

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Nauris Augulis

**DIDELIŲ DUOMENŲ KIEKIŲ SAUGOJIMAS IR
APDOROJIMAS NUTOLUSIŲ INTERNETO
CENTRŲ STEBĖJIMO IR ADMINISTRAVIMO
SISTEMOJE**

Magistro darbas

Darbo vadovas
doc. Kęstutis Motiejūnas

KAUNAS, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Nauris Augulis

**DIDELIŲ DUOMENŲ KIEKIŲ SAUGOJIMAS IR
APDOROJIMAS NUTOLUSIŲ INTERNETO
CENTRŲ STEBĖJIMO IR ADMINISTRAVIMO
SISTEMOJE**

Magistro darbas

Recenzentas

doc. Dalius Rubliauskas

2008 05 23

Vadovas

doc. Kęstutis Motiejūnas

2008 05 23

Atliko

IFM-2/2 gr. stud.

Nauris Augulis

2008 05 23

Kaunas, 2008

Turinys

| | |
|---|-----------|
| 1. ĮVADAS..... | 7 |
| 2. PROJEKTAVIMO BEI TECHNOLOGINIŲ SPRENDIMŲ ANALIZĖ..... | 7 |
| 2.1. REIKALAVIMŲ ANALIZĖ | 7 |
| 2.2. DAUGIASLUOKSNĖS SISTEMOS | 9 |
| 2.3. PATIKIMŲ SISTEMŲ PROJEKTAVIMAS | 9 |
| 2.4. INFORMACINIŲ SISTEMŲ SAUGUMAS | 10 |
| 2.5. OBJEKTINIS PROJEKTAVIMAS | 11 |
| 2.6. TIRIAMOS SISTEMOS ANALIZĖ | 12 |
| 2.7. PROGRAMŲ SISTEMOS KOKYBĖS VERTINIMO CHARAKTERISTIKOS | 20 |
| 2.8. IŠVADOS..... | 23 |
| 3. NUTOLUSIŲ INTERNETO CENTRŲ STEBĖJIMO IR ADMINISTRAVIMO SISTEMOS REALIZAVIMAS..... | 24 |
| 3.1. PANAUDOJIMO ATVEJAI | 24 |
| 3.2. FUNKCINIAI REIKALAVIMAI | 26 |
| 3.3. NEFUNKCINIAI REIKALAVIMAI | 27 |
| 3.4. PROGRAMINIO SPRENDIMO ARCHITEKTŪROS SPECIFIKACIJA | 29 |
| 3.5. IŠVADOS..... | 46 |
| 4. EKSPERIMENTINIS SISTEMOS KOKYBĖS TYRIMAS | 47 |
| 4.1. SISTEMOS FUNKCIONALUMAS | 47 |
| 4.2. SISTEMOS VARTOSENĄ | 48 |
| 4.3. SISTEMOS PATIKIMUMAS..... | 49 |
| 4.4. SISTEMOS EFEKTYVUMAS | 49 |
| 4.5. SISTEMOS PALAIKOMUMAS | 50 |
| 4.6. SISTEMOS PERKELIAMUMAS..... | 51 |
| 4.7. REZULTATAI | 51 |
| 4.8. TOLIMESNIO SISTEMOS TOBULINIMO IR PLĖTOJIMO GALIMYBĖS | 52 |
| 5. IŠVADOS | 53 |
| 6. LITERATŪRA..... | 54 |
| 7. TERMINŲ IR SANTRUMPŲ ŽODYNAS..... | 56 |
| PRIEDAS 1: ĮDIEGIMO DOKUMENTAS | 58 |
| PRIEDAS 1: APKLAUSA | 59 |

SUMMARY

Project describes specifying, designing and implementing tracking and administration system for distant internet centres. Analysis of design and technology solutions were researched during this project development. Some basic goals of system realization and potential solutions were formulated, which were presented. The architecture of the software developed is based on three layer design. This software was installed over thousand of computers and successfully used by people. Some research of system usage and user experience was done after system installation. This was done with the purpose of software quality analysis, that showed system quality is evaluated as an average, but its functionality was high.

PAVEIKSLŲ SĄRAŠAS

| | |
|---|----|
| 1 pav. Daugiasluoksnės sistemos modelis..... | 9 |
| 2 pav. Sistemos panaudojimo atvejų modelis | 25 |
| 3 pav. Sistemos architektūros modelis | 29 |
| 4 pav. Sistemos modulių diagrama..... | 31 |
| 5 pav. Prisijungimo sąsajos modulio klasė..... | 31 |
| 6 pav. Administravimo sąsajos modulio klasė | 32 |
| 7 pav. Statistikos peržiūrėjimo sąsajos modulio klasė | 32 |
| 8 pav. Prisijungimų valdiklio modulio klasės | 33 |
| 9 pav. Administravimo valdiklio modulio klasės..... | 34 |
| 10 pav. Statistikos valdiklio modulio klasės | 35 |
| 11 pav. Duomenų saugojimo valdiklio modulio klasės | 35 |
| 12 pav. Duomenų skaitymo valdiklio modulio klasės | 36 |
| 13 pav. Statistikos bei vartotojų informacijos saugojimo lentelės | 38 |
| 14 pav. Vartotojo prisijungimo prie kompiuterio sekų diagrama | 39 |
| 15 pav. Bendrosios statistikos peržiūros sekų diagrama | 39 |
| 16 pav. Administravimo sekų diagrama..... | 40 |
| 17 pav. Vartotojo prisijungimo prie kompiuterio veiklos diagrama | 41 |
| 18 pav. Bendrosios statistikos peržiūrėjimo veiklos diagrama | 42 |
| 19 pav. Administravimo veiklos diagrama..... | 43 |
| 20 pav. Sistemos būsenų diagrama..... | 44 |
| 21 pav. Sistemos išdėstymo vaizdas..... | 45 |

LENTELIŲ SĄRAŠAS

| | | |
|---|--|----|
| 1 | lentelė. Probleminiai sistemos realizavimo uždaviniai..... | 13 |
| 2 | lentelė. Duomenų bazės projektavimo užduotys | 15 |
| 3 | lentelė. Kuriamos ir esamų sistemų funkcijų palyginimai..... | 20 |

1. ĮVADAS

1.1 Dokumento paskirtis

Šis dokumentas yra skirtas pateikti magistro baigiamojo darbo aprašymo ir tyrimo dokumentaciją. Jis būtinas norint išsamiai ir vienareikšmiškai susipažinti su „Didelių duomenų kiekių saugojimo nutolusių interneto centrų sebėjimo ir administravimo sistemoje“ tyrimo duomenimis.

1.2 Santrauka

Lietuvoje sparčiai plečiantis informacinių technologijų naudojimui, kuriama vis daugiau informacinių technologijų projektų, kuriuos remia Europos Sąjunga ir kitos įvairios organizacijos. Taip pat stengiamasi pasiekti, kad informacinės technologijos būtų pasiekiamos kuo platesniam vartotojų ratui. Todėl steigiami interneto centrai kaimiškose vietovėse ir ne tik. Tačiau įsteigus tokius centrus ir norint juos tinkamai administruoti, reikia atitinkamos programinės įrangos. Deja lietuviškų produktų skirtų nutolusių interneto centrų stebėsenai ir administravimui nėra. Todėl sukūrus šią sistemą, palengvėjo interneto centrų, kuriuose ji įdiegta, administravimas.

2. PROJEKTAVIMO BEI TECHNOLOGINIŲ SPRENDIMŲ ANALIZĖ

2.1. Reikalavimų analizė

2.1.1. Reikalavimų užrašymas

Egzistuoja daug reikalavimų užrašymo formų. Jie gali būti dokumentuojami įvairiomis formomis, pavyzdžiui, natūralia kalba, vartojimo atvejais(Use cases), procesų specifikacijomis, ar į tam tikros formos dokumentus. Reikalavimai paprastai išgaunami interviu metu su užsakovu. Viena labiausiai paplitusių reikalavimų užrašymo formų yra fiksavimas į tam tikros formos dokumentus pvz: *Volere* šablonas. [7]

Volere šablonas naudojamas pradiniam sistemos kūrimo etape ir yra kaip pagrindas užregistruoti vartotojų reikalavimus. Reikalavimai skirstomi į skyrius pagal jų tipus, su detaliais aprašymais. Juose kaupiami reikalavimai, kuriuos pateikia vartotojai per interviu arba kurie nustatomi analizuojant sistemos objekto veiklą. Tai atviras šablonas, kurį galima pritaikyti konkrečiam atvejui. Skyrius, kurie netinka nagrinėjamam objektui, galima išmesti arba sukurti naują skyrių, kuris leidžia specifikuoti specifines dalykinės srities charakteristikas. Šablono

pagrindu galima aprašyti pagrindinius funkcinis bei nefunkcinis reikalavimus, sudarančius bendrą vartotojo reikalavimų modelį .

2.1.2. Programinės įrangos reikalavimų specifikacija

PĮ reikalavimų specifikacija yra pilnas sistemos, kuri kuriama, elgesio aprašymas. Šioje specifikacijoje yra įtraukti panaudos atvejai(Use cases), kurie aprašo vartotojų veiksmus su sistema. Panaudos atvejai taip pat laikomi funkciniais reikalavimais. Reikalavimų specifikacijoje taip pat specifikuojami ir nefunkciniai reikalavimai. Nefunkciniai reikalavimai aprašo sistemos modelio ar įgyvendinimo apribojimus, pavyzdžiu, sistemos greitaveikos, kokybės standartų, ar dizaino apribojimų.

2.1.3. Reikalavimų analizavimo metodai

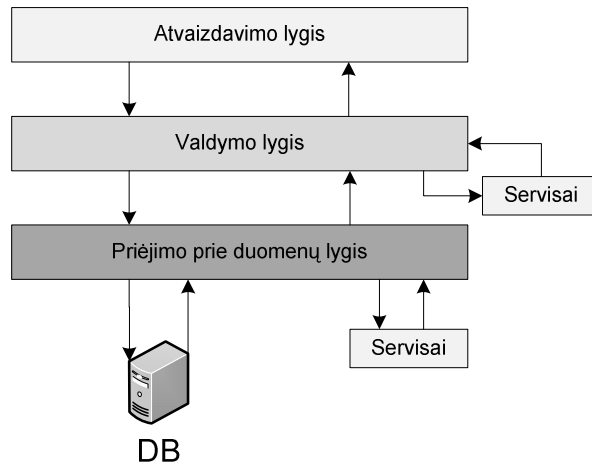
Plačiausiai žinomas ir naudojamas analizavimo metodas yra reikalavimų transformavimas į vieningos modeliavimo kalbos (UML)[17] diagramų panaudojimo atvejus. UML leidžia atlikti orientuotą į objektą sistemos analizę ir projektavimą naudojantis viena nuoseklia kalba programų sistemos rezultatams specifikuoti, vaizduoti, ir dokumentuoti .

Kitas analizavimo metodas yra sistemos specifikuojamas formaliais metodais. Pagrindinė šio metodo problema, kad ne visus reikalavimus galima formaliai specifikuoti. Taip pat daugelis formalių specifikuojamų kalbų yra gana sudėtingos, reikalaujančios daug laiko jų suvokimui ir praktiniam pritaikymui. Toks formalizmo metodas daugiau taikytinas kritinėms sistemoms, reikalaujančioms specifikuojamos teisingumo ir patikimumo įrodymo. Beje, labai dažnai formalizmu apsiribojama tik pateikiant formalias specifikuojamas, nes tolesniam plėtojimui trūksta paskatinimo .

Naudojant UML[17], vartotojo funkcinis reikalavimus betarpiškai tenkinantys rezultatai – formos, ataskaitos, meniu, kurie dažniausiai išreiškiami per sąsajos elementus, nėra pilnai specifikuojami, o specifikuojamos forma neadekvati vartotojo turimam jų įvaizdžiui.

2.2. Daugiasluoksnės sistemos

Vienas iš efektyviausių, kuriamos PĮ sistemos, sudėtingumo mažinimo būdų yra programinio kodo skaidymas į sluoksnius (lygius). Dažniausias naudojamas trijų sluoksnių principas, kai atskiriami vartotojo sąsajos, programos logikos ir duomenų priėmimo sluoksniai (1 pav.).



1 pav. Daugiasluoksnės sistemos modelis

Lygiai, kurie daugiau ar mažiau yra tik principas koku būdu organizuojamas programinis kodas į stipriai susijusias panašių užduočių zonas yra naudingi sistemos palaikomumui. Esminis faktorius yra tai, kad pakeitimų darymas viename lygyje sąlygoja minimalius pakeitimus kituose lygiuose, kas programinį kodą padaro kur kas labiau palaikomą. Lygiai taip pat suteikia tam tikrą abstrakcijos lygį nuo jų sudarančių realizacijos ypatumų. Kas įgalina pakeisti reliacinę duomenų bazę į XML be jokių pakeitimų kituose lygiuose.[11]

Pagrindinė problema taikant sluoksnius, kaip ir su visomis kitomis strategijomis, proteguojančiom ryšių tarp sistemos komponentų minimizavimą, yra papildomas nereikalingas sudėtingumas. Jūs rizikuojate perprojektuoti savo sistemą pritaikydami ją įvairiausiems pasikeitimams, kurie greičiausiai niekada neįvyks ir nėra tikrų sistemai keliamų reikalavimų dalis.

2.3. Patikimų sistemų projektavimas

Patikimų sistemų projektavimo principai grindžiami tuo, kad neįmanoma sukurti tobulo produkto, kuris neturėtų jokių defektų, tačiau galima sukurti pakankamai patikimas sistemas[13]. Panagrinėkim parašitistus. Nei vienas jų nepirks ir nepasitikės parašitais iš įmonės sakančios, kad jos parašutai visada išsiskleis. Parašitistai žino, kad nesvarbu kiek kartų viską patikrinsi, kokių procedūrų

laikysies vis tiek yra tikimybė, kad tavo pagrindinis parašutas neišsiskleis. Šiems atvejams kaip tik ir skirtas atsarginis parašutas. Tokį nelaimių vengimo būda galima vadinti pertekliško principu.

Kuriant PĮ irgi reikia laikytis panašių principų. Neįmanoma pagaminti visiškai patikimo komponento, tačiau kai vienas komponentas tikrina ne tik save bet ir kitų sistemos komponentų darbą galima surinkti vieną pakankamai patikimą sistemą. Pagrindiniai PĮ patikimumo didinimo principai:

- a) defektų mažinimas;
- b) sprendimo paprastumas – padeda išvengti klaidų;
- c) apsaugų pertekliškumas – leidžia numatyti, kad įvykus gedimui vienoje sistemos dalyje bus kita perteklinė sistema, kuri galės atlikti tas pačias funkcijas;
- d) klaidų aptikimas ir apdorojimas.

Šių punktų reikia laikytis ir įgyvendinant šio darbo PĮ, nes bet koks netikslus veikimas arba prastovos sąlygos tam tikros duomenų aibės praradimą.

2.4. Informacinių sistemų saugumas

Informacinių sistemų saugumas tapo viena iš aktualiausių problemų greitai didėjant kompiuterizacijos lygiui ir kompiuterinių tinklų integracijai. Įvairiuose internetiniuose šaltiniuose ir techninėje literatūroje galima rasti begales patarimų, metodikų ir produktų, kurie nagrinėja programinės įrangos saugumo problemas ir metodus kaip jas galima spręsti.

Nepaisant specifinių kiekvienos sistemos saugumo problemų yra išskiriamos bendrinės PĮ saugumo problemos. Šias problemas turi įvertinti bet kuris PĮ produktas, kuriam aktuali saugumo problema. Bendrinės PĮ saugumo problemos:[15]

- a) buferio persipildymas;
- b) teksto formatavimo problemos;
- c) sql kodo įterpimas;
- d) komandų įterpimas;
- e) išskylančių klaidų neteisingas apdorojimas;
- f) tarp puslapiniai skriptai;
- g) tinklo informacinių srautų nesaugojimas;
- h) „stebuklingų“ url naudojimas ir paslėptos formos;
- i) neteisingas ssl protokolo naudojimas;
- j) lengvų slaptažodžių naudojimas;

- k) nesugebėjimas išsaugoti ir apsaugoti informacijos;
- l) informacijos nutekėjimas;
- m) neleistinas failų priėjimas;
- n) masyvų ribų tikrinimas;
- o) pasitikėjimas tinklo adresų informacija;
- p) signalų lenktyniavimo situacijos;
- q) neautentifikuotas apsikeitimas raktais;
- r) nenaudojimas kriptografiškai stiprių atsitiktinių skaičių;
- s) sudėtingas sistemos panaudojamumas.

Kai kurias iš šių problemų, padeda išspręsti valdomų programavimo aplinkų pasirinkimas pvz.: Java arba .NET [12]. Tokiose sistemose atliekamas griežta kintamųjų tipų kontrolė, stebimi masyvų ir įvairių operacijų veiksmi, automatizuotai valdoma atmintis, taip minimizuojamos atminties nutekėjimo ir buferio perpildymo problemos .

Visas kitas čia paminėtas problemas būtina įvertinti sistemos projektavimo, kūrimo ir diegimo metu. Bet kuriame iš šių etapų padaryta klaida, vėliau gali kainuoti labai brangiai.

2.5. Objektinis projektavimas

Dauguma šiuolaikinių programinių sistemų kuriamos taikant objektinio projektavimo principus. Objektinis projektavimas - tai modelis, kuris leidžia į programos objektus žiūrėti kaip į juodas dėžes, kurios tik siunčia ir gauna pranešimus.

Objektinio projektavimo principai:

- a) *Inkapsuliacija* - Objektas – tai juoda dėžė, kuri tik siunčia ir gauna pranešimus. Visi objektai gali dirbti nepriklausomai ir žinoti apie kitus objektus reikia tik tiek, kad būtų galima pasinaudoti jų teikiamomis sąsajomis;
- b) *Paveldimumas* - Baziniai objektai realizuoja savo teikiamas funkcijas, kurias toliau galima lanksčiai ir lengvai išplėsti paveldimuose objektuose. Pakeitus bazinį objektą (pavyzdžiui, ištaisyus klaidą), pakeitimai persiduoda ir visiems to objekto palikuonims;
- c) *Polimorfizmas* - Tai savybė, kuri apibūdina kodo formos keitimą. Vienas aprašas gali būti naudojamas su skirtingais duomenų rinkiniais.

Tinkamas objektinių principų taikymas leidžia išskaidyti sistema į atskirus mažai tarpusavyje priklausomus modulius. Kūrių kiekvienas vykdo tam tikras funkcijas. Toks sistemos dalių atskyrimas ir

struktūrizavimas palengvina sistemos palaikymą ir vėlesnio tobulinimo darbus, kas tiesiogiai atsiliepia kuriamos sistemos projekto kaštams.

Taikant objektinį projektavimą, sistemos architektūros kūrimo metu, reikia nepamiršti, kad yra visas sąrašas sugalvotų ir plačiai dokumentuotų projektavimo šablonų. Kurie nagrinėja visas dažniausiai pasitaikančias sistemų kūrimo problemas ir leidžia net pačias sunkiausias sumažinti iki valdomų ir prognozuojamų sistemos komponentų. Tinkamas objektinio projektavimo šablonų pritaikymas gali sutaupyti nemažai laiko ir išspręsti problemas apie kurias iš pradžių gal net neįtariame [16].

2.6. Tiriamos sistemos analizė

2.6.1. Taikymo sritis ir pradiniai reikalavimai

Taikymo sritis – nutolusių interneto centrų, kitaip interneto kavinių, stebėjimas ir administravimas.

Šiame skyriuje, pateikiami reikalavimai gauti iš užsakovo, dar prieš pradėdant reikalavimų rinkimą vadovaujantis *Volere* šablonais, t.y. pradiniai vartotojo reikalavimai – norai kuriamai sistemai.

Šio projekto užsakovas yra viena konkreti įmonė KTU Distancinio mokymo centras. Jos tikslas yra kuruoti Viešųjų interneto prieigos taškų problemas susijusias su IT. Sistema buvo kuriama būtent šios įmonės užsakymu, kad sumažinti interneto centrų administravimo kaštus, pateikiant centralizuotas ataskaitas, atsakingiems asmenims.

Užsakovo pateikti pradiniai funkciniai reikalavimai:

- 1) Minimizuoti rankinį darbą skirtą interneto centrų ataskaitų pildymui;
- 2) Vartotojo autentifikavimas ir prijungimas prie sistemos
- 3) Vartotojo autentifikavimas sistemos pagalba darbui kompiuteriu
- 4) Vartotojų prisijungimo ir kita informacija integruota su vipt.lt portalo vartotojų informacija, t.y. vartotojai turintys prisijungimo duomenis prie vipt.lt portalo, gali naudotis ir nutolusių interneto centrų sistema.
- 5) Sistema leidžia lankytojui prisijungti svečio teisėmis, t.y. be autentifikacijos
- 6) Statistinių duomenų kaupimas:
 - a. Kompiuterio darbo laiko kaupimas
 - b. Vartotojo praleisto laiko prie kompiuterio kaupimas.
 - c. Vartotojo naudotų programų sąrašų kaupimas

- d. Vartotojo naudotų programų naudojimo trukmė
- e. Vartotojo aplankytų tinklapių kiekis
- 7) Ataskaitų generavimas iš sukauptų statistinių duomenų ir pateikimas atitinkamiems asmenims per WEB prieigą.
- 8) Ataskaitų generavimas apie atskirus interneto centrus
- 9) Ataskaitų generavimas apie atskirus kompiuterius
- 10) Ataskaitų generavimas apie atskirus vartotojus
- 11) Išsamus vartotojo vadovas.

2.6.2. Probleminiai uždaviniai

Iš ankstesniame skyrelyje aprašytų pradinių reikalavimų, žemiau esančioje lentelėje (1 lentelė) pateikti pagrindiniai sistemos realizavimo probleminiai uždaviniai. Siekiant sėkmingai išspręsti šiuos uždavinius, analizės ir projektavimo metu turės būti atsakyta į visus lentelėje pateiktus klausimus.

1 lentelė. Probleminiai sistemos realizavimo uždaviniai

| Probleminiai uždaviniai | Pagrindiniai klausimai |
|---|--|
| <i>Vartotojo autentifikavimas sistemos pagalba darbui kompiuteriu</i> | <ul style="list-style-type: none"> • Ką daryti jei autentifikavimo metu nėra ryšio su serveriu? • Kokią pasirinkti vartotojo duomenų apsaugą |
| <i>Statistinių duomenų kaupimas ir atvaizdavimas</i> | <ul style="list-style-type: none"> • Ką daryti jei statistikos siuntimo metu į serverį nutrūksta arba nėra interneto ryšio? • Ką daryti kai bus sukaupti dideli statistinių duomenų kiekiai? |

2.6.3. Probleminių uždavinių sprendimai

Šiame skyrelyje pateikiami 2.6.2 skyrelyje iškeltų probleminių sistemos realizavimo uždavinių sprendimų aprašymai.

Vartotojo autentifikavimas sistemos pagalba darbui kompiuteriu

Detaliau išanalizavus vartotojo prisijungimo prie kompiuterio mechanizmą, buvo iškeltos tokios pagrindinės problemos

1. Kaip užtikrinti, kad vartotojo įvesti vartotojo vardas ir slaptažodis, tinklu iki serverio keliautų apsaugoti.

2. Kaip užtikrinti, kad vartotojas prisijungtų darbui kompiuteriu, net jei nėra ryšio su autentifikavimo serveriu.

Šių dviejų problemų galimas sprendimas:

1. Kadangi sistema su serveriu duomenimis keičiasi HTTP protokolo POST metodu [1], duomenų apsaugai galima naudoti SSL/TLS šifravimo algoritmus[3]. Tačiau šiam duomenų apsikaitimo būdui reikalingi ne tik kliento dalies pakeitimai, bet ir serverio, todėl nuspręsta šio būdo nenaudoti.

Pasirinktas paprastesnis, tačiau ir mažiau saugus vartotojo duomenų apsaugos būdas. Vartotojo autentifikavimosi metu į serverį siunčiamas vartotojo slaptažodis ir prisijungimo vardas užkoduojami MD5 algoritmu.[4] Šio algoritmo privalumas yra tas, kad bet kokio ilgio simbolių eilutė užkoduojama į 32 šešioliktainių skaičių eilutę.

Naudojant taip užkoduotus vartotojo duomenis, užtikrinamas tam tikras duomenų saugumo laipsnis. Net jei kieno nors būtų gauti šie užkoduoti duomenys, jų dekoduoti praktiškai neįmanoma.

2. Siekiant išspręsti problemą su vartotojo autentifikacija, kai kompiuteris neturi ryšio su serveriu, galimi keli sprendimo būdai:
 - a. Saugoti visų vartotojų duomenų lentelę lokaliai, t.y. kiekviename interneto centro kompiuteryje, ją atnaujinant tam tikru laiko intervalu. Tokio metodo privalumas yra tas, tai užtikrintų vartotojo autentifikavimąsi, net jei kompiuteris neturi ryšio su serveriu. Tačiau iš tokio būdo atsiranda daug kitų spręstinų problemų.
 - i. Pavyzdžiui vartotojų lentelės lokaliai atnaujinamos kas valandą, o vartotojas prieš pusvalandį pasikeitė slaptažodį ir dingo interneto ryšys. Vartotojas norėdamas prisijungti darbui prie kompiuterio turėtų vesti senąjį slaptažodį.
 - ii. Vartotojų lentelėje yra apie 50 000 įrašų. Reikia paminėti, kad kuriant sistemą planuojama ją diegti mažiausiai į 1100 kompiuterių. Tad kiekvieną kartą atnaujinant duomenis lokaliai, būtų stipriai apkrautas serverio ir tinklo darbas.
 - iii. Saugumo problemos. Lokaliai saugomus vartotojo duomenis gali rasti, bet kas ir juos nuskaityti.

Dėl šių priežasčių vartotojų duomenų saugojimo kiekviename kompiuteryje buvo atsisakyta.

- b. Šios problemos sprendimui buvo pasirinktas paprastas būdas. Jei nėra interneto ryšio, vartotojui siūloma jungtis svečio teisėmis. Šiuo atveju vartotojas gauna pilną kompiuterio darbo funkcionalumą, nors prarandami kai kurie statistiniai duomenys.

Statistinių duomenų kaupimas ir atvaizdavimas

Išanalizavus statistinių duomenų kaupimą ir atvaizdavimą, buvo iškeltos tokios problemos:

1. Ką daryti jei statistikos siuntimo metu į serverį nutrūksta arba nėra interneto ryšio?
2. Ką daryti kai bus sukaupti dideli statistinių duomenų kiekiai?

Šių problemų galimi sprendimo būdai:

1. Statistiniai, kaip ir autentifikacijos, duomenys į serverį siunčiami HTTP protokolo POST metodu. Todėl sukaupti statistiniai duomenys vartotojo atsijungimo nuo kompiuterio metu išsaugomi faile kompiuteryje. Prisijungus kitam vartotojui šie duomenys siunčiami į serverį, jei to nepavyksta padaryti duomenys vėl įrašomi į failą kompiuteryje. Atsijungus kitam vartotojui failas papildomas naujais duomenimis, ir taip vyksta tol kol duomenis pavyksta išsiųsti į serverį.
2. Pagrindinė problema, kuri sprendžiama, yra didelių statistinių duomenų kiekių kaupimas, saugojimas ir panaudojimas. Šios problemos sprendimui reikalingas duomenų bazės optimizavimas. Pagrindinė optimizavimo užduotys yra duomenų bazės reliacinio modelio optimalus sudarymas, užklausų optimizavimas, duomenų analizė ir t.t. Šios užduotys buvo suskirstytos į tam tikrus lygius:

2 lentelė. Duomenų bazės projektavimo užduotys

| Nr. | Lygis | Užduotis |
|-----|-----------------------|--|
| 1. | Vienkartinės užduotys | Siekiamų apdoroti ir saugoti duomenų analizė Duomenų bazės reliacinio modelio sudarymas |

| | | |
|----|-----------------------|--|
| 2. | Daugkartinės užduotys | Užklausų optimizavimas Duomenų bazės lentelių skaldymas Nereikalingų duomenų naikinimas Duomenų transformavimas Duomenų bazės ir jos apkrovimo analizė |
|----|-----------------------|--|

Svarbiausios yra pirmojo lygio užduotys. Jas atlikus, t.y. suprojektavus ir sukūrus duomenų bazę, vėlesni jos pakeitimai gali būti labai sudėtingi. Tuo labiau, kad tas pats serveris bei DBVS aptarnauja ir daugiau projektų. Analizuojant ir atliekant pirmojo lygio užduotis, galvojama ir apie antrajame lygyje esančias užduotis, kad vėliau būtų įmanomas jų atlikimas.

Siekiamų apdoroti ir saugoti duomenų analizė reikalinga tam, kad būtų teisingai sudarytas duomenų bazės reliacinis modelis. Taip nustatoma kokius duomenis gaus serveris bei kuriuos iš jų yra tikslinga saugoti. Ši užduotis yra standartinė duomenų bazių kūrime, bet taip pat ir labai svarbi.

Duomenų bazės reliacinis modelis buvo sudaromas remiantis duomenų analize. Tai, taip pat vienas svarbiausių žingsnių, norint sukurti efektyvią duomenų bazę.

Užklausų optimizavimas vykdomas tuomet, kai pastebimas didelis duomenų bazės apkrovimas, nors siekiamų atvaizduoti duomenų kiekiai yra nedideli. Ši užduotis atliekama periodiškai stebint DBVS sunaudojamą procesoriaus laiką.

Jei užklausų optimizavimas nepadeda sumažinti DBVS naudojamą procesoriaus laiką, galimas duomenų bazės lentelių skaldymas, t.y. lentelės kuriose yra daug įrašų, skaldyti su vieno mėnesio duomenimis. Taip nuspręsta daryti, nes buvo pastebėta, kad norint iš nesuskaldytos lentelės gauti senus duomenis, šis procesas užtrukdavo ilgai ir sunaudodavo daug procesoriaus laiko. Atliekant tą patį veiksmą su naujais duomenimis, ši problema neiškildavo.

Duomenų transformavimas atliekamas senesnius duomenis transformuojant galutinį rezultatą turinčius interneto puslapius, ar duomenų bazės lenteles iš kurių duomenys gaunami greitai.

2.6.4. Realizavimo technologijos pasirinkimas

Apžvelgę sistemai keliamus reikalavimus galime iš anksto numatyti, kad jos kūrimui reikės technologijų kurios leistų realizuoti WEB statistikos generavimo ir atvaizdavimo sistemą, stebėjimo ir statistikos kaupimo bei atnaujinimų ir apribojimų programą diegiamą į kompiuterį.

Šiuo metu naudojamos trys pagrindinės WEB platformos: *PHP*, *.NET* ir *Java*. Yra ir kitų variantų, bet jie kur kas mažiau išplėtoti, neturi geros dokumentacijos, bei nėra reikalingos patirties, todėl jų plačiau nenagrinėjant buvo atmesti.

PHP tai greitai išmokstama ir pakankamai galinga programavimo kalba WEB puslapių kūrimui. Yra sukurta daug šablonų(angl. framework), padedančių programuotojui pasiekti rezultatą per daug trumpesnį laiką bei atskiriančių logiką nuo duomenų bei atvaizdavimo lygių. Tačiau, dėl nepakankamai išvystytų IDE, su ja yra sunkiau kurti.

Galiausiai statistikos generavimo ir atvaizdavimo posistemio realizavimui buvo nuspręsta pasirinkti *PHP* platformą. Apsisprendimą nulėmė kelios priežastys. Pirmiausia tai, kad sistemai pateiktuose reikalavimuose nurodyta naudoti „*MySQL*“ DBVS. Antra kūrėjai turi didesnę patirtį dirbant su *PHP* platforma nei su *Java* ar *.NET*.

Kliento dalyje nuspręsta naudoti Visual C++ programavimo kalbą.[6] Viena iš priežasčių kodėl pasirinkta ši kalba yra ta, kad jos kodas kompiliuojamas į „native kodą“, t.y. nereikia jokių papildomų virtualių mašinų ar karkasų (angl. Framework) kaip *.NET* programos paleidimui. Taip supaprastinamas programų sistemos diegimas į kliento kompiuterį.

2.6.5. Egzistuojančių sistemų lyginamoji analizė

Yra keletas komercinių realizacijų, kurios įgyvendina nutolusių interneto centrų stebėsenos ir administravimo sistemos funkcijas. Toliau nagrinėsime aplinkas, kuriose yra integruotos (o ne pavienės) pagrindinės interneto centrų administravimo funkcijos: „Smartlaunch“, „Internet Caffè Software“, „CyberCafePro“. Visos šios aplinkos turėjo savo specifinį plėtros kelią, tačiau apibendrinant galima teigti, kad visos šios sistemos veikia panašiai. Vienos jų turi funkcijas skirtas klientų leidžiančias klientams ne tik dirbti kompiuteriu, bet ir užsisakyti maisto, kitos skirtos interneto centrams, kuriuose daugiausia žaidžiami žaidimai.

SmartLaunch systems „Smartlaunch“ sprendimas

SmartLaunch interneto centrų stebėsenos ir administravimo sistema skirta žaidimų ir darbo centrams. Šioje sistemoje įgyvendintas amžiaus cenzo palaikymas pagal JAV ir Europos standartus.

Veikiant šiai programai kliento kompiuteryje standartinė Windows užduočių juosta nepanaikinama, ir pridedama papildoma SmartLaunch užduočių juosta su įvairių paslaugų pasirinkimu ir išnaudotu balansu. Šioje sistemoje taip pat įgyvendintos įvairios saugumo priemonės, kaip užblokuojamas priėjimas prie įvairių diskų skirsnių. Pagrindinės šios sistemos funkcijos:

- Kompiuterių valdymas
- Ekranu nuotraukos(screenshot) peržiūrėjimas
- Vartotojų skirstymas į kategorijas
- Įvairūs apsaugos lygiai:
- Draudimas naudoti valdymo skydą(Control Panel)
- Draudimas naudoti Task Manager
- Draudimas naudoti nustatytais diskų skirsniais
- Ataskaitų siuntimas paštu
- Finansinių ataskaitų generavimas
- Darbuotojų pardavimo ataskaitos
- Įvairių kitų ataskaitų generavimas

Šios sistemos privalumai:

- Centralizuotas kompiuterių valdymas
- Kiekvieno vartotojo duomenys išsaugojami atskirai ir jais galima naudotis bet kuriame kompiuteryje
- Turi labai daug kitų galimybių, tačiau jos nėra aktualios.

Sistemos trūkumai:

- Pritaikytas žaidimų viešiesiems interneto prieigos taškams
- Negalima organizuoti centralizuoto darbo daugiau nei viename viešųjų interneto prieigos taške
- Dideli reikalavimai techninei įrangai

Antamedia „Internet Caffe Software“.

Ši sistema panaši kaip ir aprašyta anksčiau. Ši sistema turi daug įvairių funkcijų ir naudoja kliento – serverio principą. Šios sistemos pagrindinės funkcijos:

- Pilna kompiuterių kontrolė
- Aktyvumo žurnalizavimas
- Spausdinimo stebėjimas
- Ataskaitų generavimas
- Įvairūs apmokėjimo būdai su nuolaidų sistemomis

Šios sistemos pagrindiniai privalumai:

- Daugiau nei vieno interneto preigos taško palaikymas centralizuotai
- Galima naudotis daugeliu kalbų
- Interneto srauto stebėjimas

Sistemos trūkumai:

- Apkrauta vartotojo sąsaja
- Sudėtingas valdymas

CyberCafePro

Sistema realizuota naudojant kliento – serverio technologiją. Ši sistema pakeičia pagrindinį Windows sistemos apvalkalą(shell) - Explorer. Ši sistema taip pat turi daug galimybių, pagrindinės jų:

- Kietojo disko skirsnių pasiekiamumo blokavimas
- Internet Explorer funkcijų blokavimas
- Failų siuntimos draudimas
- Programų paleidimo draudimas
- Specialiųjų klavišų kombinacijų naudojimo draudimas
- Pardavimų ataskaitų generavimas
- Ataskaitų generavimo periodiškumo nustatymas

- Automatinis ataskaitų siuntimas paštu
- Realizuotas bendravimas tarp kliento ir interneto centro administratoriaus
- Kompiuterių valdymo galimybės

3 lentelė. Kuriamos ir esamų sistemų funkcijų palyginimai

| | Nutolusių interneto centrų stebėsenos ir administravimo sistema | SmartLaunch | Internet Caffe Software | CyberCafePro |
|--|---|-------------|-------------------------|--------------|
| Ataskaitų apie centrų ir juose esančių kompiuterių darbą generavimas | + | + | + | + |
| Apribojimų nustatymas kiekvienam vartotojui | + | - | - | + |
| Paslaugų teikimo automatinis apmokestinimas | +/- | + | + | + |
| Vartotojų darbas prie skirtingų kompiuterių | + | + | + | + |
| Vartotojų aktyvumo stebėjimas ir žurnalizavimas | + | + | + | + |
| Daugelio centrų palaikymas | + | - | + | + |
| Vartotojų sąskaitos (accounts). | + | + | + | + |

2.7. Programų sistemos kokybės vertinimo charakteristikos

Programinei įrangai vertinti labai dažnai dėl savo paprastumo ir aiškumo yra naudojamas programinės įrangos kokybės užtikrinimo standartas ISO/IEC 9126 (Quality Model) [Bevan, 1999]. Tačiau atitinkamai šiam standartui autorius Gordan Gledec pateikia internetinių sistemų kokybės užtikrinimo modelį pagrįstą minėtoju standartu[9].

Kokybės modelis yra sudarytas hierarchijos principu kur yra 6 aukščiausio lygio savybės (paimtos iš ISO standarto) ir eilė žemesnio lygio savybių (subsavybių) priklausančių kiekvienai aukščiausio lygio savybei. Panašiai siūlo vertinti sudėtingesnių internetinių projektų kokybę ir kiti autoriai išskirdami minėtojo modelio pagrindines savybes[9][10]:

1. **Funkcionalumas** yra programinės įrangos gebėjimas užtikrinti funkcijas ir savybes atitinkančias reikalavimus, kai sistema yra naudojama prie apibrėžtų sąlygų. Atitinkamai programinės įrangos paskirčiai, funkcionalumas gali būti suskaidytas į tokias subsavybes kaip *tinkamumas*, *tikslumas*, *funkcinis suderinamumas* ir *apsauga*. Šių subsavybių bei jų charakteristikų svarbumas ir apimtis priklauso nuo sistemos paskirties.
2. **Vartoseną** yra nustatoma atsižvelgiant į žmogiškus veiksnius kaip gebėjimą suprasti, išmokti ir kaip estetika, kai programinė įranga yra naudojama prie apibrėžtų sąlygų. Iš esmės vartoseną atspindi lengvumą, kuriuo interneto naršyklė išnaudoja teikiamą internetinės programinės įrangos funkcionalumą. Tinkamumo subsavybės yra *naudojimo lengvumas*, *turinio suprantamumas*, *nuorodų gylis* ir *patrauklumas*.
3. **Patikimumas** yra programinės įrangos gebėjimas išlaikyti apibrėžtą našumo lygį, kai sistema yra naudojama prie apibrėžtų sąlygų. Patikimumo subsavybės yra *naudingumas*, *atsparumas gedimams* ir *saugumas*.
4. **Efektyvumas** yra programinės įrangos gebėjimas tiekti atitinkamą našumą priklausomai pagal naudojamą resursų kiekį prie nustatytų sąlygų. Internetinės sistemos pusėje resursais gali būti kiti programinės įrangos produktai, aparatinė įranga (diskai, atmintis, procesorius), finansinės išlaidos ir panašiai. Vartotojo pusėje resursais gali būti ryšio išlaidos, laikas, pastangos ir panašiai. Efektyvumo subsavybės yra *laikinė elgsena*, *resursų panaudojimas*, *mastelio keičiamumas* ir *lankstumas*.
5. **Palaikomumas** yra programinės įrangos galimybė būti modifikuota. Modifikacija gali būti laikomi pataisymai, patobulinimai arba internetinės sistemos aplinkos, reikalavimų ir funkcinės specifikacijos pritaikymas. Palaikomumo subcharakteristikos yra *analizavimo lygis*, *stabilumas*, *pakeičiamumas* ir *testuojamumas*.
6. **Perkeliamumas** yra programinės įrangos galimybė būti perkelta iš vienos aplinkos į kitą,

papildyta, testuota, suderinta ir konfigūruota. Perkeliavimo subsavybės yra *pritaikomumas, įdiegiamumas* ir *prisitaikomumas*.

Pirmos keturios savybės (funkcionalumas, vartoseną, patikimumą ir efektyvumą) daugiausia įtakoja kokybę iš vartotojo pusės. Tuo tarpu paskutinės dvi savybės (palaikomumas ir perkeliavimas) yra labiausiai juntamos administratoriaus pusėje.

Kai kurie autoriai nurodo, kad aptartų savybių sąrašas nebūtinai yra baigtinis ir privalomas[10]. Priklausomai nuo sistemos specifinių sprendimų šios savybės gali persidengti arba nebūtinai turi būti įtrauktos vertinant kokybę. Tačiau nurodoma, kad šių savybių turi užtekti įvertinti internetinės programinės įrangos kokybę.

Remiantis šiais kriterijais įvertintos sistemos kokybė turėtų būti neprastesnė kaip vidutiniška.

2.8. Išvados

1. Išanalizavus vartotojo iškeltus pradinis reikalavimus sistemai, nustatyti pagrindiniai probleminiai sistemos realizavimo uždaviniai: duomenų integralumas esant ryšio trikdžiams, tinklu perduodamų duomenų saugumas, didelių duomenų kiekių saugojimas. Visiems jiems pateikti galimi sprendimo variantai.
2. Palyginus keletą sistemų realizavimui skirtų ir plačiausiai naudojamų platformų platformų (PHP, Java, C++), taip pat atsižvelgus į sistemai keliamus reikalavimus ir numatomus poreikius jų realizavimui pasirinktos PHP bei Visual C++ sistemos kūrimo platformos.

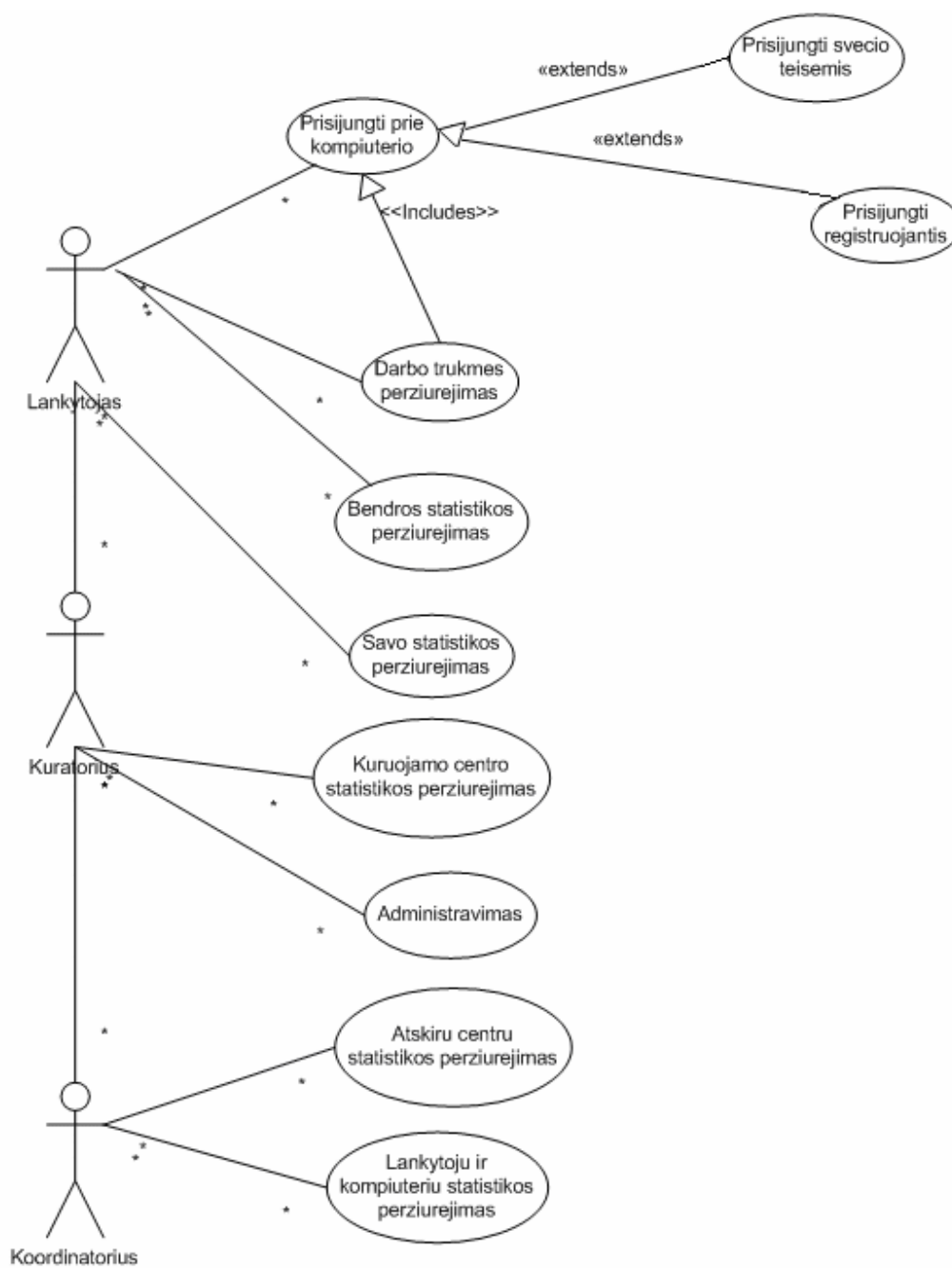
3. NUTOLUSIŲ INTERNETO CENTRŲ STEBĖJIMO IR ADMINISTRAVIMO SISTEMOS REALIZAVIMAS

Šiame skyriuje nagrinėjamos pagrindinės techninės projekto detalės. Konkretizuojami priimti sprendimai, pateikiamos architektūros, modulių ir duomenų vaizdų diagramos.

Pirmuose šio skyriaus poskyriuose trumpai aptariami priimti techniniai sprendimai. Vėliau apžvelgiamas bendras sistemos architektūros vaizdas, tuo pačiu pateikiamos esminės sistemos architektūrą paaiškinančios schemas. Galiausiai pateikiami pagrindiniai sistemos duomenų vaizdai.

3.1. Panaudojimo atvejai

Nagrinėjant surinktus reikalavimus, buvo sudaryta žemiau pateikta panaudojimo atvejų diagrama. Čia pateikiami tik pagrindiniai panaudojimo atvejai be detalesnio jų nagrinėjimo.



2 pav. Sistemos panaudojimo atvejų modelis

Panaudojimo atvejai:

- ✓ Prisijungti prie kompiuterio, vartotojas prisijungia darbui prie kompiuterio
- ✓ Prisijungti svečio teisėmis, jei vartotojas neturi vartotojo vardo ir slaptažodžio, jis gali prisijungti svečio teisėmis, tačiau darbui gali būti taikomi tam tikri apribojimai

- ✓ Prisijungti registruojantis, vartotojas jungiasi su suteiktais vartotojo vardu ir slaptažodžiu, kurie užregistruoti duomenų bazėje.
- ✓ Darbo trukmės peržiūrėjimas, vartotojas gali peržiūrėti kiek laiko jis praleido dirbdamas kompiuteriu.
- ✓ Bendros statistikos peržiūrėjimas, visų nutolusių interneto centrų bendrosios statistiko peržiūrėjimas, prieinamas visiems.
- ✓ Savo statistikos peržiūrėjimas, lankytojas gali peržiūrėti sukauptos apie jį statistikos suvestinę.
- ✓ Kuruojamo centro statistikos peržiūrėjimas, kuratorius peržiūri savo kuruojamo centro statistiką, įvairiais pjūviais.
- ✓ Administravimas, kuratorius gali nustatyti įvairius apribojimus kuruojamo centro kompiuterių naudojimui.
- ✓ Atskirų centrų statistikos peržiūrėjimas, koordinatorius gali peržiūrėti kiekvieno centro, kuriame įdiegta stebėsenos ir administravimo sistema statistiką
- ✓ Lankytojų ir kompiuterių statistikos peržiūrėjimas, koordinatorius gali peržiūrėti kiekvieno lankytojo ir kompiuterio darbo statistiką.

Aktoriai:

- Lankytojas
- Kuratorius
- Koordinatorius

3.2. Funkciniai reikalavimai

Žemiau pateikiami surinktų funkcinių reikalavimų pavadinimai. Reikalavimai buvo surinkti naudojantis Volere šablonu.[7]

1. Vartotojas gali prisijungti darbui prie kompiuterio naudojantis jam suteiktais prisijungimo duomenimis
2. Vartotojas neturėdamas jokių prisijungimo duomenų turi galimybę prisijungti darbui prie kompiuterio svečio teisėmis
3. Vartotojas gali peržiūrėti praleistą prie kompiuterio laiką
4. Vartotojas gali peržiūrėti bendrąją statistiką.

5. Vartotojas gali peržiūrėti savo darbo kompiuteriu statistiką
6. Koordinatorius gali peržiūrėti atskirų centrų statistiką
7. Kuratorius gali peržiūrėti atskirų kompiuterių ir lankytojų statistiką
8. Kuratorius ir koordinatorius turi turėti galimybę eksportuoti kompiuterių panaudojimo statistiką į CSV formatą.
9. Kuratorius turi turėti galimybę įjungti arba išjungti interneto apribojimą kiekvienam kompiuteriui jei vartotojas jungiasi svečio teisėmis
10. Sistemos vartotojų paskyros naudojamos tokios pačios kaip ir www.vipt.lt portale.
11. Vartotojai skirstomi į grupes, kurios turi skirtingus teisių nustatymus sistemoje
12. Sistemoje turi būti galimybė valdyti vartotojų teises
13. Sistema turi leisti vartotojui rikiuoti statistiką pagal atskirus jos elementus
14. Sistema turi leisti vartotojams peržiūrėti statistiką pagal vartotojų nustatytus laiko režius
15. Sistema turi leisti vartotojams nustatyti kiek rodyti eilučių statistikoje
16. Sistema turi grupuoti statistiką pagal apskritis
17. Sistema turi apskaičiuoti paskutinės savaitės bendrą kompiuterių darbo laiką
18. Sistema turi susumuoti pasirinkto centro kompiuterių darbo laiką pasirinktame arba numatytame intervale
19. Sistema turi fiksuoti paskutinių statistinių duomenų atsiuntimo laiką

3.3. Nefunkciniai reikalavimai

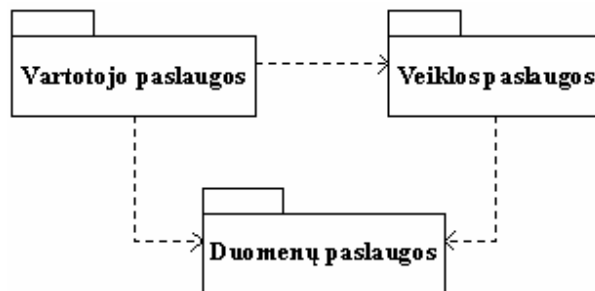
1. Sistema turi būti konkreti ir aiški
2. Aiškus vizualus duomenų pateikimas
3. Sistemoje turi būti galimybė naudoti įvairias kalbas
4. Sistema turi aptarnauti 100 klientų vienu metu
5. Sistema turi veikti populiariose interneto naršyklėse
6. Sistemos serverio patikimumas turi būti didelis

7. Esant būtinybei sistemo serverinę dalį galima perkelti į kitas operacines sistemas
8. Esant būtinybei sistemoje turi būti galimybė naudoti SSL ryšio kodavimas
9. Vartotojai turi būti skirstomi į grupes, taip atskiriant jų privilegijas

3.4. Programinio sprendimo architektūros specifikacija

3.4.1. Bendras architektūros vaizdas

Internetinių sistemų kūrimui paprastai naudojamas trijų sluoksnių architektūrinis modelis. [18] Šiai sistemai taip pat pasirinktas tipinis trijų lygių architektūros modelis. Pagal šį modelį, nutolusių interneto centrų stebėjimo ir administravimo sistemą sudaro vartotojo, veiklos paslaugos ir duomenų paslaugos, sugrupuotos i atitinkamus paketus, pateiktus 3 paveiksle.



3 pav. Sistemos architektūros modelis

3.4.1.1. Paketas „Vartotojo paslaugos“

Paketas „Vartotojo paslaugos“ yra skirtas atlikti tarpininko funkcijas tarp sistemos ir vartotojo. Jame yra sukurtos visos klasės, reikalingos pilnam vartotojo sąsajos funkcionalumui užtikrinti. Realizuota sistema sukurta naudojantis internetine sąsaja bei minimalia kliento dalies programos sąsaja. Internetiniai puslapiai pateikia vartotojui sistemoje esančią informaciją, bei suteikia administravimo galimybes. Kliento dalies programos sąsaja pateikia vartotojui prisijungimo prie kompiuterio vartotojo sąsają.

Paketas „Vartotojo paslaugos“ atsakingas už puslapių bei dinamiško turinio generavimą. Tai yra, bendradarbiaudamas su kitais lygiais, gauna tam tikrą informaciją arba duomenis ir pagal suprogramuotą struktūrą sugeneruoja informacijos pateikimo formą.

Kita šio lygio funkcija yra apdoroti gaunamą iš vartotojo informaciją ir perduoti ją paketui „Veiklos paslaugos“

3.4.1.2. Paketas „Veiklos paslaugos“

Paketas „Veiklos paslaugos“ yra skirtas atlikti tarpininko funkcijas tarp vartotojo sąsajos ir duomenų sąsajos. Jame yra sukurtos visos klasės, reikalingos pilnam sistemos funkcionalumui užtikrinti. Visos funkcijos, kurios turi būti realizuotos programoje, turi tam tikrus loginius scenarijus ir tam tikrus apribojimus arba vykdymo sąlygas. Visą šį veiklos priklausomumą galima vadinti veiklos

taisyklėmis. *Paketas „Veiklos paslaugos“* kaip tik ir yra atsakingas už visą veiklos logiką realizuotą programoje. Šis sluoksnis yra valdantis sluoksnis.

Veiklos logika apima:

- a) visus reikalingus skaičiavimus;
- b) darbo srautų valdymą (taip pat sesijos duomenų stebėjimą ir valdymą);
- c) visų *Atvaizdavimo sluoksnio* duomenų užklausų apdorojimą.

3.4.1.3. Paketas „Duomenų paslaugos“

Paketo „Duomenų paslaugos“ paskirtis yra aprūpinti *Veiklos paslaugų paketą* reikalingais duomenimis bei rašyti iš jo gaunamus duomenis į duomenų bazę. Kitaip tariant, šis sluoksnis yra tarpininkas tarp fizinės duomenų bazės ir programos, o konkrečiu atveju *Veiklos paslaugų paketo*.

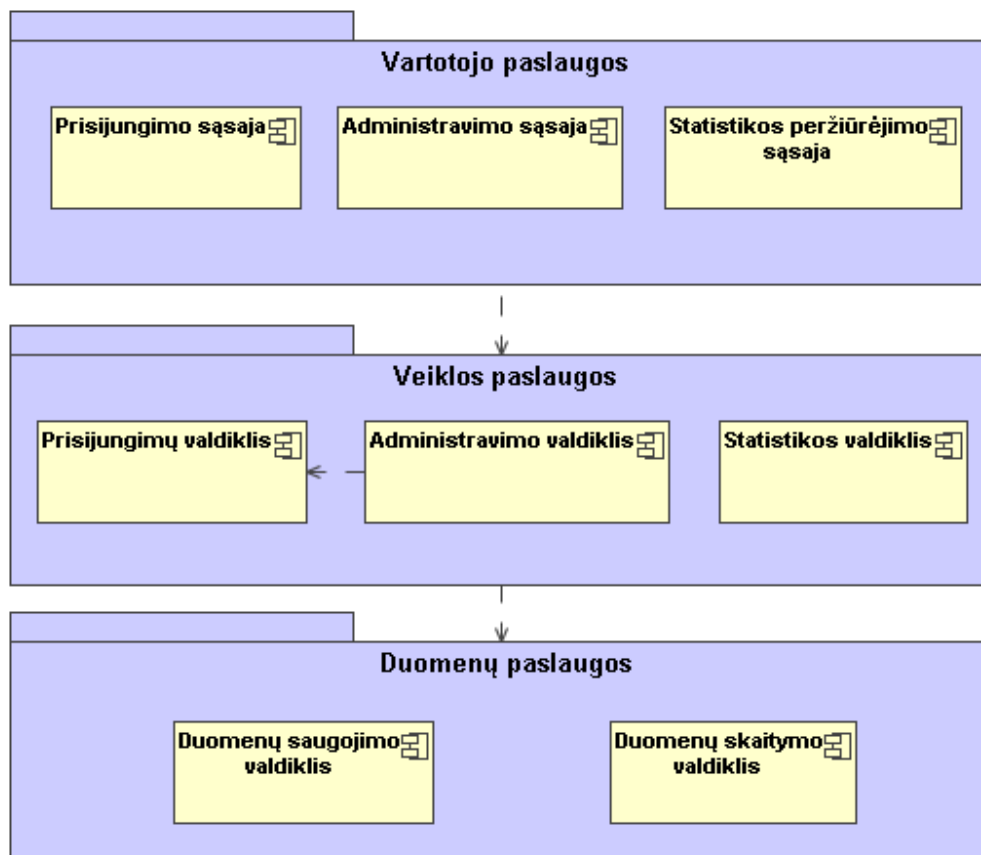
Šis sluoksnis atsako už ryšį su duomenų baze. Užtikrina ryšio palaikymą, nutraukimą ir duomenų gavimą iš duomenų bazės.

Šiame sluoksnyje yra tik minimalūs patikrinimai ar veiklos logikos elementai, kadangi visa tai turi užtikrinti *Veiklos paslaugų paketas*. Čia išlieka tik tokie veiklos logikos elementai, kaip:

- a) neegzistuojančio objekto patikrinimas;
- b) išorinio rakto ryšio apribojimai.

3.4.2. Detalus architektūros vaizdas

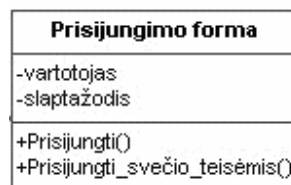
Detalus programos architektūros vaizdas apžvelgia pagrindines realizuotas klases ir jų struktūrą. Klasių aprašymai suskirstyti pagal sistemos modulius. Žemiau pateiktoje diagramoje (4 pav.) pavaizduoti visi pagrindiniai sistemos moduliai, jų grupavimas į loginius paketus ir ryšiai tarp jų.



4 pav. Sistemos modulių diagrama

3.4.2.1. Prisijungimo sąsajos modulio architektūra

Prisijungimo sąsajos modulis skirtas vartotojo prisijungimo lango atvaizdavimui vartotojo kompiuteryje. Šiame modulyje apdorojami vartotojo įvedami prisijungimo duomenys. Žemiau pateikiamos šio modulio pagrindinės klasės(5 pav.)



5 pav. Prisijungimo sąsajos modulio klasė

3.4.2.2. Administravimo sąsajos modulio architektūra

Administravimo sąsajos modulis skirtas administravimo formos pateikimui vartotojui. Šios formos pagalba nustatomi (įjungiami/išjungiami) administravimo parametrai.

Žemiau pateikiamos šio modulio pagrindinės klasės(6 pav.)

| Administravimo sąsajos forma |
|---|
| +Pasirinkti_administravimo_objektą() +Keisti_objekto_atributus() +Kurti_objektą() +Nustatyti_numatytuosius_atributus() |

6 pav. Administravimo sąsajos modulio klasė

3.4.2.3. Statistikos peržiūrėjimo sąsajos modulio architektūra

Statistikos peržiūrėjimo modulis skirtas statistikos sąsajos pateikimui vartotojui. Šioje sąsajoje vartotojas gali nustatyti statistikos filtrus, pasirinkti statistiką kurią jis nori peržiūrėti, taip pat eksportuoti visa tai CSV formatu.

Žemiau pateikiamos šio modulio pagrindinės klasės(7 pav.)

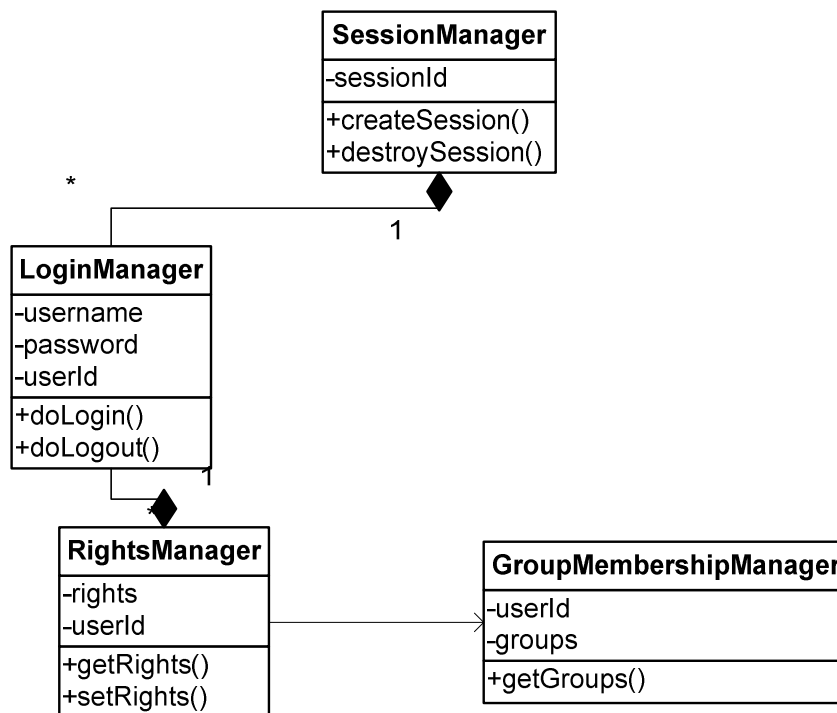
| Statistikos peržiūrėjimo forma |
|---|
| -vartotojas -vartotojo_teisės |
| +Pasirinkti_statikos_kriterijus() +Generuoti() |

7 pav. Statistikos peržiūrėjimo sąsajos modulio klasė

3.4.2.4. Prisijungimų valdiklio modulio architektūra

Prisijungimų valdiklis atsakingas už vartotojo duomenų patikrinimą bei sesijų valdymą. Vartotojo prisijungimo duomenys perduodami iš prisijungimo sąsajos modulio.

Žemiau pateikiamos šio modulio pagrindinės klasės(8 pav.)

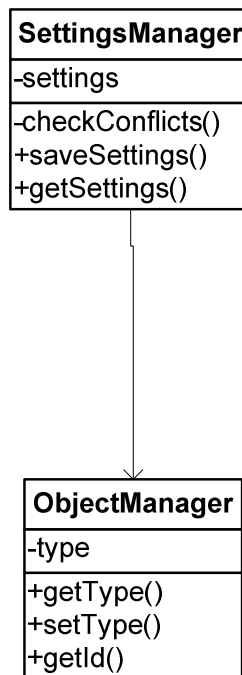


8 pav. Prisijungimų valdiklio modulio klasės

3.4.2.5. Administravimo valdiklio modulio architektūra

Administravimo valdiklio modulis atsakingas už administravimo nustatymų patikrinimą, ar nėra konfliktų tarp vartotojo pasirinktų nustatymų. Pasirinkti nustatymai gaunami iš administravimo sąsajos modulio.

Žemiau pateikiamos šio modulio pagrindinės klasės(9 pav.)

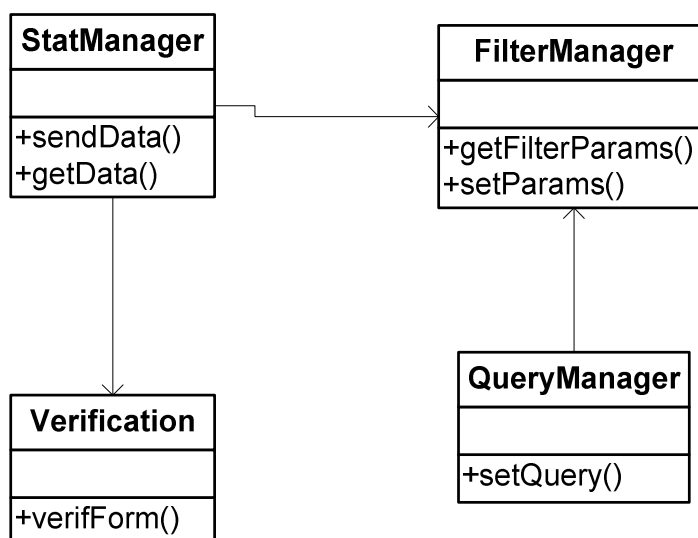


9 pav. Administravimo valdiklio modulio klasės

3.4.2.6. Statistikos valdiklio modulio architektūra

Statistikos valdiklio modulis atsakingas už statistinių duomenų suformavimą ir pateikimą statistikos peržiūrėjimo sąsajos moduliui. Šis modulis gauna vartotojo pasirinktus nustatymus kokią statistiką jis nori matyti, taip pat filtrų nustatymus, suformuoja užklausą duomenų paslaugų paketui ir pateikia suformuotus rezultatus statistikos peržiūrėjimo sąsajos moduliui

Žemiau pateikiamos šio modulio pagrindinės klasės(10 pav.)

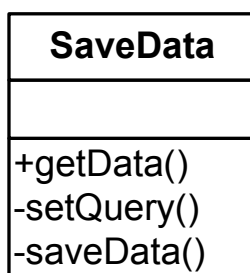


10 pav. Statistikos valdiklio modulio klasės

3.4.2.7. Duomenų saugojimo valdiklio modulio architektūra

Duomenų saugojimo valdiklis yra atsakingas už duomenų įrašymą į duomenų bazę. Duomenys gaunami iš veiklos paslaugų pakete esančių sistemos modulių

Žemiau pateikiamos šio modulio pagrindinės klasės(11 pav.)



11 pav. Duomenų saugojimo valdiklio modulio klasės

3.4.2.8. Duomenų skaitymo valdiklio modulio architektūra

Duomenų skaitymo valdiklis yra atsakingas už duomenų nuskaitymą iš duomenų bazės ir jų pateikimą tam tikra forma. Užklauskos šiam moduliui perduodamos iš veiklos paslaugų paketo komponentų, o suformuoti rezultatai į veiklos paslaugų komponentus.

Žemiau pateikiamos šio modulio pagrindinės klasės(12 pav.)

| |
|---|
| GetData |
| |
| -setQuery() +getQuery() +sendData() |

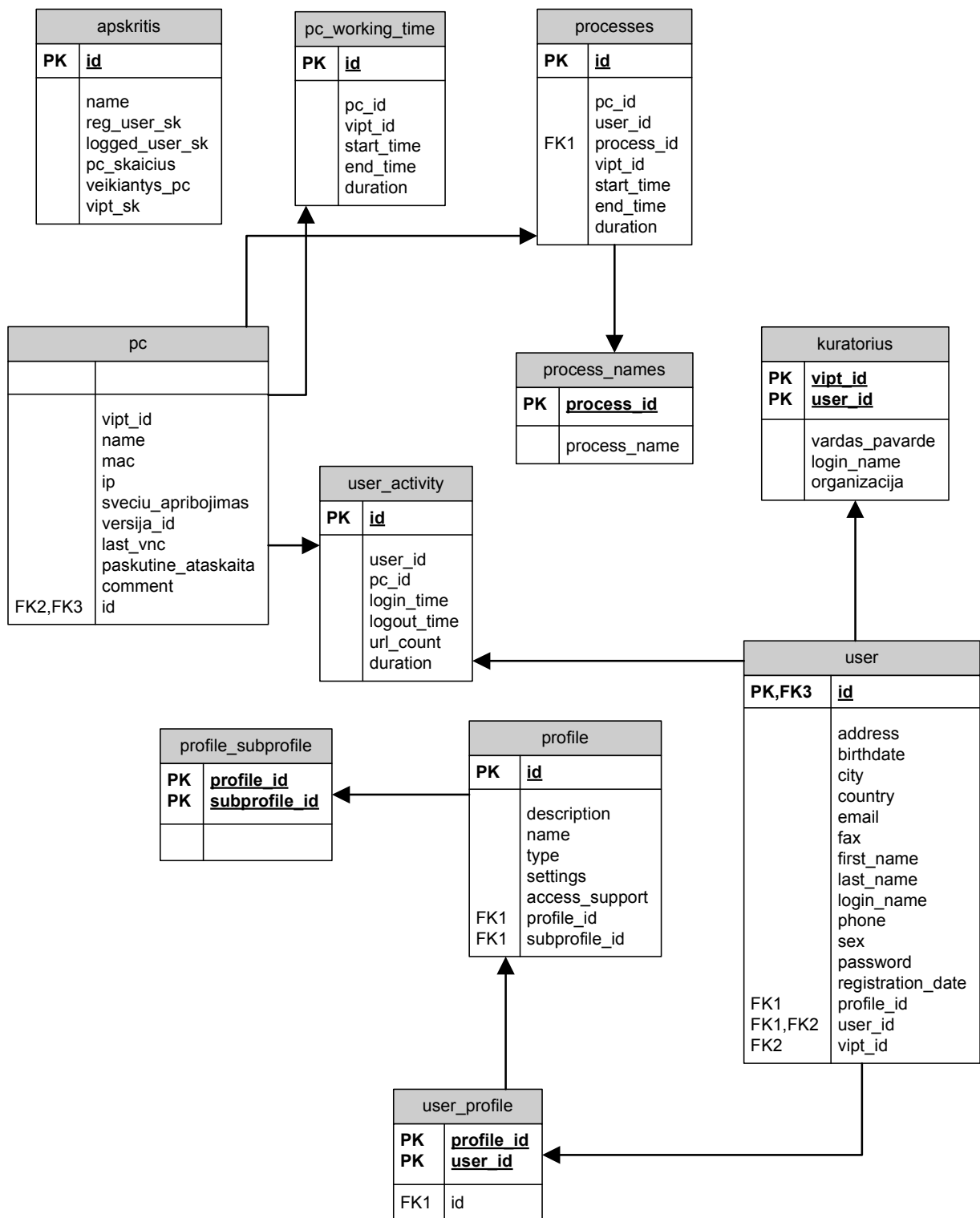
12 pav. Duomenų skaitymo valdiklio modulio klasės

3.4.3. Pagrindiniai duomenų vaizdai

Žemiau esančiuose skyreliuose aprašytos pagrindinės duomenų bazės lentelės.

3.4.3.1. Vartotojų duomenų ir statistikos saugojimo lentelės

- a) Apskritis – šioje lentelėje saugomi duomenys apie apskrityse esančius kompiuterių, registruotų vartotojų bei kompiuterių skaičius
- b) pc_working_time – šioje lentelėje saugoma informacija apie kompiuterių įjungimo ir išjungimo laikus
- c) processes – lentelė skirta saugoti informaciją apie vartotojo naudotas programas, bei kompiuterius kuriuose šios programos buvo naudotos
- d) pc – informacija apie kiekvieną kompiuterį
- e) user_activity – šioje lentelėje saugoma informacija apie vartotojų aktyvumą, prisijungimo bei atsijungimo laikus, aplankytų sveitainių skaičių ir kt.
- f) process_names – lentelė kurioje saugomi procesų pavadinimai
- g) kuratorius – informacija apie kuratorius
- h) user – informacija apie vartotojus
- i) profile – šioje lentelėje saugoma informacija apie profilius, kuriuos galima suteikti vartotojams
- j) profile_subprofile – ryšiai tarp vartotojų profilių bei subprofilų
- k) user_profile – ryšiai tarp vartotojų ir jiems priskirtų profilių

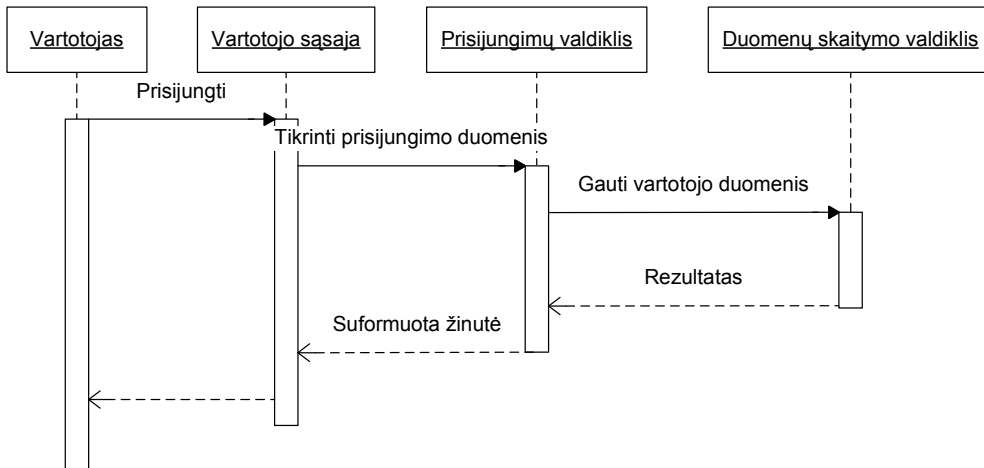


13 pav. Statistikos bei vartotojų informacijos saugojimo lentelės

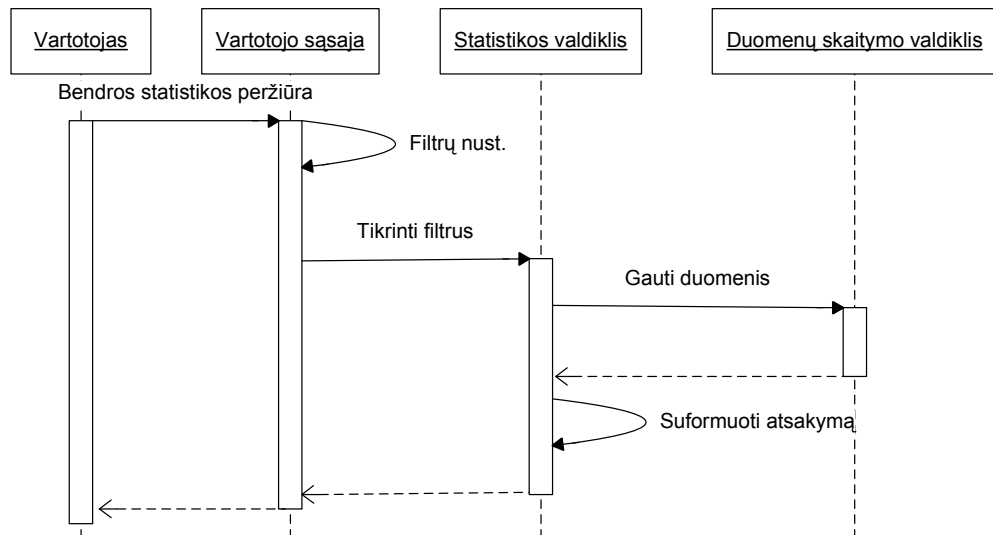
3.4.4. Dinaminiai sistemos vaizdai

Šiame skyriuje pateikiamos pagrindinė nutolusių interneto centrų stebėsenos ir administravimo sistemos sekų, veiklos ir būsenų UML diagramos.

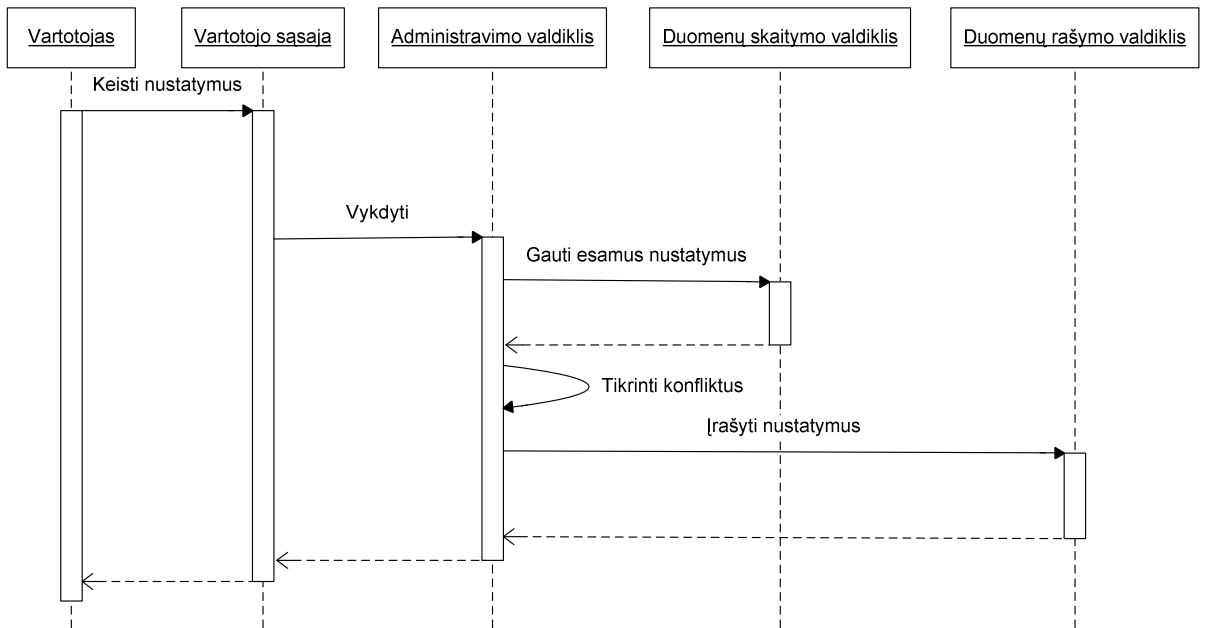
3.4.4.1. Pagrindinės sistemos sekų diagramos



14 pav. Vartotojo prisijungimo prie kompiuterio sekų diagrama

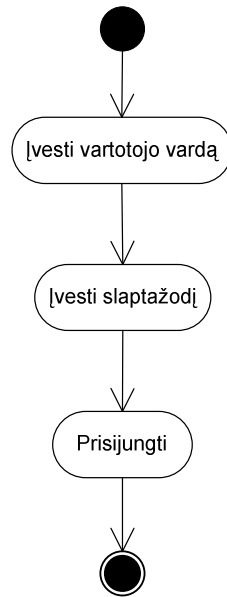


15 pav. Bendrosios statistikos peržiūros sekų diagrama

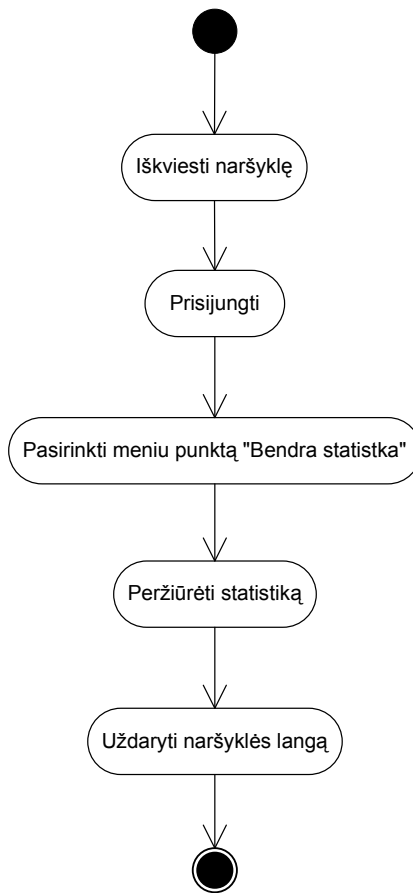


16 pav. Administravimo sekų diagrama

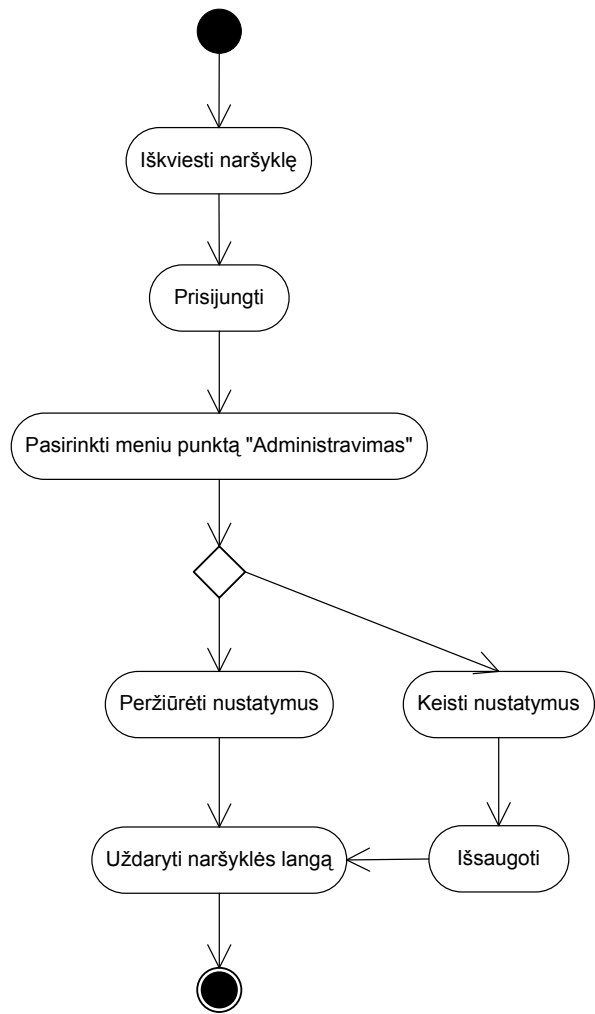
3.4.4.2. Pagrindinės sistemos veiklos diagramos



17 pav. Vartotojo prisijungimo prie kompiuterio veiklos diagrama

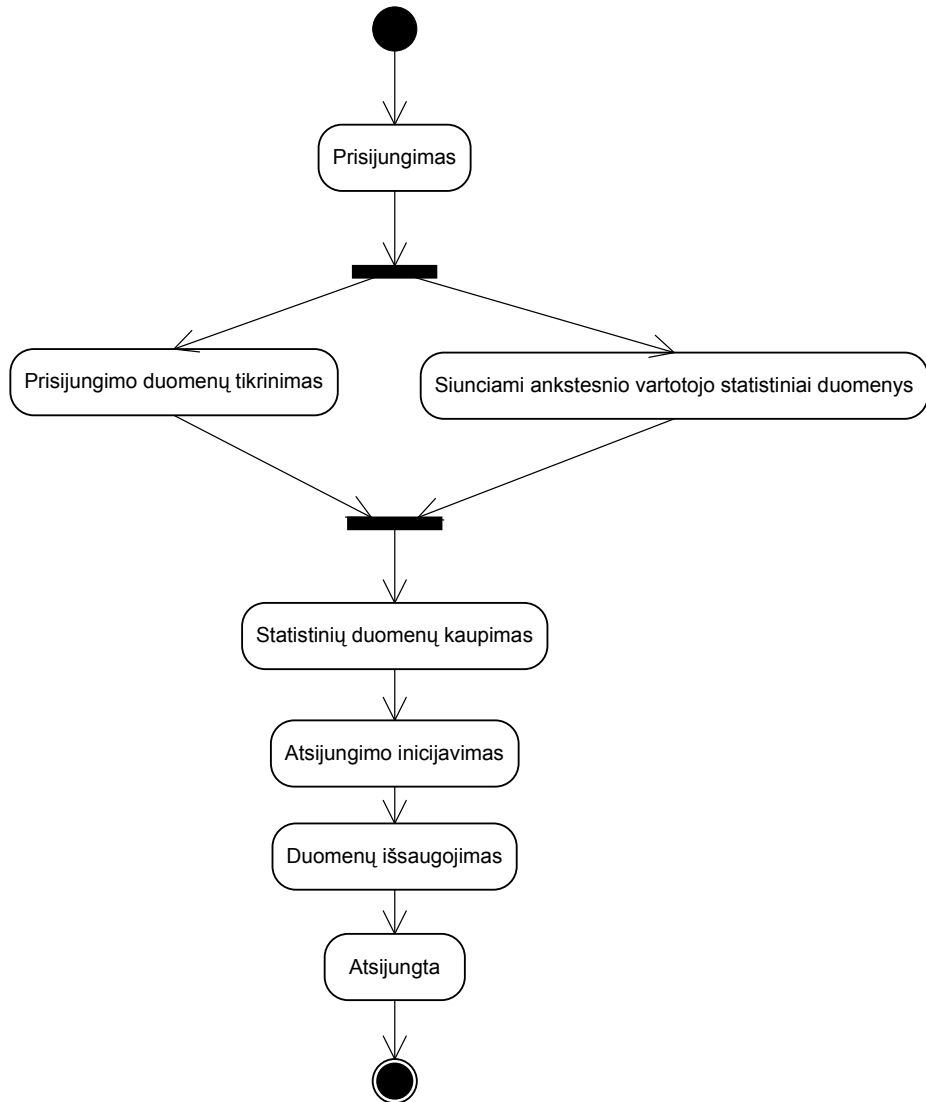


18 pav. Bendrosios statistikos peržiūrėjimo veiklos diagrama



19 pav. Administravimo veiklos diagrama

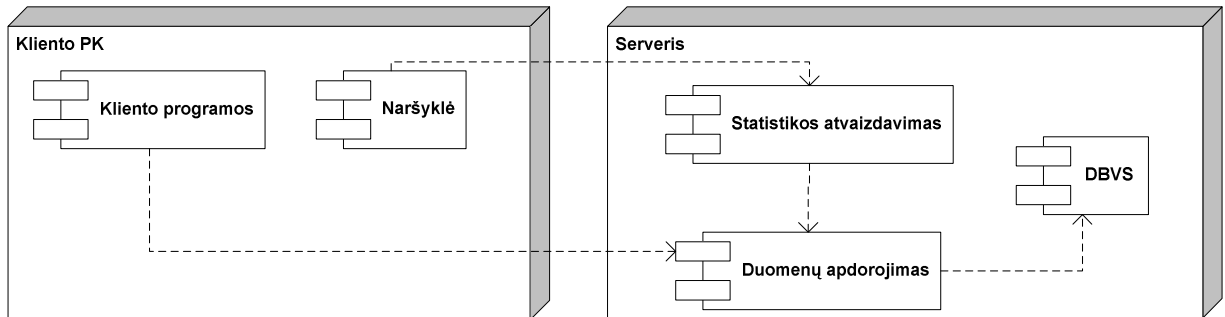
3.4.4.3. Sistemos būsenų diagrama



20 pav. Sistemos būsenų diagrama

3.4.5. Sistemos išdėstymo vaizdas

Kuriama sistema yra suprojektuota veikti pagal kliento-serverio architektūrą. Jos pagrindinių komponentų išdėstymo vaizdas pavaizduotas 21 paveiksle.



21 pav. Sistemos išdėstymo vaizdas

3.5. Išvados

1. Remiantis surinktais bei suformuluotais sistemos reikalavimais bei išanalizuotais probleminiais uždaviniais bei galimais jų sprendimo būdais suprojektuota nutolusių interneto centrų stebėjimo ir administravimo sistema.
2. Sistemai pasirinktas kliento – serverio veikimo principas bei trijų sluoksnių architektūra. Kiekvienas architektūros sluoksnis atskiria sistemos vartotojo sąsają, logiką bei darbą su duomenimis.

4. EKSPERIMENTINIS SISTEMOS KOKYBĖS TYRIMAS

Šiame skyriuje aprašomas atliktas nutolusių interneto centrų stebėjimo ir administravimo sistemos eksperimentinis tyrimas ir gauti rezultatai. Sistemos kokybė buvo vertinama stebint vartotojų darbą su ja, apklausiant juos, taip pat atsižvelgiant į vartotojų komentarus.

4.1. Sistemos funkcionalumas

Analizuojant sistemos funkcionalumą, būtina patikrinti sistemos atitikimą reikalavimų specifikacijai. Nutolusių interneto centrų stebėsenos ir administravimo sistema buvo kuriama nuolat tikrinant reikalavimų dokumentų išpildymą. Testavimo metu nuolat buvo analizuojama veiklos diagrama ir stebimas paeiliui realizuojamų funkcijų teisingumas.

Tinkamumo užtikrinimas. Sistemos tinkamumą užtikrina programinės įrangos atitikimas reikalavimų specifikacijai ir validumo testai su užsakovais. Kaip jau buvo paminėta anksčiau, sistema buvo kuriama sekant reikalavimų dokumento įrašus, todėl sistemos tinkamumas buvo užtikrintas. Jį užtvirtino nuolatinis bendradarbiavimas su užsakovais.

Įvertinimas: 9

Tikslumo užtikrinimas. Sistema buvo kuriama komponentinės architektūros principais. Sukuriamas naujas komponentas, ištestuojamas su įvairiais duomenų rinkiniais, integruojamas į sistemą. Sistemos testavimo metu buvo užfiksuota daug tikslumo defektų, kaip ataskaitų generavimo neatitikimai, tačiau derinimo metu jie buvo pastebėti ir pašalinti. Sistemos kūrėjai labai vertina įvairias pastabas iš vartotojų ir bando jas analizuoti ir priimti sprendimus.

Įvertinimas: 9

Apsaugos užtikrinimas. Vartotojo įvedamas slaptažodis yra šifruojamas kriptografinė hešavimo funkcija MD5. Tai užtikrina slaptažodžių atkūrimo apsaugą. Visi sistemos komponentai ir duomenys yra valdomi per teisių valdiklį, todėl kiekvienas vartotojas gauna tik tą informaciją, kuri jam yra skirta. Priėjimas prie duomenų bazės yra apsaugotas slaptažodžiu.

Įvertinimas: 10

4.2. Sistemos vartoseną

Kaip ir dažnos programinės įrangos kūrimo proceso metu, šios sistemos vartosenos savybės buvo ypač akcentuojamos dėl tikslinės vartotojų grupės ypatumų: vyresnio amžiaus vartotojų ir jų menkų kompetencijų patirtyje darbe su kompiuteriais.

Naudojimo lengvumas. Nutolusių interneto centrų stebėsenos ir administravimo sistemos užsakovai programinės įrangos dizaino koncepcijos kūrimo metu pabrėžė grafinių elementų minimalumo poreikį. Buvo pageidaujamas kuo paprastesnis sistemos valdymo principas.

Atsižvelgiant į išsakytus kliento poreikius buvo sukurtas vartotojo sąsaja, kurios grafika yra visiškai minimali, reikalinga tik duomenų apipavidalinimui ir aiškiam pateikimui. Sukurtos ikonėlės, kurios tiksliai nusako jų funkciją. Bet kuriuo metu bus galima pakeisti sistemos funkcinį išdėstymą pagal sistemos vartotojų pageidavimus.

Įvertinimas: 8

Turinio suprantamumas. Sistemos esminis turinys yra duomenys, formos, ataskaitos, sąrašai. Be jų yra pateikiami patarimai, nurodymai, pastabos, kokias funkcijas atlieka tam tikri komponentai sistemoje, kaip jomis naudotis. Ateityje yra numatyta sistemą papildyti išsamesniais pagalbinais puslapiais, kurie įgalintų suinteresuotus vartotojus sistemos veikti be išorinės pagalbos ar konsultacijų.

Įvertinimas: 7

Nuorodų gylis. Nors sistemos realizavimo metu visas funkcijas buvo stengiamasi realizuoti lengvai pasiekiamas ir lengvai suprantamas, bet šis tikslas nebuvo pilnai pasiektas. Pavyzdžiui, norint pasiekti tam tikro vartotojo naudojimosi kompiuteriu ataskaitą reikia pasirinkti 4 prieš tai esančius nuorodų lygius. Šioje vietoje galima teigti, kad sistema neužtikrina mažo nuorodų gylis.

Įvertinimas: 5

Patrauklumas. Sistemos kūrėjas gali džiuginti faktas, kad nutolusių interneto centrų vartotojai, kuriuose yra įdiegta sistema, tiesiog privalo ja naudotis. Kitoks prisijungimo metodas darbui kompiuteriu nėra priimtinas. Nepaisant šio fakto, informacinės sistemos realizavimo metu buvo siekiama suvokti, kas patiktų sistemos vartotojams, kokios funkcijos erzintų, kaip pripratinti juos prie sistemos, kaip aiškiai pavadinti siūlomas funkcijas.

Įvertinimas: 7

4.3. Sistemos patikimumas

Vertinant internetinės informacinės sistemos patikimumą galima skirti tokius numatytus techninius sprendimus: Ataskaitų generavimo ir peržiūros sistema atlikta naudojant PHP ir MySQL duomenų bazę, vartotojų teisių valdymas per vartotojų grupes ir jų teises į atitinkamus sistemos komponentus.

Vartotojų teisių valdymas užtikrina sistemoje teisingą funkcijų ir duomenų pateikimą teisingiems vartotojams (t.y., kam jos skirtos).

Naudingumas. Sistemos naudingumu būtų sunku sudvejoti, nes sistemą užsakinėjo asmenys, kuriems tos programinės įrangos reikėjo. Jau dabar nutolusių interneto centrų personalas iš sistemos gauna, kiek įmanoma daugiau naudos: generuoja ataskaitas, peržiūri naudojimo statistiką, pergalvoja naujus ataskaitų pūvius, kurie jiems būtų naudingi ateityje.

Įvertinimas: 9

Atsparumas gedimams. Sistemos užtikrina, kad sutrikus vienam sistemos komponentui blogiausiu atveju neveiks tik tas puslapis, kuriame jis patalpintas, bet ne visa sistema. Kadangi sistema yra internetinė, sunku užtikrinti be didesnių resursų skyrimo duomenų išlaidų dingus internetui, sugedus sistemos techninei įrangai (serveriams), tačiau yra saugoma visos sistemos duomenų kopija.

Įvertinimas: 8

4.4. Sistemos efektyvumas

Laikinė elgsena. Sistemos laikinė elgsena nėra labai gera. Pagrindinės iškilusios problemos yra su kelių ar keliasdešimties tūkstančių duomenų ataskaitų pateikimu. Pastebėjus sistemos veiklos sutrikimus, nutarta atlikti duomenų bazės užklausų optimizavimą.

Įvertinimas: 4

Resursų panaudojimas. Sistema nėra imli nei laiko, nei vartotojo kompiuterių resursams.

Įvertinimas: 9

Mastelio keičiamumas. Dalis internetinių sistemų turi fiksuotą mastelį, kitos – keičiamą priklausomi nuo naršyklės lango dydžio. Nėra vieningo sprendimo, kuris realizacijos variantas yra

geresnis. Kadangi sugeneruotoms ataskaitoms paprastai naudojamos lentelės, sistemai buvo pasirinktas keičiamo lango dydis

Įvertinimas: 10

Lankstumas. Sistemos lankstumas yra užtikrinimas jau parinkus realizacijos technologijas. Galima teigti, kad visa internetinė programinė įranga pasižymi lankstumu, vien todėl, kad pasiekama visose pasaulio vietose, kur yra kompiuteris ir internetas. Informacinė sistema yra laisvai papildoma nauja informacija, naujais komponentais, tuo pačiu naujomis funkcijomis su ja susipažinusių žmonių.

Įvertinimas: 8

4.5. Sistemos palaikomumas

Stabilumas. Sistemos stabilumas daugiausiai priklauso nuo sistemos serverio. Šiuo metu sistema patalpinta vykdytojo serveryje, kuriame veikia dar dešimtys kitų projektų, todėl kiekvieną kartą perkraunant vieną sistemą, perkraunamos ir visos kitos. Tomis keliomis minutėmis nutolusių interneto centrų sistema nėra pasiekama. Vystant sistemą toliau gali būti susidurta su sistemos modulių integracija. Pakeitus vieną sistemos modulį gali prireikti keisti ir kitus modulius.

Įvertinimas: 4

Pakeičiamumas. Kaip jau minima prie analizavimo lygio kriterijaus aprašymo, sistemos programinio kodo analizė nėra lengvas procesas. Todėl norint atlikti pakeitimus sistemoje reikia būti susipažinus su sistemos realizacija

Įvertinimas: 5

Testuojamumas. Kuriant sistemą nebuvo atsižvelgta į sistemos testavimą ir tuo pačiu metu nebuvo kuriami sistemos testai, todėl sistemos testuojamumas yra gana prastas.

Įvertinimas: 3

4.6. Sistemos perkeliamumas

Pritaikomumas. Projektuojant sistemą buvo atsižvelgta tik į Windows šeimos operacines sistemas. Sistema veikia tik Windows 2000 ir naujesnėse operacinėse sistemose. Unix šeimos operacinėms sistemoms programinė įranga nebuvo projektuojama.

Įvertinimas: 3

Įdiegiamumas. Sistemos įdiegiamumas yra gana paprasta, nes programinei įrangai veikti pasirinktame serveryje, reikalingi tik MySQL, Apache bei mod_php, kurie paprastai į serverį diegiami standartiškai. Sistemos vartotojui užtenka turėti interneto naršyklę ir surinkti reikiamą adresą jai pasiekti. Kliento dalies diegimas taip pat yra paprastas, nes naudojama NSIS diegimo sistema kuri sukuria standartinį diegimo vedlį.

Įvertinimas: 8

Prisitaikomumas. Veikimo sutrikimų nepastebėta. Sistema naudoja standartinį HTTP protokolą duomenų perdavimui. Iš vartotojo pusės nesvarbu ar naudojama Internet Explorer naršyklė, ar Firefox Mozilla, ar Opera naršyklės, sistemos funkcionalumas nepraranda kokybės. Dėl šių priežasčių galima teigti jog nutolusių interneto centrų stebėsenos ir administravimo tenkina prisitaikomumo kriterijų.

Įvertinimas: 10

4.7. Rezultatai

Įvertinus sistemos kokybę, bendras jos įvertinimo balų vidurkis gavosi 7,2, tai vidutinės kokybės įvertinimas. Tačiau atidžiau pažvelgus į atskirus vertinimo kriterijus, galima pastebėti labai prastai įvertintų sistemos vietų, pavyzdžiui, sistemos testuojamumas. Pagrindinė problema yra ta kad beveik nėra sukurta automatinių sistemos modulių testų. Kita vertus sistemos funkcionalumo įvertinimas gavo gana aukštą balą, kas rodo, kad sistema atitinka vartotojo lūkesčius.

4.8. Tolimesnio sistemos tobulinimo ir plėtojimo galimybės

Stebint sistemos bei vartotojų darbą su sistema buvo išskirtos prioritetinės sistemos tobulinimo galimybės:

Laikinių sistemos charakteristikų pagerinimas. Pradėjus naudoti sistemą realioje aplinkoje, duomenų bazėje per trumpą laiką atsirado daug perteklinių, t.y. nereikalingų duomenų. Taip pat duomenų bazės užklausų realizavimas nebuvo optimalus. Dėl šių priežasčių vartotojų peržiūrimų ataskaitų generavimas nuo sistemos naudojimo pradžios užtrukdavo vis ilgiau. Serverio galingumo padidinimas iš esmės nepakeitė sistemos laikinių charakteristikų rezultatų. Todėl šioje vietoje nutarta atlikti gilią duomenų analizę, sukurti naują duomenų bazės struktūrą, optimizuoti užklausas ir transformuoti duomenis iš senosios duomenų bazės į naująją. Kadangi sistema jau buvo pradėta naudoti, tai turėjo būti padaryta su kuo mažesne įtaka jai.

Kliento dalies pritaikymas .NET platformai. Sistemos klientinė dalis buvo kuriama naudojant Visual C++ programavimo kalbą. Šios technologijos pasirinkimas buvo grindžiamas tuo, kad jos kodas yra kompiliuojamas į mašininį kodą ir nereikia diegti papildomų platformų, taip supaprastinamas sistemos kliento dalies diegimas į vartotojų kompiuterius. Tačiau naudojant Visual C++ susidurta su atminties valdymo problemomis, dėl ko nukenčia sistemos stabilumas. Todėl ateityje nutarta įvertinti galimybę sistemos klientinę dalį perkelti į .NET platformą, kuri yra pati atsakinga už atminties valdymą bei palengvina darbą su duomenų apdorojimu.

5. IŠVADOS

1. Išanalizavus vartotojo reikalavimus sistemoai nustatyti pagrindiniai sistemos kūrimo probleminiai uždaviniai. Išanalizavus šiuos uždavinius nustatyti galimi šių problemų sprendimo būdai.
2. Surinkus ir išanalizavus projektinius ir probleminius uždavinius realizuota nutolusių interneto centrų stebėsenos ir administravimo sistema. Jos architektūra suskirstyta į tris pagrindinius lygius –atvaizdavimo, logikos bei priėjimo prie duomenų.
3. Išanalizavus sistemą pagal ISO 9126 kokybės standartą, nustatyta, kad sistema yra vidutiniškos kokybės ir surinko 7,4 balo. Detaliau panagrinėjus sistemos kokybės kriterijus nustatyta mažiausia balų surinkęs kriterijus yra sistemos palaikomumas. Tai rodo, kad reikia peržvelgti sistemos kodo struktūrą bei sukurti automatizuotus testus.
4. Išanalizavus sistemos kokybę, taip pat nustatyti pagrindiniai tobulinimo kriterijai.
5. Nutolusių interneto centrų stebėsenos ir administravimo sistema atiduota užsakovams KTU ITPI Distanciniam mokymo centrui. Sistema sėkmingai įdiegta ir naudojama apie 1300 kompiuterių 300 nutolusių interneto centrų. Šio dokumento prieduose pateikiamas „Programų sistemos perdavimo ir aprobavimo aktas“

6. LITERATŪRA

- [1] Hypertext Transfer Protocol -- HTTP/1.1 [Žiūrėta 2006.10.20]
prieiga internete <ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt>
- [2] HTTP Over TLS [Žiūrėta 2006.11.05]
prieiga internete <ftp://ftp.rfc-editor.org/in-notes/rfc2818.txt>
- [3] The SSL Protocol Version 3.0 [Žiūrėta 2006.11.05]
prieiga internete <http://www21.ocn.ne.jp/~k-west/SSLandTLS/draft302.txt>
- [4] The MD5 Message-Digest Algorithm [Žiūrėta 2006.10.20]
prieiga internete <ftp://ftp.rfc-editor.org/in-notes/rfc1321.txt>
- [5] DES encryption algorithm [Žiūrėta 2006.10.20]
prieiga internete <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
- [6] Visual C++ Developer Center [Žiūrėta 2006.09.30]
prieiga internete <http://msdn2.microsoft.com/en-us/visualc/default.aspx>
- [7] The Atlantic Systems Guild, Volere Requirements Specification Template [Žiūrėta 2006.12.19]
Prieiga internete: <http://www.systemguild.com/GuildSite/Robs/Template.html>
- [8] N.Bevan. Quality in use for all. User interfaces for all. Stephanidis, C (ed), Lawrence Erlbaum, 1999.
Prieiga internete: http://www.usabilitynet.org/papers/quality_in_use_for_all.pdf
- [9] G.Gledec. Evaluating Web Site Quality. 2005.
Prieiga internete: http://www.carnet.hr/CUC/program/papers/abs/b1_gledec_abs.pdf
- [10] J.Offutt. Web Software Applications QualityAttributes. Quality Engineering in Software Technology (CONQUEST 2002), pages 187-198, Nuremberg, Germany, September 2002.
Prieiga internete: <http://ise.gmu.edu/faculty/ofut/rsrch/papers/conquest02.pdf>
- [11] Hayden D. *Web Applications: N-Tier vs. N-Layer - Benefits and Trade-Offs*. 2005 m. [žiūrėta 2007-04-30]
Prieiga per internetą: <<http://davidhayden.com/blog/dave/archive/2005/07/22/2401.aspx>>.
- [12] Freeman A., Jones. A. *Programming .NET Security*. O'Reilly. 2003.

- [13] Gerard J., Rajeev J. *Reliable Software Systems Design: Defect Prevention, Detection, and Containment*. Jet Propulsion Laboratory, California Institute of Technology. [žiūrēta 2007-05-01]
Prieiga per internetą: <<http://vstte.ethz.ch/Files/holzmann-joshi.pdf>>.
- [14] ISO 9126: *The Standard of Reference*. [žiūrēta 2007 04 30]
Prieiga per internetą: <http://www.cse.dcu.ie/essiscope/sm2/9126ref.html>
- [15] Howard M. *The 19 Deadly Sins of Software Security*. Michael Howard's Web Log. [žiūrēta 2007-04-20]
Prieiga per internetą:
<http://blogs.msdn.com/michael_howard/archive/2005/07/11/437875.aspx>.
- [16] Montlick T. *What is Object-Oriented Software? Software Design Consultants, LLC*. 1999 m. [žiūrēta 2007-05-05.]
Prieiga per internetą: <<http://www.softwaredesign.com/objects.html>>.
- [17] Booch G., Rumbaugh J., Jacobson I. *The Unified Modeling Language User Guide*. Addison-Wesley. 1998.
- [18] Benett G. *Working Together, Apart : Intranet Journal*. [žiūrēta: 2007 04 11]
Prieiga per internetą: <<http://intranetjournal.com/features/idm0398-pm1.shtml>>.

7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

HTTP – Hyper Text Transfer Protocol(hiperteksto perdavimo protokolas)

FTP – File transfer protocol(failų perdavimo protokolas)

.NET – Microsoft sukurta sistema palengvinanti programų kūrimą

C++ – Programavimo kalba

MD5 – Duomenų kodavimo algoritmas, be galimybės juos atkoduoti(Message digest algorithm)

DES – Duomenų kodavimo standartas(Data Encryption Standard)

SSL – Saugus ryšio sluoksnis(Secure Sockets Layer)

TLS – Perdavimo sluoksnio saugumas (Transport layer security)

NSIS – Nullsoft įdiegimo sistema(Nullsoft install system)

API – Programų kūrimo interfeisas(Application programming interface)

Socket – programavimo kalbos priemonė užmegzti ryšiui tarp kompiuterių

WWW – pasaulinis duomenų tinklas(World Wide Web)

HTML – Hiperteksto žymėjimo kalba (angl. Hypertext Markup Language) - tai kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete. Ji naudojama apibrėžti dokumento struktūrai.

VIPT – viešieji interneto prieigos taškai

LIEDM – Lietuvos distancinio mokymosi tinklas

IP – Internet Protocol, esminis šiuolaikinio Interneto adresavimo protokolas.

RFC – Request For Comments, tai dokumentas kuriame detalai aprašomas protokolas/metodas/technologija.

NTP – Network Time Protocol – laiko sinchronizacijos/suderinimo protokolas

PĮ – programinė įranga

CSV – kableliais atskirtų reikšmių struktūro failas

Visual C++ - Bendros paskirties, aukšto lygio, objektiškai orientuota, programavimo kalba sukurta Microsoft korporacijoje.

PA – panaudojimo atvejis

UML - Unifikuota modeliavimo kalba, naudojama objektiškai orientuotame projektavime (angl. Unified Modeling Language)

XML – Išplečiama žymių (arba teksto išdėstymo ir struktūrizavimo) kalba (angl. eXtensible Markup Language)

CASE – Programų projektavimas kompiuterio pagalba (angl., Computer Aided Software Engineering)

PRIEDAS 1: ĮDIEGIMO DOKUMENTAS

KTU DISTANCINIO MOKYMO CENTRAS

Kauno Technologijos universitetas
Distancinio mokymo centras
Studentų 48a- 308, LT-51367 Kaunas
Tel. (8 37) 300612
Faksas (3 37) 300614

PROGRAMŲ SISTEMOS PERDAVIMO IR APROBAVIMO AKTAS

200 7 m. Rugpjūčio 39 d.

Programų sistemos pavadinimas Nutolusių interneto centrų stebėjimo ir administravimo sistema

Kūrinio tipas Programinė įranga

Programų sistemos sukūrimo data 200 7 m. Birželio 30 d.

Kūrinio įteikimo UŽSAKOVUI data 200 7 m. Liepos 30 d.

Užsakovo arba trečiojo asmens Kūrinio aprobavimo rezultatas:

Programinė įranga įdiegta 1300 kompiuterių, 300 nutolusių interneto centrų

Kūrinio aprobavimo data 200 7 m. Rugpjūčio 30 d.

Kūrinio originalo saugotojas - KTU Distancinio mokymo centras

AUTORIUS

UŽSAKOVAS

Nauris Augulis

(vardas, pavardė)

(parašas)

(vardas, pavardė)

(parašas)

Asmens pareigų pavadinimas

(Parašas)

(Vardas ir pavardė)

PRIEDAS 2: APKLAUSA

Apklaustos forma

Prašome įvertinti žemiau pateiktus nutolusių interneto centrų stebėjimo ir administravimo sistemos dalių kokybės kriterijus

Vertinimą prašome pateikti dešimtbalėje sistemoje, kur:

0 – prasta kokybė;

10 – puiki kokybė

1. Įvertinkite sistemoje realizuotas darbu reikalingas funkcijas(10 – funkcijų netrūksta, 0 – trūksta daug elementarių funkcijų)
Įvertinimas ____
2. Įvertinkite sistemos atliekamų ataskaitų generavimų tikslumą
Įvertinimas ____
3. Įvertinkite sistemoje esančių asmeninių duomenų prieinamumą pašaliniams asmenims
Įvertinimas ____
4. Įvertinkite naudojimosi sistema aiškumą ir valdymą
Įvertinimas ____
5. Įvertinkite pagalbos žinybų bei paaiškinimų kiekį
Įvertinimas ____
6. Įvertinkite naršymo patogumą ir paprastumą peržiūrėti bei generuojant ataskaitas
Įvertinimas ____
7. Įvertinkite sistemos vartotojo sąsajos išvaizdą
Įvertinimas ____
8. Įvertinkite ataskaitų generavimo trukmę
Įvertinimas ____
9. Įvertinkite kompiuterio resursų sumažėjimą įdiegus kliento programas(10 – kompiuterio darbas nepasikeitė, 0 – kompiuteris tapo daug lėtesnis)
Įvertinimas ____
10. Įvertinkite sistemos generuojamų ataskaitų atsinaujinimo trukmę
Įvertinimas ____
11. Įvertinkite sistemos konfigūravimo galimybes

Įvertinimas ____

12. Įvertinkite sistemos kliento dalies įdiegimo sudėtingumą (10 – įdiegimas paprastas ir aiškus, 0 – įdiegimas visiškai nesuprantamas)

Įvertinimas ____

Apklauso rezultatai

Apklausa buvo atliekama praėjus vienam mėnesiui po sistemos įdiegimo pas užsakovą. Apklausti šeši žmonės, visi žmonės tiesiogiai susiję su darbu nutolusiuose interneto centruose. Apklauso rezultatų suvestinė pateikiama toliau esančioje lentelėje.

| Kriterijai | 1 | 2 | 3 | 4 | 5 | 6 | Vid. |
|----------------------------------|----|----|----|----|----|----|-----------|
| Sistemos funkcijos | 10 | 10 | 9 | 10 | 9 | 10 | 10 |
| Generuojamų ataskaitų tikslumas | 9 | 10 | 9 | 10 | 10 | 9 | 10 |
| Asmeninių duomenų apsauga | 10 | 10 | 10 | 10 | 9 | 10 | 10 |
| Sistemos aiškumas ir valdymas | 7 | 8 | 6 | 8 | 9 | 8 | 8 |
| Pagalba ir paaiškinimai | 5 | 6 | 5 | 7 | 6 | 8 | 6 |
| Naršymo patogumas | 6 | 7 | 5 | 5 | 4 | 5 | 5 |
| Vartotojo sąsajos išvaizda | 5 | 8 | 6 | 7 | 6 | 7 | 7 |
| Ataskaitų generavimo trukmė | 3 | 2 | 4 | 6 | 3 | 5 | 4 |
| Resursų naudojimas | 9 | 9 | 9 | 9 | 8 | 9 | 9 |
| Duomenų atsinaujinimas | 10 | 9 | 10 | 10 | 8 | 10 | 8 |
| Sistemos konfigūravimo galimybės | 6 | 6 | 5 | 8 | 7 | 4 | 6 |
| Įdiegimo sudėtingumas | 8 | 9 | 9 | 8 | 9 | 8 | 9 |