

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Žydrūnas Dieninis

**Biomedicininės informacinės sistemos realizacija
internete**

Magistro darbas

Darbo vadovas

doc. Vacys Jusas

Kaunas

2003

TURINYS

1. SANTRAUKA ANGLŲ KALBA	3
2. SANTRUMPŲ IR TERMINŲ ŽODYNAS	4
3. ĮVADAS	5
3.1. Informacinių sistemų panaudojimo gydymo įstaigose aktualumas.	5
3.2. Bendri reikalavimai, keliami tokio pobūdžio sistemoms.....	6
3.3. Panašių sistemų apžvalga.....	7
3.4. Darbo tikslas	7
3.5. Darbo struktūra	7
4. ANALITINĖ DALIS	8
4.1. Duomenų bazių publikavimo internete principai.....	8
4.2. Galimų sistemos realizacijos priemonių apžvalga.....	13
4.2.1. Duomenų bazių valdymo sistemos.	13
4.2.2. Priemonės, naudojamos vartotojo sąsajos realizacijai.....	14
5. METODO APRAŠYMAS	18
5.1. Apibendrintas sistemos aprašas	18
5.2. Sistemos programinės įrangos posistemės.....	20
5.2.1. Prisijungimo prie duomenų bazės posistemė.....	21
5.2.2. Vartotojų administravimo posistemė.....	21
5.2.3. Duomenų apdorojimo posistemė.....	22
5.2.4. Vartotojo sąsaja.....	22
5.3. Sistemos duomenų bazė.....	23
5.3.1. Informacijos apie pacientus saugojimas.	23
5.3.2. Informacijos apie sistemos vartotojus saugojimas.....	24
5.4. Įgyvendinimo priemonių parinkimas.....	25
6. EKSPERIMENTINĖ DALIS	26
6.1. Sukurtos programinės įrangos aprašymas.....	26
6.1.1. Prisijungimo prie duomenų bazės posistemė.....	28
6.1.2. Vartotojų administravimo posistemė.....	28
6.1.3. Duomenų apdorojimo posistemė.....	29
6.1.4. Vartotojo sąsajos posistemė.....	29
6.2. Sukurtos programinės įrangos vieta sistemoje.....	30
6.3. Eksperimentinės programinės įrangos veikimas.....	31

6.3.1. Įvadas.....	31
6.3.2. Pagrindinis meniu.....	31
6.3.3. Sistemos administravimas.....	32
6.3.4. Duomenų įvedimo, redagavimo ir šalinimo formos.....	34
6.3.5. Ataskaitos.....	38
7. IŠVADOS.....	40
8. LITERATŪRA.....	41

1. SANTRAUKA ANGLŲ KALBA

Realization of informational system in the field of biomedicine

The business of healthcare management is changing rapidly, heading in the direction of managed care, capitation, and integrated delivery systems. Information management is crucial to the success and competitiveness of these new care delivery systems. A major goal of this work is to develop a model for a computer-based patient medical record system which could be accessed through the world wide web. The fundamental function of such system is to record, monitor, retrieve all events associated with an encounter between the patient and the healthcare system.

This document presents a description of informational system in the field of biomedicine which is appointed to patients, with coronary heart disease.

First of all, urgency, applicability and functions of such informative system are presented.

Further the main points of the problem, which consists of system realization and publishing on the web, and its solutions are analyzed in details.

Keywords: medical record, informational system in the field of biomedicine, world wide web, web publishing.

2. SANTRUMPŲ IR TERMINŲ ŽODYNAS

API (<i>Application Program Interface</i>)	programos sąsaja su vartotoju.
ASP (<i>Active Server Pages</i>)	dinaminių tinklapių kūrimo technologija.
CGI (<i>Common Gateway Interface</i>)	tinklo serveryje vykdomi scenarijai.
DB	duomenų bazė.
DBVS	duomenų bazės valdymo sistema.
HTML (<i>HyperText Markup Language</i>)	programavimo kalba, skirta hipertekstinių dokumentų kūrimui.
HTTP (<i>HyperText Transfer Protocol</i>)	protokolas, kuriuo perduodama informacija tarp serverio ir interneto naršyklės.
ODBC (<i>Open Database Connectivity</i>)	prisijungimo prie DBVS serverio sąsajos standartas.
OS	operacinė sistema.
PHP (<i>PHP Hypertext Preprocessor</i>)	programavimo kalba, sukurta specialiai tinkliniam programavimui ir skirta tinklo serveryje vykdomų scenarijų rašymui.
SQL (<i>Structured Query Language</i>)	stuktūrinių užklausų kalba.
TCP/IP	protokolų rinkinys, naudojamas į tinklą sujungtų kompiuterių komunikacijai.

3. ĮVADAS

3.1. Informacinių sistemų panaudojimo gydymo įstaigose aktualumas.

Vis daugiau šiuo metu kuriamų informacinių sistemų skirtos darbui globaliam tinkle. Panaudojant WWW technologijas, informacinės sistemos tampa prieinamos vartotojams iš bet kurio pasaulio taško ir reikalauja minimalių vartotojo darbo naršykle žinių ir įgūdžių. Tokias sistemas paprastai sudaro duomenų bazė ir per internetą prieinama vartotojo sąsaja.

Ypatingas tokių sistemų poreikis jaučiamas įvairiausiose veiklos srityse. Viena iš jų - medicina.

Pacientas sveikatos apsaugos sistemoje susiduria su eile sveikatos priežiūros grandžių. Jis gydomas eilėje sveikatos priežiūros įstaigų, jam atliekama eilė tyrimų, procedūrų, operacija. Šiame procese dalyvauja įvairių sveikatos priežiūros grandžių specialistai ir institucijos, kuriose pacientai konsultuojami arba gydomi.

Ligonio administraciniai ir medicininiai duomenys kaupiami įvairiuose diagnostikos, gydymo ir reabilitacijos etapuose, todėl tikslinga šį procesą aktyviai valdyti, panaudojant bendrą medicininių ir administracinių duomenų banką, kuriame saugoma informacija turėtų būti prieinama visiems gydytojams ir specialistams, bendravusiems su ligoniu.

Tokios sistemos įgyvendinimas leis gydytojams operatyviai gauti naujausią informaciją apie paciento gydymo eigą ir esant reikalui vykdyti neatidėliotinas konsultacijas. Kadangi duomenys apie paciento būklę paprastai kaupiami įvairiose medicinos įstaigose, kurios dažnai yra išsidėstę įvairiuose miestuose, centralizuotas duomenų saugojimas ir valdymas suteiktų galimybę įvairių medicininių įstaigų specialistams pildyti paciento duomenis, peržiūrėti informaciją apie jam atliktus tyrimus, gydymo eigą.

Sistemos įjungimas į globalų tinklą leidžia dinamiškesnį duomenų srautų judėjimą. Pacientų medicininių duomenų bazės naudojimo WWW tinkle ir jų būklės pastovaus stebėjimo ir ilgalaikės reabilitacijos programos vykdymo efektyvumo vertinimo sukūrimas panaudojant šiuolaikines technologijas pagerins diagnostiką ir būklės įvertinimą, padidins operatyvumą keičiantis medicinine informacija.

3.2. Bendri reikalavimai, keliami tokio pobūdžio sistemoms

Šiuo metu medicinos įstaigose informacija apie pacientus ir jų gydymo eigą saugoma pačia įvairiausia forma. Gydymo įstaigos dažniausiai nutolę viena nuo kitos ir neturi galimybės operatyviai keistis turima informacija, todėl tikslinga siekti visą šia informaciją apibendrinti ir saugoti centralizuotai, bei padaryti prieinamą kuo didesniam kiekiui nutolusių vartotojų. Atsižvelgiant į tokioms sistemoms teikiamas rekomendacijas [1], galima suformuluoti tokius bendrus reikalavimus:

- ❑ Didelių duomenų apimčių saugojimas.
- ❑ Duomenų saugojimo patikimumas (asmens duomenų apsauga, poreikis saugoti duomenis ne vienerius metus).
- ❑ Prieinamumas nutolusiems vartotojams (per internetą).
- ❑ Didelis darbo greitis.

Duomenų saugojimas.

Sistema turi suteikti galimybę saugoti įvairią informaciją: paciento asmens duomenis, informaciją apie gydymą, vartojamus medikamentus, gydymo rezultatus.

Kadangi saugomos informacijos apimtys labai didelės ir nuolat didės, turi būti suprojektuota atitinkama duomenų bazė ir parinkta patikima, greita ir saugi duomenų bazių valdymo sistema. Taip pat duomenų bazių

Duomenų pateikimas vartotojui.

Sistemoje saugomi duomenys turi būti prienami maksimaliam kiekiui galimų vartotojų (gydymo įstaigų darbuotojų), todėl turi būti suprojektuota ir realizuotas patikimas priėjimas prie sistemos tiek lokaliame, tiek globaliame tinkle. Taip pat turi būti užtikrinta:

- ❑ Patikima vartotojų autorizacija.
- ❑ Saugomų duomenų peržiūros, redagavimo, papildymo galimybės.
- ❑ Naudojimosi paprastumas.
- ❑ Kuo didesnis darbo greitis.

.3.3. Panašių sistemų apžvalga.

Galima spėti, kad panašių sistemų, skirtų medicininių duomenų valdymui, yra sukurta tikrai ne viena, bet detalią informaciją apie jas rasti yra labai sudėtinga, nes pačios sistemos nėra publikuojamos. To priežastis - panašios sistemos kuriamos specializuotoms organizacijoms (šiuo atveju – gydymo įstaigoms), atsižvelgiant į jų poreikius, todėl jos nėra skirtos plačiam vartotojų ratui. Be to, turint omenyje, kad tokiose sistemose yra dirbama su asmens duomenimis (paciento asmeniniai duomenys, įrašai apie susirgimus, gydymą), kurie paprastai negali būti paviešinti be asmens sutikimo, priėjimas prie jų (sistemų) suteikiamas tik labai ribotam kiekiui asmenų (gydymo įstaigų darbuotojams) ir yra griežtai kontroliuojamas.

Iš Lietuvoje naudojamų panašių sistemų galima paminėti **KMU Psichofiziologijos ir reabilitacijos institute** naudojamą internetinį medicininių duomenų banką [2], skirtą saugoti informacijai apie pacientų, kuriems atlikta širdies operacija, reabilitacijos procesą. Sprendžiant iš instituto svetainėje pateiktos informacijos, ši sistema skirta kaupti bei valdyti duomenims apie reabilitacijos klinikos pacientus, jiems atliktus tyrimus, procedūras, būseną bei peržiūrėti sukauptą informaciją naudojantis internetu.

Būtent analogiškos sistemos suprojektavimas ir sukūrimas yra šio darbo tikslas.

3.4. Darbo tikslas

Projekto tikslas - realizuoti informacinę sistemą, kuri leistų nutolusiems vartotojams naudotis centralizuotoje medicininių duomenų bazėje saugoma informacija.

3.5. Darbo struktūra

4 skyriuje pateikiami bendri duomenų bazių publikavimo internete principai, apžvelgiamos galimos realizacijos priemonės.

5 skyriuje analizuojamas sistemos realizavimo būdas, pateikiama jo metodika.

6 skyriuje aprašoma pasirinkta metodika realizuota eksperimentinė sistema.

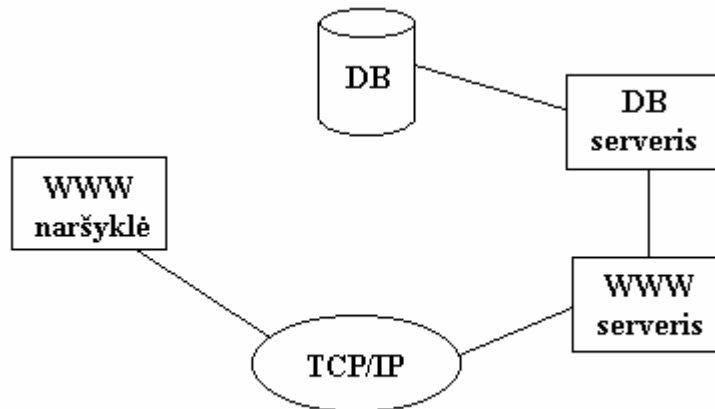
7 skyriuje pateikiamos darbo išvados ir rekomendacijos.

4. ANALITINĖ DALIS

4.1. Duomenų bazių publikavimo internete principai

Tipinę duomenų bazės publikavimo internete sistemą sudaro: interneto naršyklė su vartotojo sąsaja, serveris su duomenų baze (duomenų saugojimui) ir tinklo serveris, kuris apjungia du anksčiau paminėtus komponentus. Tinklo serverio programinė įranga supaprastina ir standartizuoja vartotojo sąsają bei duomenų atvaizdavimą, o DBVS organizuoja ir standartizuoja duomenų saugojimą ir pateikimą.

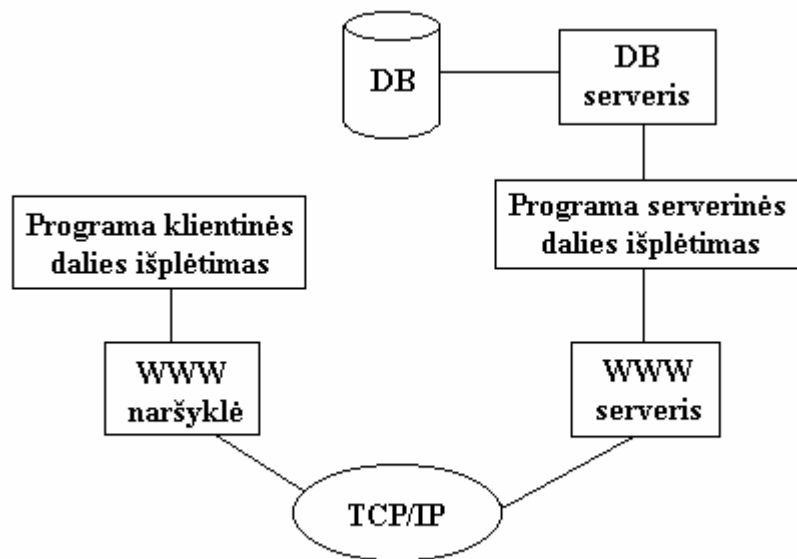
Vienintelė programa, kuri turi būti įdiegta vartotojo kompiuteryje – interneto naršyklė. Vartotojas komunikuoja su duomenų baze formos, atvaizduotos tinklapyje, pagalba.



1 pav. Supaprastinta internetinės duomenų bazės struktūra.

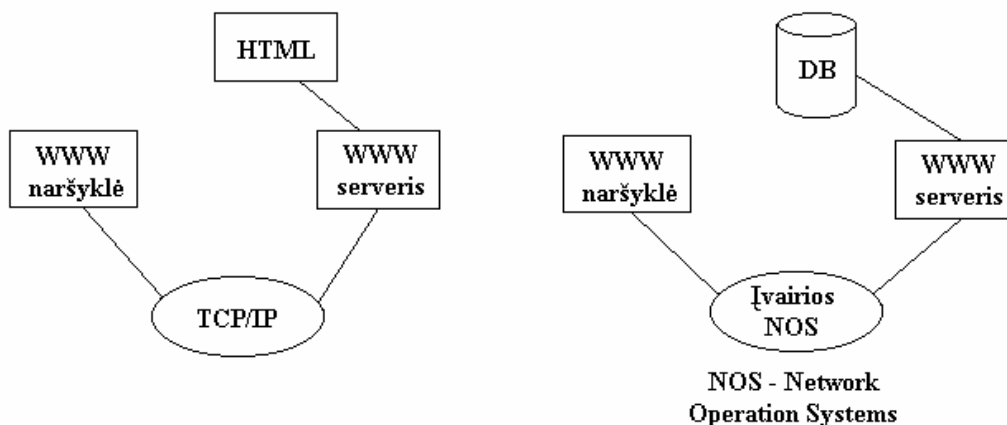
Informacija, paimta iš duomenų bazės, tinklapyje gali būti pateikiama tekstine, grafine ar kitokia forma.

Schemoje parodytas modelis gali pasirodyti labai paprastas, bet daugybė interneto sistemų dirba būtent pagal tokią schemą. Sistemoms, reikalaujančioms sudėtingesnės vartotojo sąsajos, egzistuoja standartinės priemonės, išplečiančios interneto naršyklės galimybes; sistemoms, kurioms reikalingas sudėtingesnis duomenų apdorojimas, egzistuoja standartinės priemonės išplečiančios tinklo serverio galimybes.



2 pav. WWW duomenų bazė su klientinės ir serverio dalies išplėtimais.

Internetinių duomenų bazių pagrindu tapo dvi technologijos, kurios vystėsi nepriklausomai viena nuo kitos ir turėjo skirtingus tikslus.



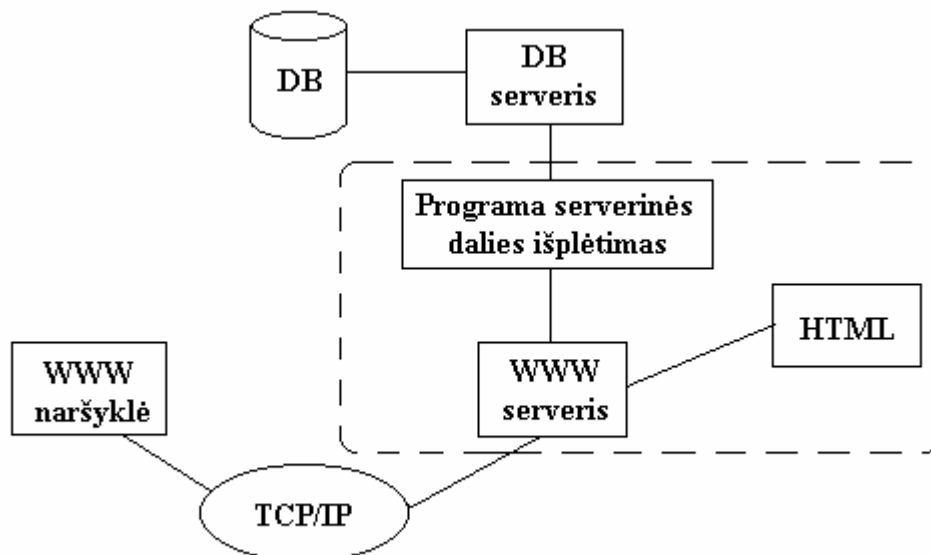
3 pav. WWW ir DBVS vystėsi kaip dvi nepriklausomos technologijos.

Kiekviena iš šių technologijų turėjo eilę svarbių privalumų:

- Tinklo sistemos turi paprastą ir aišką vartotojo sąsajos apibrėžimą (tinklapiai, nuorodos), gerai apgalvotą duomenų aprašymo kalbą (HTML – HyperText Markup Language), greitą duomenų perdavimo protokolą, kuris leidžia perduoti didelius informacijos kiekius per laiko vienetą (HTTP – HyperText Transport Protocol), multimedinės informacijos perdavimo priemonės ir, svarbiausia, kad šios sistemos nepriklauso nuo platformos ir gana nebrangiai kainuoja.

- Duomenų bazių sistemos turi aiškiai apibrėžtą duomenų modelį, greitą duomenų išsaugojimą/pateikimą (SQL, ataskaitos), programines priemones skirtas vartotojo sąsajai realizuoti (kliento/serverio priemonės), gerai išvystytas saugumo ir autentifikavimo priemones, patikimus metodus, skirtus duomenų valdymui ir integralumo užtikrinimui.

Tiek tinklo sistemos, tiek duomenų bazių sistemos remiasi dviejų lygių kliento/serverio architektūra. Duomenų bazių publikavimo internete sistemos apjungia šias dvi architektūras į trijų lygių kliento/serverio architektūrą. Kliento lygį užima interneto naršyklė, serverio lygyje yra duomenų bazės serveris, o tarpiniame lygyje – tinklo serveris ir jo praplėtimo programa.



4 pav. Trijų lygių WWW duomenų bazės realizacija.

Toks architektūrinis sprendimas leidžia sumažinti tinklo apkrovimą, daro atskirus komponentus lengvai pakeičiamais ir padidina sistemos saugumą. Veikimo principas:

1. Naršyklė siunčia tinklo serveriui užklausą pateikti reikalingus duomenis arba tinklapį.
2. Tinklo serveris vykdo užklausą užkrauti tinklapį, o užklausas duomenims perduoda serverinės dalies praplėtimo programai.
3. Praplėtimo programa priima užklausą ir transformuoja ją į duomenų bazės serveriui suprantamą formą (pvz. ODBC SQL). Transformuota užklausa perduodama DB serveriui.

4. DB serveris aptarnauja užklausą ir grąžina rezultatą serverinės dalies praplėtimo programai.
5. Praplėtimo programa transformuoja rezultatą į interneto naršyklei suprantamą formą (pvz. HTML) ir perduoda jį tinklo serveriui.
6. Tinklo serveris perduoda rezultatą į interneto naršyklę.

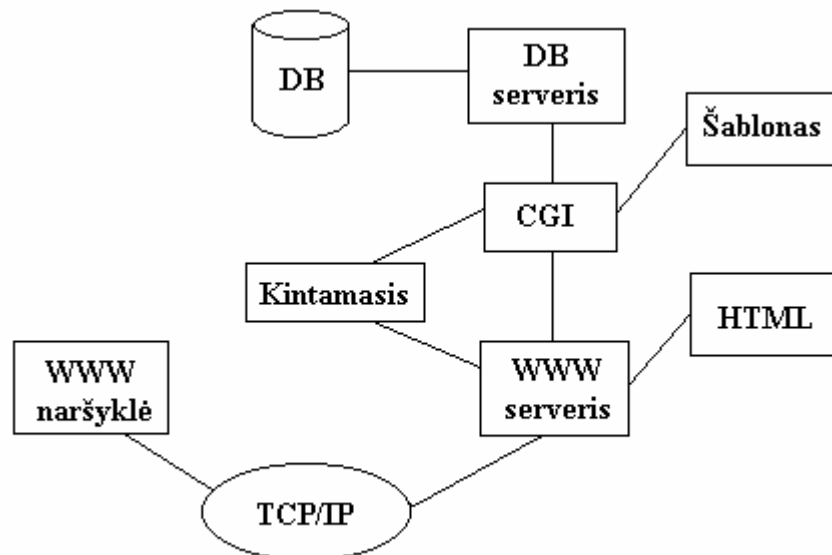
Serverinės dalies praplėtimo programos.

Viena iš pagrindinių praplėtimo programų panaudojimo priežasčių tarpiniame lygyje tarp tinklo ir DB serverių – būtinybė transliuoti tarp jų vykstančią komunikaciją.

Egzistuoja įvairūs serverio dalies praplėtimo būdai. Juos būtų galima suskirstyti į tris kategorijas: paprastas CGI, hibridinis CGI ir API [3].

CGI

CGI buvo pirmasis protokolas suteikęs galimybę programuotojams kurti programas, praplečiančias tinklo serverių funkcionalumą. Dauguma pirmųjų interneto duomenų bazių sistemų buvo sukurta būtent CGI pagalba ir ši architektūra iki šiol yra labai populiari.



5 pav. WWW duomenų bazė naudojanti CGI.

Tinklo serveris komunikuoja su CGI-programa per aplinkos kintamuosius ir standartinę operacinės sistemos įvestį. Tokie parametrai kaip URL ir vartotojo IP adresas perduodami per aplinkos kintamuosius, o vartotojo įvedimas į formas – per standartinę įvestį.

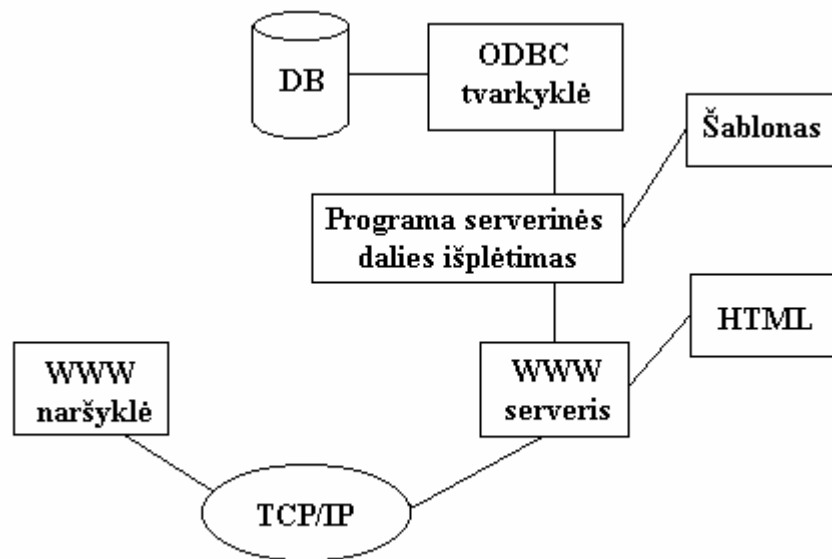
Serverio dalies išplėtimo programas, naudojančias CGI galima rašyti bet kokia programavimo kalba, bet dažniausiai yra naudojamos Perl, C, PHP. Programavimo kalbos pasirinkimas gali turėti didelę įtaką sistemos darbo efektyvumui.

ODBC Internetinėms duomenų bazių sistemoms.

Daugelis serverio dalies išplėtimo programų yra paremtos architektūra, į kurią yra įtrauktas ODBC.

ODBC – tai API sąsaja, leidžianti unifikuoti užklausas skirtas duomenų bazei. Ši standartą palaiko dauguma įvairių programinių produktų. Naudojant ODBC, net pakeitus duomenų bazę, naudojami šablonai ir ODBC SQL operatoriai išliks nepakitę.

ODBC funkcionuoja sudarydamas papildomą lygį tarp užklausa siunčiančios programos ir duomenų bazės.



6 pav. WWW duomenų bazė panaudojant ODBC tvarkyklę.

4.2. Galimų sistemos realizacijos priemonių apžvalga.

Kuriamą sistemą sudaro dvi pagrindinės sudedamosios dalys: duomenų bazė ir vartotojo sąsaja, todėl tikslinga apžvelgti šiuo metu naudojamas priemones, skirtas šioms, panašių sistemų, sudedamosioms dalims realizuoti.

4.2.1. Duomenų bazių valdymo sistemos.

Oracle

Oracle – moderni duomenų bazių valdymo sistema, pritaikyta darbui įvairiose operacinėse sistemose: Windows NT, Unix, Linux.

Oracle DBVS galimybės:

- ❑ Daugialygis duomenų saugumo užtikrinimas.
- ❑ Priemonės, užtikrinančios duomenų integralumą.
- ❑ Įvairių kalbų palaikymas.
- ❑ Patikimas vartotojų administravimas, teisių, privilegijų suteikimas.
- ❑ Priėjimui prie duomenų naudojamas SQL ir PL/SQL.
- ❑ Oracle prekompiliatorius SQL'ui.
- ❑ Galimybė naudoti išorines funkcijas iš DLL bibliotekų ir OCI sąsaja, funkcijų iškvietimui.
- ❑ Paskirstyto programavimo technologijos palaikymas.
- ❑ API sąsaja papildomų programinių priedų kūrimui.

Kaip vieną iš svarbiausių naujovių Oracle DBVS, galima išskirti perėjimą nuo reliacinių duomenų bazių prie objektinių-reliacinių, kas leidžia dirbti su objektiniais tipais. Dar viena Oracle naujovė – LOB tipų realizacija.

Ms SQL Server

SQL Server – Microsoft kompanijos produktas, todėl šios DBVS panaudojimas apsiriboja Windows aplinka. Tai bene didžiausias šios sistemos trūkumas, nes galimybėmis ir vartotojui teikiamais servisais ji nenusileidžia aukščiau minėtam Oracle. Išskirtiniu SQL Server privalumu (lyginant su Oracle) galima laikyti patogią grafinę vartotojo sąsają.

Kaip ir Oracle, SQL Server yra įgyvendintos šios darbo su DB priemonės:

- Duomenų saugumo užtikrinimo mechanizmai.
- Priemonės, užtikrinančios duomenų integralumą.
- Priėjimui prie duomenų naudojamas SQL ir TRANSACT-SQL.
- Galimybė naudoti išorines arba pačioje DB saugomas funkcijas.

MySQL

MySQL – nekomercinė DBVS, pritaikyta darbui Windows, Unix, Linux operacinėse sistemose ir paskutiniu metu ypatingai plačiai naudojama dinaminių tinklapių kūrimui. Nors ši sistema ir nepasižymi daugeliu savybių, kurios būdingos komerciniams tokio pobūdžio produktams, bet yra vertinama už įdiegimo ir priežiūros paprastumą bei darbo greitį – labai svarbų faktorių publikuojant duomenų bazes internete. Paprastai naudojama nesudėtingoms ir nedidelės apimties duomenų bazėms įgyvendinti.

4.2.2. Priemonės, naudojamos vartotojo sąsajos realizacijai.

CGI

CGI (Common Gateway Interface) skriptai - tai programos, kurios veikia tinklo serveryje. Jų pagalba galima išplėsti serverio funkcionalumą. Teoriškai, CGI skriptų panaudojimas, leidžia interpretuoti vartotojo įvestus duomenis HTML formose, tuos duomenis apdoroti ir grąžinti vartotojui rezultatą. Praktiškai, tai yra programos, kurios leidžia vartotojui lengvai palaikyti ryšį su serveriu. Naudojant CGI skriptus nereikia žinoti tinklo ar serverio veikimo smulkmenų.

CGI programas galima rašyti bet kokia programavimo kalba, kuri turi priemones, leidžiančias bendrauti su serveriu. Šios programos galės dirbti su bet kuriuo serveriu, kuris palaiko CGI protokolą. Naudojant CGI skriptus neiškyla WWW naršyklių senumo klausimas, nes CGI programas vykdo ne naršyklė bet serveris.

Rašant CGI programas, itin svarbu teisingai pasirinkti programavimo kalbą. Pasirenkama programavimo kalba turi sugebėti: perduoti duomenis į standartinį duomenų išvedimo srautą, nuskaityti duomenis iš standartinio įvedimo srauto ir nuskaityti aplinkos kintamuosius.

Šias savybes palaiko daugelis programavimo kalbų. Tačiau dažniausiai CGI programos rašomos panaudojant C arba Perl programavimo kalbas bei paskutiniu laiku populiarėjančia PHP. Trumpai apie kiekvienos jų pranašumus ir trūkumus:

- **Perl** programavimo kalba buvo sukurta specialiai darbui su tekstu, be to ja parašytos programos yra itin greitai vykdomos.
- **C** programavimo kalba mažiau efektyvi, be to neturi patogių priemonių, kurios galėtų greitai ir patogiai nuskaityti duomenis iš HTML formų. Be to šia kalba parašytos taikomosios programos yra lėtesnės už analogiškas programas parašytas Perl kalba.
- **PHP** – kalba skirta būtent serverio išplėtimo programų rašymui, joje realizuotos tokios pat funkcijos kaip ir kitose kalbose, skirtose CGI kūrimui: duomenų perdavimas/nuskaitymas, dinaminių tinklapių turinio generavimas. Pagrindinis PHP privalumas – suderinamumas su daugeliu plačiai naudojamų domenų.bazių valdymo sistemų.

ASP

ASP (Active Server Pages) – Microsoft kompanijos sukurta technologija, skirta programavimui WWW tinkle. Palaikoma Windows platformos ir veikia IIS (Internet Information Server) aplinkoje. Pagal nutylėjimą palaiko Visual Basic Script ir Java Script programavimo kalbas (yra galimybė įdiegti papildomas, pavyzdžiui Perl), suteikia galimybę naudoti išorinius COM objektus. ASP technologija paremtų programinių produktų veikimo principas analogiškas CGI – tai dažniausiai serveryje veikiančios programos, skirtos dinaminiam WWW tinklapių generavimui (iš HTML formų paimama informacija, serveryje vykdomas apdorojimas, rezultatai perduodami vartotojui vėl HTML forma).

ASP technologija palaiko objektų panaudojimą – tiek įdiegtų pagal nutylėjimą, tiek parašytų vartotojo. Galimas įvairių papildomų komponentų įdiegimas (pvz. darbui su elektroniniu paštu).

Darbą su duomenų bazėmis ASP užtikrina naudodama dvi tarpines technologijas – ODBC ir ADO. Naudojant ODBC standartą galima dirbti su daugeliu Windows platformai pritaikytų DBVS (SQL Server, Oracle, Access, FoxPro). Palaikymas ir ryšio su DB parametru nustatymas vykdomas naudojantis ODBC priemonėmis (tvarkyklių bibliotekomis) Windows aplinkoje. ADO – rinkinys objektų, kurie pasiekiami iš ASP ir įgalina kreiptis į ODBC

palaikomas duomenų bazes. Praktiškai naudojami 2 objektai – *Connection*, ryšio su DB užmezgimui ir *Recordset*, informacijos perduodamos iš DB perėmimui.

PHP

Egzistuoja dvi pagrindinės sritys, kuriose naudojami PHP skriptai [8]:

- **Serverio išplėtimai.** Tai pagrindinė PHP taikymo sritis, ir tam reikalingi trys dalykai: PHP interpretatorius, tinklo serveris ir interneto naršyklė. PHP scenarijai naudojami dinaminiam interneto tinklapiams kurti (tinklapių turinio generavimas, duomenų transakcijos su duomenų bazėmis).
- **Iškvietimas komandinės eilutės pagalba.** Galima paleisti PHP skriptą nesinaudojant tinklo serveriu ir interneto naršykle. Tokiam PHP skripto iškvietimui reikalingas tik PHP interpretatorius, ir toks iškvietimo būdas dažniausiai naudojamas paprastam teksto apdorojimui.

PHP gali būti naudojamas daugelyje operacinių sistemų, įskaitant Linux, daugelį UNIX versijų (HP-UX, Solaris ir OpenBSD), Microsoft Windows, Msc OS X, RISC OS ir kt. Taip pat PHP yra palaikomas daugelio tinklo serverių, tokių kaip Apache, Microsoft Internet Information Server, Personal Web Server, Caudium, Xitami, OmniHTTPd ir daugelio kitų. Daugumai serverių PHP turi modulį palaikantį CGI standartą.

Naudojant PHP gaunama operacinių sistemų ir tinklo serverių pasirinkimo laisvė, taip pat galima rinktis tarp procedūrinio ir objektiškai orientuoto programavimo arba naudoti mišrų variantą.

PHP neriboja standartinis HTML informacijos išvedimas, nes į PHP galimybes įtraukti: grafikos, PDF failų ir Flash animacijos išvedimas. Taip pat galima laisvai išvesti bet koki tekstą.

Viena iš didžiausių ir labiausiai pastebimų PHP naujovių - platus palaikomų duomenų bazių spektras, kas labai supaprastina duomenų bazių publikavimą internete.

Šiuo metu PHP palaikomos DBVS:

Adabas D	Ingres	Oracle
Dbase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro	MsSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

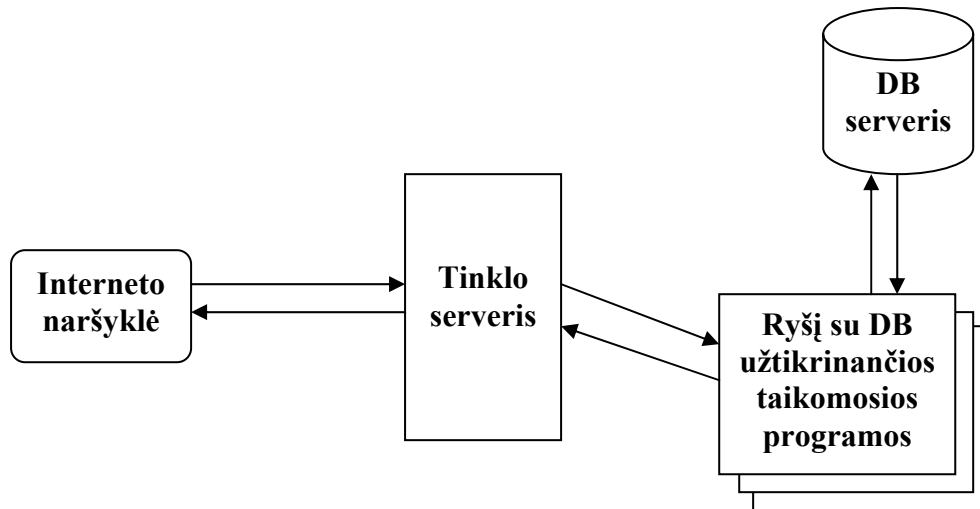
Taip pat PHP palaiko ODBC(Open Database Connectivity) standartą, kuris įgalina prisijungti prie bet kurios jį palaikančios duomenų bazės.

PHP gali komunikacijai naudoti tokius protokolus kaip LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM ir kt.

Egzistuoja Java objektų palaikymas, leidžiantis naudoti juos kaip PHP objektus, bei CORBA išplėtimas, naudojamas priėjimui prie nutolusių objektų.

5. METODO APRAŠYMAS

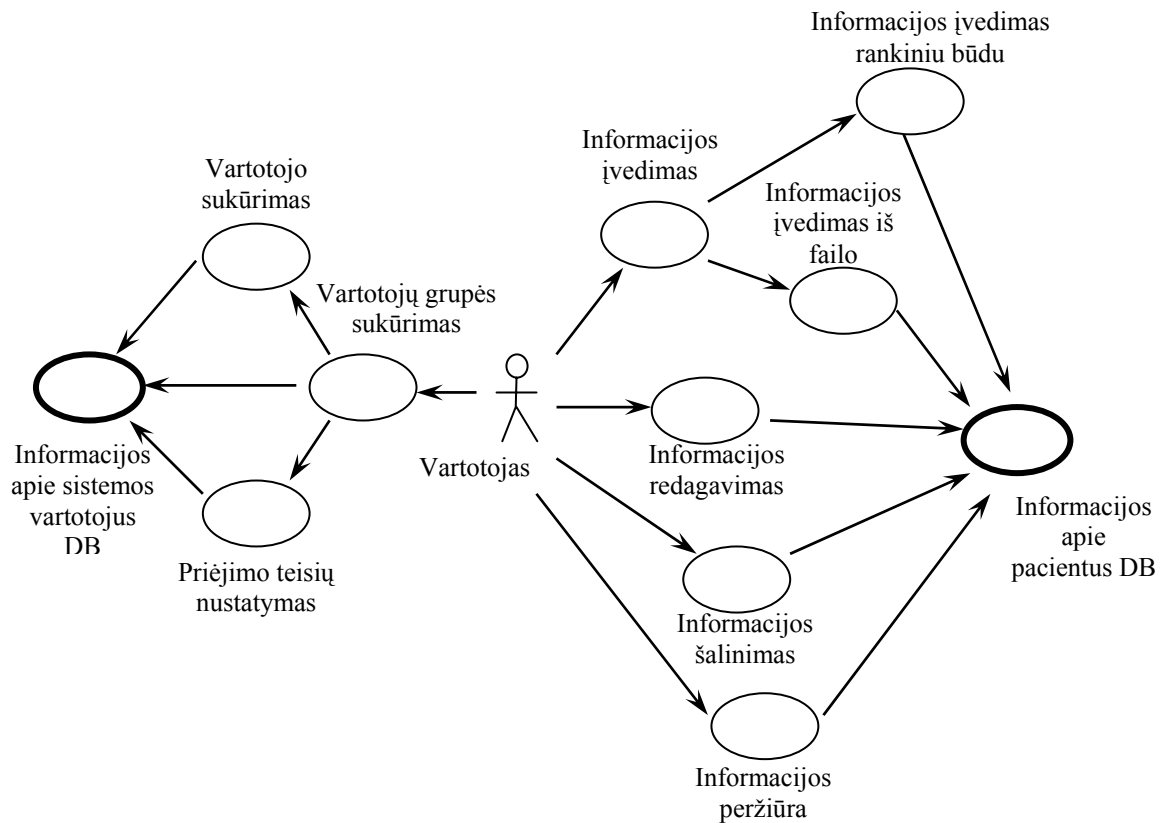
5.1. Apibendrintas sistemos aprašas



7 pav. Bendra sistemos schema.

Sistemos realizacijos pagrindas - „kliento - serverio“ architektūra. Vartotojo sąsaja realizuojama naudojant dinaminių interneto puslapių kūrimo priemones ir saugoma tinklo serveryje. Vartotojas prisijungia prie sistemos naudodamasis interneto naršykle. Vartotojo užklausa perduodama į serverį ir ten ją apdoroja sistemos serverinės dalies taikomosios programos. Šios programos pateikia vartotojo užklausa duomenų bazių valdymo sistemai ir užklausa rezultatus perduoda vartotojui dinamiškai sugeneruoto interneto tinklapio forma. Taip pat veikia ir duomenų į DB įvedimas ar keitimas, tik šiuo atveju į DBVS perduodamos ne užklausa informacijai gauti, o komanda nurodančios keisti vieną ar kitą duomenų bazės elementą.

Būtent vartotojo ryšį su duomenų baze užtikrinančių taikomųjų programų visuma (sistema) ir yra šiame darbe realizuojamas objektas.



8 pav. Sistemos panaudojimo atvejų diagrama.

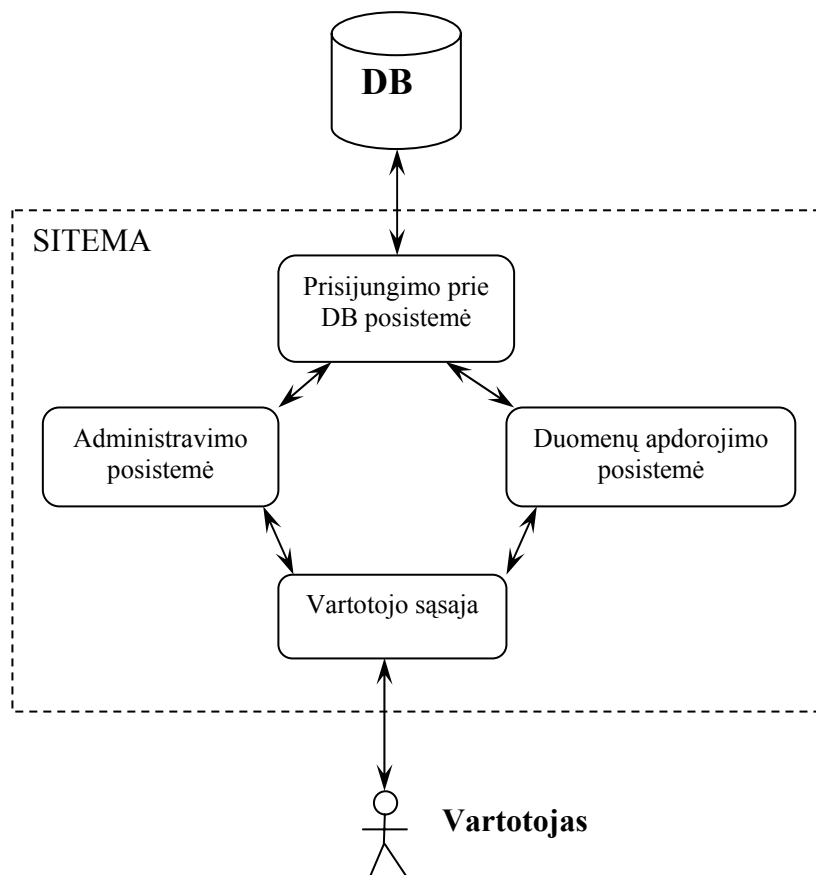
Šioje diagramoje parodytos sistemos funkcijos, kurias galima išskirti į dvi kategorijas:

- Sistemos vartotojų administravimas – sistemos administratoriui skirtų funkcijų rinkinys, įgalinantis kurti vartotojus, apjungti juos į vartotojų grupes bei priskirti naudojimosi sistema teises. Kaip galima pastebėti, šis funkcijų rinkinys naudojasi duomenų bazės dalimi, kuri naudojama tik informacijai apie vartotojus saugoti.
- Informacijos apie pacientus valdymas – šis funkcijų rinkinys savo apimtimi yra daug didesnis už anksčiau minėtą ir yra skirtas informacijos apie pacientus peržiūrai, įvedimui, redagavimui, šalinimui. Įvedimas gali būti vykdomas tiek rankiniu būdu, tiek automatiškai – pateikiant sistemai failą su paruoštais duomenimis. Šios funkcijos naudojamos naudotis didesniąja duomenų bazėje saugomos informacijos dalimi (informacija apie pacientus – klinikiniai įrašai, medicininių duomenų žinynai).

5.2. Sistemos programinės įrangos posistemės

Sistemos sudėtinės dalys:

- ❑ Prisijungimo prie duomenų bazės posistemė. Universalus modulis perduodantis sistemos generuojamas SQL užklausas į duomenų bazę ir gražinantis rezultatus.
- ❑ Vartotojų administravimo posistemė. Skirta užtikrinti, kad su sistema dirbs tik tie vartotojai, kuriems suteikta teisė ja naudotis, ir jie atliks tik tas operacijas, kurių atlikimui jiems suteikta teisė (vartotojų administravimas).
- ❑ Duomenų apdorojimo posistemė. Realizuoja informacijos įvedimo, redagavimo, šalinimo ir išrinkimo bei pateikimo vartotojui operacijas, naudodama prisijungimo modulyje realizuotas funkcijas, palaiko ryšį su vartotoju vartotojo sąsajos langų pagalba.
- ❑ Vartotojo sąsaja. Sistemos dalis, kurią tiesiogiai mato vartotojas. Tai funkcijų rinkinys, skirtas vartotojo darbo aplinkai sukurti ir komunikacijai tarp vartotojo ir kitų sistemos dalių užtikrinti (menu, įvedimo/išvedimo formos, ataskaitų langai ir pan.)



9 pav. Sistemos struktūra.

Sistemos struktūros schemoje (8 paveikslas) pavaizduotas sistemos posistemų išsidėstymas ir tarpusavio sąveika. Vartotojas prisijungęs prie sistemos pirmiausiai susiduria su vartotojo sąsaja, o per ją su administravimo posisteme, kuri patikrina, ar vartotojas gali naudotis sistema ir kokius veiksmus jam leidžiama sistemoje atlikti. Toliau vartotojas renkasi jam reikiamą duomenų apdorojimo posistemės dalį (pasirenka dominančios informacijos kategoriją atitinkantį meniu punktą) ir dirba su pateikta informacija. Kaip galima pastebėti administravimo ir duomenų apdorojimo posistemės yra glaudžiai susiję su prisijungimo prie duomenų bazės modulių, nes komunikavimui su duomenų baze yra naudojamos šioje posistemėje aprašytos darbo su duomenų baze funkcijos.

5.2.1. Prisijungimo prie duomenų bazės posistemė.

Prisijungimo prie duomenų bazės posistemė - tai funkcijų, transliuojančių vartotojo veiksmus sistemoje į duomenų bazių valdymo sistemai suprantamas SQL užklausas, rinkinys. Ši posistemė vartotojui yra visiškai nematoma, vartotojas mato tik jos darbo rezultatus, prieš tai apdorotus duomenų apdorojimo posistemės.

Prisijungimo prie duomenų bazės posistemė vykdo:

- prisijungimą prie duomenų bazės, vartotojui prisijungus prie sistemos;
- kitų sistemos modulių generuojamas užklausas transliuoja į SQL užklausas, perduoda jas į duomenų bazę, pateikia rezultatus.

5.2.2. Vartotojų administravimo posistemė.

Vartotojų administravimo posistemė vykdo:

- prie sistemos prisijungiančio vartotojo teisių tikrinimą;
- suteikia galimybes sistemos administratoriui kurti naujus vartotojus, suteikti jiems naudojimosi viena ar kita sistemos dalimi teises.

5.2.3. Duomenų apdorojimo posistemė.

Duomenų apdorojimo posistemė vykdo:

- ❑ bendravimą su vartotoju, naudojantis vartotojo sąsajos teikiamomis priemonėmis;
- ❑ vartotojo užklausų perdavimą prisijungimo prie duomenų bazės posistemei;
- ❑ užklausų rezultatų pateikimą vartotojui, per vartotojo sąsają.

Vartotojo veiksmai sistemoje, kurie gali būti realizuoti duomenų apdorojimo posistemėje:

- ❑ paieška - informacijos paieška duomenų bazėje pagal tam tikrus raktinius identifikatorius;
- ❑ įvedimas - naujų duomenų įvedimas į duomenų bazę;
- ❑ redagavimas - jau įvestos į duomenų bazę informacijos redagavimas, atnaujinimas;
- ❑ šalinimas - informacijos šalinimas iš duomenų bazės.

Šie veiksmai skirti darbui su pagrindine duomenų bazėje saugoma informacija ir būtent jie skirti daugumai vartotojų.

Žemiau išvardinti veiksmai turi būti prieinami tik sistemos administratoriui:

- ❑ naujo vartotojo sukūrimas;
- ❑ vartotojų grupės sukūrimas;
- ❑ vartotojų teisių nustatymas - galimų vartotojo veiksmų sistemoje apibrėžimas;
- ❑ informacijos apie vartotoją peržiūra/redagavimas/šalinimas;

5.2.4. Vartotojo sąsaja.

Vartotojo sąsaja tai funkcijų rinkinys kuris užtikrina sistemos ryšį su vartotoju. Šios funkcijos turi užtikrinti:

- ❑ sistemos sistemos grafinių elementų, skirtų komunikavimui su vartotoju, atvaizdavimą (menu langai, informacijos įvedimo/išvedimo formos, laukai, mygtukai ir pan.);

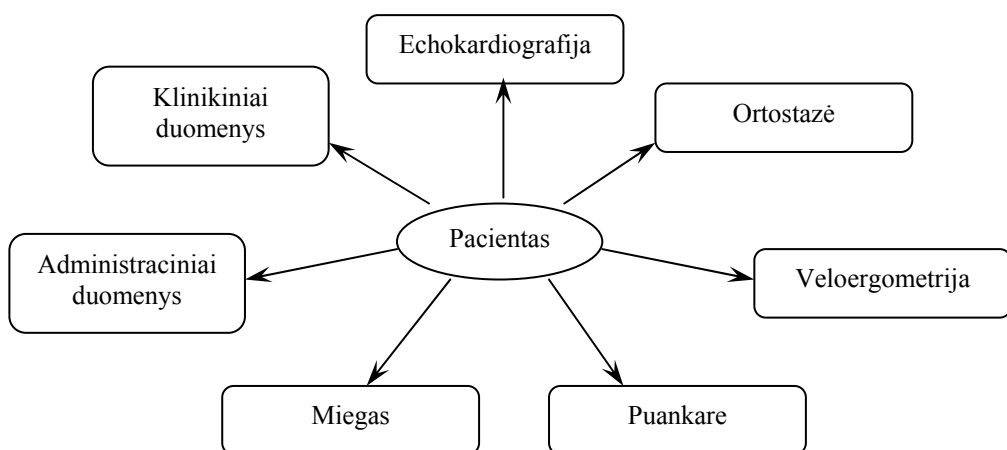
- vartotojo įvestos informacijos perdavimą duomenų apdorojimo bei administravimo posistemėms ir apdorotos informacijos pateikimą vartotojui suprantama forma.

5.3. Sistemos duomenų bazė.

Ne mažiau svarbi sistemos dalis – duomenų bazė, kurioje saugoma vartotojus dominanti informacija. Pagal pasirinktą modelį ir atsižvelgiant į konkretų sistemos taikymą (medicininės informacijos saugojimas) duomenų bazę turi sudaryti dvi dalys – dalis, skirta informacijai apie pacientus saugoti, bei dalis skirta efektyviam sistemos vartotojų administravimui užtikrinti.

5.3.1. Informacijos apie pacientus saugojimas.

Pagrindinė sistemos paskirtis – informacijos apie pacientus saugojimas. Duomenų bazėje ši informacija turėtų sudaryti apie 95 % viso informacijos kiekio. Kadangi projektuojama sistema yra specializuota , t.y. turi konkrečią paskirtį (informacijos apie pacientų, kuriems atlikta širdies operacija, reabilitacijos procesą saugojimas ir valdymas), bendradarbiaujant su **KMU Psichofiziologijos ir reabilitacijos institutu**, buvo sudaryta griežtai apibrėžta saugomų medicininių duomenų struktūra.



10 pav. Pacientų informacijos duomenų struktūra.

Kaip parodyta diagramoje (10 pav.), medicininė informacija duomenų bazėje suskirstyta į septynias pagrindines kategorijas. Kiekvienai kategorijai priskirta informacija detalizuojama

ir saugoma vienoje ar keliose duomenų bazės lentelėse. Iš viso duomenų bazę sudaro 64 lentelės (didelė jų dalis skirta pagalbinei informacijai – medikamentų, susirgimų žinynams ir pan.), kuriose atributų kiekis varijuoja nuo 2 iki 67. Žemiau pateikiamas vieno *Ortostazės* kategorijos tyrimo lentelės aprašymas.

1 lentelė. Tyrimo *OVperproc (Ortostazė-Stovi)* informacijos struktūra.

Nr.	Pavadinimas	Trumpas vardas	Tipas	Reikšmių aibė
1.	<u>Paciento kodas</u>	Kodas	char(6)	
2.	<u>Tyrimo pobūdis</u>	T_pob	ryšys į Tyrimo pobūdis	
3.	<u>Tyrimo numeris</u>	T_nr	sveikasis	1-9
4.	<u>Mėginio tipas</u>	M_tip	char(1)	O, V
5.	RR reikšmė B taške	RRb	sveikasis	300-1500
6.	Skirtumas tarp 1-ojo etapo RR vidurkio ir RRB (ΔRR_B)	dRRb	sveikasis	0-1000
7.	Skirtumas tarp 1-ojo etapo RR vidurkio ir RRB % ($\Delta RR_B\%$)	dRRBproc	sveikasis	0-100
8.	RR reikšmė C taške	RRc	sveikasis	300-1500
9.	Laikas tarp taškų A ir B	lab	sveikasis	1-30
10.	Laikas tarp taškų A ir C	lac	sveikasis	0-100
11.	Atsistatymo laikas (iki 75%)	l75proc	sveikasis	0-300
12.	Pereinamojo proceso klasė	RGO	sveikasis	1-110
13.	Sinusinio ritmo klasė	RG	sveikasis	1-10

Kaip galima pastebėti, didžiausią informacijos dalį medicininuose įrašuose sudaro tam tikro skaitinio diapazono reikšmės.

5.3.2. Informacijos apie sistemos vartotojus saugojimas.

Mažesnioji duomenų bazės dalis (apie 5 %), skirta informacijai apie sistemos vartotojus saugoti. Jos sudedamosios dalys:

- informacija apie vartotojus (vartotojo asmeniniai duomenys, prisijungimo vardas, slaptažodis, priklausomybė vartotojų grupei);
- informacija apie vartotojų grupes (grupės aprašymas, grupei suteiktos teisės sistemoje);

Ši informacija turėtų būti naudojama vartotojų autorizacijai ir kontrolei ir prieinama tik sistemą administruojantiems asmenims.

5.4. Įgyvendinimo priemonių parinkimas.

Projektui įgyvendinti buvo pasirinktos šios priemonės:

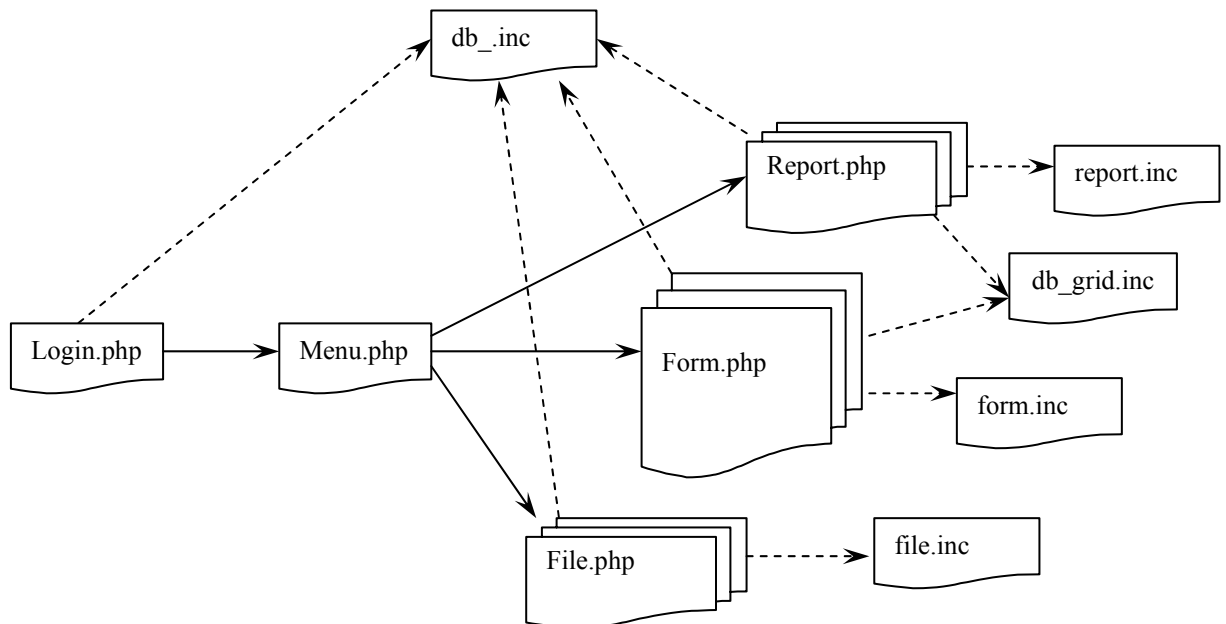
- **Duomenų bazei** – Oracle 8i DBVS. Ms SQL Server buvo atmestas dėl priklausomybės tik Windows platformai ir iš to sekančio mažesnio darbo greičio bei patikimumo[16]. Kadangi Oracle sistema palaikoma tiek Unix, tiek Windows platformų, jos pasirinkimas pasirodė universaliausias. MySQL, nors ir pritaikyta darbui su būtent tokio tipo, kaip projektuojamoji, sistemomis ir vertinama dėl didelio darbo greičio taip pralaimėjo Oracle, nes paaiškėjo, kad jos darbo greitis labai sumažėja, augant duomenų bazės dydžiui (MySQL optimaliausiai dirba tik su mažos arba vidutinės apimties duomenų bazėmis bei jos greitis pasireiškia tik išrenkant duomenis, tuo tarpu duomenų įvedimas arba keitimas užtrunka daug ilgiau lyginant su Oracle, kuri pritaikyta darbui su didelės apimties duomenų bazėmis [11] [14]).
- **Vartotojo sąsajai** – PHP, kurios pagrindiniai privalumai prieš kitas technologijas (CGI, ASP) [9] [10] [12] [13]:
 - greitis;
 - daugiaplatformiškumas (Unix/Windows);
 - DBVS palaikymas;
 - atviras kodas.

6. EKSPERIMENTINĖ DALIS

6.1. Sukurtos programinės įrangos aprašymas

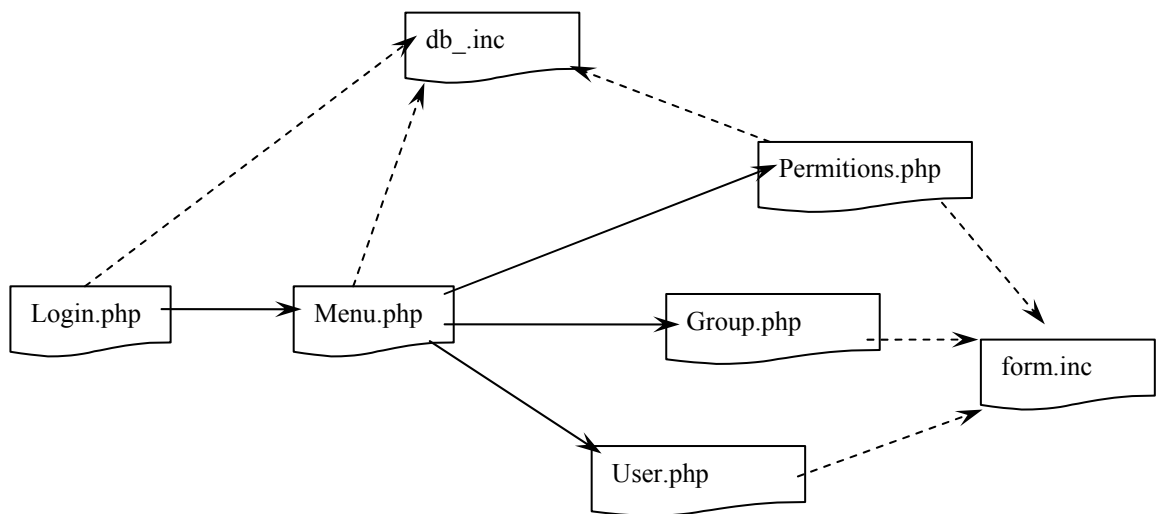
Naudojantis 5 skyriuje aprašyta metodika buvo sukurta programinė įranga, kurios pagrindinis uždavinys – priėjimo prie informacijos, saugomos centralizuotoje duomenų bazėje, užtikrinimas nutolusiems vartotojams. Naudojant objektinę programavimo technologiją, PHP programavimo kalbos priemonėmis, buvo realizuota visuma sudėtinių projektuojamos sistemos dalių (objektų), kurie apjungti į vieną visumą sudarė sistemą atitinkančią 5 skyriuje aprašytą modelį. Kadangi sistema buvo realizuota PHP priemonėmis, gauta unikali sistema funkcionuojanti tiek Windows, tiek UNIX/Linux operacinėse sistemose, bei lengvai suderinama su dauguma populiarių DBVS.

Žemiau pateiktose schemose parodyti pagrindiniai sistemos komponentai bei jų tarpusavio ryšiai. Ištininės linijos schemoje vaizduoja tiesioginį sistemos dalių veikimo eiliškumą (t.y. kuri dalis ką suaktyvina, vartotojui dirbant su sistema). Brūkšninės linijos – sistemos komponentų tarpusavio priklausomybės (objektų panaudojimas, kiekviena sistemos dalis yra susieta su vienu ar keliais objektais ir naudojami tame objekte aprašytomis funkcijomis).



11 pav. Sistemos komponentų ir jų tarpusavio sąveikos schema (vartotojo funkcijos).

Šioje schemoje pavaizduotos sistemos dalys, kurios užtikrina vartotojui priėjimą prie duomenų bazėje saugomos informacijos apie pacientus. Pirmiausiai vartotojas prisijungia prie sistemos (login.php), prisijungus patenka į pagrindinį meniu (menu.php), kuriame pasirinkęs reikiamą punktą iškviečia informacijos įvedimo/redagavimo/šalinimo formą (form.php), ataskaitų langą (report.php) arba automatinio duomenų įvedimo langą (file.php). Kiekvienas iš iškviestų komponentų formuojamas naudojant atitinkamą funkcijų biblioteką.



12 pav. Sistemos komponentų ir jų tarpusavio sąveikos schema (administratoriaus funkcijos).

Ši schema parodo tik sistemos administratoriui prieinamas funkcijas realizuojančius sistemos komponentus. Kaip ir eilinis sistemos vartotojas, administratorius jungiasi prie sistemos, patenka į pagrindinį meniu ir gali atlikti visas kitiems vartotojams priskirtas funkcijas. Taipogi administratorius papildomai prieina prie tokių funkcijų kaip naujo vartotojo arba vartotojų grupės sukūrimas, vartotojų teisių nustatymas.

6.1.1. Prisijungimo prie duomenų bazės posistemė.

Prisijungimo prie duomenų bazės posistemė - tai funkcijų, perduodančių SQL užklausas duomenų bazių valdymo sistemai rinkinys. Vartotojas mato tik jos darbo rezultatus, prieš tai apdorotus duomenų apdorojimo posistemės.

Prisijungimo prie duomenų bazės modulis susideda iš:

- ❑ **db.inc** - prisijungimo prie duomenų bazės ir komunikavimo su duomenų baze funkcijų biblioteka. Joje taip pat nurodyti prisijungimo prie konkrečios DBVS ir duomenų bazės atributai.

Kadangi sistema buvo projektuojama siekiant maksimalaus suderinamumo su dauguma duomenų bazių valdymo sistemų, ši sistemos objektą sudaro tik trys funkcijos, kurios skirtos:

- ❑ prisijungimui prie DBVS ir konkrečios duomenų bazės;
- ❑ galutiniam SQL užklausių formavimui ir perdavimui į DBVS;
- ❑ rezultatų iš DBVS priėmimui ir perdavimui duomenų apdorojimo posistemėi.

Ekspperimentų metu nustatyta, kad būtent šiame objekte, keičiant komunikavimo su DBVS funkcijas galima pasiekti kad sistema dirbtų su įvairiomis DBVS, nekeičiant visų kitų sistemos komponentų.

6.1.2. Vartotojų administravimo posistemė.

Vartotojų administravimo posistemė susideda iš:

- ❑ **login.php** - prisijungimo prie sistemos forma (tikrinimas ar yra toks vartotojas, pagrindinio meniu pateikimas atsižvelgiant į vartotojo teises).
- ❑ **user.php** – forma skirta vartotojų administravimui. Naudoja objekto form.inc metodus.
- ❑ **group.php** – vartotojų grupės administravimo forma. Naudoja objekto form.inc metodus.
- ❑ **permissions.php** - vartotojų teisių suteikimo/keitimo forma. Naudoja objekto form.inc metodus.

Šis sistemos komponentas prieinamas tik sistemos administratoriui.

6.1.3. Duomenų apdorojimo posistemė.

Duomenų apdorojimo posistemė susideda iš:

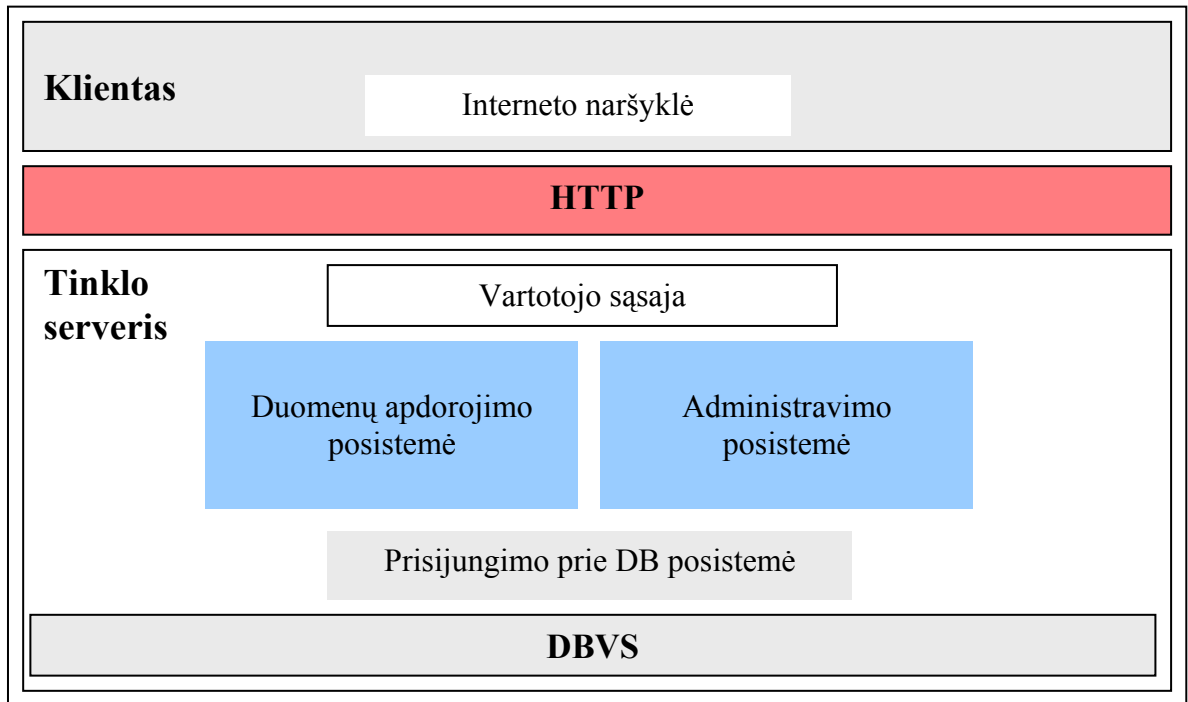
- **form.php** - formos, kurios dinamiškai generuojamos priklausomai nuo pasirinkto veiksmo (naujas įrašas\redagavimas\šalinimas). Iš jų vartotojo įvesta informacija generuojama į SQL užklausas ir perduodama prisijungimo prie DB posistemai arba į formą perduodama informacija kurią pateikė DBVS. Naudoja objektų *form.inc* ir *db_grid.inc* metodus.
- **report.php** - dinamiškai generuojamos ataskaitų formos. Vartotojui pateikiama forma su pagal užduotus kriterijus išrinktos informacijos sąrašu. Naudoja objektų *report.inc* ir *db_grid.inc* metodus.
- **file.php** – forma skirta automatiniam duomenų įvedimui (joje nurodžius failo vardą, paleidžiamos objekto *file.inc* procedūros failo turinį paverčiančios į SQL užklausas ir perduodančios jas prisijungimo prie DB posistemai). Naudoja objekto *file.inc* metodus.

6.1.4. Vartotojo sąsajos posistemė.

Vartotojo sąsajos posistemė susideda iš:

- **menu.php** – pagrindinis sistemos meniu, kurio punktų aprašas saugomas duomenų bazėje. Pagal prie sistemos prisijungusio vartotojo teises formuojamas individualiai (vartotojai turintys skirtingas darbo sistemoje teises, negali naudotis vienais ar kitais meniu punktais). Informaciją apie meniu formavimą ši sistemos dalis gauna iš *db.inc* objekto.
- **form.inc** - funkcijų biblioteka kurioje aprašyti sistemos formų bei langų elementai, kurie naudojami formų generacijai (formos parametrai, įvedimo laukai, mygtukai, informacijos įvedimo kontrolės procedūros).
- **report.inc** - funkcijų biblioteka kurioje aprašyti ataskaitų elementai ir jų formavimo procedūros. Savo struktūra ir sudedamosiomis dalimis objektas panašus į *form.inc*.
- **db_grid.inc** - funkcijų biblioteka kurioje aprašytos duomenų pateikimo vartotojui funkcijos (įrašų išrinktų duomenų bazėje pateikimas lentelės forma).
- **file.inc** - funkcijos skirtos duomenų nuskaitymo iš failo formai generuoti.

6.2. Sukurtos programinės įrangos vieta sistemoje



13 pav. Sistema.

6.3. Eksperimentinės programinės įrangos veikimas

6.3.1. Įvadas.

Šiame skyriuje bendrais bruožais aprašomas sukurtos programinės įrangos veikimas.

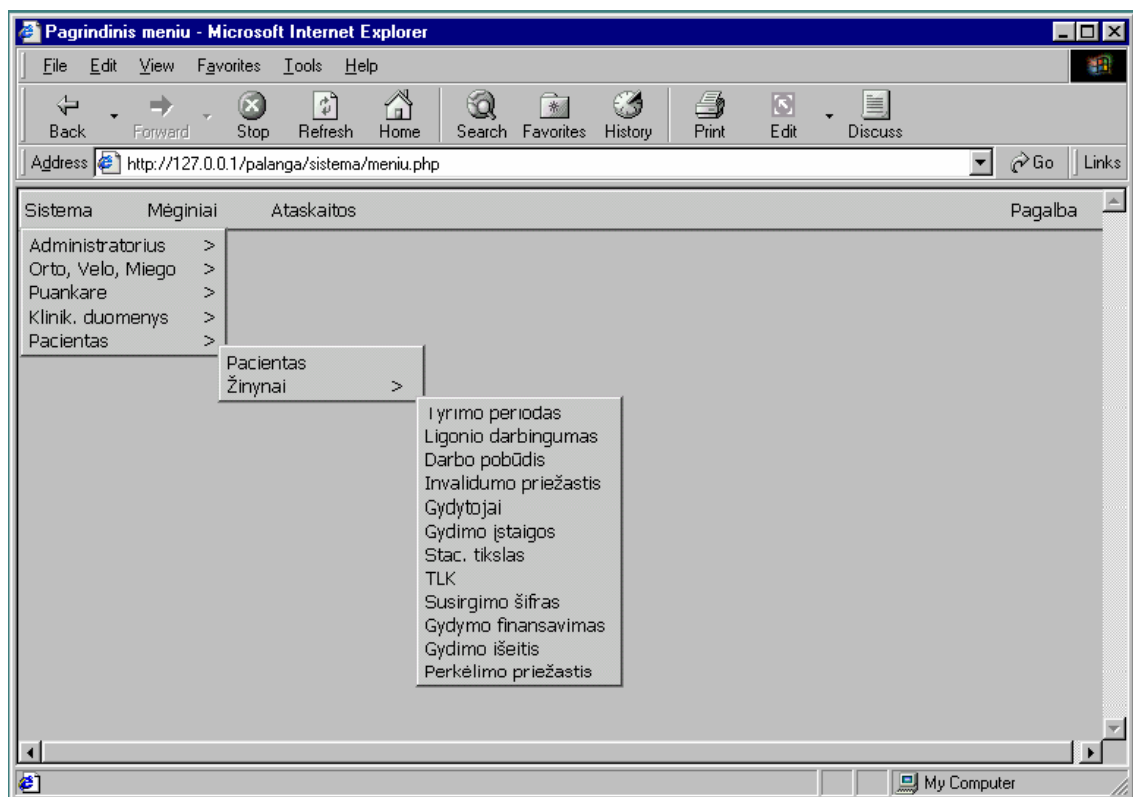
Žemiau aprašoma eksperimentinė programinė įranga buvo įgyvendinta:

- Vartotojo sąsaja – PHP priemonėmis.
- Duomenų bazė – Oracle 8i duomenų bazių valdymo sistemoje.

Sistemos veikimas buvo testuojamas tiek Windows, tiek Linux operacinių sistemų aplinkose. Sistema be didesnių pakeitimų efektyviai funkcionavo abiejose terpėse.

6.3.2. Pagrindinis meniu.

Prisijungimas prie sistemos vykdomas interneto naršyklės adreso lauke surinkus URL adresą (paprastas jungimasis prie tinklapio). Pasirodžius prisijungimo langui įvedamas vartotojo vardas ir slaptažodis. Jei vartotojas sistemoje užregistruotas, jam užkraunamas pagrindinio meniu langas.



14 pav. Pagrindinis meniu.

Priklausomai nuo vartotojo teisių, ne visi meniu punktai būna aktyvūs. Jei vartotojas neturi teisės naudotis vienokia ar kitokia informacija, jam neleidžiama pasirinkti atitinkamo meniu punkto.

Meniu hierarchija

Visi meniu punktai suskirstyti į keturis lygius:

□ **I lygis:**

- **Sistema** - papildomos informacijos, užtikrinančios sistemos darbą valdymas.
- **Mėginiai** – pacientui atliekamų tyrimų informacijos valdymas.
- **Ataskaitos** - informacijos apie atliktus tyrimus peržiūra.

□ **II, III, IV lygiai:**

- Šiuose lygiuose meniu punktai sugrupuoti pagal loginę prasmę (mėginio tipą), pvz.: Sistema -> Pacientas -> Žinynai (informacija reikalinga pildant medicininės informacijos žinynus, kuriuose saugoma informacija naudojama sistemos darbe).

6.3.3. Sistemos administravimas.

Vartotojų teisės užtikrina, kad kiekvienas vartotojas atliks tik jam leistas duomenų apdorojimo operacijas.

Sistemoje realizuotas grupinių teisių tipas: teisės suteikiamos ne kiekvienam vartotojui atskirai, o vartotojų grupei ir galioja visiems tos grupės nariams. Teisių veikimo principas - leidžiama viskas, kas neuždrausta, t.y., kad vartotojas galėtų atlikti vienas ar kitas duomenų apdorojimo operacijas, jo grupei turi būti suteiktas leidimas tų operacijų atlikimui (pvz., jei sukuriama nauja vartotojų grupė, jos nariams yra uždraustos visos duomenų apdorojimo operacijos, kol nenustatomos šios grupės teisės).

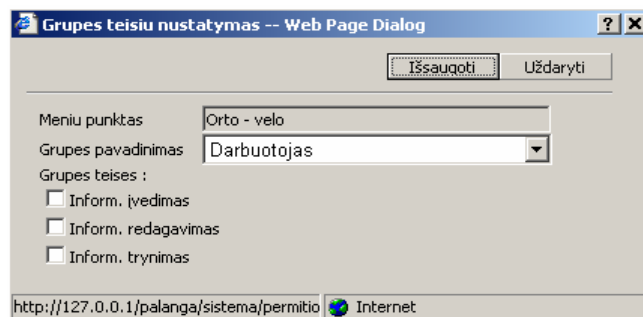
Teisių nustatymas

Teisių nustatymo forma atidaroma, paspaudus dešinę pelės klavišą ant norimo meniu punkto.

Teises nustatyti gali tik administratoriaus grupės vartotojai. Administratoriaus grupės vartotojams pagal nutylėjimą suteiktos neribotos teisės, kitų grupių vartotojams teises būtina nustatyti.

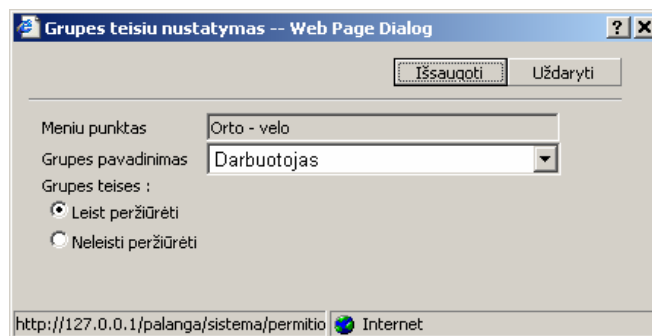
Teisių nustatymo formos

Sistemoje naudojamos teisių nustatymo formos:



A forma

15 pav. Priėjimo prie informacijos valdymo forma.



B forma

16 pav. Priėjimo prie ataskaitų peržiūros forma.

Formų elementai:

- ❑ mygtukai: „Išsaugoti“ - išsaugoti vartotojo teises; „Uždaryti“ - atšaukti operaciją.
- ❑ pasirinkimo laukas „Grupės pavadinimas“ - naudojamas pasirinkti vartotojų grupę, kuriai nustatomos teisės.
- ❑ nustatymo laukai :
 - *Priėjimo prie informacijos valdymo forma:*
 - „Inform. įvedimas“ - leisti/neleisti įvesti duomenis;
 - „Inform. redagavimas“ - leisti/neleisti redaguoti duomenis;
 - „Inform. trynimas“ - leisti/neleisti šalinti duomenis;
 - *Priėjimo prie ataskaitų peržiūros forma:*
 - „Leisti peržiūrėti“ - leisti naudotis ataskaitos duomenimis;
 - „Neleisti peržiūrėti“ - neleisti naudotis ataskaitos duomenimis;

6.3.4. Duomenų įvedimo, redagavimo ir šalinimo formos.

Informacijos įvedimui, redagavimui, šalinimui sistemoje naudojamos formos, kurios atsidaro naujame lange „suaktyvinus meniu punktą.

Sistemoje naudojamų formų tipai:

Pavadinimas	Aprašymas
Administratorius	Leidžiama viskas
Darbuotojas	Leidžiama viskas išskyrus vartotojų administravimą
Lankytojas	Leidžiama peržiūrėti ataskaitas

I forma

17 pav. I forma.

II forma

18 pav. II forma.

Išrinkti Atšaukti

Paciento kodas ?? Tyrimo pobūdis 1 Meteo-ryte (Ortostazė) pirmadienis

Tyrimo numeris Etapo kodas 1 Ortostazė-Guli

Įvedimo laukas Pasirinkimo/įvedimo laukas Pasirinkimo laukas

III forma

19 pav. III forma.

IV forma

20 pav. IV forma.

Formų paskirtis:

- operacijos(įvedimo, redagavimo, trynimo) pasirinkimas - formos I, II;
- įrašo išrinkimas - formos I, III;
- duomenų įvedimas, redagavimas, šalinimas - forma IV.

Formų elementai:

- **mygtukai** - naudojami operacijai pasirinkti (naujo įrašo sukūrimas - „Naujas“, įrašo redagavimas - „Redaguoti“, įrašo šalinimas - „Trinti“, įrašo išrinkimas (redagavimo arba trynimo operacijoms) - „Išrinkti“, įvestos/pakeistos informacijos išsaugojimas - „Išsaugoti“, operacijos atšaukimas - „Atšaukti“, formos uždarymas - „Uždaryti“;
- **įvedimo laukai** - naudojami duomenų įvedimui;
- **pasirinkimo laukai** - naudojami informacijos pasirinkimui iš sąrašo. Sąrašas gali būti suformuotas dinamiškai iš žinyno arba statiškai aprašytas programoje;
- **įvedimo/pasirinkimo laukai** - naudojami duomenų išrinkimui iš žinyno. Išrinkimas gali būti atliekamas dviem būdais :
 - įvedant įrašo kodą;
 - paspaudus mygtuką .

Naudojant antrą išrinkimo metodą, atsirado sąrašas, kuriame galima pasirinkti norimą įrašą.

- **žinyno įrašų sąrašas** - naudojamas atvaizduoti žinyne saugomą informaciją, bei įrašo išrinkimui redagavimo ar trynimo operacijoms.

Duomenų kontrolė formose.

Duomenų kontrolė atliekama dviem etapais:

- duomenys tikrinami jų įvedimo metu;
- kontroliuojamas informacijos pilnumas ir unikalumas duomenų išsaugojimo metu.

Pirmas duomenų kontrolės etapas

Šiame etape, atsižvelgiant į duomenų tipą, tikrinamas duomenų formatas ir reikšmė.

Sistemoje realizuoti duomenų tipai, jų formatai ir kontrolė:

- tekstas:
 - formatas: bet kokių simbolių, išskyrus „“, junginys;
 - kontrolė: tekstinės informacijos ilgis;
- sveikas skaičius:
 - formatas: simbolių - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 - junginys;
 - kontrolė: skaičiaus reikšmė;
- realus skaičius:
 - formatas: simbolių - ., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 - junginys;
 - kontrolė: skaičiaus reikšmė;
- data:
 - formatas: pilnas - YYYY.MM.DD; sutrumpintas - YY.MM.DD ir YYMMDD; čia Y - metų skaitmuo, M - mėnesio skaitmuo, D - dienos skaitmuo.
 - kontrolė: datos reikšmė;
- laikas:
 - formatas: pilnas - HH:MM:SS arba HH:MM; sutrumpintas - HHMMSS ir HHMM; čia H - valandos skaitmuo, M - minutės skaitmuo, S - sekundės skaitmuo.
 - kontrolė: laiko reikšmė;
- mišrus duomenų tipas:
 - formatas: tekstas arba skaičius, pvz.: pildant kai kuriuos žinytus, atributo reikšmė gali būti įvedama tiek skaičiumi(25-500), tiek tekstu („neg“);
 - kontrolė: duomenų reikšmė;
- žinyto įrašo kodas:
 - formatas: priklauso nuo žinyto;
 - kontrolė: kodo reikšmė;

Jeigu duomenų formatas ar reikšmė neteisinga, vartotojas įspėjamas ir jam leidžiama pakartoti įvedimą .

Antras duomenų kontrolės etapas

Šiame etape tikrinam ar pateikta visa būtina informacija ir ar informacija nesidubliuoja (pvz., antra kartą bandoma įvesti tą patį tyrimą). Jei pateikta nepilna informacija, t.y., ne visi būtini laukai užpildyti, išvedamas pranešimas: „Tuščias laukas“ ir kursorius pastatomas į tuščią lauką. Informacijos dubliavimosi atveju duomenys neišsaugomi.

Duomenų apdorojimas

Informacijos apdorojimas susideda iš šių operacijų:

- duomenų įvedimas į sistemos duomenų bazę;
- duomenų redagavimas;
- duomenų šalinimas;
- duomenų išrinkimas ir pateikimas vartotojui.

Duomenų įvedimas

Duomenų įvedimas atliekamas naudojant duomenų įvedimo formas (I, II, III, IV formos žr. 6.3.3. skyrių). Šis būdas gali būti naudojamas visiems papildomos informacijos bei mėginių žinynams pildyti.

Papildomos informacijos (išskyrus paciento kortelę) įvedimui naudojamos I ir IV formos (žr. 6.3.3. skyrių).

Mėginių įvedimui ir paciento kortelės pildymui naudojamos II, III ir IV formos (žr. 6.3.3. skyrių).

Pildant paciento kortelę, sistema patikrina paciento kodą ir, jei toks kodas jau egzistuoja, išvedamas pranešimas apie jau esamą įrašą. Įvedus paciento asmens kodą, sistema duomenų bazėje ieško paciento su tokiu asmens kodu ir, jei suranda, užpildo dalį formos laukų. Neaktyvių laukų pildyti nereikia.

Duomenų redagavimas

Papildomos informacijos (išskyrus paciento kortelę) redagavimui naudojamos I ir IV formos (žr. 6.3.3. skyrių).

Mėginių ir paciento kortelės redagavimui naudojamos II, III ir IV formos (žr. 6.3.3. skyrių).

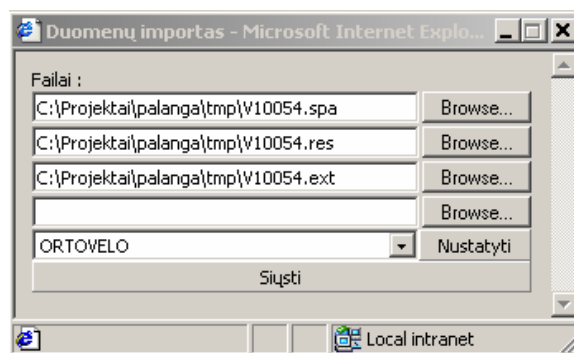
Duomenų šalinimas

Papildomos informacijos (išskyrus paciento kortelę) šalinimui naudojamos I ir IV formos (žr. 6.3.3. skyrių).

Mėginių ir paciento kortelės įrašų šalinimui naudojamos II, III ir IV formos (žr. 6.3.3. skyrių).

Duomenų įvedimas iš failo

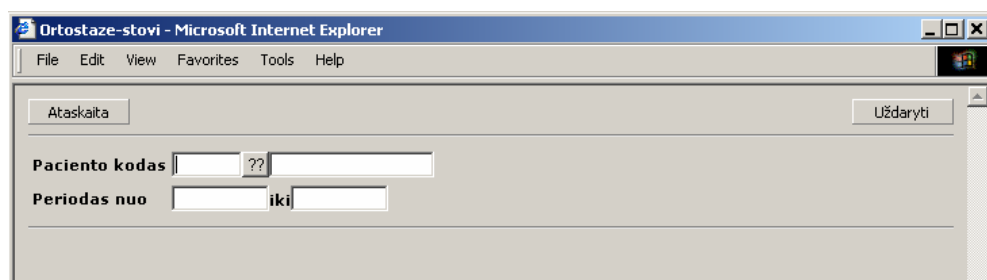
Sistemoje taip pat realizuotas automatinis duomenų įvedimas iš medicininės įrangos generuojamų tekstinių failų. Jei vartotojas pasirenka šį įvedimo būdą, jam tereikia nurodyti failus, iš kurių išrinkta informacija turi būti įvesta į duomenų bazę.



21 pav. Informacijos įvedimo iš failo forma.

6.3.5. Ataskaitos.

Ataskaitose sąrašo forma pateikiami visų konkrečiam pacientui atliktų, tam tikros rūšies tyrimų informacija:



22 pav. Informacijos išrinkimo forma.

Formos elementai:

- mygtukai: „Ataskaita“ - ataskaitos sukūrimas; „Uždaryti“ - uždaryti informacijos formą;
- įvedimo/pasirinkimo laukas (žr. 6.3.3. skyrių): „Pacientas“ - paciento išrinkimui (privalomas);

- įvedimo laukai(žr. 6.3.3. skyrių): „Periodas nuo“ ir „iki“ - tyrimų periodui nurodyti (neprivalomi);

Atlikus informacijos išrinkimą vartotojui pateikiama sugeneruota ataskaita.

Ortostazė - Stovi											
Pacientas	Nikolajus Novikovas										
Tyrimo pobūdis	Tyrimo numeris	Mėginio tipas	RR reikšmė B taške	Skirtumas tarp 1-ojo etapo RR vidurkio ir RRB	Skirtumas tarp 1-ojo etapo RR vidurkio ir RRB %	RR reikšmė C taške	Laikas tarp taškų A ir B	Laikas tarp taškų A ir C	Atsistatymo laikas (iki 75%)	Pereinamojo proceso klasė	Sinusinio ritmo klasė
Meteo-ryte (Ortostazė) pirmadienis	1	O	1000	100	100	1000	10	100	100	100	10
TESTAS	1	V	862	62	7	924	26	28	2	7	5
TESTAS	2	V	862	62	7	924	26	28	2	7	5
TESTAS	3	V	862	62	7	924	26	28	2	7	5

23 pav. Ataskaitos pavyzdys.

7. IŠVADOS

Šiame darbe buvo iškelta pagrindinė problema – sukurti sistemą leidžiančią per globalų tinklą prieiti prie centralizuotos duomenų bazės ir efektyviai naudotis toje bazėje saugoma informacija.

Nagrinėti šie problemos aspektai:

- Sistemos prieinamumo internete realizavimas.
- Sistemos vartotojui teikiamų paslaugų realizavimas.
- Sistemos ir jos vartotojų administravimas.

Išnagrinėti tokių sistemų realizavimo principai, apibrėžiantys duomenų bazių publikavimo internete galimybes ir metodiką, naudojamas priemonės.

Remiantis išnagrinėtomis metodikomis buvo sukurtas informacinės sistemos kūrimo modelis. Buvo aprašyti pagrindiniai sistemos elementai, jų veikimo principai, bendri reikalavimai atskiroms dalims.

Sukurtam modeliui pagrįsti buvo sukurta specializuota biomedicininė informacinė sistema, skirta gydymo įstaigoms, t.y. duomenų apie pacientus registracijai ir efektyviam valdymui.

Sukurta sistema pilnai atitiko kūrimo modelyje iškeltus reikalavimus ir tokiu būdu įrodė pasirinkto metodo ir pasirinktų įgyvendinimo priemonių efektyvumą.

Remiantis atliktu tyrimu buvo išskirtos šios rekomendacijos tolesniam sistemos tobulinimui:

- Sistemos komunikacijai su DBVS buvo panaudotos paprasčiausios priemonės, kurios pilnai neišnaudojo didesnės dalies DBVS galimybių. Nors šios priemonės sudarė galimybę panaudoti sistemą su maksimaliai dideliu kiekiu šiuo metu naudojamų DBVS, ateityje rekomenduotina papildyti prisijungimo prie duomenų bazės posistemę labiau specifinėmis funkcijomis ir efektyviau panaudoti DBVS galimybes.
- Atsižvelgiant į aukščiau minėtą punktą (platesnį DBVS galimybių panaudojimą), patartina papildyti sistemą moduliais leidžiančiais apdoroti platesnį duomenų saugojimo formatų spektrą (grafinė informacija, vaizdo, garso įrašai).

8. LITERATŪRA

1. William B. Hanlon, Ethan F. Fener, Jeffrey W. Downs. Data Storage and Management Requirements for the Multimedia Computer-Based Patient Medical Record. [žiūrēta: 2003-05-15]. Prieiga per internetą: <http://www.computer.org/conferences/mss95/Data Storage and Management Requirements for the Multimedia .htm>.
2. KMU Psichofiziologijos ir rehabilitacijos institutas.. [žiūrēta: 2003-05-10]. Prieiga per internetą: <http://www.pri.kmu.lt/telemedicina/index2.html>.
3. Kurt Lang, Jeff Chow. Database publishing on the web and intranets. The Coriolis Group, Inc. 1996
4. David M.Kroenke. Database processing. Pearsen Education Inc. 2002
5. William Page, David Austin, Nicholas Chase. Using Oracle 8/8i. Que Corporation. 1999
6. В.А. Матросов, А.О. Сергеев, М.П. Чаунин. HTML 4.0. БХВ - Петербург. 1999
7. Luke Weeling, Laura Thomson. PHP and MySQL web deveopment. Sams. 2000
8. PHP Manual. [žiūrēta: 2002-04-20]. Prieiga per internetą: <http://lt.php.net/manual/en>
9. 7 Reasons Why PHP is Better than ASP. [žiūrēta: 2003-05-10]. Prieiga per internetą: http://php.weblogs.com/php_asp_7_reasons.
10. Comparing PHP with Perl for Dynamic Web Pages. [žiūrēta: 2003-05-10]. Prieiga per internetą: http://php.weblogs.com/php_versus_perl.

11. Flying with Oracle and MySQL: Benchmarking Performance. [žiūrēta: 2003-05-10].
Prieiga per internetą: http://php.weblogs.com/oracle_mysql_performance.
12. PHP versus ASP. [žiūrēta: 2003-05-10].
Prieiga per internetą: http://php.weblogs.com/php_vs_asp.
13. PHP vs CGI. [žiūrēta: 2003-05-10].
Prieiga per internetą: <http://www.javascriptkit.com/howto/phpcgi.shtml>.
14. SQL Server vs MySQL. [žiūrēta: 2003-05-12]. Prieiga per internetą:
http://www.mssqlcity.com/Articles/Compare/sql_server_vs_mysql.htm.
15. Oracle Inside Out - Linux vs Windows. [žiūrēta: 2003-05-12]. Prieiga per internetą:
<http://www.interealm.com/technotes/roby/pentmark.html>.
16. Oracle vs. SQL Server. [žiūrēta: 2003-05-12]. Prieiga per internetą:
http://www.clientsandprofits.com/sql/SQL_Oracle_v_SQLserver.html.
17. Scott F. Wilson, Bruce Maples, Tim Landgrave. Analyzing requirements and defining solution architectures.
Microsoft Corporation. 1999