



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS**

Šarūnas Slanys

Ext JS karkaso našumo tyrimas

Baigiamasis magistro projektas

Vadovas

Dr. Šarūnas Packevičius

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Ext JS karkaso našumo tyrimas

Baigiamasis magistro projektas
Programų sistemų inžinerija (621E16001)

Vadovas

Dr. Šarūnas Packevičius

Recenzentas

Dr. Dominykas Barisas

Projekta atliko

Šarūnas Slanys

KAUNAS, 2016



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Šarūnas Slanys

(Studento vardas, pavardė)

Programų sistemų inžinerija, 621E16001

(Studijų programos pavadinimas, kodas)

„Ext JS karkaso našumo tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 ____ m. _____ d.
Kaunas

Patvirtinu, kad mano, **Šarūno Slanio**, baigiamasis projektas tema „Ext JS karkaso našumo tyrimas“ 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)

TURINYS

1. ĮŽANGA	10
1.1. Mokslinis naujumas	10
1.2. Darbo tikslas ir uždaviniai	10
2. ANALITINĖ DALIS	11
2.1. Analizės tikslas	11
2.2. Egzistuojantys sprendimai	11
2.2.1. Kliento/Serverio užduočių valdymo sistemos sprendimas	11
2.2.2. „iTeamWork“ ir „OnProject“ sprendimai.....	12
2.2.3. „Microsoft Project“ programinė įranga.....	12
2.3. Panaudojamumo tyrimas.....	12
2.3.1. Kas yra panaudojamumas?.....	13
2.3.2. Jakob Nielsen‘o 10 panaudojamumo euristicinių metodų.....	13
2.3.3. Panaudojamumo testavimas ir vertinimas	14
2.4. Platformos pasirinkimas.....	15
2.4.1. Ext JS	16
2.4.2. „Android“	17
2.4.3. „iOS“	18
2.4.4. „Microsoft Windows Phone“	18
2.4.5. Mobilios programos	19
2.4.6. Mobilios programos architektūra	20
2.5. Analitinės dalies išvados	20
3. Projektinė dalis.....	21
3.1. Projekto kūrimo pagrindas	21
3.2. Sistemos realizacijos metodai ir technologijos	21
3.2.1. MVC architektūra	21
3.2.2. REST serveris	22
3.2.3. Kitos naudotos technologijos	22
3.3. Reikalavimų analizė.....	22
3.3.1. Sistemos funkcijos	23
3.3.2. Vartotojų rolės	24
3.4. Panaudos atvejų aprašymai	25
3.4.1. Kontaktų modulio PA	25

3.4.2. Darbuotojų modulio PA	26
3.4.3. Dokumentų modulio PA	27
3.4.4. Užduočių modulio PA.....	28
3.4.5. Projektų modulio PA.....	30
3.5. Užduočių gyvavimo ciklas.....	31
3.6. Sistemos projektavimas.....	33
3.6.1. Sistemos architektūros tikslai ir apribojimai.....	33
3.6.2. Sistemos statinis vaizdas.....	33
3.6.3. Sistemos dinaminis vaizdas	36
3.6.4. Sistemos išdėstymo vaizdas	40
3.6.5. Duomenų vaizdas.....	41
4. Tyrimo dalis	43
4.1. Tikslas	43
4.2. Tyrimo aprašymas.....	43
4.3. Tyrimui naudota techninė įranga	43
4.4. Tyrimo eiga.....	43
4.5. Tyrimo išvados.....	44
5. Eksperimentinė dalis	45
5.1. Eksperimentinio tyrimo tikslas	45
5.2. Eksperimentinių tyrimų aprašymai	45
5.3. Našumo eksperimentas kintant duomenų eilučių skaičiui	45
5.4. Našumo eksperimentas kintant duomenų stulpelių skaičiui	46
5.5. Sistemos našumo eksperimentas naudojant skirtingas naršykles	47
5.6. Eksperimento rezultatai.....	49
6. Išvados	50
7. Literatūra.....	51
8. Terminų ir santrumpų žodynas	53
9. PRIEDAI.....	54
9.1. Diegimą patvirtinanti aktas	54
9.2. Užduočių ir projektų valdymo sistemos, paremtos Ext JS karkasu, kūrimas ir tyrimas.....	55

LENTELIŲ SĄRAŠAS

Lentelė 1. „JavaScript“ karkasų palyginimas	16
Lentelė 2. PA „Kontaktų paieška“ aprašymas	25
Lentelė 3. PA „Kontaktų segmentavimas“ aprašymas.....	25
Lentelė 4. PA „Kontaktų valdymas“ aprašymas	26
Lentelė 5. PA „Darbuotojų valdymas“ aprašymas.....	26
Lentelė 6. PA „Darbuotojų užimtumo valdymas“ aprašymas	27
Lentelė 7. PA „Darbuotojų teisių valdymas“ aprašymas	27
Lentelė 8. PA „Dokumentų saugojimas“ aprašymas	28
Lentelė 9. PA „Dokumentų priskyrimas užduotims ir projektams“ aprašymas	28
Lentelė 10. PA „Užduočių planavimas“ aprašymas	29
Lentelė 11. PA „Užduočių atlikimo stebėjimas“ aprašymas	29
Lentelė 12. PA „Užduočių kalendoriaus peržiūra“ aprašymas	29
Lentelė 13. PA „Užduočių eksportas CSV, XLSX formatais“ aprašymas	30
Lentelė 14. PA „Užduočių importas CSV, XLSX formatais“ aprašymas	30
Lentelė 15. PA „Ganto diagramos peržiūra“ aprašymas.....	31
Lentelė 16. PA „Resursų valdymas“ aprašymas	31
Lentelė 17. Duomenų bazės lentelių aprašymai.....	42
Lentelė 18. Sistemos tyrimo rezultatai prieš patobulinimus	44
Lentelė 19. Sistemos tyrimo rezultatai po patobulinimo	44
Lentelė 20. „Ext JS 5.1.1“ našumo tyrimas	45
Lentelė 21. „Ext JS 6.0.1“ našumo tyrimas	46
Lentelė 22. „Ext JS“ našumo tyrimas, kintant duomenų stulpelių skaičiui	47
Lentelė 23. Eksperimento rezultatai naudojant skirtingas naršykles	48

PAVEIKSLĖLIŲ SĄRAŠAS

1 pav. Mobiliųjų įrenginių struktūra	15
2 pav. „Android“ platformos architektūra.....	17
3 pav. „iOS“ sluoksniai.....	18
4 pav. „Windows Phone“ API.....	19
5 pav. Sistemos funkcijos.....	23
6 pav. Moduliai pasiekiami pagal vartotojo rolę.....	24
7 pav. Užduoties gyvavimo ciklas	32
8 pav. Sistemos paketų diagrama	33
9 pav. Bendravimo su DB paketas	34
10 pav. Vartotojo sąsajos paketas	35
11 pav. Administratoriaus sąsajos paketas.....	36
12 pav. Vartotojo prisijungimo būsenų diagrama	37
13 pav. Užduoties kūrimo būsenų diagrama	37
14 pav. Užduočių importavimas CSV ir XLSX formatais.....	38
15 pav. Vartotojo prisijungimo sekų diagrama	39
16 pav. Užduoties užbaigimo sekų diagrama.....	39
17 pav. Sistemos išdėstymo vaizdas	40
18 pav. Sistemos duomenų bazės vaizdas.....	41
19 pav. „Ext JS“ karkaso evoliucinis našumo tyrimas, pagal kintantį duomenų eilučių skaičių.....	46
20 pav. „Ext JS“ karkaso evoliucinis našumo tyrimas, pagal kintantį duomenų stulpelių skaičių.....	47
21 pav. „Ext JS“ našumo kitimas, naudojant skirtingas naršykles	48

Slanys, Šarūnas. Ext JS karkaso našumo tyrimas. Magistro baigiamasis projektas / vadovas dr. Šarūnas Packevičius; Kauno technologijos universitetas, Informatikos fakultetas.

Reikšminiai žodžiai: užduočių valdymas, projektų valdymas, Ext JS, MVC, REST, našumas.

Kaunas, 2016. 58 p.

SANTRAUKA

Užduočių ir projektų valdymas yra svarbus kiekvienos įmonės uždavinys. Tokioms įmonėms reikalinga užduočių ir projektų valdymo programinė įranga, kuri suteiktų galimybę efektyviai valdyti įmonės procesus, orientuotus į sėkmingą, greitą ir kokybišką projekto įgyvendinimą. Šiame darbe tiriamas užduočių ir projektų valdymo sistemos programinės įrangos projektavimo ir kūrimo procesas siekiant sukurti įmonei pažangią, greitą ir vartotojo konfigūruojamą užduočių ir projektų valdymo sistemą. Sistema suteikia tokias funkcijas, kaip įmonės darbuotojų valdymas, užduočių planavimas, darbuotojų našumo stebėjimas, užduočių kalendoriaus vaizdavimas, dokumentų valdymas ir kita. Užduočių ir projektų valdymo sistema gali sumažinti įmonės projektų valdymo išlaidas, pagerinti projektų kokybę ir padidinti įmonės produktyvumą. Naudojant „Ext JS“ karkasą ir MVC architektūrą, įgyvendinta sistema suteikia vartotojams lanksčią vartotojo sąsają ir skirtingas platformas palaikančią funkcionalią internetinę programą.

Siekiant gauti aukštą sistemos našumą, šiame darbe aprašomi našumo bandymai, kurių metu stebimas sistemos veikimo laikas, kai sistemai tenka apdoroti vis augančias duomenų imtis. Taip pat yra atliekami bandymai, naudojant skirtingas naršykles, vertinant naršyklių naudojamus skirtingus „JavaScript“ apdorojimo variklius. Atlikta eksperimentinių rezultatų analizę leido nustatyti, kokios technologijos turi būti naudojamos siekiant pagerinti sukurtos sistemos ir „Ext JS“ karkaso našumą, kai sistema yra apkraunama dideliais duomenų kiekiais.

Slanys, Šarūnas. Research of Ext JS framework performance: Master's thesis in information technologies / supervisor assoc. dr. Šarūnas Packevičius. The Faculty of Informatics, Kaunas University of Technology.

Key words: Task management, project management, Ext JS, MVC, REST, performance.

Kaunas, 2016. 58 p.

SUMMARY

Task and project management is an important task of each company. Such companies need task and project management software to efficiently manage processes, focused on the successful, rapid and high quality implementation of the project. This paper researches tasks and projects management system software design and development process, in order to give the company an innovative, quick and user-configurable system. The system provides functions such as management of company's employees, task scheduling, employee employment tracking, task calendar, document management, etc. Task and project management system can reduce enterprise project management costs, improve project quality and increase enterprise productivity. Based on the Ext JS framework and MVC architecture, implemented system provides responsive user interface and feature-rich cross-platform web application.

In order to obtain high system performance, this paper describes performance tests, which are carried out by using different amount of data and monitoring operation of system. Also there are performed tests using different browsers, for evaluating performance of JavaScript engines which are used by browsers. The analysis of experimental results led to conclusion which technologies have to be used, to improve system and Ext JS framework performance while working with large amount of data.

1. ĮŽANGA

Šiuo metu įmonės suteikiančios darbuotojams lankstų darbo grafiką arba galimybę dirbti iš darbuotojui patogios ir kintančios darbo vietos, susiduria su problemomis, kaip suvaldyti darbuotojų užimtumą ir išlaikyti įmonės darbo našumą. Sėkmingas ir patogus darbų planavimas ir darbų paskirstymas yra svarbus kiekvienos įmonės veiklos uždavinys. Tokioms įmonėms reikalinga užduočių ir projektų valdymo programinė įranga, skirta efektyviai valdyti procesus, sutelktus į unikalų tikslo pasiekimą (pvz. produkto ar paslaugos sukūrimą), turintį savo pradžią ir pabaigą, bei ribotus išteklius. Visas įmonės veiklos procesas turi būti valdomas taip, kad mažiausiomis sąnaudomis būtų galima administruoti projektus, užduotis, dokumentus ir kitus įmonėje vykstančius kasdienes įvykius.

Naudojantis šiuolaikinėmis technologijomis galima paspartinti ir supaprastinti visą užduočių valdymo procesą, apjungiant visus užduočių ir projektų duomenis, dokumentus, bei įmonės darbuotojus į vieną bendrą sistemą. Tokio pobūdžio sistemomis, vartotojams tenka naudotis kasdien, todėl svarbu, kad projektuojant sistemą būtų atsižvelgta į sistemos panaudojamumą, taip sukuriant patogią ir draugišką vartotojui sistemą.

1.1. Mokslinis naujumas

1. Atliktas „Ext JS“ karkaso evoliucinis našumo tyrimas.

1.2. Darbo tikslas ir uždaviniai

Darbo tikslas - sukurti įmonės projektų ir užduočių valdymo sistemą ir atlikti sukurtos sistemos ir „Ext JS“ karkaso našumo tyrimus.

Darbo uždaviniai:

- Išanalizuoti užduočių ir projektų valdymo sistemai keliamus reikalavimus.
- Suprojektuoti ir realizuoti užduočių ir projektų valdymo sistemą.
- Atlikti sukurtos sistemos ir „Ext JS“ karkaso evoliucinį našumo tyrimą ir eksperimentus.
- Atlikti populiariausių naršyklių „JavaScript“ variklių eksperimentus.
- Įdiegti sistemą įmonėje.

2. ANALITINĖ DALIS

2.1. Analizės tikslas

Analizės tikslas yra surinkti ir detalizuoti reikalavimus, keliamus įmonės projektų ir užduočių valdymo sistemai, kuri jungtų svarbiausias užduočių valdymo sistemos funkcijas. Tyrimo metu siekiama išanalizuoti esamų užduočių valdymo sistemų panaudojamumą, našumą ir produktyvumo įtaką įmonėms. Remiantis tyrimo rezultatais realizuoti sistemą naudojantis šiuolaikinėmis technologijomis ir pritaikyti populiariausioms platformoms.

2.2. Egzistuojantys sprendimai

Yra daug realizuotų komercinių ir nemokamų sistemų, skirtų užduočių ir projektų valdymui. Dauguma jų yra realizuotos keliose platformose ir vienokiu ar kitokiu būdu suteikia užduočių valdymo funkcionalumą, bet dauguma jų yra gana sudėtingos ir reikalauja daug laiko jas pildant, tai yra neefektyvu darbuotojams, kurie dažnai turi atlikti skirtingas mažas užduotis, taip pat dažnai dirbant prie skirtingų projektų.

2.2.1. Kliento/Serverio užduočių valdymo sistemos sprendimas

Naujojo Pietų Velso universitete buvo kuriama programinė įranga, kuri skirta projektų valdymui, kurių vykdymas yra išsidėstęs per kelias vietas [1]. „DSPMTool“ (Distributed Software Project Management Tool), tai yra paskirstyta kliento/serverio programinė įranga, skirta projektų informacijos valdymui, kuri apima įrankius ir metodus, naudojamus surinkti, analizuoti, integruoti bei platinti įvairių programinės įrangos projektų valdymo procesų rezultatus. Šis įrankis pasiskolino idėjas iš naujos grupinės kliento/serverio valdymo technologijos, vadinamos „Groupware“, kuris yra esminis technologijų rinkinys ir suteikia galimybę sujungti žmones iš skirtingų organizacijų, ar net žmones iš skirtingų pasaulio vietų, tam kad galėtų kartu rinkti ir dalintis informacija.

Sprendžiama problema apėmė pirmojo „DSPMTool“ prototipo analizę, kurios metu buvo nustatyti trūkumai, kad sistema nebuvo suprojektuota siekiant valdyti, vadovauti ir kontroliuoti visą programinės įrangos projekto vystymosi eigą ir negebėjo nustatyti tam tikrus projekto tikslus.

Antrasis prototipas buvo realizuotas sėkmingai ir praplėtė jau egzistuojantį įrankį papildomomis funkcijomis, kurios sujungė viską į vieną bendrą sąsają. Įrankis buvo papildytas šiomis pagrindinėmis funkcijomis:

- Projekto failų kontroliavimas;
- Komunikacija tarp asmenų, susijusių su projektu;
- Tikslus kiekvieno darbuotojo vaidmens apibrėžimas projekte.

Viskas buvo ganėtinai supaprastinta, vartotojams tiesiog reikėjo sukurti užduotis, nurodyti jų pradžios ir pabaigos datas, o toliau pats įrankis rūpinasi užduoties vykdymu ir jos įvykdymu laiku.

2.2.2. „iTeamWork“ ir „OnProject“ sprendimai

„iTeamWork“ [3] ir „OnProject“ [4] yra internetinės paslaugos, vadinasi nereikalinga atsisiųsti jokios programinės įrangos, o sistema pasiekama tiesiog per internetinę naršyklę. Naudotis tokiomis sistemomis yra rizikinga, nes visi projektų resursai ir failai yra saugomi nutolusiame serveryje, kuris priklauso net ne pačiai įmonei. Tokiu atveju visus failus ir duomenis tektų patikėti išorinei sistemai, lygiai taip pat patikint projektų saugumą ir jų vientisumo užtikrinimą. Tokia sistema gali būti tik patikima, jei mes patys visą sistemą įdedame į savo serverį ir patys saugome visus duomenis arba perkame tokio pobūdžio sistemą iš patikimo tiekėjo, kuris prisiims atsakomybę už duomenų saugumą ir vientisumą.

2.2.3. „Microsoft Project“ programinė įranga

„Microsoft Project“ suteikia aplinką planuoti ir kontroliuoti užduočių ir projektų vykdymo eigą, kartu skirta padėti projektų vadovams valdyti išteklių priskyrimą užduotims, valdyti projekto biudžetą, analizuoti darbo krūvį. Ši programinė įranga taip pat gali būti naudojama ir mobiliuose įrenginiuose su „iOS“ ir „Windows Phone“ operacinėmis sistemomis, ir tai suteikia sistemai patogumo ir lankstumo. Pagrindinis sistemos trūkumas - ši sistema neturi mobilios versijos „Android“ operacinei sistemai, kuri yra plačiausiai paplitusi mobili operacinė sistema.

„Microsoft Project“ yra atskira programinė įranga, todėl visi vartotojai norintys peržiūrėti projekto failus turi turėti „MS Project“ programinę įrangą, kuri yra mokama. Programinės įrangos kaina yra ganėtinai didelė, todėl mažesnėms įmonėms jos įsigijimas gali būti per brangus.

Sistemos naudojimas mažiems projektams nėra veiksmingas, jei vartotojams reikia pastoviai dirbti vis prie skirtingų projektų, todėl dažnai pildant užduočių sistemą nėra pasiekiamas maksimalus sistemos našumas.

2.3. Panaudojamumo tyrimas

Svarbu atlikti įvairių programinės įrangos ir aparatūros produktų panaudojamumo vertinimą, prieš paskelbiant sukurtą produktą visiems vartotojams [5]. Taigi įrangos palaikymo ir personalo mokymų kaina gali būti sumažinta ir tai žymiai padidins produkto panaudojamumą. Yra daug būdų atlikti panaudojamumo testus, bet problema yra pasirinkti tinkamą vertinimo parametą programinės įrangos panaudojamumui iširti.

2.3.1. Kas yra panaudojamumas?

Žinomas panaudojamumo ekspertas Jakob Nielsen apibrėžia panaudojamumą taip: „Panaudojamumas yra kokybės požymis, kuris įvertina, kaip paprastai vartotojas gali naudotis vartotojo sąsaja. Žodis „panaudojamumas“ taip pat nurodo metodus, kurie skirti pagerinti naudojimo paprastumą, projektavimo proceso metu“ [6].

Pagal Jakob Nielsen panaudojamumas yra apibrėžiamas penkiais kokybės komponentais:

- Išmokstamumas: kaip yra lengva vartotojams atlikti paprastas užduotis, kai jie pirmą kartą susiduria su vartotojo sąsaja?
- Efektyvumas: kai vartotojas išmoko vartotojo sąsają, kaip greitai jis gali atlikti užduotis?
- Įsimenamumas: kai vartotojas sugrįžta į vartotojo sąsają, po tam tikro laiko jos nesinaudojimo, kaip lengvai jie gali atkurti naudojimosi įgūdžius?
- Klaidos: kiek klaidų vartotojai daro, kokio sudėtingumo yra šios klaidos ir kaip lengvai jie gali ištaisyti klaidas?
- Pasitenkinimas: kaip yra malonu naudotis vartotojo sąsaja?

2.3.2. Jakob Nielsen'o 10 panaudojamumo euristinių metodų

Yra 10 pagrindinių vartotojo sąsajos projektavimo principų. Jie vadinami „euristiniais“, nes jie yra labiau taisyklių pobūdžio, nei konkrečios panaudojamumo projektavimo gairės [7].

1. Sistemos būklės matomumas – sistema turėtų visada informuoti vartotojus apie tai, kas vyksta, per kiek įmanoma trumpesnę laiką.
2. Varžybos tarp sistemos ir realaus pasaulio – sistema turėtų „kalbėti“ vartotojų kalba, žodžiais, frazėmis, pažįstamomis vartotojui, o ne sistemai orientuotais terminais. Reikalinga sekti realaus pasaulio konvencijų, įgyvendinat, kad informacija būtų parodoma natūralia ir logiška tvarka
3. Vartotojo valdymas ir laisvės – vartotojai dažnai renkasi sistemos funkcijas su klaidomis ir nori aiškiai matyti „avarinį išėjimą“, kuriuo galėtų išeiti iš nepageidaujamos būsenos, neinant per visą sistemos dialogą ar formą. Taip pat turėtų palaikyti anuliavimo ir perdarymo funkcijas.
4. Nuoseklumas ir standartai – vartotojai neturėtų susidurti su situacijomis, kurių metu turėtų galvoti ar skirtingi žodžiai, situacijos ar veiksmai reiškia tą patį dalyką. Sekite platformos standartus.
5. Klaidų prevencija – yra geriau, kai vartotojui rodomi aiškūs klaidų pranešimai, kurie yra kruopščiai suprojektuoti ir padeda suprasti apie galimas vykdymo pasekmes. Taip pat

rekomenduojama pašalinti ar bent sumažinti klaidos padarymo tikimybę arba pareikalauti vartotojo patvirtinimo prieš atliekant operaciją.

6. Geriau atpažinimas, nei prisiminimas – sumažinti vartotojo atminties apkrovą projektuojant matomus objektus, veiksmus ir pasirinkimus. Vartotojui neturėtų reikėti prisiminti informaciją iš vienos dialogo dalies, kurią reikia naudoti kitoje dialogo dalyje. Sistemos naudojimosi instrukcija turėtų būti matoma ar lengvai pasiekama, kai tik reikia.
7. Lankstumas ir naudojimo efektyvumas – greitikliai, dažniausiai nematomi pradedančiųjų vartotojų, gali paspartinti patyrusių vartotojų sąveiką su sistema, taip kad sistema gali būti pritaikyta tiek patyrusiems, tiek nepatyrusiems vartotojams.
8. Estetinis ir minimalistinis dizainas – dialoguose neturėtų būti informacijos, kuri yra nesvarbi ar retai panaudojama. Kiekvienas papildomas informacijos vienetas konkuruoja su svarbia informacija ir mažina svarbios informacijos santykinį matomumą.
9. Pagalba vartotojui atpažinti, diagnozuoti ir išvengti klaidos – klaidų pranešimai turi būti išreikšti aiškia kalba (ne kodais), tiksliai nurodyti problemą ir konstruktyviai pasiūlyti sprendimą.
10. Pagalba ir dokumentacija – net jei sistema nereikalauja dokumentacijos, ji yra būtina siekiant suteikti vartotojams pagalbą. Bet kokia informacija turi būti lengvai pasiekama, turėti paprastą paiešką, turi būti orientuota į vartotojo užduotis, konkrečių veiksmų sąrašus, kurie turi būti atliekami ir dokumentacija neturi būti per didelės apimties.

2.3.3. Panaudojamumo testavimas ir vertinimas

Panaudojamumo testavimas yra metodas, naudojamas įvertinti produktą, kai jį išbando vartotojai. Tai yra vertinama, kaip nepakeičiama panaudojamumo praktika, nes ji suteikia tiesioginį atsakymą, kaip iš tikrųjų vartotojai gali naudotis produktu.

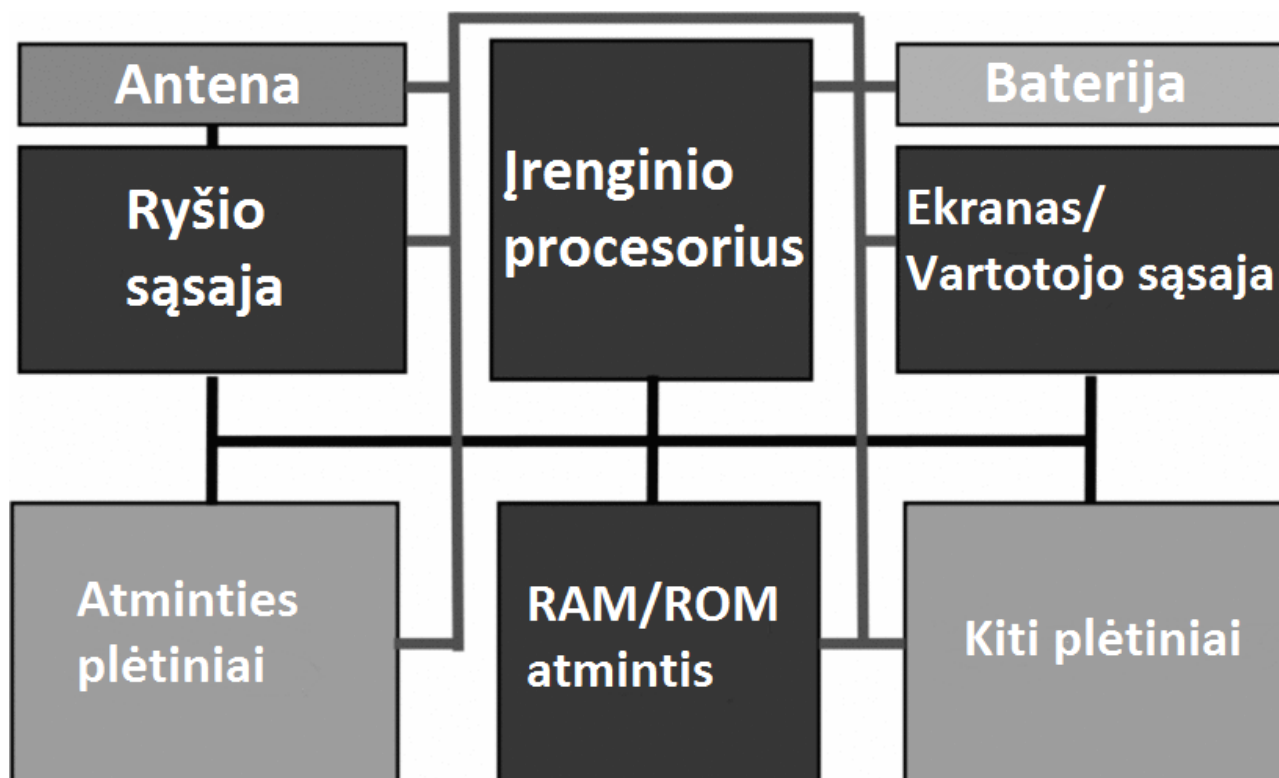
Panaudojamumo vertinimui gali būti naudojami šie metodai [8]:

- Apklausos ir anketos – daugelį panaudojamumo aspektų gali būti geriausiai išaiškinti paprasčiausiai klausiant vartotojų. Apklausos ir interviu yra naudingi metodai, iširti kaip vartotojai naudoja sistemą ir kokios funkcijos jiems labiausiai patinka arba nepatinka.
- Stebėjimas – paprasčiausias vartotojų lankymas ir jų stebėjimas, kai jie atlieka savo užduotis realioje aplinkoje. Panaudojamumo įvertinimo metodas yra labai svarbus tiek užduoties analizei, tiek informacijos surinkimui apie įdiegtos sistemos panaudojamumą. Stebėjimas yra patogiausias panaudojamumo metodas tol, kol jis apima vieno ar kelių vartotojų stebėjimą, neįsiterpiančią į jų darbą ir netrukdančią jo.

- Euristinis vertinimas – yra panaudojamumo inžinerijos metodas išsiaiškinti panaudojamumo problemas, esančias suprojektuotoje vartotojo sąsajoje, tam, kad į šias problemas būtų atkreipiamas dėmesys kartojant projektavimo procesą. Euristinis vertinimas apima mažą vertinimo kriterijų rinkinį, kurie tikrina vartotojo sąsają ir sprendžia jos atitikimą pripažintiems praktiškumo principams.

2.4. Platformos pasirinkimas

„Mobilumas gali būti apibrėžtas kaip galimybė judėti arba būti lengvai perkeltam. Mobilųjų įrenginių kontekste mobilumas susijęs su žmonių naudojamais nešiojamais ir funkcionaliais įrenginiais, kurie siūlo galimybę atlikti programų funkcijas, esant „neprisirišusiam“ prie vietos, taip pat galimybę prisijungti, gauti duomenis ir išsiųsti duomenis kitiems vartotojams, programoms, sistemoms“ [9]. Svarbu projektuojant mobiliąsias programas atkreipti dėmesį į mobiliųjų įrenginių ekrano atributus, tokius kaip: dydis, skiriamoji geba, spalvų gylis, apšvietimas ir energijos suvartojimas. Mobilųjų įrenginių ekrano naudojimas suvartoja didelį kiekį įrenginio energijos, todėl į tai turi būti ypatingai atsižvelgta. Taigi mobiliųjų įrenginių operacinės sistemos ir taikomosios programos turėtų taikyti metodus, kurie leistų taupyti sunaudojamą energiją. Energijos valdymas yra svarbus mobiliuose įrenginiuose, dėl ribotos baterijos gyvavimo trukmės, todėl svarbu efektyviai naudoti energiją naudojant tiek techninę įrangą, tiek programų projektavimą [10]. Mobilųjų įrenginių struktūra pateikta 1 pav. [11].



1 pav. Mobilųjų įrenginių struktūra

„Mobiliųjų programų platforma yra apibrėžiama, kaip operacinės sistema, skirta tam tikrai aibei palaikomų įrenginių su programų kūrimo bibliotekomis, programų kūrimo sąsajomis (API) ir programavimo įrankiais“ [12]. Mobiliųjų operacinių sistemų poreikis, kurios leidžia kurti programas, išsaugo dėl labai išaugusių mobiliųjų įrenginių platinimo. Pagrindinis šių platformų tikslas yra sukurti mobilią aplinką, kur vartotojai ir programų kūrėjai gali kurti programas. Pačios populiariausios komercinės platformos yra: „Android“, „iOS“, „Windows Phone“ [13].

2.4.1. Ext JS

„Ext JS“ karkasas yra atvirojo kodo „JavaScript“ biblioteka, kuri suteikia daug išplečiamų komponentų, kurie suteikia galimybę naudotis asinchroninėmis AJAX užklausomis [19], DOM, CSS, HTML ir kitomis technologijomis tam, kad sukurti funkcionalią ir skirtingas platformas palaikančią internetinę programą orientuotą į kompiuterius, planšetes ir mobiliuosius telefonus. „Ext JS“ naudoja modernias, naršyklių, HTML5 funkcijas, taip pat išlaiko palaikymą ir funkcionalumą senoms naršyklėms.

„Ext JS“ suteikia šimtus aukšto kokybės vartotojo sąsajos valdiklių, kurie yra sukurti siekiant patenkinti tiek paprasčiausių, tiek sudėtingų internetinių programų poreikius. „Ext JS“ šablonai ir išdėstymo tvarkytuvai suteikia pilną ekrano kontrolę, nepriklausomai nuo ekrano dydžio. Išplėstinis grafikų paketas leidžia vizualizuoti didelius duomenų kiekius. Ši sistema apima duomenų paketų valdymą ir priėmimą iš įvairių serverio pusės šaltinių. „Ext JS“ taip pat siūlo jau paruoštų ir konfigūruojamų temų paketą. „Ext JS“ palaiko prieinamumo paketą (ARIA) žmonėms su negalia pagal „Section 508“ standartus [20].

Yra daug panašių „JavaScript“ kalbos pagrindu sukurtų karkasų. „JavaScript“ karkasų, naudojančių MVC architektūrą. Palyginimas pagal svarbiausias funkcijas, kurios yra reikalingos užduočių ir projektų sistemos kūrimui pateiktos 1 lentelėje.

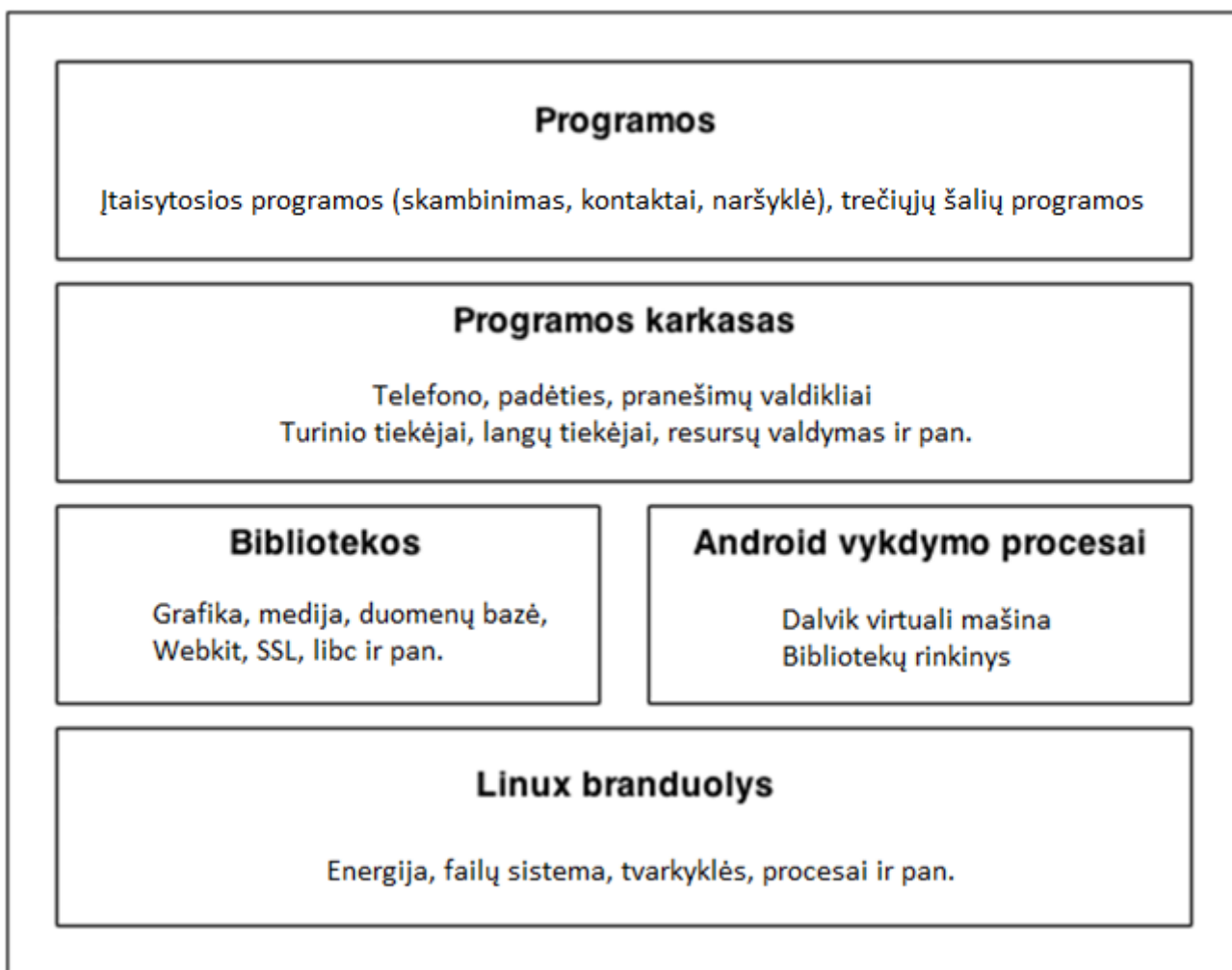
Lentelė 1. „JavaScript“ karkasų palyginimas

	AngularJs	Ext JS	Backbone	React	Ember
Vartotojo sąsajos saistymas	✓	✓		✓	✓
Daugkartinio panaudojimo komponentai	✓	✓	✓	✓	✓
Maršrutų programavimas	✓	✓	✓		✓
Kompoziciniai vaizdiniai		✓			✓
Tinklelis	✓	✓			
Raiškaus teksto redaktorius		✓		✓	
Diagramos		✓			
Daugiplatformis palaikymas	✓	✓	✓	✓	✓
Atjungtinė atmintinė	✓	✓			
Kūrėjo įrankiai		✓			

Kaip matome pagal palyginimo rezultatus, „Ext JS“ karkasas pirmąją pagal savo funkcionalumą ir suteikia visus, kuriamai sistemai svarbius komponentus. Prie viso to papildomai yra suteikiama „Ext JS“ karkaso programavimo aplinką.

2.4.2. „Android“

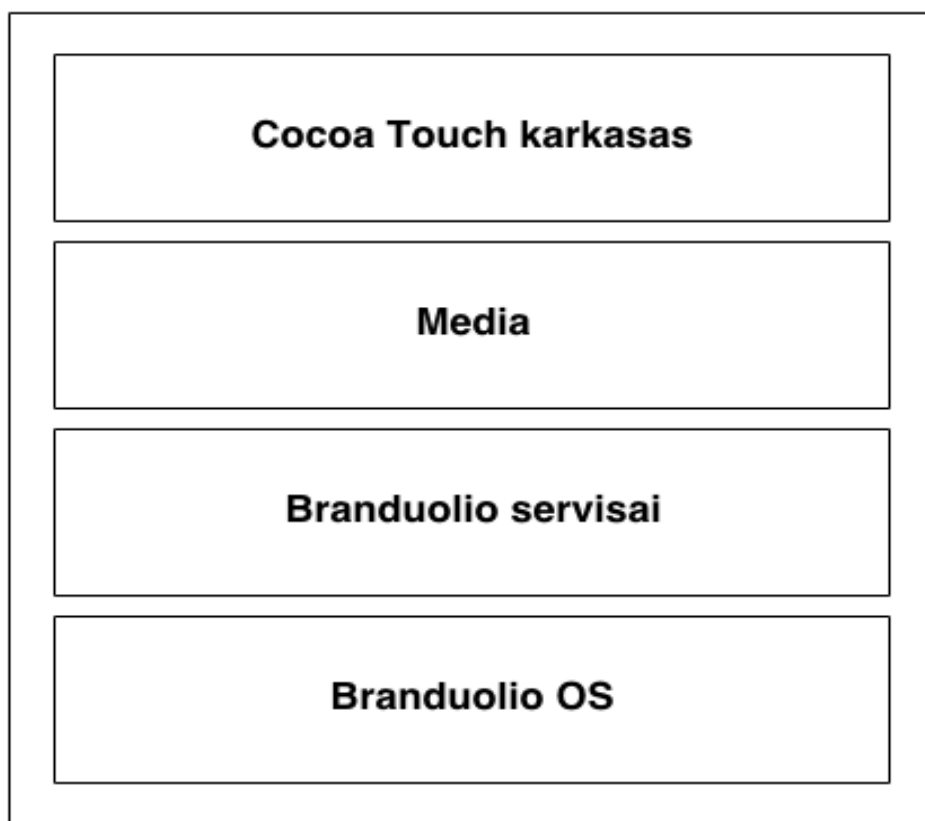
„Android“ yra atvirojo kodo platforma su atvirojo kodo licencija mobiliesiems įrenginiams, turintiems „Linux“ pagrindą. Ši platforma yra sukurta „Linux“ branduolio pagrindu, standartinių bibliotekų ir „Android“ programų sąsajų. „Linux“ branduolio pagrindinės paslaugos (įskaitant aparatinės tvarkyklės procesus ir atmintį, saugumo ir energijos valdymo) yra tvarkomos naudojant 3.4 versijos ar naujesnius branduolius. Bibliotekos veikia branduolio viršuje. „Android“ programos yra pirmiausia parašomos Java programavimo kalba ir sukompiliuojama į „Dalvik“ vykdomąjį (DEX) formatą. Kiekviena programa vykdoma atskirame procese, su atskira „Dalvik“ virtualia mašina [13]. Platformos atvirumas gali sukelti fragmentacijos problemų. „Android“ mobiliosios platformos architektūra pateikta 2 pav. [11].



2 pav. „Android“ platformos architektūra

2.4.3. „iOS“

Operacinė sistema valdo įrenginio techninę įrangą ir teikia reikalingas technologijas plėsti standartines programas. „iOS“ programinės įrangos kūrimo rinkinys (SDK) apima įrankius ir reikalingas sąsajas, kurios skirtos kurti, įdiegti, testuoti ir vykdyti programas. Programos yra sukurtos naudojant sisteminių karkasą ir objektinę C programavimo kalbą, kurios yra vykdomos tiesiai „iOS“ platformoje. Aukščiausiam lygyje „iOS“ veikia kaip tarpininkas tarp pagrindinės techninės įrangos ir programų, kurios yra rodomos ekrane. Programos bendrauja su technine įranga per apibrėžtas sistemos sąsajas, kurios apsaugo programą nuo aparatūrinės įranga pakeitimų. Visa tai leidžia lengvai kurti programas, kurios nuosekliai dirba skirtingų įrangos galimybių prietaisuose. „iOS“ technologijų įgyvendinimas gali būti pavaizduotas kaip sluoksnių rinkinys, kaip parodyta 3 pav.

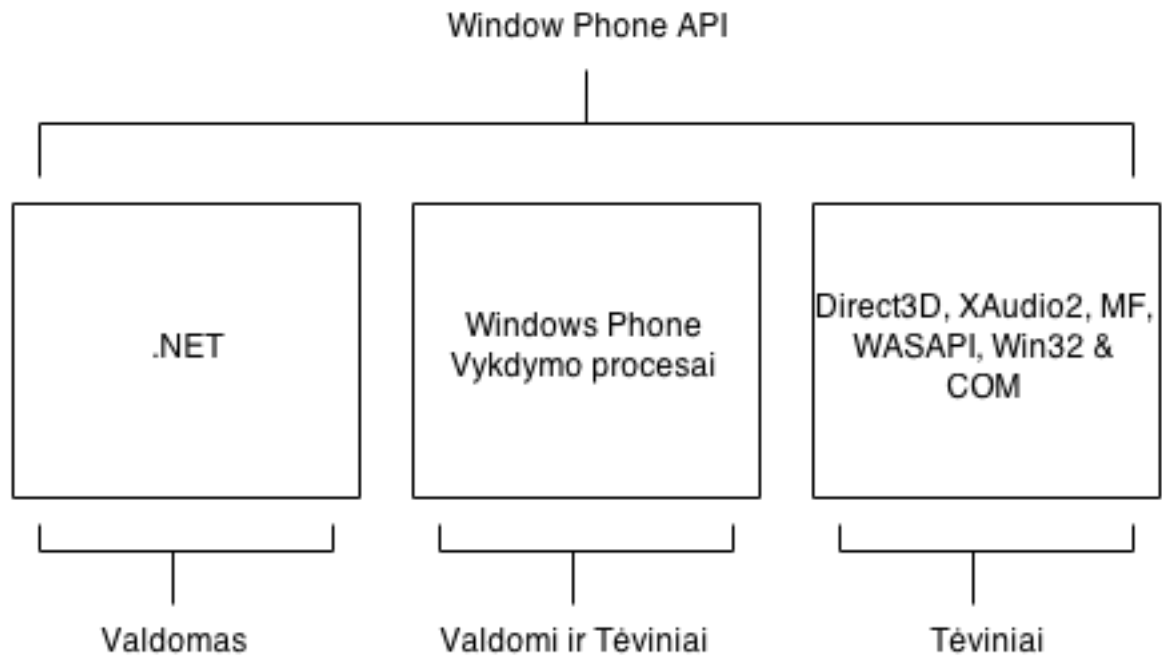


3 pav. „iOS“ sluoksniai

Žemiausiame sistemos lygyje yra servaisai ir technologijos, kurios priklauso visoms programoms. Aukštesni sluoksniai apima sudėtingesnius servaisus ir technologijas.

2.4.4. „Microsoft Windows Phone“

„Microsoft Windows Phone“ pristato galimybę naudoti C++ per XAML programą ir žaidimus parašytus naudojant „Direct3D“. „Windows Phone“ API apima pilną programinių objektų rinkinį, naudojamų „Windows Phone 8“ platformoje. 4 pav. iliustruoja kokios API egzistuoja ir kaip jos yra integruotos į „Windows Phone“ API.



4 pav. „Windows Phone“ API

„.NET“ API turi klases ir tipus iš pagrindinės vardų srities. „Windows Phone“ vykdymas yra vietinis API poaibis, kuris yra sukurtas operacinėje sistemoje. Visa tai įgyvendinta C++ ir VB.NET kalbomis ir suteikia galimybę lengvai prisitaikyti prie kitos kalbos. Be šių API, dar galima pasiekti „Win32“ API, kuris suteikia prieigą prie platformos žemo lygio funkcijų [14].

2.4.5. Mobilios programos

Mobilioji programa yra kompiuterinė mobilioji programa sukurta atlikti užduotis ar teikti vartotojui įvykius. Mobilios programos kūrimas turi keletą reikalavimų, kurie nėra dažnai pasitaikantys, kuriant tradicinę programinę įrangą [15]:

1. Sąveika su kitomis programomis – dauguma mobiliųjų prietaisų turi daug programų iš skirtingų šaltinių, todėl naujai sukurtos programos dažniausiai turi bendrauti su jau įdiegtomis programomis.
2. Jutiklių valdymas – programos turi suteikti galimybę mobiliajam įrenginiui naudoti jutiklius, kad pagerinti vartotojo naudojimosi patirtį.
3. Tėvinės ir hibridinės programos – mobilieji įrenginiai dažnai apima programas, kurios teikia paslaugas per telefono ryšį arba internetu ir įtakoja duomenų atvaizdavimą prietaise.
4. Vartotojo sąsaja – turi būti draugiška ir intuityvi vartotojui. Vartotojo sąsaja taip pat turi atsižvelgti į įrenginio apribojimus, tokius kaip baterijos gyvavimo trukmė, procesoriaus pajėgumas.

5. Energijos sunaudojimas – programos daugeliu aspektų įtakoja prietaiso energijos sunaudojimą, todėl neatsižvelgus į tai sukurta programa gali naudoti per didelį kiekį įrenginio energijos.

2.4.6. Mobilios programos architektūra

Yra daug galimų kombinacijų renkantis programos architektūra. Pagrindinės mobilios programos architektūros gali būti tokios [9]:

1. Klientas-Serveris – dauguma mobiliųjų programų yra modeliuojamos pagal šią architektūrą, kurioje vienas ar daugiau įrenginių pateikia užklausas serveriui. Serveris atsako su prašyta informacija.
2. Klientas (įrenginys) – mobilus įrenginys gali dirbti ir kaip įrenginys ir kaip serveris pats sau.
3. Centralizuotas serveris – šią architektūrą sudaro nuo vieno iki trijų kodo sluoksnių, įgyvendintų vienoje ar trijose pakopose. Kitos architektūros yra grįstos pakopų skaičiumi, kurias įgyvendina serveris ir daugeliu atveju, kartu yra naudojama duomenų bazė.
4. „Peer-to-Peer“ – ši architektūrą yra skirta dalintis kompiuterinius išteklius, pavyzdžiui turinį, atmintį ar procesorių tiesiogiai jais keičiantis, o ne reikalaujant pateikti išteklius iš centralizuoto serverio.
5. Debesis – debesų kompiuterija yra apibrėžta nacionalinio technologijų ir standartų instituto: „Debesų kompiuterija yra modelis leidžiantis patogią, pagal poreikimą prieigą prie bendrai patalpintų skaičiavimo išteklių (pavyzdžiui serveriai, atmintis, programos, servisai), kurie gali būti greitai paleisti su minimaliomis valdymo pastangomis ar minimaliu paslaugų tiekėjo įsikišimu“ [16].

2.5. Analitinės dalies išvados

1. Atlikus panašių sistemų analizę nustatyta, kad aktualiausia realizuoti sistemą, palaikančią prieigą per kompiuterių naršykles ir mobiliuosius įrenginius.
2. Išanalizavus „JavaScript“ karkasus, nustatyta, kad „Ext JS“ karkasas funkcionalumu lenkia kitus tirtus „JavaScript“ karkasus ir yra tinkamiausias kuriant internetinę užduočių ir projektų valdymo sistema, palaikančią prieigą per kompiuterių naršykles ir mobiliuosius įrenginius.
3. Išanalizavus galimus programų architektūros tipus, nustatyta kad Klientas-Serveris architektūra yra dažniausiai naudojama ir labiausiai tinkama architektūra kuriamai sistema.

3. PROJEKTINĖ DALIS

3.1. Projekto kūrimo pagrindas

Įmonės, kuriančios programinę įrangą, susiduria su problemomis, kai sunku suvaldyti visą programų kūrimo procesą, tam tikros užduotys yra neįvykdytos arba yra įvykdomos nekorektiškai ir nėra dėl to atsakingų asmenų. Taip pat informacijos perdavimas, bei komandos valdymas yra sudėtingas, jei visa tai nėra sujungta į vieną bendrą sistemą. Daugumos esamų sistemų problema yra ta, kad sistemos yra neintuityvios ir reikalauja daug laiko ir resursų jas pildant.

Programinė įranga skirta, valdyti užduotis ir projektus, bei darbuotojus, yra svarbi įmonėms atliekančioms programavimo darbus, nes naudojant vieningą sistemą, informacija gali būti matoma vienoje vietoje. Taip pat, bet kuriuo metu gali būti matomas darbo progresas, bei valdomas darbuotojų užimtumas.

Projekto tikslas – naudojant „Ext JS“ karkasą sukurti užduočių ir projektų valdymo sistemą.

Projekto realizacijos metu atlikti darbai:

- Reikalingų technologijų pasirinkimas;
- Reikalavimų analizė ir specifikavimas;
- Užduoties gyvavimo ciklo modelio sudarymas;
- Projekto realizavimas;
- Projekto programinės įrangos testavimas ir kokybės vertinimas;
- Projekto diegimas.

3.2. Sistemos realizacijos metodai ir technologijos

3.2.1. MVC architektūra

Sistemos realizacijai pasirinkta MVC sistemos architektūra, kurios veikimas paremtas programos dalių išskaidymu [21]. Tai reiškia, kad programa yra skaidoma į dalis, pagal kiekvienos dalies funkcionalumą arba poreikius. MVC architektūroje programa yra išskiriama į tris dalis:

- Modelis – atsakingas už duomenų valdymą ir duomenų pateikimą.
- Valdiklis – atsako už modelio valdymą, bei rodinių atnaujinimą pagal vartotojo veiksmus.
- Rodinys – suteikia programai išvaizdą, kuri būtų suprantama vartotojui. Tai yra sistemos formos, sąrašai ir pan.

3.2.2. REST serveris

REST yra architektūrinis stilius, skirtas paskirstytoms sistemoms kurti. REST ignoruoja komponentų realizacijos technologiją ir protokolų sintaksę, tam kad sutelkti visą dėmesį į sudedamųjų dalių funkcijas ir šių dalių sąveikavimą su kitais komponentais [17, 18].

Realizuojamoje sistemoje pasirinkta, supaprastina, atvirojo kodo programinė įranga, skirta PHP programavimo kalbai, kuri buvo parašyta Jacob'o Wright'o [22].

REST atlieka universalių išteklių adresų formatavimą ir suteikia galimybę paprasčiau apdoroti užklausas tarp vartotojo sąsajos ir serverio. Pavyzdžiui, jei programa siunčia HTTP GET užklausą į serverį (pvz. adresu: projekto_adresas.lt/užduotys), serveris apdorojęs tokią užklausą grąžina visą duomenų rinkinį apie užduotis JSON formatu. Jei siųstume patikslintą užklausą į serverį (pvz. adresu: projekto_adresas.lt/užduotis/:užduoties_id) – tokiu atveju serveris turėtų grąžinti tik vieną užduotį pagal gautą užduoties identifikacijos numerį. Tokiu pačiu būdu realizuojami ir HTTP POST užklausos, skirtos sukurti naujus įrašus, HTTP PUT užklausos, skirtos duomenų atnaujinimui ir HTTP DELETE užklausos, duomenų trynimui.

3.2.3. Kitos naudotos technologijos

Kitos technologijos ir programavimo kalbos naudotos kuriant projektą:

- Programavimo kalbos – PHP, „JavaScript“;
- Žiniatinklio serveris – Internet Information Services (IIS) 7;
- Duomenų bazė - Microsoft SQL Server;
- Serverio operacinė sistema - Microsoft Windows Server Service Pack 2.

3.3. Reikalavimų analizė

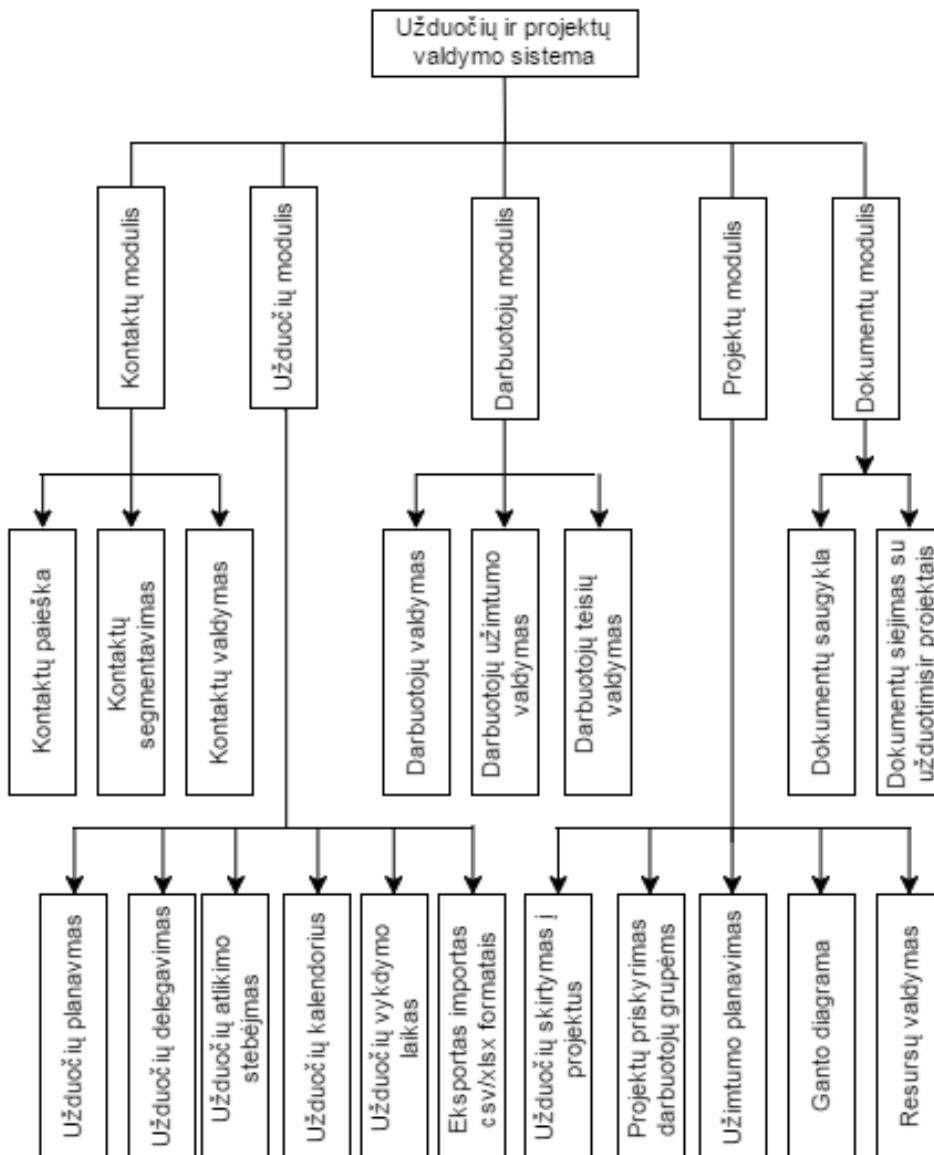
Kartu su sparčiu darbo tempu ir skubiais užduočių pasikeitimais sistema turi integruotis į kitas jau vartotojų naudojamas sistemas ir pateikti pranešimus įvairiais patogiais būdais. Elektroninių laiškų siuntimas į vartotojų pašto dėžutes ir pranešimų siuntimas į vartotojų mobiliuosius įrenginius sumažina laiką nuo užduoties pateikimo iki vartotojo pasiekimo, taip pat iki užduoties vykdymo pradžios.

Užduočių ir projektų sistema turi dvi vartotojų roles: projektų vadovus ir vartotojus. Projektų vadovai turi aukščiausias sistemos teises ir turi galimybę valdyti įmonės kontaktų bazę, projektus, darbuotojus, kurti vartotojams užduotis, tvarkyti užduočių failus, peržiūrėti Ganto diagramą, eksportuoti/importuoti užduotis XLSX, CSV formatu. Taip pat projektų vadovų rolė paveldi visas vartotojo rolės teises. Vartotojų rolė skirta, priskirtų užduočių peržiūrai, vykdymui, papildomos informacijos tiekimui apie užduotį. Užduotys skirstomos į penkias būsenas: projektavimas,

vykdymas, testavimas, atiduota klientui ir užbaigta. Pasikeitimai apie užduočių sukūrimą, būsenos pasikeitimą, siunčiami elektroniniu paštu su visa užduoties informacija.

3.3.1. Sistemos funkcijos

Atsižvelgiant į keliamus sistemai reikalavimus, reikalinga internetinė programa skirta užduočių ir projektų valdymui per internetinę naršyklę. Realizuojant sistemą reikalinga įgyvendinti šiuos paketus: vartotojo sąsajos paketas, projektų vadovo sąsajos paketas, el. laiškų siuntimo paketas, bendravimo su duomenų baze paketas. Sistema vartotojai gali naudotis tik po sėkmingos autorizacijos, naudojantis asmeniniais prisijungimo duomenimis. Tokiu būdų visi vartotojo atlikti veiksmai yra saugomi duomenų bazėje. Kadangi sistema vartotojams turi būti pritaikyta kasdieniniam naudojimui, reikalinga intuityvi vartotojo sąsaja, kurios pagalba vartotojas gali išdėlioti programos objektus taip kaip jam yra patogiu, taip padidinant sistemos panaudojamumą. Sistemos funkcijų vaizdas pateiktas 5 pav.



5 pav. Sistemos funkcijos

Kaip matome paveiksle sistema, pagal funkcionalumą, yra išskaidoma į penkis pagrindinius modulius:

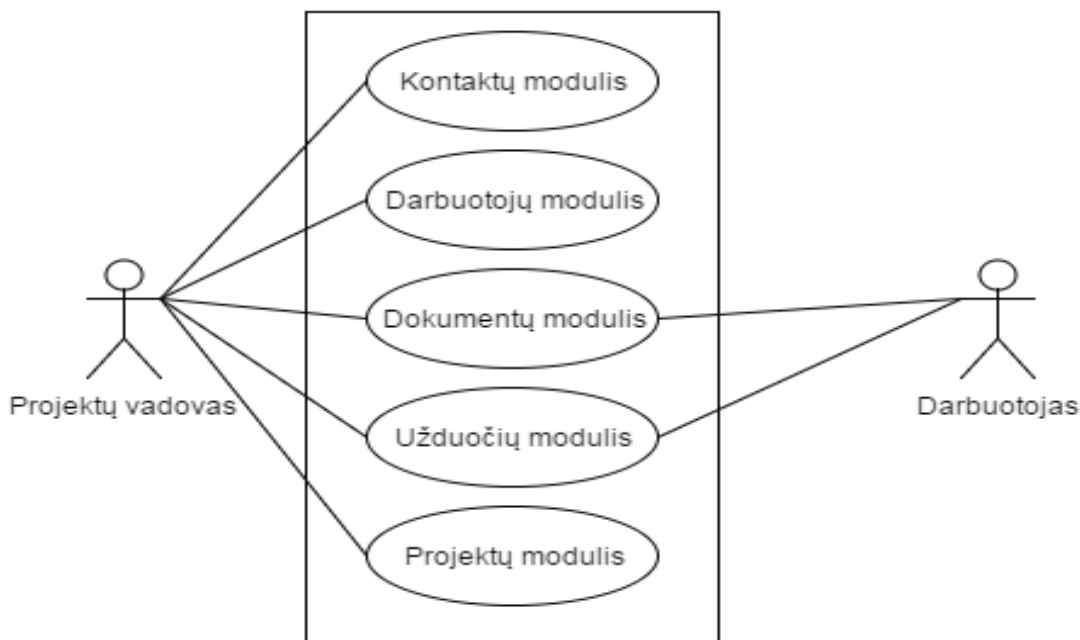
- Kontaktų modulis – skirtas kontaktų valdymui ir jų segmentavimui.
- Darbuotojų modulis – skirtas darbuotojų duomenų valdymui, darbuotojų užimtumo valdymui, darbuotojų teisių valdymui.
- Dokumentų modulis – skirtas saugoti failus, susijusius su užduotimis ir projektais.
- Užduočių modulis – skirtas užduočių planavimui, užduočių delegavimui, užduočių vykdymo procesų stebėjimui, užduočių kalendoriaus peržiūrai, duomenų importavimui ir eksportavimui.
- Projektų modulis – skirtas skirstyti užduotis į projektus ir darbuotojų grupes, planuoti darbuotojų užimtumą ir resursų valdymą, peržiūrėti kritinį projekto kelią, naudojant ganto diagramą.

3.3.2. Vartotojų rolės

Siekiant išskaidyti sistemos funkcionalumą pagal vartotojus, išskirtos dvi vartotojų rolės:

- Projektų vadovai – asmenys atsakingi už bendravimą su įmonės klientais, užduočių projektavimą ir jų delegavimą darbuotojams. Taip pat valdo ir planuoja užduočių procesą.
- Darbuotojai – asmenys, kurių pagrindinė funkcija sistemoje yra peržiūrėti ir vykdyti jiems priskirtas užduotis.

Pagal priskirtą vartotojo rolę, sistemos funkcionalumas yra apribojimas pagal konkrečiai rolei priskirtas funkcijas. Sistemos moduliai pasiekiami pagal vartotojo roles pavaizduoti 6 pav.



6 pav. Moduliai pasiekiami pagal vartotojo rolę

Matome, kad projektų vadovas gali naudotis visais sistemos moduliais, tuo pačiu ir visu sistemos funkcionalumu. Vartotojai turintys tik darbuotojo rolę turi teisę naudotis tik užduočių ir dokumentų moduluose esančiu funkcionalumu.

3.4. Panaudos atvejų aprašymai

Šiame skyriuje pateikiami visų sistemos panaudos atvejų aprašymai, išskirstant panaudos atvejus pagal sistemos modulius.

3.4.1. Kontaktų modulio PA

Kontaktų valdymo modulis - skirtas projektų vadovams, valdyti duomenis apie įmonės klientus ir saugoti, bei peržiūrėti visą su jais susijusią informaciją. Toliau pateikiami kontaktų modulio panaudos atvejų aprašymai.

Lentelė 2. PA „Kontaktų paieška“ aprašymas

1. Panaudojimo atvejis: Kontaktų paieška	
Tikslas:	Lengvai ir greitai surasi reikiamą kontaktinį asmenį.
Aktoriai:	1. Projektų vadovas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Vartotojas pasirenka kontaktinio asmens paieška arba spaudžia CTRL+F.
Po sąlygos:	Surastas reikiamas kontaktinis asmuo.
Pagrindinis scenarijus:	<ol style="list-style-type: none"> 1. Įvedama dalis ieškomos frazės. 2. Spaudžiama ieškoti. 3. Ieškomas kontaktinis asmuo pagal visus suvestus kontaktinio asmens atributus. 4. Grąžinamas rezultatas.
Alternatyvinis scenarijus:	Neradus ieškomo kontaktinio asmens apie tai pranešama vartotojui.

Lentelė 3. PA „Kontaktų segmentavimas“ aprašymas

2. Panaudojimo atvejis: Kontaktų segmentavimas	
Tikslas:	Suskirstyti kontaktinius asmenis į grupes pagal įmones, projektus.
Aktoriai:	1. Projektų vadovas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Vartotojas pasirenka kontaktinį asmenį ir spaudžia segmentavimo mygtuką.
Po sąlygos:	Kontaktinis asmuo priskirtas grupei.
Pagrindinis scenarijus:	<ol style="list-style-type: none"> 1. Atsidariusiame lange matomas pasirinktas kontaktinis asmuo. 2. Pasirenkamas priskiriamos grupės tipas. 3. Pasirenkama grupė iš sąrašo. 4. Patvirtinami pakeitimai.
Alternatyvinis scenarijus:	Nepavykus kontaktinio asmens priskyrimui grupei, priežastis pranešama vartotojui.

Lentelė 4. PA „Kontaktų valdymas“ aprašymas

3. Panaudojimo atvejis: Kontaktų valdymas	
Tikslas:	Suvesti, redaguoti, pašalinti duomenis apie kontaktinius asmenis.
Aktoriai:	1. Projektų vadovas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Vartotojas pasirenka naujo kontakto kūrimą, arba du kartus pelės pagalba spaudžia ant esamo.
Po sąlygos:	Apdorota kontaktinio asmens informacija.
Pagrindinis scenarijus:	<ol style="list-style-type: none"> 1. Atsidariusioje formoje matomi visi kontaktinio asmens informacijos laukai. 2. Užpildoma ir redaguojama informacija formoje. 3. Patvirtinami pakeitimai.
Alternatyvinis scenarijus:	Vedant netinkamą informaciją, įvedimo laukai iš karto žymimi raudonai. Nepavykus pakeisti kontaktinio asmens duomenų, priežastis pranešama vartotojui.

3.4.2. Darbuotojų modulio PA

Darbuotojų modulis - skirtas valdyti informacija apie įmonės, kuri naudosis sukurta sistema, darbuotojus, peržiūrėti darbuotojų užimtumo grafiką, skirstyti darbuotojams sistemos teises. Toliau pateikiami darbuotojų modulio panaudos atvejų aprašymai.

Lentelė 5. PA „Darbuotojų valdymas“ aprašymas

4. Panaudojimo atvejis: Darbuotojų valdymas	
Tikslas:	Suvesti, redaguoti, pašalinti duomenis apie įmonės darbuotojus.
Aktoriai:	1. Projektų vadovas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Vartotojas pasirenka naujo darbuotojo kūrimą, arba du kartus pelės pagalba spaudžia ant esamo.
Po sąlygos:	Apdorota darbuotojo informacija.
Pagrindinis scenarijus:	<ol style="list-style-type: none"> 1. Atsidariusioje formoje matomi visi darbuotojo įvedimo laukai su užpildytais turimais duomenimis. 2. Užpildomi ir redaguojami duomenys formoje. 3. Patvirtinami pakeitimai.
Alternatyvinis scenarijus:	Vedant netinkamą informaciją, įvedimo laukai iš karto žymimi raudonai. Nepavykus pakeisti darbuotojo duomenų, priežastis pranešama vartotojui.

Lentelė 6. PA „Darbuotojų užimtumo valdymas“ aprašymas

5. Panaudojimo atvejis: Darbuotojų užimtumo valdymas	
Tikslas:	Peržiūrėti ir keisti darbuotojo užimtumą.
Aktoriai:	1. Projektų vadovas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Vartotojas pasirenka darbuotoją ir spaudžią užimtumo peržiūros mygtuką.
Po sąlygos:	Peržiūrėtas ir pakeistas darbuotojo užimtumas.
Pagrindinis scenarijus:	<ol style="list-style-type: none"> 1. Atsidariusioje formoje matomas darbuotojų užimtumo grafikas, su priskirtomis užduotimis ir projektais. 2. Užvedus pelę ant užduoties galima peržiūrėti užduoties informaciją ir ją keisti, bei perstumdyti darbuotojo darbus. 3. Pasirinkus „+“ mygtuką, galima pridėti naują, jau sukurtą, bet niekam nepriskirtą užduotį. 4. Atlikus reikiamus veiksmus, uždaroma forma.
Alternatyvinis scenarijus:	Vartotojas įspėjamas apie lygiagrečias vartotojui priskirtas užduotis.

Lentelė 7. PA „Darbuotojų teisių valdymas“ aprašymas

6. Panaudojimo atvejis: Darbuotojų teisių valdymas	
Tikslas:	Priskirti ir pakeisti vartotojui sistemos rolę.
Aktoriai:	1. Projektų vadovas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Vartotojas spaudžia ant pasirinkto darbuotojo rolės stulpelio du kartus.
Po sąlygos:	Priskirta vartotojo rolę.
Pagrindinis scenarijus:	<ol style="list-style-type: none"> 1. Atsiradusiame pasirenkamajame langelyje, pasirenkama vartotojo rolę. 2. Po pasirinkimo, pasirenkamasis langelis savaime užsidaro.
Alternatyvinis scenarijus:	Vartotojui pranešama, jei nepavyko išsaugoti pasirinkimo.

3.4.3. Dokumentų modulis PA

Dokumentų modulis - skirtas saugoti failus, susijusius su projektais ir užduotimis. Įkeliami failai saugomi duomenų bazės lentelėje, dvejetainiu formatu, išsaugant kartu failo MIME tipą ir failo vardą. Įkelti failai į sistemą gali būti priskiriami keliems projektams ir kelioms užduotims. Toliau pateikiami dokumentų modulis panaudos atvejų aprašymai.

Lentelė 8. PA „Dokumentų saugojimas“ aprašymas

7. Panaudojimo atvejis: Dokumentų saugojimas	
Tikslas:	Įkelti, ištrinti failą.
Aktoriai:	1. Projektų vadovas 2. Darbuotojas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Pasirenkamas naujo failo įkėlimo mygtukas.
Po sąlygos:	Apdorota failo informacija.
Pagrindinis scenarijus:	1. Pasirenkamas failo paieškos mygtukas. 2. Kompiuteryje pasirenkamas įkeliamas failas. 3. Papildomai jei reikalinga pasirenkamas priskiriamas projektas(-ai) ar užduotis(-ys). 4. Patvirtinami pakeitimai.
Alternatyvinis scenarijus:	Vartotojas informuojamas, jei viršytas failo dydis.

Lentelė 9. PA „Dokumentų priskyrimas užduotims ir projektams“ aprašymas

8. Panaudojimo atvejis: Dokumentų priskyrimas užduotims ir projektams.	
Tikslas:	Priskirti įkelta dokumentą prie užduočių ar projektų.
Aktoriai:	1. Projektų vadovas 2. Darbuotojas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Spaudžiama du kartus ant pasirinkto failo.
Po sąlygos:	Failas priskirtas užduotims ir projektams.
Pagrindinis scenarijus:	1. Pasirenkami ir keičiami priskirti projektai. 2. Pasirenkamos ir keičiamos priskirtos užduotys. 3. Patvirtinami pakeitimai.
Alternatyvinis scenarijus:	Vartotojas informuojamas, jei saugant pasirinkimus įvyko klaida.

3.4.4. Užduočių modulis PA

Užduočių modulis - skirtas užduočių, kitiems sistemos vartotojams kūrimui, užduočių planavimui, bei viso užduoties vykdymo proceso stebėjimui. Taip pat užduotys gali būti eksportuojamos XLSX ir CSV formatais ir importuojamos, tiek naujos užduotys, tiek jau sistemoje esančios užduotys papildant jas informacija. Toliau pateikiami užduočių modulis panaudos atvejų aprašymai.

Lentelė 10. PA „Užduočių planavimas“ aprašymas

9. Panaudojimo atvejis: Užduočių planavimas	
Tikslas:	Sukurti, redaguoti užduotis ir jas priskirti vykdytojams.
Aktoriai:	1. Projektų vadovas 2. Darbuotojas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Pasirenkamas naujos užduoties kūrimas.
Po sąlygos:	Apdorota užduoties informacija.
Pagrindinis scenarijus:	1. Atsidariusioje formoje matomi visi užduoties įvedimo laukai, su užpildytais turimais duomenimis. 2. Užpildomi ir redaguojami duomenys formoje. 3. Priskiriami atsakingi asmenys. 4. Jei reikalinga, įkeliami su užduotimi susiję failai. 5. Patvirtinami pakeitimai.
Alternatyvinis scenarijus:	Vedant netinkamą informaciją, įvedimo laukai iš karto žymimi raudonai. Vartotojas informuojamas, jei viršytas įkelto failo dydis.

Lentelė 11. PA „Užduočių atlikimo stebėjimas“ aprašymas

10. Panaudojimo atvejis: Užduočių atlikimo stebėjimas	
Tikslas:	Peržiūrėti užduoties vykdymo etapus.
Aktoriai:	1. Projektų vadovas 2. Darbuotojas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Pasirenkama užduotis ir spaudžiamas istorijos peržiūros mygtukas.
Po sąlygos:	Peržiūrėta užduotis vykdymo istorija.
Pagrindinis scenarijus:	1. Atsidariusioje formoje matomi visi užduoties atlikti pasikeitimai, įvykusių etapų pradžios ir pabaigos datos. 2. Atlikus peržiūrą, uždaroma forma.

Lentelė 12. PA „Užduočių kalendoriaus peržiūra“ aprašymas

11. Panaudojimo atvejis: Užduočių kalendoriaus peržiūra	
Tikslas:	Peržiūrėti vartotojo užduočių išsidėstytą laike, pagal dienas.
Aktoriai:	1. Projektų vadovas 2. Darbuotojas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Pasirenkama kalendoriaus peržiūra.
Po sąlygos:	Peržiūrėtas kalendorius.
Pagrindinis scenarijus:	1. Atsidariusioje formoje matomas visas mėnesio vaizdas su sužymėtomis užduotimis. 2. Užvedus pelę ant užduoties, parodomas išsamesnis užduoties aprašymas. 3. Tempiant užduotį galima užduotį perkelti į kitą dieną. 4. Atlikus peržiūrą uždaroma forma.
Alternatyvinis scenarijus:	Vartotojas informuojamas, atliekant veiksmus įvyko klaida.

Lentelė 13. PA „Užduočių eksportas CSV, XLSX formatais“ aprašymas

12. Panaudojimo atvejis: Užduočių eksportas CSV, XLSX formatais	
Tikslas:	Eksportuoti užduotis CSV, XLSX formatais.
Aktoriai:	1. Projektų vadovas 2. Darbuotojas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Pasirenkamas eksporto mygtukas.
Po sąlygos:	Eksportuotos užduotys pasirinktu formatu.
Pagrindinis scenarijus:	1. Atsidariusioje eksportavimo formoje pasirenkami kriterijai, pagal kuriuos bus eksportuojamos užduotys. 2. Pasirenkamas eksportavimo formatas. 3. Spaudžiama generuoti. 4. Išsaugomas sugeneruotas failas kompiuteryje.
Alternatyvinis scenarijus:	Vartotojas informuojamas, jei generuojant failą įvyko klaida.

Lentelė 14. PA „Užduočių importas CSV, XLSX formatais“ aprašymas

13. Panaudojimo atvejis: Užduočių importas CSV, XLSX formatais	
Tikslas:	Eksportuoti užduotis CSV, XLSX formatais.
Aktoriai:	1. Projektų vadovas 2. Darbuotojas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Pasirenkamas importo mygtukas.
Po sąlygos:	Importuotos užduotys iš failo į sistemą.
Pagrindinis scenarijus:	1. Pasirenkamas failo paieškos mygtukas. 2. Kompiuteryje pasirenkamas įkeliamas failas. 3. Patvirtinami failo įkėlimas.
Alternatyvinis scenarijus:	Vartotojas informuojamas, jei nepavyko įkelti failo ar failo analizavimas buvo nesėkmingas.

3.4.5. Projektų modulio PA

Projektų modulis - skirtas kurti naujus, bei redaguoti jau esamus projektus, taip pat projektams priskirti atsakingus asmenis. Projekto peržiūra suteikia galimybę peržiūrėti dirbančių prie projekto darbuotojų užimtumą. Naudojant Ganto diagramą vaizdžiai matyti projekto užduočių išsidėstymą, bei kritinį kelią. Naudojant resursų valdymo įrankį, papildomai, prie projektų galima priskirti transporto priemonės, bei kitą bendrai naudojamą įrangą. Toliau lentelėse pateikiami projektų modulio panaudos atvejų aprašymai.

Lentelė 15. PA „Ganto diagramos peržiūra“ aprašymas

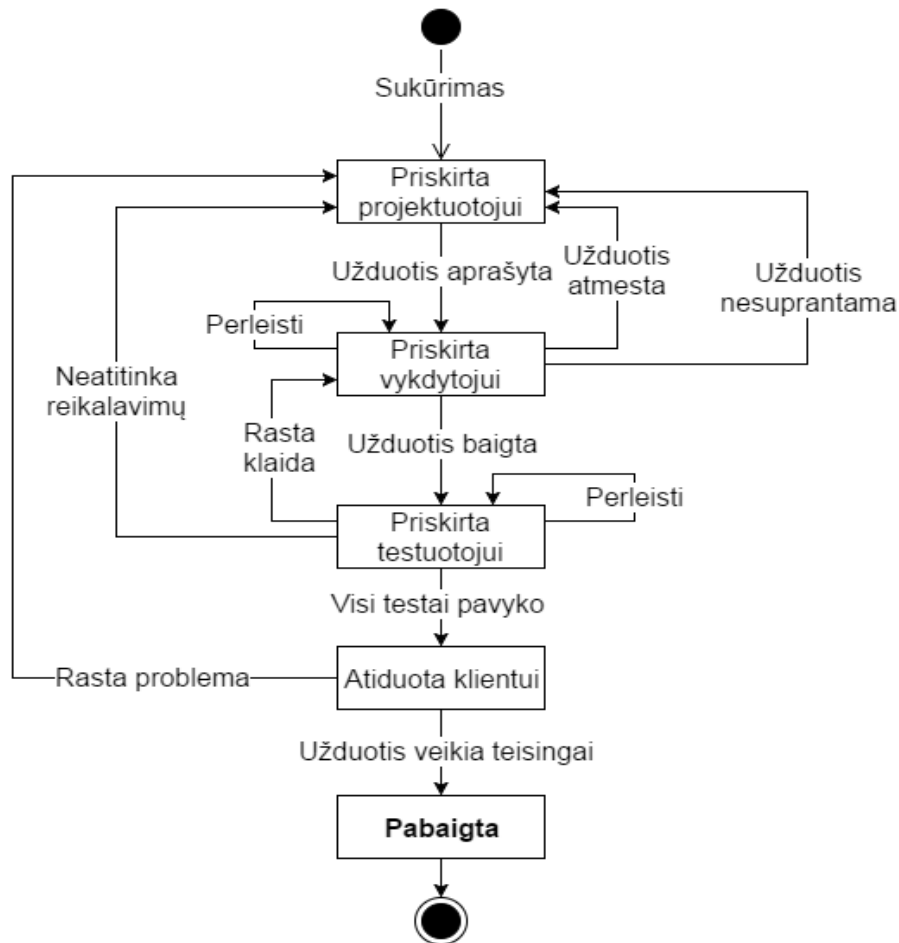
14. Panaudojimo atvejis: Ganto diagramos peržiūra	
Tikslas:	Peržiūrėti projekto kritinį kelią, naudojant Ganto diagramą.
Aktoriai:	1. Projektų vadovas 2. Darbuotojas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Pasirenkamas projektas ir spaudžiamas Ganto diagramos mygtukas.
Po sąlygos:	Peržiūrėtas kritinis projekto kelias.
Pagrindinis scenarijus:	1. Atsidariusiame lange matomas atvaizduotas projekto kritinis kelias. 2. Užvedus pelę ant vaizduojamos užduoties, galima peržiūrėti užduoties informaciją. 3. Pasirinkus užduotį, galima tempti ją į kitą diagramos vietą. 4. Atlikus peržiūrą, forma – uždara.
Alternatyvinis scenarijus:	Vartotojas informuojamas, jei nepavyko atlikti diagramos keitimo operacijos.

Lentelė 16. PA „Resursų valdymas“ aprašymas

15. Panaudojimo atvejis: Resursų valdymas	
Tikslas:	Rezervuoti projektui reikalingus resursus reikalingu metu.
Aktoriai:	1. Projektų vadovas 2. Darbuotojas
Prieš sąlygos:	Prisijungta prie sistemos.
Sužadinimo sąlygos:	Pasirenkamas projektas ir spaudžiamas resursų valdymo mygtukas.
Po sąlygos:	Pridėti resursai prie projekto.
Pagrindinis scenarijus:	1. Atsidariusiame lange matomi jau priskirti prie projekto resursai. 2. Naujose eilutės galima pasirinkti naują įmonės resursą ir resurso rezervacijos datos intervalą. 3. Jei resursas nereikalingas, galima jį pašalinti. 4. Atlikus visus pakeitimus, spaudžiamas saugojimo mygtukas.
Alternatyvinis scenarijus:	Vartotojas informuojamas, jei pasirinkti resursai persidengia su kituose projektuose naudojamais resursais.

3.5. Užduočių gyvavimo ciklas

Užduotys yra pagrindinis užduočių ir projektų valdymo sistemos objektas. Projektuojant realizuojamą sistemą, užduočių gyvavimo ciklo procesas buvo suskaidytas į penkis etapus. Užduoties gyvavimo ciklo būsenų diagrama pateikta 7 pav.



7 pav. Užduoties gyvavimo ciklas

Užduoties gyvavimo ciklą sudaro šie etapai:

- Priskirta projektuotojui – būseną, kurios metu projektuotojas projektuoja užduotį, ją aprašo ir pateikia visa reikalingą informaciją kuri bus reikalinga užduoties vykdymo metu. Projektuotojas taip pat priskiria asmenis atsakingus už užduoties vykdymą ir testavimą. Projektavimo pabaigoje, projektuotojas perduoda užduotį vykdytojui.
- Priskirta vykdytojui - būseną, kurios metu užduotis yra vykdoma. Vykdytojas, jei negali atlikti užduoties ar pagal pateiktą aprašymą jos negali suprasti, gali grąžinti užduotį projektuotojui. Taip pat užduotis gali būti perduoda kitam vykdytojui. Atlikus užduotį, užduotis perduodama testavimui.
- Priskirta testuotojui – būseną, kurios metu atliekamas užduoties testavimas. Radus klaidą, užduotis grąžinama vykdytojui. Pastebėjus jei užduotis neatitinka reikalavimų ar specifikacijos, užduotis grąžinama į projektavimo būseną. Sėkmingai atlikus testavimą, užduotis perduodama užduoties užsakovui.
- Atiduota klientui – būseną, kai užduotis atlikta, ištestuota ir atiduoda naudojimuisi užsakovui. Užsakovui pastebėjus netikslumą, užduotis grąžinama projektuotojui su pastabomis ir visas užduoties procesas pradedamas nuo pradžių. Jei užsakovo

pastebėjimų apie neteisingą sistemos veikimą nėra arba užsakovas patvirtina veikimo teisingumą – užduotis yra užbaigiama.

3.6. Sistemos projektavimas

Projektuojant sistemą, suprojektuotos paketų diagramos, veiklos diagramos, būsenų diagramos, sekų diagramos ir bendradarbiavimo diagramos, kurios nusako sistemos dinaminį, bei sisteminį vaizdą. Taip pat suprojektuotas sistemos išdėstymo vaizdas ir duomenų bazės modelio diagrama.

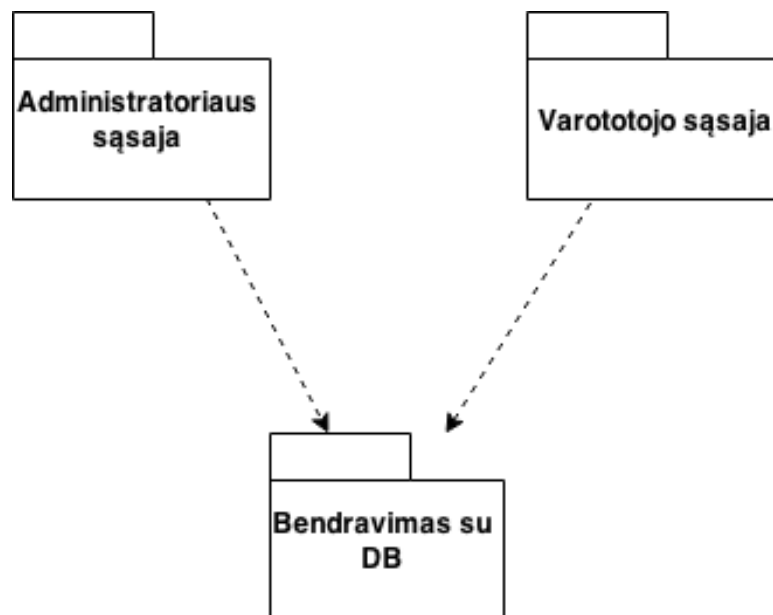
3.6.1. Sistemos architektūros tikslai ir apribojimai

Prieš pradėdant projektavimo etapą, numatyti apribojimai keliami sistemai, kurie įtakoja sistemos architektūra. Sistemos realizacijos ir architektūros sprendimus įtakoja šie reikalavimai:

1. Sistema pasiekama per internetinę naršyklę ir per mobiliuosius įrenginius, todėl naudojama kliento-serverio architektūra.
2. Internetinė vartotojo sąsaja projektuojama taip, kad atitiktų MVC architektūros standartus.
3. Duomenų saugumo užtikrinimui nuo pašalinių asmenų, naudojamas vartotojų autentifikavimas.
4. Ilgai trunkančios skaičiavimų operacijos, perkeliamos į atskirą giją arba į serverį, taip nestabdant ir nesulėtinant vartotojo darbo su sistema.
5. Internetinė sistema realizuojama naudojant „Ext JS“ karkasą.

3.6.2. Sistemos statinis vaizdas

Aukščiausiam lygyje programinė įranga suskaidyta į paketus (8 pav.).



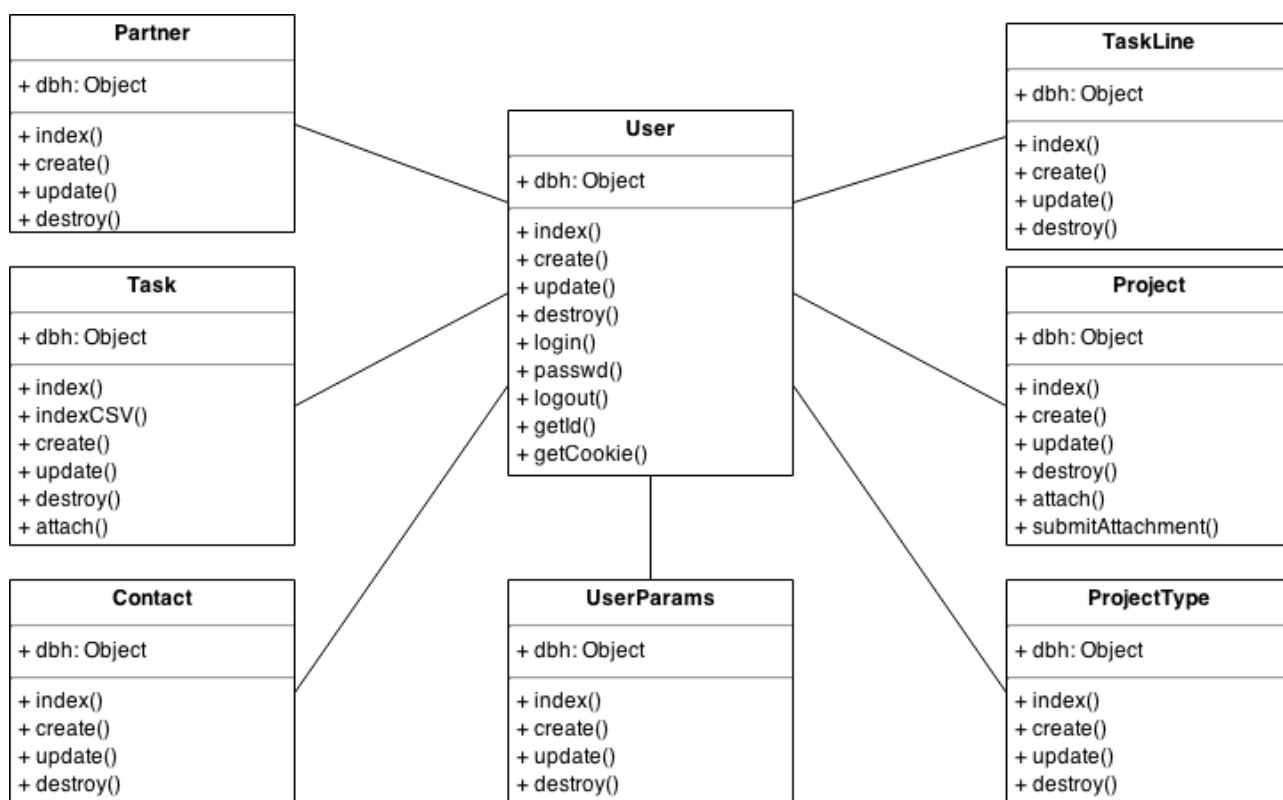
8 pav. Sistemos paketų diagrama

Sistemą sudaro trys paketai, kurie išskirsto programinę įrangą į tris posistemas:

- Bendravimas su DB – serverio pusės paketas, realizuotas REST serviso pagrindu, kuris apdoroja užklausas ir bendrauja su DB.
- Vartotojo sąsaja – darbuotojo ir projektų vadovo darbo aplinka, iš kurios vartotojai gali pasiekti jiems priskirtas panaudojimo atvejų funkcijas.
- Administratoriaus sąsaja – projektų vadovo darbo aplinka skirta valdyti kontaktų, darbuotojų ir projektų modulius.

3.6.2.1. Bendravimo su DB paketas

Bendravimo su duomenų baze paketas realizuotas PHP programavimo kalba ir atlieka vieną pagrindinį tikslą – priima iš vartotojo sąsajos HTTP užklausas, jas apdoroja ir grąžina suformuotą atsakymą JSON formatu. Šio paketų klasių diagrama pateikta 9 pav.



9 pav. Bendravimo su DB paketas

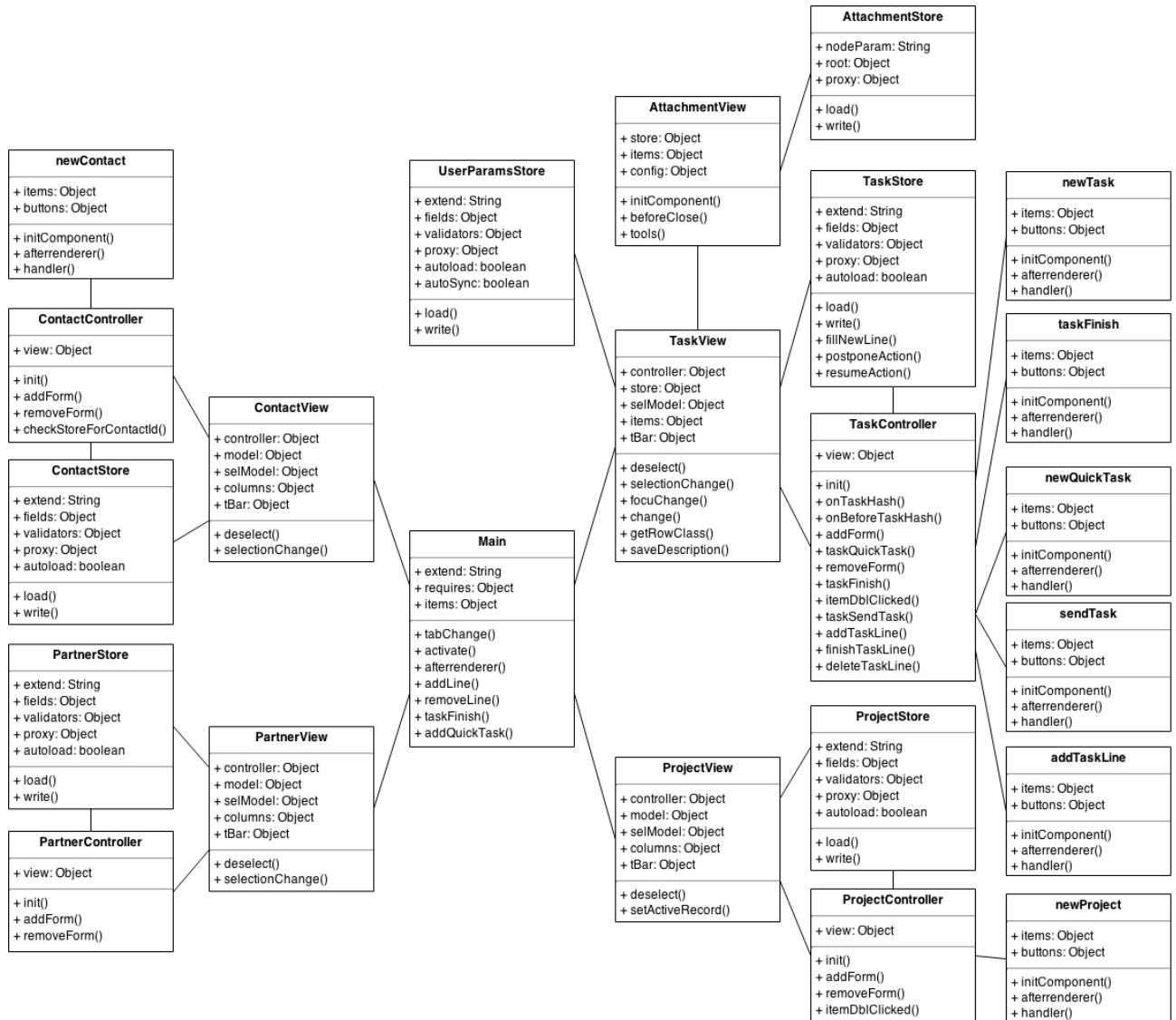
Kaip matome visos šio paketo klasės realizuoja keturi pagrindinius metodus:

- index() – atitinkamos duomenų aibės paėmimas iš duomenų bazės ir atidavimas vartotojo sąsajai.
- create() – naujo įrašo(-ų) sukūrimą duomenų bazėje realizuojantis metodas.
- update() – įrašo atnaujinimą realizuojantis metodas.
- destroy() – įrašo ištrynimą realizuojantis metodas.

Visos paketo klasės yra sujungtos su „User“ klase, tokiu būdų užtikrinamas duomenų saugumas ir duomenys yra prieinami tik autorizuotiems asmenims, kurie turi teisę pasiekti konkrečius duomenis pagal rolę priskirtą prie vartotojo.

3.6.2.2. Vartotojo sąsajos paketas

Šis paketas sukuria vartotojo aplinką, suteikiant galimybes valdyti užduotis, užduočių vykdymą, projektus, klientus, taip pat valdyti užduočių ir projektų failus, iškviešti kitas su šiais duomenimis susijusias formas. Vartotojo sąsajos paketas pateiktas 10 pav.

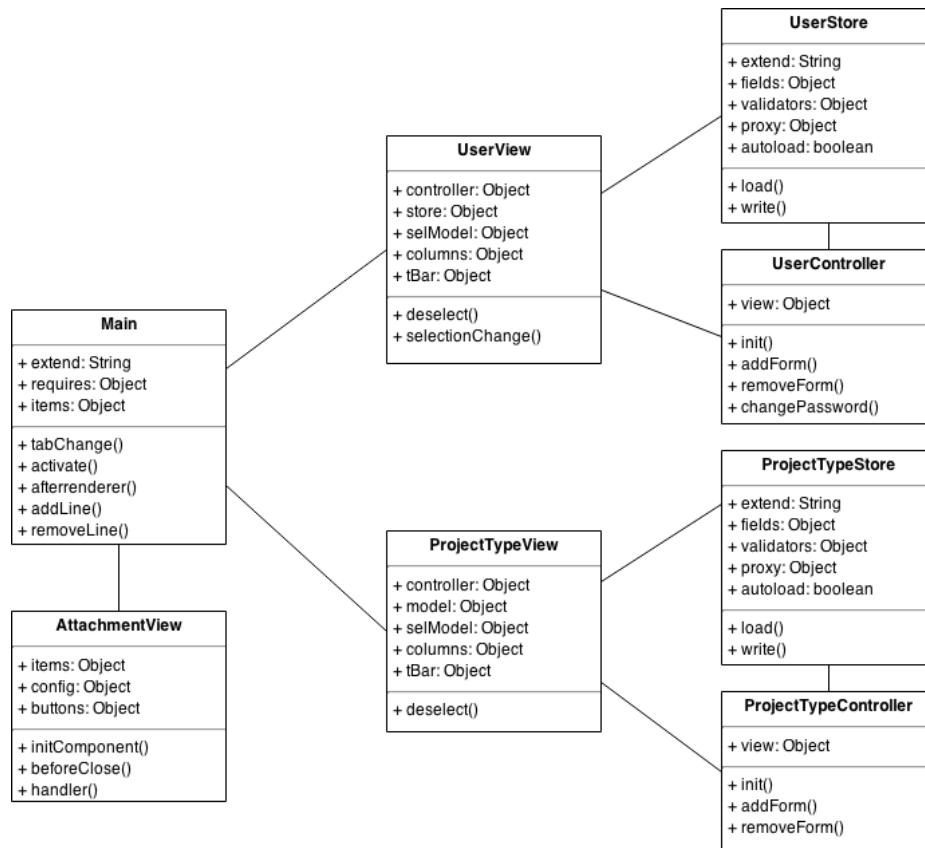


10 pav. Vartotojo sąsajos paketas

Vartotojo sąsajos paketo klasių diagramoje galime matyti naudojamą MVC architektūrą. Dauguma objektų yra sujungti grupėmis po tris, realizuojant modelio, valdiklio ir rodinio objektus. Papildomai prie valdiklių yra prijungtos kitos klasės, realizuojančios sistemos formas ir skaičiavimo klases.

3.6.2.3. Administratoriaus sąsajos paketas

Administratoriaus sąsajos paketas realizuoja projektų vadovo rolės papildomą funkcionalumą. Šio paketo funkcionalumas apima darbuotojų valdymą, eksportuojamų failų, apie užduotis ir projektus, generavimą, projektų tipų valdymą ir kitą papildomą projektų vadovo funkcionalumą. Administratoriaus sąsajos paketo realizacijos diagrama pateikta 11 pav.



11 pav. Administratoriaus sąsajos paketas

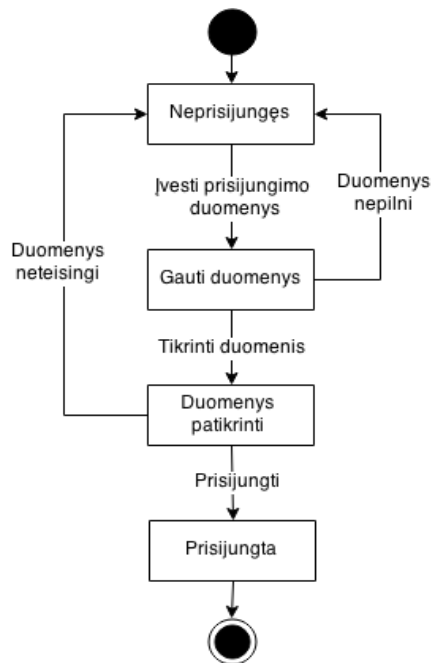
3.6.3. Sistemos dinaminis vaizdas

Šiame skyriuje pateikiamos svarbiausios sistemos dinaminio vaizdo diagramos, kurios išsamiau detalizuoja panaudos atvejus. Dinaminio sistemos vaizdo vaizdavimui naudojamos šios diagramos:

- Būsenų diagramos;
- Sekų diagramos.

3.6.3.1. Būsenų diagramos

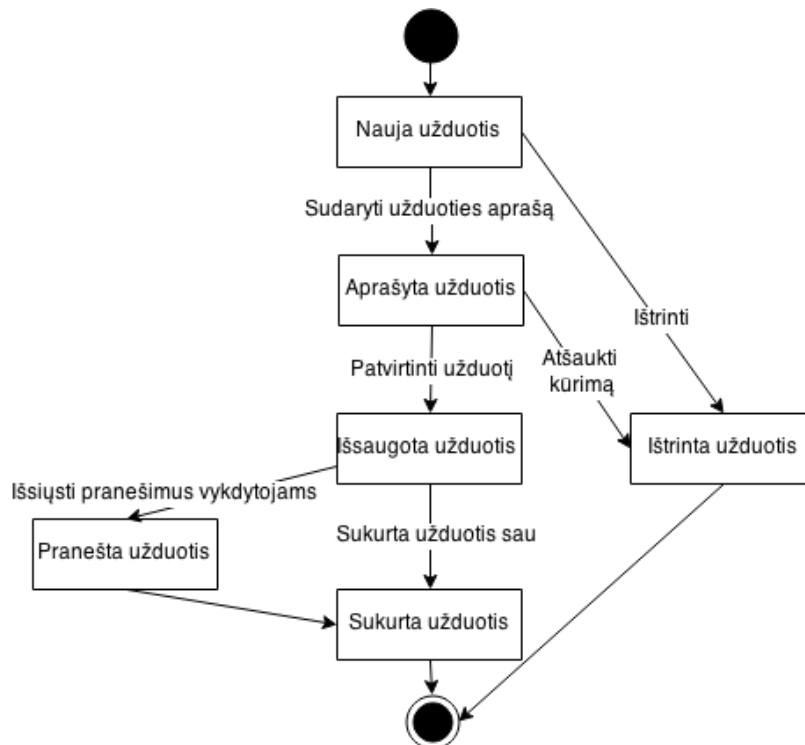
Pirmą kartą vartotojas paleidęs sistemą, turi prisijungti, su asmeniniais prisijungimo duomenimis, tam kad vartotojui sistemoje būtų priskirtos jam reikalingos funkcijos ir teisės, taip pat pateikiama tik konkrečiam vartotojui reikalinga informacija. Vartotojo prisijungimo būsenų diagrama pateikta 12 pav.



12 pav. Vartotojo prisijungimo būsenų diagrama

Kaip matome prisijungimo procesas susideda iš keturių būsenų. Pradžioje vartotojas įveda prisijungimo duomenis ir iš karto patikrinama ar įvestas ir vartotojo vardas ir slaptažodis. Jei bent vienas įvedimo laukas yra tuščias, vartotojo prašoma pakartoti įvedimą. Kai yra įvesti visi prisijungimo duomenys, duomenys yra siunčiami į serverį ir ten tikrinami. Jei duomenys yra teisingi, vartotojas sėkmingai prisijungia prie sistemos.

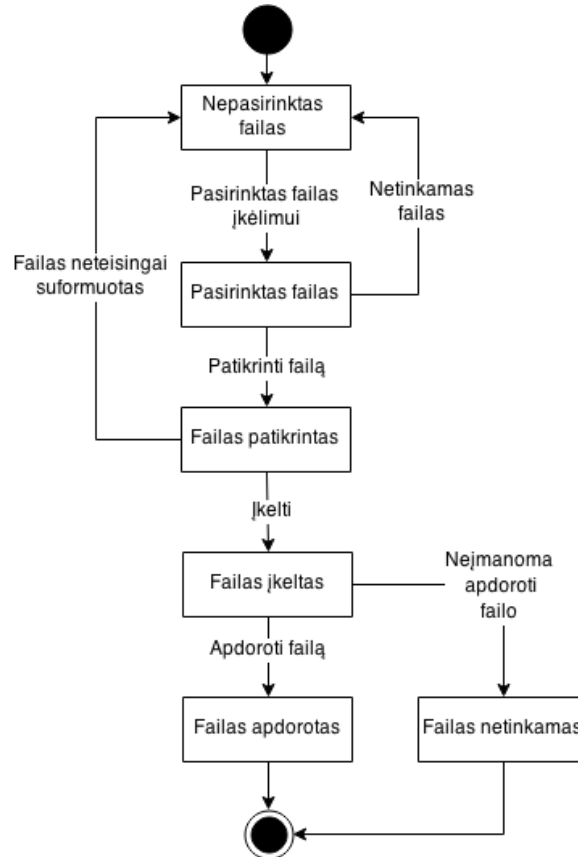
Kai vartotojas yra prisijungęs, vartotojas jau gali kurti užduotis ir jas priskirti tiek sau, tiek kitiems sistemos vartotojams. Užduoties kūrimo panaudos atvejo būsenų diagrama pateikta 13 pav.



13 pav. Užduoties kūrimo būsenų diagrama

Pradėjus kurti užduotį, užduotis turi būti aprašoma. Kai tik užduotis yra aprašyta ir yra priskirti užduoties vykdytojai, užduotis yra išsaugoma. Saugojimo metu, jei užduotis yra kuriama kitiems sistemos vartotojams, informacija apie naują užduotį yra siunčiama vartotojams elektroniniu paštu.

Užduočių failo importavimo metu, gali būti įkeliami nauji užduočių duomenys iš kitų sistemų, taip pat atnaujinamos jau sistemoje esančios užduotys. Užduočių importavimo CSV ir XLSX formatais būsenų diagrama pateikta 14 pav.

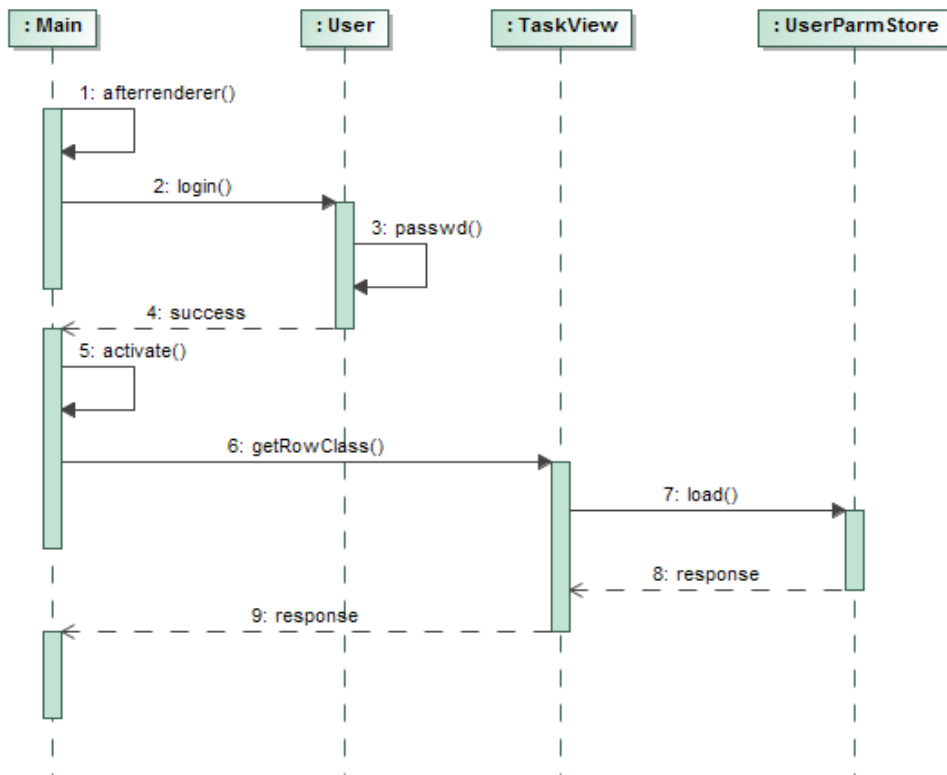


14 pav. Užduočių importavimas CSV ir XLSX formatais

Pradžioje vartotojas turi pasirinkti keliamą į sistemą failą. Kai failas yra pasirinktas, tikrinama ar jo formatas atitinka laukiamus failų formatus. Jei formatas neatitinka - vartotojas yra apie tai informuojamas. Jei failo formatas yra tinkamas, toliau tikrinama ar galime sėkmingai išanalizuoti gautą failą ir išskirti užduočių duomenis iš gauto failo. Jei failo analizavimas pavyksta sėkmingai, atnaujinamos sistemoje esančios užduotys pagal gautus naujus duomenis.

3.6.3.2. Sekų diagramos

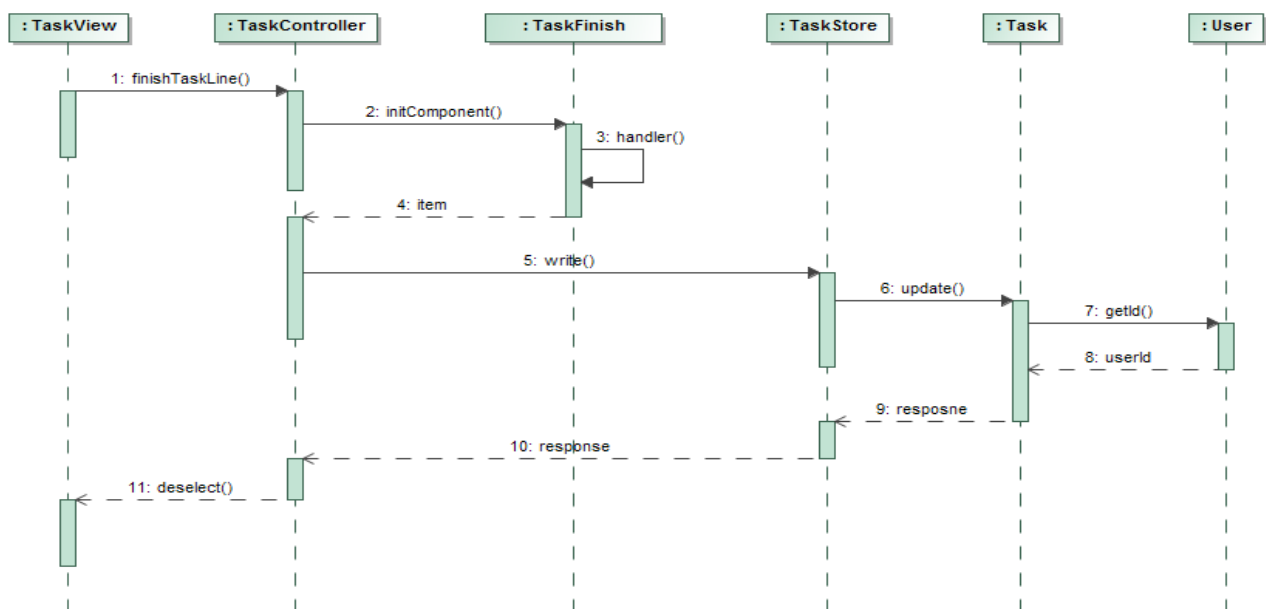
Vartotojų prisijungimo sekų diagrama, remiantis klasių diagramomis ir suprojektuotais metodais pateikta 15 pav.



15 pav. Vartotojo prisijungimo sekų diagrama

Paleidus sistemą parodoma vartotojo prisijungimo forma. Suvedus duomenis ir jungiantis prie sistemos, kreipiamasi į serverį ir patikrinami duomenys. Jei duomenys yra teisingi, vartotojas yra prijungiamas prie sistemos ir papildomai, atskira užklausa, atsiunčiami vartotojo išsaugoti parametrai.

Vartotojui sėkmingai atlikus užduotį, sistemoje tokia užduotis yra žymima kaip pabaigta arba perkeliama į sekančią užduoties gyvavimo ciklo būseną. Užduoties užbaigimo sekų diagrama, remiantis klasių diagrama, pavaizduota 16 pav.



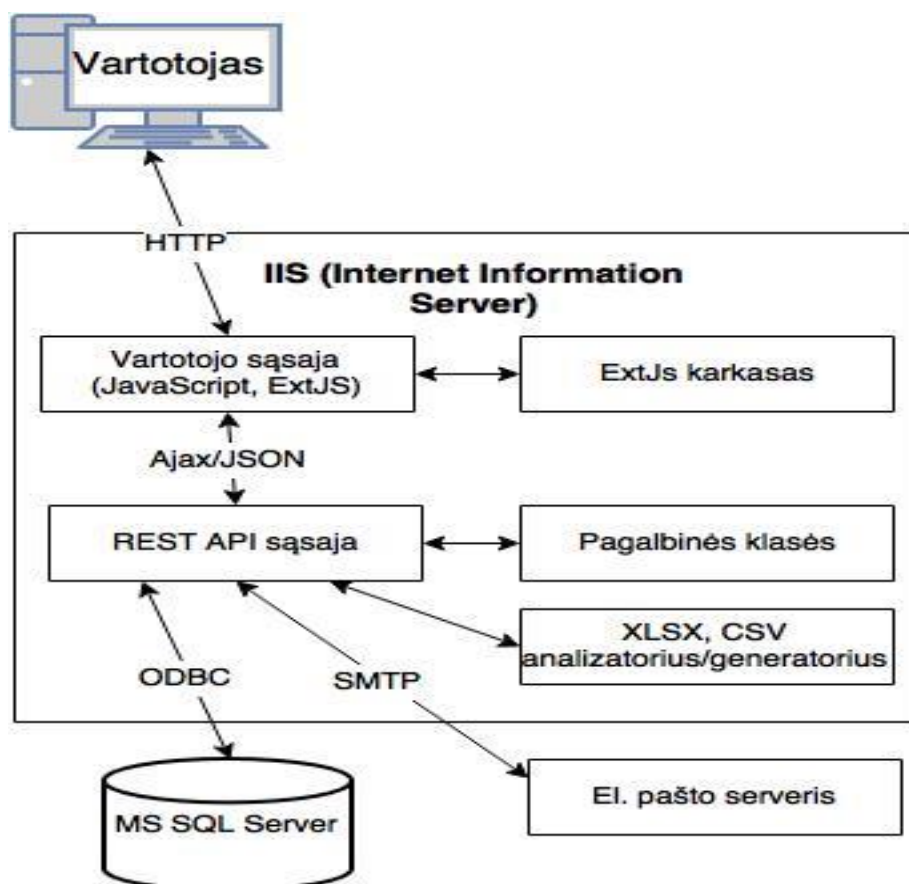
16 pav. Užduoties užbaigimo sekų diagrama

Kaip matome užbaigiant užduotį, per užduočių valdiklio klasę yra iškviečiama užduoties užbaigimo forma, kurią patvirtintus, pakeisti duomenys siunčiami į serverį. Serveryje patikrinami vartotojo duomenys ir pakeičiama užduoties būsena. Atliktus visus veiksmus, grąžinamas atsakymas vartotojui.

3.6.4. Sistemos išdėstymo vaizdas

Sistema susideda iš IIS serverio, „MS SQL Server“ duomenų bazės ir elektroninių laiškų serverio. IIS serveryje talpinamas visas programinės įrangos kodas.

Pagrindinis elementas - vartotojo sąsajos modulis, talpinamas IIS serveryje, kuris yra realizuotas „Ext JS“ karkaso pagrindu, naudojant MVC architektūra. Šis modulis jungia dviejų vartotojų rolių darbo aplinką. Panaudojus MVC architektūrą, sistema gali būti bet kada praplėsta papildomais moduliais. Taip pat, IIS serveryje talpinami „Ext JS“ karkaso pirminiai failai ir suprogramuota REST API sąsaja, kuri apima duomenų bazės valdymo metodus ir XLSX/CSV failų generavimo, analizavimo metodus. Ryšiai tarp sistemos objektų ir komunikavimo protokolai parodyti 17 pav.



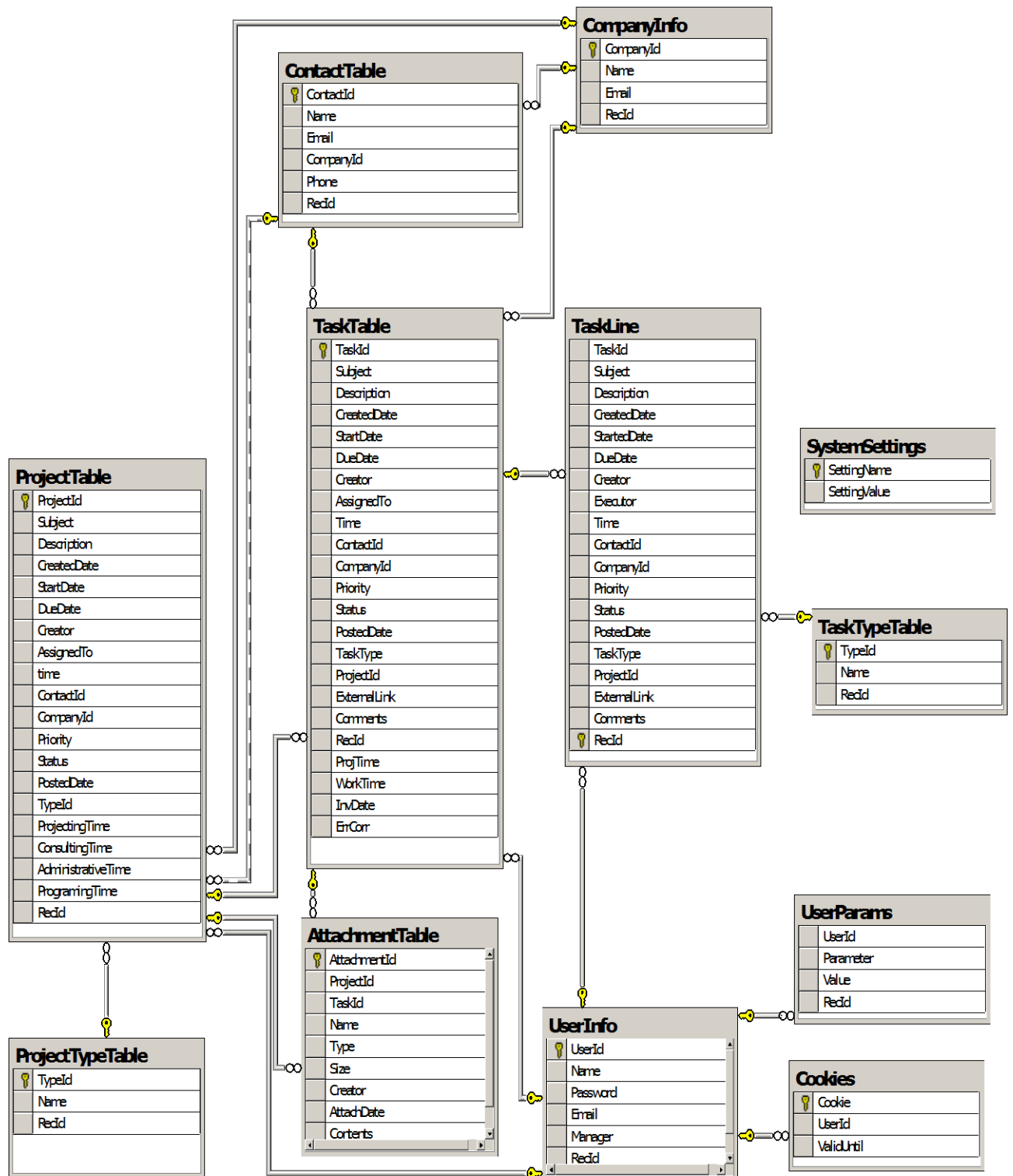
17 pav. Sistemos išdėstymo vaizdas

Vartotojas naudodamasis naršykle jungiasi prie sistemos HTTP protokolu. Vartotojo sąsaja bendrauja su serveriu naudojant AJAX užklausas, kurios leidžia atlikti kreipinius sinchroniškai.

Duomenys perduodami JSON formatu. Serveris pateikia užklausas duomenų bazei naudojant ODBC API sąsają, elektroniniai laišakai siunčiami per SMTP protokolą.

3.6.5. Duomenų vaizdas

Šiame darbe duomenų saugojimui pasirinkta „Microsoft SQL Server“ duomenų bazė. Suprojektuotos duomenų bazės modelis pateiktas 18 pav.



18 pav. Sistemos duomenų bazės vaizdas

Duomenų bazę sudaro dvylika, tarpusavyje sujungtų, lentelių, kurių kiekvienos aprašymas pateiktas 17 lentelėje.

Lentelė 17. Duomenų bazės lentelių aprašymai

Duomenų bazės lentelė	Aprašymas
CompanyInfo	Lentelė skirta saugoti sistemoje suvestas klientų įmones. Šios lentelės duomenys gali būti naudojami projektų, užduočių ir kontaktų lentelėse.
ContactTable	Lentelėje saugomi duomenys apie įmonės klientus. Kontaktinis asmuo gali būti susietas su keliais projektais, taip pat keliomis užduotimis ir tik viena įmone.
ProjectTable	Lentelėje saugoma projektų informacija. Kiekvienas projektas yra susietas su atsakingu kliento įmonės asmeniu, taip pat kiekvienas projektas gali turėti daug užduočių, bei prisegtų failų.
ProjectTypeTable	Lentelėje saugoma projektų tipų informacija.
TaskTable	Lentelė skirta saugoti užduotis. Kiekviena užduotis turi konkretų projektą, gali turėti vieną ir daugiau užduoties vykdymo pranešimų, bei vieną ar daugiau užduoties vykdytojų. Prie užduoties taip pat gali būti prisegti failai.
TaskLine	Lentelėje saugomi užduočių vykdymo įrašai.
TaskTypeTable	Lentelėje saugomi įrašai apie galimus užduočių tipus, kurie yra naudotini sistemoje.
UserInfo	Lentelėje saugoma informacija apie sistemos vartotojus. Ši lentelė siejasi su „Cookies“ ir „UserParams“ lentelėmis.
Cookies	Lentelėje saugoma informacija apie vartotojų prisijungimus.
UserParams	Lentelė, kurioje saugomi vartotojų parametrai.
AttachmentTable	Lentelėje saugomi failai, kurie buvo prisegti prie užduočių ar projektų, failo turinys saugomas „Contents“ atribute.
SystemSettings	Lentelė skirta saugoti duomenims apie sisteminius nustatymus, pvz. elektorinių laiškų siuntimo serverio nustatymai.

4. TYRIMO DALIS

4.1. Tikslas

Šio tyrimo tikslas yra nustatyti, sukurtos užduočių ir projektų valdymo sistemos, programinės įrangos našumą ir įvertinti galimus sistemos našumo patobulinimus.

4.2. Tyrimo aprašymas

Kuriant užduočių ir projektų valdymo sistemą, buvo naudojama „Ext JS 4.2.1.833“ stabili versija. Naudojantis sukurta sistema, paremta šia „Ext JS“ karkaso versija, buvo pastebėtas sistemos veikimo lėtėjimas, daugėjant duomenų kiekiui. Šiuo pagrindu buvo nuspręsta atlikti sistemos veikimo tyrimą, keičiant duomenų aibės dydį.

4.3. Tyrimui naudota techninė įranga

Tyrimui naudota aparatūrinė ir techninė įranga:

- Kompiuterio centrinis procesorius: Intel Core i5 CPU 2.53 GHz;
- Operatyvioji atmintinė: DDR3 3 GB 1333 MHz;
- Operacinė sistema: Microsoft Windows 7, 32 bitai.

Tyrimo rezultatai gali skirtis, jei tyrimai atliekami su kitokio galingumo centriniu procesoriumi ar su kitu operatyvios atmintinės tipu ar dydžiu.

4.4. Tyrimo eiga

Atliekant sistemos našumą tyrimą buvo paimtos visų (2780 užduočių) sistemoje sukurtų užduočių duomenys, iš kurių atsitiktiniu būdu buvo generuojami didesni duomenų masyvai. Atliekant tyrimą, pasirinktos duomenų eilučių imtys, buvo tokios (skliausteliuose nurodytas vidutinis sugeneruotų duomenų dydis):

- 500 eilučių (800 KB);
- 1000 eilučių (1,5 MB);
- 5000 eilučių (7,4 MB);
- 10000 eilučių (15,2 MB);
- 100000 eilučių (147 MB).

Siekiant iširti sistemos našumą, sistemai perduodamas vis augantis duomenų kiekis ir stebima kiek laiko trunka objektų su šiais duomenimis sukūrimas, bei atvaizdavimas. Pagrindinis duomenų atvaizdavimo objektas, naudojamas sistemoje, yra tinklelis. Kiekvienu atveju buvo vykdoma po 200 pakartotinių testų ir iš gautų rezultatų skaičiuojamas vidurkis. Tyrimo rezultatai pateikti 18 lentelėje.

Lentelė 18. Sistemos tyrimo rezultatai prieš patobulinimus

	Duomenų generavimo laikas, ms	Duomenų objekto sukūrimo laikas, ms	Tinklelio objekto sukūrimo laikas, ms	Duomenų atvaizdavimo laikas, ms	Viso, ms
500 eilučių	15,72	15,32	52,46	2114,75	2198,25
1000 eilučių	37,09	24,07	52,15	4180,23	4293,54
5000 eilučių	153,07	68,71	43,55	20866,47	21131,80
10000 eilučių	314,85	233,11	76,50	41653,33	42277,79
100000 eilučių	2479,27	2272,10	130,21	N/A	N/A

Pagal tyrimo rezultatus matome, kad net mažos duomenų imties atvaizdavimas trunka virš dviejų sekundžių ir trukmė auga, augant duomenų kiekiui pagal tiesinę priklausomybę, tol kol galiausiai naršyklė nesugeba įvykdyti užklauso ir nutraukia programos vykdymą.

Įvertinus gautus tyrimo rezultatus, perprogramuotas duomenų atvaizdavimo tinklelio objektas panaudojant „Ext JS“ karkaso „Buffered Renderer“ technologiją, kurios pagalba, nepaisant didelės duomenų aibės, duomenys atvaizduojami dalimis, tuo metu kai vartotojas juos peržiūri. Pritaikius šią technologiją tyrimas buvo pakartotas dar kartą, gauti rezultatai pateikti 19 lentelėje.

Lentelė 19. Sistemos tyrimo rezultatai po patobulinimo

	Duomenų generavimo laikas, ms	Ext JS duomenų objekto sukūrimo laikas, ms	Tinklelio objekto sukūrimo laikas, ms	Duomenų atvaizdavimo laikas, ms	Viso, ms
500 eilučių	18,42	16,45	55,18	946,83	1036,88
1000 eilučių	40,12	26,27	51,84	900,00	1018,23
5000 eilučių	148,78	65,23	47,21	911,14	1172,36
10000 eilučių	321,66	234,41	65,50	890,10	1511,67
100000 eilučių	2554,39	2338,91	159,78	1001,67	6054,75

Atlikus pakartotinį tyrimą ir panaudojus „Buffered Renderer“ „Ext JS“ technologiją matome, kad rezultatai žymiai pagerėjo, ir duomenų atvaizdavimo laikas svyravo apie 1 sekundę, nepaisant kokią duomenų imtį duodame sistemai. Šiuo atveju ilgesnį suminį atvaizdavimo laiką įtakojo tik duomenų generavimo trukmė ir „Ext JS“ duomenų objekto kūrimo laikas.

4.5. Tyrimo išvados

1. Tyrimo metu, atlikus sukurtos sistemos našumo bandymus, nustatyta, kad naudojama „Ext JS 4.2.1.833“ karkaso versija, lėtai apdoroja didelius (didesnius nei 500 eilučių) duomenų kiekius (500 eilučių vidutiniškai apdorotos ir atvaizduotos per 2,1 s, 100000 – 42,2 s).
2. Duomenų atvaizdavimui pritaikius „Ext JS“ karkaso „Buffered Renderer“ technologiją, tyrimo rezultatai pagerėjo 2-7 kartus (500 eilučių vidutiniškai apdorotos ir atvaizduotos per 1 s, 100000 – 6 s).

5. EKSPERIMENTINĖ DALIS

5.1. Eksperimentinio tyrimo tikslas

Eksperimento tikslas – patikrinti sukurtos sistemos našumą naudojant naujesnes „Ext JS“ karkaso versijas ir patikrinti sistemos našumą naudojant skirtingas naršykles.

5.2. Eksperimentinių tyrimų aprašymai

Eksperimentų metu, atliekamas „Ext JS“ karkaso evoliucinis našumo tyrimas, naudojant tris skirtingas sąlygas:

1. Duomenų atvaizdavimo našumo eksperimentas, kintant duomenų eilučių skaičiui.
2. Duomenų atvaizdavimo našumo eksperimentas, kintant duomenų stulpelių skaičiui.
3. Sistemos našumo eksperimentas naudojant skirtingas naršykles.

Pirmi du eksperimentai atliekami naudojant tik „Opera“ naršyklę, paskutinis – atliekamas naudojant „Opera“ ir kitas populiarias naršykles, kad būtų galima palyginti rezultatus nepriklausomai nuo naršyklių „JavaScript“ variklių.

Visi eksperimentai atlikti naudojant tris paskutines „Ext JS“ karkaso stabilias laidas, kurios yra platinamos nemokamai:

- „Ext JS 4.2.1.833“ – išleista 2013 m. gegužės 16 d.
- „Ext JS 5.1.1“ – išleista 2015 m. gegužės 7 d.
- „Ext JS 6.0.1“ – išleista 2015 m. rugsėjo 8 d.

5.3. Našumo eksperimentas kintant duomenų eilučių skaičiui

Šio eksperimento metu, atliekamas pakartotinis tyrimas aprašytas skyriuje 4.4. „Tyrimo eiga“, tiriant duomenų atvaizdavimo našumą naudojant skirtingas „Ext JS“ karkaso versijas. „Ext JS 4.2.1.833“ versijos tyrimas jau buvo atliktas 19 lentelėje, todėl papildomai atliekame tik „Ext JS 5.1.1“ ir „Ext JS 6.0.1“ versijų tyrimus. Tyrimų rezultatai pateikti 20 ir 21 lentelėse.

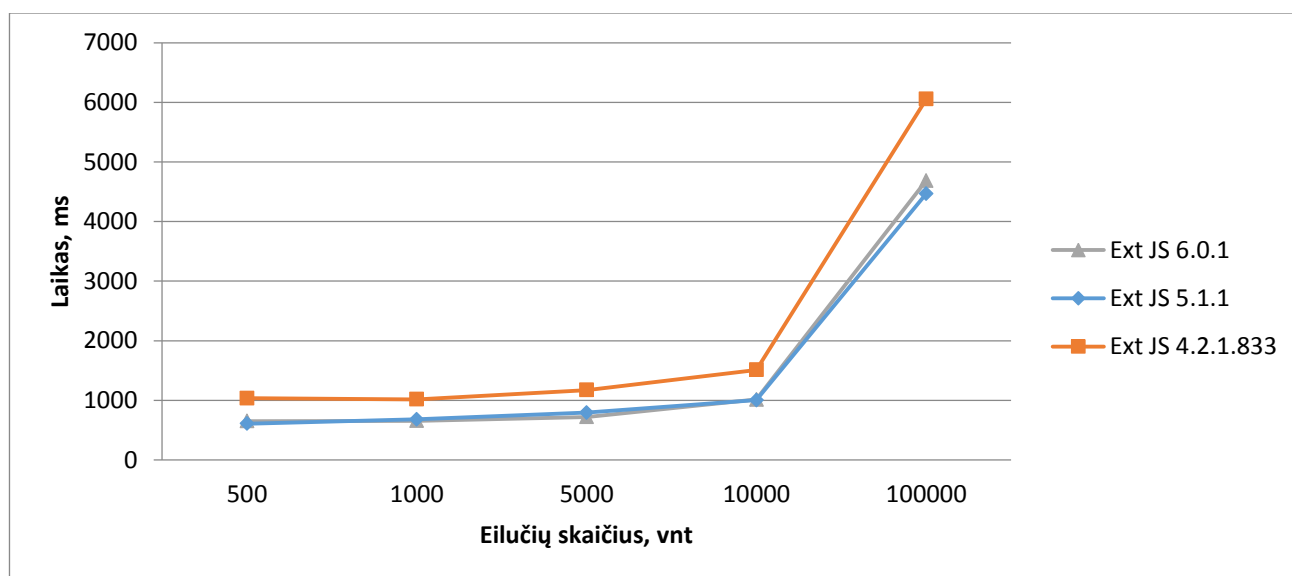
Lentelė 20. „Ext JS 5.1.1“ našumo tyrimas

	Duomenų generavimo laikas, ms	Ext JS duomenų objekto sukūrimo laikas, ms	Tinklelio objekto sukūrimo laikas, ms	Duomenų atvaizdavimo laikas, ms	Viso, ms
500 eilučių	10,76	7,24	336,41	254,84	609,25
1000 eilučių	35,68	8,19	258,41	380,34	682,62
5000 eilučių	142,10	27,55	240,84	383,91	794,40
10000 eilučių	278,28	75,33	422,27	227,06	1002,94
100000 eilučių	2826,44	993,21	490,30	155,83	4465,78

Lentelė 21. „Ext JS 6.0.1“ našumo tyrimas

	Duomenų generavimo laikas, ms	Ext JS duomenų objekto sukūrimo laikas, ms	Tinklelio objekto sukūrimo laikas, ms	Duomenų atvaizdavimo laikas, ms	Viso, ms
500 eilučių	15,41	7,66	393,47	235,83	652,37
1000 eilučių	26,15	17,71	443,85	169,05	656,76
5000 eilučių	120,17	39,93	378,27	184,64	723,01
10000 eilučių	295,11	129,57	456,04	132,96	1013,68
100000 eilučių	2178,53	1808,48	548,53	151,07	4686,61

Atlikus eksperimentą, matome, kad našumo rezultatai yra maždaug pusę sekundės geresni naujesnėse „Ext JS“ karkaso versijose. Taip pat matome kad naujesnėse „Ext JS“ versijose, buvo perkurtas tinklelio objektas, nes tinklelio objekto sukūrimo trukmė žymiai padidėjo, bet duomenų atvaizdavimo trukmė atitinkamai sumažėjo. „Ext JS“ karkaso evoliucinio tyrimo palyginimas pagal suminį atvaizdavimo laiką, kai kinta duomenų eilučių skaičius, pavaizduotas 19 pav.

**19 pav.** „Ext JS“ karkaso evoliucinis našumo tyrimas, pagal kintantį duomenų eilučių skaičių

Pagal gautą palyginimo grafiką matome, kad „Ext JS 6.0.1“ ir „Ext JS 5.1.1“ versijos našumu prilygsta viena kitai, tuo tarpu „Ext JS 4.2.1.833“ versija yra akivaizdžiai lėtesnė.

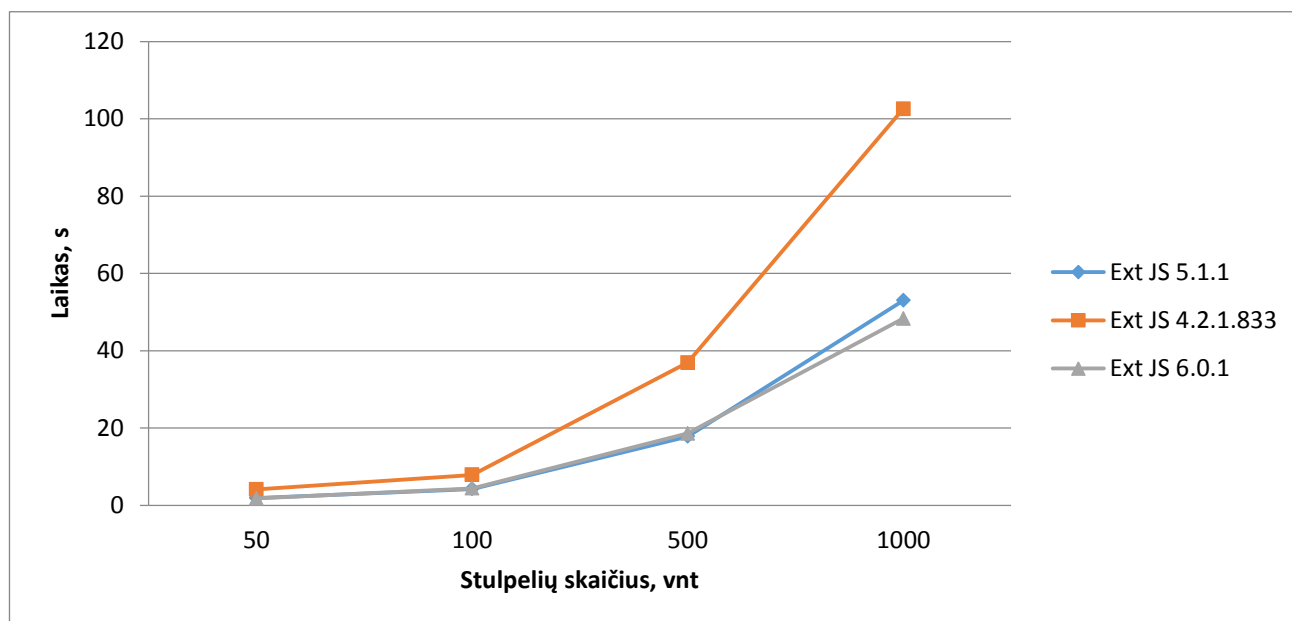
5.4. Našumo eksperimentas kintant duomenų stulpelių skaičiui

Šiam tyrimui, kurio metu buvo tiriama duomenų atvaizdavimo greitis, pagal didėjantį duomenų stulpelių skaičių, buvo paimta vienoda, 1000 eilučių, duomenų imtis ir keičiamas tik atvaizduojamų duomenų stulpelių skaičius šiais dydžiais: 50, 100, 1000. Eksperimento rezultatai pateikti 22 lentelėje.

Lentelė 22. „Ext JS“ našumo tyrimas, kintant duomenų stulpelių skaičiui

	Vykdymo laikas pagal karkaso versiją, s		
	Ext JS 4.2.1.833	Ext JS 5.1.1	Ext JS 6.0.1
50 stulpelių	4,105	1,853	1,785
100 stulpelių	7,876	4,196	4,389
500 stulpelių	36,862	17,807	18,566
1000 stulpelių	102,530	52,984	48,323

Šio eksperimento rezultatai patvirtino ankstesnio tyrimo našumo rezultatus ir parodė, kad tirtos naujesnės „Ext JS“ karkaso versijos yra našesnės, nei sena „Ext JS“ karkaso versija. Taip pat matome, kad „Ext JS“ karkasas nėra pritaikytas atvaizduoti duomenis kurie turėtų daug įrašo atributų. Grafiškas palyginimas, kaip kinta vykdymo laikas pagal stulpelių skaičių pavaizduotas 20 pav.



20 pav. „Ext JS“ karkaso evoliucinis našumo tyrimas, pagal kintantį duomenų stulpelių skaičių

Pagal gautą karkaso versijų palyginimo grafiką matome, kad „Ext JS 4.2.1.833“ versija yra du kartus lėtesnė, nei „Ext JS 5.1.1“ ir „Ext JS 6.0.1“ versijos.

5.5. Sistemos našumo eksperimentas naudojant skirtingas naršyklės

Siekiant, kad tyrimo rezultatai būtų tikslesni ir nepriklausomi nuo pasirinktos naršyklės naudojimo, papildomai atliktas tyrimas, įvykdant eksperimentus naudojant 1000x100 (1000 duomenų eilučių, 100 duomenų stulpelių) duomenų imtį naudojant šias populiariausias naršyklės:

- „Opera“ (versija 37.0) – „Carakan“ „JavaScript“ variklis;
- „Google Chrome“ (versija 50.0) – „V8“ „JavaScript“ variklis;

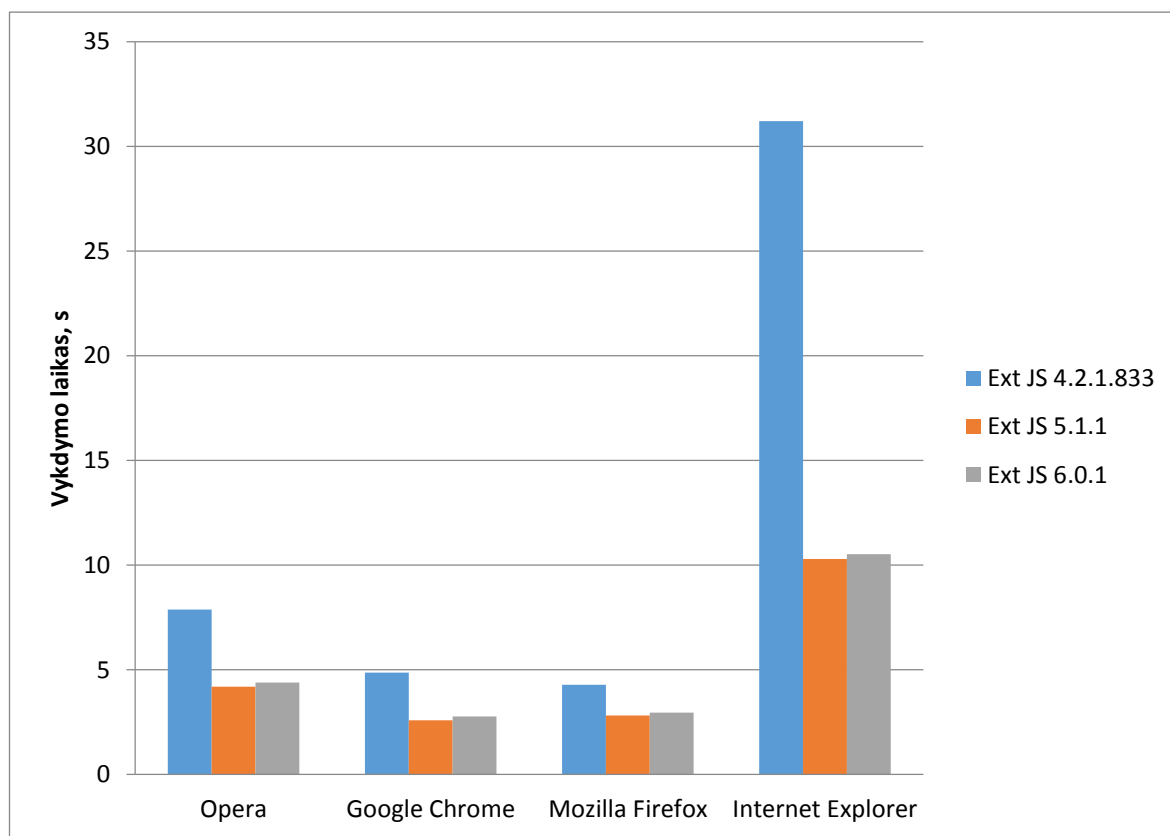
- „Mozilla Firefox“ (versija 46.0.1) – „SpiderMonkey“ „JavaScript“ variklis;
- „Internet Explorer“ (versija 11) – „Chakra“ „JavaScript“ variklis.

Atliktų eksperimentų rezultatai pateikti 23 lentelėje.

Lentelė 23. Eksperimento rezultatai naudojant skirtingas naršykles

	Vykdymo laikas pagal karkaso versiją, s		
	Ext JS 4.2.1.833	Ext JS 5.1.1	Ext JS 6.0.1
Opera	7,876	4,196	4,389
Google Chrome	4,864	2,579	2,767
Mozilla Firefox	4,287	2,814	2,948
Internet Explorer	31,201	10,290	10,513

Pagal gautus rezultatus matome, kad „Ext JS“ karkaso 4.2.1.833 versija išlieka lėčiausia naudojant visas tirtas naršykles. Naršyklių palyginimo diagrama pateikta 21 pav.



21 pav. „Ext JS“ našumo kitimas, naudojant skirtingas naršykles

Palyginus visų tirtų naršyklių našumą, nustatyta kad „Internet Explorer“ naršyklė lėčiausiai apdorojo „JavaScript“ programinį kodą. „Google Chrome“ ir „Mozilla Firefox“ naršyklių našumas skyrėsi minimaliai ir buvo geriausi. „Opera“ naršyklė eksperimento metu buvo perpus lėtesnė nei „Google Chrome“ ir „Mozilla Firefox“ naršyklės.

5.6. Eksperimento rezultatai

1. Atlikus eksperimentus ir atlikus rezultatų analizę, nustatyta, kad „Ext JS“ karkaso 4.2.1.833 versija yra 2 kartus lėtesnė, nei naujos „Ext JS“ karkaso versijos, todėl siekiant geresnio sistemos našumo, tikslinga naudoti „Ext JS 5.1.1“ arba „Ext JS 6.0.1“ karkaso versiją.
2. Atlikus „Ext JS“ karkaso, tinklelio objekto eksperimentus ir rezultatų tyrimą, nustatyta, kad augant duomenų imčiai, duomenų atvaizdavimo laikas kinta nežymiai. Nustatyta, kad atvaizduojant 500-100000 eilučių duomenų imtis, duomenų tinklelio objekto sukūrimo laikas svyruoja tarp 400-500 ms, o duomenų atvaizdavimo laikas svyruoja tarp 100-200 ms.
3. Atlikus naršyklių greitaveikos eksperimentus, nustatyta kad „Google Chrome“ ir „Mozilla Firefox“ naršyklės greičiausiai apdoroja „JavaScript“ programinį kodą ir tuo pačių greičiausiai vykdo „Ext JS“ karkaso programas. Tuo tarpu „Opera“ naršyklė yra du kartus lėtesnė, nei „Google Chrome“ ir „Mozilla Firefox“ naršyklės, o „Internet Explorer“ naršyklės našumas buvo lėčiausias (lėtesnis maždaug 3 kartus).

6. IŠVADOS

1. Nustatyti sistemai keliami reikalavimai, kurių pagrindu buvo realizuota internetinė daugiaplatformė užduočių ir projektų valdymo sistema.
2. Realizuota užduočių ir projektų valdymo sistema sudaryta iš 5 modulių – kontaktų modulio, darbuotojų modulio, dokumentų modulio, užduočių modulio ir projektų modulio.
3. Nustatyta, kad pritaikius „Ext JS“ karkaso „Buffered Renderer“ technologiją, didelio kiekio duomenų atvaizdavimas pagreitėjo 2-7 kartus (500 eilučių vidutiniškai apdorotos ir atvaizduotos per 1 sekundę, 100000 – 6 sekundes).
3. Nustatyta, kad „Ext JS“ karkaso 4.2.1.833 versija yra 2 kartus lėtesnė, negu „Ext JS 5.1.1“ arba „Ext JS 6.0.1“ karkaso versijos, todėl siekiant geresnio sistemos našumo, pasirinkta „Ext JS 5.1.1“ karkaso versija.
4. Nustatyta, kad „Google Chrome“ ir „Mozilla Firefox“ naršyklės greičiausiai apdoroja „JavaScript“ programinį kodą ir tuo pačių greičiausiai vykdo „Ext JS“ karkaso programas. „Opera“ naršyklė yra maždaug du kartus lėtesnė, nei „Google Chrome“ ir „Mozilla Firefox“ naršyklės, o „Internet Explorer“ naršyklės našumas buvo lėčiausias (lėtesnis maždaug 3 kartus).
5. Sukurta sistema sėkmingai įdiegta dvejose įmonėse ir yra aktyviai naudojama.
6. Šiame darbe atliktas tyrimas buvo pristatytas XXI tarpuniversitetinėje magistrantų ir doktorantų konferencijoje „IVUS 2016“ ir išspausdintas šios konferencijos pranešimų medžiagos leidinyje.

7. LITERATŪRA

- [1] Hai Eric Lam ir Piyush Maheshwari. Task and Team Management in the Distributed Software Project Management Tool, 2001, 401 – 408p. ISBN 0-7695-1372-7.
- [2] „iTeamWork“. Prieiga internete: <http://www.iteamwork.com> [Žiūrėta 2014 11 21].
- [3] „onProject“. Prieiga internete: <http://www.onproject.com> [Žiūrėta 2014 11 21].
- [4] Trautmann, N., Baumann, P., „An iterative backward/Forward Technique for the scheduling of resource-constrained projects within Microsoft Project“, Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on, 2010 gruodžio 7 – 10 d., ISBN 978-1-4244-8501-7.
- [5] A. Lodhi, Usability Heuristics as an assessment parametre: For performing Usability Testing. Software Technology and Engineering (ICSTE), 2010 2nd International Conference on, 2010 Spalio 3-5 d, ISBN 978-1-4244-8667-0.
- [6] J. Nielson. „What is Usability“. Usability Engineering, 1993, 26p., ISBN 0-12-518406-9
- [7] Jakob Nielsen . „Ten usability heuristics“, 2005, ISSN 1548-5552.
- [8] J. Nielson. „Usability Assessment Methods beyond Testing“. Usability Engineering, 1993, 207 - 209p., ISBN 0-12-518406-9.
- [9] Valentino Lee, Heather Schneider, and Robbie Schell. „Mobile Applications: Architecture, Design, and Development“. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004. ISBN:0-13-117263-8.
- [10] Majeed Ahmad. „Smartphone: Mobile Revolution at the Crossroads of Communications, Computing and Consumer Electronics.“ CreateSpace, 2011. ISBN 1-46-103315-2.
- [11] Hernandez, I.M.T., Viveros, A.M., Rubio, E.H., „Analysis for the design of open applications on mobile devices“, Electronics, Communications and Computing (CONIELECOMP), 2013 International Conference on, 2013 Kovo 11-13 d, ISBN 978-1-4673-6156-9.
- [12] Pei Zheng and Lionel Ni. „Smart Phone and Next Generation Mobile Computing (Morgan Kaufmann Series in Networking (Paperback))“. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. ISBN 0-12-088560-3.
- [13] D. Gavalas and D. Economou. „Development platforms for mobile applications: Status and trends“. Software, IEEE, 28(1), 77 – 86 p., 2011 Sausis-Vasaris. ISSN 0740-7459.
- [14] „Windows Phone API reference,, 2014 liepos 21 d. Prieiga internete: [http://msdn.microsoft.com/library/windows/apps/ff626516\(v=vs.105\).aspx](http://msdn.microsoft.com/library/windows/apps/ff626516(v=vs.105).aspx) [Žiūrėta 2014 11 29].

- [15] Anthony I. Wasserman. „Software engineering issues for mobile application development”. In Proceedings of the FSE/SDP workshop on Future of software engineering research, FoSER'10, pages 397-400, New York, NY, USA, 2010. ACM. ISBN: 978-1-4503-0427-6.
- [16] Qi Zhang, Lu Cheng, and Raouf Boutaba. „Cloud computing: stateof- the-art and research challenges”. Journal of Internet Services and Applications, 7 - 18 p., 2010. ISSN 1867-4828.
- [17] Roy Thomas Fielding, Richard Taylor, „Principled design of the modern Web architecture”. Software Engineering, 2000, pp. 407-416.
- [18] Roy Thomas Fielding, „Architectural Styles and the Design of Network-based Software Architectures“. Doctoral dissertation, University of California, Irvine, 2000.
- [19] Sun Zhaoyun, Zhang Xiaobo, Zhao Li, „The Web asynchronous communication mechanism research based on Ajax”. Education Technology and Computer (ICETC), 2010, pp. 370-372.
- [20] Rosalind Rogoff, „Making electronic information accessible to everyone”. Professional Communication Conference, 2001, pp. 231-236.
- [21] Liu Xudong, Xu Xiaofei, Zhan Dechen, He Ting, „A Novel Approach for Developing Web-Based Enterprise Information Systems”. Computer Science and Software Engineering, 2008, pp. 548-551.
- [22] Rest server. Prieiga internete: <https://github.com/jacwright/RestServer> [Žiūrėta 2016 03 15] .

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

API (angl. Application Programming Interface) – aplikacijų programavimo sąsaja.

CSV (angl. comma separated values) – failų formatas, skirtas saugoti duomenis lentelėms.

DB – duomenų bazė.

DSPMTool (angl. Distributed Software Project Management Tool) – paskirstytos programinės įrangos projekto valdymo įrankis.

HTTP (angl. Hypertext Transfer Protocol) – standartizuotas protokolas pasiekti informaciją pasauliniame tinkle.

JSON (angl. JavaScript Object Notation) - atviro formato standartas kuris skirtas perduoti duomenų objektus poromis: atributas – reikšmė.

PA – Panaudos atvejis.

SDK (angl. software development kit) – programinės įrangos kūrimui skirtas priemonių rinkinys.

MVC (angl. Model-View-Controller) - sistemos architektūra, atskirianti programos dalis pagal funkcionalumą ar poreikius.

XAML (angl. Extensible Application Markup Language) – skelbiamoji XML pagrindu parašyta žymėjimo kalba.

9. PRIEDAI

9.1. Diegimą patvirtinanti aktas

UAB „GM Consult Group“

KAM: Kauno technologijos universitetui

UŽDUOČIŲ IR PROJEKTŲ VALDYMO SISTEMOS DIEGIMĄ PATVIRTINANTIS AKTAS

2016-05-17
Kaunas

Patvirtiname, kad UAB „GM Consult Group“ įmonėje buvo įdiegta Šarūno Slanio sukurta užduočių ir projektų valdymo sistema.

UAB „GM Consult Group“
Direktorius

Marijus Vaičys



UAB „GM Consult Group“
Savanorių pr. 284, LT-49475, Kaunas

Research and implementation of task and project management system based on Ext JS framework

Šarūnas Slanys
Kaunas University of Technology
Kaunas, Lithuania
sarunas.slany@gmail.com

Šarūnas Packedvičius
Kaunas University of Technology
Kaunas, Lithuania
sarunas@iee.org

Abstract — Task and project management is an important task of each company. Such companies need task and project management software to efficiently manage processes, focused on the successful, rapid and high quality implementation of the project. This paper researches tasks and projects management system software design and development process, in order to give the company an innovative, quick and user-configurable system. The system provides functions such as management of company's employees, task scheduling, employee employment tracking, task calendar, document management, etc. Task and project management system can reduce enterprise project management costs, improve project quality and increase enterprise productivity. Based on the Ext JS framework and MVC architecture, implemented system provides responsive user interface and feature-rich cross-platform web application.

Keywords — Task management; Project management; MVC; Ext JS; REST;

I. INTRODUCTION

At present, companies offering employees a flexible work schedule or the ability to work in a comfortable and always changing workplace, encounter problems, how to manage employees employment and maintain the company's productivity [1]. Successful and comfortable work planning and distribution of work is an important task of each company. Such companies need task and project management software to effectively manage processes, focusing on the unique achievement of the objective (e.g. the creation of a product or service) that has a beginning and an ending, and limited resources [2]. All business processes must be managed at the lowest possible cost to manage projects, tasks, documents, and other daily events taking place in the company.

This paper analyses the task and project management system development, using MVC (Model-View-Controller) architecture. This architecture allows separating presentation, content and logic layers to adapt to the complex user interface design requirements for the development of an online web system [3]. REST service coupled with the MVC architecture simplifies the query communication between the user and the server-side part, increases query and programming code readability and legibility of the modules, simplifies the system by providing simpler system scalability.

Ext JS framework provides comprehensive and convenient data management functionality that implements asynchronous communication between the client requests and

server responses reduces the queries response and data display time [4]. Sending requests in JSON format, allows saving transferred data traffic.

II. SYSTEM REQUIREMENTS ANALYSIS AND FEATURES

A. System requirements analysis

Along with the fast pace of work and urgent task changes the system must be integrated into other systems used by users on various convenient ways. Sending emails to user mailboxes and messages to consumers' mobile devices reduces the time from the point of task submission to the moment when task actually reaches a consumer.

Task and project management system has two kinds of user roles: project managers and tasks executors. Project managers have the highest rights and have the ability to manage the company's database of contacts, projects, employees, also create tasks to other users, manage tasks files, export / import tasks in XLSX, CSV format. Also, project manager's role inherits all rights of task executor. Task executor role dedicated to the assigned tasks preview, tasks execution or submitting additional information about the task. Tasks are divided into five stages: design, execution, testing, delivered to the client and completed. Notifications about task creation or changes of the task status are sent by e-mail with all required information about the task.

B. System features

According to given requirements for the system there is a need for internet based application designed for task and project management through a web browser. To create system there is necessary to implement these packages: user interface module, a project manager interface module, email sending module and communication with the database module. Users can access system only after successful authorization with the personal data in order to store all users performed actions in the database. As long as users have to use the system on a daily basis, there is implemented an intuitive user interface that allows users to arrange objects according to their needs to reach system comfortability and usability. System features design is showed in Figure 1.

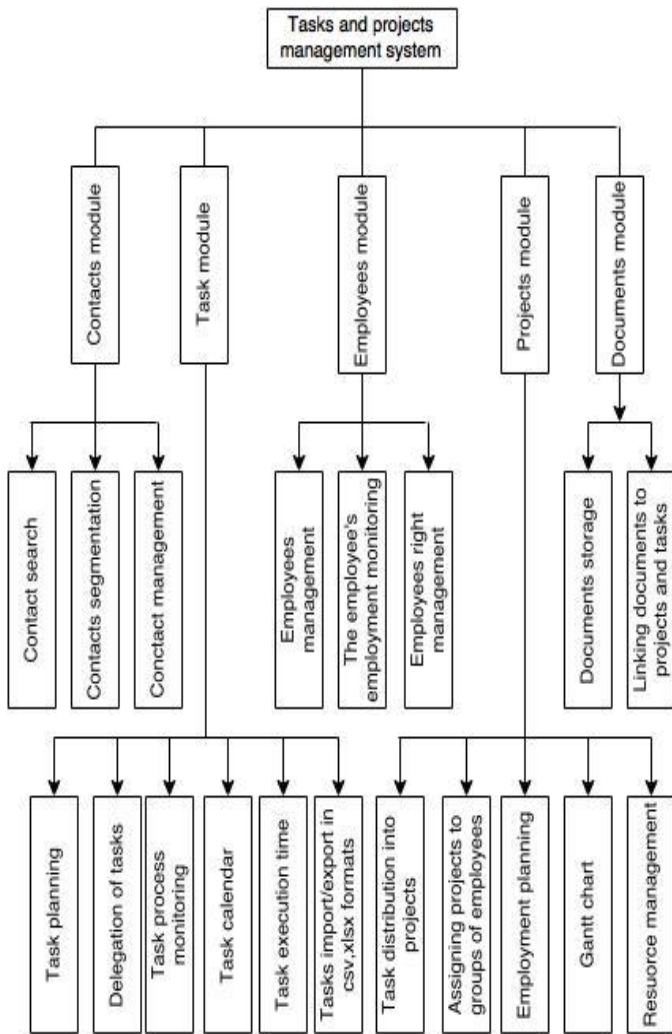


Figure 1. System features design

According to Figure 1 system can be distinguished into five user interface modules: contacts module, tasks module, employees module, projects module, documents module.

III. TECHNOLOGY SELECTION

A. MVC achitecture

MVC operation is based on the separation of concerns (SoC) principle [5]. This means that the program is decomposed into the different parts that their functionality or requirements do not overlap. This principle is applied in many places. One example would be HTML and CSS. HTML is designed to specify the document structure while CSS defines how the document should look like. In object-oriented programming languages this principle is reflected by splitting tasks into objects and their methods.

The principle of the MVC gets the separation of logic, presentation and content layers. The three parts of MVC are responsible for their own activity:

- *Model* – data management or application unit of certain area.

- *View* – gives the appearance of a program model suitable for the user to understand (forms, lists, etc.). Web application is nothing more than the generated HTML / CSS code.
- *Controller* – responsible for the model management and view changes according to appropriate user interaction like button press, text entry.

B. ExtJS framework

Ext JS framework is an open source JavaScript library that provides extensive component library which makes it possible to use asynchronous Ajax requests [6], DOM, CSS, HTML and other technologies for building feature-rich cross-platform Web applications targeting the desktop, tablets, and smartphones. Ext JS leverages HTML5 features on modern browsers while maintaining compatibility and functionality for legacy browsers.

Ext JS provides hundreds of high-quality UI widgets that are meticulously designed to meet both the needs of the easiest web applications, as well as the most complex web applications. Ext JS templates and layout manager gives complete control of the screen, regardless of the devices and screen sizes. Advanced charting package allows visualize large amounts of data. This system includes a robust data packet that can use data from any backend data source. Ext JS also offers a number of out-of-the-box themes and full theming support. Ext JS also includes the availability of a package (ARIA) for people with disabilities to help Section 508 compliance [7].

There are many different JavaScript frameworks available. JavaScript MVC architecture frameworks comparisons, according to important features which are needed for task and project management system development are shown in table 1.

JAVASCRIPT FRAMEWORKS COMPARISON

	Angular.js	ExtJs	Backbone	React	Ember
UI binding	✓	✓		✓	✓
Reusable components	✓	✓	✓	✓	✓
Routing	✓	✓	✓		✓
Composed views		✓			✓
Grid	✓	✓			
Rich text editor		✓		✓	
Charts		✓			
Cross broswer support	✓	✓	✓	✓	✓
Offline storage	✓	✓			
Developer tools		✓			

Ext JS framework chosen to establish a system adapted for convenient everyday use, depending on the usability of the system. Ext JS framework provides many objects that can be modified according to each user's needs. Integrating some of these objects to task and management system we obtain an intuitive, user-configurable user interface.

C. REST server

Representational State Transfer (REST) is architectural style for distributed hypermedia systems. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements [8, 9].

This paper uses simplified open source REST service designed for PHP programming language, which was written by Jacob Wright [10].

REST is a way of formatting the URL of tasks and projects management web application. For example, if application is sending HTTP GET request to server with URL *domain.com/tasks* – server will return JSON object containing data about all tasks. Also if we are sending HTTP GET request with *domain.com/tasks/:id* – server returns only one task by given ID. In the same way we can use HTTP POST requests to create new records, HTTP PUT requests to update record by given ID and HTTP DELETE requests to delete record by given ID.

D. Other used technologies

Other technologies and the programming language that was used during the development of the task and project management system are listed below:

- *Programming languages* – PHP 5.5.10, JavaScript
- *Web server* – Internet Information Services (IIS) 7
- *Database* – Microsoft SQL server 2008
- *Operating system* – Microsoft Windows Server 2008 Service Pack 2

IV. DEVELOPED SYSTEM

Task and project management system goal – improve enterprise tasks management process. At any time, while using system we can see currently running processes in the company, state of running tasks and their progress. Also the employment of employees can be evenly distributed.

A. System design

The system consists of the IIS server, SQL Server database and mail server. In the IIS server there is placed all software code.

A key element of the system is placed in the IIS server - the user interface module based on Ext JS framework and using the MVC architecture combines the working environment of two user roles. Using that kind of the architecture of the system can always be extended with additional modules. Also, in the IIS server hosted Ext JS framework source files and programmed REST API, which includes database management methods and XLSX / CSV file

generation, parsing methods. Connections between system object structure is showed in Figure 2.

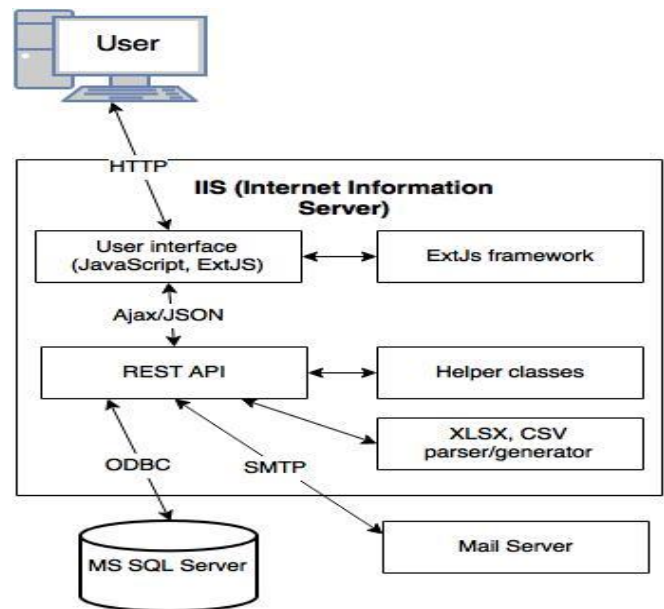


Figure 2. Connections between system objects

User using computer browser connects to the system via HTTP protocol. The user interface communicates with the server using Ajax requests, which enables to make queries asynchronously. The data transmitted in the JSON format. The server provides query the database using ODBC API, emails sent via the SMTP protocol.

B. Task lifecycle

Tasks are the main object of tasks and projects management system. Task lifecycle and state diagram is showed in Figure 3.

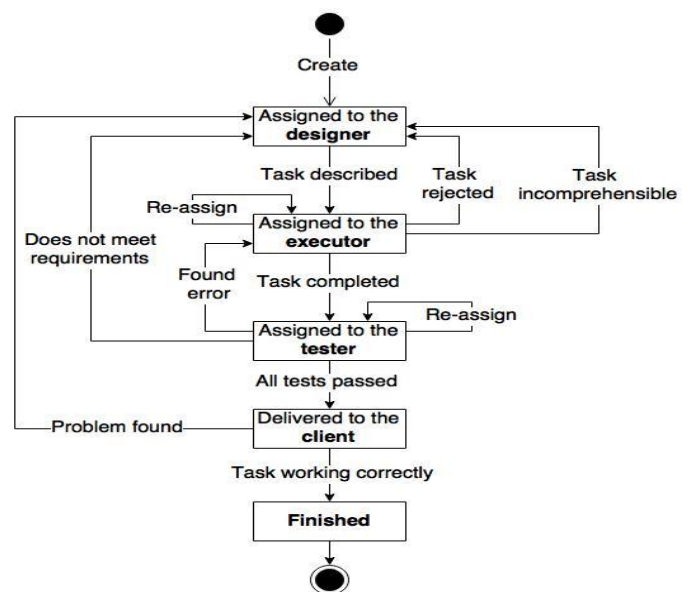


Figure 3. Task lifecycle diagram

In this paper task lifecycle is divided into five stages:

- *Assigned to the designer* – status in which the designer designs the task to extract and provide all the necessary information about a task that will be required during the execution of the task. The designer also assigns a person responsible for the task execution and testing. At the end of the design, the designer transmits task to executor.
- *Assigned to the executor* – state in which the task is executed. If executor cannot perform the task or task is incomprehensible according to the description then it can be returned to design status. Also, the task can be re-assigned to another employee who can complete the task. If the task has been completed successfully, then task can be transferred to the testing stage.
- *Assigned to the tester* – status in which task testing is performed. If error is found during testing, the task is returned to the executor. If there is observed that the task does not meet requirements or specifications, the task is returned to the design state. After successful testing process, the task is being delivered to customer.
- *Delivered to the client* – state when the task has been completed, tested and given to the client for further use. If customer notices inaccuracies, the task is returned to design stage with customer comments and all the task process starts from the beginning. If the customer does not provide observations about the incorrect functioning or customer confirms the correctness of the task operation – the task is completed.
- *Finished* – state when task is completed and marked as finished. Finished tasks are stored in the system database, in order to display project and task history and statistics related to task.

V. CONCLUSIONS

- Developed task and project management system allows companies manage and monitor the company's task and project management process, improving the traceability of tasks and increasing the company's productivity.
- According to requirements developed an online system, accessible through a computer, tablet or smartphone web browser with maintaining compatibility and functionality for legacy browsers.
- The system is realized based on Ext JS framework using the MVC architecture, which realizes the feature-rich cross-platform web application. Feature-rich widgets provide users comfortable responsive workplace.
- In this paper, the task lifecycle is divided into five main stages: design, execution, testing, delivered to client and finished. Every task stage have assigned

responsible person, who is responsible for particular stage quality.

- A system can be used in various companies for tasks employment planning, tasks process monitoring, performed work history trail. Email services inform system users about new tasks and task status changes, to ensure faster task information delivery.

REFERENCES

- [1] Gamze Karayaz, Charles B. Keating, Morgan Henrie, „Designing Project Management Systems“. System Sciences(HICSS), 2011, pp. 1-10.
- [2] Rahat Iqbal, Richard Rider, Nazaraf Shah, Anne James, „Project management system review and redesign using user-centred design methodology“. Computer Supported Cooperative Work in Design (CSCWD), 2011.
- [3] Hamid Mcheick, Yan Qi, „Dependency of components in MVC distributed architecture“. Electrical and Computer Engineering (CCECE), 2011, pp. 139-145.
- [4] Libin Yang, Yilai Zhang, Hailin Li, „Information Search Based on Test Mining and EXTJS“. Software Engineering, 2009, pp. 386-391.
- [5] Liu Xudong, Xu Xiaofei, Zhan Dechen, He Ting, „A Novel Approach for Developing Web-Based Enterprise Information Systems“. Computer Science and Software Engineering, 2008, pp. 548-551.
- [6] Sun Zhaoyun, Zhang Xiaobo, Zhao Li, „The Web asynchronous communication mechanism research based on Ajax“. Education Technology and Computer (ICETC), 2010, pp. 370-372.
- [7] Rosalind Rogoff, „Making electronic information accessible to everyone“. Professional Communication Conference, 2001, pp. 231-236.
- [8] Roy Thomas Fielding, Richard Taylor, „Principled design of the modern Web architecture“. Software Engineering, 2000, pp. 407-416.
- [9] Roy Thomas Fielding, Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000.
- [10] Rest server. Internet access: <https://github.com/jacwright/RestServer>