

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Tomas Kartašovas

**PROJEKTŲ VALDYMO SISTEMŲ ATVIRO KODO
SISTEMŲ BAZĖJE EFEKTYVUMO DIDINIMAS,
PANAUDOJANT VIZUALIUS KOMPONENTUS**

Magistro baigiamasis darbas

Darbo vadovas: doc. Eimutis Karčiauskas

Kaunas, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Tomas Kartašovas

**PROJEKTŲ VALDYMO SISTEMŲ ATVIRO KODO
SISTEMŲ BAZĖJE EFEKTYVUMO DIDINIMAS,
PANAUDOJANT VIZUALIUS KOMPONENTUS**

Magistro baigiamasis darbas

Recenzentas

doc. Aleksas Riškus

2011-05-

Vadovas

doc. Eimutis Karčiauskas

2011-05-

Atliko

IFM-9/2 gr. stud.

Tomas Kartašovas

2011-05-

Kaunas, 2011

TURINYS

1. ĮVADAS	12
2. PROJEKTŲ VALDYMO SISTEMŲ ANALIZĖ	14
2.1. Problema.....	14
2.2. Projektų valdymo prototipų apžvalga	14
2.3. Pagrindinis projektų valdymo sistemų apibendrintas funkcionalumas	17
2.4. Projektų valdymo PĮ tikslai	17
2.5. Situacijos Lietuvoje įvertinimas.....	18
2.6. Projektų sėkmės faktoriai	18
2.7. Projektų valdymo sistemų klasifikacija.....	18
2.8. Pasirinktas programinės įrangos kūrimo procesas	19
2.8.1. Žiniatinklio tipo programinės įrangos privalumai:	19
2.8.2. Žiniatinklio tipo programinės įrangos trūkumai:	19
2.9. Pasirinkta realizavimo architektūra	20
2.10. Projekto koncepcinė schema	21
2.11. Išvados.....	21
3. PROJEKTŲ VALDYMO SISTEMOS PROTOTIPAS	21
3.1. Sistemos tikslai.....	21
3.2. Projektų valdymo PĮ esybių analizė	22
3.3. Bendri apribojimai.....	25
3.4. Funkciniai reikalavimai.....	26
3.4.1. Veiklos kontekstas	26
3.4.2. Veiklos padalinimas	26
3.5. Papildomi reikalavimai	27
3.6. Projektavimas	27
3.7. Naudojamos technologijos	27
3.8. Duomenų bazė.....	28
3.9. Programavimo kalbos.....	28
3.10. Web Serveris	29
3.11. Papildomi įrankiai	29

3.12. Diegimo aplinka	30
3.13. Bendradarbiaujančios sistemos	30
3.14. Numatoma darbo vietos aplinka.....	30
3.15. Svarbūs faktai ir prielaidos.....	30
3.15.1. Svarbūs faktai.....	30
Kuriama sistema bus pateikta kaip atviro kodo programinė įranga visiems norintiems ja naudotis.	30
3.15.2. Prielaidos.....	30
3.16. Vartotojai.....	30
3.16.1. Vartotojo charakteristikos	30
3.16.2. Vartotojo problemos	31
3.16.3. Vartotojo tikslai.....	31
3.17. Projekto išėja	31
3.18. Egzistuojantys sprendimai.....	31
3.18.1. Pagamintos sistemos, kurios gali būti panaudotos	31
3.18.2. Pagaminti komponentai, kurie gali būti panaudoti	31
3.19. Panaudojimo atvejai	33
3.20. Reikalavimai duomenims.....	34
3.21. Galimas pakartotinas panaudojimas.....	34
3.22. Loginis vaizdas.....	34
3.23. Komponentų detalizavimas	35
3.23.1. Sistemos branduolys	35
3.23.2. Administravimo komponentas	36
3.23.3. Užduočių tvarkymo komponentas.....	37
3.23.4. Priminimų komponentas	37
3.23.5. Duomenų bazės valdymo komponentas.....	37
3.23.6. Projektų tvarkymo komponentas.....	38
3.23.7. Resursų tvarkymo komponentas	38
3.23.8. Grafikų piešimo komponentas	39
3.23.9. Žinučių komponentas	39
3.23.10. Ataskaitų generavimo komponentas	39
3.23.11. Interneto naršyklė.....	39

3.24. Procesų vaizdas	40
3.24.1. Sistemos veikimas.....	40
3.24.2. Vartotojo veiksmus naudojantis šia sistema apsprendžia run controleris:	40
3.24.3. Projektų tvarkymas	41
3.24.4. Užduočių tvarkymas	42
3.24.5. Sistemos vartotojų administravimas	43
3.24.6. Sistemos administravimas.....	43
3.25. Išdėstymo vaizdas	44
3.26. Kokybė	45
3.27. Testavimas.....	45
3.27.1. Testavimo tikslai ir objektai.....	45
3.27.2. Testavimo apimtis	46
3.27.3. Pagrindiniai apribojimai.....	46
3.27.4. Strategija	47
3.27.5. Vienetų testavimas	47
3.27.6. Integravimo testavimas	48
3.27.7. Priėmimo testavimas	48
3.27.8. Aukšto lygio testavimas	48
3.27.9. Testavimo resursai	48
3.27.10. Testavimo įrankiai ir aplinka	48
3.28. Vartotojo dokumentacija	49
3.28.1. Sistemos struktūra.....	49
3.28.2. Vartotojo instrukcija	50
3.28.3. Diegimas	50
3.28.4. Klaidos	50
3.28.5. Autorių teisės ir gretutinės teisės	50
3.28.6. Pakeitimų sąrašas	50
4. TYRIMAS	51
4.1. Pagrindinis projektų valdymo sistemos funkcionalumas.....	51
4.2. Mano siūlomas projektų valdymo sistemos meniu	51
4.2.1. Apžvalga	52
4.2.2. Meniu punktas Projektai.....	54

4.2.3.	Menu punktas Užduotys	57
4.2.4.	Menu punktas Bendradarbiavimas	66
4.2.5.	Menu punktai: resursai, ataskaitos, administravimas.....	67
4.3.	Svarbiausi aspektai meniu punktų grafinių atvaizdavimų aprašymuose.....	68
4.4.	Vizualinių komponentų svarba.....	68
5.	EKSPERIMENTAS.....	69
5.1.	Eksperimentas patvirtinantis, kad didelio funkcionalumo PĮ su prasta vartotojo sąsaja ir skurdžiais vizualiniais komponentais nėra tinkamas šio tipo įrankio realizavimo būdas.....	69
5.2.	Praktinėje dalyje įgyvendinto funkcionalumo tinkamumo patikrinimas	70
5.3.	Netinkamai realizuoti vizualiniai sprendimai ir jų galimi pataisymai	74
6.	IŠVADOS.....	76
7.	TERMINŲ ŽODYNAS.....	77
8.	LITERATŪROS SĄRAŠAS.....	79
9.	PRIEDAI	81

Paveikslukai

Paveikslukas 1. Programų realizavimo architektūra	19
Paveikslukas 2 Konceptinė projekto schema	21
Paveikslukas 3 Veiklos kontekstas	26
Paveikslukas 4 Panaudojimo atvejai	33
Paveikslukas 5 Duomenų modelis	34
Paveikslukas 6 Loginis vaizdas	35
Paveikslukas 7 Sistemos branduolys	36
Paveikslukas 8 Kalendoriaus komponento klasių diagrama	36
Paveikslukas 9 Duomenų bazės valdymo komponento klasių diagrama	38
Paveikslukas 10 Sistemos būsenos	40
Paveikslukas 11 Projektų valdymo diagrama	41
Paveikslukas 12 Užduočių valdymo diagrama	42
Paveikslukas 13 Vartotojų administravimo diagrama	43
Paveikslukas 14 Sistemos administravimo diagrama	43
Paveikslukas 15 Sistemos išdėstymas	44
Paveikslukas 16 Sistemos failų išdėstymas	45
Paveikslukas 17 Basecamp'q pradinis puslapis	53
Paveikslukas 18 Siūlomos gantt tipo diagramos modifikacijos pavyzdys	56
Paveikslukas 19 Projektų peržiūra	57
Paveikslukas 20 Užduoties lauko pavyzdys	59
Paveikslukas 21 Darbų eigos atvaizdavimo pavyzdys	60
Paveikslukas 22 Vieno projekto darbų eigos pavyzdys	62
Paveikslukas 23 Darbų eigos šablono sudarymo pavyzdys	63
Paveikslukas 24 Sudėtingo šablono sudarymo pavyzdys	64

Paveiksliukas 25 Minčių dėstymo pavyzdys	66
Paveiksliukas 26 Pirmo lygio pasirinkimo meniu pavyzdys.....	67
Paveiksliukas 27 Užduočių atvaizdavimas „Žmonės” būdu	71
Paveiksliukas 28 Užduoties perskyrimas	72
Paveiksliukas 29 Užduoties spalvos pasikeitimas.....	72
Paveiksliukas 30 Projektų medžio vaizdas	74
Paveiksliukas 31 Projekto užduočių terminų modifikavimas	75

Lentelēs

Lentelē 1 Veiklos padalinimas.....	27
------------------------------------	----

Santrauka

Pastaruoju metu ypatingai išaugo interneto naršyklių galimybės, kurias suteikia Javascript kalbos bibliotekos, bei css3 ir html5 technologijos. Tai skatina didesnių galimybių vizualinių komponentų kūrimą sistemoms, kuriomis vartotojai naudojami interneto naršyklės pagalba. Šiame darbe aprašoma realizuota PĮ, kuri ypatingą dėmesį skiria darbų paskirstymui pagal turimus išteklius, pasinaudojant vizualiniai komponentais. Planavimo etapas daro didžiausią įtaką projekto biudžetui, vykdymo terminui, pasiektų rezultatų kokybei. Tinkamas darbų paskirstymas yra tinkamas plano sudarymas. Projekto planavimas yra sunkus uždavinys, kurį galima palengvinti pasitelkiant šiuolaikinės technologijas. Panaudoti komponentai aiškiau atvaizduoja pateiktą informaciją, suteikia patogesnę būdą atlikti tam tikrus veiksmus. Pagal PĮ vartotojo kategoriją skiriasi PĮ vartotojo tikslai. Todėl planavimo etapo palengvinimui pritaikyti komponentai išskirti pagal PĮ vartotojų kategorijas. Sukurta PĮ leidžia lengviau kontroliuoti projekto planavimo eigą.

Tomas Kartašovas. Open source project management system productivity increase methods using visual components: Master's work in science of informatics supervisor doc. E. Karčiauskas; Department of Software Engineering, Faculty of Informatics, Kaunas University of Technology. – Kaunas, 2011.

Summary

These days web browsers retain the unbelievable possibilities with html5, css3 and javascript. This promotes increased opportunities for the development of visual components for systems in which users interact with web browser. This work describes the implemented software that works with special attention to the allocation of the available resources, using the visual components. The planning phase of the project has the greatest impact on the budget, period of performance and the quality of the results. The proper division of labor is appropriate plan. Project planning is a difficult task, which can be facilitated by means of modern technology. Used components provides more convenient way to carry out certain actions. According to different categories of software user, different and software user goals. Therefore, the planning stage to facilitate the adaptation of software components identified under the categories of users providing different user interface and different visual components. Developed software makes it easier to control the project planning process.

1. ĮVADAS

Vieningas, struktūrizuotas ir stebimas procesas – tik tokio proceso dėka projektų valdymas bus efektyvus. Projektų valdymas, ypač programinės įrangos kūrimo, yra sudėtingas procesas, apimantis projekto užsakovus, vykdytojus ir projekto procesų paskirstymą tarp vykdytojų.

Projektų valdymo programinė įranga turi užtikrinti darnų projekto vykdytojų darbą su minimaliomis laiko sąnaudomis, maksimaliai efektyviai paskirstant projekto veiklas.

Tikslas - sukurti projektų valdymo sistemą, pateikiamą informaciją ir galimybes ją valdyti suteikiančią vizualių komponentų pagalbą. Tai būtų komponentų rinkinys, organizuojantis projekto procesą. Išanalizuoti rinkoje esančias tokio pobūdžio sistemas, panaudoti jų teigiamiausius bruožus ar net atvirojo kodo komponentus kuriamoje PĮ. Projektą įgyvendinti naudojantis tik nemokamais, užsibrėžtą funkcionalumą įgyvendinančiais komponentais ir įrankiais. Kuriama sistema leis efektyviau panaudoti turimas galimybes. Visų pirma vartotojai nori bendrauti su patogią ir aiškia vartotojo sąsają turinčią PĮ ir peržiūrėti nesudėtingai atvaizduojamą informaciją. Tuo tikslu bus parinkta labiau draugiška vartotojo sąsaja, patogesni ir funkcionalesni informacijos vizualizavimo komponentai tiek administravimo posistemei, tiek vartotojo posistemei.

Tyrimo objektas – projektų valdymo įrankio efektyvumo didinimas vizualiais komponentais.

Aktualumas – daugelio atvirojo kodo projektų valdymo sistemų vartotojo sąsajos neišbaigtos, neatkreiptas dėmesys į vizualinius komponentus, ypač į jų atnaujinimą, koncentruojantis tik išgauti reikiamą funkcionalumą projektų valdymo procesuose. Tokių PĮ įrankių daug, tačiau jų naudojimas nėra žymus.

Šiame darbe buvo išskelti šie uždaviniai:

- Egzistuojančių programinės įrangos kūrimo procesų analizė ir ją palaikančios programinės įrangos analizė.

- Pasirinkto programinės įrangos kūrimo proceso analizė, esminių savybių išskyrimas. Detalus planavimo esybių išskyrimas.
- Projektų valdymo sistemos prototipo sukūrimas, kuris remtųsi pasirinktu programinės įrangos kūrimo proceso modeliu.
- Eksperimentas, skirtas patikrinti projektuojamos sistemos pagrįstumą.

Pirmoje darbo dalyje nagrinėjami: egzistuojantys sprendimai, apibendrinamos vyraujančios tendencijos projektų valdyme, išskiriami programinės įrangos kūrimo procesų modeliai, klasifikacija, pagrindžiamas tokių sistemų būtinumas, poreikis, apžvelgiamos sukurtos sistemos, išskiriamos pagrindinės jų savybės ir klasifikacija, atvirojo kodo ir mokamų analogų, pagrindžiamos projektų valdymo sistemos kūrimui naudojamos technologijos.

Projektinėje dalyje akcentuojami: priimti esminiai architektūriniai sprendimai kuriant sistemą, detalizuojami pagrindiniai sistemos komponentai ir jų tarpusavio ryšiai, atskleidžiama sąsaja tarp projektų valdymo metodikos ir realizuotos projektų valdymo sistemos, akcentuojamos esminės sukurtos sistemos savybės ir jų panaudojimo galimybės projektų valdymo procesų eigoje.

Prototipo kūrimo dalyje pasirinkti ir praktiškai realizuoti užduočių planavimui ir tvarkymui skirti komponentai. Taipogi įgyvendintas „gantt“ diagramos tipo projekto valdymo komponentas, išplėstų galimybių užduočių tvarkymo kalendoriaus tipo įrankis.

Eksperimentinė dalis skirta praktiškai įvertinti, pasirinktų komponentų ir metodų tinkamumą realioje projektų valdymo situacijoje. Įvertinamos darbų trukmės, turimi resursai, patikrintas vizualių komponentų teikiamas naudingumas.

2. PROJEKTŲ VALDYMO SISTEMŲ ANALIZĖ

2.1. Problema

Kokybiškas projekto veiklos organizavimas. Kad projektas taptų sėkmingas būtina užtikrinti šias sąlygas:

- Projekto vykdymo terminų laikymasis;
- Projektui skirto biudžeto neviršijimas;
- Realizuotas specifikacijoje nurodytas funkcionalumas.

Dažniausiai pasitaikanti ir labiausiai pabrėžiama projekto rizika yra netinkamas laiko įvertinimas ir resursų paskirstymas. Pirmajam punktui užtikrinti yra būtinas tinkamas projekto eigos plano sudarymas. Tai pažymi projektą sudarančių užduočių tinkamo paskirstymo svarbumą. Užduočių paskirstymą ir tvarkymą galime realizuoti grafiniais komponentais, kurie aiškiai ir paprastai atvaizduoja užduočiai priskirtus resursus ir vykdymo terminus.

Užtikrinus pirmo punkto įgyvendinimą, sumažėja projekto biudžeto viršijimo galimybė. Šiuo atveju pažymėtina tai, kad vizualiniai komponentai atlieka ir projekto rizikų mažinimo funkciją.

2.2. Projektų valdymo prototipų apžvalga

- **Project-open.** Tai tarptautinis atvirojo kodo projektas, kurio būstinė yra Barselonoje, Ispanijoje, o projektui vadovaujama iš Vokietijos. Projekto komandos pagrindas: Frank Bergmann, Klaus Hofeditz, Christof Damian. Tai ilgus metus dirbantys ir didelę projektų veiklos valdymo patirtį sukaupę žmonės. Project-open sistema skirta teikti verslo sprendimus mažoms ir vidutinio dydžio įmonėms. Project-open bendradarbiauja su kitais atvirojo kodo pasekėjais, tokiais kaip: OpenACS.org, AOLServer.org ir Linux. Sistema yra atvirojo kodo, o tai reiškia, kad galima laisvai naudotis programinės įrangos kodu. Kūrėjų svetainėje [1] taip pat pateikiama nuoroda išbandyti sistemą. Tai labai didelis privalumas, kadangi galima išbandyti sistemą jos neįsidiegiant. Taip pat pateikiami reikalavimai sistemos veikimui. Internetu prieinama ir sistemos dokumentacija.

- **E-groupware.** Sistema [3] sudaryta iš atskirų modulių – programų. Pažymima tai, kad sistema gali veikti daugelyje platformų ir naudotis ja galima daugeliu populiarių įrankių, tokiais kaip Kontakt, Evolution, Outlook. Taip pat svarbus pranašumas yra tai, kad verčiama net į 25 pasaulio kalbas, tarp jų įskaitant ir lietuvių kalbą. Sistemos svetainėje pateikiama demonstracinė versija, taigi sistemą galime išbandyti neįsideddami jos. Sistema naudotis neturėtų būti sudėtinga, kadangi svetainėje pateikiama visa dokumentacija, taip pat veikia forumas, bendruomenė. Sistema platinama pagal [GNU General Public License \(GPL\)](#) licenciją, tai reiškia kad naudotis ja galime laisvai ir nemokamai, taip pat galime ją ir tobulinti pagal licencijoje numatytus punktus.
- **Dotproject.** Šios sistemos [4] kūrėjai remiasi idėja, kad vartotojas supranta kas yra projektų valdymas, todėl pateikia tik paprastą projekto valdymo šabloną ir įrankius jo sėkmingam įgyvendinimui tvarkyti. Ši sistema, projekto įgyvendinimui stebėti ir valdyti, siūlo su juo susijusių užduočių vykdymo sekimo įrankį/tvarkaraštį. Kadangi taip bus galima stebėti visų susijusių funkcijų, tokių kaip planavimas, sutarties pasirašymas, rizikos valdymas, išlaidos priklausomybę. Sistemos pagrindą sudaro užduočių tvarkymo, laiko žymėjimo, bendradarbiavimo ir informacijos mainų įrankiai. Sistema palaiko ir keletą priedų, papildomų įrankių, kurias vartotojai taip pat gali įsdiegti. Sistema palaiko daugiakalbiškumą, tačiau nėra lietuviškos versijos. Pateikti sistemos veikimui reikalingi resursai bei aprašyta dokumentacijoje naudojama technologija. Pateikta pilna sistemos dokumentacija ir naudojimosi instrukcija. Sistema platinama, redaguojama ir gali būti naudojama nemokamai. Svetainėje pateikiama nuoroda išbandyti demonstracinę versiją, tačiau šiuo metu atliekami demonstracinės versijos puslapio tvarkymo darbai. Ši sistema pateikiama kaip nemokama alternatyva Microsoft kompanijos produktui – Microsoft office Project.
- **VisionProject.** Vision project [5] remiasi lengvumo ir „agile“ idėjomis valdyti projektus. Naudoja informaciją atvaizduotą grafikais projekto eigai stebėti. Pateikia įprastas pagalbos teikimo galimybes, tokias kaip problemos sprendimas elektroninio pašto, telefono, forumo pagalba. Sistema naudoja labai patrauklią vartotojo sąsają.

Pastebėtina tai, kad gausu ir funkcionalumo. Taip yra dėl to, kad sistema naudotis galima tik žiniatinklyje ir ši paslauga yra mokama. Tačiau sistemą išbandyti galima ir nemokamai. Užpildžius nedidelę formą elektroniniu paštu, gaunami prisijungimo duomenys ir išbandyti sistemą galima net 20 dienų. Ši sistema kaip ir pateikti lietuviškų sistemų variantai nėra priimtini, kadangi tai komerciniai produktai, nors mano nuomone, iš Vision Project būtų galima pasimokyti kokia turėtų būti vartotojo sąsaja.

- **ProjectPier.** Ši sistema [6], iš visų aptartų, valdoma turbūt paprasčiausiai. Vartotojo sąsaja nors ir primityvi, tačiau pilnai suteikianti funkcionalumą. Be to, tai turėtų būti lengvai plėtojama sistema, kadangi naudoja populiarias bibliotekas. Tačiau, norint tobulinti šia sistemą tektų rimtai padirbėti prie vartotojo sąsajos kokybės, nes pilnai funkcinės galimybes atliekanti sistema gali būti nepatraukli vartotojui ir jis ja nesinaudos. Sistemos svetainėje pateikiama dokumentacija, vartotojo vadovas, forumas. Nėra galimybės išbandyti demonstracinę versiją. Naudotis galima pagal „Creative Commons License, Attribution-ShareAlike 2.0“.
- **Achievo.** Atvirojo kodo sistema [7], parašyta panaudojant PHP programavimo kalbą ir SQL duomenų bazes. Skelbianti save verslui skirtu projektų valdymo įrankiu, kuris lengvai ir efektyviai pateikia informaciją ir jos valdymą. PĮ puslapyje rašoma, kad numatyta šio įrankio plečiamumo galimybė ir tai padaryti nėra sunku. Pateikiama nemokama pagalba forume, bei galimybė išbandyti demonstracinę versiją. Taip pat turi klaidų registravimo sistemą.
- **Basecampgh.** Labai aiškios vartotojo sąsajos įrankis [8], tačiau mano nuomone labiau skirtas bendradarbiavimui. Projektų valdymas čia vykdomas tik tokiu būdu: sukuriame objektą „projektas“, tada galime priskirti užduotis, rašyti komentarus, žymėti numatomas užduotis. Bet tokie komponentai kaip kalendorius, mano nuomone, yra įgyvendinti netinkamai, pernelyg supaprastinti. Galėtų būti pasirinkimas vartotojui, norinčiam matyti sudėtingesnio tipo kalendorių ir tvarkyti jame užduotis. Taip pat neradau ir projektų palyginimo ar kokių kitų diagramų, atvaizduojančių projekto išsidėstymą laike.

Prototipo gynimo metu pateiktoje magistrinio darbo ataskaitoje pateikiamas platesnis projektų valdymo PĮ palyginimas.

2.3. Pagrindinis projektų valdymo sistemų apibendrintas funkcionalumas

- Apžvalga (angl. dashboard, angl. overview)
- Projektai (portfolio, planas)
- Užduotys (angl. tasks, angl. milestones, angl. to-dos, angl. tickets, angl. issues, angl. bugs)
- Laikas (kalendorius, laiko diagrama)
- Bendradarbiavimas (angl. collaboration, angl. chat, forumas, komentarai)
- Resursai (dokumentai, failai, žmonės, inventorių)
- Klientai, kontaktai (angl. companies, angl. clients, angl. contacts, angl. addressbook)
- Žinynas (angl. wiki per project, angl. knowledgebase)
- Sąskaitos
- Ataskaitos (angl. reports, angl. gantt charts)
- Paieška
- Administravimas (šablonai, vartotojai)

2.4. Projektų valdymo PĮ tikslai

- Projekto proceso automatizavimas, optimizavimas.
- Projekte dalyvaujančių asmenų apjungimas, informavimas, koordinavimas.
- Resursų, darbų, eigos, žinių, informacijos, failų valdymas, valdymo optimizavimas automatizuojant įvairius projekto proceso etapus.
- Rezultatų pateikimas grafine vizualine forma (grafikais).
- Priminimai apie etapų terminus.
- Ataskaitų generavimas.

- Galimybės dirbti nuotoliniu būdu bendraujant virtualioje aplinkoje užtikrinimas.

2.5. Situacijos Lietuvoje įvertinimas

Lietuvoje negausu tokio pobūdžio sistemų, ypač nemokamų. Paminėti galima tokias sistemas kaip Dokas, Linditex, Dekonos projektų valdymo sistemas, tačiau visi šie produktai yra komerciniai.

2.6. Projektų sėkmės faktoriai

Nagrinėjant projektų valdymo metodiką, galime išskirti šiuos sėkmės faktorius:

1. Projekto palaikymas – ši sąvoka apima projekto užsakovą ir jo skiriamas lėšas projekto vystymui.
2. Būsimo vartotojo įtraukimą į projekto eigą – iš jo galima sužinoti daug probleminių sričių, pastiklinamosios informacijos, kaip derėtų spręsti iškilusias problemas.
3. Patyręs projektų vadovas.
4. Pagrindinių projekto tikslų nustatymas.
5. Projekto apimties nustatymas.
6. Projekto skaidymas į smulkesnes užduotis.
7. Aiškiai apibrėžtas projekto valdymo procesas.
8. Automatizuotų įrankių naudojimas.
9. Nuolatinis rizikos vertinimas.

1, 2, 6, 7, 8, 9 punktus tiesiogiai arba netiesiogiai gali įtakoti projektų valdymo programinės įrangos pasirinkimas.

2.7. Projektų valdymo sistemų klasifikacija

Projektų valdymo (dauguma PĮ) gali būti skirstomos pagal veikimo pobūdį ir terpę į šias kategorijas:

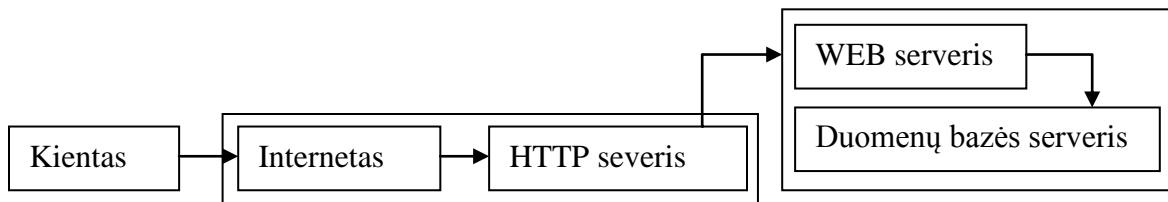
- Žiniatinklio – vartotojo naršyklės pagalba veikianti programinė įranga.
- Darbastalio – vartotojo kompiuteryje veikianti programinė įranga.
- Asmeninė – vienam vartotojui skirta nesudėtinga programinė įranga.
- Integruota – šios sistemos apima CRM, kalendorius, užduočių tvarkykles.

- Bendradarbiavimo – teikia galimybes dirbti bendradarbiaujant, pasiskirstant atsakomybes tarpusavyje. Dažniausiai žiniatinklyje veikianti programinė įranga.
- Mobiliosios – skirtos mobiliems įrenginiams su minimaliu funkcionalumu arba daliniu pilnos sistemos funkcionalumu. Pritaikytos pagal operacinę įrenginio sistemą.

2.8. Pasirinktas programinės įrangos kūrimo procesas

Apžvelgus programinės įrangos kūrimo metodikas ir dabartines technologijų galimybes, buvo nuspręsta, kad bus kuriama žiniatinklio tipo PĮ. Tokio tipo aplikacijos dažniausiai būna realizuojamos trijų lygių architektūra:

- Interneto naršyklė;
- WEB serveris;
- Duomenų bazės serveris;



Paveikslukas 1. Programų realizavimo architektūra

2.8.1. Žiniatinklio tipo programinės įrangos privalumai:

- Pakankamai lengvai realizuojama ir diegiama;
- Nesudėtingas programinės įrangos atnaujinimas;
- Santykinai nebrangus palaikymas;
- Nereikalinga papildoma programinė įranga vartotojo kompiuteryje (diegiama);
- Užtikrinamas duomenų centralizuotumas ir saugumas. Priklausomai nuo serverių saugumo lygio.

2.8.2. Žiniatinklio tipo programinės įrangos trūkumai:

- Naudojama HTML tekstų internetui interpretavimo kalba ne visose naršyklėse atvaizduojama vienodai, ypatingai DOM elementų panaudojimas. O HTML5 šiuo

metu palaikomas tik populiariausių naršyklių, tokių kaip Mozilla Firefox, Google Chrome, Safari, Opera, tačiau labai sunkiai suderinamas su naujausia Microsoft korporacijos naršykle Internet explorer 9.0 ir visiškai nesuderinama su senesnėmis šios didžiosios programinės įrangos gamintojos naršyklėmis.

- Duomenų validumo užtikrinimas. Programavimo kalba, kaip Php, neturi griežtai priskiriamų kintamųjų tipų, todėl visada reikia papildomai tikrinti gaunamų duomenų tinkamumą.
- Atitinkamas sistemos veikimo greitis priklausomai nuo kliento interneto spartos.

Taip pat numatoma esant sėkmingam PĮ realizavimui ir naudojimui, įdiegimas mobiliesiems įrenginiams, kurie naudoja iOS ar android operacinę, aplikacijų būdų ar tiesiog mobiliojo įrenginio naršykle. Kadangi šių įrenginių vartotojo sąsajos pakankamai gerai išvystytos ir turi plačias panaudojimo galimybes, nesudėtinga jose atvaizduoti kalendorius, užduočių valdymo komponentus, projektų atvaizdavimo komponentus.

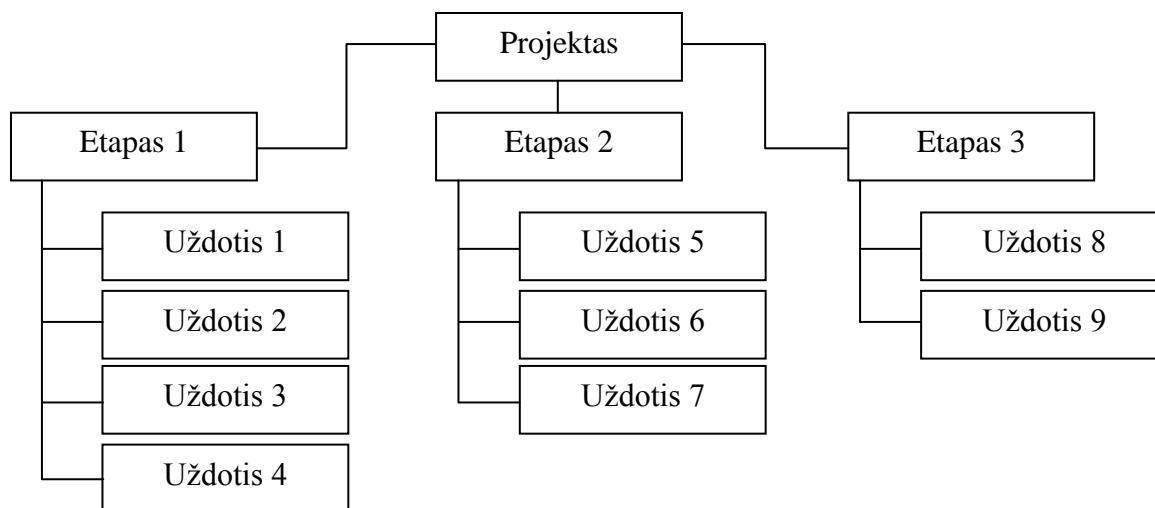
Atsižvelgiant į dabartinę interneto plėtros būseną ir „Cloud“ aplikacijų spartų populiarėjimą, nėra tikslinga kurti tokio tipo ir sudėtingumo PĮ, skirtą darbataliui.

2.9. Pasirinkta realizavimo architektūra

Vartotojų kompiuteriuose veikianti programinė įranga bus atsakinga už komponentų tinkamą atvaizdavimą. Labai priklausys komponentų kokybė ir jų atvaizdavimas nuo vartotojo interneto naršyklės javascript, css3, html5 palaikymo procento. Duomenims saugoti pasirinktas Mysql duomenų bazės serveris, o visus atliekamus veiksmus duomenų bazės lygyje pasinaudojant Adodb php programavimo kalbos biblioteka. Ši biblioteka turi integruotą užklausų kešavimą, aktyvių sesijų valdymą, skirtingų duomenų bazių prisijungimo ir darbo su jomis galimybę. Programinės įrangos loginę dalį realizuos php kalba aprašyti aplikacijos logikos moduliai. Web server tradiciškai šiai programavimo kalbai realizuoti renkamasi Apache web serveris.

2.10. Projekto koncepcinė schema

Tinkamai organizuoti projektą, sudėtingesnius jo etapus, būtina suskaidyti į mažesnes užduotis tam, kad visi projekto žingsniai būtų tinkamai įvertinti visais atžvilgiais.



Paveikslukas 2 Koncepcinė projekto schema

2.11. Išvados

Apibendrinti atliktos analizės rezultatai:

- Atsakyta į klausimą kodėl pasirinktas tokio tipo projektas;
- Išanalizuoti rinkoje esantys programinės įrangos analogai;
- Pasirinktas programinės įrangos kūrimo tipas.

3. PROJEKTŲ VALDYMO SISTEMOS PROTOTIPAS

Šiame skyriuje aprašyti esminiai sukurtos sistemos architektūriniai ir vizualių komponentų panaudojimo sprendimai.

3.1. Sistemos tikslai

Projekto tikslas yra sukurti projekto kontroliavimo programinę įrangą. Tai bus posistemių rinkinys, organizuojantis projekto procesą. Užduočių vadovams ir jų atlikėjams bus sudarytos didesnės galimybės tvarkyti resursus, susijusius su projektais.

Naudodamasis sistema darbų vadovas galės atlikti rutininius darbus daug greičiau bei paprasčiau ir galės daugiau laiko skirti svarbesniems dalykams.

Sistema leis:

- Tvarkyti projektus.
- Tvarkyti užduotis.
- Tvarkyti resursus.
- Saugoti versijas.
- Sudaryti projekto planą.
- Generuoti ataskaitas.
- Tvarkyti kalendorius.
- Ieškoti sistemoje.
- Kurti darbuotojų profilius.
- Atvaizduoti diagramos rezultatus.
- Bendrauti (komentarai, forumas, žinynas, pokalbiai).
- Tvarkyti kontaktus.

3.2. Projektų valdymo PĮ esybių analizė

Projektuojant projektų valdymo sistemą apibrėžtos šios pagrindinės esybės:

- Projektas
- Užduotis
- Vartotojas
- Terminas
- Vartotojo kategorija (tipas)
- Teisė

Siekiant padaryti kuriamą programinę įrangą universalesne, buvo papildomi projekto esybės atributai. Projekto esybė papildyta tokiais atributais:

- Pavadinimas;
- Aprašymas;
- Kas atsakingas;
- Pradžios data;
- Pabaigos data;
- Įvykdymo požymis;
- Įvykdymo procentas;
- Aktyvumo požymis.

Užduoties esybė papildyta tokiais atributais:

- Aprašymas;
- Kam skirta atlikti;
- Projektas kuriam priklauso;
- Laiko periodas (pradžios data, pabaigos data, pasikartojamumas, pilna diena);
- Vietovė;
- Numatyta trukmė;
- Įvykdymo požymis;
- Paveldimumas;
- Užduotims priskiriant du spalvos kodus. Vienas skirtas vartotojo konfigūravimui, kokios spalvos užduočių blokus matys. O antras skirtas aukštesnėmis teisėmis esančiam vartotojui atskirti vieno nuo kito vartotojo užduočių blokus, pasirenkant skirtingus spalvų kodus. Spalvų kodai pasirenkami visiškai laisvai.

Vartotojo esybė papildyta tokiais atributais:

- Nario vardas;
- Vaidmuo;
- Vardas;
- Pavardė;
- Pseudonimas;
- El.paštas;
- Pranešimų programų pseudonimai;
- Apie.

Vartotojo kategorijos esybė papildyta tokiais atributais:

- Identifikuojantis laukas. Skirtas programinės įrangos kokybiškesniam realizavimui. Php kalbos funkcijose, objektų metoduose manipuluoti kitais didelės svarbos objektais manipuluoti ne pagal automatiškai priskirtą objekto identifikatorių, o nustatytu iš anksto.
- Pavadinimas. Vartotojui suprantamas kategorijos pavadinimas.

Tai pagrindiniai atributai kiekvienai esybei. Šiuos visus esybių atributus nuspręsta dar papildyti bendriniais atributais:

- Visų tipų esybėms priskirti atributus sukūrimo data, modifikavimo data, skirtą duomenų modifikavimo, pasikeitimų sekimui.
- Visų tipų esybes papildyti atributais kas sukūrė. Kad visada būtų galima atsekti kas ir ką keitė sistemoje. Kitas šio atributo panaudojimo atvejis numatytas kuriant veiksmų istorijos sekimą, registruoti kas, kokius duomenų objektus sukūrė naudodamasis programine įranga.

- Visų tipų esybės papildytos atributų rikiavimo pasirinkimais ir rikiavimo tipais. Tai supaprastins duomenų sąrašų rikiavimus vartotojo posistemėje.

Išskyrus esybes ir papildžius jas naujas atributais, galima konkrečiau suformuluoti reikalavimus kuriamai PĮ. Tolimesniame skyriuje detalizuosime PĮ keliamus reikalavimus atsižvelgiant į išskirtas esybes ir jų atributus. Bus detalizuota pasirinkta sistemos architektūra, sukurto prototipo architektūriniai sprendimai.

3.3. Bendri apribojimai

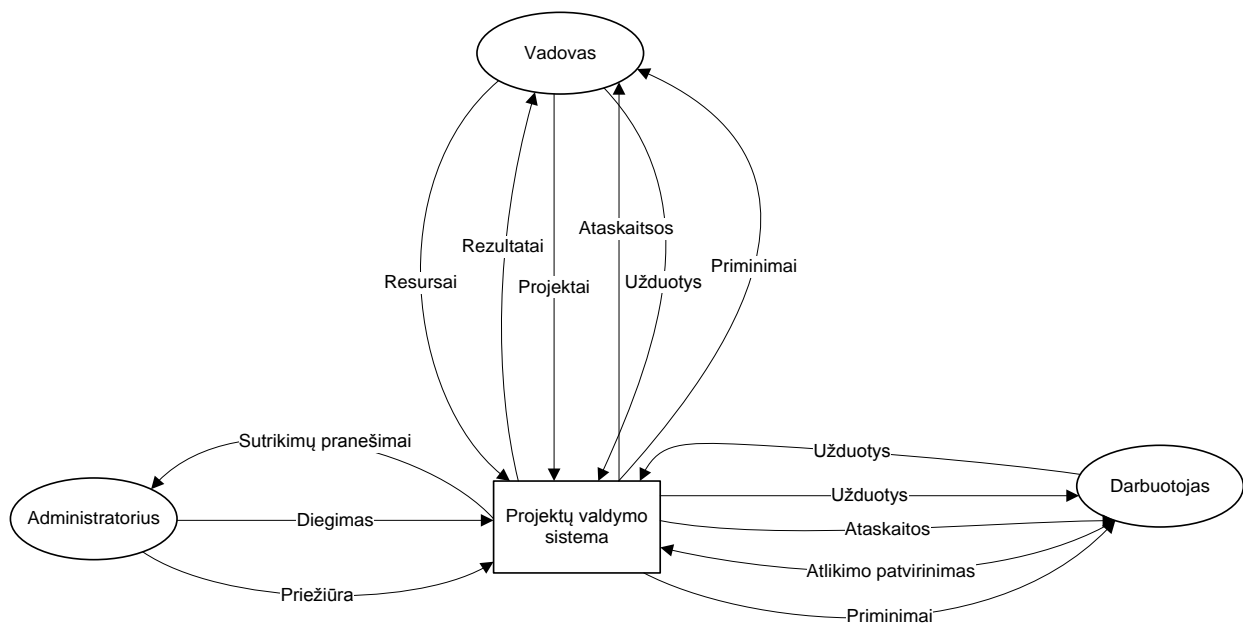
Pagrindiniai sistemai taikomi apribojimai:

- Sistema turėtų būti kuriama naudojant evoliucinį sistemos projektavimo modelį, t.y. pradiniam žingsnyje yra sukuriama sistemos prototipas, o vėliau jis vystomas ir tobulinamas iki tol, kol gaunama galutinė, išbaigta sistema.
- Vienas pagrindinių sistemos reikalavimų - garantuoti apsaugą ir autorizavimą: vartotojai, neturintys tam tikrų jiems suteiktų teisių, negali atlikti jiems uždraustų veiksmų arba padaryti nepataisomos žalos sistemai. Pvz., registruotas vartotojas negali peržiūrėti, keisti ar pasisavinti kitų vartotojų informacijos, valdyti kai kurias kitas tik administratoriams ar vadovams galiojančias sritis, kur jam nėra suteiktos valdymo teisės.
- Sistema privalo užtikrinti vartotojų prisijungimo slaptažodžių slaptumą ir saugumą. Tiek registruotas vartotojas, tiek vadovas, tiek administratorius negali matyti nei savo, nei kitų vartotojų slaptažodžių, administratorius gali tik pakeisti savo, kito administratoriaus bei registruoto vartotojo slaptažodį, o registruotas vartotojas – tik savo slaptažodį. Slaptažodžiai privalo būti saugomi duomenų bazėje, užkoduoti pasirinktu kodavimo algoritmu, siekiant užtikrinti maksimalią apsaugą.
- Sistema privalo būti suderinama su populiariausiomis interneto naršyklėmis, tokiomis kaip Internet Explorer, Mozilla, Opera, Netscape. Tai pasiekti galima nuodugniai testuojant sukurtą sistemą ir taikant atitinkamas priemones nesuderinamumų pašalinimui.
- Sistema turi būti pasiekiami iš bet kurios vietos, kur yra interneto ryšys. Sistemoje turėtų būti naudojamas “plono kliento” modelis, tam, kad klientui nereikėtų papildomos programinės įrangos bei didelių kompiuterinės įrangos resursų – pagrindinė sistemos logika atliekama serverio pusėje.

- Sistema turi būti sukurta naudojant šias programines priemones: PHP [10], MYSQL [11], JavaScript [12, 13, 14], HTML, XHTML, HTML/5 [20], CSS2, CSS3, AJAX [16], JQUERY [17, 18], JQUERY UI [19].
- Sistemos vykdymui ir palaikymui reikalingas Apache web serveris.
- Duomenų saugojimui turi būti sukurtas atskiras MYSQL vartotojas, kuriam turi būti sukurtas aukšto sudėtingumo lygio slaptažodis bei atskira MYSQL duomenų bazė. Sistemoje turi būti apsauga nuo atsitiktinio klaidingo duomenų įvedimo.

3.4. Funkciniai reikalavimai

3.4.1. Veiklos kontekstas



Paveikslukas 3 Veiklos kontekstas

3.4.2. Veiklos padalinimas

Eil. Nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1.	Naujas projektas	Projekto duomenys (in)
2.	Nauja užduotis	Užduoties duomenys (in)

3.	Naujas darbuotojas	Darbuotojas duomenys (in)
4.	Naujas resursas	Resurso duomenys (in)
5.	Naujas kontaktas	Kontakto duomenys(in)
6.	Darbuotojas įvykdė darbą	Darbo vertinimas (in)
7.	Pateikti projektų rezultatus	Projektų rezultatai (out)
8.	Pateikti užduočių rezultatus	Užduočių lankomumas (out)
9.	Pateikti darbų ataskaitas	Darbų (projektų atliktų užduočių) ataskaitos duomenys (out)
10.	Kalendoriaus peržiūra	Užduočių grafikas (out), Projektų grafikas (out)
11.	Rezultatų peržiūra	Projektų apibendrinimų rezultatai (out)

Lentelė 1 Veiklos padalinimas

3.5. Papildomi reikalavimai

Sukurtas arba panaudotas kalendoriaus komponentas turi būti kuo artimesnis google kalendoriui, t.y. turi būti panaši vartotojo sąsaja, bei pagrindinis funkcionalumas. Magistrinio darbo reikalavimai ir paaiškinimai pateikiami [9].

3.6. Projektavimas

Sistemos projektavimui turėtų būti naudojama UML projektavimo kalba. Bei vienas iš įrankių „magic draw“.

3.7. Naudojamos technologijos

Kadangi sistema turi būti kuriama atvirojo kodo sistemų bazėje [26], tai ir jos kūrime panaudotos technologijos turi būti taip pat nemokamos. Kadangi tai sistema skirta

naudojimuisi žiniatinklyje, tai ir bus naudojamos populiariausios nemokamos bibliotekos bei įrankiai.

3.8. Duomenų bazė

Labiausiai žiniatinklyje paplitęs duomenų bazės valdymo įrankis MySQL. „MySQL – viena iš reliacinių duomenų bazių valdymo sistemų (liet. santrumpa RDBVS [21], angl. – RDBMS), palaikanti daugelį naudotojų, dirbanti SQL kalbos pagrindu. MySQL yra atviro kodo programinė įranga (GPL [22] ir kitos licencijos). Nors prieigai prie MySQL duomenų bazių dažniausiai pasirenkama PHP kalba, ją taip pat galima pasiekti įvairiomis kitomis programinėmis priemonėmis – C, C++, C#, Java, Perl, Python ir kitomis. Kiekvienai šių kalbų sukurtos specialios bibliotekos (API [23]). Taip pat MySQL duomenų bazėms yra sukurta ODBC sąsaja MyODBC, leidžianti duomenis pasiekti bet kuria kalba, neturinčia specialios bibliotekos, tačiau palaikančia ODBC [24] komunikavimo mechanizmą. Todėl numatant PĮ plečiamumo galimybę nebus sudėtinga prisijungti prie duomenų baze naudojant kitokias operacines sistemas.

3.9. Programavimo kalbos

Sistemos turėtų būti programuojamos php programavimo kalba. PHP – plačiai paplitusi dinaminė interpretuojama programavimo kalba (en: Hypertext Preprocessor), sukurta 1997 m. ir specialiai pritaikyta interneto svetainių kūrimui. Tačiau ji nėra skirta tikai svetainių kūrimui, ji pakankamai efektyviai realizuoja ir didesnes sistemas. PHP sintaksė panaši į daugelį struktūrinių kalbų, ypač į C bei Perl. PHP kalba yra atviro kodo ir tai yra viena priežasčių dėl ko kalba, nors ir yra nesudėtinga, bet gana lanksti – veikia daugumoje operacinių sistemų, palaiko nemažai reliacinių duomenų bazių bei veikia su dauguma interneto serverių – CGI, FastCGI, ISAPI ir kitais protokolais. Nors PHP yra dažniausiai naudojama interneto puslapių kūrimui, bet yra labai galingas įrankis atlikti kitas funkcijas komandinėje eilutėje. Php skirta atlikti logikos veiksmus, o vartotojo sąsajos programavimui galėtų būti panaudotos javascript programavimo kalbos ir atitinkamai HTML, XHTML, DHTML, HTML5 DOM technologijos.

JavaScript – objektiškai orientuota skriptų programavimo kalba, besiremianti prototipų principu. Dažniausiai kalba naudojama internetinių puslapių interaktyvumo realizacijai, bet

taip pat naudojama ir kaip galimybė skriptais manipuluoti tam tikromis programomis. Kalba sukurta Brendano Eicho Netscape kompanijoje ir pavadinta Mocha, vėliau pervadinta į LiveScript, ir galiausiai tapo JavaScript. Vienas iš argumentų pervadinant kalbą buvo sintaksinis panašumas su Java kalba. Paskutinė JavaScript versija – 1.8.

Taip pat ir HTML, XHTML, DHTML, HTML5 žymių kalba, CSS2, CSS3(kaskadinės stilių lentelės) vartotojo sąsajos formatavimui.

Ajax – technologija taip pat turėtų būti naudojama, kadangi suteikia sistemai interaktyvumo įspūdį. “Ajax (sutrumpinta sąvoka įvardyti Asinchroninis JavaScript ir XML) yra tarpusavyje susijusių interneto svetainių kūrimo metodų kliento pusėje technologija, kad būtų sukurta interaktyvių interneto programų. Panaudojant ajax technologiją, puslapiuose duomenys gali būti atnaujinti neatnaujinant jų pačių. Duomenys paprastai atnaujinami naudojant XMLHttpRequest objektą.

3.10. **Web Serveris**

Kadangi sistema turi būti pasiekiamą internetu, sistemos veikimui reikalingas serveris. Tai įgyvendinti būtų galima į pagalbą pasitelkus patį populiariausią nemokamą http tinklo serverį - apache. Apache HTTP serverį kuria atvira kūrėjų bendrija, kurią palaiko Apache Software Foundation fondas. Apache palaiko įvairias operacines sistemas: Microsoft Windows, Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X. Apache HTTP serveris gali būti naudojamas pagal Apache licenciją ir tai yra laisvoji atviro kodo programinė įranga. Serverio operacinė sistema turėtų būti parinkta iš linux operacinės sistemos distribucijų. Kadangi apache serveris bei php programavimo kalba turi daugiau galimybių šios šeimos operacinėse sistemose.

3.11. **Papildomi įrankiai**

Sistemos kūrimui numatomi papildomai reikalingi įrankiai.

- Notepad++, JEdit – kodo redagavimui.
- Windows XP, 7 – operacinė sistema.
- Putty – prisijungimui prie sistemos serverio, serverio nustatymų keitimui.

- Filezilla – failų talpinimui į web server bei jų tvarkymui.
- Adobe fireworks – dizaino pakeitimams atlikti.

3.12. **Diegimo aplinka**

Sistema diegiama kliento turimame kompiuteryje su operacine sistema kuriame yra galimybė įdiegti „apache“ web serverį ir „mysql“ duomenų bazę, bei pasiekiamumą iš tinklo ir interneto prie minėto kompiuterio.

3.13. **Bendradarbiaujančios sistemos**

Kuriamos sistemos dalį komponentų bus galima susieti su kitomis sistemomis, kadangi dalis komponentų duomenis priims xml, html, json formatu.

3.14. **Numatoma darbo vietos aplinka**

Darbo vieta bus mobili. Tačiau visada bus galima pasiekti visus duomenis iš pagrindinio serverio, bet kuriuo metu.

3.15. **Svarbūs faktai ir prielaidos**

3.15.1. Svarbūs faktai

Kuriama sistema bus pateikta kaip atviro kodo programinė įranga visiems norintiems ja naudotis.

3.15.2. Prielaidos

Sistemos kūrimo metu galima panaudoti visą sistemos funkcionalumą įgyvendinačią nemokamą programinę įrangą, įgyvendinant panaudojimo atvejų modelį.

3.16. **Vartotojai**

3.16.1. Vartotojo charakteristikos

Sistemos vartotojai yra sistemos administratoriai, įvairių lygių projektų vadovai, darbuotojai (su projekto įgyvendinimų susiję žmonės), klientai (projekto užsakovai). Kadangi visi vartotojai bendrauja per grafines sąsajas ypatingų kompiuterinių žinių nereikia, nebent susipažinti su sistemos vartotojo vadovu.

3.16.2. Vartotojo problemos

Kuriama sistema leis efektyviau panaudoti turimas galimybes. Visų pirma vartotojai nori naudotis patogią ir aiškią vartotojo sąsają turinčią PĮ. Tuo tikslu bus parinkta draugiškesnė vartotojo sąsaja tiek administravimo posistemei, tiek vartotojo naudojimosi posistemei.

3.16.3. Vartotojo tikslai

- Suprantamas programos bendravimas, pranešimai vykstant tam tikriems įvykiams arba klaidoms.
- Aiški dokumentacija.
- Paprasta ir aiški vartotojo sąsaja.
- Paprastas projektų kūrimas, tvarkymas, eigos sekimas.
- Papildomos galimybės projekto proceso organizavime, tokios kaip užduočių, resursų valdymas.
- Sistemos patikimumas.
- Sistemos našumas.
- Duomenų atstatymo/eksportavimo/importavimo galimybė.

3.17. Projekto išeiga

Sistema turi būti kuo universalesnė ir pritaikoma įvairių tipų projektų valdymui.

3.18. Egzistuojantys sprendimai

3.18.1. Pagamintos sistemos, kurios gali būti panaudotos

Yra nekomercinių realizacijų, kurios įgyvendina tam tikras sistemos funkcijas. Jeigu esamų produktų komponentų sąsajos lengvai įgyvendina bendradarbiavimą su kitomis sistemomis, tai juos bus galima pakartotinai panaudoti.

3.18.2. Pagaminti komponentai, kurie gali būti panaudoti

- Htmlx scheduler – užduočių tvarkymo kalendorius.
- Htmlx grid – duomenų atvaizdavimo lentelės komponentas su galimybe redaguoti.
- mailer – priminimų, pranešimų siuntimas elektroniniu paštu.

- Google Chart Tools – rezultatų atvaizdavimui grafiškai. jQuery UI – vartotojo sąsajos kūrimui.

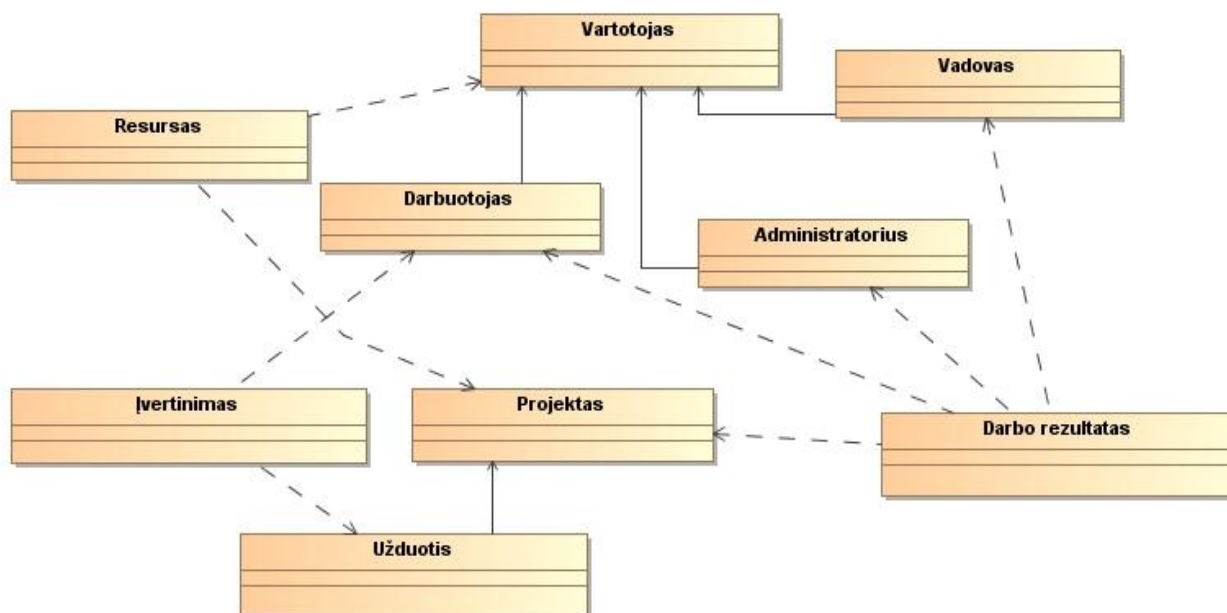
3.19. Panaudojimo atvejai



Paveikslukas 4 Panaudojimo atvejai

3.20. Reikalavimai duomenims

Pradinis duomenų modelis pateiktas sekančiame paveikslėlyje.



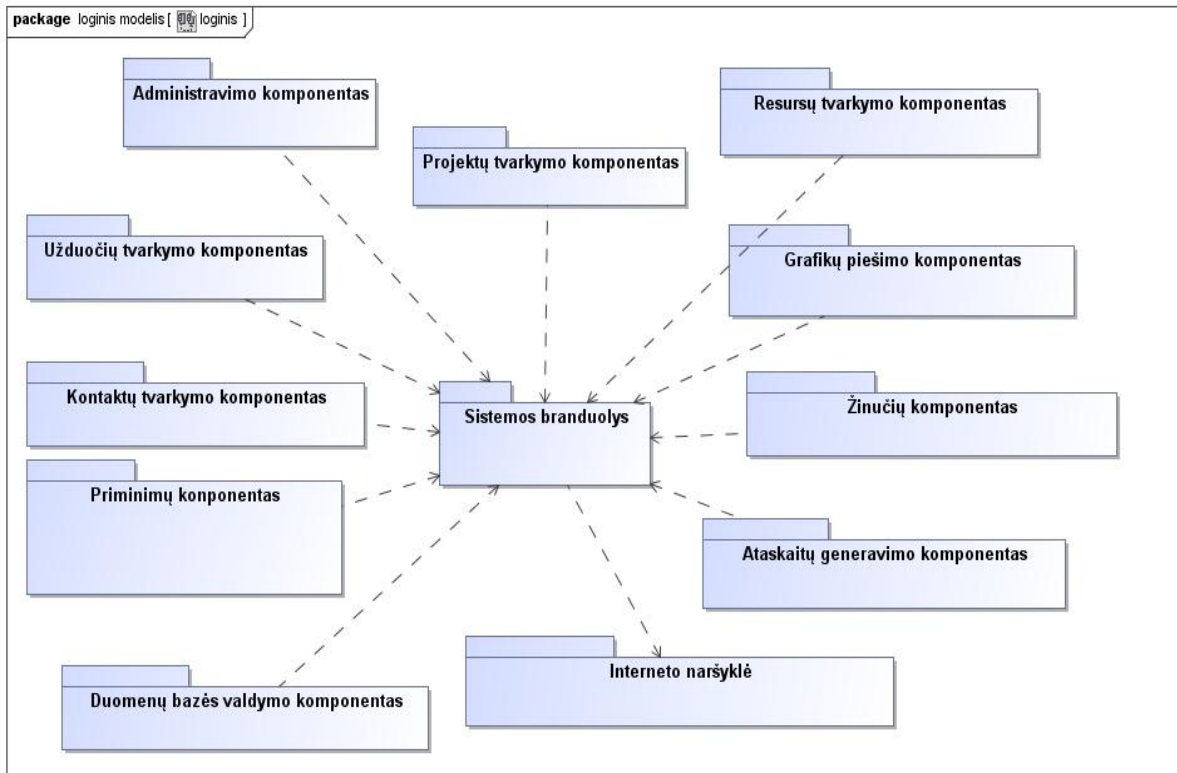
Paveikslukas 5 Duomenų modelis

3.21. Galimas pakartotinas panaudojimas

Bakalauro darbe buvo integruota projektų valdymo ir sąskaitų faktūrų tvarkymo sistema, kur galima tinkamus komponentus perkelti į naująją sistemą.

3.22. Loginis vaizdas

Sistemą sudaro dalys – komponentai. Komponentams apjungti php programavimo kalba sukurtas ‚branduolys‘ kuris valdys veiksmus sistemoje.



Paveikslukas 6 Loginis vaizdas

3.23. Komponentų detalizavimas

3.23.1. Sistemos branduolys

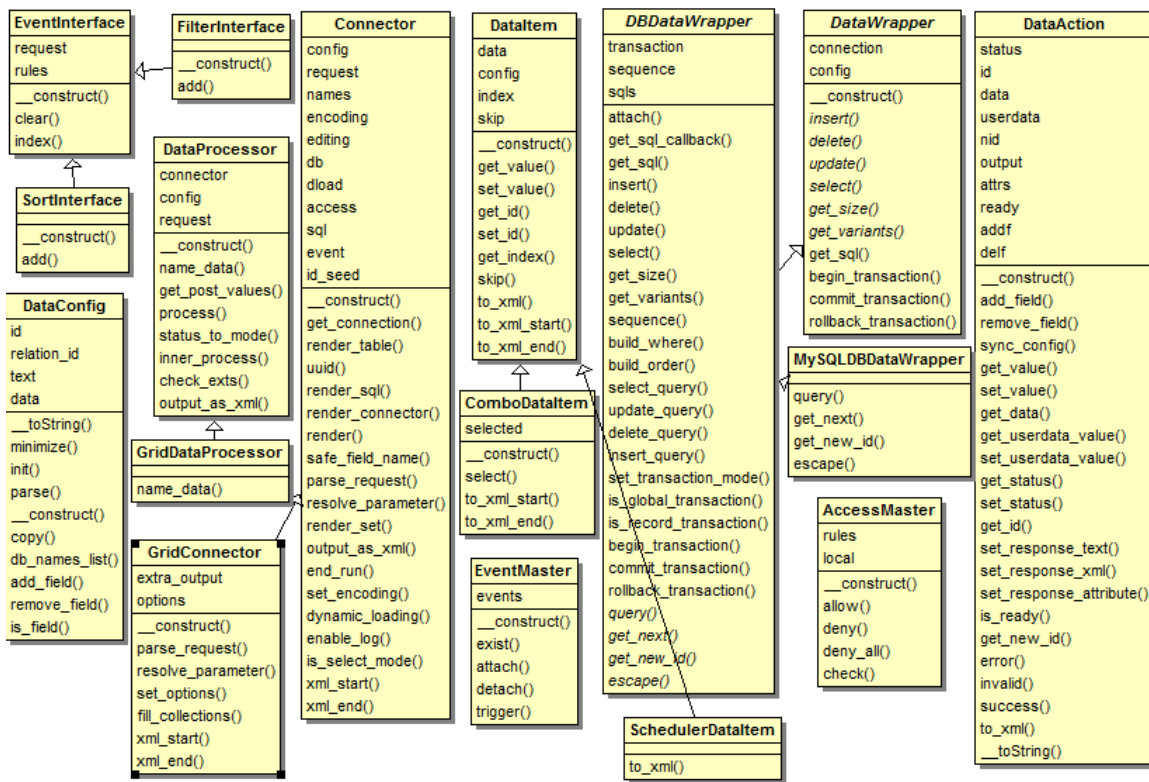
Šis komponentas realizuoja WEB sąsają, kuria vartotojas gali pasiekti per naršyklę. Šis komponentas bus realizuojamas pasinaudojant MVCH technika: Model – View – Controller - Hierarchical. Kiekviena šio komponento esybė (Session, User, Capabilities, Request, Template, Language, Run) turės modelį – patį objektą, kontrolerį – vartotojo galimi atlikti veiksmai su modeliu ir vaizdą – kaip kiekvienas objektas bus atvaizduojamas (HTML). Taip pat gali prireikti paslaugų sluoksnio (Service Layer), kuris realizuotų veiksmus su objektais, kurie vartotojui yra nematomi. Table modelis saugo prisijungimą prie duomenų bazės, abstractobject modelis atlieka mysql funkcijas pagal perduotą objekto id, session modelis keičia abstractobject modelio metodus su savo objekto id. Aktualiausias šios komponento modeliai yra adodb, table, abstarctobject, session, user, language, request, run, tablerow, tablerowlist, template, kadangi jie yra atsakingi už tai ką vartotojas daro su sistema.



Paveikslukas 7 Sistemos branduolys

3.23.2. Administravimo komponentas

Šis komponentas tikrina vartotojų teises. Jo pagalba kuriami, redaguojami, šalinami sistemos vartotojai, keičiamos jų teisės. Žemiau pateikiamas komponentą sudarančių klasių diagrama:



Paveikslukas 8 Kalendoriaus komponento klasių diagrama

3.23.3. Užduočių tvarkymo komponentas

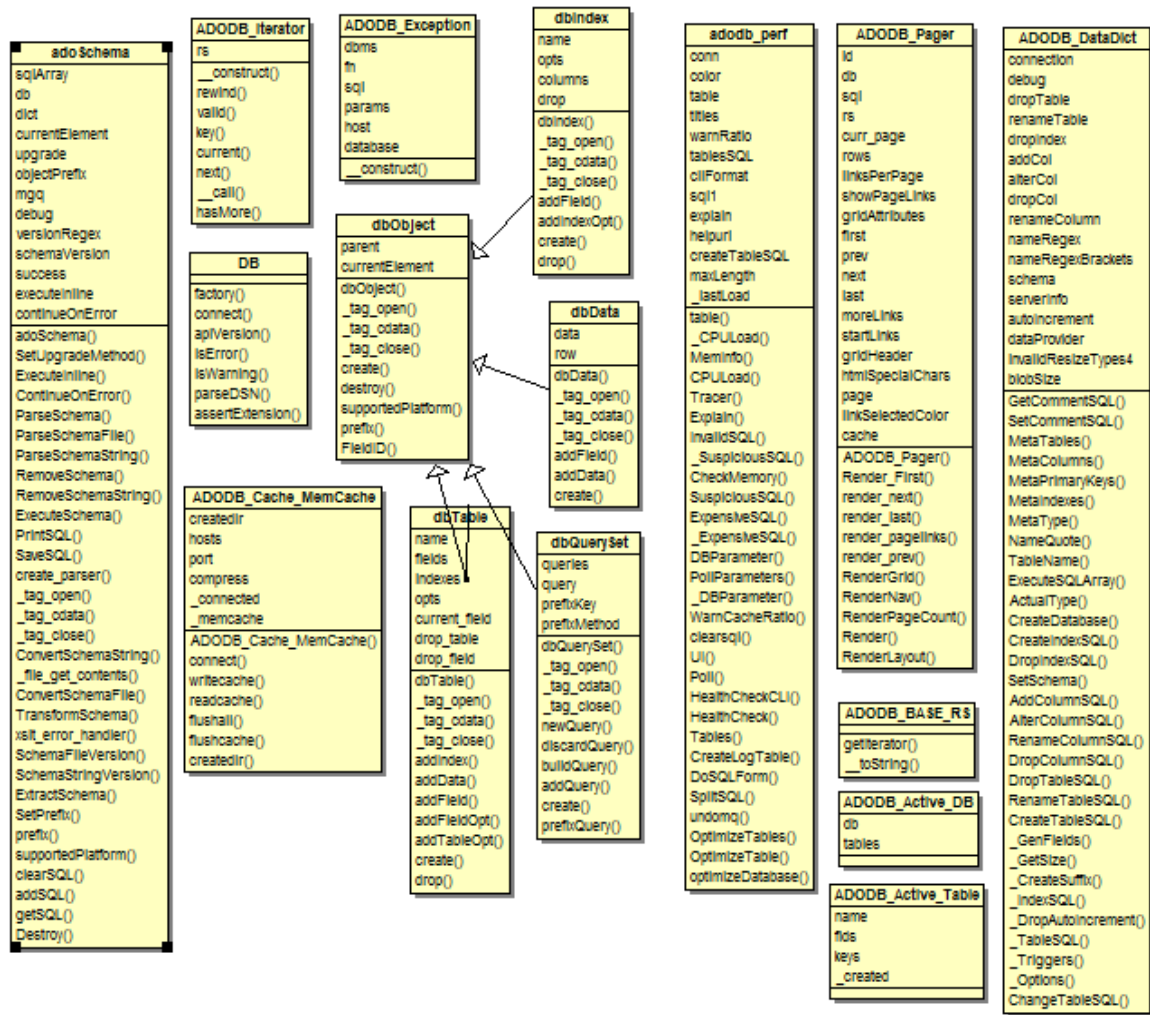
Vadovas arba darbuotojas turi galimybę tvarkyti užduotis ,susietas arba ne, su projektais. Sistema pateiks artimiausių dienos, savaitės, mėnesio, užduočių vaizdą. Suteiks galimybę kurti, keisti, šalinti užduotis ir susieti jas su projektu, darbuotoju. Šio komponento pagrindas „dhtmlx sheduler“. Užduočių ataskaitos peržiūros generavimui panaudotas „dhtmlx grid“ elementas.

3.23.4. Priminimų komponentas

Sistema turi galimybę siųsti automatinius priminimus. Šiam komponento įgyvendinimui panaudota „cron jobs“ technologija, kuri automatiškai įvykdys komponento metodus.

3.23.5. Duomenų bazės valdymo komponentas

Tai yra yra duomenų bazių abstrakcijų biblioteka adodb, kuri palaiko labai daug duomenų bazių valdymo sistemų, tokių kaip: MySQL, PostgreSQL, Interbase, Firebird, Informix, Oracle, MsSQL, Foxpro, Access, ADO, Sybase, FrontBase, DB2, SAP DB, SQLite, Netezza, LDAP, ODBC, ODBTP ir dar nemažai kitų. Sistemoje šį komponentą naudoja table modelis, atlieka tarpinį vaidmenį perduodant parametrus tarp sistemos objekto ir adodb objekto. Žemiau pateikiama komponentą sudarančių klasių diagrama:



Paveikslukas 9 Duomenų bazės valdymo komponento klasių diagrama

3.23.6. Projektų tvarkymo komponentas

Šis komponentas suteiks galimybę peržiūrėti grafiškai atvaizduotą projekto sudėtį. Tai įgyvendinama pavaizduojant projektą, jo etapus ir priskirtas užduotis medžio tipu, kur matomi ir užduočių, etapų, bei projekto įgyvendinimo terminai.

3.23.7. Resursų tvarkymo komponentas

Resursai – tai laikas, užduotys, darbuotojai, dokumentai (failai) susieti su projektais. Komponentas pateiks informaciją apie projektui paskirtus terminus, darbuotojus, papildomą informaciją.

3.23.8. Grafikų piešimo komponentas

Komponentas skirtas atvaizduoti projekto įgyvendinimo terminui, t.y. priskirtų etapų ir užduočių atlikimo laikui. Komponentas remiasi „google charts“ biblioteka, kuri suteikia galimybę atvaizduoti duomenis stulpelių, pyrago, taškinių, linijinių tipų grafikais.

3.23.9. Žinučių komponentas

Komponentas suteiks galimybę vartotojams keisti žinutėmis sistemoje. Tokia funkcija reikalinga norint palikti kokį nors komentarą apie numatytą užduotį, atliktą užduotį ar pan.

3.23.10. Ataskaitų generavimo komponentas

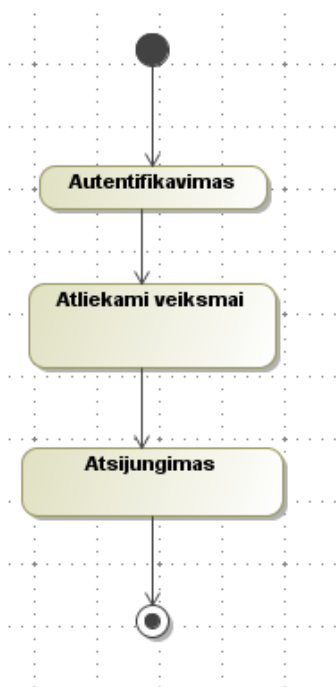
Komponentas paremtas „dhmlx grid“ biblioteka. Skirtas projekto statistikos peržiūrėjimui ir jos perkėlimui į popierinį variantą. Komponentas atvaizduos informaciją naršyklės lange tinkama atsispausdinti forma ant standartinio A4 formato lapo.

3.23.11. Interneto naršyklė

Viena iš šių interneto naršyklių: opera 9.0 arba aukštesnės versijos, google chrome 2.0 arba aukštesnės versijos, mozilla firefoz 3.0 arba aukštesnės versijos, internet explorer 6.0, 7.0, 8.0 arba aukštesnės versijos, safari 2.0 arba aukštesnės versijos. Su suteikta galimybe naudoti javascript, bei „sausainukus“.

3.24. Procesų vaizdas

3.24.1. Sistemos veikimas

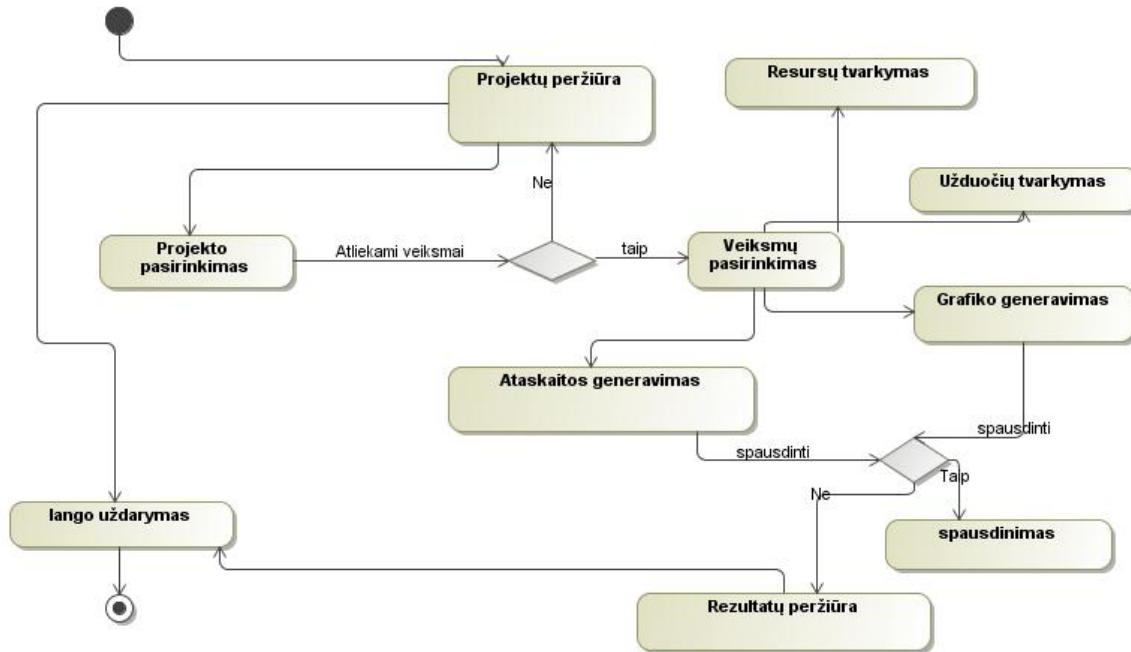


Paveikslukas 10 Sistemos būsenos

3.24.2. Vartotojo veiksmus naudojantis šia sistema apsprendžia run controleris:

- Sukuriama nauja sesija;
- Sukuriamas naujas (sesijai) vartotojas;
- Nustatomos vartotojo teisės;
- Parenkama kalba;
- Parenkamas užklauso vykdyimo kontroleris;
- Parenkamas vartotojo sąsajos šablonas;

3.24.3. Projektų tvarkymas



Paveikslukas 11 Projektų valdymo diagrama

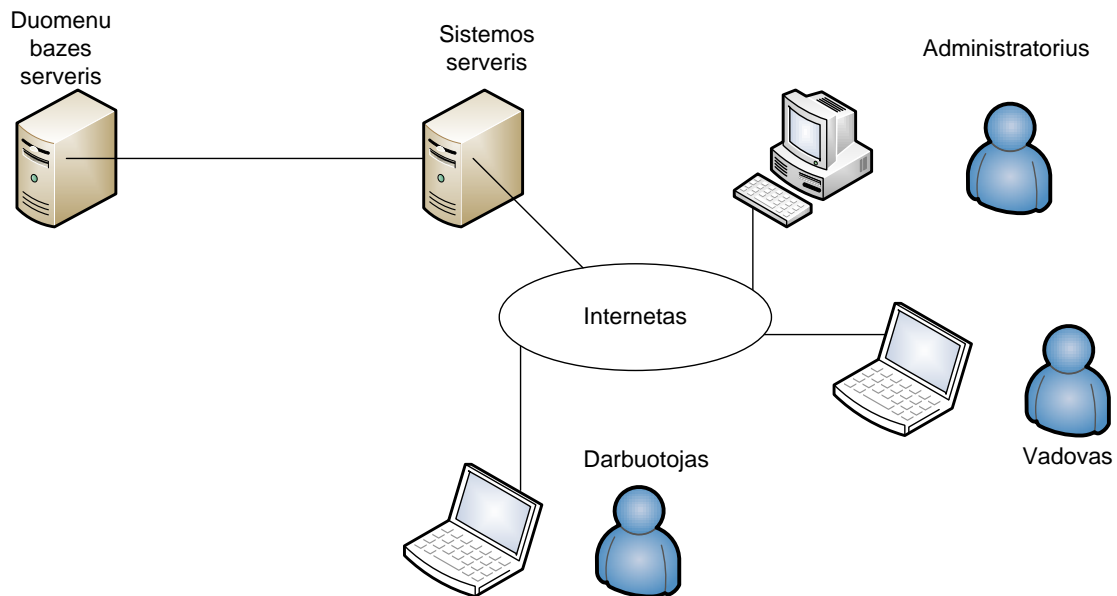
Vartotojo veiksmus naudojantis šia sistema apsprendžia projektų tvarkymo komponentas:

- Pateikiama projektų peržiūra;
- Pateikiama galimybė atlikti veiksmus pagal vartotojo teises;
- Pasirinkimas tvarkyti resursus;
- Pasirinkimas tvarkyti užduotis;
- Pasirinkimas pavaizduoti duomenis grafiškai arba atskaitos forma su galimybe atsispausdinti.

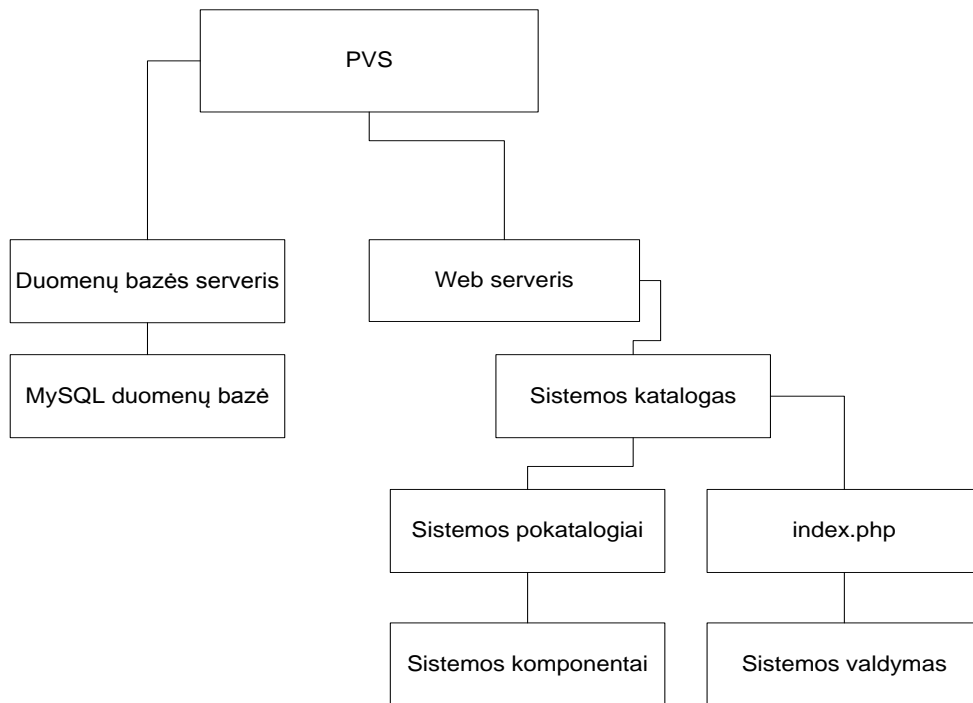
- Tvarko duomenų bazės įrašus;
- Tvarko sistemos fizinius failus (talpos atžvilgiu, grąžinimo);
- Daro sistemos atsargines kopijas.

3.25. Išdėstymo vaizdas

Paveikslėlyje pavaizduotas fizinis sistemos išdėstymas. Tačiau galima teigti, kad sistemos serveris ir duomenų bazės serveris veikia viename fiziniame kompiuteryje.



Paveikslukas 15 Sistemos išdėstymas



Paveikslukas 16 Sistemos failų išdėstymas

3.26. **Kokybė**

Paprasta ir interaktyvi vartotojo sąsaja iš keleto lengvai pasiekiamų langų leis šia sistema pasinaudoti ir nepatyrusiems vartotojams.

Sistemos funkcionalumo pasiekimas per interneto naršyklę šiuo metu yra pats patogiausias, kadangi savo kompiuterio aplinkoje nereikia diegti jokių papildomų programų, o ir sąsaja yra paprasta ir intuityvi.

Sistemos kūrimui panaudotas MVCH technika leis daryti pakeitimus vienoje komponentuose, neįtakojant kitų sistemos komponentų. Taip pat ši technika leidžia rašyti nepriklausomus vienetų testus.

3.27. **Testavimas**

3.27.1. **Testavimo tikslai ir objektai**

Laikoma, kad programinė įranga yra gera, jei ji atlieka numatytus veiksmus taisyklingai. Nėra įmanoma sukurti programinę įrangą neturinčią jokių defektų. Tačiau yra įmanoma sukurti programinę įrangą su kuo mažesniu defektų skaičiumi. Norint tai pasiekti, prieš

galutinai pristatant programinę įrangą, reikia su ja atlikti visą eilę skirtingų testų, ypač jei programinė įranga yra kuriama per sąlyginai trumpą laiką. Tačiau turint detalią testų specifikaciją galima pasiekti aukštą programinės įrangos patikimumo lygį. Aukštas patikimumo lygis bus pasiektas, jei kūrimo komandai bus pateikta detali testavimo specifikacija, ir komanda testuos specifikacijos atvejus vieną po vieno, tikrins objektą po objekto, kol patikrins visus (arba bent daugumą) reikalingus objektus – modulius bei funkcijas, duomenų srautus, programinės įrangos galimybes ir apribojimus.

Šis dokumentas turėtų apibrėžti galimus pavojų šaltinius ir išskirti vietas, kur numatomi galintys iškilti sunkumai. Čia bus atspindėtos tiek tipinės klaidos, į kurias reikėtų atkreipti ypatingą dėmesį kuriant produktą, tiek netipinės, kurios yra būdingos tik šiam produktui. Dokumentas skirtas aprašyti veiklos skirtos klaidų mažinimui strategijas.

3.27.2. Testavimo apimtis

Testavimas apims modulių įeinančių į sistemą ir jų susijusių dalių, bei bendro veikimo testavimą. Dokumente yra pateikiamos testavimo strategijos, testavimo atvejai ir kiti reikalavimai sistemos testavimui.

Sistemos klaidų aptikimo testavimai:

- Funkcinis testavimas
- Struktūrinis testavimas

Šie testavimai smulkiau aprašyti dokumento antrame skyriuje.

3.27.3. Pagrindiniai apribojimai

- Mažas testuotojų skaičius - dėl šios priežasties prarandama galimybė užtikrinti, kad viskas bus atlikta iki galo ir laiku. Neturint pakankamai testuotojų programinė įranga gali būti ištestuota nepilnai ir nežiūrint gilesnių lygių, arba gali būti praleisti kai kurie testavimo specifikacijos punktai, kurie turi didelę svarbą programinės įrangos efektyvumui.

- Patirties trūkumas - testavimo komandai neturint pakankamai patirties testuojant programinę įrangą, ji gali neatkreipti dėmesio į kai kuriuos neaiškius sutrikimus, su kuriais dar nebuvo susidūrusi.

3.27.4. Strategija

Sistemos prototipo testavimo strategija:

- Sistemos metodų funkcinis testavimas;
- Sistemos integracijos testavimas;
- Vartotojo sąsajų testavimas.

3.27.5. Vienetų testavimas

Bus atliekamas automatinis sistemos vienetų klasių, jų metodų testavimas. Metodai bus vykdomi su nustatytais pradiniais testiniais duomenimis. Pagal sistemos specifikaciją testiniams duomenims yra nustatomi laukiami rezultatai. Bendrai, yra sudaromi testavimo atvejai, kurių pagalba yra nustatoma ar metodas veikia korektiškai.

Testavimo atvejai sudaromi remiantis metodo struktūra. Tokiu atveju yra sudaromi struktūriniai testai. Numatomi testavimo atvejai, kurie leistų patikrinti metodo struktūrą pagal tokius kriterijus: patikrinami visi metodo operatoriai, patikrinamos visos metodo šakos, patikrinami visi metodo keliai.

Funkcinis testavimas kitaip vadinamas "Juodos dėžės" testavimu. Testavimui naudojant šį metodą remiamasi įėjimo, išėjimo ir vidinių būsenų kintamaisiais. Testavimo atvejai gaunami remiantis sistemos specifikacija.

Funkcinis testavimas bus taikomas pasirinktinai keliems programų sistemos komponentams arba klasių metodams. Testiniai atvejai bus sudaromi naudojantis sistemos specifikacija. Remiantis ekvivalentiniu testavimo metodu bus numatomos įėjimo reikšmės prie kurių gaunami numatyti išėjimai.

3.27.6. Integravimo testavimas

Atlikus pavienių sistemos vienetų testavimą atliekamas vienetų integravimo testavimas. Vienetų grupei taikomi tie patys testavimo kriterijai kaip ir pavieniams vienetams ir vienetų visuma gali būti laikoma kaip sudėtinis vienetas.

Papildomai turi būti užtikrinta tai, kad vieneto sutrikimas negali sąlygoti kitų sistemos vienetų nesklandumų, t.y. klaidos neturi plisti tarp vienetų ar jų grupių sąlygodamos sistemos neveiksnumą ar klaidingą veikimą.

3.27.7. Priėmimo testavimas

Šio etapo metu tikrinama ar sukurta sistema atitinka vartotojo reikalavimus ir poreikius. Užsakovas turi įvertinti nefunkcinius reikalavimus sistemai, kurie detalčiai yra pateikti „Reikalavimų specifikacijos“ dokumente.

3.27.8. Aukšto lygio testavimas

Sistemą tikrina užsakovas, po jos priėmimo. Pastebėtos klaidos yra ištaisomos, nauji reikalavimai daromi pagal atskirą susitarimą.

3.27.9. Testavimo resursai

Nedetalizuotas testavimo resursų sąrašas:

- Programinė įranga sistemos veikimui
 - WEB serveris su LINUX šeimos operacine sistema
 - MYSQL duomenų bazė
 - Naršyklė Firefox su priedu FIREBUG (stebėti ajax užklausas)
 - Programos kodo redaktorius
- Programos kodas
- Asmeninis kompiuteris;

Testams atlikti reikalingas vienas žmogus. Tai gali būti tiek sistemos kūrimu užsiimantis asmuo, tiek užsakovas (aukšto lygio testavimo atveju).

3.27.10. Testavimo įrankiai ir aplinka

Testavimui naudojami įrankiai rašomi ta pačia programavimo kalba kaip kad testuojamas kodas. Testavimas atliekamas bet kurioje operacinėje sistemoje, su įdiegta modernia

interneto naršykle (Firefox >= 3.6, google chrome >= 3.0, opera >= 10.00, safari >= 4.0, internet explorer >= 7.0).

3.28. Vartotojo dokumentacija

3.28.1. Sistemos struktūra

Sistema sudaro atskiri moduliai: informacija, kalendorius, projektai, žinutės, sąskaitos, resursai, žinynas, administravimas, greitisis meniu.

Informacija – prisijungus vartotojui prie sistemos parodo naujus įvykius, kuriuos fiksuoja sistema.

Kalendorius – skirtas užduočių tvarkymui, išdėstymui laike, priskyrimui asmenims, projektams ir pan.

Projektai – projekto užduočių plano sudarymas. Grafiškai leidžia išdėstyti projekto užduotis laike. Paskaičiuoja viso projekto trukmę pagal išdėstytas užduotis.

Žinutės – modulis skirtas vidiniam sistemos vartotojų bendravimui. Galima siųsti privačias arba viešas žinutes. Privačias gauna tik vienas asmuo, viešas gali gauti visi pasirinkti vartotojai jeigu yra sutikę gauti tokio tipo žinutes (vartotojo nustatymai).

Sąskaitos – sąskaitų faktūrų generavimas, jų redagavimo istorijos kaupimas.

Resursai – sistemoje talpinami ir visiems vartotojams matomi failai.

Žinynas – skirtas vidiniam naudojimui ir matomas tik pagal tam tikras suteiktas teises vartotojui. Šio modulio pagalba galima kaupti video, audio, vaizdų, tekstų formatų informaciją.

Administravimas – vartotojų valdymas, vartotojų teisių valdymas.

Greitisis meniu – „sparčiųjų“ klavišu pagalba arba pelės pagalba pasirenkami projektai greitam redagavimui arba projekto/užduoties pasirinkimui pasiekiamas iš visų meniu punktų. Atvaizduoja projektus ir užduoties medžio tipu pagal esamus susiejimus tarp projektų ir užduočių.

3.28.2. Vartotojo instrukcija

Detali vartotojo instrukcija su aiškinamaisiais paveikslėliais pateikta dokumente „Vartotojo dokumentacija“ pristatyto „Informacinės technologijos projektavimo vadyboje“.

3.28.3. Diegimas

Diegti sistemos nereikia. Sistema yra įdiegta ir paruošta naudojimui. Sistema pasiekama interneto ir interneto naršyklės pagalba.

3.28.4. Klaidos

Pastebėjus klaidos pranešimą ar netinkamai suveikus sistemai rašyti adresu:

tomas@deform.lt, test@deform.lt . Taip pat reikėtų prisegti klaidos įvykio ekranvaizdį.

3.28.5. Autorių teisės ir gretutinės teisės

Autorius visas programinės įrangos ir programinio kodo teises perduoda užsakovui. Užsakovui suteiktos teisės: išleisti kūrinį; adaptuoti ar kitaip perdirbti kūrinį; platinti kūrinio originalą ar jo kopijas juos parduodant, nuomojant, teikiant panaudai ar kitaip perduodant nuosavybėn arba juos valdyti; importuoti kūrinio originalą ar jo kopijas ir kt.

Kitiems asmenys taip pat gali laisvai naudotis sistemos programiniu kodu ne komerciniams tikslams.

3.28.6. Pakeitimų sąrašas

- Kontaktų tvarkymo atsisakyta, įtraukta į vartotojų tvarkymą.
- Resursų ir failų valdymas apsiriboja tik sistemai prisikirtų failų valdymu.
- Bendravimo modulis pakeistas į žinučių modulį.
- Ataskaitų generavimo modulis ir grafikų piešimo modulis šiuo metu pakeistas į pdf failo formato sugeneruotas ataskaitas.

4. TYRIMAS

4.1. Pagrindinis projektų valdymo sistemos funkcionalumas

Žemiau pateikiamas dažniausiai sutinkamas ir apibendrintas projektų valdymo sistemų, užduočių tvarkymo, planavimo įrankių funkcionalumas su angliškais paaiškinimais skliausteliuose:

- Apžvalga (dashboard, overview);
- Projektai (portfolio, plan) ;
- Užduotys (tasks, milestones, to-dos, tickets, issues, bugs);
- Laikas (calendar, timesheet);
- Bendradarbiavimas (collaboration, chat, forum, comments) ;
- Resursai (documents, files, people, inventory);
- Klientai, kontaktai (companies, clients, contacts, addressbook) ;
- Wiki (wiki per project, knowledgebase) ;
- Ataskaitos (reports, gantt charts);
- Sąskaitos (invoice);
- Paieška ;
- Administravimas (templates, users).

4.2. Mano siūlomas projektų valdymo sistemos meniu

Norint kad sistema būtų nesudėtinga naudotis, reikėtų apjungti funkcionalumą į grupes. Meniu punktus, apžvalga ir paieška, galima pateikti tikrai grafiškai suprantamomis ir intuityviai, suprantamai, kad reikia spausti ikonomis, kadangi apžvalga yra pradinis sistemos vaizdas, tai galima panaudoti „namai“ ikoną, kuri labai plačiai naudojama internetiniuose puslapiuose, bei „didinamasis stiklas“ ikoną ant kurios paspaudus išsiskleistų paieškos laukas į dešinę nuo pačios ikonos. Pagrindinis meniu sudarytas iš tokių punktų:

- Ikonos (namai, paieška);
- Projektai;
- Užduotys;

- Bendradarbiavimas;
- Resursai;
- Ataskaitos;
- Sąskaitos (matomas tik tam tikrai vartotojų kategorijai);
- Administravimas (matomas tik tam tikrai vartotojų kategorijai).

Vartotojo vardas (pateikiami kiekvienam vartotojui suasmenintas funkcionalumas: nustatymai, atsijungti ir pan..)

Meniu grupavimo tikslas – pateikti labai parastą ir būtiną funkcionalumą vartotojui, t.y. sudaryti tinkamą pirmą įspūdį apie sistemą. Šiuo būdu mes pateiktume aišku ir labai paprastą meniu sistemos vartotojui, kuriame nereikėtų ilgai klaidžioti spaudinėjant vienus ar kitus meniu punktus norint kažką konkrečiau atlikti, kadangi kiekvienas atskiras sistemos funkcionalumo tipas yra labai aiškiai sugrupuotas. O papildomą funkcionalumą pateiktume vartotojui per meniu punkto „vartotojo vardas -> nustatymai“ pasirinkimo galimybes. Vartotojas pats galėtų įjungti arba išjungti papildomą (teikiamą) funkcionalumą, kurio jis pats ieškotų per sistemos nustatymus. Sekančiuose punktuose platesni paaiškinimai apie kiekvieną punktą.

4.2.1. Apžvalga

Tai yra pagrindinis (pradinis) sistemos vaizdas, kuris būtų matomas prisijungus prie sistemos arba pasirinkus meniu punktą „namai“. Tinkamiausias egzistuojantis sprendimas manau yra realizuotas „basecamp“ . Kadangi viskas yra pateikiama labai paprastai ir aiškiai. Visi nauji įvykiai sistemoje turi susietas skirtingas spalvas, ilgesnį laiką pasinaudojus įrankių tai labai įsisavinama ir iš karto matomi tam tikro tipo akimis ieškomi įvykiai. Tačiau, visiškai nenaudingai išnaudota dešinioji pusė. Šioje vietoje būtų galima pateikti nuorodas į greituosius veiksmus su „mini“ formomis. Taip būtų suteikta galimybė tik prisijungus prie sistemos priskirti nesudėtingą aprašymą turinčią užduotį arba siūlomo duomenų sąrašų atvaizdavimo komponentu (aprašytas toliau) pateikti

naujausius įvykius ir virš jo arba po juo pateikti greitojo meniu nuorodas. Pavyzdys egzistuojančio „basecamphq“ sprendimas pateikiamas žemiau:

The screenshot displays the 'Latest activity across your projects' section of the Deform application. It features a sidebar with the Deform logo and a main content area with several project-specific activity lists.

Late Milestones

- 27 days late: [Future actions](#) [AL-ROUND | AI - Round | DEFORM]
- 30 days late: [Preferred Developer Consultant](#) [DEFORM | Deform - the new era! | Egle Šertvyčiūtė]
- 58 days late: [Deform's reforma](#) [DEFORM | Deform - the new era! | DEFORM]

AL-ROUND — AI-Round Cabinets

Task	Assigned to	Due Date
Improvements and fixes (will be specified later) Testing	Assigned to DEFORM	Yesterday
Testing Testing	Assigned to Egle Š.	Yesterday
Programming the ROOM preview Information specifications	Assigned to Justas	Yesterday
Programming the 3D preview Information specifications	Assigned to Laurynas	Yesterday
Programming drag and drop functions Information specifications	Assigned to Tomas K.	Yesterday
Design details Information specifications	Assigned to Vidmantas	Yesterday
Preparing the design specifications Information specifications	Assigned to Arthur B.	Yesterday
Preparing the specifications for programming Information specifications	Assigned to Egle Š.	Yesterday

AL-ROUND — AI - Round

Activity	Posted by	Date
Comment Re: Navigation	Posted by Harald J.	Apr 14
Comment Re: Navigation	Posted by Harald J.	Apr 12
Comment Re: Technologie page	Posted by Harald J.	Apr 12
Comment Re: Navigation	Posted by Egle Š.	Apr 12
Message Technologie page	Posted by Harald J.	Apr 12
Message Navigation	Posted by Egle Š.	Apr 11
Comment Re: Gallery - Galerie - urgent -	Posted by Egle Š.	Apr 11
Comment Re: Gallery - Galerie - urgent -	Posted by Harald J.	Apr 11

DEFORM — D-Blast

Activity	Posted by	Date
Comment Re: Žaidimo komentarai	Posted by Egle Š.	Mar 30
Comment Re: Žaidimo komentarai	Posted by Arthur B.	Mar 28
Message Žaidimo komentarai	Posted by Egle Š.	Mar 28
To-do Surašyti visus komentarus štai taip: > Messages > Add a comment Komentarai apie žaidimą	Assigned to DEFORM	Mar 28
To-do Pažaisli žaidimą Komentarai apie žaidimą	Assigned to DEFORM	Mar 28

DEFORM — Deform - the new era!

Activity	Completed by	Date
To-do 4- "Sukurti vaizdą" - naujo įvaizdžio suformavimas Reforma step-by-step	Completed by Egle Š.	Dec 21
File Deform is a web company that thrives on excellence and innovation.docx	Uploaded by Egle Š.	Dec 16
File Company information.docx	Uploaded by Egle Š.	Dec 15
To-do 2- Suformuoti mišinį (išskirti opsibrėžti, kas mes esame ir ko siūlome) Reforma step-by-step	Completed by Egle Š.	Dec 14
To-do 10- Sukurti DEFORM FB puslapį Reforma step-by-step	Completed by Egle Š.	Dec 14
To-do 3- Sukurti darbo metodiką-koip mes dirbame (vidaus struktūra ir procedūros) ir koip mes aptarnaujame klientus (step-by-step) Reforma step-by-step	Completed by Egle Š.	Dec 13
To-do 1- Išgryninti paslaugas (išskirti pagrindines ir antrines paslaugų grupes) Reforma step-by-step	Completed by Egle Š.	Dec 13
To-do 1- Surinkti informaciją apie kompaniją ir parašyti trumpą aprašymą (250 žodžių) Tapimas preferred facebook developer	Completed by Egle Š.	Dec 13

Your global feeds ([read me first](#))

- [Global RSS Feed](#): Be notified about major activity across all your projects.
- [Global iCalendar](#): Get milestones from all your projects in a single iCalendar feed.

Paveikslukas 17 Basecamphq pradinis puslapis

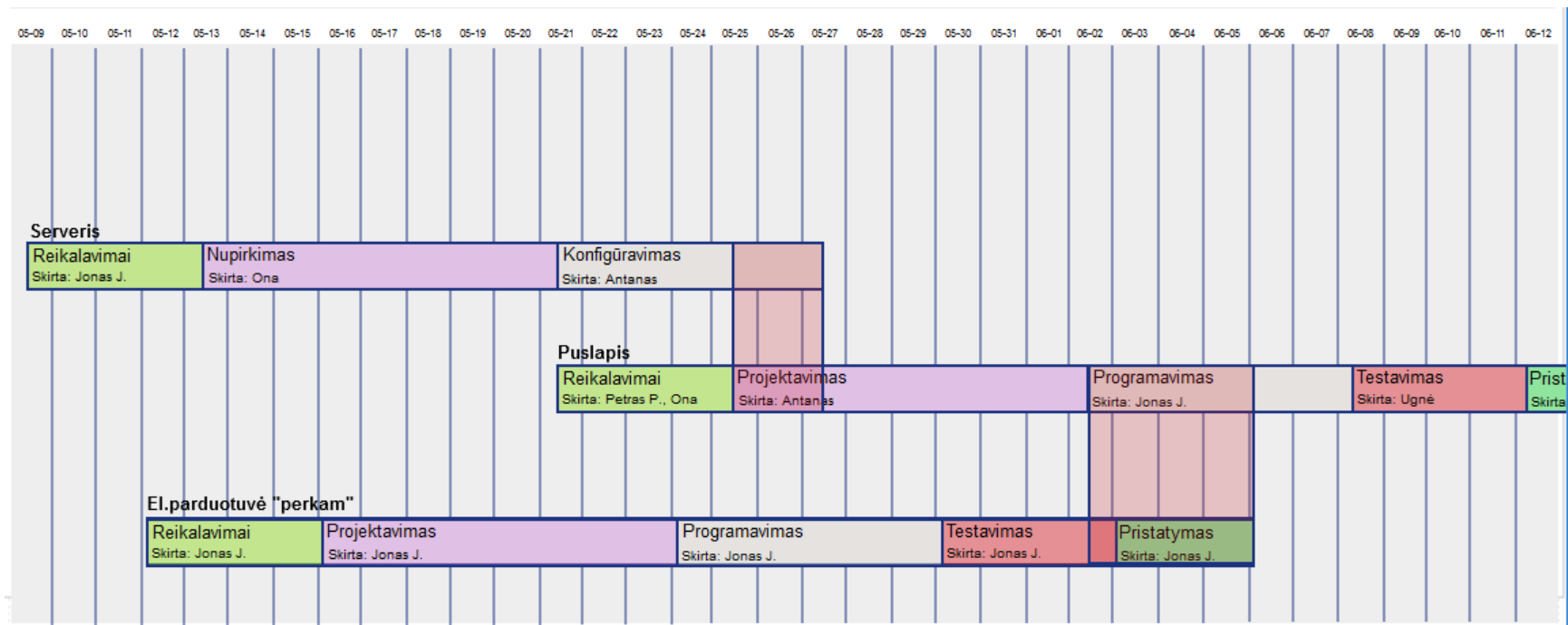
Modifikuoto sprendimo nauda būtų pačios sistemos detalus įvykių pateikimas pagal tai, su kokiomis užduotimis ir projektais yra susietas konkretus vartotojas, nes sistema gali teikti informaciją apie visus įvykusius įvykius ir elektroniniu paštu (ar mobiliųjų tel. sms) priminimų tipu, bet prisijungus prie sistemos iš jos taip pat norima gauti kažkokią naudingą informaciją. Šiuo būdu būtų siekiama užtikrinti, kad niekas nėra atliekama be žinios. Jeigu vartotojui buvo priskirta užduotis arba parašytas komentaras apie užduoties atlikimą, arba, kad užduotis nepatvirtinta, arba padaryti papildomi paaiškinimai kaip

atlikti užduotį, tai nereikėtų net eiti į meniu punktą „užduotys“, o būtų iškart matoma tik prisijungus. O greitojo meniu nuorodos, suteiktą galimybę kartais atlikti nesudėtingas funkcijas iškart po prisijungimo, nepasirenkant papildomo sistemos meniu punkto. Greitasis meniu bus pasirenkamas pačio vartotojo daugiau, ir ne taip kad tik keli pateikti pavyzdžiai iš kurių jis gali būti sudarytas, o pavyzdžiui, kaip android sistemoje, pateikiamas visų galimų pasirinkti sistemos atliekamų funkcijų sąrašas.

4.2.2. Meniu punktas Projektai

Egzistuoja stereotipas, kad projektai turi būti vaizduojami gantt tipo diagramomis. Gantt tipo diagrama yra naudingas atvaizdavimo būdas tik vienam projektui, tačiau ne kai jų turime kelis ir norime iškart palyginti ekrane. Gantt tipo diagrama taip pat nėra tinkama kai projektas yra sudarytas iš labai daug etapų ar užduočių, kadangi diagrama tampa labai sunkiai skaitoma, nes priėjus jos pabaigą būna sunku prisiminti, kas buvo vaizduojama jos pradžioje, taip pat tai galioja ir resursų sąrašui, kuo jis ilgesnis tuo sunkiau jį visą skaityti. Gantt tipo diagrama būtų galima keisti PERT diagrama, bet tai dar sudėtingesnė diagrama, nors, galbūt mano nuomone ir labiau tinkama. Kadangi mažai žmonių nesusidūrusių su grafų teorija ir nemokėdami pagrindinių grafų teorijų taisyklių nesupras kaip redaguoti tokio tipo diagramą. Mano siūlymas atvaizduoti projektą, ne kaip vieną valdomą ir peržiūrimą objektą, o kelis juos su galimybe palyginti tai būtų gantt tipo diagramos modifikacija. Ši modifikacija neturėtų resursų sąrašo kairėje diagramos pusėje ir neturėtų laiptuoto etapų ir užduočių atvaizdavimo. Projektas būtų atvaizduojamas vienoje eilutėje, kur skirtingos spalvos ir papildoma laukelio informacija suteiktų visą reikalingą informaciją apie projektą. Laukeliai eitų vienas paskui kitą priskirtų laiko intervalų tvarka. Projektai pagal sukūrimo laiką būtų išdėstyti vienas po kito atitinkamai, kur būtų matomi jų resursų persidengimai. Tai suteiktų svarbios informacijos apie tai, kaip tinkamiau paskirstyti resursus, šiuo atveju žmones, kurie priskirti projektui, įgyvendinti. Tai būtų daug paprasčiau skaitoma diagrama nei gantt ar PERT, bet labai aiškiai pateiktą informaciją apie netinkamai skirstomus resursus atliekant daugiau nei vieną projektą. Dažniausiai praktikoje labai retai būna, kad įmonė vienu metu turėtų tik vieną projektą, kurį galėtų sėkmingai valdyti pasitelkus gantt tipo diagrama besiremiančius projektų valdymo įrankius. Žemiau pateikiamas pavyzdys, tai nėra pilnas

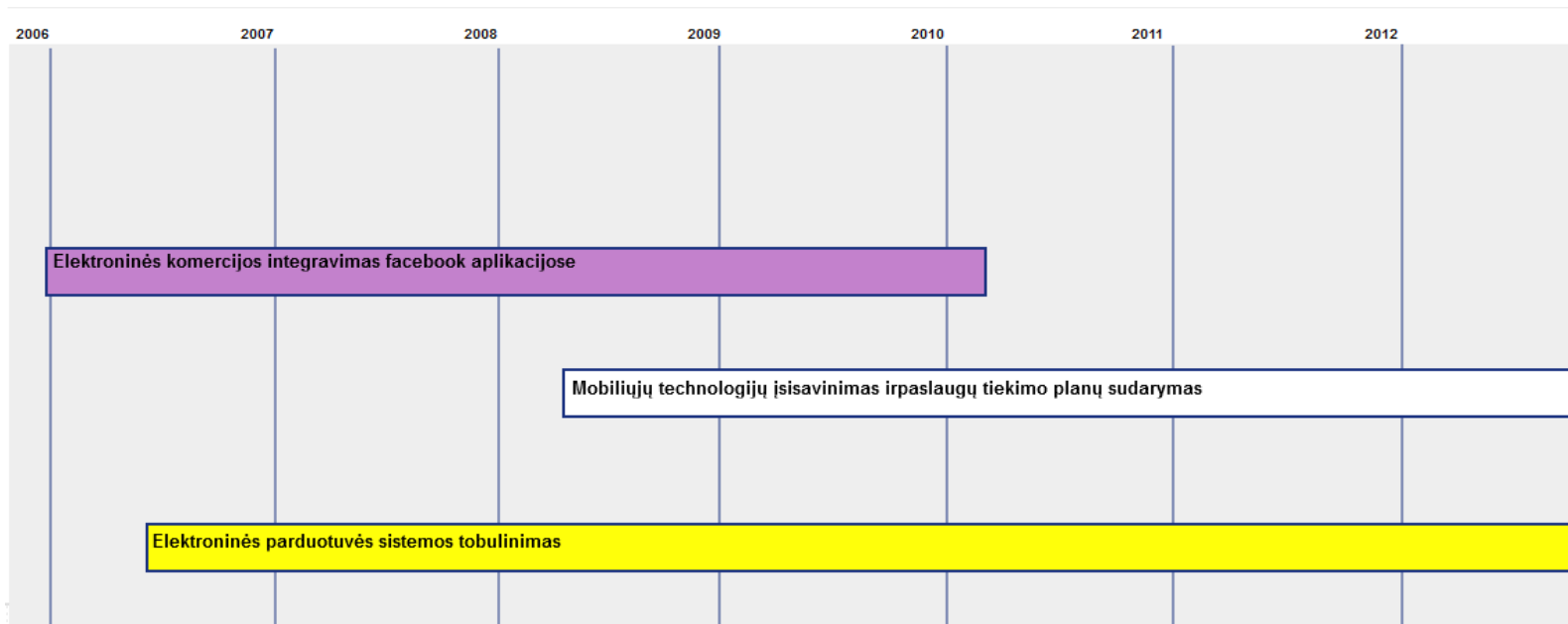
diagramos pavyzdys, pateiktas fragmentas kuriame puikiai matomas resursų persidengimas:



Paveikslukas 18 Siūlomos gantt tipo diagramos modifikacijos pavyzdys

Iš diagramos matome, kad Antanui ir Jonui J. reikės atlikti du darbus vienu metu, kad nėra gerai. Darbas gerai gali būti atliekamas kai yra daromas vienas, o ne keli vienu metu. Žmogus nėra mašina, kuri gali atlikti kelias sudėtingas užduotis vienu metu.

Pačios vartotojo sąsajos detalesnis aprašymas bus pateiktas prie užduočių valdymo aprašymo (slankiojimas ir panašūs dalykai). Šiuo atvaizdavimo tipu pasinaudojus taip pat galima atvaizduoti numatytą ateities planą vizualizavimą, o taip pat ir generuoti bendrų vykdomų veiklos sričių vaizdus. Taip būtų išlaikomas informacijos atvaizdavimo vientisumas. Pavyzdys pateikiamas žemiau:



Paveikslukas 19 Projektų peržiūra

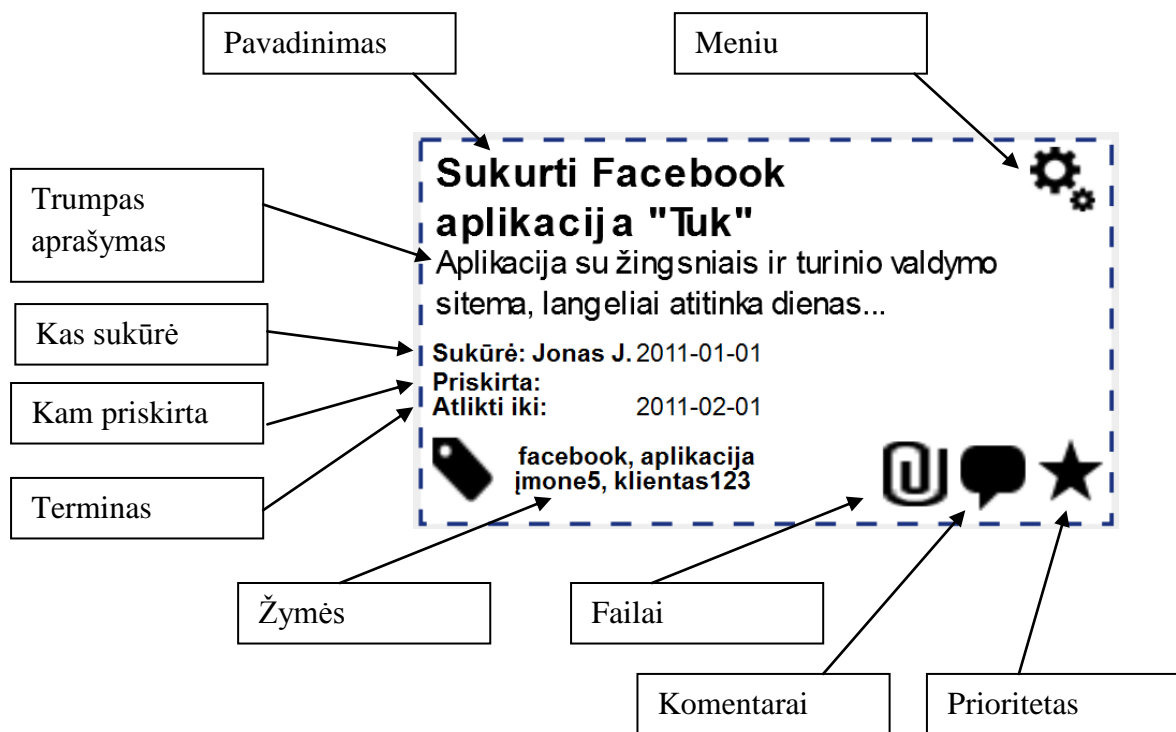
4.2.3. Meniu punktas Užduotys

Užduočių valdymo funkcija yra neatsiejama projektų valdymo sistemoje. Nemažai užduočių valdymo sistemų ir įrankių naudojamos tiesiog kaip objektų valdymo įrankiai arba bent tokį įvaizdį apie įrankių teikiamas galimybes suteikia aprašymuose. Dažniausiai užduočių tvarkymui pasitelkiamas kalendorius. Kalendoriaus galimybė atvaizduoti užduotis „dienes“, „mėnesio“, „metų“, „sąrašo“ vaizduose yra tinkamos ir priimtinos. Tai jau kaip tradicija, stebėti kalendoriujė išdėstytas užduotis arba naudotis užduočių sąrašu. Tačiau, manau yra vienas atvaizdavimo būdas šiek tiek praktiškesnis. Tai būtų darbų eigos sąrašas angl. „workflow“. Atitinkamai formuojant šį sąrašą galima sudaryti kelis užduočių atvaizdavimo būdus, kurie teiktų daugiau informacijos nei kalendorius.

Užduotis kaip objektas gali saugoti tokią reikalingą informaciją:

- Pavadinimas;
- Trumpas aprašymas;
- Kas sukūrė;
- Kam priskirta;
- Sukūrimo data;
- Terminas;
- Prioritetas;
- Komentarai;
- Failai.

Pavadinimas, aprašymas, kas sukūrė, kam priskirta (dažniausiai skirtingos spalvos būdu), prioritetas šio tipo informacija dažniausiai yra pateikiama užduočių tvarkymo kalendoriuose arba užduočių valdymo kalendoriuose su papildomu funkcionalumu angl. „advanced“. Terminas būna nurodomas kaip užduotis priskirta keliems laiko intervalams (dienoms). Visą šią informaciją galima tinkamai pateikti viename užduotį atvaizduojančiame lauke. Tokio laukelio pavyzdys pateikiamas žemiau:

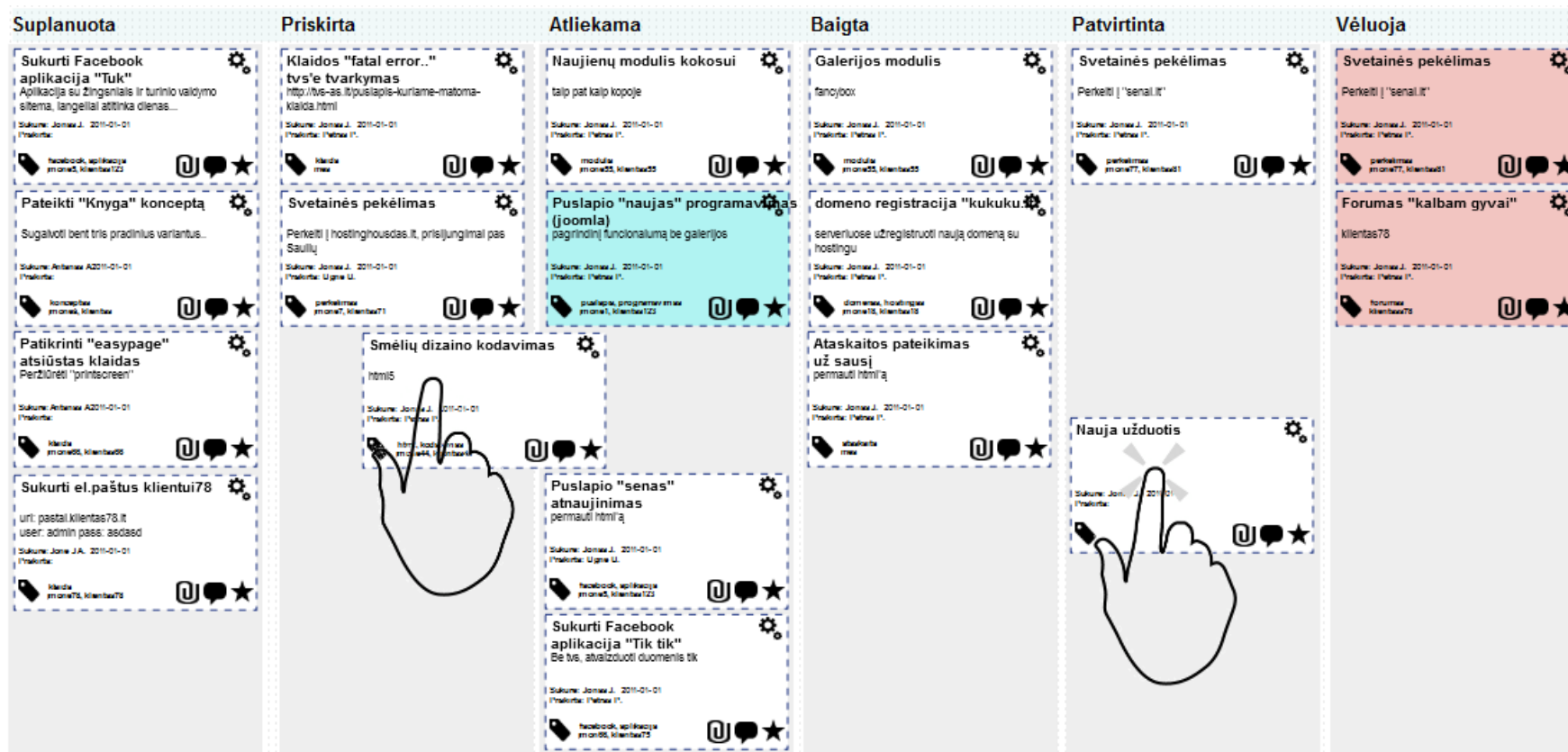


Paveikslukas 20 Užduoties lauko pavyzdys

Skirtumas nuo visų kitų, būtų žymės, kurios nėra privalomos, tačiau suteiktų galimybę realizuoti virtualius projektus. Tokius projektus, kurie nebūtų pažymėti projektų sąrašė atskiru objektu, tačiau įvykdžius paiešką pagal tokio projekto raktinius žodžius, sistema pateiktų visą turimą informaciją apie juos. Taip pat komentarai bei failai. Komentarai reikalingi tam, kad staiga pažymėjus užduotį kaip svarbesnę už visas kitas, būtų galimybė tai paaiškinti. Failai gali būtų įvairaus tipo ir skirti įvairiems tikslams, gali pateikti ilgus ir sudėtingus užduoties aprašymus ar paaiškinimus, pvz.: diegimo papildomų įrankių reikalingų užduočiai įgyvendinti. Papildomas meniu skirtas užduoties susiejimui su projektu. Meniu punktas gali pateikti galimybę peržiūrėti užduotį projektų vaizde, sudėtingą užduoties redagavimo formą su daug pasirinkimo galimybių, užduoties pašalinimo pasirinkimą.

Šiuo būdu aprašomus užduočių sąrašus galima atvaizduoti darbų eigos komponentu. Principas užduočių terminams nurodyti panašus kaip kalendoriuje, tačiau kalendorius negali būti taip lengvai transformuojamas pagal nurodytą šabloną į kitokio tipo atvaizdavimą. Taip pat kalendorius negali atvaizduoti užduočių grupių (etapų) taip

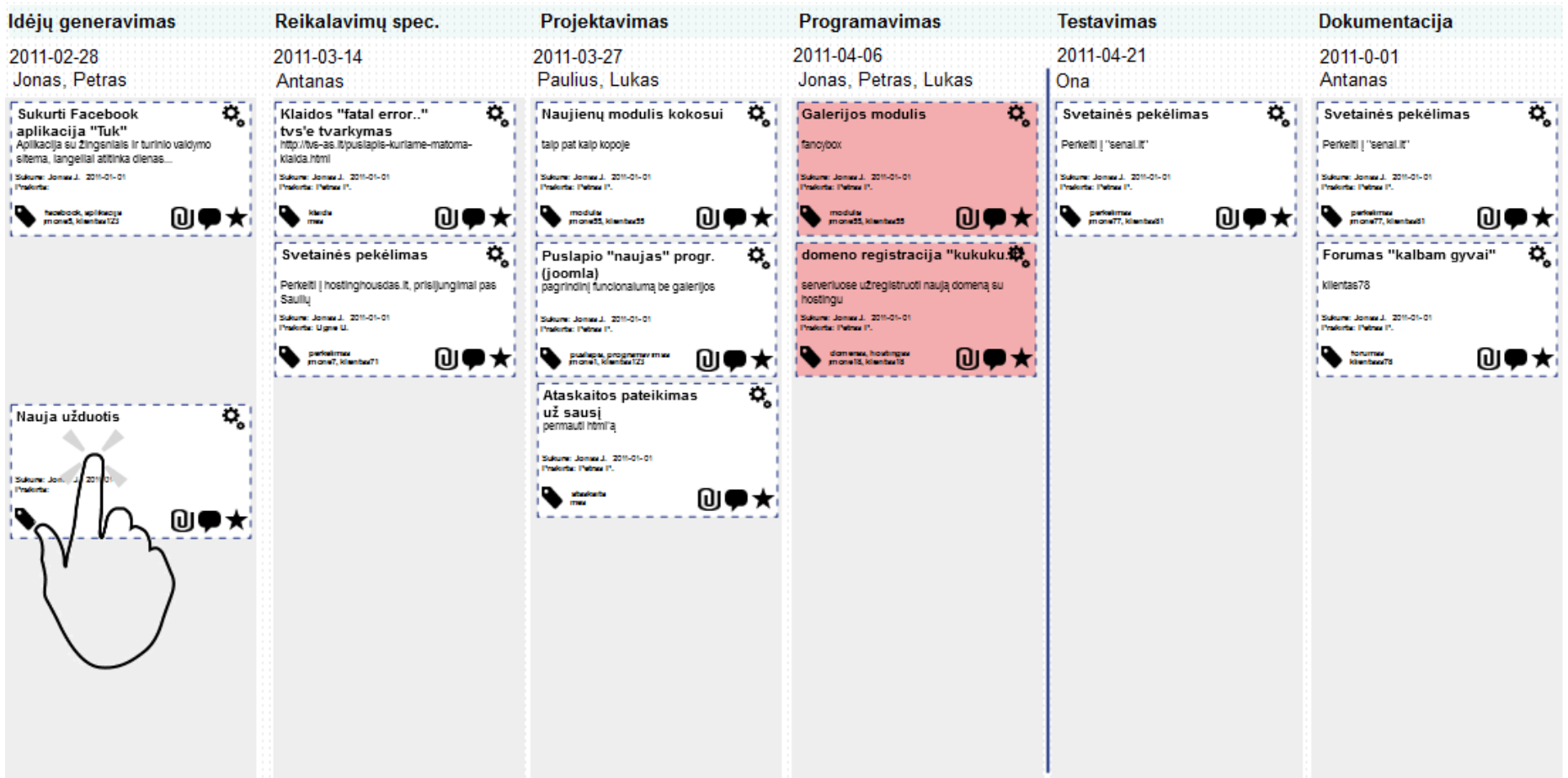
informatyviai kaip galima tai atlikti naudojanti darbų eigos sąrašą. Pavyzdys pateikiamas žemiau:



Paveikslukas 21 Darbų eigos atvaizdavimo pavyzdys

Pavyzdyje matome, kad užduotys negali būti priskirtos projektams, kadangi laukas „žymė“ nėra privalomas. Svarbios užduotys pažymėtos mėlsva spalva, o užduotys kurių terminas pasibaigęs automatiškai persikelia į paskutinį stulpelį „Vėluoja“. Užduočių valdymas vyksta labai sklandžiai pertempimo būčių perkeliant užduotį iš vieno stulpelio į kitą. Naujos užduoties sukūrimas taip pat labai paprastas,

tereikia du kartus spragtelėti ant pasirinkto stulpelio. Šis atvaizdavimo būdas tinkamiausias kai užduočių neskirstome griežtai projektams, o turime tam tikrus užduočių atlikimo etapus. Tada galime stebėti visas esamas užduotis ir jų būsenas. Klausimas išlieka tik toks, kada užduotį automatiškai pašalinti iš stulpelio „patvirtinta“, nes atliekamas pats veiksmas patvirtinimas ir norima, kad asmenys kuriems ta informacija yra svarbi, kurį laiką dar matytų, kad užduotis atlikta ir patvirtinta. Galbūt šis veiksmas „užduoties pašalinimas“, turėtų būti ne automatinis, o priskirtas vienai vartotojo kategorijai su aukštesnėmis teisėmis. Vėluojančių užduočių perkėlimas ir išryškėjimas kita spalva atliekamas automatiškai, kadangi jos persikelia iš stulpelio priskirta arba atliekama. Nes informacija apie priskyrimą saugos pats užduoties objektas, o stulpelyje atliekama laikyti vėluojančią užduotį nėra logiška. Toliau pateikiamas pavyzdys kaip šis darbo eigos atvaizdavimo būdas gali sutapti su kalendoriaus vaizdu ir pateikti užduotis laiko intervaluose taip pat specifikuotai tik vienam konkrečiam projektui, pagal tai kokius etapus numatėme projekto vykdymo eigoje:



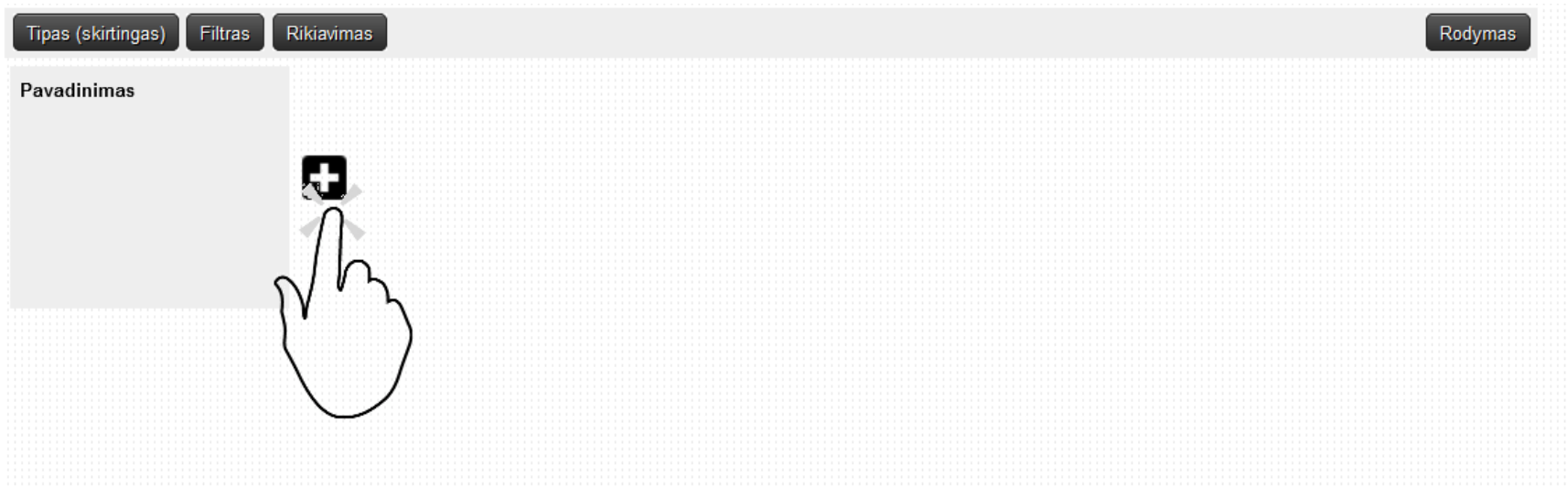
Paveikslukas 22 Vieno projekto darbų eigos pavyzdys

Pateiktame pavyzdyje aiškiai matome projekto etapus ir priskirtus žmones, kurie yra už tai atsakingi, bei vykdymo terminus.

Atvaizduojami etapai ir priskirti žmonės, atitinka Ganto diagramos kairiąją pusę. Tačiau šiuo atveju esant didesniam etapų skaičiui ar

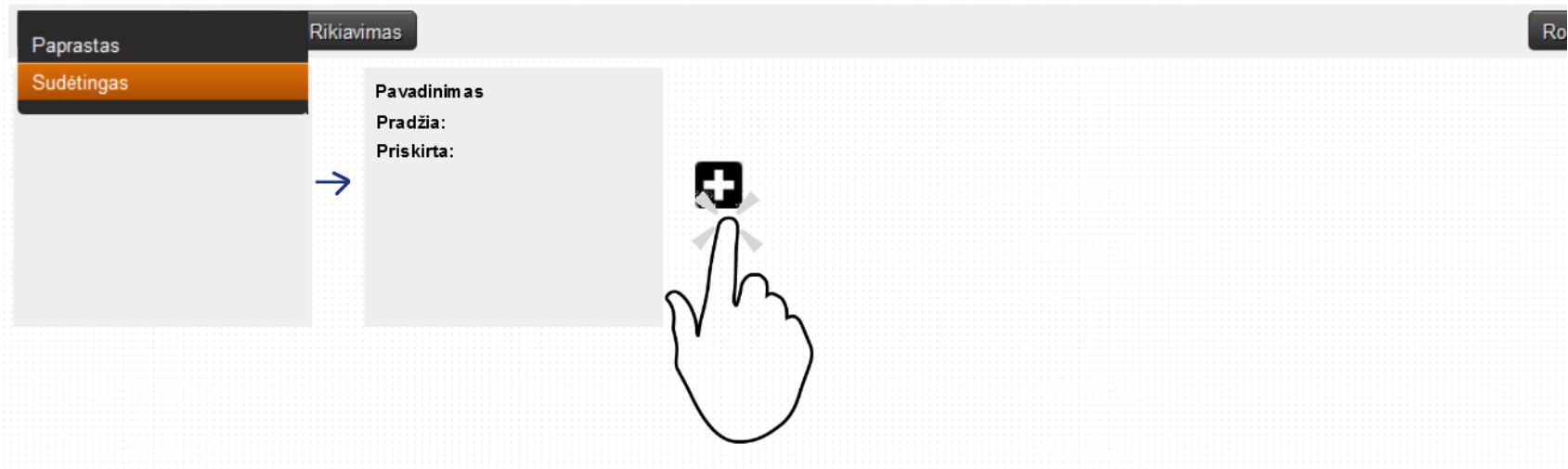
užduočių skaičiui priskirtų vienam etapui diagramos (atvaizdavimo) skaitymas yra lengvesnis. Kadangi vienam etapui priskirtas užduotis peržiūrėti tereikės slinkti vaizdą žemyn, o norint peržiūrėti visus projekto etapus reikės vaizdą slinkti į kairę arba į dešinę. Ryški linija vaizduoja esamą datą. Taip lengvai pabrėžiame dabartinę projekto eigą kur šiuo metu esame projekto vykdymo eigoje (kiek padaryta iš viso). Šis atvaizdavimo būdas užduočių tvarkymui suteiktų universalumo, kadangi jį galima taikyti tiek visiškai su projektais nesiejamomis užduotimis iki vienam konkrečiam projektui skirtu stulpelių išdėstymo pagal projekto vykdymo eigos etapus, taip pat iki skirtingas veiklas vykdančių įmonių užduočių suskirstymų pagal jų darbo procesus.

Taip pat pažymėtinas ir šio darbų eigos atvaizdavimo būdo, naujų atvaizdavimo būdų sudarymo paprastumas. Administravimo srityje suteikiama galimybė kurti vartotojui (su atitinkamomis teisėmis) šablonus darbų eigos atvaizdavimo. Darbų eigos šablonai ir yra tas principas kuriuo šis atvaizdavimo būdas tampa universaliu. Darbų eigos šablono kūrimo pavyzdys pateikiamas žemiau:



Paveikslukas 23 Darbų eigos šablono sudarymo pavyzdys

Pavyzdyje taip pat matomas antro lygio meniu pasirinkimai tik tam, kad suprastume bendrą vaizdą koks paprastas šablonų kurimo punktas. Atliekamas tik vienas veiksmas “ pridėti” ir antras “saugoti”, kuriant paprastą “numatytąjį” darbų eigos šabloną. Kuriant sudėtingą šabloną, reikėtų pasirinkti iš meniu punkto “Tipas” tipą “sudėtingas”, tada vartotojui būtų suteikiama galimybė priskirti šablono stulpeliams (projekto etapams) vykdymo terminus, resursus – žmonės kurie tas užduotis vykdys. Pavyzdys pateikiamas žemiau:

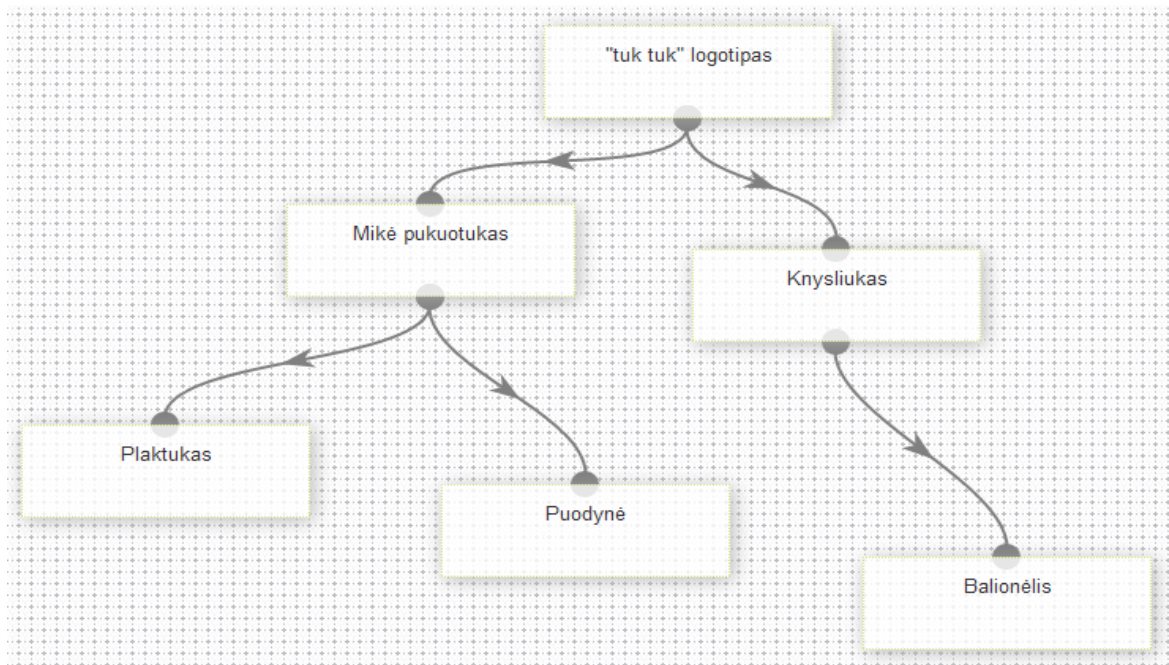


Paveikslukas 24 Sudėtingo šablono sudarymo pavyzdys

Norint, kad užduočių atvaizdavimo tipas “darbų eiga” taptų visiškai universalus užduočių tvarkymo įrankis, reikėtų suteiktų vartotojui galimybę pasirinkti net laukus/tipus pagal kuriuos bus išskirti darbų stulpeliai. Tačiau norint pasiekti šį funkcionalumą visų pirma tektų į projektų valdymo sistemą ingertuoti automatinio formų kūrimo įrankį, o antra išvengti nelogiškų vartotojo grupavimo pasirinkimų, kas turbūt tirkai nėra lengva. Mano nuomone pakaktų universalumui ir tų laukų, kurie būtų numatyti, leidžiami vartotojui pasirinkti stulpelių grupavimui, kadangi taip jau gauname tris skirtingus tipus, kurie atitinka pagrindinius užduočių tvarkymo atvejus.

Kalendorius – šis vaizdavimo tipas nėra atmestinas. Labiau akcentuoju darbų eigos vaizdavimo tipą, nes jis tinkamesnis vartotojo kategorijai „Vadovas“, „direktorius“, bet yra tinkamas ir varotojo kategorijai „Darbuotojas“, tačiau tik tuo atveju, kai vartotojas „darbuotojas“ priskiria užduotį pats, kuri nėra susieta su atlikimo terminais, bet norima pažymėti apie papildomai atliktą užduotį. Tada pateiktame sistemos greitajame meniu arba pasirinkus projektų eigos atvaizdavimo tipą ir stulpelį „Baigta“ būtų galima greitai pažymėti atliktą užduotį ir vartotojui „Vadovas“ teliktų patvirtinti (pažymėti tai, kad jis matė). Kaip bebūtų vartotojui „darbuotojas“ aktualesnis yra kalendoriaus užduočių atvaizdavimo tipas, nes jis pateikia dienos ar kelių, ar savaitės ir pan. laiko tarpo užduotis skirtas tik jam, kurios šios kategorijos vartotojui ir yra aktualiausios. Taip pat pažymėtina yra tai, kad praktinės dalies darbo metu, suteikus užduočiai tipą „susitikimas“ ir galimybę tai pažymėti žemėlapyje vartotojams labai patiko. Manau ši užduočių atvaizdavimo galimybė taip pat turėtų išlikti sistemoje. Iš to seka, kad užtuotims suteikus tipus (užduotis, susitikimas, klaidą) sistemoje, ne vartotojo pasirinkimą, vėliau šią projektų valdymo sistemą bus galima lengvai plėsti naudojantis ta pačia duomenų bazės struktūra. Kadangi bus galima žymėti ne tik užduotis, bet ir susitikimus, aptiktas klaidas, nes sukurti įrankį kuris teikia klientams paslaugą apie suteiktų paslaugų klaidų žymėjimą, kurių pranešimai būtų apdorojami ir valdomi pasitelkiant šią projektų valdymo sistemą (bet tai nukrypimas į funkcionalumą). Tačiau taip pat svarbus, kadangi jis išplaukia iš pačio vizualizavimo būdo sukūrimo (užduoties tipo, reikalingo vizualizavimui). Dar pažymėtina ir tai, kad kalendorius taip pat paprastai valdomas, tai įgyvendinau praktinėje dalyje.

Prie užduočių valdymo taip pat reikia pažymėti tai, kad ne visos užduotys turi būti kuriamos, kad būtų priskirtos, arba turėtų užduotis – reikia atlikti statusą. Angl. “mindmap” – sistema turi teikti šį funkcionalumą. Kadangi tai yra projektų valdymo sistema, teikia užduočių valdymo funkciją, bet gali teikti ir tai kas yra prieš projekto pradžią, t.y. minčių lietaus valdymą. Tam įgyvendinti tinka grafos. Pasitelkiant html5 ir javascript technologijas tai galima įgyvendinti. Šia minčių dėstymo ir susiejimo diagrama būtų galima atvaizduoti net įvairius procesus (uml, reikia tik skirtingų rodyklių, bet paprasčiau). Pavyzdys pateikiamas žemiau:



Paveikslukas 25 Minčių dėstymo pavyzdys

Šio komponento valdymo principai:

- Spaudžiant du kartus pelyte ant tuščios vietos sukuriamas teksto laukelis.
- Užslenkant pelytę ant teksto laukelio krašto, atsiranda žymės (vidurys, kampas ir t.t.) ant kurio nuspaudus pelytės KK ir užslinkus pelytę ant kito teksto laukelio jie sujungiami.
- Užsilinkus pelytę ant pačios sujungimo linijos pateikiamas jos tipo pasirinkimas.

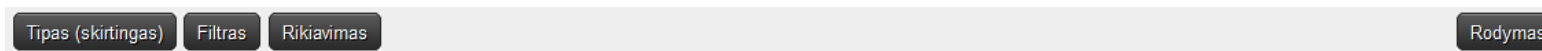
Taip galima suteikti ir galimybę „linijos pakeitimo principu“ pakeisti ir teksto laukelio tipą (grafiškas atvaizdavimas: ovalas, kvadratas, stačiakampis ir t.t.)

4.2.4. Meniu punktas Bendradarbiavimas

Tai nėra esminis projektų valdymo sistemos funkcionalumas. Tai plačiau būtų galima aptarti jeigu būtų akcentuojamas ne pačio projekto valdymas, o bendradarbiavimas projekto vykdymo metu.

4.2.5. Meniu punktai: resursai, ataskaitos, administravimas

Šių meniu punktų informacija dažniausiai pateikiama sąrašų principu. Tačiau, kad vartotojui būtų suteikta didesnė laisvė atvaizdavimo tipo pasirinkime, visiems sąrašams atvaizduoti naudosis tokį principą:



Paveikslukas 26 Pirmo lygio pasirinkimo meniu pavyzdys

Svarba – sukurti vieningą visiems sistemos sąrašams papildomų pasirinkimų meniu. Tai padėtų greičiau varotojui įsisavinti sistemą.

Tai bus antro lygio meniu. Žemiau pirmo lygio meniu. Mygtukų paaiškinimai ir atliekamos funkcijos:

- Tipas – pavadinimas ne galutinis, kiekvienam pagrindiniam meniu punktui skirtingas tačiau vieta ta pati. Pavyzdys.: užduočių valdyme sukeisti stulpelių šablonus (šablonų paaiškinimas aukščiau) arba atvaizdavimo tipus.
- Filtras – visuose atvaizdavimo tipuose pavadinimas vienodas, skirtingas tik pateikiamas pasirinkimas. Tarkime turime resursų (darbuotojų) sąrašą, tai galime pasirinkti filtruoti pagal vardą, pavardę ir pan.
- Rikiavimas – visuose atvaizdavimo tipuose pavadinimas vienodas, skirtingas tik pateikiamas pasirinkimas.

Rodymas – visuose atvaizdavimo tipuose pavadinimas vienodas, skirtingas tik pateikiamas pasirinkimas. Tarkime užduočių valdyme būtų galima pasirinkti rodyti užduotis kalendoriumi, darbo eigos sąrašu. Paprastuose sąrašuose kaip tarkime resursai pateiktu galimybių atvaizduoti sąrašu einančiu žemyn, stulpeliais, „talpiais“ stulpeliais.

Tokiu būdu visa likusi laisva vieta po meniu mygtuko „rikiavimas“, visada galėtų būti dinaminė ir pateikti vartotojui tokias galimybes kaip sukurti naują, redaguoti ar pašalinti pasirinktą objektą. Taip prieš tai išvardintų dinaminių pasirinkimų ir jų pasirodymų vartotojui vieta būtų aiški bei lauktina., t.y. kad kažką pasirinkus – pažymėjus šioje vietoje pamatysime papildomus galimus atlikti veiksmus.

4.3. Svarbiausi aspektai meniu punktų grafinių atvaizdavimų aprašymuose

Visi užduočių ir projektų (objektų) atvaizdavimai turėtų remtis naujųjų mobiliųjų telefonų, bei planšetinių įrenginių vartotojo sąsajų principais. Tinkamas naudotis sistema pelytės pagalba, bet orientuotas ir į ateitį labiau ir minėtus įrenginius, t.y. į liečiamą ekraną.

Slinkimas vaizdo į šoną turėtų būti „gudrus“, tai kad nuo kiek laiko palaikome nuspaudę ar prilietę taip priklausomai ir pasislinkimo dydis keičiasi bei greitis.

Visų objektų savybes kurias galima atlikti „perkėlimu“ ir atlikti perkėlimu. Pavyzdys užduoties priskyrimas kitam stulpeliui, tai tiesiog pertempiame iš vieno stulpelio į kitą ir viskas, tai būtų labai patogų veiksmą atliekant ranka, liečiamame ekrane. Kitas pažymėtinas punktas užduoties failų priskyrimas, šiuo metu programinės galimybės yra, kad galima failus įkelti į serverį tiesiog tempiant juos iš kompiuterio į naršyklės langą ant pažymėtos vietos (ne visose naršyklėse pavyzdžiui Internet Explorer).

Kitas pavyzdys: paieška viršuje. Norint pasinaudoti paieška reikia paspausti ikonėlę, tai yra veiksmas, tačiau persigalvojus ir norint pasiekti pagrindinį meniu, tereikia paslinkti pelytę truputį į kairę ir paspausti ant paieškos laukelyje esančio „iksiuko“ simbolio.

4.4. Vizualinių komponentų svarba

Pagrindinis tikslas – sukurti kitokį projektų valdymo įrankį. Atsisakyti sudėtingų diagramų, didžiąją jų teikiamo informatyvumo bei funkcionalumo dalį pakeičiant kitais duomenų vizualizavimo būdais, lengviau suprantamomis diagramomis. Jeigu vartotojui patinka tai kuo jis naudojasi, tai dažnai pasitaiko praktikoje tokių atvejų, kad naudodamasis ir paprastesnėmis priemonėmis/įrankiais pasiekia tokių pat gerų rezultatų kaip ir aukštesnės kokybės įrankiais (didesnio, sudėtingesnio funkcionalumo PĮ). Pavyzdys būtų: piešimas su „Microsoft Paint“ programa, kuri skirta tik minimaliam paveikslėlių redagavimui, palyginus su „Adobe Photoshop“, tačiau vaizdelių portale „youtube“ galima rasti labai didelį kiekį ne vieno vartotojo „Microsoft Paint“ entuziasto, kuris turi ganėtinai didelį portfelį įspūdingų grafikos darbų, atliktų būtent šia programa.

Pagrindinis tikslas Nr.2 – sukurti panašų, artimą funkcionalumą kitų analogiškų sistemų funkcionalumui, pateikiant vartotojui patrauklesnę formą. Patrauklesnių vizualinių

komponentų pagalba pasiekti tai, kad vartotojas būtų suinteresuotas naudotis būtent šiuo įrankiu.

5. EKSPERIMENTAS

5.1. Eksperimentas patvirtinantis, kad didelio funkcionalumo PĮ su prasta vartotojo sąsaja ir skurdžiais vizualiniais komponentais nėra tinkamas šio tipo įrankio realizavimo būdas

Pasirinkta „Achievo“ PĮ su labai plačiomis galimybėmis funkcionalumo atžvilgiu. PĮ buvo išbandoma UAB „Reklamos vizija“ programavimo padalinio „Deform“ darbuotojų. Ši PĮ bandymui pasiūlyta projektų vadovo, kadangi šis asmuo buvo girdėjęs gerų atsiliepimų apie PĮ funkcionalumą. Šios PĮ apžvalga atlikta literatūros analizėje. Naudota PĮ versija 1.3.0 šiuo metu randama internete <http://www.achievo.org/download/>.

Sistemos diegimas nesudėtingas pateiktas PĮ interneto puslapyje. Įdiegta sistema pasiekama adresu [25].

Sistemos funkcionalumo aiškinimuisi buvo skirtas vienos savaitės terminas ir vienas programuotojas (Tomas Kartašovas). Tikslas – suprasti kaip naudotis sistema, po to apmokyti naudotis ja kolegas, bei išsiaiškinti kokios sistemos plėtimo galimybės, t.y. kiek tai užims laiko.

Eksperimento rezultatai:

- Sistemos įdiegimas – nesudėtinga ir lengvai atliekama užduotis
- Sistemos įsisavinimas naudojimuisi – sudėtingas procesas užėmęs daug laiko ir pareikalavęs papildomų šaltinių (tokių kaip PĮ pagalbos forumas, google paieška) informacijos. Sistema labai paini naudojimuisi, norint atlikti konkretų veiksmą kartais tenka pasirinkti tris ar daugiau meniu punktų, kas dažnai priveda iki pasimetimo ką norima padaryti.

- Informacijos apie PĮ pasidalinimas –skirta 2 kartus atskiromis savaitės bėgyje dienomis po 4h laiko supažindinti su sistema kolegas. Projektų vadovas pasisavino perteiktą informaciją pirmo supažindinimo metu, tačiau bandymų metu atlikti konkrečius veiksmus reikėjo padėti surasti ką reikia atlikti, kad būtų atliktas norimas veiksmas. Kiti kolegos, kurie užduotims žymėti naudojos Google kalendoriumi, po supažindinimo su sistema, pareiškė nenorą ja naudotis, kadangi negalėjo jiems įprastu ir lengvu būtu greitai pažymėti atliktų užduočių.

Galiausiai nuspręsta, kad sistema yra per sudėtinga naudotis, nors suteikia daug didesnę funkcionalumą nei užduočių žymėjimas kalendoriuje. Taip pat nuspręsta, kad PĮ vartotojo sąsaja ir visos duomenų grafinis vizualizavimas yra per daug primityvus, vartotojai supažindinti su šia sistema nepanoro ja naudotis.

Kadangi įrankis netinkamas, tai bandymams išvaistytas laikas paskaičiuotas pagal formulę
asmuo * valandos * kartai:

- $1 * 8 * 5 = 40$ h
- $6 * 4 * 2 = 48$ h

Išvada – ieškoti ir išbandyti kitą šio tipo PĮ, bandyti mokamas šio tipo PĮ, kurti patiems.

5.2. Praktinėje dalyje įgyvendinto funkcionalumo tinkamumo patikrinimas

Atliekant praktinę dalį daug dėmesio skyriau užduočių tvarkymui, kadangi kaip pradinis sistemos kūrimo pavyzdys buvo Google kalendorius [27]. Manau šią dalį įgyvendinau tinkamai, kadangi atliekant analogų apžvalgą ir ieškant nemokamų vizualinių komponentų atradau DHTMLX kalendoriaus komponentą skirtą užduočių tvarkymui, kuris vizualiai patraukliai pateikia užduotis mums įprastame grafiškai atvaizduojame kalendoriuje.

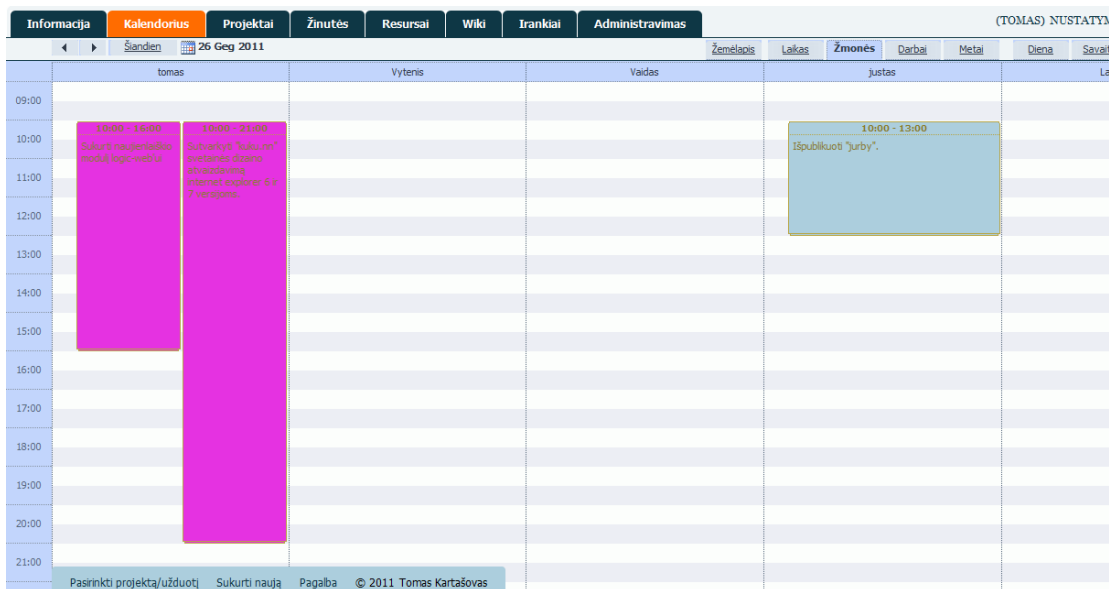
Komponentas randamas internete adresu <http://www.dhtmlx.com> [28]. Pasirinkus šį komponentą teko praplėsti jo galimybes iki užduočių apjungimo ir susiejimo su projektu. Tai buvo sudėtingas procesas, kadangi javascript programavimo kalba parašyta biblioteka sudėtinga ir didelės apimties. DHTML kalendoriaus komponentas sudėtingas dėl to, kad jis turi galimybes dalintis duomenimis su visais kitais šio PĮ gamintojo komponentais. Yra šio

komponento ir mokama versija, joje suteikiamos papildomos galimybės užduočių eksportavimui kitais duomenų formatais, bet apjungti užduotis į projektą ar panašių galimybių vistiek nėra.

Panaudoto komponento funkcionalumas dalinai didesnis už Google kalendoriaus, nes įtraukti tokie užduočių atvaizdavimo būdai kaip „Žmonės“, „Laikas“, „Žemėlapis“. Kiekvienas šių punktų turi savų pliusų. Kiekvieną iš jų trumpai galima apžvelgti:

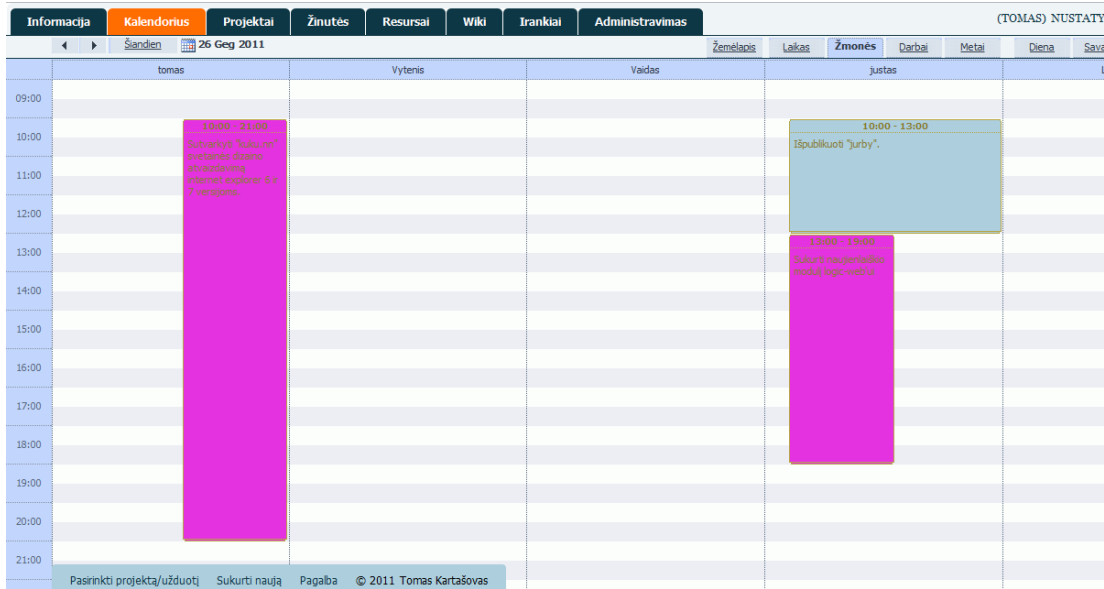
„Žmonės“ – šis atvaizdavimo būdas pateikė galimybę peržiūrėti dviejų ar daugiau asmenų, galinčių atlikti tą patį darbą, užimtumą ir paprastu KP klavišo nuspaudimu ant užduoties antraštės ir tempimu iki kito darbuotojo stulpelio. Žemiau pateikiamas tokios situacijos pavyzdys:

Darbuotojas Tomas turi atlikti dvi užduotis nuo 10h.



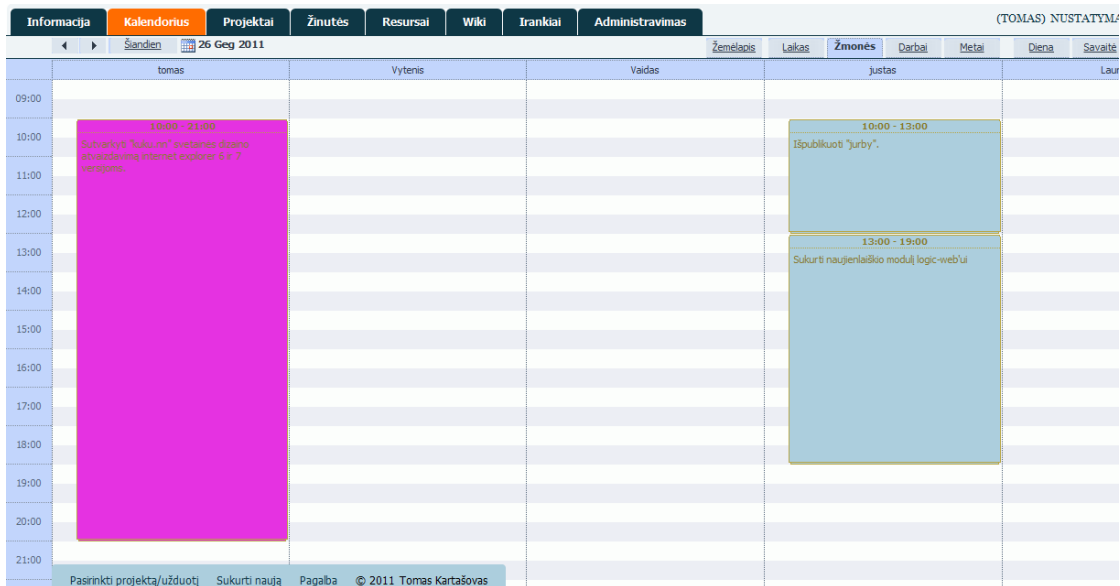
Paveikslukas 27 Užduočių atvaizdavimas „Žmonės“ būdu

Tačiau darbuotojas Justas turi tik vieną užduotį, kurią atliks per tris valandas ir neturės jokių kitų užduočių. Tai kodėl neperkėlus laiko atžvilgiu trumpesnės užduoties iš Tomo Justui?



Paveikslukas 28 Užduoties perskyrimas

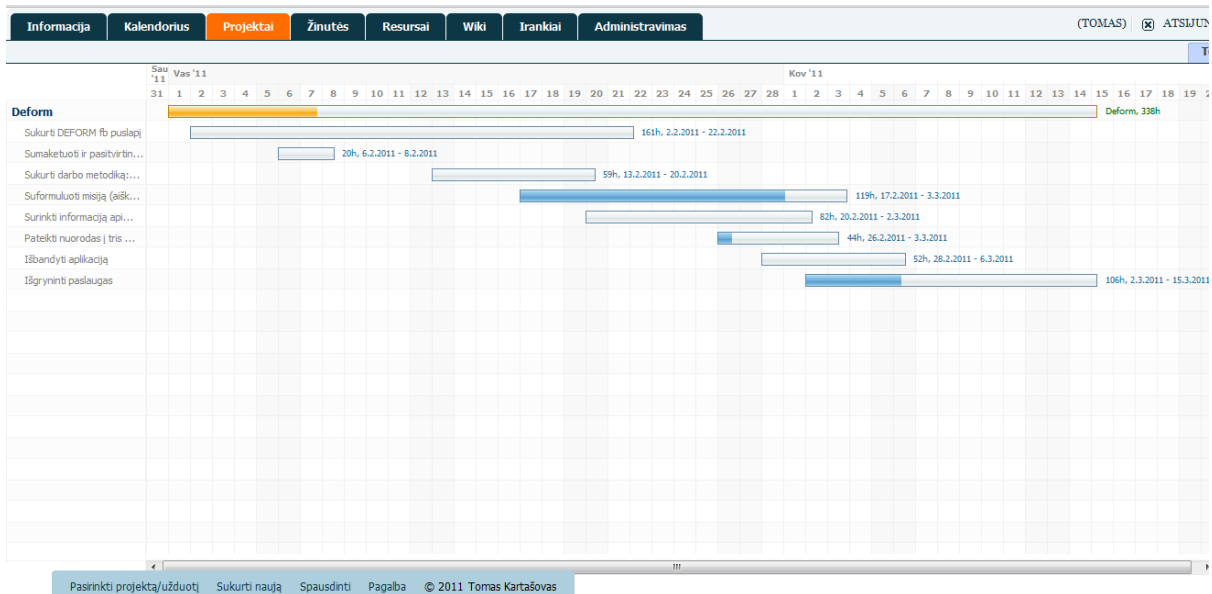
Atlikus veiksmą pasikeičia užduoties spalva iš priskirtos Tomui į priskirtą Justui:



Paveikslukas 29 Užduoties spalvos pasikeitimas

Taip abu gauname efektyvesnę 3 užduočių paskirstymą patogiu ir lengvai atliekamu būdu.

- „Laikas” – tai visiškas funkciniu požiūriu analogas „dienos” atvaizdavimo būdai. Tačiau įprasta kad laikas atvaizduojamas iš kairės į dešinę įvairiose diagramose ir pan (pavyzdys gantt tipo diagrama), tai šio atvaizdavimo būdo privalumas yra, kad pateikiamas darbuotojo dienos užimtumas nuosekliai iš kairės į dešinę pateiktomis 15min laiko intervaluose užduotimis (tiek kiek viena užduotis užima).
- „Žemėlapis” – kuriant užduotį galima priskirti vietovę. Tai atliekama taip pat kaip naudojantis įprastomis žemėlapių peržiūrų programomis <http://maps.google.com> ar <http://maps.lt>, kur paieškos laukelyje tereikia nurodyti vietovės pilna arba dalinį pavadinimą. Kadangi šis komponentas integruoja tą patį Google žemėlapi. Be to, šis priskyrimas atliekamas bet kuriame užduočių atvaizdavimo pasirinkime, nes visur naudojama ta pati užduoties sukūrimo forma. Taip pat vietovę galima priskirti ir projektui. Bendrai, tai nėra pagrindinė ar būtina funkcija projektų valdymo sistemoje, tačiau bandymo metu, vartotojai labai žavėjosi galimybe sukurti užduotį pavyzdžiui pavadinimu „Susitikimas” ir galimybe priskirti vietovę. Galvojant apie sistemos tolimesnį kūrimą, tai naudinga funkcija, kadangi sukūrus PĮ versiją skirtą mobiliesiems telefonams turintiems GPS navigaciją, atsirastų galimybė patikrinti užmirštą maršrutą ar tiesiog pasirinkti naudojamame įrenginyje, kad norima vykti į nurodytą vietovę. Pastebėtas papildomas šio atvaizdavimo būdo aspektas, kuris nėra taip aktualus programavimo inžinerijos sričiai, bet teisingai žymint susitikimus su klientais, arba tiksliau naujų klientų pirmą pažymėjimą kaip užduotį su nurodyta vietove iš karto, gauname savo vykdomos veiklos geografinio išplitimo vaizdą. Tokios



Paveikslukas 31 Projekto užduočių terminų modifikavimas

Galbūt projekto atvaizdavimas medžio tipo vaizdu, kaip sprendimas atvaizduoti visus projektus viename naršyklės lange ir buvo tinkamas, suteikiama galimybė net palyginti projektus, jeigu įgyvendintas projektų papildomas rikiavimas šiame medyje. Tačiau visa laiką buvo galima išskleisti tik vieną projektą, realizuoti galima, kad išsiskleistų iš karto ir keli, bet jeigu projektams priskirta bent kiek daugiau nei 10 užduočių tai jau nebebus matoma viename naršyklės lange (pagal monitoriaus rezoliuciją, šiuo atveju 1366x678). Tai nesuteikia tokios informacijos kaip užduočių persidengimas darbuotojams atliekant daugiau nei vieną projektą vienu metu. O tai yra labai svarbu, kadangi ne tik neefektyviai, bet nuostolingai gali būti paskirstytos užduotys, kai nematomas vaizdas kada ir kokias vienu metu vykdo vienas darbuotojas iš kelių skirtingų projektų. Darbuotojas nėra mašina kuri įvykdytų visas priskirtas skirtingas užduotis jeigu jos yra sudėtingos arba laike nėra greičiau įvykdomos negu numatyta. Tokiu būdu praleidžiamos ir neįvykdomos kai kurios užduotys, kas vėliau lemia projektų uždelsimą. Projekto ir užduočių vizualinių sprendimų idėjas aprašiau tiriamojoje dalyje. Manau kad atlikdamas praktinę dalį netinkamai sukūriau šio tipo duomenų atvaizdavimą, kurį dabar galėčiau padaryti tinkamesnį.

6. IŠVADOS

1. Atlikta atviro kodo projektų analizė rodo, kad kol kas daugelis jų pasižymi ganėtinai skurdžiomis vizualių komponentų savybėmis, kas sumažina jų galimybes efektyviai valdyti programinės įrangos projektų kūrimą.
2. Magistrinio darbo metu buvo sukurti ir ištirti vizualiniai komponentai su išplėstomis galimybėmis ir darantys projektavimo procesą efektyvesniu:
 - darbo eigos komponentas - galimybė valdyti užduotis pagal priskirtą būseną arba pagal projekto etapą. Pirmas pasirinkimas suteikia galimybę valdyti pavienes užduotis, antras vieno projekto užduotis, suskirstytas pagal etapus.
 - projektų išsidėstymo laike komponentas – galimybė peržiūrėti projektų išsidėstymą laike, su pažymėtais resursų persidengimais (jeigu tokie yra).
 - kalendorius – didelis pasirinkimas išsidėstymo laike peržiūros būdu, pasiskirstymo pagal darbuotojus peržiūra, paskirstymo pagal vietovę peržiūra.
3. Eksperimentinių tyrimų metu bandymo būdu patvirtintas vizualių komponentų reikalingumas tokio pobūdžio programinėje įrangoje.
4. Atliekant šį darbą realiai įgyvendintas projekto valdymas, įgyvendinant magistrinio darbo etapų planavimą. Keturiis semestrus buvo dokumentuojama įvykdytos užduotys ir informacija kaupiama magistro duomenų bazėje prieinamoje internetu <http://tomas.nzn.lt/magistras/>.
5. Naujai įgautos žinios šio darbo metu, ateityje turėtų padėti sukurti kokybiškesnę programinę įrangą ir efektyviau panaudoti atviro kodo projektus.
6. Įgyvendinus visus siūlomus vizualizavimo komponentus, programinė įranga galėtų būti labai naudinga kompiuteriuose su liečiamuoju monitoriumi, mobiliuosiuose įrenginiuose, turinčiuose didesnę liečiamąjį ekraną ir virtualioje lentoje, nes siūlomi vizualiniuose komponentuose tokios funkcijos lengvai modifikuojamos.

7. TERMINŲ ŽODYNAS

HTTP - HTTP (trumpinys nuo angl. HyperText Transfer Protocol) – pagrindinis metodas pasiekti informaciją pasauliniame tinkle (WWW). Pradinė protokolo paskirtis – pateikti standartinį būdą HTML puslapių skelbimui ir skaitymui.

Apache – labiausiai paplitęs nemokamas HTTP serveris internete.

Php (Hypertext Preprocessor) - hiperteksto scenarijų kalba.

Mysql – nemokama duomenų bazių valdymo sistema.

Html (Hypertext Markup Language) – hiperteksto žymėjimo kalba, kuri interneto tinkle plačiausiai naudojama aprašant tinklalapius. HTML nusako puslapio komponentų išdėstymą, šriftų stilius, grafinius elementus bei nuorodas į kitus interneto dokumentus.

Css (Cascading Style Sheets) – HTML elementų stilių aprašas.

HTML5 – atnaujintas HTML žymėjimo formatas.

DHTML (Dynamic HTML) – HTML galintis interaktyviai keisti išvaizdą. Paprastai tai atliekama kurios nors scenarijų kalbos (dažniausiai Javascript arba vbscript) pagalba.

AJAX - Asynchronous JavaScript and XML for web development.

Nuor. – nuoroda.

Akt. – aktyvūs.

Neakt. – neaktyvūs.

Admin. – administratorius(ių).

Modulis – rinkinys studentams skaitomų paskaitų, vedamų užsiėmimų, laboratorinių darbų.

Komponentas – papildomą panaudojimo galimybę turintis, vizualinę ar funkcinę reikšmę turintis sistemos elementas.

Darbuotojas – Žmogus atlikeantis vadovo paskirtas užduotis.

Vadovas – Už projektus atsakingas žmogus.

Užduotis – Projekto dalis turinti resursus, įvertinimą ir laiką jos atlikimui.

Resursas – laiko vienetas, darbuotojas, piniginis vienetas.

Vartotojas – sistemos vartotojas (administratorius, vadovas, darbuotojas).

PVS – projektų valdymo sistema

UML - Unifikuota modeliavimo kalba (Unified Modeling Language)

MVCH - Programų kūrimo technika

XML - Duomenų saugojimo formatas

MYSQL - Duomenų bazių valdymo sistema

JQUERY, UI - Javascript biblioteka vartotojo sąsajos kūrimui

DHTMLX - Javascript biblioteka kalendoriui ir sąrašo atvaizdavimui

PA - panaudojimo atvejai (Use Case)

Greitas menu - Meniu punktas matomas visą laiką nepriklausomai nuo kitų sistemos menu punktų. Suteikia galimybę redaguoti projektą ar užduotį arba projekto ar užduoties pasirinkimą perėjimui į meniu punktą „projektai“

8. LITERATŪROS SĄRAŠAS

- [1] Open-project aprašymas. Prieiga internetu: <http://www.project-open.com>
- [2] Linditex aprašymas. Prieiga internetu:
http://www.linditex.lt/projektu_valdymo_sistema.html
- [3] E-groupware sistemos svetainė. Prieiga internetu: <http://www.egroupware.org>
- [4] dotProject sistemos svetainė. Prieiga internetu: <http://www.dotproject.net/>
- [5] VisionProject sistemos svetainė. Prieiga internetu: <http://www.visionproject.se/>
- [6] ProjectPier sistemos aprašymas. Prieiga internet: <http://www.projectpier.org/>
- [7] Achievo sistemos aprašymas. Prieiga internetu: <http://www.achievo.org/>
- [8] Basecamphq sistemos aprašymas. Prieiga internetu: <http://basecamphq.com/>
- [9] Darbo užduoties atlikimo aprašymas. Prieiga internetu:
http://proin.ktu.lt/~virga/mag_atmintine/1sem/index.html
<http://dev.mysql.com/doc/refman/4.1/en/index.html>
- [10] PHP kalbos funkcijų dokumentacija: <http://www.php.net/manual/en/>
- [11] MySQL duomenų bazių valdymo įrankio dokumentacija internete:
<http://dev.mysql.com/doc/>
- [12] Javascript komponentų ir bibliotekų archyvas internete
<<http://www.dhtmlgoodies.com/>>
- [13] Nemokamos programinės įrangos paieškai internete
<<http://www.osalt.com/search?q=>>
- [14] Javascript knyga Object oriented javascript, pdf versi Stoyan Stefanov 2008m
- [15] Javascript programavimo šablonų paaiškinimas, Prieiga internetu:
<http://www.bennadel.com/blog/2047-Javascript-Patterns-By-Stoyan-Stefanov.htm>

- [16] AJAX – ajax technologijos naudojimas jQuery bibliotekos pagalba. Prieiga internetu: <http://api.jquery.com/jquery.ajax/>
- [17] JQuery bibliotekos dokumentacija ir pavyzdžiai. Prieiga internetu: http://docs.jquery.com/Main_Page
- [18] jQuery bibliotekos pavyzdžiai, pamokos. Prieiga internetu: <http://www.jquery4u.com/>
- [19] jQuery UI pavyzdžiai su paaiškinimais. Prieiga internetu: <http://jqueryui.com/demos/>
- [20] HTML5 aprašymas. Prieiga internetu: <http://en.wikipedia.org/wiki/HTML5>
- [21] RDBVS paaiškinimas. Prieiga internetu: http://lt.wikipedia.org/wiki/Reliacin%C4%97_duomen%C5%B3_baz%C4%97
- [22] GPL licenzijos aprašymas. Prieiga internetu: <http://www.gnu.org/licenses/gpl.html>
- [23] API paaiškinimas. Prieiga internetu: http://en.wikipedia.org/wiki/Application_programming_interface
- [24] ODBC paaiškinimas. Prieiga internetu: <http://lt.wikipedia.org/wiki/ODBC>
- [25] Bandyams skirtoms programinė įranga <http://www.achievo.org/>
- [26] Nemokamų įrankių pasirinkimo sąrašas. Prieiga internetu: <http://www.freecomponent.com/>
- [27] Sistemos prototipo kalendoriaus komponento kūrimo pavyzdys: <http://calendar.google.com>
- [24] Panaudotas komponentas kalendoriaus vaizdo realizavimui. Prieiga internetu: <http://dhtmlx.com/docs/products/dhtmlxScheduler/index.shtml>
- [26] Vizualių komponentų naudojimas. Prieiga internetu: <http://www.targetprocess.com/blog/page/2>

9. PRIEDAI

UAB REKLAMOS VIZIJA

Kauno Technologijos Universiteto

Informatikos fakultetui

ATLIKTO EKSPERIMENTO PAŽYMA

2011m. gegužės 30 d.

Kaunas

UAB „Reklamos vizija“ užsakymu programuotojas Tomas Kartašovas įdiegė atvirodo kodo projektų valdymo sistemą „Achievo“, bei atliko šios programinės įrangos funkcionalumo tyrimą.

Projektų vadovas Vaidas Bagdonas

UAB REKLAMOS VIZIJA

Kauno Technologijos Universiteto

Informatikos fakultetui

PAŽYMA

2011m. gegužės 30 d.

Kaunas

UAB „Reklamos vizija“ programuotojo Tomo Kartašovo praktinės magistrinio darbo dalies metu sukurta programinė įranga bus naudojama kuriant ir tobulinant vidiniams įmonės tikslams skirtą programinę įrangą.

Projektų vadovas Vaidas Bagdonas