamas naują darbą vartotojas turi galimybę jį prioretizuoti. Redagavimas atliekamas iš sąrašo pasirinkus vieną darbą ir pakeitus jo informaciją arba prioritetą. Informacijos peržiūra atliekama prieš jį redaguojant, arba gavus pakvietimą atlikti darbą.



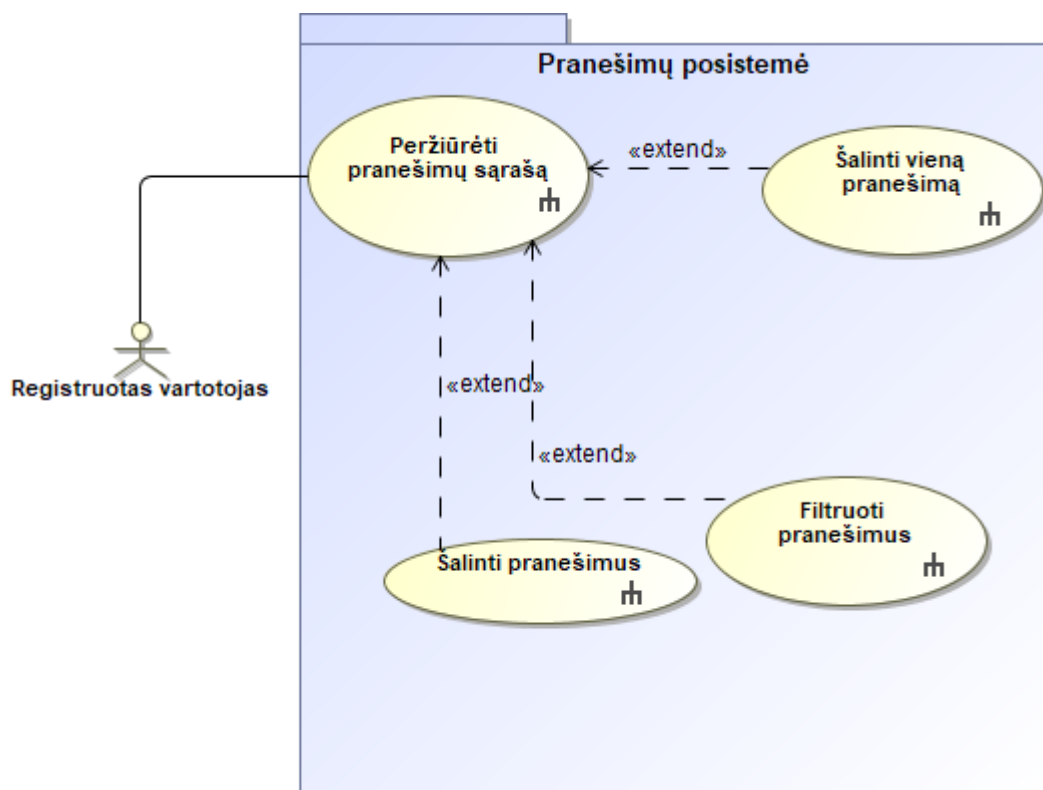
2.4 pav. Nustatymų posistemė

Nustatymai (Žr. 2.4 pav.) yra svarbi sistemos sudedamoji dalis, kadangi tai leidžia vartotojui patogiau jaustis sistemoje. Pavyzdžiui, kai nustatomas puslapio dydis 10 elementų, tokiu atveju visoje sistemoje kur yra pasiskirstymas puslapiai bus atsižvelgta į prieš tai nustatytą reikšmę. Darbo pradžios ir pabaigos nustatymai leis atsižvelgti į vartotojo norus ir apskaičiuoti darbams tinkamas vietas pagal šiuos laiko režius. Jei vartotojas mano, jog paskyra jam nereikalinga, jam suteikiama teisė ją pašalinti.



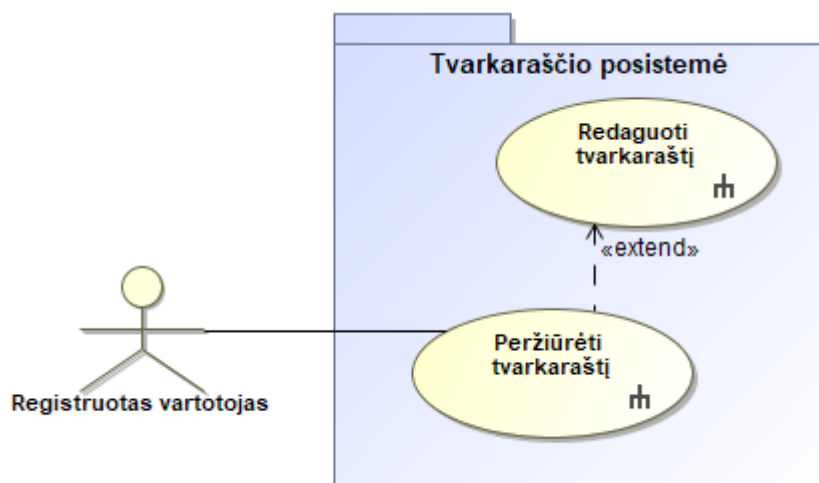
2.5 pav. Pakvietimų posistemė

Pakvietimų posistemė skirta paprasčiau dalintis darbais. Vartotojui galima išsiųsti pakvietimą kitam sistemoje registruotam asmeniui. Kvietimą gavęs asmuo gali jį priimti arba atmesti. Kaip ir visuose puslapiuose, čia turėtų būti filtras skirtas palengvinti reikalingų duomenų suradimą.



2.6 pav. Pranešimų posistemė

Pranešimų posistemė (Žr. 2.6 pav.) – skirta vartotojo informavimui apie tam tikrus įvykius. Pagrindinis funkcionalumas yra atliekamas sistemos, kuri kuria pranešimus ir perduoda juos vartotojui. Vartotojas čia neturi daug funkcijų, tačiau gali peržiūrėti ir pašalinti pranešimus. Taip pat kaip ir daugelyje posistemų yra pranešimų filtravimo galimybė.



2.7 pav. Tvarkaraščio posistemė

Tvarkaraščio posistemė (Žr. 2.7 pav.) – skirta vartotojo tvarkaraščio peržiūrai ir redagavimui. Vartotojas gali pakeisti darbų pozicijas perkedamas juos į kitą vietą ir sistema perskaičiuos naujus pradžios ir pabaigos laikus.

2.1.3. Vartotojo sąsajos specifikacija

Vartotojo sąsajai reikalavimai sudaryti nebuvo.

2.1.4. Realizacijai keliami reikalavimai

- Tik autentifikuoti vartotojai gali naudotis sistema.
- Vartotojai turi gauti tik savo informaciją. t. y. kitų vartotojų informacija nėra pasiekama (išimtis: jei kitas vartotojas atsiunčia pakvietimą).
- Dizaino prisitaikymas mobiliems įrenginiams.
- Vartotojo sąsaja turi būti paprasta, vartotojai turi greitai išmokyti ja naudotis.

2.1.5. Techninė specifikacija

Norint savo kompiuteryje pasileisti šį projektą, reikalinga programinė įranga:

- Windows operacinė sistema.
- Visual Studio 2019
- Microsoft Sql Server Management Studio DBVS.

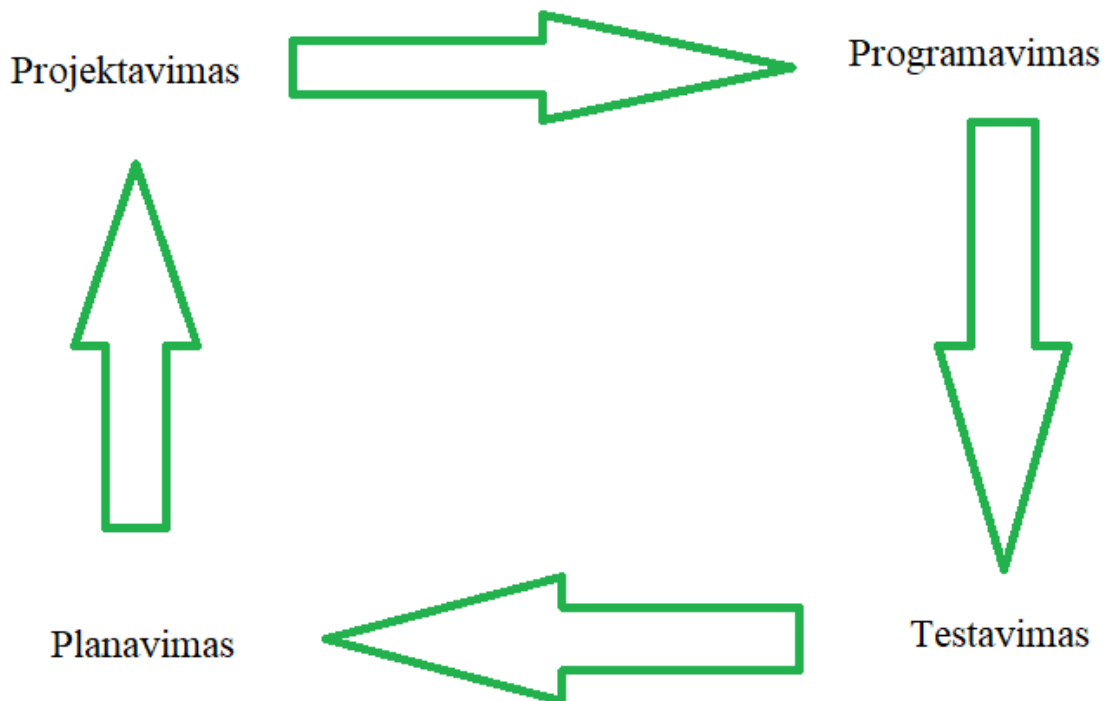
Taip pat reiktų nepamiršti būti įsirašius Angular 8. To pilnai pakanka norint dirbti su šiuo projektu.

2.2. Projektavimo metodai

2.2.1. Projektavimo valdymas ir eiga

Projektas buvo kuriamas naudojant iteracinį projektavimo modelį (Žr. 2.8 pav.).

Iteracinis projektavimas



2.8 pav. Iteracinis projektavimo modelis

Visas atliktas darbas skirstomas į sprintus. Kiekvieno jų ilgis buvo po 2 savaites. Iš viso atlikti 6 sprintai. 2 paskutiniai šiek tiek neatitiko iteracinio projektavimo modulio, kadangi jie paskirti tik dokumentacijos rengimui bei jau sukurtos sistemos testavimui, bandant visus įprastus atvejus bei ieškant galimų sistemos spragų.

Iteracijos pagal užduotis:

1. Pasiruošimas, reikalingos programinės įrangos diegimas, pavyzdinio projekto, duomenų bazės sukūrimas.
2. Darbų, pakvietimų posistemė, jų testavimas
3. Pranešimų, nustatymų posistemė, jų testavimas.
4. Tvarkaraščio posistemė, jos testavimas.
5. Testavimas, dokumentacija, kritinių klaidų paieška sistemoje.
6. Testavimas, dokumentacija, kritinių klaidų paieška sistemoje.

2.2.2. Projektavimo technologija

Modeliavimui naudotas įrankis Magic Draw, kuris palaiko UML modeliavimo kalbą. Buvo panaudota standartinių elementų kaip klasės, komponentai ir kita. Iš jų sudėta keletas dažniausiai naudojamų diagramų pavyzdžiui klasių, paketų, komponentų, diegimo, būsenų bei sekų.

2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos

Sistema sukurta C# programavimo kalba. Programa sukompiliuojama ir jos paketai vykdomi .Net Core 3.1 aplinkoje. Sistema kurta Windows aplinkoje, tačiau naudojant .Net Core veiktų ir Linux ar MacOS aplinkose.

Programavimui naudota Windows10 aplinka, bei Visual Studio 2019 programavimo aplinką.

DBVS – Microsoft Sql Server Management Studio 2019.

Vartotojo sąsajai programuoti naudota Angular 8 karkasas, kuris yra paremtas TypeScript programavimo kalba, kartu su HTML5 žymėjimo ir CCS3 stilių aprašymo kalbomis.

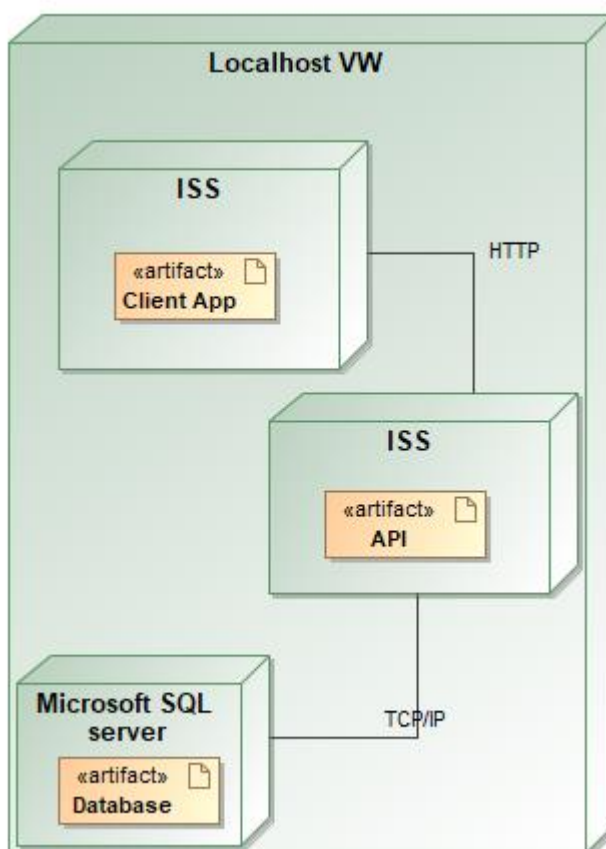
Testavimu naudota Xuint testavimo įrankių paketas.

2.3. Sistemos projektas

2.3.1. Statinis sistemos vaizdas

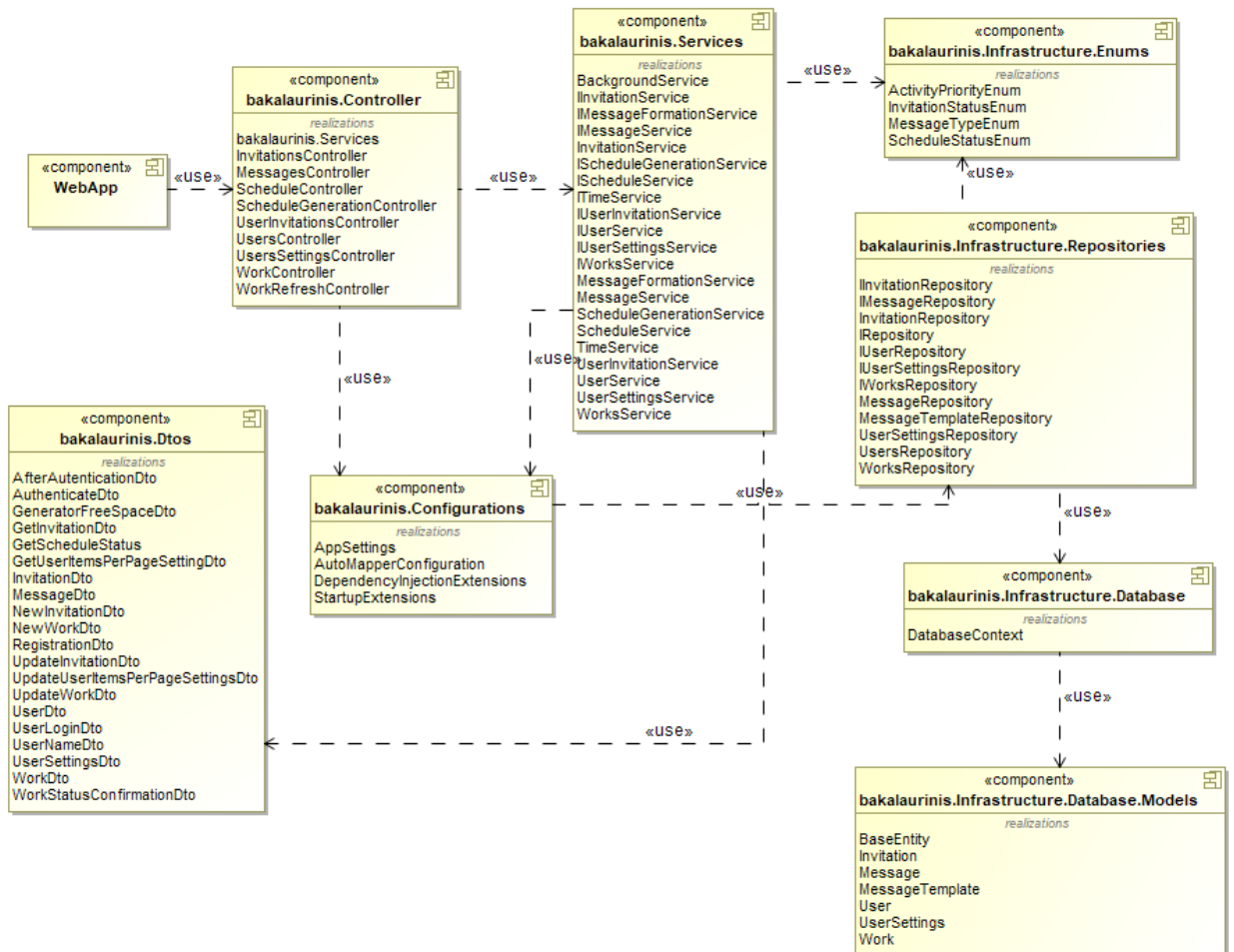
Statinis sistemos vaizdas pateikiamas UML diagramomis. Šiame skyriuje bus pateiktos šios diagramos.

1. Išdėstymo (Žr. 2.9 pav.)
2. Komponentų (Žr. 2.10 pav.)
3. Paketų (Žr. 2.11 pav.)
4. Klasų (Žr. 2.12 pav.)
5. Duomenų bazės (Žr. 2.13 pav.)



2.9 pav. Sistemos išdėstymo diagrama

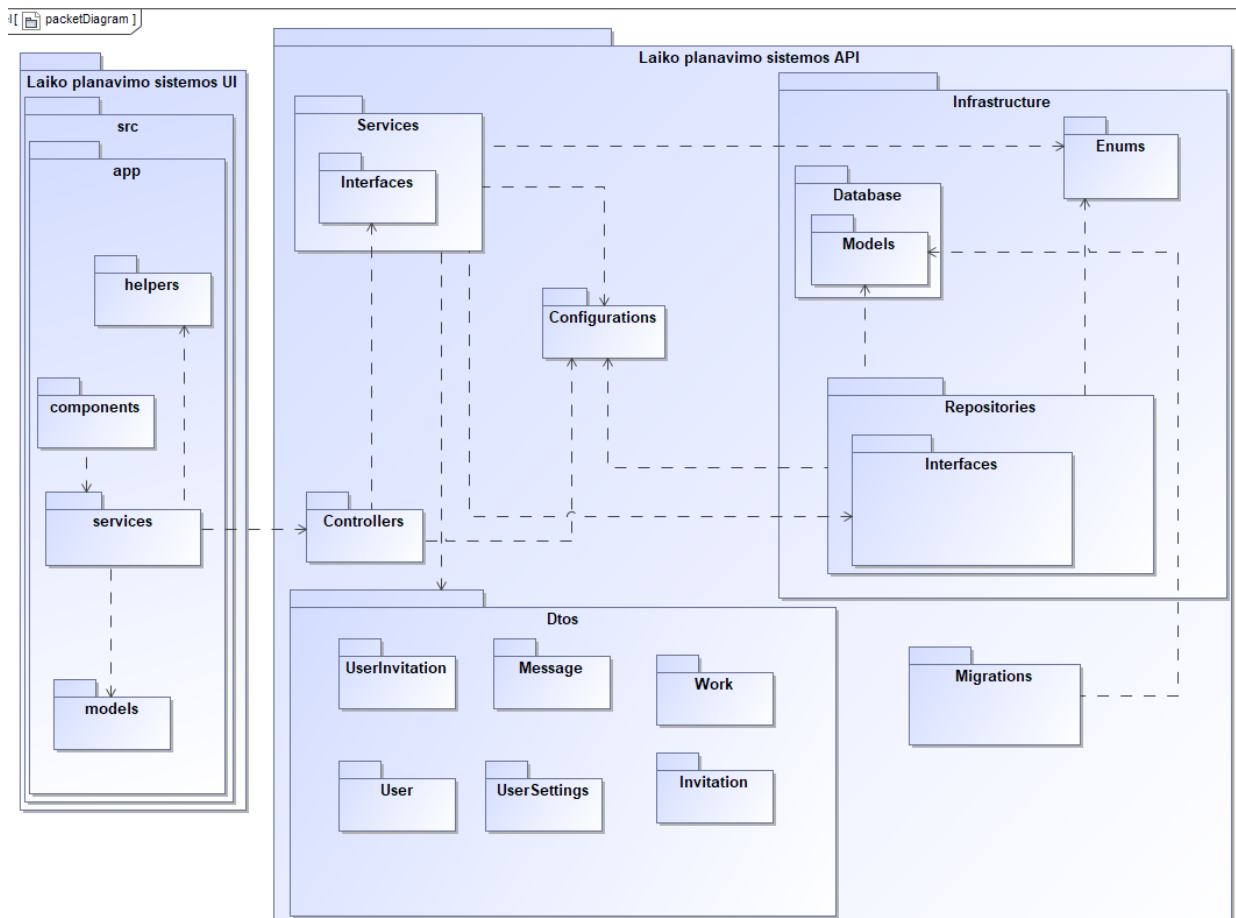
Šiuo metu viskas veikia tik lokaliai, tai yra viename įrenginyje. Paleidžiami du procesai. Jie tarpusavyje bendrauja naudodami HTTP protokolą. Tuo metu duomenys iš serverio imami naudojant TCP/IP protokolą.



2.10 pav. Sistemos komponentų diagrama

Sistemos komponentų diagrama (Žr. 2.10 pav.) – pavaizduoja komponentų tarpusavio sąsajas.

Platesni kiekvienos sistemos dalies paskirties paaiškinimai bus pateikta toliau.



2.11 pav. Sistemos paketų diagrama

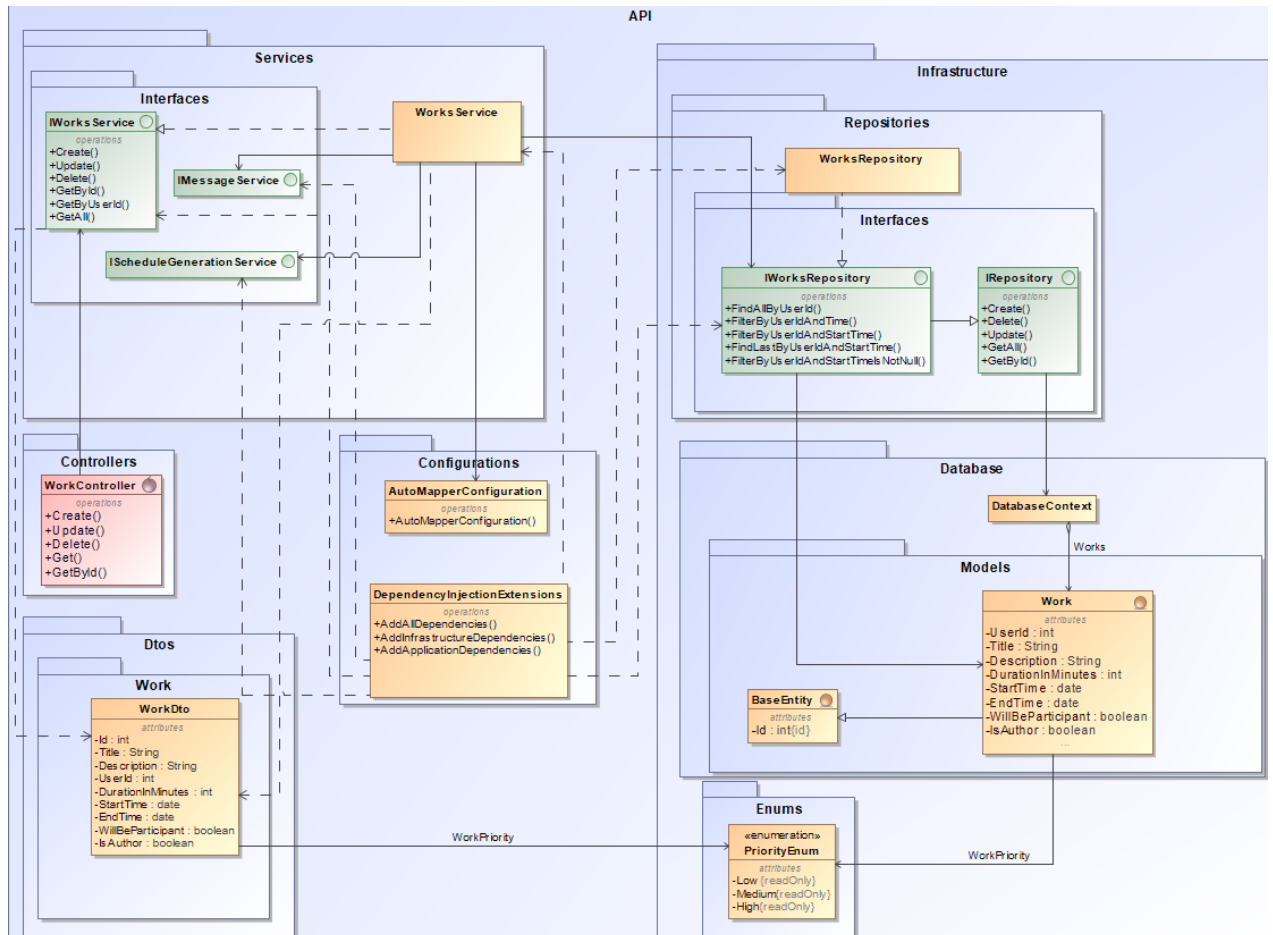
Laiko planavimo sistemos paketų diagrama (Žr. 2.11 pav.) parodo, kad pati sistema sudaryta iš dviejų dalių. Tai yra API skirtas duomenų apdorojimui ir perdavimui, bei UI (angl. – *user interface*) vartotojo sąsaja su visu reikalingu funkcionalumu. Duomenys atiduodami naudojantis įvairias valdikliais siunčiant HTTP užklausą.

API detalizavimas:

- Dtos (angl. *data transfer object*) – skirti duomenų perdavimui tarp servisų ir tarp servisų – valiklių. Taip pat jie perduodami kartu su HTTP užklausomis.
- *Controllers* – valdikliai, skirti duomenų perdavimui ir surinkimui.
- *Services* – dalis atsakinga už duomenų transformavimą ir reikalingą formą, skaičiavimų atlikimą.
- *Repositories* – atsakingos už duomenų surinkimą iš duomenų bazės.
- *Database(Models)* – modelis atitinka duomenų bazės lentelę.
- *Migrations* – tai naudojant sugeneruojamas kodas, iš kurio bus sukurta duomenų bazė. Kodo generavimas vyksta naudojant esamus modelius.

UI detalizavimas:

- *Models* – struktūra atitinka Dtos struktūrą, tačiau naudojami norint atvaizduoti duomenis vartotojui arba gražinti pakeistus duomenis į valdiklius
- *Services* – duomenų, surenka duomenis jos perduoda į komponentus
- *Helpers* – pagalbinės funkcijos susijusios su vartotojų autentifikavimu
- *Components* – apjungiami dizainais su duomenimis, informacija atvaizduojama vartotojui

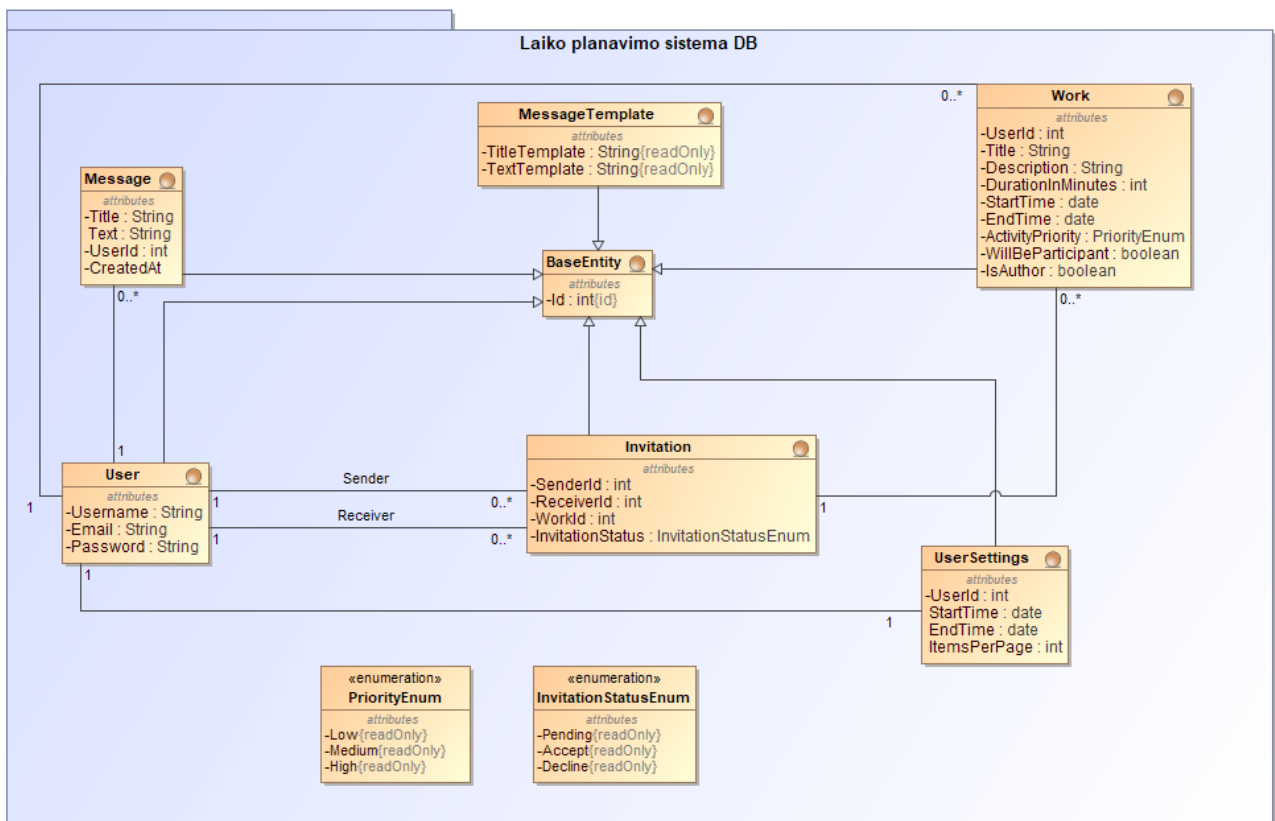


2.12 pav. Darbų posistemės klasių diagrama

Ši klasių diagrama (Žr. 2.12 pav.) atitinka dalies sistemos vaizdą, kitos dalys yra labai panašios. Tačiau tai yra puikus pavyzdys kaip duomenys iš duomenų bazės pasiekia valdiklį ir vėliau vartotoją. Tokiam atvaizdavimui pasirinkta darbų posistemė.

Taigi klasė *Work* yra darbo informaciją apibūdinantis modelis. Ji paveldi klasę *BaseEntity* (plačiau apie tai bus kitoje dalyje). Iš *DatabaseContext* klasės duomenys paimami naudoja *WorksRepository*, kuri paveldi interfaisei *IWorksRepository* ir įgyvendina jame aprašytą funkcionalumą. Tada *Work* modelis perduodamas į servisą *WorkService*, kuris paveldi interfaisei *IWorkService*, taip pat du kitus servusus su savo funkcionalumu, kuris šiuo atveju nėra svarbus, bet turi įtakos kitoms posistemėms. Toliau šis servisas per *Dto*, šiuo konkrečiu atveju *WorkDto* į valdiklį.

Dar vienas svarbus dalykas *Configurations* paketas. Jo klasės turi labai svarbias funkcijas. *AutoMapperConfiguration* padeda paversti vieno tipo objektą į kito tipo, taip supaprastinant kodą ir padidinant jo skaitomumą, o *DependencyInjectionExtensions* leidžia vietoj klasės naudoti jos interfaisei.



2.13 pav. Duomenų bazės schema

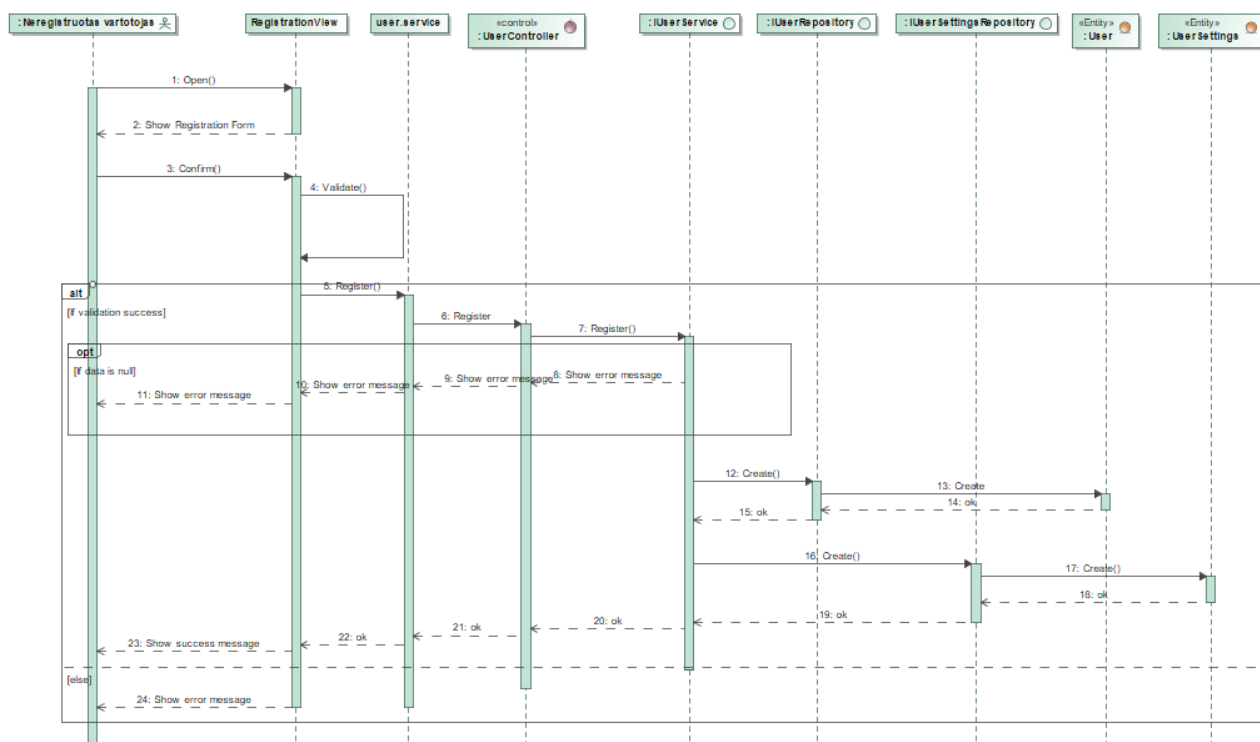
Duomenų bazė yra sudaryta iš 6 lentelių (Žr. 2.13 pav.), tai pat dviejų išvardijimų. Lentelė *User* saugo informacija apie visus užsiregistravusius vartotojus. Čia saugomi visi identifikaciniai duomenys, juos naudojant galima pasiekti gerokai platesnės asmeninės informacijos kaip pavyzdžiui darbus, kurie saugomi *Work* lentelėje. Tai yra svarbiausia lentelė, kuri saugo visą kiekvieno darbo informaciją. Dar viena ne mažiau svarbi lentelė yra *UserSettings*. Jos tikslą atspindi ir pavadinimas, saugoti visus vartotojų nustatymus. Kiekvienas vartotojas turi tik vieną įrašą nustatymų lentelėje. Paskutinė lentelė – *Message* sauganti visus vartotojo gautus pranešimus. Pranešimų šablonai imami iš lentelės *MessageTemplate*, papildomi reikalinga informacija ir sudedami i *Message* ir taip vartotojas gali juos pamatyti.

Taip pat pateiktos dvi išvardinamojo duomenų tipo klasės, kurios saugomos jau pačiame kode. Tai yra *PriorityEnum* ir *InvitationStatusEnum*.

Visos klasės atitinkančios duomenų bazės lenteles paveldi klasę *BaseEntity*. Ši klasė turi tik vieną kintamąjį *Id*. *Id* galima laikyti įrašo numeriu. Tai reiškia, kad kiekvienoje lentelėje bus įrašo numerio stulpelis.

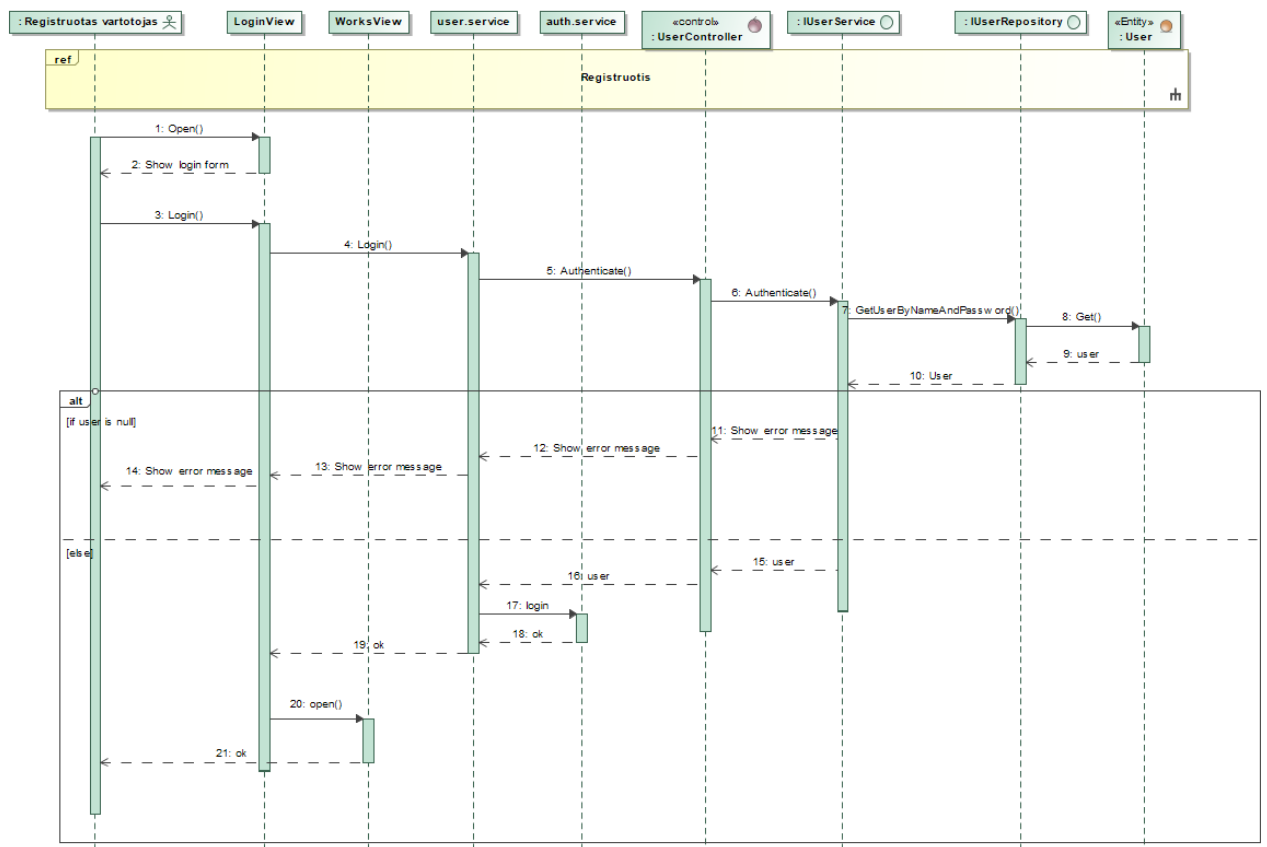
2.3.2. Dinaminis sistemos vaizdas

Dinaminiam sistemos vaizdui pateikti pasirinkta darbų posistemė, taip pat kelios svarbesnės kitų posistemų sekų diagramos, tokios kaip registracijos (Žr. 2.14 pav.), prisijungimo (Žr. 2.15 pav.).



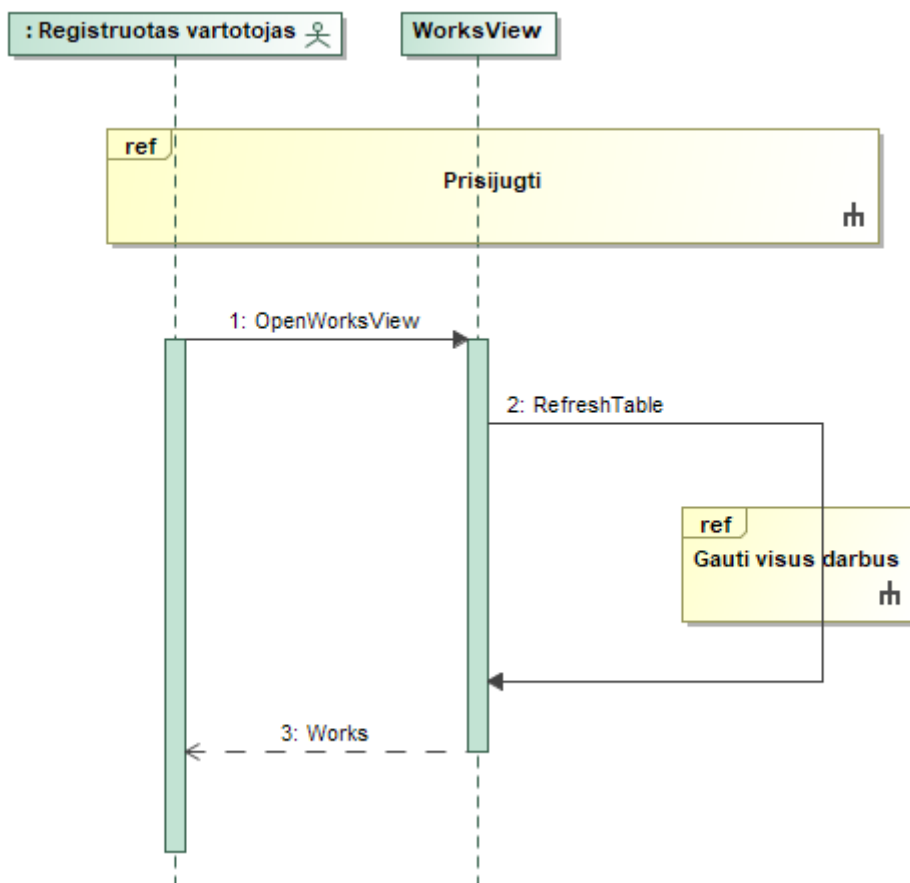
2.14 pav. Registracijos sekų diagrama

Registracijos sekų diagrama (Žr. 2.14 pav.) parodo kaip vyksta nauja vartotojo registravimas. Neregistruotas vartotojas atidaro registracijos puslapį ir užpildo registracijos formą. Tada patvirtinus registracija patikrinami duomenys. Atsiradus klaidai/klaidoms parodomas pranešimas. Jei klaidų nėra duomenys siunčiami į *back end* ir ten patikrinama ar toks vartotojas dar nebuvo sukurtas. Jei vartotojas jau buvo sukurtas parodomas pranešimas, priešingu atveju sukuriamas naujas vartotojas ir pranešama, jog registracija sėkminga.



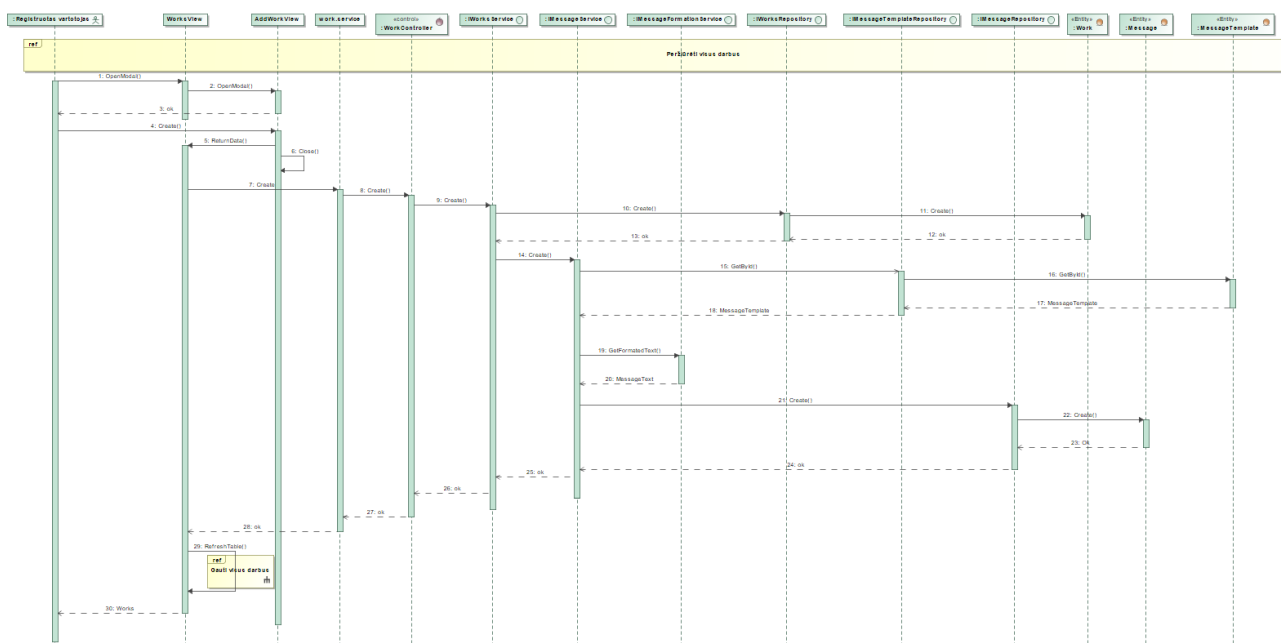
2.15 pav. Prisijungimo sekų diagrama

Norint prisijungti prie sistemos (Žr. 2.15 pav.) yra būtina prieš tai atlikti registraciją (Žr. 2.14 pav.). Įvedamas vartotojo vardas bei slaptažodis. Jei toks vartotojas egzistuoja sistemoje, bei slaptažodis yra tinkamas įvyksta prijungimas prie sistemos. Kitu atveju rodomas klaidos pranešimas.



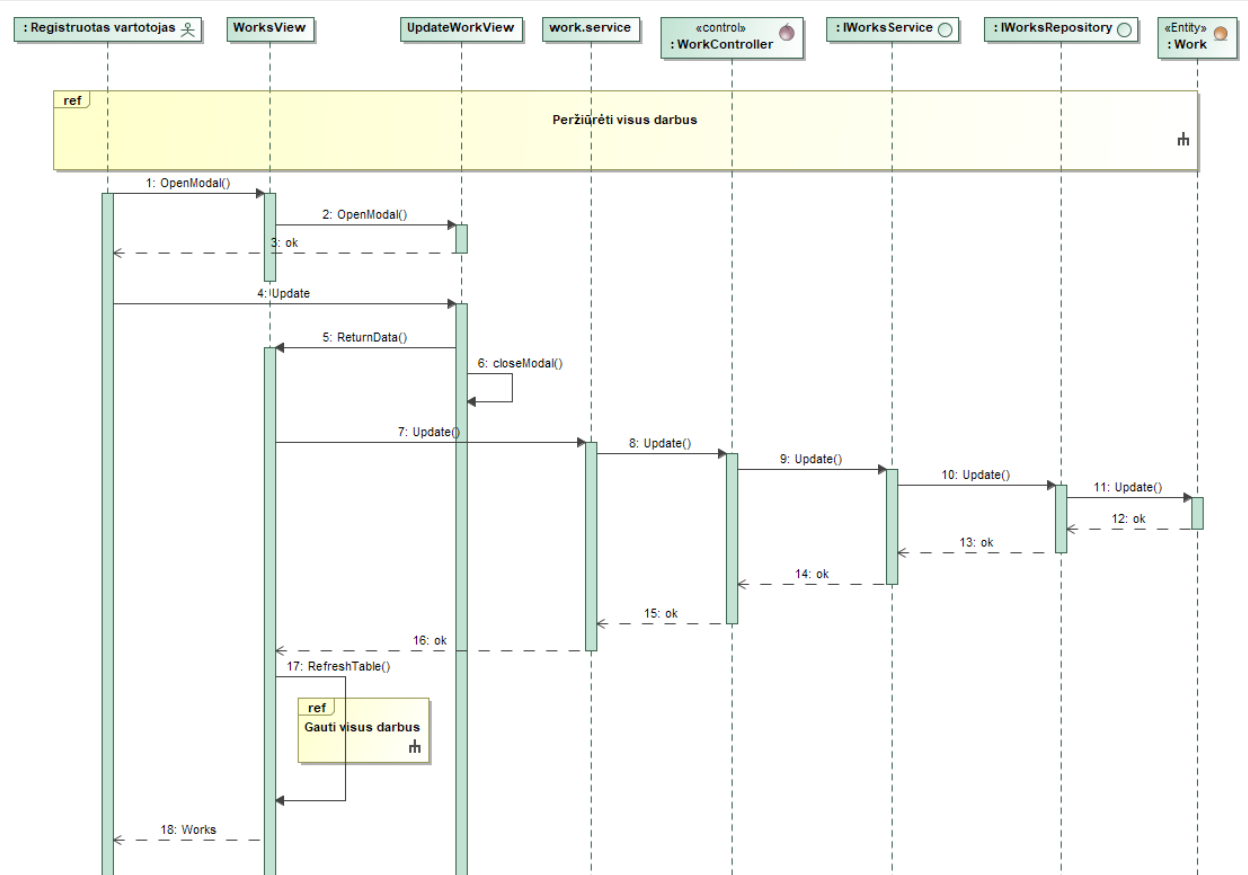
2.16 pav. Visų darbų peržiūros sekų diagrama

Prieš atlikdamas darbų peržiūrą (Žr. 2.16 pav.) vartotojas turi būti prisijungęs prie sistemos (Žr. 2.15 pav.). Jis atsidaro darbų peržiūros langą. Tada sistema paima visus šio vartotojo darbus (Žr. 2.21 pav.) esančius duomenų bazėje ir parodo vartotojui darbų peržiūros lange.



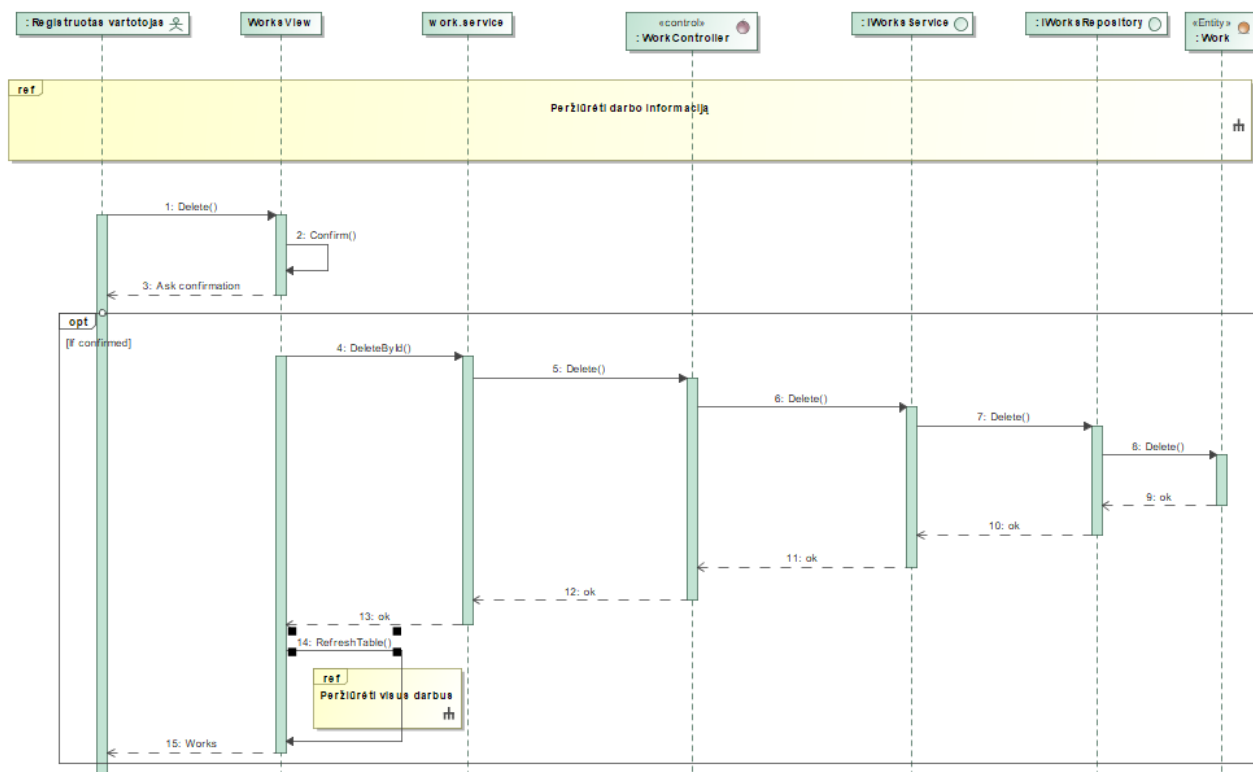
2.17 pav. Darbo sukūrimo sekų diagrama

Kuriant naują darbą (Žr. 2.17 pav.) atidaro darbų sukūrimo langą, jame užpildo duomenis ir paspaudžia išsaugojimo mygtuką. Sistema uždaro darbo kūrimo langą bei siunčia duomenis iki repozitorijos. Išsaugojus darbą, surandamas atitinkamas žinutės šablonas, jis pertvarkomas į žinutę pagal duotas taisykles ir gauta žinutė išsaugoma. Tada gražinamas atsakymas, jog viskas pavyko, atnaujinamas darbų sąrašas (Žr. 2.21 pav.) ir parodomas vartotojui.



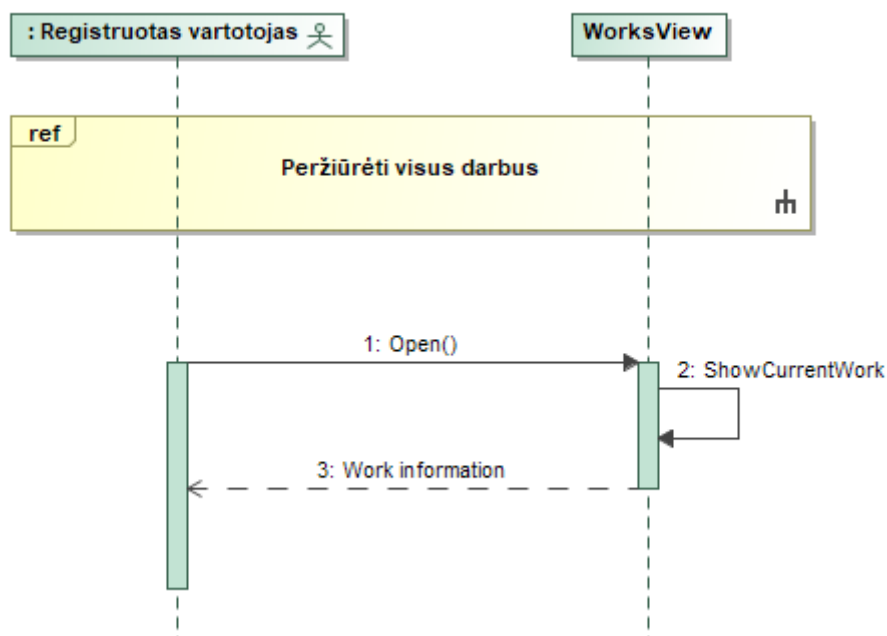
2.18 pav. Darbo redagavimo sekų diagrama

Darbo redagavimas (Žr. 2.18 pav.) atliekamas vartotojui atsidarius tam skirtą formą. Tada išsaugojus naują informaciją, ji siunčiama ir įrašoma į duomenų bazę. Kai gaunamas teigiamas atsakymas atnaujinamas darbų sąrašas ir vartotojas mato pasikeitusią informaciją.



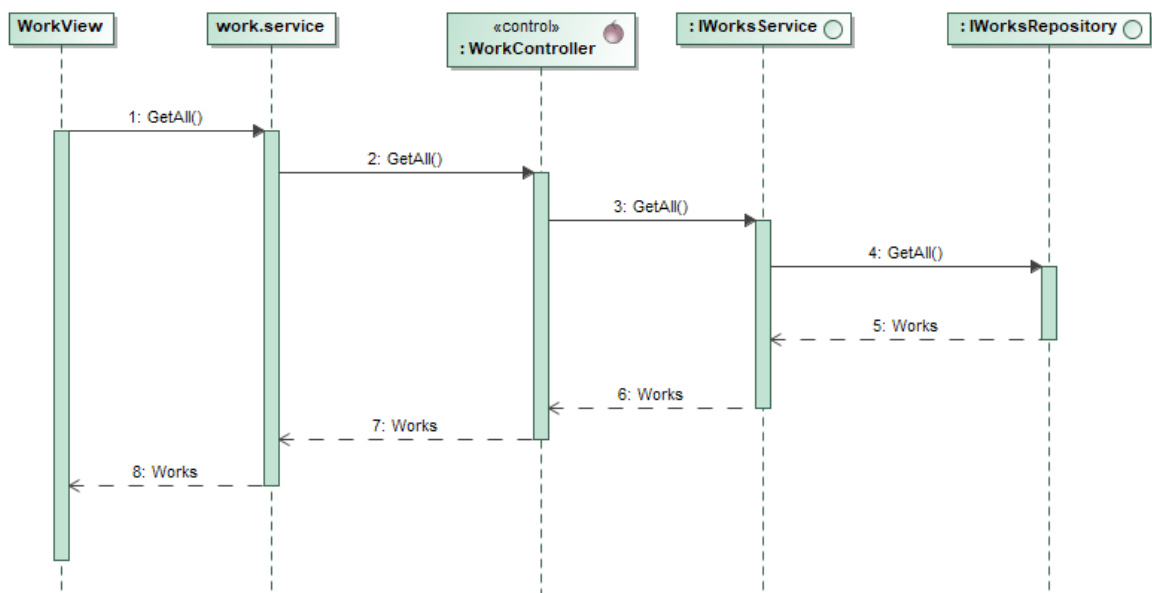
2.19 pav. Darbo šalinimo sekų diagrama

Norėdamas pašalinti darbą (Žr. 2.19 pav.) reikia darbų sąrašą (Žr. 2.16 pav.) paspausti ištrynimo mygtuką. Atsiradus patvirtinimo langui, šalinimas patvirtinamas ir tada atnaujinamas darbų sąrašas (Žr. 2.21 pav.).



2.20 pav. Darbo informacijos peržiūros sekų diagrama

Norėdamas peržiūrėti konkretaus darbo informaciją (Žr. 2.20 pav.) vartotojas turi būti atsidaręs visų darbų sąrašą (Žr. 2.16 pav.). Jame paspaudęs pasirinktą darbą pamatys išplėstinę jo informaciją.



2.21 pav. Darbų sąrašo atnaujinimo sekų diagrama

Darbų sąrašų atnaujinimo diagrama (Žr. 2.21 pav.) naudojama kitose diagramose, tačiau tai nėra atskiras panaudojimo atvejis. Taip yra todėl, kad kiekvieną kartą darant naujo darbo pridėjimą (Žr. 2.17 pav.), redagavimą (Žr. 2.18 pav.) ar šalinimą (Žr. 2.19 pav.) visi duomenys rodomi vartotojui turi būti atnaujinti.

Kaip jau buvo paminėta tai tik kelios sekų diagramos. Nenorint labai išsiplėsti, jų visų pavaizduota nebus. Taip pat dauguma jų yra labai panašios ar turi nemažai vienodų dalių.

3. Testavimas

Skyrius skirtas testavimo plano sudarymui, bei jo įgyvenimui. Bus pateiktos naudojamos technologijos bei aprašyti gauti rezultatus.

3.1. Testavimo planas

Sistemai būtinas testavimas įsitikinti, kad jo veikimas atitinka projektavimo metu nustatytus tikslus. Kad tikslus įgyvendinti buvo sudarytas toks testavimo planas:

1. Atlikti kiekvieno serviso metodo vienetų testus.
2. Atlikti API metodų testus naudojant Swagger (Žr. 3.3 pav.) įrankį.
3. Atlikti statinę kodo analizę, nepaliekant kodo dalių kurios ateityje gali turėti įtakos klaidos atsiradimui.
4. Atlikti svetainės grafinės vartotojo sąsajos testavimą rankiniu būdu.

3.2. Testavimo kriterijai

Kad testavimas būtų sėkmingai įvykdytas, turi būti įgyvendinti šie kriterijai:



















1. Servisų kodo padengimas virš 90%.
2. API testavimas visada gražina rezultatus kurių tikėtasi.
3. Statinė kodo analizė negali turėti nei vieno klaidos pranešimo, taip pat kuo mažiau perspėjimų.
4. Visi sistemos funkcijos turi būti atliekamos taip, kaip nurodyta specifikacijoje.

3.3. Komponentų testavimas

3.3.1. Automatinis testavimas

Komponentų testavimas bus naudojamas tik servisams, nes juose naudodami visi kiti metodai. Taip jie bus ištestuojami kartu. Kontrolerių testuoti vienetų testais nėra prasmės, jie tik kviečia servisu.

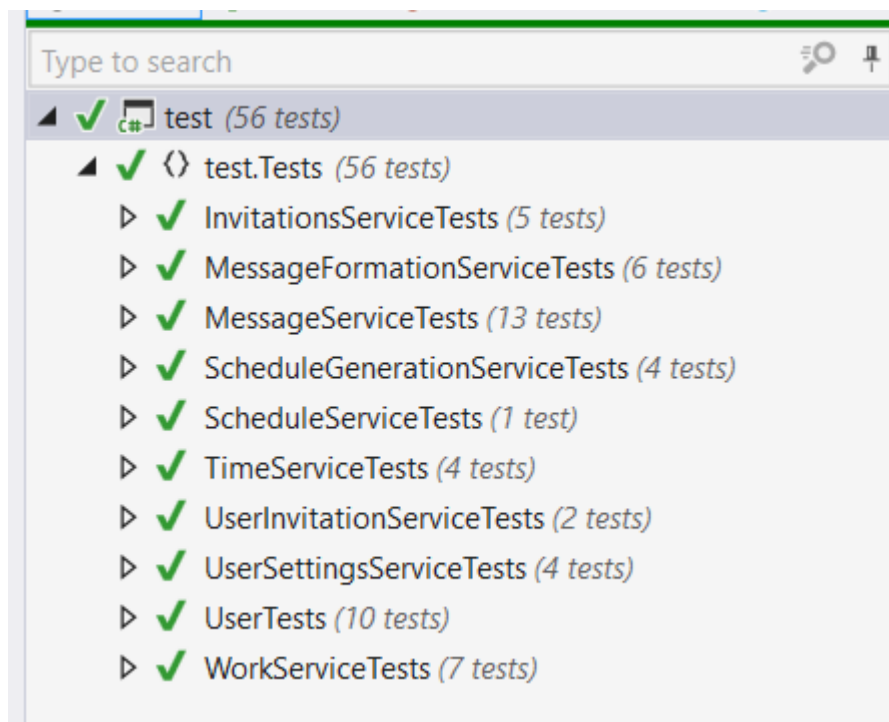
Servisų kodo padengimo kriterijus įgyvendintas sėkmingai, kadangi pateiktas 90 % kriterijus yra pasiektas (Žr. 3.1 pav.).

Symbol	Coverage (%) ▲	Uncovered/Total Stmts.
▲  Total	88%	145/1230
▲  bakalaurinis	83%	144/868
▲  bakalaurinis	83%	144/868
▷  Dtos	71%	37/126
▷  Infrastructure	79%	63/295
▲  Services	90%	44/444
▷  UserService	81%	11/59
▷  ScheduleG	83%	25/151
▷  WorksServ	88%	6/49
▷  InvitationS	96%	2/52
▷  ScheduleS	100%	0/10
▷  UserInvitat	100%	0/18
▷  TimeServic	100%	0/19
▷  MessageFc	100%	0/22
▷  UserSettin	100%	0/29
▷  MessageSe	100%	0/35
▷  Constants	100%	0/3
▷  test	99%	1/362

3.1 pav. Vientų testų kodo padengimas

Plačiau apie komponentų testavimą galima pasakyti, jog atlikti 56 testai. Kaip matome iš paveikslėlio (Žr. 3.2 pav.) visi testai buvo sėkmingi. Kodo padengimas (Žr. 3.1 pav.) svyruoja nuo 81% iki 100%. Tas parodo, kad testavimas atliktas gan nuosekliai, patikrinant daugelį galimų scenarijų.

Iš to galime teigti, kad komponentų testavimas buvo įvykdytas.



3.2 pav. Visi atlikti vienetų testai

Kiekvienam servisui parašyta nuo 1 iki 13 skirtingų testavimo scenarijų. Toks skaičius priklauso nuo serviso dydžio bei atliekamų funkcijų kiekio.

3.3.2. Rankinis API testavimas

Rankinio API testavimo metu bus kviečiami visi metodai. Šiai užduočiai atlikti sudaryti scenarijai, pateikiami lentelėmis (Žr. 3.1 lentelė – 3.5 lentelė), kuriose nurodyta metodo paskirtis, įvedami duomenys bei gauti rezultatai.

3.1 lentelė. Naujo darbo sukūrimas

Metodo paskirtis	Sukurti naują darbą
Užklaustos duomenys	<pre>{ "title": "work title", "description": "long work description ", "userId": 1, "durationInMinutes": 20, "activityPriority": 0, "willBeParticipant": false }</pre>
Sėkmingo veikimo scenarijus	Visi duomenys nurodyti
Sėkmingo veikimo rezultatas	Gražinamas 200 kodas, ir naujo darbo id
Nesėkmingo veikimo scenarijus	Trūksta dalies ar visų duomenų
Nesėkmingo veikimo rezultatas	Gražinamas 404 kodas

3.2 lentelė. Visų darbų paėmimas

Metodo paskirtis	Gauti visus darbus
Užklaustos duomenys	{}

Sėkmingo veikimo scenarijus	Bent vienas darbas sukurtas
Sėkmingo veikimo rezultatas	Gražinamas 200 kodas ir visų darbų sąrašas
Nesėkmingo veikimo scenarijus	Nėra nei vieno darbo
Nesėkmingo veikimo rezultatas	Gražinamas 404 kodas

3.3 lentelė. Vieno darbo paėmimas

Metodo paskirtis	Gauti darbą pagal jo id
Užklauso duomenys	{ }
Sėkmingo veikimo scenarijus	Visi duomenys nurodyti
Sėkmingo veikimo rezultatas	Gražinamas 200 kodas, ir naujo darbo id
Nesėkmingo veikimo scenarijus	Trūksta dalies ar visų duomenų
Nesėkmingo veikimo rezultatas	Gražinamas 404 kodas

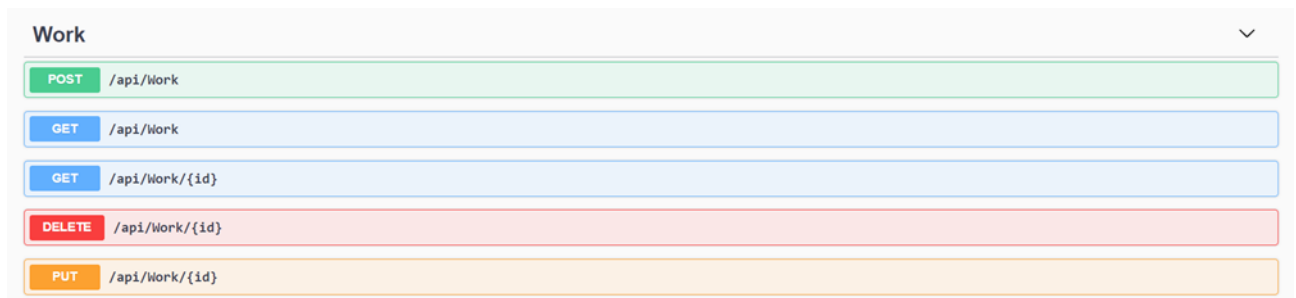
3.4 lentelė. Darbo redagavimas

Metodo paskirtis	Redaguoti darbą
Užklauso duomenys	{ "title": "updated title", "description": "long updated description", "userId": 1, "durationInMinutes": 15, "activityPriority": 1, "willBeParticipant": true }
Sėkmingo veikimo scenarijus	Visi duomenys nurodyti
Sėkmingo veikimo rezultatas	Gražinamas 204 kodas
Nesėkmingo veikimo scenarijus	Trūksta dalies ar visų duomenų
Nesėkmingo veikimo rezultatas	Gražinamas 404 kodas

3.5 lentelė. Darbo šalinimas

Metodo paskirtis	Šalinti darbą
Užklauso duomenys	{ }
Sėkmingo veikimo scenarijus	Darbas buvo pašalintas
Sėkmingo veikimo rezultatas	Gražinamas 204 kodas
Nesėkmingo veikimo scenarijus	Darbo nebuvo rastas
Nesėkmingo veikimo rezultatas	Gražinamas 404 kodas

Visi 5 anksčiau pavaizduoti testai (Žr. 3.1 lentelė – 3.5 lentelė) buvo atlikti naudojant Swagger grafinę sąsają (Žr. 3.3 pav.). Iš viso atlikta 27 API metodų testavimo atvejai ir jie visi buvo sėkmingi.



3.3 pav. Swagger grafinė vartotojo sąsaja

3.4. Statinė kodo analizė

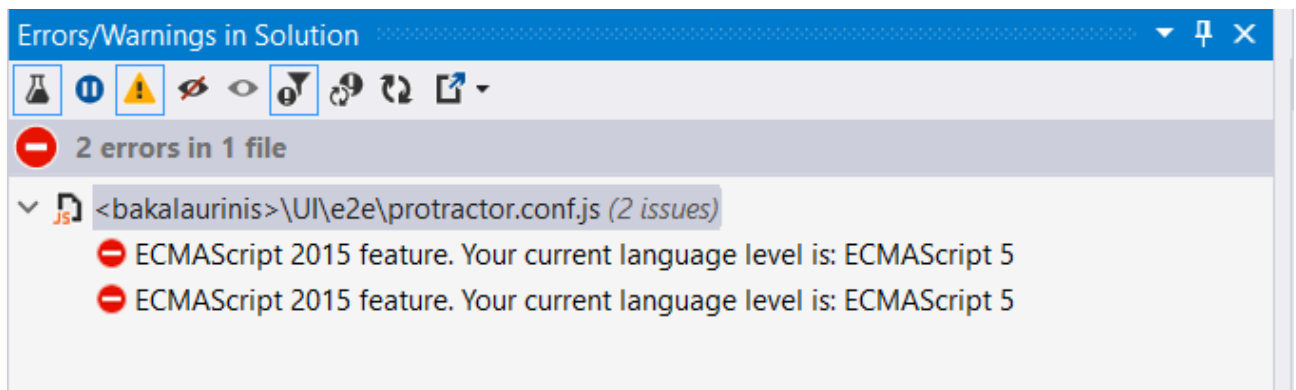
Statinė kodo analizė buvo atlikta naudojant Resharper įrankį. Šis įrankis sugeba aptikti TypeScript, CSS, HTML bei C# kalbų, naudotų šiame projekte, klaidas.

Pagrindinės šio įrankio aptinkamos klaidos yra šios:

- kintamųjų pavadinimo rašymo (išskyrus HTML ir CSS)
- formatavimo
- kodo stiliaus

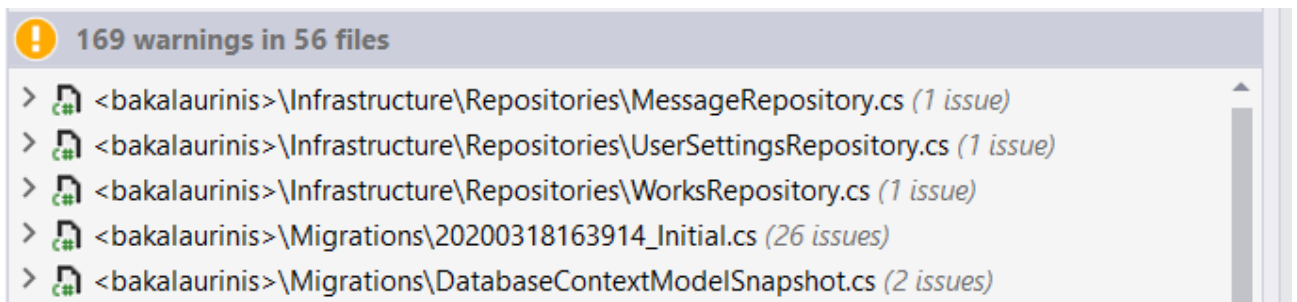
Ir daug kitų, kurias ne mažiau svarbu aptikti, klaidų.

Statinės kodo analizės metu aptiktos 2 klaidos (Žr. 3.4 pav.). Tiesa, jų nebuvo galima ištaisyti, kadangi tai yra Angular projektui paleisti reikalingas failas. Todėl ši statinės kodo analizės dalis laikoma sėkmingai įvykdyta.



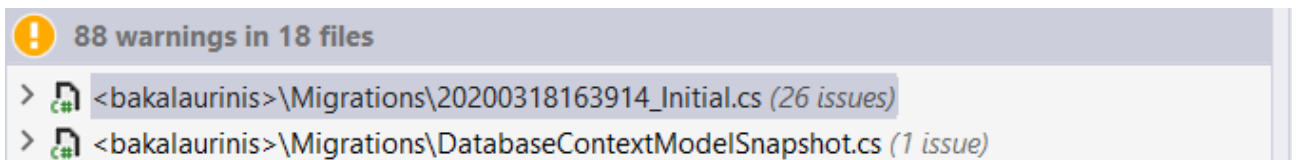
3.4 pav. Resharper aptiktos klaidos

Peržiūrėjus informaciją apie klaidas buvo nuspręsta peržiūrėti ir perspėjimus. Iš viso aptikti 169 perspėjimai (Žr. 3.5 pav.). Jie buvo net 56 skirtinguose failuose. Pamačius, jog jų skaičius yra gana didelis buvo nuspręsta patikrinti tuos failus ir įsitikinti ar galima jų kiekį sumažinti, taip pagerinant kodo kokybę.



3.5 pav. Perspėjimai dėl kodo kokybės, prieš peržiūrą

Tada buvo atlikta rankinė kodo peržiūra tuose failuose, kuriuose aptikti perspėjimai. Visi perspėjimai, kurie neturėjo įtakos kodo veikimui buvo ištaisyti ir liko tik tie, kurių nebuvo galima pakeisti, pavyzdžiui migracijose, kurios yra generuojamos įvedus komandą ir jų redagavimas nerekomenduojamas. Rezultatai (Žr. 3.6 pav.) rodo gerokai sumažėjusį perspėjimų skaičių.



3.6 pav. Perspėjimai dėl kodo kokybės, po peržiūros

Galime padaryti išvadą, kad statinis kodo testavimas padėjo padaryti projekto kodą kokybiškesnį, paprasčiau suprantamą ir padidino jo skaitomumą. Taip pat mažina klaidos tikimybę rašant didesnės apimties projektą.

3.5. Vartotojo sąsajos testavimas

Vartotojo sąsajai testuoti pasirinkau rankinį testavimo būdą. Pateikiama testavimo lentelė (Žr. 3.6 lentelė) su visais scenarijais bei tikėtinais rezultatais.

Kiekviena užduočiai atlikti yra būtina registracija ir prisijungimas, todėl atskiras šių panaudojimo atvejų testavimas nėra įtrauktas.

3.6 lentelė. Vartotojo sąsajos testavimas

Veiksmai	Aprašymas	Rezultatas
Sukurti naują darbą	Spaudžiamas „Works“ mygtukas pagrindiniame meniu Spaudžiama „Add new work“ mygtukas Užpildomi laukai Spaudžiamas „Create“ mygtukas	Sąrašas atsirado naujas darbas
Redaguoti darbą	Spaudžiamas „Works“ mygtukas pagrindiniame meniu Atsidariusiame sąrašas paspausti mygtuką redaguoti Pakeisti kelis laukus Spausti „Edit“	Darbas buvo atnaujintas
Pašalinti darbą	Spaudžiamas „Works“ mygtukas pagrindiniame meniu Atsidariusiame sąrašas paspausti raudoną „X“ mygtuką esantį sąrašas	Darbas buvo ištrintas

Pakviesti kitus vartotojus atlikti darbą	Spaudžiamas „Works“ mygtukas pagrindiniame meniu Spausti „Invite“ mygtuką esanti sąrašė Įvesti norimo vartotojo vardą Spausti „Send invitation“	Jei vartotojas yra atsiranda pranešimas apie išsiųstą pakvietimą, jei ne pranešimas apie klaidą
Perskaičiuoti darbų laiką	Paspausti ant „Schedule“ pagrindiniame meniu Paimti vieną darbą ir pertemti į kitą vietą	Darbų laikas perskaičiuotas
Keisti dieną	Paspausti ant „Schedule“ pagrindiniame meniu Pereiti dieną pirmyn ir atgal, pažiūrėti kaip keičiasi dienos naudojant „Next“ ir „Previous“ mygtukus	Diena keičiasi teisingai, po „Previous“ paspaudimo mažėja, po „Next“ didėja
Atnaujinti tvarkaraštį	Paspausti ant „Schedule“ pagrindiniame meniu Paspausti mygtuką „Refresh“	Tvarkaraštis atsinaujins
Peržiūrėti žinutes	Paspausti ant „Messages“ mygtuko arba voko simbolio pagrindiniame meniu	Žinučių sąrašas matomas
Filtruoti žinutes	Paspausti ant „Messages“ mygtuko arba voko simbolio pagrindiniame meniu Įvesti įvairius filtrus, pažiūrėti kaip kinta žinučių sąrašas	Žinučių sąrašas filtruojamas
Pašalinti visas žinutes	Paspausti ant „Messages“ mygtuko arba voko simbolio pagrindiniame meniu Paspausti mygtuką „Delete all“	Visos žinutės pašalintos
Pašalinti vieną žinutę	Paspausti ant „Messages“ mygtuko arba voko simbolio pagrindiniame meniu paspausti raudoną „X“ mygtuką esantį žinučių sąrašė	Pasirinkta žinutė ištrinta
Peržiūrėti pakvietimų sąrašą	Paspausti ant „Invitation“ mygtuko pagrindiniame meniu Peržiūrėti ar sąrašė yra pakvietimų	Pakvietimų sąrašas matomas
Priimti/atmesti pakvietimą	Paspausti ant „Invitation“ mygtuko pagrindiniame meniu Priimti/atmesti pakvietimą Jei pakvietimas priimtas, pažiūrėti ar darbų sąrašė atsirado naujas darbas	Pakvietimas pašalinamas iš sąrašo, priėmimo atveju darbas atsiranda darbų sąrašė
Peržiūrėti darbo į kurį kviečiama informacija	Paspausti ant „Invitation“ mygtuko pagrindiniame meniu Spausti mygtuką „Details“ ant pasirinkto pakvietimo	Darbo informacija matomos
Pakeisti darbo pradžios ir pabaigos nustatymus	Paspausti ant vartotojo vardo pagrindiniame meniu Pasirinkti laikus Spausti „Update“ mygtuką	Laikai atnaujinti
Šalinti paskyrą	Paspausti ant vartotojo vardo pagrindiniame meniu Paspausti mygtuką „Delete“ Patvirtinti paskyros šalinimą Pabandyti prisijungti su buvusios paskyros vardu ir slaptažodžiu	Paskyra ištrinta, prisijungti negalima
Nustatyti puslapio dydį	Paspausti ant vartotojo vardo pagrindiniame meniu Pasirinkti iš sąrašo tinkama dydį Spausti „Update“ mygtuką	Puslapio dydis pasikeitė
Atsijungti	Spausti „Log out“ mygtuką pagrindiniame meniu	Bus atsijungta nuo paskyros, informacija taps nepasiekiamą

Testuojant vartotojo sąsają jokių problemų neaptikta. Vartotojas gali atlikti visus nurodytus veiksmus (Žr. 3.6 lentelė).

4. Dokumentacija naudotojui

Šioje dalyje aprašoma sukurtos programinės įrangos funkcijos skirtos vartotojams.

4.1. Apibendrintas sistemos galimybių aprašymas.

Sistema sukurta darbams planuoti, taip taupant laiką. Yra galimybė peržiūrėti sistemos sukurta tvarkaraštį, suteikti prioritetus darbams, siųsti ir gauti pakvietimus bei atlikti kitas užduotis. Rūpindamiesi vartotojų patogumu, daugelyje puslapių sukūreime paieškos laukus, leidžiančius greičiau surasti reikalingą informaciją. Sukurtas ir nustatymų puslapis, kuriame vartotojas gali sistemą pritaikyti savo poreikiams.

4.2. Vartotojo vadovas

Norint naudotis sistema yra būtina registracija. Tą galima padaryti pasirinkus *Registration* meniu juostoje (Žr. 4.1 pav.).



4.1 pav. Neprisijungusio vartotojo meniu juosta

Toliau reikia užpildyti registracijos formą (Žr. 4.2 pav.). Ji nėra sudėtinga, reikia įvesti 4 laukus.

4.2 pav. Registracijos forma

Toliau pamatysite pranešimą ar registracija sėkminga. Jei viskas gerai, galite prisijungti prie sistemos. Priešingu atveju, bus klaidos pranešimai po kiekvienu įvedamu lauku, taip pat ir puslapio apačioje.

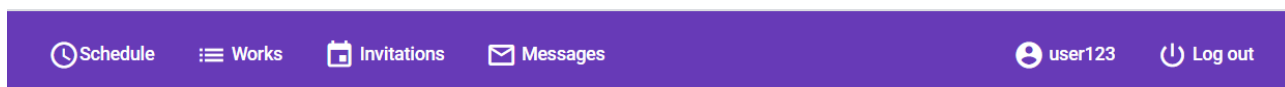
Norint prisijungti prie sistemos pagrindiniame meniu (Žr. 4.1 pav.) pasirinkti mygtuką *Log In*. Tada atsidarys prisijungimo forma (Žr. 4.3 pav.).

A login form titled "Login" with two input fields: "Username *" and "Password *". Below the fields is a purple button labeled "Log in".

4.3 pav. Prisijungimo prie sistemos forma

Užpildžius šią formą jau galima matyti kas vyksta sistemos viduje, žinoma jei duomenys buvo įvesti ir teisingi. Kitu atveju pamatysite klaidos pranešimą.

Po prisijungimo matomas šiek tiek pasikeitęs meniu (Žr. 4.4 pav.), iš kurio galima pasiekti visą reikalingą informaciją. Taip pat baigus darbą ir paspaudus mygtuką *Log out* atsijungti nuo sistemos.



4.4 pav. Pagrindinis meniu po prisijungimo

Paspaudę mygtuką *Schedule* būsite perkelti į tvarkaraštį (Žr. 4.5 pav.). Pradžioje jis bus tuščias. Vėliau bus parodyta kaip jį papildyti. Taip pat šiame puslapyje matomas jo užimtumas (*Busyness*).

A table titled "Day (2020-04-25) works list" with navigation arrows on the sides. The table has a header row labeled "Busyness" and three data rows showing time slots and their status.

Busyness
test (14:00:00 - 16:30:00)
ooo (16:30:00 - 17:22:00)
Veikia (17:22:00 - 17:47:00)

4.5 pav. Pasirinktos dienos darbų sąrašas

Tvarkaraštis (Žr. 4.5 pav.) yra vienos dienos darbų sąrašas. Jis be užimtumo rodymo turi dar vieną galimybę – nutempti pasirinktą darbą į kitą vietą, sistema perskaičiuos pradžios ir pabaigos laiką. Taip pat galite paspausti rodykles, taip keisdami datą ir matydami kitų dienų tvarkaraščius.

Add new work

Title

Priority

Search...

ooo

High

Edit

Invite

Veikla

Medium

Edit

Invite

test

Low

Edit

Invite

Items per page: 10

1 – 3 of 3

4.6 pav. Bendras darbų sąrašas

Kitas svarbus dalykas – bendras darbų sąrašas (Žr. 4.6 pav.). Jį galite atsidaryti sistemos meniu (Žr. 4.4 pav.) paspaudę *Works*. Tai vienas pagrindinių sistemos puslapių. Iš jo galima pasiekti darbo redagavimo bei sukūrimo puslapius (Žr. 4.7 pav.) atitinkamai paspaudus *Edit* ir *AddNewWork* mygtukus. Yra galimybės ir juos pašalinti, naudojant raudoną X mygtuką, tačiau šalinimas reikalaus patvirtinimo. Dar yra sukurtas darbų filtras, atrenkantis darbus pagal įrašytą pavadinimą (arba jo dalį). Kita šio puslapio galimybė pakviesti sistemos vartotojus atlikti tą patį darbą. Tam naudojamas *Invite* mygtukas, atidarantis (Žr. 4.8 pav.) langą. Paskutinė šio puslapio funkcionalumo dalis – paspaudus eilutę parodoma platesnė pasirinkto darbo informacija (Žr. 4.9 pav.).

New work

Title *

Title cannot be empty.

Description *

Trukmė *

Select priority

☐ Will be more participant

Close

Create

Edit work

Title *

Veikla

Description *

ukuhujkh

Trukmė *

25

Select priority

Medium

☒ Will be more participant

Close

Edit

4.7 pav. Naujo darbo pridėjimo bei redagavimo formos

Naujo darbo pridėjimo bei redagavimo formose (Žr. 4.7 pav.) reikia užpildyti/redaguoti tris laukus jei reikia, pasirinkti prioritetą. Dar yra galimybė pažymėti ✓, jei numatoma šiuo darbu dalintis su kitais vartotojais.

4.8 pav. Pakvietimų atlikti darbą puslapis

Pakvietimai (Žr. 4.8 pav.) siunčiami įvedant gavėjo vardą. Įvykus klaidai, pavyzdžiui vartotojas jau anksčiau buvo pakviestas ar tokio vartotojo registruoto sistemoje nėra, jūs gausite pranešimą, su klaidos paaiškinimu.

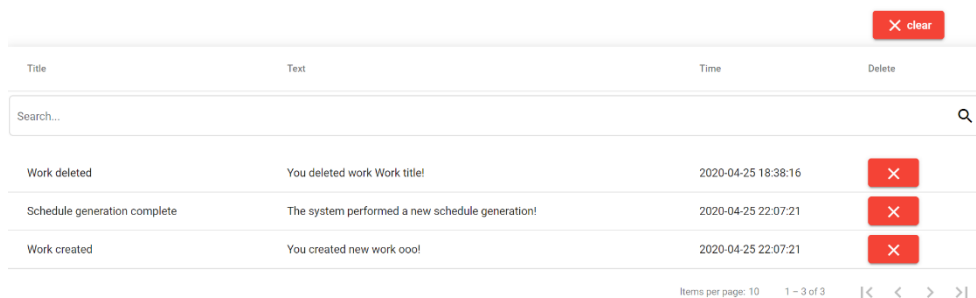
4.9 pav. Darbo informacijos peržiūros puslapis

Darbo peržiūros puslapis (Žr. 4.9 pav.) parodo platesnę darbo informaciją nei matoma bendrame darbų sąraše (Žr. 4.6 pav.) arba pakvietimuose (Žr. 4.10 pav.).

4.10 pav. Pakvietimų sąrašas

Atsidaręs pagrindiniame meniu (Žr. 4.4 pav.) puslapį *Invitations* vartotojas gali peržiūrėti, priimti arba atmesti jam atsiųstus kitų vartotojų darbų pakvietimus, tam naudodamas ✕(atmetimui) ir

✓(priėmimui). Taip pat paspaudę ant bet kurio teksto galite pamatyti platesnę darbo informaciją (Žr. 4.9 pav.).

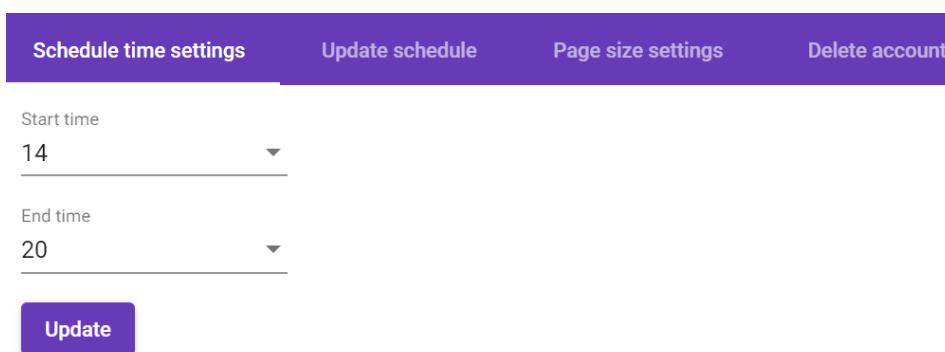


Title	Text	Time	Delete
Work deleted	You deleted work Work title!	2020-04-25 18:38:16	
Schedule generation complete	The system performed a new schedule generation!	2020-04-25 22:07:21	
Work created	You created new work ooo!	2020-04-25 22:07:21	

4.11 pav. Pranešimų sąrašas

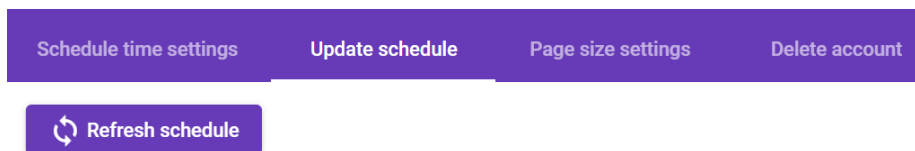
Pranešimų sąrašė (Žr. 4.11 pav.) vartotojo galimybės yra ribotos. Jis gali pašalinti pranešimus (visus naudojant *clear* arba po vieną), ir atlikti filtravimą įrašant pasirinktą tekstą ar jo dalį. Beje visi šalinimai reikalauja patvirtinimo.

Toliau pateikiamas nustatymų puslapis. Jis pasiekiamas pagrindiniame meniu (Žr. 4.4 pav.) spaudžiant ant vartotojo vardo (šiuo atveju user123).



4.12 pav. Darbo pradžios ir pabaigos laikų nustatymas

Pirmas nustatymas – darbų vykdymo pradžios ir pabaigos laikas (Žr. 4.12 pav.). Vartotojas gali pakeisti esamą laiką į savo norimą. Jei įvyksta klaida pasirenkant pradžios ir pabaigos darbo laiką vartotojas gauna įspėjimą apie klaidą.



4.13 pav. Tvarkaraščio atnaujinimo mygtukas

Antras nustatymas – mygtukas, kuris atnaujiną tvarkaraštį (Žr. 4.13 pav.).

Schedule time settings Update schedule **Page size settings** Delete account

Select page size

10 ▼

Update

4.14 pav. Elementų kiekio puslapyje nustatymas

Trečias nustatymas – elementų skaičius puslapyje (Žr. 4.14 pav.). vartotojas pasirenka vieną iš duotų variantų ir spaudžia *Update*.

Schedule time settings Update schedule Page size settings **Delete account**

Delete

4.15 pav. Paskyros šalinimas

Po paskyros pašalino (Žr. 4.15 pav.) mygtuko paspaudimo sistema paprašo patvirtinti ar tikrai norite pašalinti paskyrą. Jeigu trynimasis patvirtintas, paskyra tampa ištrinta, kartu su ja ir visi duomenys, tad prisijungimas prie sistemos tampa nebegalimas. Šioje stadijoje dar galite atšaukti paskyros trynimą jo nepatvirtindami.

Rezultatai ir išvados

Atlikus darbą padarytos tokios išvados:

1. Atlikus konkurentų analizę buvo surastos panašų funkcionalumą turinčios sistemos. Paaiškėjo, jog dauguma sistemų turi labai didelį kiekį funkcijų. Vartotojui tampa sudėtinga išmokti naudotis sistema. Todėl nuspręsta kurti naują sistemą, kuri būtų paprastesnė nei rinkoje esantys konkurentai.
2. Atlikus technologijų analizę, paaiškėjo, jog tokį patį rezultatą galima ir naudojant ir kitas tinklapių programavimo technologijas, todėl pasirinkimą lėmė šių technologijų populiarumas ir didelis pernaudojamų komponentų pasirinkimas.
3. Atlikus projektavimą, pavyko supaprasti itin sudėtingą tvarkaraščio generavimo algoritmą, kurio įgyvendinimas buvo didelė kliūtis kuriant sistemą. Taip pat sumažinti klaidų tikimybę programavimo metu ir sumažinti laiką, reikalingą sukurti sistemai.
4. Atlikus testavimą įsitikinta korektišku sistemos veikimu. Aukštas ištestuoto kodo procentas leidžia įsitikinti, kad vartotojas visada gaus teisingus rezultatus ir sumažės klaidų tikimybę plečiant sistemą.
5. Ruošiant dokumentaciją, didžiausias dėmesys buvo kreipiamas į paprastumą ir detalumą, pateikta daug vaizdinės informacijos. Pagrindinė priežastis yra galimas vartotojų įgūdžių trūkumas. Detalesnė dokumentacija leidžia vartotojui be pagalbos išspręsti daugumą problemų.
6. Sistema yra pilnai funkcionuojanti ir atitinkanti visus iškeltus reikalavimus. Esant reikalui paliktos galimybės plėsti sistemą.

Literatūros sąrašas

- [1] „JavaScript Frameworks 2020“ [Tinke] <https://dev.to/webapp007/javascript-frameworks-2020-699> [Kreiptasi: 2020 m. sausio 30 d.]
- [2] „The 15 Best Time Management Apps You Should Start Using Yesterday“ [Tinke] <https://www.wrike.com/blog/the-15-best-time-management-apps-you-should-start-using-yesterday/> [Kreiptasi: 2020 m. vasario 13 d.]
- [3] Appointy [Tinkle] <https://www.appointy.com/online-class-scheduling-software/> [Kreiptasi: 2020 m. vasario 13 d.]
- [4] Calendly [Tinkle] <https://calendly.com/pages/features> [Kreiptasi: 2020 m. vasario 13 d.]
- [5] Fantastical2 [Tinkle] <https://flexibits.com/fantastical/pricing> [Kreiptasi: 2020 m. vasario 13 d.]
- [6] Sling [Tinkle] <https://getsling.com/> [Kreiptasi: 2020 m. vasario 13 d.]
- [7] „Iterative Model: What Is It And When Should You Use It?“ [Tinkle] Prieiga internete: <https://airbrake.io/blog/sdlc/iterative-model> [Kreiptasi: 2020 m. kovo 22 d.]
- [8] „Unit Test Frameworks for C#: The Pros and Cons of the Top 3“ [Tinkle] <https://stackify.com/unit-test-frameworks-csharp/> [Kreiptasi: 2020 balandžio 16d.]
- [9] Swagger UI [Tinkle] <https://swagger.io/tools/swagger-ui/> [Kreiptasi: 2020 balandžio 16d.]