



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

Rolandas Juodžbalis

Įmonės vidinio tinklo taikomoji programa

Magistro darbas

Darbo vadovas

dr. A. Ostreika

Kaunas
2004

1. Turinys

1.	Turinys	2
2.	Įvadas	3
3.	Sistemos architektūra	5
3.1.	Aukščiausio lygio duomenų srautų diagrama	7
3.2.	Duomenų srautų diagrama (nulinio lygio)	8
3.3.	USE CASE diagrama	9
4.	Technologinių platformų analizė	10
4.1.	Operacinės sistemos	10
4.2.	Duomenų bazės	11
4.3.	Programavimo kalbos	12
4.4.	HTTP serveriai	14
5.	Programinės įrangos projektas	15
5.1.	Reikalavimai sistemai	15
5.2.	Technologinis sprendimas	15
5.3.	Duomenų struktūra	15
6.	Programinės įrangos UML diagramos	17
7.	Sistemos aprašymas	22
8.	Sistemos įdiegimas	23
8.1.	LAMP platformos įdiegimas	23
8.2.	Programinės įrangos įdiegimas	24
9.	Darbas su sistema	26
9.1.	Sistemos administravimas	26
9.2.	Vartotojo darbas	31
9.3.	Remonto meistro darbas	34
10.	Bandomosios eksploatacijos aprašymas	36
11.	Išvados	40
12.	Literatūra	41
13.	Terminų ir santrumpų žodynelis	42
14.	Summary	43
15.	Priedai	44
15.1.	Duomenų bazės kodas	44

2. Įvadas

Programinės įrangos taikymas įvairiuose verslo aspektuose pagerina jų darbo kokybę ir didina produktyvumą. Rinkoje yra daugybė programinių sprendimų, skirtų kompiuterizuoti įvairias įmonės veiklos sritis. Plačiausiai paplitę – buhalterijos apskaitos sistemos, palengvinančios buhalterinės apskaitos vedimą, turinio valdymo sistemos, palengvinančios Interneto portalo valdymą, bei kitos. Galima pabrėžti, kad šie sprendimai įgalina didesnę rinkos skverbti, pilnutinį ir skaidrų informacijos valdymą bei auditą, individualizuotą nuolatinį ryšį su klientu.

Nepaisant gausybės egzistuojančių sistemų egzistuoja specializuotos programinės įrangos trūkumas, kuris ir yra adresuojamas šiame magistriniame darbe. Šio magistrinio darbo objektas – asmeninių kompiuterių ir komponentų remonto informacinė sistema. Šioje srityje nėra specializuotų, reikalingų užduočiai atlikti programinės įrangos sprendimų. Daugiau yra universalių sprendimų, kurie nėra visiškai tinkami šiai užduočiai.

Tyrimo metu bandoma analizuoti egzistuojančias sistemas, jas vertinti ir taikyti užduoties atlikimui. Apžvelgus esamus programinės įrangos sprendimus yra suprojektuota ir realizuota specifikacijoje nusakytus poreikius ir reikalavimus atitinkanti informacinė sistema. Nauja programinė įranga yra specifinė, adaptuota ir taikoma konkrečios įmonės darbe.

Siekiant pilnai sukurti specifinę programinę įrangą, nepakanka tirti vien tik ją ir panaudojimo sritį. Papildomai reikia analizuoti alternatyvias projekto realizacijos technologines platformas: duomenų bazę, programavimo kalbą, operacinę sistemą. Visa tai apima pilną platformą, kurią ir reikia parinkti bei sukurti kaip sprendimą pasirinktai darbo sferai. Šiuo metu labiausiai paplitusios dvi pagrindinės operacinės sistemos, kurių pasirinkimas labiausiai nusprendžia kokia bendra sprendimo platforma bus naudojama. Microsoft Windows operacinė sistema paplitusi pasaulyje kaip komercinė ir universali. Linux (Unix) operacinė sistema yra naudojama kaip nemokama arba dalinai nemokama alternatyva Microsoft Windows operacinei sistemai. Tyrimo metu analizuojamos abi operacinės sistemos ir pagal nustatytus kriterijus parenkama viena iš jų. Vėliau parenkami kiti kompleksiniam sprendimui reikalingi komponentai: programavimo kalba, duomenų bazė, pasiekimo sistema. Tai būtų trumpas darbo krypties aprašymas.

Pagrindinis darbo objektas yra programinės įrangos, skirtos įmonės pasirinktos veiklos sričiai, sukūrimas. Programinei įrangai kurti buvo paruoštas projektas:

- Atlikta esamos programinės įrangos ir sistemų analizė, išskirti jos privalumai ir trūkumai;
- Sukurti ir išanalizuoti vartotojo reikalavimai;
- Išskirti funkciniai reikalavimai;
- Sudarytas programinės įrangos modelis;

- Nustatytos duomenų struktūros;
- Nustatytas programinės įrangos kūrimo grafikas;
- Sukurtos duomenų bazės lentelės;
- Sukurtas vartotojo interfeisas;
- Sukurtas sistemos tvarkytojo interfeisas;
- Sukurti sistemos moduliai ir ryšiai tarp jų;
- Nustatytas sistemos testavimo planas;
- Parašyta vartotojo dokumentacija, susidedanti iš šių dalių: programinės įrangos funkcinio aprašymo, programinės įrangos vadovo, sistemos reikalavimų ir instaliavimo aprašymo ir programinės įrangos tvarkytojo aprašymo.

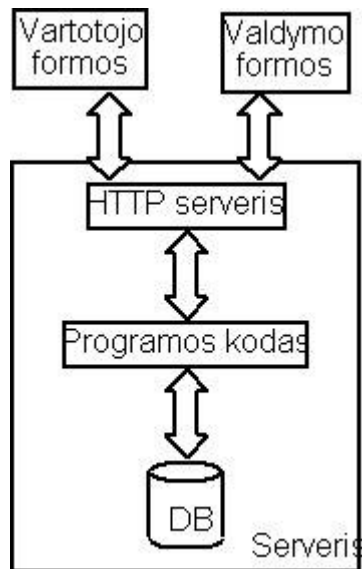
Šio dokumento struktūra yra sudaryta sekančiai:

- Sistemos architektūra – aprašoma pati sistema, jos sudėtinės dalys ir sąveikos tarp jų;
- Technologinių platformų analizė – aprašomos įvairios galimos platformos ir jų kombinacijos;
- Programinės įrangos projektas – aprašomas konkrečiam darbui pasirinktas technologinis sprendimas bei funkciniai ir nefunkciniai reikalavimai;
- Programinės įrangos dokumentavimas – aprašoma dokumentacija, skirta įvairiems programinės įrangos vartotojams;
- Programinės įrangos kokybės įvertinimas – aprašomas bandomasis diegimas ir testavimas.

3. Sistemos architektūra

Vidinė informacinės sistemos struktūra turi šiuos blokus:

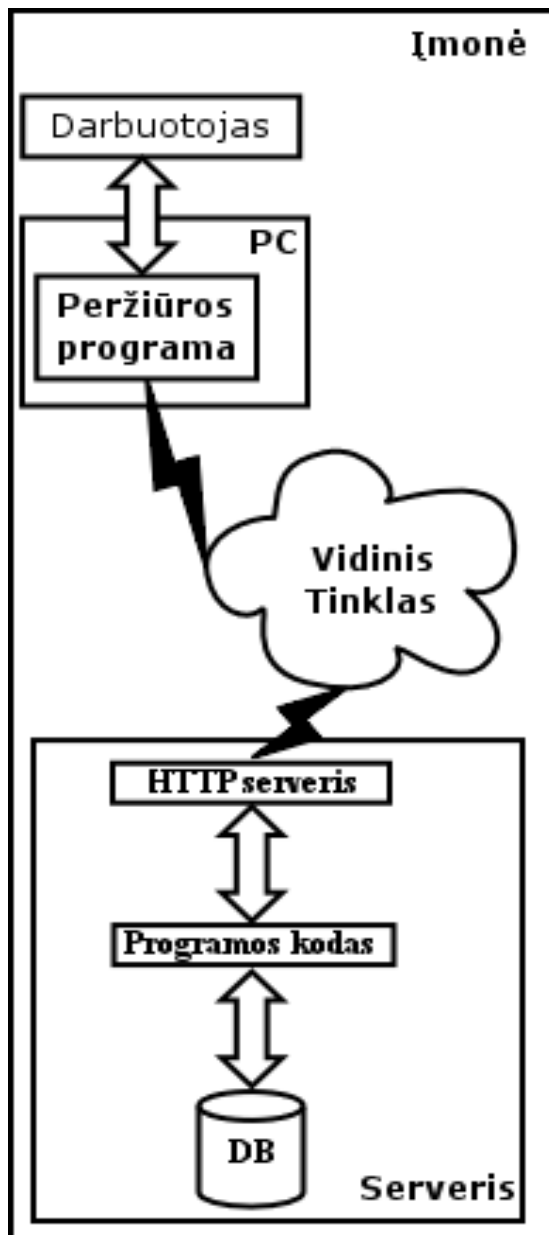
- Vartotojo formos. Funkcija – informacijos įvedimas, išvedimas;
- Valdymo formos. Funkcija – valdymo informacijos įvedimas, išvedimas;
- HTTP serveris. Funkcija – perdavinėti informaciją tarp formų ir programos kodo;
- Programa (kodas). Funkcija – vykdyti įvedimo, išvedimo operacijas, valdyti duomenų bazę;
- Duomenų bazė. Funkcija – duomenų, reikalingų sistemos funkcionavimui saugojimas.



1 pav. Vidinė informacinės sistemos struktūra

Darbas su sistema vyksta tokia tvarka:

- Darbuotojas pasinaudodamas Interneto peržiūros programa prisijungia prie sistemos naudodamasis savo vartotojo vardu ir slaptažodžiu. Meniu pasirenka jam reikalingą modulį ir gali dirbti;
- Informacinė sistema darbuotojui formas išvedinėja HTTP serverio pagalba;
- Informacinė sistema duomenis saugo duomenų bazės serveryje.

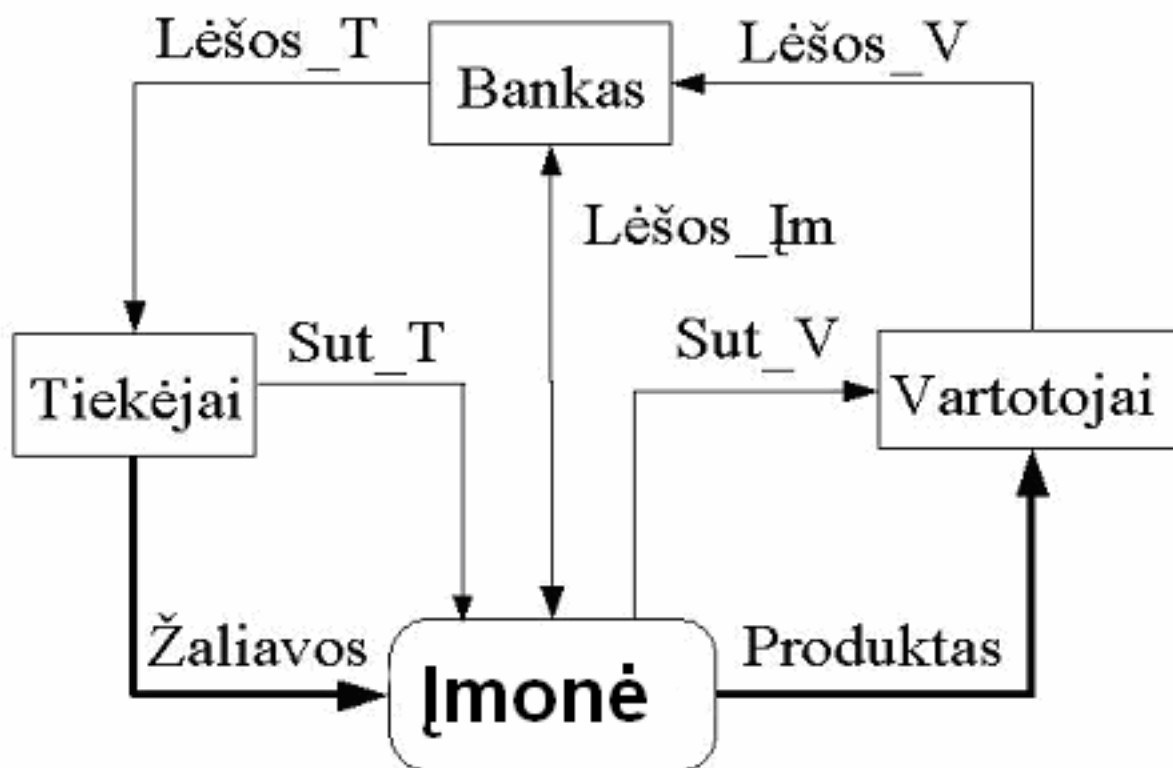


2 pav. Darbas su sistema

Toliau pateikiamos duomenų srautų diagramos, parodančios tokius duomenų srautus:

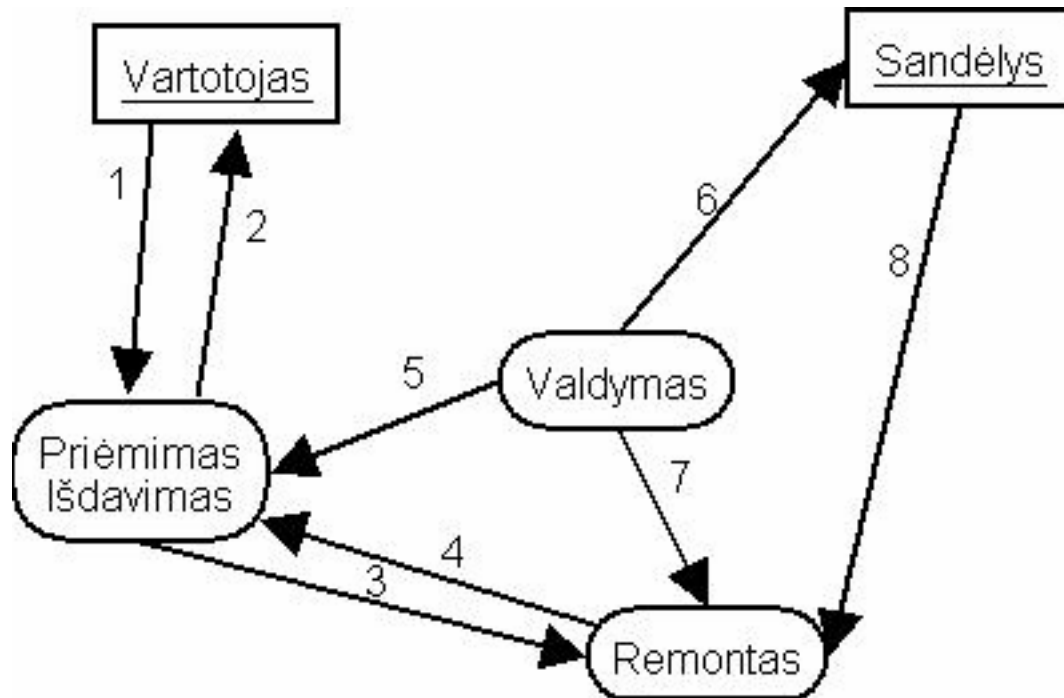
- Aukščiausio lygio diagrama, parodanti įmonės komunikaciją su išore (kitoms įmonėmis, bankais, tiekėjais, klientais);
- Nulinio lygio diagrama, parodanti sistemos vidinį darbą.

3.1. Aukščiausio lygio duomenų srautų diagrama



3 pav. Aukščiausio lygio duomenų srautų diagrama

3.2. Duomenų srautų diagrama (nulinio lygio)

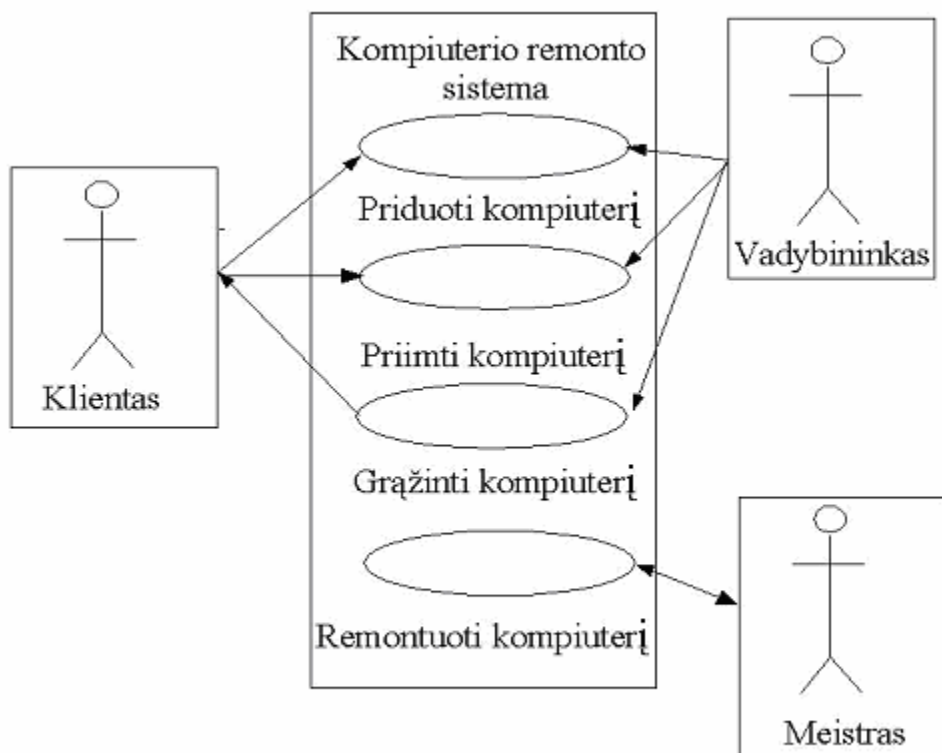


4 pav. Nulinio lygio duomenų srautų diagrama

Duomenų srautų lentelė:

1	Remonto priėmimas
2	Remonto atidavimas
3	Kompiuterio perdavimas į remontą
4	Kompiuterio gražinimas iš remonto
5	Priėmimo/išdavimo valdymas
6	Sandėlio valdymas
7	Remonto valdymas
8	Medžiagų išdavimas

3.3. USE CASE diagrama



5 pav. USE CASE diagrama

4. Technologinių platformų analizė

Pagrindinė užduotis: Kuriamos programinės įrangos technologinių platformų analizė.

Kad sukurti projektą, reikia įvertinti jau esančių sistemų bei programinių įrangų parametrus ir galimybes. Toliau nagrinėjami kelios operacinės sistemos, duomenų bazės, programavimo kalbos, programinės įrangos veikimo modeliai leidžia pasirinkti labiausiai poreikius atitinkantį sprendimą. Atsižvelgiama į privalumus, trūkumus, kliento pageidavimus, integraciją tarpusavyje.

4.1. Operacinės sistemos

- *Microsoft Windows 2003 Standart Edition.*

Gamintojas – kompanija „Microsoft“.

„Windows Server 2003 Standart Editon yra patikima tinklo operacinė sistema, greitai ir nesudėtingai pateikianti verslo sprendimus. Ši lanksti tinklo operacinė sistema - idealus pasirinkimas mažų įmonių poreikiams ir žinybinėms reikmėms tenkinti.“

Pagrindiniai šios operacinės sistemos trūkumai yra šie:

- a) Nemaži reikalavimai pačio serverio techninei įrangai;
- b) Didelė kaina.

Pagrindiniai privalumai:

- a) Nesudėtingas diegimas;
- b) Didelis komercinis techninis palaikymas.

- *Debian GNU/Linux.*

Gamintojo nėra.

Tai operacinė sistema, kuriama savanoriškai virš 900 žmonių pasaulyje. Debian - laisva operacinė sistema, kuri naudoja Linux branduolį.

Pagrindiniai trūkumai:

- a) Instaliavimo sudėtingumas;
- b) Sutvarkymo sudėtingumas.

Pagrindiniai privalumai:

- a) Nededeli diegimo ir palaikymo kaštai;
- b) Platus pasirinkimas įvairios programinės įrangos, leidžiančios lengviau pasiekti norimą rezultatą;
- c) Gerokai didesnės testavimo galimybės, esančių klaidų fiksavimas ir stebėjimas;
- d) Pati operacinė sistema ir jos palaikymas yra nemokami.

4.2. Duomenų bazės

- *Microsoft Access duomenų bazė:*

Gamintojas – kompanija Microsoft. Tai duomenų bazė, standartiškai įeinanti į Microsoft Office programų paketą. Ji leidžia sukurti galingas, lengvai susiejamas duomenų bazines, kurios lengvai integruojamos į Internetą ir įmonės duomenų struktūrą.

Pagrindiniai privalumai:

- a) Aiškus ir lengvas darbas;
- b) Lengva integracija.

Pagrindiniai trūkumai:

- a) Galima naudoti tik Microsoft Windows platformoje;
- b) Sunku integruoti su ne Microsoft korporacijos produktais;
- c) Didelė kaina.

- *Microsoft SQL duomenų bazė:*

Gamintojas – kompanija Microsoft. Tai duomenų bazė, skirta įvairiems vartotojų poreikiams. Paprastai yra naudojama kai reikia saugoti didelius informacijos kiekius, reikalingas stabilumas ir sparta.

Pagrindiniai privalumai:

- a) Didelės galimybės;
- b) Stabilumas, saugumas ;
- c) Galimybė dirbti per kelis kompiuterius.

Pagrindiniai trūkumai:

- a) Per didelė mažiems projektams;
- b) Labai didelė kaina.

- *MySQL duomenų bazė:*

Gamintojas – kompanija MySQL AB. Tai duomenų bazė skirta įvairiems vartotojams. Ji gali būti naudojama mažuose ir vidutiniuose įmonių projektuose.

Pagrindiniai privalumai:

- a) Didelės galimybės;
- b) Stabilumas, saugumas;
- c) Nemokamas;
- d) Įeina į LAMP sudėtį.

Pagrindiniai trūkumai:

- a) Ne visos standartinės duomenų bazių funkcijos yra palaikomos;

- *PostgreSQL duomenų bazė:*

Gamintojo nėra. Ją kuria ir palaiko grupė žmonių. Pagrindinis centras Kanadoje.

Duomenų bazė, skirta visiems atvejams. Ji turi visas reikalingas funkcijas, yra greita, stabili.

Pagrindiniai privalumai:

- a) Didelės galimybės;
- b) Stabilumas, saugumas;
- c) Nemokama.

Pagrindiniai trūkumai:

- a) Itin dideliems projektams trūksta papildomų funkcijų, kurias turi kitos duomenų bazės;
- b) Lėta.

- *Oracle duomenų bazė:*

Gamintojas – Oracle korporacija. Reliacinių duomenų bazių lyderis. Turi visas reikalingas funkcijas.

Į ją orientuojasi dauguma kitų reliacinių duomenų bazių.

Pagrindiniai privalumai:

- a) Palaikomos visos reikalingos funkcijos;
- b) Didelės galimybės;
- c) Greita dirbant su dideliais duomenų kiekiais.

Pagrindiniai trūkumai:

- a) Per didelė nedideliems projektams;
- b) Didelė kaina.

4.3. Programavimo kalbos

- *PHP programavimo kalba:*

PHP yra plačiai naudojami hyper teksto pirminio apdorojimo programavimo kalba. Ji yra optimizuota WWW tinklapių kūrimui. Šią kalbą kuria ir palaiko grupė žmonių.

Pagrindiniai privalumai:

- a) Labai lengvas kodas;
- b) Lengvai įterpiamas į jau egzistuojanti HTML kodą;
- c) Veikia daugelyje operacinių sistemų;
- d) Nemokama;
- e) Įeina į LAMP sudėtį.

Pagrindiniai trūkumai:

- a) Nepilnai palaiko objektinį programavimo metodą;

b) Dėl negriežtos sintaksės labai lengva net ir veikiančioje sistemoje padaryti daug klaidų.

- *Perl programavimo kalba:*

Perl yra plačiai naudojama kalba skirta tekstams apdoroti. Pagrindinis funkcinis elementas – reguliariosios išraiškos. Dėl didelio populiarumo yra sukurta daugybė išplėtimų. Viena iš pagrindinių kalbų skirtų operacinių sistemų scenarijams rašyti. Galima naudoti HTTP puslapiams generuoti, tekstui tvarkyti ir grafinėms programoms kurti. Ją kuria ir palaiko grupė žmonių.

Pagrindiniai privalumai:

- a) Labai populiarūs ir turi daug išplėtimų;
- b) Labai efektyvi;
- c) Greita ir saugi;
- d) Nemokama;
- e) Įeina į LAMP sudėtį;

Pagrindiniai minusai:

- a) Sudėtinga sintaksė;
- b) Reguliariųjų išraiškų rašymas yra labai sudėtingas.

- *ASP programavimo kalba:*

Tai Microsoft kompanijos gaminys. Ji kaip ir PHP programavimo kalba yra optimizuota WWW puslapių kūrimui. Tai jos pagrindinė panaudojimo sritis.

Pagrindiniai privalumai:

- a) Lengvai integruojasi į Microsoft Windows operacinę sistemą;
- b) Optimizuota ir greitai dirbant su Microsoft kompanijos produktais;
- c) Lengva sintaksė.

Pagrindiniai trūkumai:

- a) Normaliai veikia tik Microsoft Windows aplinkoje;
- b) Sunku arba neįmanoma panaudoti dirbant su ne Microsoft kompanijos gaminiais.

- *Python programavimo kalba:*

Python kaip ir Perl yra universali programavimo kalba, kurią kuria ir palaiko grupė žmonių. Ji veikia daugelyje operacinių sistemų ir yra objektinė programavimo kalba.

Pagrindiniai privalumai:

- a) Universalumas;
- b) Veikimas daugelyje operacinių sistemų;
- c) Sintaksės aiškumas;
- d) Turi daug išplėtimų.

Pagrindiniai trūkumai:

- a) Kodo matomumas. Nėra galimybių kodą sukompiliuoti, tik interpretuoti;
- b) Sudėtingas kodas naudojant tik tinklapių aplikacijose.

4.4. HTTP serveriai

- *IIS http serveris:*

Interneto informacijos serveris. Microsoft kompanijos gaminys, skirtas Microsoft Windows operacinei sistemai.

Pagrindiniai privalumai:

- a) Integruotas į Microsoft Windows operacinę sistemą;
- b) Lengvas ir intuityvus valdymas.

Pagrindiniai trūkumai:

- a) Veikia tik Microsoft Windows operacinėje sistemoje;

- *Apache http serveris:*

Interneto informacijos serveris. Jį kuria ir palaiko žmonės, oficialiai priklauso Apache programinės įrangos fondui. Labiausia pasaulyje paplitęs http serveris (2003 metų gruodžio mėnesį kompanijos Netcraft duomenimis Apache naudojo 67.43% visų http serverių (12).

Pagrindiniai privalumai:

- a) Veikia daugelyje operacinių sistemų;
- b) Pripažintas pasaulyje;
- c) Nemokamas;
- d) Stabilus ir greitas;
- e) Įeina į LAMP sudėtį.

Pagrindiniai trūkumai:

- a) Sudėtingas tvarkymas.

Apžvelgus ir įvertinus technologijas buvo pasirinkta LAMP platforma pačios informacinės sistemos sukūrimu ir tolesniam veikimui. Ši platforma pasirinkta dėl šių priežasčių:

- Platformos programinė įranga nieko nekainuoja;
- PC reikalavimai yra mažesni lyginant su Microsoft Windows operacine sistema;
- Pasirinktos platformos komponentai turi daugiau privalumų negu trūkumų;
- Jei leidžia galimybės, užsakovas prašė viską integruoti į jau veikiančią elektroninio pašto serverį;
- Platformos vientisumas užtikrina nepriekaištingą suderinamumą tarpusavyje.

Toliau aptarsime pačios sistemos projektą, funkcinius ir nefunkcinius reikalavimus, duomenų struktūrą.

5. Programinės įrangos projektas

Pagrindinė užduotis: Kuriamos programinės įrangos projektas pasirinktai platformai.

Pagal reikalavimus sukurti programinės įrangos projektą tenkinantį visus reikalavimus.

5.1. Reikalavimai sistemai

- Duomenys įvedami ranka, naudojantis specialiomis formomis.
- Duomenys išvedami formose ir į spausdintuvą.
- Programa turi naudoti duomenų bazę informacijos saugojimui.
- Programa turi būti saugi.
- Programa turi turėti kelis vartotojų galimybių lygius.
- Programa turi turėti administravimo interfeisą vartotojams ir sistemai tvarkyti.
- Programa turi naudoti mažai operacinės sistemos resursų.
- Programa turi naudoti mažai vidinio įmonės tinklo resursų.
- Programa turi būti kliento-serverio architektūros.
- Interfeisas turi būti aiškus ir patogus.

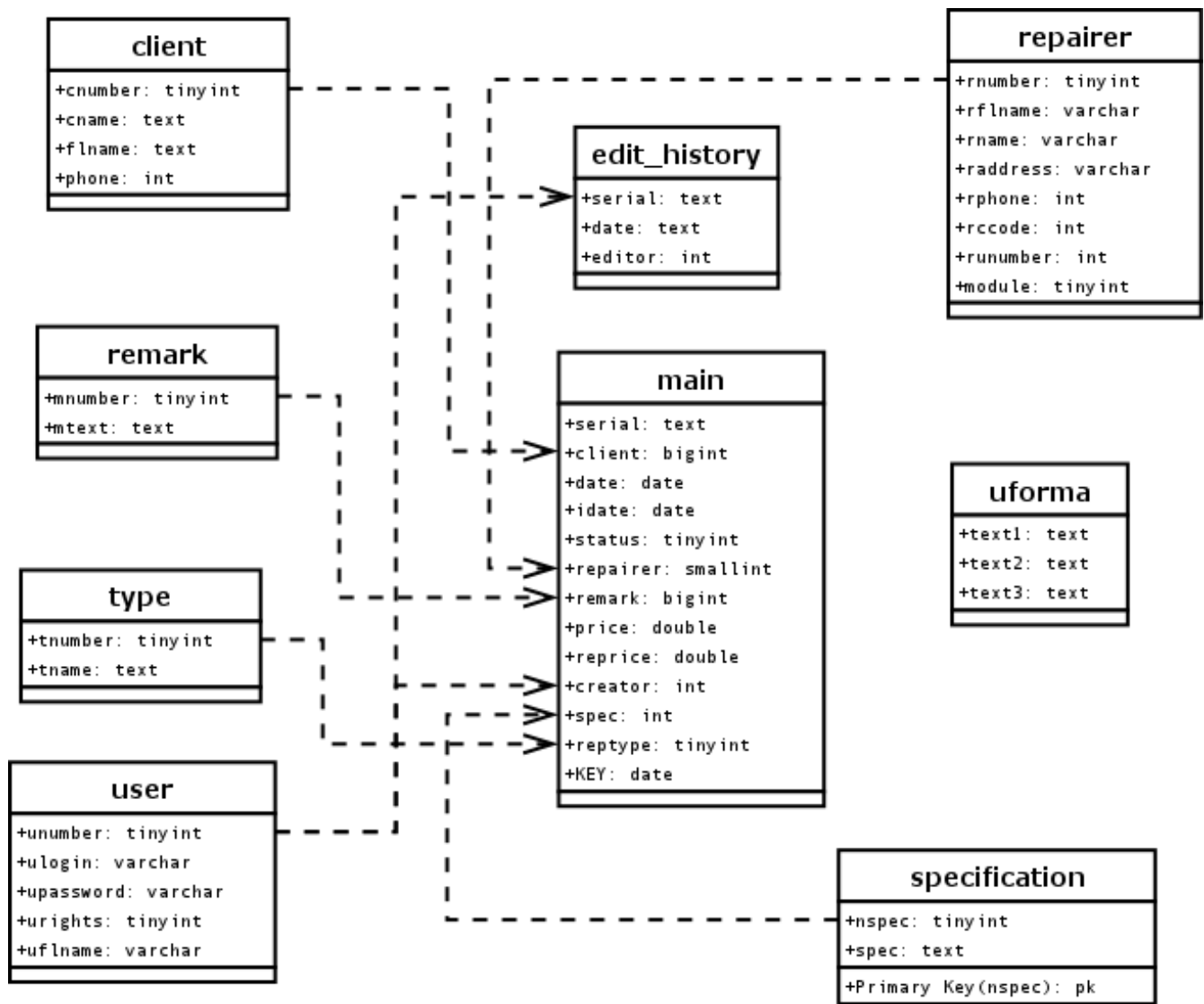
5.2. Technologinis sprendimas

Pagal reikalavimus, atlikus technologinių platformų analizę ir įvertinus užsakovo pageidavimus buvo pasirinkta LAMP platforma . Ši platforma susideda iš Linux operacinės sistemos, Apache http serverio, MySQL duomenų bazės ir PHP programavimo kalbos. Šioje platformoje PHP programavimo kalba leidžia visus komponentus apjungti tarpusavyje ir garantuoja sistemos veikimo patikimumą.

5.3. Duomenų struktūra

Duomenų struktūra buvo projektuojama phpmyadmin programos pagalba. Programa phpmyadmin suteikia grafinį MySQL duomenų bazės valdymo interfeisą. Duomenų bazės struktūra priede.

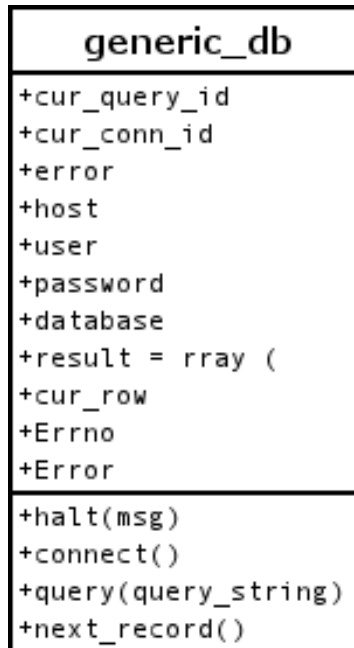
Žemiau parodyta duomenų bazės diagrama:



6 pav. Duomenų struktūra

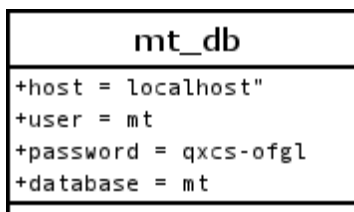
6. Programinės įrangos UML diagramos

Programinė įranga buvo sumodeliuota naudojantis UML. Žemiau pateiktoje diagramoje matosi duomenų bazės abstrakcijos klasė. Ji aprašo duomenų bazės funkcijų prototipus, kurie vėliau bus naudojami programinės įrangos veikime.



7 pav. Abstrakti duomenų bazės klasė

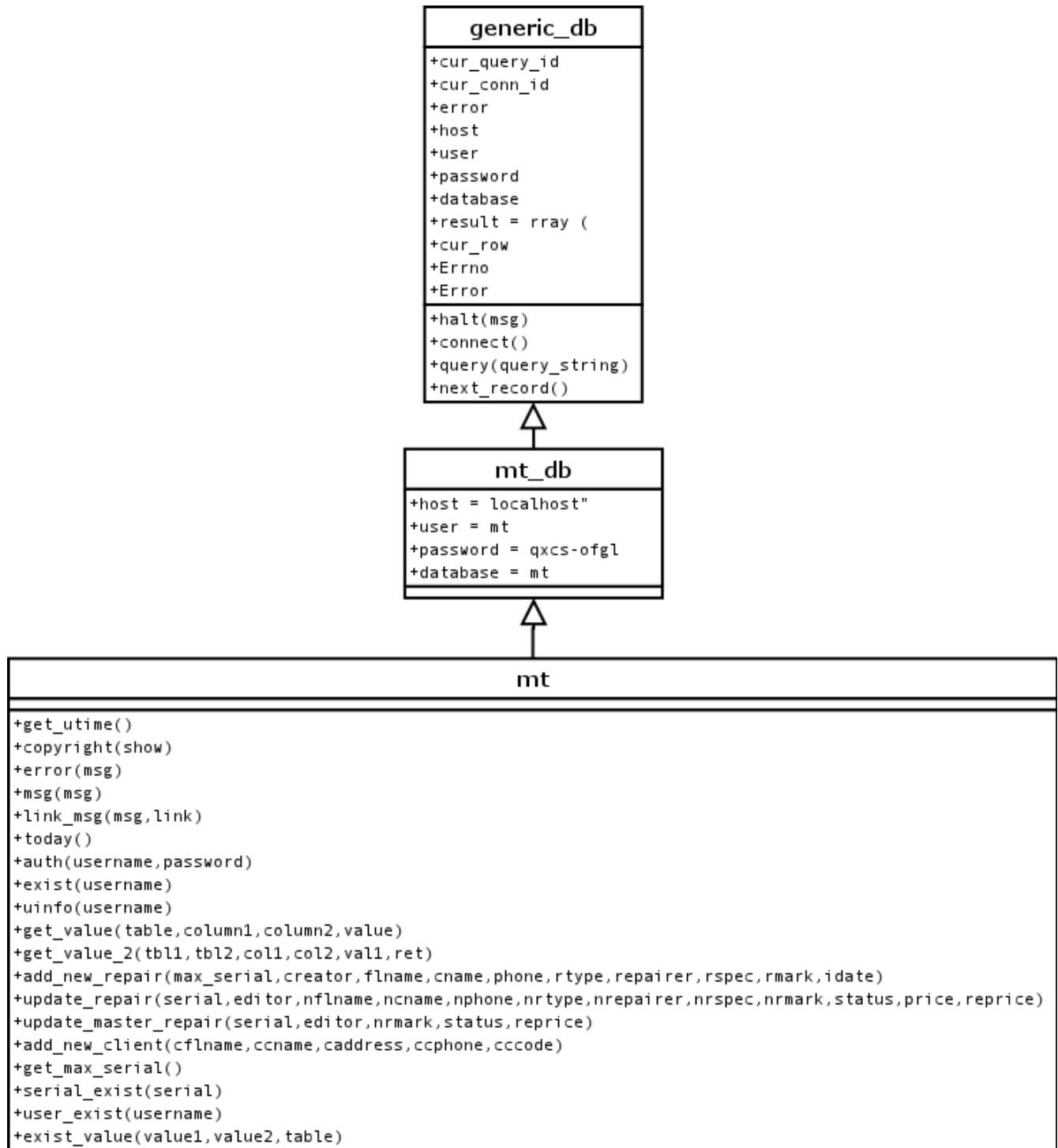
Šioje diagramoje parodyta vaiko klasė, kuri modifikuoja tėvo klasės prototipų funkcijas:



8 pav. Abstrakčios duomenų bazės klasės vaikas

Šios dvi klasės programinėje įrangoje naudojamos valdyti darbą su MySQL duomenų baze. Tai labai palengvina darbą rašant kodą pačioje programoje, kadangi nebereikia dirbti tiesiogiai su pačia baze. Yra naudojami tik db klasės metodai `halt(msg)`, `connect()`, `query(query_string)` ir `next_record()`. Jie atlieka prisijungimą prie duomenų bazės, tiesioginį kreipimąsi, rezultatų išvedimą.

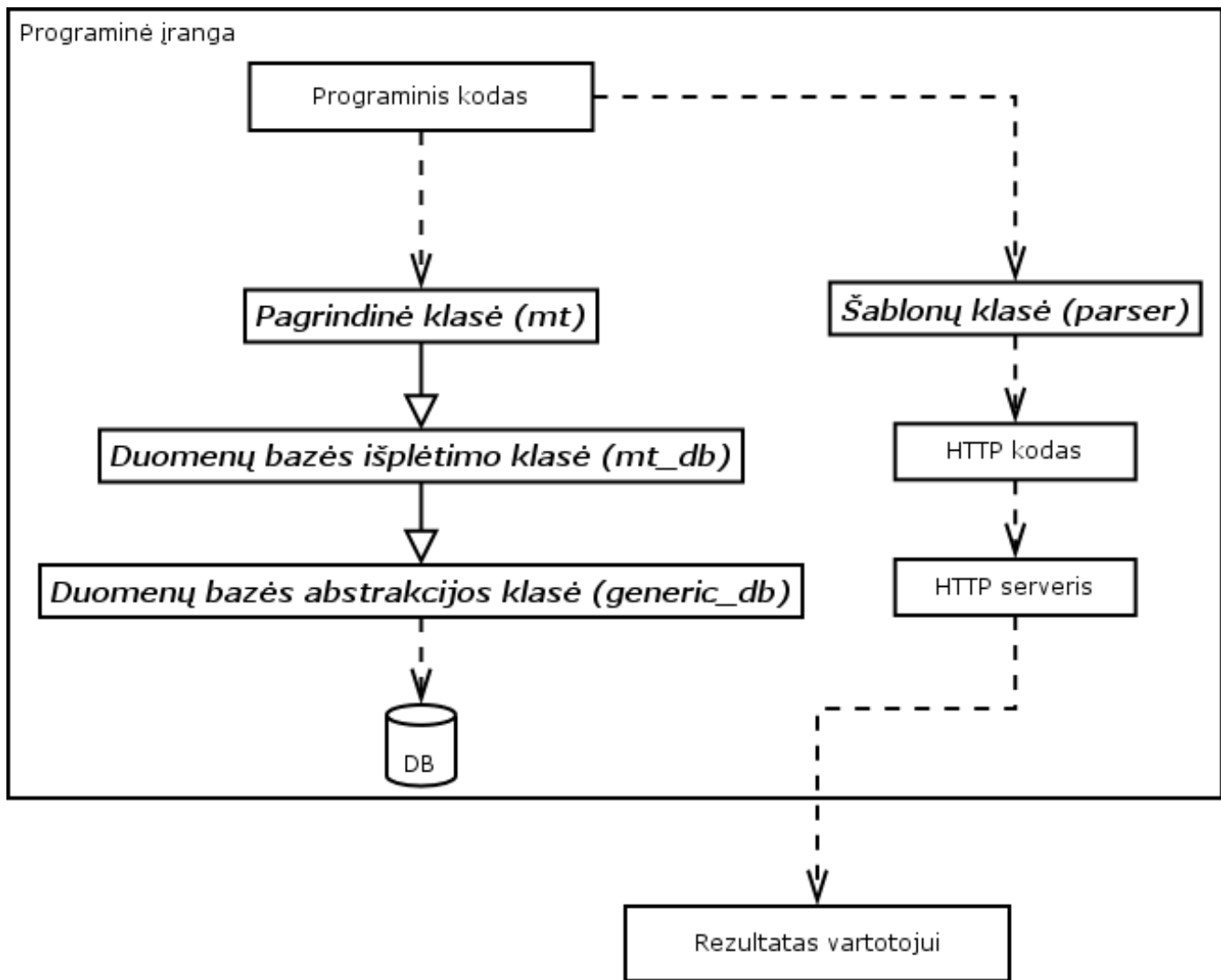
Kitame modelyje yra parodytos tėvo (db), vaiko (db_mt) ir dar vieno vaiko (mt) klasių ryšiai tarpusavyje. Paskutinė klasė yra kartu ir pagrindinė. Ji atlieka pagrindinį darbą visoje sistemoje. Jos metodai yra specifiniai, skirti konkrečioms veiksmams atlikti ir universalūs, leidžiantys atlikti universalesnius veiksmus (gauti rezultatą iš bet kurios duomenų bazės lentelės ir panašiai).



9 pav. Klasių ryšiai ir metodai

Papildomai tobulinant sistemą, siekiant ją padaryti universalesnę, buvo įvesta speciali klasė, skirta HTML kodo apdirbimui. Ši klasė vadinama šablonu ir leidžia programinį kodą atskirti nuo HTML kodo. Tai atlikus programinės įrangos ir visos sistemos išvaizda tapo nepriklausoma nuo programinio kodo. Ja pasidarė lengviau interpretuoti ir integruoti į egzistuojančius sprendimus. Tai žymiai pagerina sukurto kodo antrinį panaudojimą ir leidžia pačiai programinei įrangai nepriklausyti nuo išvaizdos. Taip pasiektas programinės įrangos kodo antrinis panaudojimas (reusability).

Žemiau pateikiama vidinė programinės įrangos struktūra ir ryšiai tarp klasių. Šioje diagramoje labai aiškiai matosi kaip HTML kodas atsiskiria nuo programinio kodo ir kaip šablonų klasės pagalba rezultatas išvedamas vartotojui:

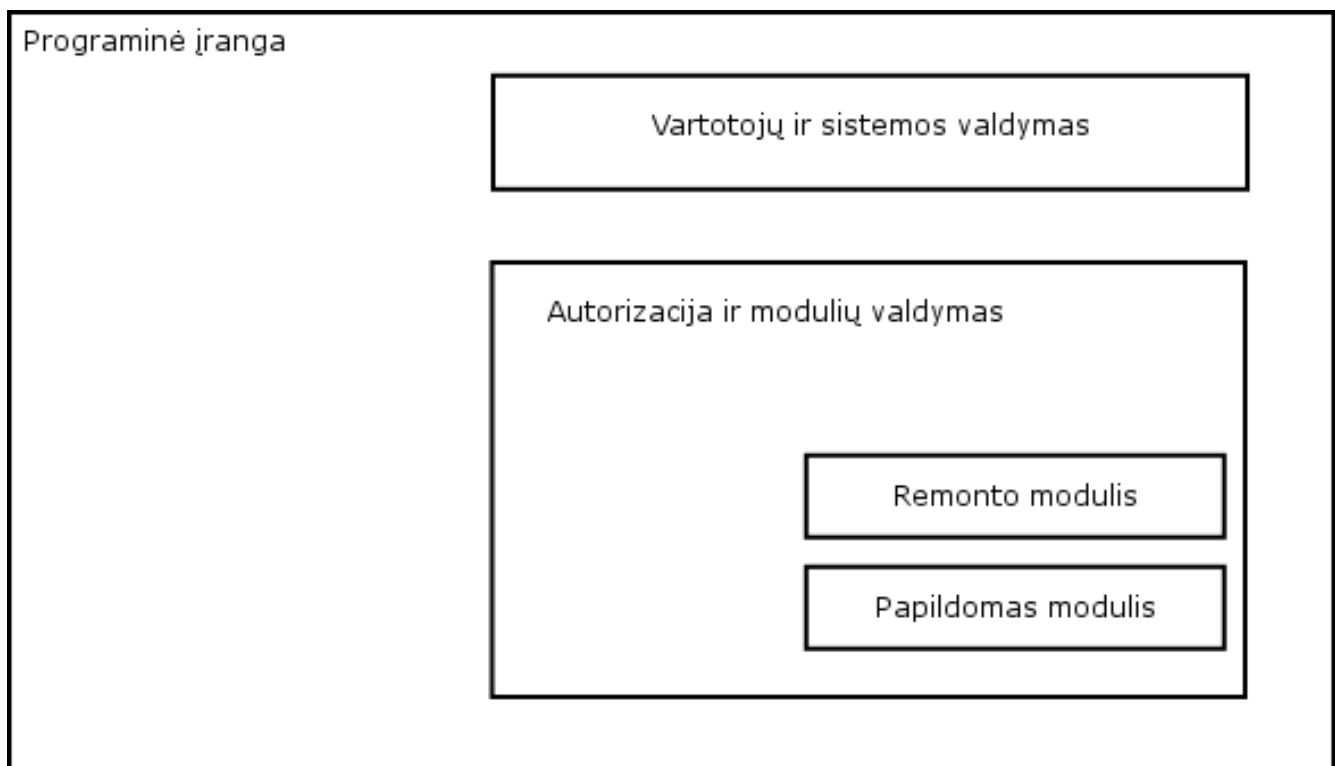


10 pav. Klasių sąveika programinėje įrangoje

Programinė įranga skaidoma į tokias dalis:

- Autorizacijos arba pagrindinė;
- Vartotojų ir sistemos valdymas;
- Moduliai.

Ši struktūra garantuoja gerą sistemos išplėtimą. Modulinė struktūra leidžia lengvai pridėti papildomus modulius, kurie atliktų kitas funkcijas. Šiuo atveju išlieka globali vartotojų, klientų duomenų bazė, kuri leidžia dar labiau pagerinti darbo kokybę. Diagramoje pateiktoje žemiau, puikiai matosi kaip modulinė struktūra leidžia prailginti sistemos gyvavimo trukmę, tuo padidindama universalumą ir ilgaamžiškumą:



11 pav. Modulinė programinės įrangos struktūra

Suprojektavus sistemą, buvo sukurtos vartotojo formos. Visas formas galima suskirstyti į keletą grupių, kurios apibrėžia arba atliekamą funkciją arba priklausomybę kažkokiai didesnei formų grupei.

- Pagrindinių formų grupė:
 1. Prisijungimo forma;
 2. Meniu forma;
 3. Slaptažodžio keitimo forma.
- Administravimo formų grupė:

1. Administravimo meniu forma;
 2. Vartotojų administravimo forma;
 3. Remonto meistrų administravimo forma;
 4. Modulio administravimo forma.
- Modulių formų grupė:
 - Remonto modulis:
 - Pagrindinė forma;
 - Redagavimo forma;
 - Naujo remonto forma;
 - Pastabų forma;
 - Kliento informacijos forma.

Papildomos funkcijos, skirtos sistemos funkcionavimui pagerinti yra išvardintos žemiau. Jos paremtos konkrečios panaudotos programavimo kalbos galimybėmis.

- Sesijų panaudojimas. Sesijos konkrečioje sistemoje skirtos duomenų saugojimui. Jos saugomos serverio o ne vartotojo pusėje, tuo padidindamos bendrą sistemos saugumą.
- Visi kintamieji patenkantys iš formų yra tikrinami kliento pusėje ir pačioje sistemoje.
- Kelių lygių apsaugos sistema. Apsauga šioje sistemoje vykdoma sesijų pagalba. Kiekvienam prisijungiančiam vartotojui generuojamas sesijos numeris su tam tikru terminuotu galiojimo laiku. Pagal šį sesijos numerį tikrinamos vartotojo teisės kreipiantis į bet kurią autorizacijos reikalaujančią programinės įrangos funkciją. Esant nesankcionuotiems kreipimams, sesija naikinama, o vartotojas siunčiamas į pradinę prisijungimo formą, reikalaujant atlikti autorizaciją iš naujo.
- Visiškam slaptažodžių saugumui garantuoti rekomenduojama naudoti SSL papildymą Apache HTTP serveryje.

7. Sistemos aprašymas

Ši sistema yra skirta kompiuterizuoti personalinių kompiuterių ir jų komponentų remonto apskaitą. Ji yra modulinės struktūros. Išskiriamos trys vartotojų grupės:

- Sistemos tvarkytojai;
- Paprasti vartotojai;
- Remonto meistrai.

Sistema veikia priklausomai nuo prisijungusio vartotojo: kiekvienas vartotojas gali atlikti tik jam skirtas funkcijas. Skiriasi ir vartotojų meniu.

Esant poreikiui sistemą galima papildyti reikalingais papildomais moduliais. Tai leistų panaudoti klientų duomenų bazę, vartotojų duomenų bazę, įvairias formas.

Saugumui užtikrinti naudojamos įvairios priemonės:

- Įvedamų duomenų tikrinimas;
- Perduodamų duomenų šifravimas;
- Sesijų saugojimas serveryje.

8. Sistemos įdiegimas

Sistemos įdiegimas susideda iš dviejų pagrindinių dalių:

- LAMP platformos įdiegimo;
- Programinės įrangos įdiegimo.

8.1. LAMP platformos įdiegimas

Reikalavimai prieš instaliavimą:

- Intel x86 architektūros kompiuteris, turintis galimybę startuoti iš CD-ROM;
- Debian GNU/Linux 2.2 (Potato) operacinės sistemos kompaktinis diskas;
- Internet prieiga;
- Patirtis dirbant su Linux operacinių sistemomis.

Diegimas:

CD diskas įdedamas į CD-ROM įrenginį. Kompiuteris nustatomas kad OS krovimasis vyktų iš CD-ROM. Sistema kraunama. Pačioje pradžioje, tik užsikrovus "boot loader" LILO, bus parodytas Debian GNU/Linux sveikinimas:

Software in the Public Interest presents

Debian GNU/Linux 2.2

Keletą kartų paspaudus "Enter" klavišą, jūsų paklaus apie naudojamą klaviatūrą. Pasirenkama naudojama klaviatūra. Kitas žingsnis - labai svarbus. Tai kieto disko skirsnių (partitions) sukūrimas. Tam Debian GNU/Linux galima naudoti standartiškai siūlomą priemonę cfdisk. cfdisk pakankamai paprasta ir aiški vartotojui programa. Štai kas patariama dėl diskų skirsnių:

- a) Būtinai reikia pasidaryti nedidelį (10-15Mb) skirsnį, kurį vėliau reikia prijungti kaip /boot. Tai padės išvengti galimų nesusipratimų su LILO;
- b) swap partcijai (swap - atmintis kietame diske, kurią savo darbui naudoja Linux) rekomenduojama skirti 1-2 kartus didesnį kiekį negu RAM kiekis kompiuteryje. Pvz.: RAM kiekis yra 128MB, swap dydis - 256MB.
- c) Likusią laisvą vietą galima skirti pagrindiniam skirsniui (root arba /).

Sukūrus skirsnius kietame diske, aktyvuojama swap skirsnis (jei jį sukūrėte). Vėliau pasiūloma suaktyvuoti Linux skirsnį. Čia reikia pasirinkti ta skirsnį, kuris skirtas root arba / . Tai atlikus ir

prijungus skirsnį kaip /, pasirenkama alternatyvus punktas: dar kartą "Initialize Linux partition". Tai leis suaktyvuoti ir prijungti /boot skirsnį.

Kai įdiegimo programa jums pasiūlys išsirinkti reikalingus modulius, kuriuos sistema pasistengs užkrauti, reikia pasirinkti tinklo plokštės tvarkyklę. Tai atlikus sėkmingai, reikės įvesti tinklo parametrus. Kitas žingsnis - pagrindinės ("Base") sistemos įdiegimas. Reikia pasirinkti instaliavimą iš CD-ROM įrengimo. Paklausus apie LILO įdiegimo vietą reikia pasirinkti į pagrindinio disko boot sektorių. Paskutinis žingsnis - kompiuterio perkrovimas. Jei viskas vyko sėkmingai, po perkrovimo jūsų kompiuteris turėtų krauti Debian GNU/Linux operacinę sistemą.

Pasikrovus sistemai, prisijungiama root vartotoju. Tai padarius, redaguojamas failas /etc/apt/sources.list arba pasileidžiama programą apt-setup.

a) rašoma: vi /etc/apt/sources.list ir redaguojama, kad failas turėtų šias eilutes:

```
deb http://ameba.sc-uni.ktu.lt/debian/ stable main non-free contrib
```

```
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
```

```
deb http://security.debian.org testing/updates stable contrib non-free
```

Išsaugoma ir išeinama iš failo.

b) rašoma: *apt-setup*

Atsiradusiame lange pasirenkamas serveris ameba.sc-uni.ktu.lt kaip lokalus Debian veidrodis.

Pabaigus vieną iš šių variantų, vykdomos komandos *apt-get update && apt-get dist-upgrade*.

Programos parsiūs reikalingus failus operacijų sistemos atnaujinimui ir juos sutvarkys. Į galimus klausimus reikia savo nuožiūra.

Viską atlikus, renkama tokią komandą:

```
apt-get install apache-ssl php4 php4-pear mysql-server mysql-client mysql-common .
```

Pasibaigus šiam etapui, serveris bus su visa reikalinga programine įranga. Reikia tik ją sutvarkyti. Tam reikia redaguoti failą /etc/apache-ssl/httpd.conf : prie DirectoryIndex opcijoje esančių indeksinių failų įrašoma index.php

Toliau nustatomas MySQL duomenų bazės pagrindinis slaptažodis. Tai atliekama su komanda *mysqladmin*. Rašoma: *mysqladmin password* slaptažodis (slaptažodis – jūsų sugalvotas).

8.2. Programinės įrangos įdiegimas

Pabaigus serverio diegimo darbus, reikia įdiegti pačią programinę įrangą. Tai daroma taip: einama į katalogą /var/www. Tam atlikti rašoma: *cd /var/www*.

Tada parsiunčiamas programinės įrangos archyvas. Parsiuntus programinės įrangos archyvą, reikia jį išskleisti. Tam atlikti rašoma: *tar xzvf archyvas.tar.gz*. Pasibaigus išskleidimui, ją sutvarkote.

Tvarkymas prasideda nuo failų ir direktorijų išdėstymo. Svarbiausia, kad direktorija common būtų nepasiekiamas per www. Todėl ji nukeliama viena direktorija žemiau.

Nukėlus, einama į ten ir redaguojamas failas db.php. Ten suvedama duomenų bazės pavadinimas, vartotojas, slaptažodis ir kompiuteris kuriame ji veikia (šiuo atveju localhost – lokalus kompiuteris). Išsaugojus redaguojamas failą db.sql. Ten į pirmą eilutę įrašoma ta pati duomenų bazė. Į antrą eilutę įrašoma duomenų bazė, vartotojas ir slaptažodis. Tokius pačius kaip ir db.php faile. Išsaugojus šį failą, informacija įkeliamas į duomenų bazę. Tam atlikti rašoma: `mysql -u root -p < db.sql`. Įvedamas root vartotojo slaptažodis, nustatytas su mysqladmin. Jei nebuvo gauta pranešimų apie klaidas, vadinasi duomenų bazė ir jos vartotojas sukurti sėkmingai. Dabar reikia sukurti duomenų bazės struktūrą. Tam atlikti rašoma: `mysql -u vartotojas -p < mt.sql`. Čia įvedamas slaptažodis, įrašytas db.sql faile.

Po šių veiksmų programinė įranga yra įdiegta serveryje. Pradinis tvarkymo vartotojo vardas admin, pradinis slaptažodis secret. Juos vėliau būtinai reikia pasikeisti. Toliau pradeda dirbti su pačia sistema.

9. Darbas su sistema

Darbas sistemoje susideda iš trijų etapų. Pirmas etapas – darbas kaip sistemos tvarkytojas. Antras etapas – darbas kaip paprastas vartotojas. Trečias – darbas kaip remonto meistras. Pirmiausiai aprašysima kaip dirbti sistemos tvarkymo režime.

Reikalavimai:

- Įdiegta programinė įranga;
- Sistemos tvarkymo vartotojo informacija (prisijungimo vardas ir slaptažodis);
- Personalinis kompiuteris su Internet naršykle.

Naudojantis personaliniu kompiuteriu ir Internet naršykle užkraunamas pradinis puslapis. Tai turėtų būti serverio adresas. Užsikrovus matomas toks langas:



The image shows a web form titled "Login". It has a light blue header with the word "Login" in bold. Below the header, there are two input fields: "Username" and "Password". The "Username" field is on top, and the "Password" field is below it. Both fields have a light blue border and a small blue icon on the left. Below the "Password" field, there is a "Login" button with a light blue background and a white border.

12 pav. Autorizacijos langas

Tai autorizacijos langas. Įvedus vartotojo informaciją, matomas pagrindinis sistemos meniu, kuris priklausomai nuo vartotojo gali būti sistemos tvarkytojo meniu, paprasto vartotojo meniu arba remonto meistro meniu.

9.1. Sistemos administravimas

Žemiau aprašomas sistemos tvarkytojo darbas. Jo pagrindinis meniu:






The image shows a web page titled "Menu". It has a light blue header with the word "Menu" in bold. Below the header, there are four links: "Bazės vartotojai", "Remonto meistras", "Formos redagavimas", and "Atsijungti". The links are in blue text and are underlined. Below the links, there is a copyright notice: "© Rolandas Juodžbalis 2002-2003".

13 pav. Sistemos administratoriaus pagrindinis meniu

Esančiame meniu galima rinktis norimą tvarkymo punktą:

- Bazės vartotojai;
- Remonto meistrai;
- Formos redagavimas;

Pasirinkus Bazės vartotojų tvarkymo punktą, matomas visų sistemos vartotojų sąrašas:

Bazės vartotojai					
Naujas bazės vartotojas Atgal Atsijungti					
Numeris	Loginas	Teisės	Slaptažodis	Redaguoti	
33	user	Skaitymas/rašymas	Keisti slaptažodį		
31	admin	Administravimas	Keisti slaptažodį		

14 pav. Bazės vartotojų pagrindinis meniu

Šioje valdymo formoje galima atlikti visus sistemos vartotojų tvarkymo veiksmus:

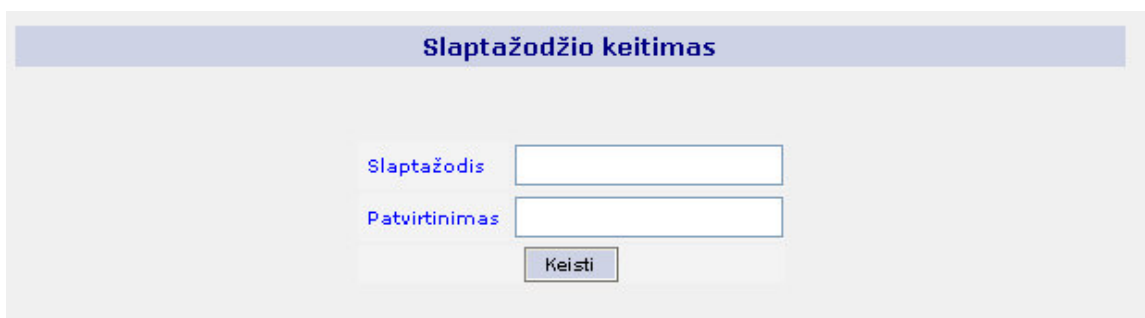
- Pakeisti teises;
- Pakeisti slaptažodį;
- Pašalinti;
- Įvesti naują.

Vartotojo teisės keičiamos naudojantis redagavimo forma:

Vartotojo redagavimas	
Loginas	<input type="text" value="user"/>
Teisės	<input type="text" value="Skaitymas/rašymas"/> ▼
<input type="button" value="Keisti"/>	

15 pav. Vartotojo teisių redagavimo forma

Vartotojo slaptažodis keičiamas naudojantis slaptažodžių keitimo forma:



The screenshot shows a web form titled "Slaptažodžio keitimas" (Password Change). It contains two input fields: "Slaptažodis" (Password) and "Patvirtinimas" (Confirmation). Below the fields is a button labeled "Keisti" (Change).

16 pav. Slaptažodžio keitimo forma

Vartotojas šalinamas paspaudus šiukšlių dėžės mygtuką pagrindinėje formoje.



17 pav. Šiukšlių dėžės mygtukas

Naujas vartotojas įvedamas naudojantis naujo vartotojo įvedimo forma:



The screenshot shows a web form titled "Naujas bazės vartotojas" (New Database User). It contains four input fields: "Loginas" (Username), "Slaptažodis" (Password), "Patvirtinimas" (Confirmation), and "Teisės" (Rights). The "Teisės" field is a dropdown menu with "Išjungtas" (Disabled) selected. Below the fields is a button labeled "Įvesti" (Add).

18 pav. Naujo bazės vartotojo įvedimo forma

Pasirinkus remonto meistrų tvarkymo punktą, matomas visų sistemos remonto meistrų sąrašas:

Remonto meistrai									
Naujas meistras Atgal Atsijungti									
Numeris	Loginas	Vardas Pavardė	Pavadinimas	Adresas	Telefonas	Įm. kodas	Slaptažodis	Redaguoti	
13	meistras1	Meistras Vienas			123		Keisti		

19 pav. Remonto meistrų sąrašas

Šioje valdymo formoje galimas atlikti visus remonto meistrų tvarkymo veiksmus:

- Pakeisti slaptažodį;
- Pašalinti;
- Įvesti naują.

Remonto meistro slaptažodis keičiamas naudojantis slaptažodžių keitimo forma:

Slaptažodžio keitimas	
Slaptažodis	<input type="text"/>
Patvirtinimas	<input type="text"/>
<input type="button" value="Keisti"/>	

20 pav. Slaptažodžio keitimo forma

Remonto meistras šalinamas paspaudus šiukšlių dėžės mygtuką pagrindinėje formoje.



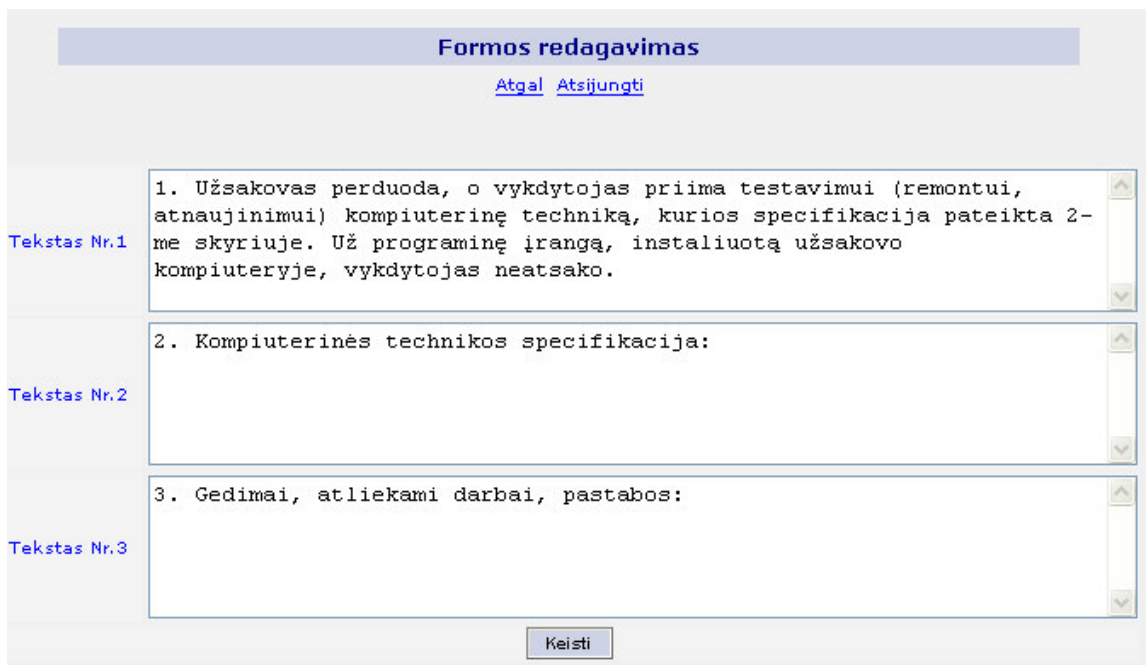
21 pav. Šiukšlių dėžės mygtukas

Naujas remonto meistras įvedamas naudojantis naujo remonto meistro įvedimo forma:



22 pav. Naujo remonto meistro įvedimo forma

Pasirinkus formos redagavimo punktą, galima redaguoti remonto modulio tekstus, kurie yra naudojami naujo remonto įvedimo formoje:



23 pav. Remonto modulio redagavimo forma

9.2. Vartotojo darbas

Toliau aprašomas paprasto vartotojo darbas. Jo pagrindinis meniu po prisijungimo:



24 pav. Paprasto vartotojo pagrindinis meniu

Čia matomas sistemoje esančių modulių sąrašas, slaptažodžio keitimo forma, atsijungimo forma ir veikimo trukmės laikas. Pagal veikimo trukmės laiką galima spręsti ar sistema apkrauta ar ne. Pasirinkus modulį „Komponenčių remontas“ patenkama į remonto modulį, kurio pagrindinis meniu yra toks:

Menu

[Atgal](#) [Logout](#)

[Naujas remontas](#) [Rodyti viską](#)

Kodas	Tvarko	Būsena	Priimta	Kaina	Rem Kaina	Klientas	Pastaba
020424-1	Meistras Vienas	Sutvarkyta	0000-00-00	300	230	Petras	Uporeidas I P4-

Runtime: 0.016s

25 pav. Pagrindinis remonto modulio menu

Čia matoma informacija apie visus duomenų bazės įrašus, kurių būsena kitokia negu „Atiduota ir apmokėta“. Būsenos leidžia nuspręsti kokia yra remonto eiga. „Atiduota ir apmokėta“ būsena reiškia kad remontas jau baigtas ir klientas už jį atsiskaitė. Kitos būsenos yra šios:

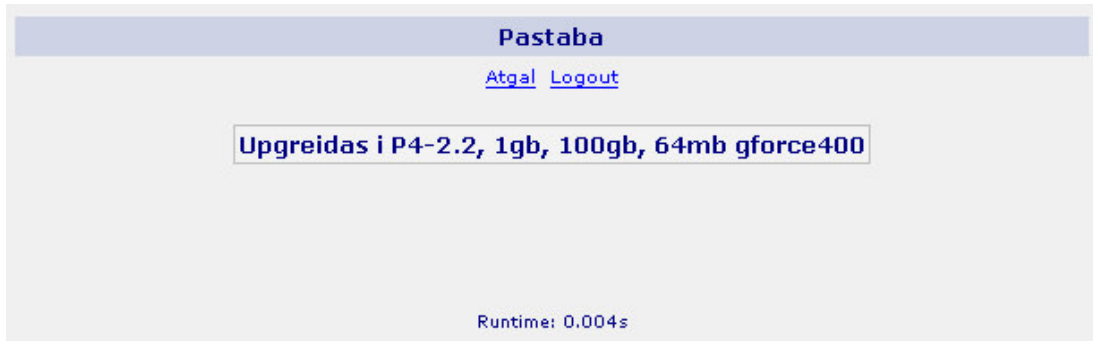
- Nesutvarkyta – reiškia kad remontas dar nebaigtas;
- Sutvarkyta - reiškia kad remontas jau pabaigtas sėkmingai;
- Nesutvarkoma – reiškia kad remontas jau pabaigtas, tačiau nesėkmingai;
- Atiduota ir neapmokėta – reiškia kad remontas jau pabaigtas sėkmingai, viskas atiduota klientui, tačiau jis dar neapmokėjo.

Tvarkant informaciją apie remontą galima atlikti šiuos veiksmus:

- Pasižiūrėti pastabą;

- Pasižiūrėti kliento informaciją;
- Redaguoti remontą;
- Įvesti naują remontą.

Pastabų peržiūrai naudojama pastabų peržiūros forma:



26 pav. Pastabų peržiūros forma

Kliento informacijos peržiūrai naudojama klientų peržiūros forma:



27 pav. Kliento peržiūros forma

Remonto redagavimui yra naudojama remonto redagavimo forma:

Remonto redagavimas

Kodas: 020424-1

Užsakovas: Petras

Firma:

Telefonas:

Garantinis Negarantinis Modernizavimas

Tvarko: Meistras Vienas Būsena: Sutvarkyta EU kaina: 300 Remontas: 230

1. Užsakovas perduoda, o vykdytojas priima testavimui (remontui, atnaujinimui) kompiuterinę techniką, kurios specifikacija pateikta 2-me skyriuje. Už programinę įrangą, instaliuotą užsakovo kompiuteryje, vykdytojas neatsako.

2. Kompiuterinės technikos specifikacija:

PIII 533, 128, 15GB, 16Mb RivaTNT

3. Gedimai, atliekami darbai, pastabos:

Upgreidas i P4-2.2, 1gb, 100gb, 64mb gforce400

Keisti Atgal

Runtime: 0.012s

28 pav. Remonto redagavimo forma

Naujo remonto įvedimo naudojama naujo remonto įvedimo forma:

Naujas remontas

[Atgal](#) [Logout](#)

Užsakovas:

Firma:

Telefonas:

Garantinis Negarantinis Modernizavimas

Tvarko: Meistras Vienas

1. Užsakovas perduoda, o vykdytojas priima testavimui (remontui, atnaujinimui) kompiuterinę techniką, kurios specifikacija pateikta 2-me skyriuje. Už programinę įrangą, instaliuotą užsakovo kompiuteryje, vykdytojas neatsako.

2. Kompiuterinės technikos specifikacija:

3. Gedimai, atliekami darbai, pastabos:

Įvesti

29 pav. Naujo remonto įvedimo forma

9.3. Remonto meistro darbas

Toliau aprašomas remonto meistro darbas. Po prisijungimo pagrindinis meniu atrodo taip:



30 pav. Remonto meistro pagrindinis meniu

Čia matomas sistemoje esančių modulių, kurie yra prieinami remonto meistriui, sąrašas, slaptažodžio keitimo forma, atsijungimo forma ir veikimo trukmės laikas. Pasirinkus modulį „Komponenčių remontas“ patenkama į remonto modulį, kurio pagrindinis meniu yra toks:



31 pav. Meistro komponenčių remonto modulio forma

Šioje formoje remonto meistras mato tik savo remontus. Kitų meistrų remontų jis nemato.

Taip pat remontų rodymas priklauso nuo remonto būsenos. Remonto būsenos meistriui yra šios:

- Sutvarkyta – Remontas nerodomas jei nepaspausta „Rodyti viską“;
- Nesutvarkyta – Remontas rodomas;
- Nesutvarkoma – Remontas nerodomas jei nepaspausta „Rodyti viską“.

Pagrindinėje formoje remonto meistras gali atlikti šiuos veiksmus:

- Pasižiūrėti kliento informaciją;
- Pasižiūrėti pastabas;
- Redaguoti remontą.

Kliento informacijos peržiūrai naudojama kliento informacijos peržiūros forma:

Kliento informacija	
Atgal Logout	
Vardas Pavardė	Petras
Pavadinimas	Nėra
Telefonas	Nėra

Runtime: 0.113s

32 pav. Kliento informacijos peržiūros forma

Pastabos peržiūrai naudojama pastabos peržiūros forma:

Pastaba
Atgal Logout
Upgreidas i P4-2.2, 1gb, 100gb, 64mb gforce400

Runtime: 0.004s

33 pav. Pastabos peržiūros forma

Remonto redagavimui naudojama remonto redagavimo forma:

Remonto redagavimas	
Upsakovas	Petras
Firma	
Telefonas	
Būsena	Nesutvarkoma
Remonto kaina	230
2. Kompiuterinės technikos specifikacija:	PIII 533, 128, 15GB, 16Mb RivaTNT
3. Gedimai, atliekami darbai, pastabos:	Upgreidas i P4-2.2, 1gb, 100gb, 64mb gforce400
<input type="button" value="Keisti"/>	

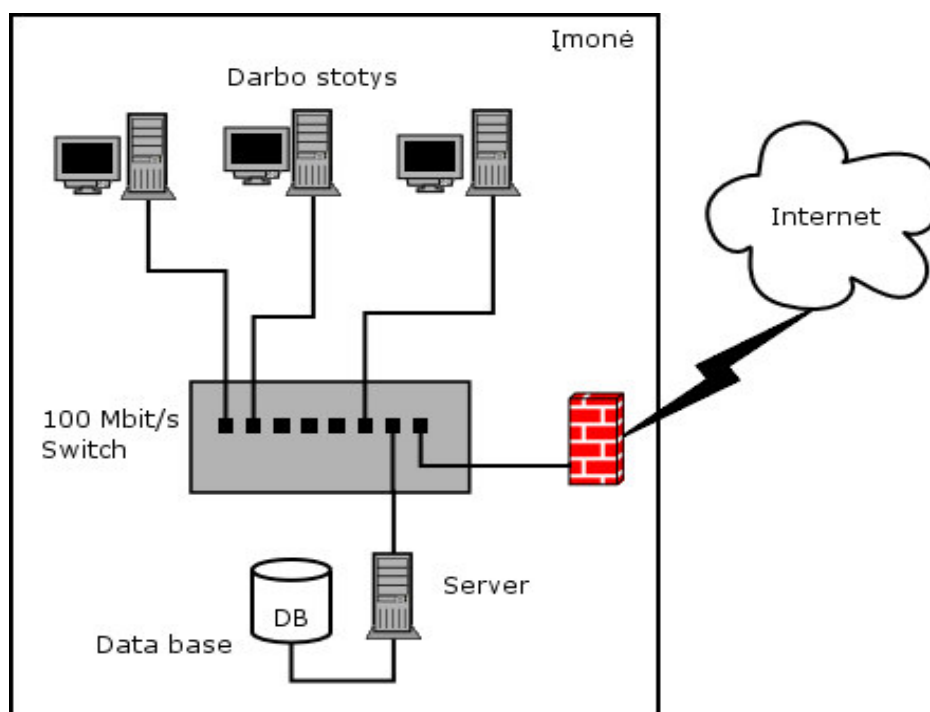
34 pav. Remonto redagavimo forma

Šioje formoje remonto meistras gali keisti tik jam reikalingą informaciją: būseną, remonto kainą ir pastabą. Kita informacija yra nekeičiama.

10. Bandomosios eksploatacijos aprašymas

Bandomoji eksploatacija buvo vykdoma kelias dienas. Sistemos bandymas vyko tame pačiame serveryje, kuriame ir buvo kuriama. Papildomų instaliacijų prieš bandymą nebuvo. Bandymas parodė, kad sistema veikia stabiliai. Aptiktos kelios smulkios klaidos. Jas ištaisius sistema funkcionuoja stabiliai. Sistemos išplėtimui sukurta sistemos vartotojų tvarkymo forma. Šiuo metu sistema yra plėtimo stadijoje. Nuspręsta praplėsti programą keliais papildomais moduliais, kurie padidins sistemos kokybę.

Papildomai atlikta Apache serverio dirbančio su sukurta programine įranga veikimo matavimai. Matavimo tikslas: įsitikinti, kad prie 1000 vartotojų sistema vidiniame tinkle veiks taip pat kaip ir prie 10. Toliau pateikiama lokalaus tinklo sujungimo schema:



35 pav. Įmonės vidinio tinklo schema

Buvo tikrinamos šios nuorodos:

- <http://mt.home.lt/index.php>;
- <http://mt.home.lt/vbid?id=040102-1>.

Rezultatai pateikti toliau:

Naudota programa: ab (12)

Pirma nuoroda:

Complete requests: 1000
Failed requests: 0
Broken pipe errors: 0
Requests per second: 48.56 [#/sec] (mean)
Time per request: 20.59 [ms] (mean)
Time per request: 20.59 [ms] (mean, across all concurrent requests)
Transfer rate: 97.55 [Kbytes/sec] received

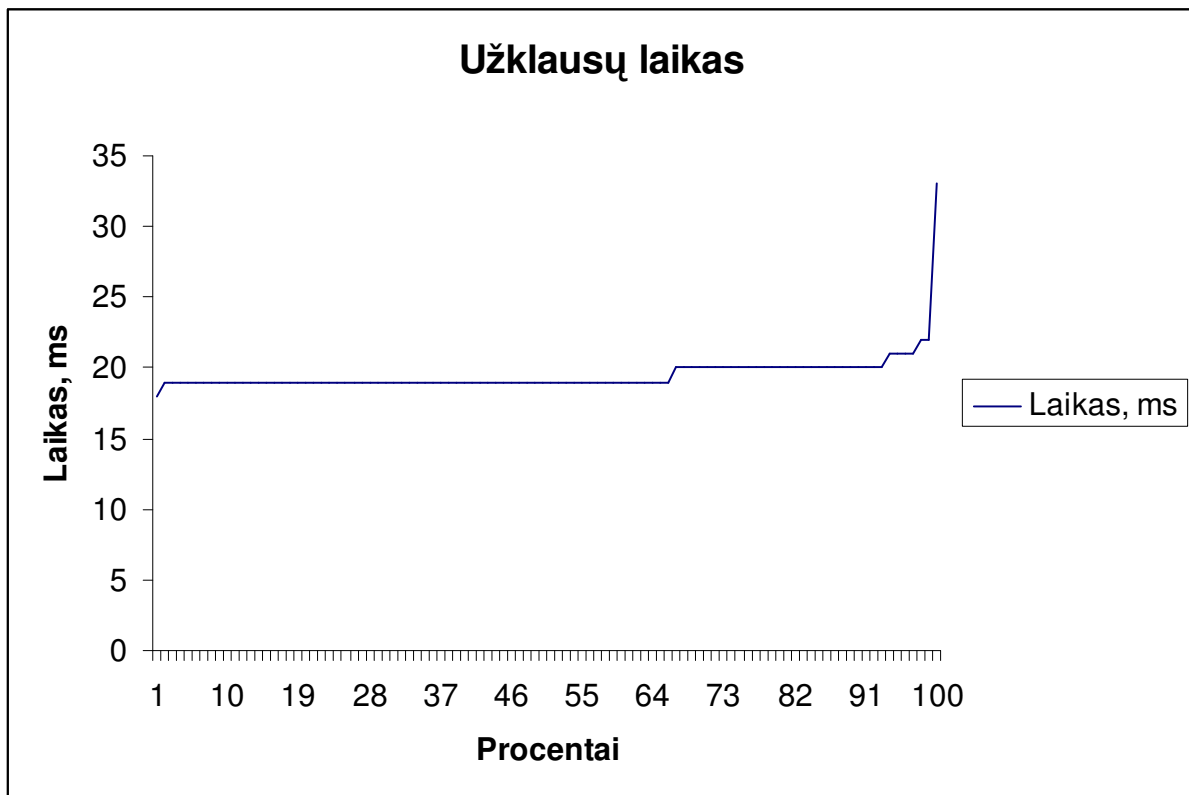
Connnection Times (ms)

	min	avg	max
Connect:	0	0	2
Processing:	18	20	374
Total:	18	20	374

Percentage of the requests served within a certain time (ms)

50%	19
66%	20
75%	20
80%	20
90%	20
95%	21
98%	22
99%	31
100%	374 (last request)

Žemiau yra pateikiama grafinė atsakymo laikų priklausomybė nuo užklausų skaičiaus procentais:



36 pav. Pirmos nuorodos bandymų rezultatų grafikas

Antra nuoroda

Complete requests: 1000

Failed requests: 0

Broken pipe errors: 0

Requests per second: 44.22 [#/sec] (mean)

Time per request: 22.61 [ms] (mean)

Time per request: 22.61 [ms] (mean, across all concurrent requests)

Transfer rate: 8.45 [Kbytes/sec] received

Connnection Times (ms)

min avg max

Connect: 0 0 37

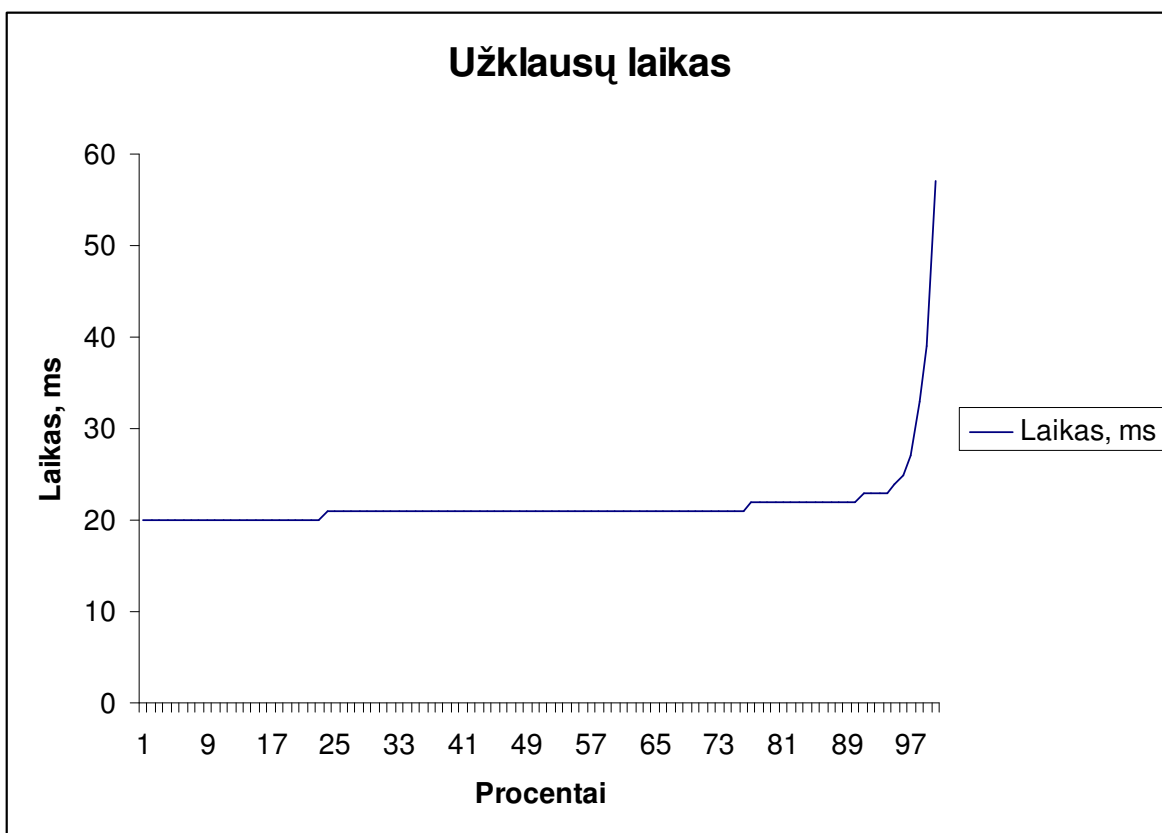
Processing: 20 22 89

Total: 20 22 89

Percentage of the requests served within a certain time (ms)

50%	21
66%	21
75%	21
80%	22
90%	23
95%	25
98%	39
99%	55
100%	89 (last request)

Žemiau yra pateikiama grafinė atsakymo laikų priklausomybė nuo užklausų skaičiaus procentais



37 pav. Antros nuorodos bandymų rezultatų grafikas

11. Išvados

1. Atlikta įvairių technologinių platformų analizė ir parinkta optimaliausia (LAMP).
2. Suprojektuota modulinė sistemos architektūra.
3. Architektūra realizuota naudojant LAMP platformos technologijas.
4. Sukurtoje sistemoje įdiegta šablonu posisteme, atskirianti programas nuo dinamiškai generuojamų puslapių.
5. Naudojantis ApacheBench testavimo priemonėmis iširtas sistemos stabilumas ir patikimumas.
6. Sukurta įmonės vidinio tinklo infrastruktūra leidžianti stebėti kompiuterijos darbų būklę.
7. Realizuotos papildomos duomenų šifravimo ir saugumo posistemės užtikrina saugų vartotojų duomenų pernešimą nevienalytėje Internet aplinkoje.

Sukaupta patirtis publikuota autoriaus vadovaujamame portale <http://debian.home.lt>.

Darbo rezultatai panaudoti naujų lietuviškų Linux distribucijų kūrime.

12. Literatūra

1. Windows 2003 Server aprašymas (2003), Internetė:
<http://www.microsoft.com/lietuva/windowsserver2003/overview/default.asp>
2. Debian GNU/Linux aprašymas (2003), Internetė <http://www.debian.org/index.lt.html>
3. MySQL duomenų bazės aprašymas (2003), Internetė: <http://www.mysql.com>
4. PostgreSQL duomenų bazės aprašymas (2003), Internetė: <http://www.postgresql.org>
5. Perl programavimo kalba (2003), Internetė: <http://www.perl.org>
6. PHP programavimo kaba (2003), Internetė: <http://www.php.net>
7. Microsoft Access duomenų bazės aprašymas (2003), Internetė:
<http://www.microsoft.com/lietuva/officexp/programs/access/>
8. Microsoft SQL duomenų bazės aprašymas (2003), Internetė:
<http://www.microsoft.com/sql/evaluation/overview/default.asp>
9. ASP programavimo kalba (2003), Internetė:
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnasp/html/asptutorial.asp?frame=true&hidetoc=true>
10. Apache http serverio aprašymas (2003), Internetė: <http://www.apache.org>
11. IIS http serverio aprašymas (2003) , Internetė:
<http://www.microsoft.com/WindowsServer2003/iis/default.msp>
12. Straipsnis apie http serverių paplitimą WWW (2004), Internetė:
http://news.netcraft.com/archives/2004/01/01/january_2004_web_server_survey.html
13. ApacheBench programa (2004) <http://httpd.apache.org>
14. Grigin, I., PHP4: specialnij spravochnik, Piter 2002
15. M. Widenius, MySQL reference manual, O'Reilly and Associates 2002
16. J. Castagnetto, Professional PHP programming, Wrox 1999
17. L. Welling, PHP and MySQL Web Development, SAMS publishing, 2001

13. Terminų ir santrumpų žodynis

- TVS – turinio valdymo sistemos (angl. CMS – Content Management Systems)
- PHP – hyper teksto pirminio apdorojimo programa (angl. Hypertext Preprocessor)
- ASP – aktyvūs serverio puslapiai (angl. Active Server Pages)
- CGI – dažnas išėjimo interfeisas (angl. Common Gateway Interface)
- HTTP – hyper teksto siuntimo protokolas (angl. Hyper Text Transfer Protocol)
- Perl – praktiška informacijos manipuliavimo ir ataskaitų kalba (angl. Practical Extraction and Report Language)
- WWW – pasaulinis žiniatinklis (angl. World Wide Web)
- HTML – hyper teksto žymėjimo kalba (angl. Hyper Text Markup Language)
- LAMP – dinaminių tinklapių kūrimo platforma, į kurios sudėtį įeina Linux, Apache, MySQL, PHP/Python/Perl
- UML – universali modeliavimo kalba (angl. Universal Modeling Language)
- SSL – saugus duomenų kanalas (angl. Secure Socket Layer)
- BIOS – pradinė kompiuterių įkrovimo programa (angl. Basic Input Output System)
- CD-ROM – kompaktinių plokštelių nuskaitymo įrenginys (angl. Compact Disc – Read Only Memory)
- SWAP – laikinų failų saugojimo vieta
- LILO – Linux operacijų sistemos įkrovimo programa (angl. Linux Loader)

14. Summary

There is a great variety of software solutions which improve quality of work and increase productivity. Nevertheless there still is a demand for specialized software solutions, often coupled to particular production systems. In this work author approaches such a concrete subject of computer hardware and repair in small and medium enterprise. Proposed information system not only allows internal administration of service requests within a company, but also provides complete status and process overview for customers accessing company's intranet. This is a unique solution of a kind as others are too general and do not fulfill specific demands posed in the specification.

The main subject of this work is software design, development and investigation. Analysis of current software technology platforms was performed and the best deployment platform LAMP (Linux-Apache-MySQL-PHP) was selected. Modular system design provided easy system development and future extendibility as well as integration into other portals. Final resulting information system incorporates a number of advanced networking and programming technologies: template engine, certificate based security, data encryption engine and others. Implemented software is customized to meet the needs of particular enterprise and is currently used as part of its' intranet and service network. Legal open source technologies were mostly used for developing the product.

15. Priedai

Žemiau yra pateikiami šie priedai:

- Duomenų bazės kodas, tinkantis MySQL duomenų bazei

15.1. Duomenų bazės kodas

```
#
# Sukurta duomenų struktūra lentelei `client`
#

CREATE TABLE client (
  cnumber tinyint(4) NOT NULL auto_increment,
  cname text NOT NULL,
  flname text NOT NULL,
  phone int(10) NOT NULL default '0',
  KEY cnumber (cnumber),
  FULLTEXT KEY cname (cname)
) TYPE=MyISAM AUTO_INCREMENT=28 ;

# -----

#
# Sukurta duomenų struktūra lentelei `edit_history`
#

CREATE TABLE edit_history (
  serial text NOT NULL,
  date text NOT NULL,
  editor int(11) NOT NULL default '0'
) TYPE=MyISAM;

# -----

#
# Sukurta duomenų struktūra lentelei `main`
#

CREATE TABLE main (
  serial text NOT NULL,
  client bigint(11) NOT NULL default '0',
  date date NOT NULL default '0000-00-00',
  idate date NOT NULL default '0000-00-00',
  status tinyint(1) NOT NULL default '0',
  repairer smallint(4) NOT NULL default '0',
  remark bigint(11) NOT NULL default '0',
  price double NOT NULL default '0',
  reprice double NOT NULL default '0',
  creator int(11) NOT NULL default '0',
  spec int(11) NOT NULL default '0',
  reptype tinyint(4) NOT NULL default '0',
  KEY client (client),
  KEY date (date),
  KEY idate (idate)
) TYPE=MyISAM;

# -----
```

```

#
# Sukurta duomenų struktūra lentelei `remark`
#

CREATE TABLE remark (
  mnumber tinyint(4) NOT NULL auto_increment,
  mtext text NOT NULL,
  KEY mnumber (mnumber)
) TYPE=MyISAM AUTO_INCREMENT=38 ;

# -----

#
# Sukurta duomenų struktūra lentelei `repairer`
#

CREATE TABLE repairer (
  rnumber tinyint(4) NOT NULL auto_increment,
  rflname varchar(50) NOT NULL default '0',
  rname varchar(50) NOT NULL default '0',
  raddress varchar(50) NOT NULL default '0',
  rphone int(10) NOT NULL default '0',
  rccode int(10) NOT NULL default '0',
  runumber int(11) NOT NULL default '0',
  module tinyint(4) default NULL,
  KEY rnumber (rnumber)
) TYPE=MyISAM AUTO_INCREMENT=14 ;

# -----

#
# Sukurta duomenų struktūra lentelei `specification`
#

CREATE TABLE specification (
  nspec tinyint(4) NOT NULL auto_increment,
  spec text NOT NULL,
  PRIMARY KEY (nspec),
  KEY nspec (nspec)
) TYPE=MyISAM AUTO_INCREMENT=12 ;

# -----

#
# Sukurta duomenų struktūra lentelei `type`
#

CREATE TABLE type (
  tnumber tinyint(4) NOT NULL auto_increment,
  tname text NOT NULL,
  KEY tnumber (tnumber)
) TYPE=MyISAM AUTO_INCREMENT=15 ;

# -----

#
# Sukurta duomenų struktūra lentelei `uforma`
#

```

```
CREATE TABLE uforma (  
  text1 text NOT NULL,  
  text2 text NOT NULL,  
  text3 text NOT NULL  
) TYPE=MyISAM;  
  
# -----  
  
#  
# Sukurta duomenų struktūra lentelei `user`  
#  
  
CREATE TABLE user (  
  unumber tinyint(4) NOT NULL auto_increment,  
  ulogin varchar(10) NOT NULL default '0',  
  upassword varchar(35) NOT NULL default '0',  
  urights tinyint(1) NOT NULL default '0',  
  uflname varchar(50) NOT NULL default '0',  
  KEY unumber (unumber)  
) TYPE=MyISAM AUTO_INCREMENT=39 ;
```