

ŠIAULIŲ UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Adomas Buškus
Informatikos specialybės II magistrantūros (gam.) kurso (neakivaizdinis skyrius) studentas

KLIENTŲ APTARNAVIMO SISTEMŲ TYRIMAS

MAGISTRO DARBAS

Darbo vadovas:
Prof. G. Kulvietis

Recenzentas:
Lekt. V. Giedrimas

Šiauliai, 2006 - 2007 m.m.

Turinys

I.Įvadas.....	4
II.Teorinė dalis.....	5
1)CRM sistemų apibrėžimas, raida.....	5
2)Sistemų taikymas.....	6
3)Populiariausių CRM sistemų apžvalga.....	7
4)CRM sistemose naudojamų technologijų ir realizavimo metodų analizė.....	9
4.1 Duomenų bazių valdymo sistemos.....	9
4.2 RDBVS duomenų analizė.....	11
4.3 Duomenų saugyklos.....	11
4.4 OLAP interaktyvūs analizės įrankiai.....	13
4.5 OLAP architektūra.....	14
4.6 ROLAP pranašumai.....	15
4.7 ROLAP trūkumai.....	15
4.8 OLAP operatoriai SQL kalboje.....	15
5)Aplikacijų dalies technologijos.....	17
5.1 Java platforma.....	17
5.2 ASP .NET technologija.....	18
5.3 PHP interpretatorius.....	18
III.Projektinė dalis. Tikslinė klientų aptarnavimo sistema (TKAS).	20
1)Įvadas.....	20
2)Reikalavimų specifikacija.....	20
2.1 Pagrindiniai TKAS tikslai:.....	20
2.2 TKAS funkciniai reikalavimai.....	20
2.3 TKAS reikalavimai.....	21
3)TKAS projektinė dalis.....	21
3.1 Tikslas.....	21
3.2 Projekto vykdymo planas.....	22
3.3 TKAS pradinis projekto aprašymas.....	22
3.3.1 Aukščiausio lygio duomenų srautų diagrama.....	22
3.3.2 Nulinio lygmens duomenų srautų diagrama.....	23
3.3.3 Sistemos dinaminis vaizdas ir panaudojimo atvejai.....	23
3.3.4 Panaudojimo atvejų sąrašas ir aprašai.....	24
3.3.5 Duomenų bazės duomenų struktūra ir ryšiai.....	26
3.3.6 Priemonių ir technologijų pasirinkimas.....	26
3.3.7 Architektūros pateikimas.....	27
3.3.8 Apibendrinimas.....	27
IV.Darbų eigos aprašymas.....	28
1)Darbų eigos grafas.....	28
2)Testavimas.....	28
Testas 1: prisijungimas.....	29
Testas 2: paieška ir rezultatų pateikimas.....	29
Testas 3: duomenų įvedimas.....	29
Testas 4: komentavimas.....	30
Testas 5: paskyrų suteikimas.....	30
3)Galutinio projekto būvio aprašymas.....	30

4)Darbo rezultatu analizē	30
V.Išvados.....	32
VI.Literatūra.....	33
VII.Anotacija.....	34

I. Įvadas

Sparčiai plečiantis paslaugų sferai, didėjant klientų srautams, vis dažniau tenka susidurti su klientų problemų registravimo ir aptarnavimo uždaviniais. Tuo pat metu stipriai į įmonių veiklą ir žmonių gyvenimą skverbiasi informacinės technologijos. Įvertinus šias tendencijas tenka pasirinkti įrankius ir technologijas klientų srautams aptarnauti, kad nereikėtų vesti popierinių žurnalų, darbas vyktų sparčiau, intelektualiau.

Šio darbo tikslas yra išanalizuoti ryšiams su klientais skirtų valdyti sistemų technologijas, funkcijas, įvertinti klientams aptarnauti skirtų įrankių galimybes. Pagrindinis dėmesys skiriamas būtent klientų aptarnavimo sričiai.

Komercinių ir nekomercinių ryšiams su klientais valdyti skirtų sistemų lyginimu, turimų klientų aptarnavimo įrankių ir universalumo aspektu siekiama atskleisti darbo srities naujumą.

Tyrimu siekiama:

- Supažindinti su CRM sistemų raida ir būtinybe.
- Išanalizuoti populiariausių komercinių ir nekomercinių CRM sistemų funkcines galimybes.
- Išsiaiškinti, kokios technologijos naudojamos CRM sistemose.
- Susipažinti su analizės įrankiais ir žinių gavimo metodais.
- Įvertinti tikslinių priemonių, skirtų klientams aptarnauti funkcijas.
- Įvertinti TKAS (tikslinė klientų aptarnavimo sistema) būtinumą.
- Suprojektuoti, suprogramuoti ir ištestuoti TKAS

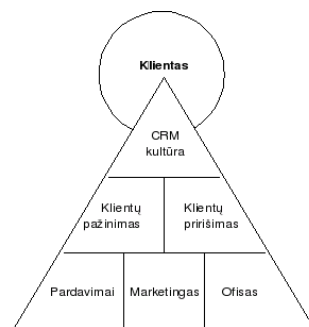
Projekto praktinė vertė yra sistemos universalumas, lankstumas. Galimybė ją taikyti bet kurią paslaugos šaką teikiančiai įmonei. Projektas gali būti modulis CRM sistemoje.

II. Teorinė dalis

1) CRM sistemų apibrėžimas, raida

Terminas „Ryšius su klientais valdymas“ (CRM) vis dažniau vartojamas šiandieniniame versle. Daugelis įmonių, kurių verslo centras yra klientas, investuoja į šią sritį, tikėdamiesi pažinti, įgyti pasitikėjimą, stiprius ryšius, kliento lojalumą. Apibrėžti šią sistemą galima taip:

CRM yra strategija ilgalaikiams ryšiams su klientais užmegzti ir palaikyti.



Pav. 1: Klientas įmonės strategijoje

Iliustracija parodo, kaip įvairūs organizacijos padaliniai siejasi tarpusavyje, keisdami informaciją apie klientų poreikius.

Elektroninių CRM sistemų atsiradimas buvo pirmas stiprus žingsnis link efektyvesnio strategiškai svarbių klientų aptarnavimo. Nors ankstyvieji sprendimai turėjo savitarnos galimybę, skambučių centrų funkcijas, tačiau šios sistemos netiko organizacijų strategijos vykdymui. Kiekviena organizacija yra unikali ir negali dirbti pagal pateiktą šabloną. Atsiranda analitinių įrankių poreikis.

Klientų aptarnavimo problema labiausiai aktuali įmonėse, kurios dirba aptarnavimo srityse, pvz:

- Telekomunikacijų kompanijos;
- Interneto tiekėjai;
- Mažmeninės finansų kompanijos;
- Komunalinių paslaugų tiekėjai;
- Didieji prekybos centrai;
- Kitos sritys, tiesiogiai bendraujančios su klientais.

CRM sistema nėra vien formali programa – aplikacija, tai įvairių technologijų junginys, informacijos resursai ir procesai, kurie sudaro galimybę kiek įmanoma labiau pažinti klientą. CRM apima daugiau nei tradicinius pardavimus, marketingą, tiekimą, biuro darbą ar aptarnavimo kokybę.

2) Sistemų taikymas

CRM sistemos apima šias sritis:

Rinkodaros automatizavimas (angl. *Marketing automation*) – programų, aplikacijų rinkinys, skirtas rinkodaros procesams automatizuoti:

- Kurti ir vykdyti marketingo kampanijas, akcijas;
- Scenarijų kūrimas ir išlaikymas;
- Literatūros saugojimas elektroniniame formate;
- Konkursų informacijos saugojimas;
- Pasiekimų ir praradimų informacijos saugojimas, kurią vėliau galima analizuoti;
- Importavimas ir eksportavimas įvairių duomenų; pvz: kontaktų;
- Galimybė kurti daugelį laiškų, panaudoti HTML technologiją;
- Galimybė įrašyti keletą kontaktų vartotojo kortelėje;
- Pašto kodo analizavimas – kliento vietai nustatyti;
- Automatinis skambinimas naudojantis CRM duomenų baze.

Įvykių valdymas (angl. *case management*) – galimybė stebėti veiksmus, susijusius su pasirinktais klientais.

Skambučių centras (angl. *call center*) – centralizuota ofiso dalis, skirta priimti ir apdoroti dideliems telefoninių skambučių srautams.

Servisas ir aptarnavimas (angl. *service and support*) – incidentų, paslaugų fiksavimas, bilieto priskyrimas, tolimesnių būsenų kontroliavimas, tranzakcijų saugojimas. Paslaugų apskaitos funkcijos.

Rezultatų valdymas (angl. *funnel management*) – sistema leidžianti valdyti ir stebėti įvairius verslo rodiklius, rezultatus. Pvz: Biudžeto augimas ar mažėjimas).

Pardavimų prognozės (angl. *sales forecasting*) – ataskaitos, grafikai, suteikiantys galimybę prognozuoti pardavimus.

Pardavimų automatizavimas (angl. *sales automation*) – naudojant informaciją apie klientus, statistinius duomenis, šablonus, prezentacijas ir kitus įrankius, suteikiama galimybė automatizuoti pardavimų procesus.

Ryšių su partneriais valdymas (angl. *partner relationship management*) – partnerių kontaktinių duomenų, verslo srities, gaminamos produkcijos ar informacijos apie paslaugas valdymas.

Vartotojų ir klientų valdymas (angl. *account/customer management*) – informacijos apie klientus valdymas – kontaktai, adresai, veiklos sritys, kita informacija.

Verslo analizės ir duomenų gavybos įrankiai (angl. *business intelligence*) – skirti gauti informaciją įmonės operaciniams, taktiniams veiksams, strateginiams sprendimams.

3) Populiariausių CRM sistemų apžvalga

Remiantis nepriklausomais tyrimų duomenimis¹, buvo išrinkta keletas populiariausių ryšiams su klientais valdyti skirtų sistemų. Pagrindinių funkcijų palyginimas pateiktas 1 lentelėje:

	<i>Act!</i>	<i>Goldmine</i>	<i>SalesLogix</i>	<i>Siebel CRM</i>	<i>Microsoft CRM</i>	<i>Sage ACCPAC</i>	<i>Salesforce.com</i>	<i>OpenCRX</i>
Rinkodara								
● Kurti ir vykdyti marketingo kampanijas, akcijas	(•)		•	•		•	•	•
● Scenarijų kūrimas ir išlaikymas				•		•	•	
● Literatūros saugojimas elektroniniame formate	•		•	•		•	•	•
● Konkursų informacijos saugojimas			•	•		•	•	•
● Pasiekimų ir praradimų informacijos saugojimas	•		•	•		•	•	•
● Importavimas ir eksportavimas įvairių duomenų	•		•	•		•	•	
● Pašto įtraukimas	•		•	•		•	•	•
● Elektroninio pašto įtraukimas	•		•	•		•	•	•
● Daugelio adresų priskyrimas			•	•		•	•	•
● Pašto kodo analizavimas	(•)		(•)	(•)			•	
● Automatinis skambinimas	•		•	•		•		
Pardavimų automatizavimas								
● Sandėrių valdymas			•	•		•	•	•
● Prognozės	•		•	•		•	•	•
● Planų stebėjimas	•		•	•		•	•	•

1 Šaltinis: „CRM Software Selection Kit 2006 08“

	<i>Act!</i>	<i>Goldmine</i>	<i>SalesLogix</i>	<i>Siebel CRM</i>	<i>Microsoft CRM</i>	<i>Sage ACCPAC</i>	<i>Salesforce.com</i>	<i>OpenCRX</i>
● Teritorijos valdymas			•	•		•	•	
● Rinkos prisotinimo valdymas			•	•		•	•	
● Kursų / pasiūlymų valdymas	(•)		(•)	•		•	•	
● Specifinių pardavimų tipų palaikymas	(•)		•	•		(•)	•	•
● Grafikų sudarymas	(•)		(•)	•		(•)	•	•
Servisas ir aptarnavimas								
● Incidentų fiksavimas			•	•		•	•	•
● Automatinis priskyrimas ir perspėjimas apie incidentus			•	•		•	•	•
● Automatinis naujų taisyklių skelbimas			•	•		•	•	
● Istorijos kaupimas			•	•			•	
● Incidentų ir paslaugų apskaita				•		•	•	
Naudojamos duomenų bazės								
● Microsoft SQL Server	•	•	•	•	•	•		•
● Oracle 8i, 9i, 10g		•	•	•		•		•
● DB2				•		•		•
● MS ACCESS 2000/XP/2003						•		
● Kitos DBVS (MySQL, PostgreSQL)								•
Serverio operacinė sistema								
● Windows 2000/2003 server	•	•	•	•	•	•		•
● Linux				•				•
● Kitos				•				•
Klientinės dalies technologijos								
● HTTP sąsaja	•	•	•	•	•	•	•	•
● Diegiama aplikacija	•	•	•		•			

	<i>Act!</i>	<i>Goldmine</i>	<i>SalesLogix</i>	<i>Siebel CRM</i>	<i>Microsoft CRM</i>	<i>Sage ACCPAC</i>	<i>Salesforce.com</i>	<i>OpenCRX</i>
Klientinės dalies operacinė sistema								
● Windows XP/2000/ME/98	•	•	•	•	•	•	•	•
● Linux							•	•
● Kitos							•	•

Lentelė 1: Klientų aptarnavimo sistemų funkcionalumo palyginimas

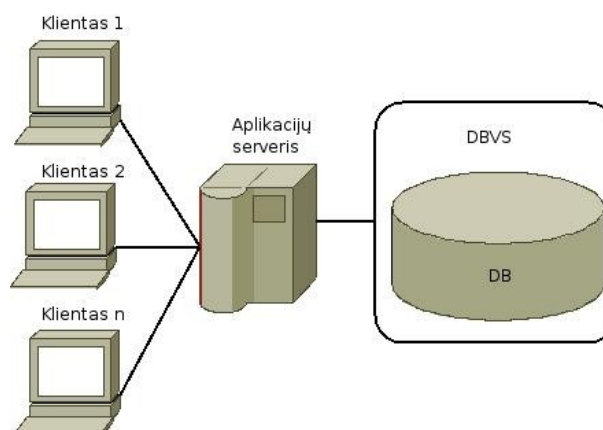
4) CRM sistemose naudojamų technologijų ir realizavimo metodų analizė

4.1 Duomenų bazių valdymo sistemos

Duomenų baze (DB) vadiname tarpusavyje susijusių duomenų rinkinį, kuriems apdoroti pasitelkiamos užklauskos, funkcijos. Tipinė duomenų bazės struktūra yra dvimatė lentelė, susidedanti iš stulpelių ir eilučių.

Duomenų bazių valdymo sistema (DBVS) atlieka vartotojo sąsajos su duomenų baze vaidmenį.

Tipinė duomenų bazės, valdymo sistemos ir aplikacijų struktūra:



Pav. 2: Klientas – DB, tipinė schema

Aplikacijų serveris (AS), duomenų bazė ir valdymo sistema gali būti tame pačiame lokaliame kompiuteryje. Taip pat AS vietoje gali būti kita sistema, tarpininkaujanti tarp kliento ir DBVS, pvz.:

HTTP serveris ir interpretatorius.

DBVS funkcijos:

Duomenų pertekliaus minimizavimas. DBVS privalo minimizuoti duomenų perteklių, kiek įmanoma labiau išvengti duomenų dubliavimosi.

Efektyvumas. DB yra skirta plačiam vartotojų ratui, todėl DBVS turi užtikrinti efektyvų bendrą DB vartojimą.

Vientisumas. DB yra vadinama vientisa (integrali), jei ji tenkina tam tikrus apribojimus (sąlygas) duomenims ir išsaugo tuos apribojimus modifikuojant (keičiant, šalinant, įterpiant duomenis).

DB neprieštarīgumas. DB vadinama neprieštarīginga, jei bet kuriuo laiko momentu, į visų vartotojų užklausas reaguojama vienodai.

Duomenų saugumas. Duomenų apsauga nuo tyčinio ar netyčinio duomenų sunaikinimo, pakeitimo ar peržiūros.

Efektyvus užklausų vykdymas. DBVS keliamas uždavinys – dideliuose duomenų masyvuose efektyviai atlikti ne tik planuotas užklausas (pagal iš anksto numatytus kriterijus), bet ir neplanuotas.

Duomenų bazių vartotojai manipuliuodami duomenimis atlieka šiuos pagrindinius veiksmus:

- naujo įrašo įtraukimas;
- esamo įrašo keitimas;
- esamo įrašo šalinimas;
- duomenų išrinkimas.

Dauguma DBVS naudoja reliacinį modelį (sąvokų rinkinį ir jų vartojimo taisykles); angl. *relation* (santykis) – matematinis terminas lentelės atitikmuo. Reliacinė DB – tai DB, kurią vartotojai priima kaip kelių lentelių rinkinį.

Pagrindinės reliacinio modelio sąvokos yra šios: lentelė, eilutė, stulpelis.

Pagrindinis reliacinės sistemos principas – vartotojui pateikti užklausų rezultatus lentelių pavidalu ir ne kitaip. Taip pat vartotojui pateikiami operatoriai, leidžiantys iš esamų lentelių gauti naujas.

Nereliacinės sistemos vartotojai DB duomenis mato kitokios struktūros nei reliacinės sistemos lentelėse. **Hierarchinėse** sistemose duomenys vaizduojami medžio struktūromis, o tarp operacijų yra

judėjimo hierarchinėmis struktūromis į viršų ir žemyn operacijos. **Tinklinės** sistemos yra hierarchinių sistemų apibendrinimas, duomenys vaizduojami orientuotais grafais. Tinklinės sistemos iki šiol taikomos geografinėse IS.

Duomenimis operuojama naudojantis formalia kalba, pvz.: SQL (struktūrizuota užklausa kalba). SQL - populiariausia iš šiuo metu naudojamų kalbų, skirtų aprašyti duomenis ir manipuluoti jais reliacinių duomenų bazių valdymo sistemose (sukurta XX a. 8-ame dešimtmetyje). Pirmoji šią kalbą komerciniame produkte pritaikė korporacija, dabar žinoma Oracle vardu.

SQL yra neprocedūrinė (deklaratyvi) programavimo kalba. Kalboje išskiriamos trys sakinių grupės: duomenų apibrėžimo, manipuliavimo ir valdymo sakiniai.

4.2 RDBVS duomenų analizė

Duomenų analizę galima atlikti RDBVS, kurios kartu naudojamos ir tranzakcijoms realiuoju laiku vykdyti (*online transaction processing*, OLTP). OLTP duomenų bazės pritaikytos tam, kad galėtų aptarnauti daug vienu metu į jas besikreipiančių vartotojų, kurie intensyviai įterpia bei keičia duomenis. Paprastai, tokios duomenų bazės turi sudėtingą struktūrą (didelė normalizacija), o duomenys jose parodo esamą įmonės situaciją, tačiau ne istoriją. Jose esančių duomenų analizė dažniausiai remiasi įprastų užklausa, parašytų SQL kalba, vykdymu.

Tačiau, kai duomenų susikaupia pernelyg daug ir organizacija mėgina atlikti strateginę analizę naudodama tą pačią OLTP duomenų bazę, kaip ir einamojo duomenų apdorojimo atveju, dažnai susiduriama su keliomis problemomis:

- didelių skaičiavimo pajėgumų reikalaujančios intensyviai vykdomos analitinės užklauskos neigiamai veikia darbinės sistemos produktyvumą;
- Jeigu darbinėje duomenų bazėje saugoma istorinė informacija, nereikalinga kasdienėms operacijoms atlikti, tokios sistemos produktyvumas mažėja, nes duomenų laikui bėgant vis daugėja.²

Dėl minėtų problemų atsiranda poreikis kurti duomenų saugyklas ir įrankius žinioms gauti.

4.3 Duomenų saugyklos

Duomenų saugyklos (DS) – periodiškai papildoma, istorinius duomenis kaupianti duomenų bazė, skirta analitiniams procesams, užklauskoms atlikti. Paprastai duomenų saugykla (sandėlys) būna

² NK „Interaktyvios duomenų analizės įrankiai šiuolaikinėje įmonėje. OLAP duomenų bazės“ Ernestas Sysojevas 2003-06-25

atskirtas nuo darbinės RDBVS. DS vienas iš svarbiausių tikslų kiek įmanoma greičiau pateikti rezultata, suformuotą iš istorinių duomenų, neapkraunant darbinės duomenų bazės. DS pasižymi šiomis savybėmis:

- orientuoti temomis – galimybė sisteminti pagal temas, sritis, pvz.: studentai, klientai, produktai;
- integravimas – duomenų struktūrų sudarymas iš įvairių šaltinių;
- laikotarpių fiksavimas – galimybė atlikti nuokrypių, tendencijų ir pasikeitimų analizę pagal pasirinktą laikotarpį;
- duomenų skaitymas – duomenys nėra keičiami, tik skaitomi ir periodiškai papildomi naujais.

Sudėtinės DS dalys vadinamos duomenų centrais (angl. *Data mart* – duomenų vitrina). Duomenų sandėlyje duomenų centras paprastai realizuojamas kaip viena faktų lentelė su keliomis ją lydinčiomis dimensijų lentelėmis (daugiamatis kubas su dimensijų aprašymais)³. Duomenų centrai paprastai yra ribotos erdvės.

Duomenų sandėlių privalumai:

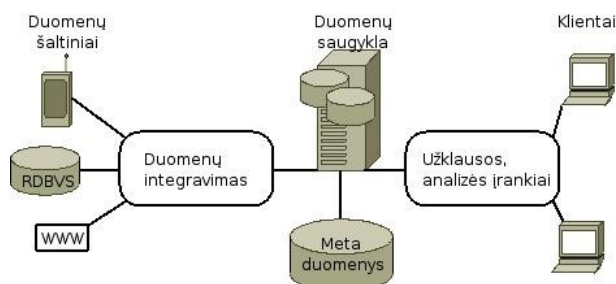
- greitas užklausų (angl. *query*) vykdymas;
- užklausos nematomos už sandėlio ribų;
- lokalus duomenų apdorojimas nedaro įtakos saugyklai;
- galima operuoti duomenimis, kai šaltiniai nepasiekiami;
- įmanoma operuoti duomenimis, nesaugomais DBVS;
- papildomos informacijos saugojimas.

DS modeliai:

- reliacinis;
- žvaigždės arba snaigės;
- daugiamaciai kubai.

Duomenų saugyklos architektūra:

3 „Informacinės technologijos“ 2002 pavasaris (13). 13 puslapis.



Pav. 3: Tipinė duomenų sandėlio architektūra

Duomenų šaltiniai – bet kokia informacijos kaupimo, saugojimo terpė. Gali būti reliacinė DBVS, tekstiniame dokumente saugoma informacija (CSV tipas), XML tipo duomenys, www informacija ir t.t.

Duomenų integravimas – periodišką veiksmą, kurio metu duomenys paimami iš šaltinių, sisteminami, normalizuojami, taisomos klaidos, skirstomi periodais, keliama į duomenų saugyklą.

Meta duomenys – lygiagreti informacinė sistema, kuri saugo duomenis apie duomenų saugyklą. Čia turėtų būti: DS objektų (lentelių, laukų ir pan.) adresai ir jų aprašai; kiekvieno duomenų kilmė, kuri ypač svarbi, jei duomenys į DS patenka iš skirtingų DB ir turi jose skirtingas žymes; sistemos vartotojai ir jų atributai; transformacijos ir verifikacijos taisyklės; DS istorija su pakeitimų datomis; agregatų sudarymo taisyklės ir kt.⁴

Užklauso, analizės įrankiai – iš anksto suformuotos užklauso statistiniams duomenims pateikti, OLAP (OnLine Analytical Processing) programinė įranga. OLAP – tai programiniai produktai, kurie leidžia visapusiškai analizuoti verslo informaciją realiuoju laiku. Iš DS žinios gali būti gaunamos ir įprastais OLTP įrankiais, tačiau jie nėra patogūs, reikalaujama specialių žinių norint suformuoti tinkams užklauso.

4.4 OLAP interaktyvūs analizės įrankiai

Darbinėse reliacinėse duomenų bazėse ar duomenų saugyklose saugomi pradiniai duomenys transformuojami ir sukuriama optimizuota duomenų saugojimo struktūra - OLAP duomenų kubai (angl. *data cubes*), kurie specialiai pritaikyti greitai duomenų analizei.

OLAP kubuose didelė tarpinių skaičiavimų dalis atliekama dar iki duomenų kubo naudojimo, o vieną kartą atliktų tarpinių skaičiavimų rezultatais gali pasinaudoti visi prieigos teises turintys šios duomenų bazės vartotojai.

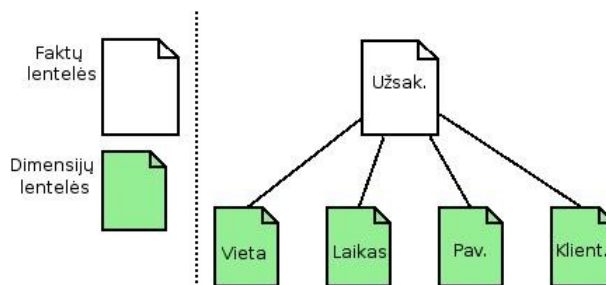
Aprašant OLAP duomenų struktūras (kubus) naudojama pjūvių sąvoka (dimensions). OLAP

4 „Informacijos mokslai“ 2002 Nr.22. Laima Zalieckaitė, Zenonas Brazaitis „Sprendimų paramos sistemų plėtra“.

duomenų bazėje esantys įvykiai (eilutė lentelėje) gali būti analizuojami pagal įvairius parametrus - įvairiais pjūviais (dimensijomis).

OLAP DB gali būti kelių struktūrų:

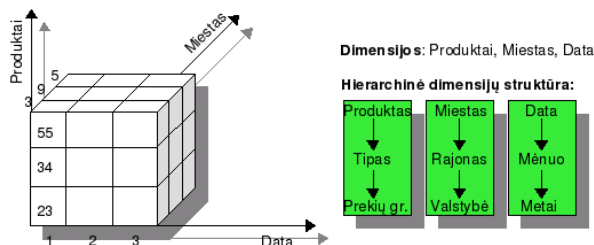
Žvaigždės. Pagrindinės sąvokos yra faktų ir dimensijų lentelės. Faktų lentelėje saugomos skaitinės reikšmės ir sąryšiai su dimensijų lentelėmis. Kiekviena dimensijų lentelė sukuriami atskiroms reikšmėms; pvz.: (vieta, klientas, laikas).



Pav. 4: Žvaigždės struktūros OLAP bazės pavyzdys

Snaigės. Kai žvaigždės struktūra pernelyg išsiplečia, tenka skaidyti smulkesniais segmentais.

Duomenų kubai. Populiariausias duomenų, esančių DS, vaizdavimo būdas. Daugiamačiai kubai yra užpildyti skaitinėmis reikšmėmis, kurios ir yra analizės objektai. Kiekviena reikšmė priklauso dimensijų lentelėms.



Pav. 5: Daugiamačio kubo pavyzdys

4.5 OLAP architektūra

ROLAP veikia reliacinės duomenų bazės pagrindu kur sukuriamos užklauskos, formuojančios duomenų kubus. Kiekvieną kartą analizuojant kubo duomenis, kubo reikšmės perskaičiuojamos. ROLAP užklauskos nėra tokios greitos kaip MOLAP, ypač dirbant su dideliais duomenų kubais, tačiau čia saugoma detali informacija ir galima pasinaudoti „Drill - In“ galimybėmis. ROLAP duomenų kubą paprasčiau papildyti naujomis dimensijomis.

MOLAP (angl. *multidimensional OLAP*) - tai duomenų analizės priemonė, paremta duomenų

kubo modeliui. MOLAP saugo jau susistemintus kubo duomenis, darant užklausą nereikia kiekvieną kartą skaičiuoti kubo reikšmių, todėl MOLAP priemonės pasižymi greitu atsakymu į užklausą. Tačiau į MOLAP kubą sudėtingiau pridėti naujas dimensijas, be to, kube nesaugomi detalizuoti duomenys, todėl negalima pasinaudoti „Drill - In“ galimybėmis.

HOLAP (angl. *hybrid OLAP*) - tai ROLAP ir MOLAP priemonės, apjungtos į vieną. HOLAP gali naudoti jau susistemintas reikšmes bei reliacinės bazės duomenų šaltinius.

4.6 ROLAP pranašumai

- Duomenys saugomi standartinėje reliacinėje duomenų bazėje todėl galima panaudoti SQL ataskaitų (angl. *reporting*) įrankius, (įrankiai nebūtinai turi būti OLAP įrankiai);
- ROLAP priemonės leidžia lanksčiau valdyti didelius duomenų kiekius (milijonai įrašų);
- ROLAP įrankiai geriau apdoroja neskaičiuojamas reikšmes (angl. *non-aggregatable facts*), pvz.: tekstinius duomenis, MOLAP priemonės lėtai vykdo užklausas su tokiais duomenimis;
- ROLAP priemonės leidžia sudaryti duomenų modelius, kurie „netilptų“ į griežtą dimensijų modelį.

4.7 ROLAP trūkumai

- ROLAP lėčiau veikia esant dideliame duomenų kiekiui;
- pakelti ROLAP greičiui reikia kurti skaičiuojamųjų reikšmių lenteles, tai užima daugiau laiko.
- dauguma ROLAP priemonių praleidžia skaičiuojamųjų reikšmių lentelių kūrimą, tada nukenčia užklaustos vykdymo greitis, nes reikia apdoroti didesnius duomenų kiekius;
- ROLAP naudoja bendros paskirties duomenų bazes, kurios neturi specialų MOLAP naudojamų priemonių (pvz.: hierarchinis indeksavimas); tačiau modernios ROLAP priemonės jau turi SQL kalboje numatytus specialius operatorius (CUBE, ROLLUP);
- ROLAP naudoja SQL užklausas visiems skaičiavimams, todėl ROLAP įrankiai netinka sudėtingiems modeliams, kuriuos aprašyti SQL kalba labai sudėtinga.

4.8 OLAP operatoriai SQL kalboje

SQL kalboje naudojama GROUP BY konstrukcija (naudojant grupines funkcijas SUM, AVG, COUNT ir kt.) leidžia agreguoti tam tikro stulpelio reikšmes pagal nurodytą stulpelį arba stulpelių sąrašą. Jei reikia paskaičiuoti papildomas reikšmes, tenka naudoti UNION operatorių, užklausa tampa sudėtinga, susidedanti iš daugelio elementų. Papildomi GROUP BY konstrukcijos operatoriai ROLLUP ir CUBE leidžia išspręsti šias problemas vienoje užklausoje.

ROLLUP funkcija leidžia pamatyti visas sumas pagal visas stulpelių, įeinančių į GROUP BY ROLLUP sąrašą, kombinacijas (tai yra tas pats, kaip ir paprastas GROUP BY) ir, papildomai, tarpines sumas pagal visas galimas pirmųjų stulpelių reikšmes.

ROLLUP sakinio pavyzdys:

```
SELECT parduot, preke, sum(pard) „Pard“
      from pardavimai
GROUP BY ROLLUP (tipas,preke);
```

Rezultatas:

parduot	preke	pard
1	7620	1300
1	7617	2450
1	7614	5000
1		8750
2	7601	6000
2	7620	1900
2	7603	2975
2		10875
		19625

Skirtingai nuo funkcijos ROLLUP, funkcija CUBE papildomai suformuoja sumas pagal visas galimas grupavimo sąraše esančių stulpelių (o ne tik pirmųjų, kaip ROLLUP atveju) kombinacijas.

CUBE sakinio pavyzdys:

```
SELECT parduot, preke, sum(pard) „Pard“
      from pardavimai
GROUP BY CUBE (tipas,preke);
```

Rezultatas:

parduot	preke	pard
1	7620	1300
1	7617	2450
1	7614	5000
1		8750
2	7601	6000
2	7620	1900
2	7603	2975
2		10875
	7620	3200
	7617	2450
	7614	5000
	7601	6000
	7603	2975
		19625

Apribojimai

CUBE arba ROLLUP atveju grupinės funkcijos viduje negalima naudoti DISTINCT (tai yra, negalima naudoti, pavyzdžiui, *sum(distinct pard)*). Maksimalus GROUP BY sakinyje (nepriklausomai nuo CUBE ar ROLLUP naudojimo) naudojamų stulpelių skaičius yra 255.

5) Aplikacijų dalies technologijos

Visos nagrinėtos ryšiams su klientais valdyti skirtos sistemos yra pritaikytos dirbti tinkle. Vienos dirba per instaliuojamas aplikacijas klientinėje dalyje, kitos naudojasi aplikacijų serveriais. Dažniausiai pasitaikančios ir labiausiai tarpusavyje lyginamos technologijos yra Java, ASP .NET ir PHP.

5.1 Java platforma

Java šiuo metu yra viena populiariausių programavimo kalbų pasaulyje. Ši programavimo kalba sukurta remiantis objektiškai orientuotos kalbos C++ sintakse ir Smalltalk bibliotekomis. Java platforma patogi tuo, kad jos komponentai vykdomi aplikacijų serveryje, o klientinė dalis pasiekama per interneto naršyklę. Tai leidžia vartotojui neprisišti prie operacinės sistemos ar kitos programinės įrangos.

Java programavimo kalba pasižymi šiomis savybėmis: paprasta, objektiškai orientuota, orientuota tinklinių sistemų kūrimui, atspari klaidoms, saugi, nepriklausoma nuo architektūros, nepriklausoma nuo operacinės sistemos, interpretuojama, daugiagijė, dinamiška, atvira.

Tačiau yra ir neigiamų kriterijų: nepakankamas greitis, priklausomybė nuo Java virtualios mašinos (JVM), išeities kodo atstatymas, konstrukcijų trūkumas.

Šiuo metu išskiriamos 4 specializuotos Java technologijų platformos:

- J2SE (Java 2 Platform Standard Edition) – apimantis pagrindinius Java kūrimo įrankius bei svarbiausias Java kalbos bibliotekas, kurios naudojamos ir atskirų programų kūrimui;
- J2EE (Java 2 Platform Enterprise Edition) – skirta informacinėms verslo sistemoms; pagal specifikaciją ši technologija realizuojama J2EE serveriuose, kurie naudojami J2EE programų kūrimui ir diegimui; veikia J2SE pagrindu;
- J2ME (Java 2 Micro Edition) – technologijų rinkinys, skirtas programuoti mini įrenginiams: buitinei technikai, mobiliems telefonams ir kt.;
- Java Card – technologija, kuri skirta adaptuoti Java platformą protingoms kortelėms (angl. *smart card*) ir kitiems panašioms įrenginiams.

Java turi JDBC (Java DataBase connectivity) biblioteką, kuri yra skirta reliacinių duomenų bazių valdymui pagrįstam SQL kalba ir nepriklausomam nuo bazės sistemos tiekėjo (Oracle, SQL Server, DB2 ir kt.). JDBC apibrėžia sąsajas, kurių realizacijos turi būti pateikiamos duomenų bazių sistemos JDBC tvarkyklėse.

5.2 ASP .NET technologija

ASP .NET technologija yra Microsoft kompanijos programinės įrangos produktas. Ši technologija skirta kurti dinamines WEB sistemas smulkioms, vidutinėms ir stambioms internetinėms sistemoms. Šios sistemos pasižymi sparčiu vykdymo laiku, turi daug standartizuotų programavimo įrankių, glaudi sąsaja su dauguma Microsoft produkcijos. Prisirišus prie Windows platformos suteikiamas didelis lankstumas. Dažniausiai programuojama dviem programavimo kalbomis: C# ir Visual Basic .NET.

Nagrinėjama technologija pasižymi šiomis įpatybėmis:

- ASP .NET turi tvarkykles tik MSSQL ir Oracle DBVS. Prie kitų serverių jungiamasi ODBC pagalba;
- veikia tik Windows operacinėje sistemoje;
- galimas darbas tik su IIS ir su komercine Apache 2.0 versija.

5.3 PHP interpretatorius

PHP (PHP hypertext preprocessor) tai skriptų kalba (angl. *scripting language*), kuri pačioje pradžioje buvo orientuota tik į internetą, nors šiuo metu ją galima „drąsiai“ pavadinti programavimo kalba. Be to, su ja galima programuoti ne tik internetinius puslapius. Pavyzdžiui, su „php-gtk“ galima kurti pilnavertiškas cross platformines programas su GUI (angl. *graphical user interface*), konsolines programėles ir t.t. PHP skriptai yra interpretuojami ir įvykdomi serverio pusėje. Pvz.:

```
<html>
  <head>
    <title>Pavyzdys</title>
  </head>
  <body>
    <?php echo "Sveikas aš esu PHP skriptas";?>
  </body>
</html>
```

Po šio skripto įvykdymo turėsime rezultatą (naršyklėje): „Sveikas aš esu PHP skriptas“. Pradžioje pagrindinis PHP privalumas buvo tai, kad PHP skriptas galėjo būti lengvai įterpiamas į HTML kodą kaip CGI. Reikėjo visą HTML išvesti iš CGI skripto. Kuo toliau tuo labiau didesniuose projektuose naudojami šablonai.

PHP skirtumas nuo „JavaScript“ yra tas, jog PHP skriptai yra atliekami serverio pusėje, o vartotojui yra grąžinamas rezultatas (HTML ar XML forma). Rezultatą vartotojas mato savo naršyklėje. O „JavaScript“ yra visiškai perduodamas kliento mašinai (kompiuteriui) ir atliekamas klientinėje dalyje. PHP sintaksės pagrindai yra paimti iš C, Perl, Java, todėl, programavusiems šiomis kalbomis, yra nesudėtinga pradėti programuoti PHP.

PHP interpretavimo kalbos galimybės

Kadangi PHP kodas yra vykdomas serveryje, tai su PHP galima suprogramuoti viską, kas įmanoma su bet kuria CGI programavimo kalba (pvz.: apdoroti HTML formų duomenis, generuoti dinamiškus puslapius, siųsti ir priimti slapukus (angl. *cookies*) ir t.t.).

PHP programavimo kalba yra naudojama trijose srityse:

- serverio pusės programavimas; tai tradicinė PHP taikymo sritis; tam reikia PHP interfeiso (CGI arba serverio modulio), web serverio ir interneto naršyklės;
- komandinės eilutės programavimas;
- grafinės vartotojo sąsajos programų kūrimas.

PHP gali būti naudojama visose pagrindinėse operacinėse sistemose, įskaitant Linux, daugumą Unix sistemos atmainų (UX, Solaris, OpenBSD, FreeBSD ir t.t.), Microsoft Windows, Mac OS X, RISC OS ir daug kitų. Be to, PHP turi palaikymą daugeliui web serverių, tarp jų Apache, Microsoft Internet Information Server, Personal Web Server, Netscape ir iPlanet serveriai, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd ir daugelis kitų. Didžiajai daugumai serverių PHP turi modulius, kitiems serveriams - CGI palaikymą.

Taigi, PHP naudojantys programuotojai turi galimybę pasirinkti operacinę sistemą ir web serverį. Be to, galima rinktis ką naudoti: 1) procedūrinį programavimą, 2) objektinį programavimą, 3) abu programavimo tipus. Nors PHP ir nepalaiko visiškai visų objektinio programavimo standartų, daugelis didesnių projektų yra suprogramuoti naudojant vien tik objektinį PHP programavimą.

PHP niekaip neriboja generuojamo HTML kodo. Su PHP taip pat galima dinamiškai generuoti paveikslukus, PDF failus ir net Flash animaciją.

Vienas iš svarbiausių PHP privalumų yra tai, jog PHP palaiko daug duomenų bazių valdymo sistemų. Programuoti duomenų bazėmis paremtus interneto puslapius yra labai paprasta.

III. Projektinė dalis. Tikslinė klientų aptarnavimo sistema (TKAS).

1) Įvadas

Pasaulyje egzistuojančių ryšių su klientais valdymo sistemų pagrindiniai funkciniai uždaviniai išvardinti tyrimo pradžioje. Nagrinėjamose sistemose trūksta arba visai nėra patogių įrankių serviso ir aptarnavimo funkcijoms atlikti. Tikslinės klientų aptarnavimo sistemos (TKAS) modelis ir architektūra galėtų būti taikomi bet kuriai ryšiams su klientais valdymo sistemai.

Kuriamas produktas tai – grupinė darbo priemonė, skirta įmonėms, fiksuojančioms klientų nusiskundimus, darbų broką, gedimus ar kitas užduotis. Sistema yra pritaikyta įmonėms, teikiančioms įvairias paslaugas ar keletą skirtingų paslaugų. Produkto lankstumas ir informacijos sisteminimas suteikia galimybę, greitai ir efektyviai spręsti iškilusias problemas naudojant sukauptą informaciją.

2) Reikalavimų specifikacija

2.1 Pagrindiniai TKAS tikslai:

- sistemos universalumas;
- paslaugų šakų pildymas;
- vartotojų teisių skirstymas pagal veiklos sritį, regionus;
- greitas ir tikslus nusiskundimų ar kitų užduočių registravimas;
- grupinis darbas;
- paskyrų sistema;
- darbų eigos stebėjimas;
- darbuotojų našumo įvertinimas;
- analizės įrankiai.

2.2 TKAS funkciniai reikalavimai

Vartotojų autorizacija – sistema leidžia prisijungti tik registruotiems vartotojams. Vartotojai sukuriama sisteminiai, su teisių grupėmis.

Užduočių registracija – sistema turi patogų langą užduotims registruoti, nurodant kontaktinius duomenis, paslaugos tipą, prioritetą, komentarą.

Paieškos organizavimas – sistema suprojektuota taip, kad būtų galima atlikti paiešką pagal raktinį žodį arba pagal paslaugų tipus ir datą.

Paieškos rezultatų pateikimas – rezultatai pateikiami lentelės forma. Tokia pati lentelė naudojama pagrindiniam darbui su baze organizuoti.

Užduočių vykdymas – kiekvienas darbuotojas privalo užduočiai (darbui) suteikti statusą, komentarą, gedimo tipą, jei reikia ir paskyrą.

Statistikos pateikimas – galimybė organizuoti užklausas statistiniams duomenims gauti, pasirenkant reikiamus kriterijus.

Istorijos kaupimas – saugomas kiekvienas įrašas duomenų bazėje, fiksuojami visi darbuotojų žingsniai. Tai suteikia galimybę atlikti statistinius skaičiavimus.

2.3 TKAS reikalavimai

Sistemos modelis buvo projektuojamas, realizuojamas ir testuojamas pasirenkant žemiau išvardintus reikalavimus.

Vartotojo sąsajai – vartotojo prieiga organizuojama per interneto naršyklę. Siekiant lankstumo, nedarant pakeitimų vartotojo sistemoje (nediegiant papildomų aplikacijų esant TKAS galimybių plėtimui).

Nesėkmės atvejui – testavimo metu išspręsti pastebėtas spragas, reguliariai atlikti sistemos kopijas (angl. *dump*).

Duomenų įvedimo kontrolė – kontrolė duomenų įvedimo etapuose: apsauga nuo klaidingo kodo įterpimo, kintamojo tipo nustatymas, duomenų ilgio fiksavimas, duomenų dubliavimosi prevencija.

Duomenų išvedimo formatai – duomenys pateikiami interneto naršyklės lange.

Sistemos perkeliamumas – galimybė sistemą perkelti neprisirišant prie platformos (OS), neprarandant funkcionalumo ir duomenų.

Mobilių įrenginių prieiga – galimybė prie sistemos jungtis naudojant mobiliuosius įrenginius.

3) TKAS projektinė dalis

3.1 Tikslas

Šios dalies paskirtis – išnagrinėti kuriamos sistemos projektinius aspektus, remiantis TKAS funkciniais reikalavimais: veiklos, duomenų srautų, architektūros diagramas; pateikti loginį vaizdą, duomenų struktūras. Išsamiai atlikus projektinę dalį sėkmingai įgyvendinti projektą, programuoti remiantis rezultatais.

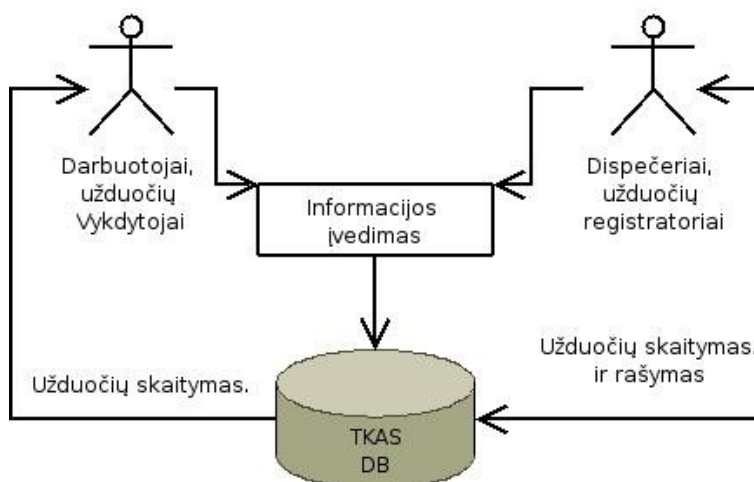
3.2 Projekto vykdymo planas

Projekto realizavimas bus atliekamas remiantis nustatytu vykdymo planu:

1. TKAS duomenų srautų diagramų sudarymas.
2. Sistemos dinaminio vaizdo pateikimas.
3. Panaudojimo atvejų pateikimas ir aprašymas remiantis 2 punktu.
4. Duomenų struktūrų ir ryšių pateikimas.
5. Priemonių pasirinkimas.
6. Architektūros pateikimas.
7. Sistemos programavimas.
8. Testavimas.
9. Galutinio projekto būvio aprašymas.
10. Darbo rezultatų analizė.

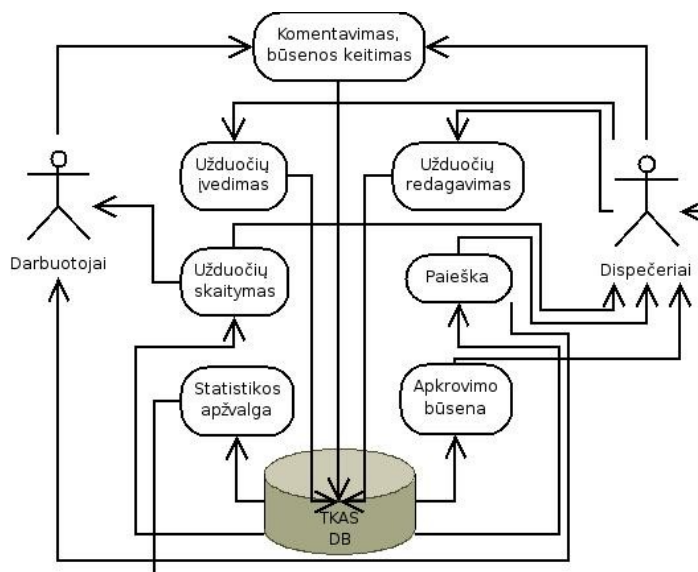
3.3 TKAS pradinis projekto aprašymas

3.3.1 Aukščiausio lygio duomenų srautų diagrama



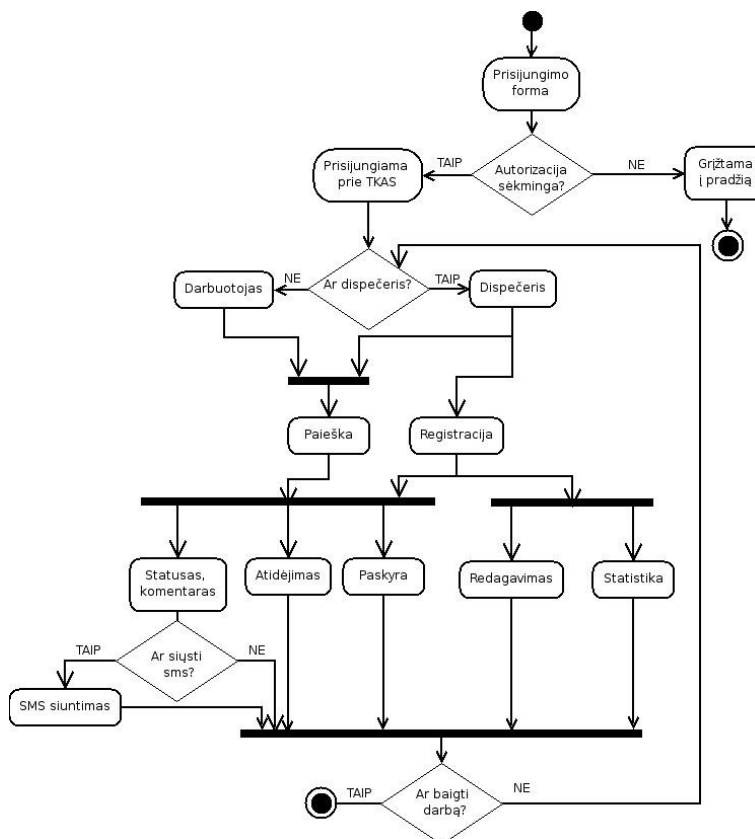
Pav. 6: Aukščiausio lygio TKAS duomenų srautų diagrama

3.3.2 Nulinio lygmens duomenų srautų diagrama



Pav. 7: Nulinio lygmens duomenų srautų diagrama

3.3.3 Sistemos dinaminis vaizdas ir panaudojimo atvejai



Pav. 8: Sistemos dinaminis vaizdas ir panaudojimo atvejai

3.3.4 Panaudojimo atvejų sąrašas ir aprašai

1. **Panaudojimo atvejis:** prisijungimas

Aprašas: skirtas vartotojų autorizacijai, ir teisių nustatymams.

Vartotojas / aktorius: dispečeris ir paprasti vartotojai.

Prieš sąlyga: autorizacijos formos aktyvavimas.

Sužadinimo sąlyga: prisijungimo mygtuko paspaudimas.

Po-sąlyga: patekimas į TKAS.

2. **Panaudojimo atvejis:** paieška

Aprašas: užduočių sąrašo atvėrimas.

Vartotojas / aktorius: dispečeris ir paprasti vartotojai.

Prieš sąlyga: sėkminga autorizacija.

Sužadinimo sąlyga: paieškos laukelio sužadinimas nurodant paieškos kriterijų arba paslaugų medžio pasirinkto punkto sužadinimas; kalendoriuje nurodytos datos ir paslaugų tipo pasirinkimas.

Po-sąlyga: pateikiamas užduočių sąrašas.

3. **Panaudojimo atvejis:** registracija

Aprašas: naujų užduočių registravimas.

Vartotojas / aktorius: dispečeris.

Prieš sąlyga: sėkminga autorizacija, dispečerio teisių turėjimas.

Sužadinimo sąlyga: užduočių registravimo mygtuko paspaudimas.

Po-sąlyga: užduotis užregistruota duomenų bazėje.

4. **Panaudojimo atvejis:** redagavimas

Aprašas: skirtas registruotos užduoties duomenims koreguoti.

Vartotojas / aktorius: dispečeris.

Prieš sąlyga: sėkmingos paieškos atlikimas.

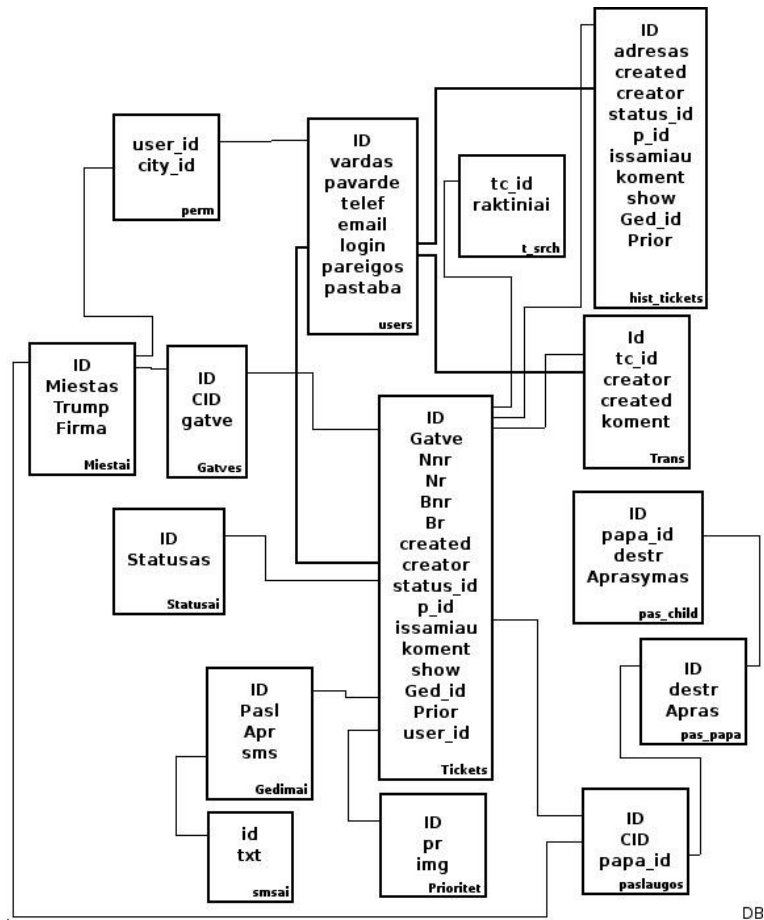
Sužadinimo sąlyga: redagavimo mygtuko paspaudimas prie pasirinktos užduoties.

Po-sąlyga: pasirinktos užduoties duomenys pakeisti.

5. **Panaudojimo atvejis:** užduoties statuso keitimas ir komentaro įvedimas.
- Aprašas:** skirtas užduoties statusui keisti: „atlikta“, „vykdoma“, „naujas“, komentarui palikti.
- Vartotojas / aktorius:** dispečeris ir paprasti vartotojai.
- Prieš sąlyga:** reikiamos užduoties pasirinkimas.
- Sužadinimo sąlyga:** informacijos pildymo mygtuko paspaudimas.
- Po-sąlyga:** pakeistas užduoties statusas, įvestas ar papildytas komentaras.
6. **Panaudojimo atvejis:** paskyros nustatymas.
- Aprašas:** skirtas pasirinktos užduoties atlikimą įpareigoti nurodytam darbuotojui.
- Vartotojas / aktorius:** dispečeris ir paprasti vartotojai.
- Prieš sąlyga:** reikiamos užduoties pasirinkimas.
- Sužadinimo sąlyga:** informacijos pildymo mygtuko paspaudimas.
- Po-sąlyga:** užduotis paskirta atlikti nurodytam darbuotojui.
7. **Panaudojimo atvejis:** statistikos pateikimas.
- Aprašas:** pateikiama statistika apie darbuotojų našumą, atliktų užduočių kiekį.
- Vartotojas / aktorius:** dispečeris.
- Prieš sąlyga:** statistikos mygtuko sužadinimas.
- Sužadinimo sąlyga:** statistikos pateikimo mygtuko sužadinimas, prieš tai nurodžius reikiamus kriterijus.
- Po-sąlyga:** pateikiama statistika įvertinus nurodytus kriterijus.
8. **Panaudojimo atvejis:** trumposios žinutės siuntimas.
- Aprašas:** klientams išsiunčiama trumpoji žinutė iš anksto aprašytais atvejais.
- Vartotojas / aktorius:** dispečeris ir paprasti vartotojai.
- Prieš sąlyga:** iš anksto aprašytos sąlygos tenkinimas.
- Sužadinimo sąlyga:** informacijos pildymo mygtuko sužadinimas pasirinktai užduočiai.

Po-salyga: išsiunčiama trumpoji žinutė.

3.3.5 Duomenų bazės duomenų struktūra ir ryšiai



Pav. 9: TKAS DB duomenų struktūra

3.3.6 Priemonių ir technologijų pasirinkimas

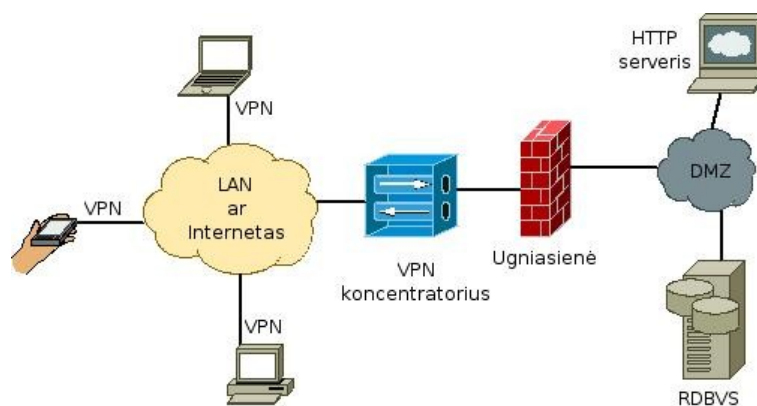
1. Duomenų bazių serverio operacinė sistema: Debian GNU/Linux 4.0
Pagrindiniai kriterijai: saugumas, stabilumas, GNU GPL licenzija.
2. RDBVS: PostgreSQL 8.2
Sparti, funkcionali sistema. PSQL ir PL/PSQL turi panašumų į Oracle duomenų bazių sintaksę, GNU GPL licenzija.
3. WEB serverio operacinė sistema: Debian GNU/Linux 4.0
4. HTTP serveris: Apache 2.0
Lankstus, saugus, turintis daug modulių, populiarus serveris. Pasulyje užima apie 70 % rinkos⁵.

5 Interneto prieiga "http://news.netcraft.com/archives/web_server_survey.html". Paskutinį kartą žiūrėta 2007-04-21

5. Aplikacijų dalies technologijos: PHP interpretatorius.
Paprasta, populiarė, lanksti interpretavimo kalba. Sugeneruotas HTML kodas lengvai atverčiamas interneto naršyklėje.
6. Klientinė dalis: interneto naršyklė.
Suteikia lankstumo, operatyvumo naudojant mobiliuosius įrenginius. Nereikalingos diegiamos aplikacijos klientinėje dalyje.
7. Saugumas: VPN, ugniasienės, HTTP serverio nustatymai, DBVS nustatymai, vartotojų autorizacija duomenų bazėje.
8. Sistemos kūrimo, projektavimo, programavimo priemonės:
 1. „pgAdminIII“ - PostgreSQL DB klientas.
 2. „Quanta Plus“ - internetinių tinklapių kūrimo priemonė. PHP kodo įvedimas.
 3. „GIMP“ – grafiniam apipavidalinimui naudojama programa.
 4. „Dia Diagram Editor“ - diagramų braižymui naudota priemonė.

3.3.7 Architektūros pateikimas

TKAPS gali veikti viename kompiuteryje (serveryje) arba būti paskirstyta specialios paskirties serveriuose. Galima architektūra pateikiama 10 paveiksle.



Pav. 10: Galima TKAS architektūra

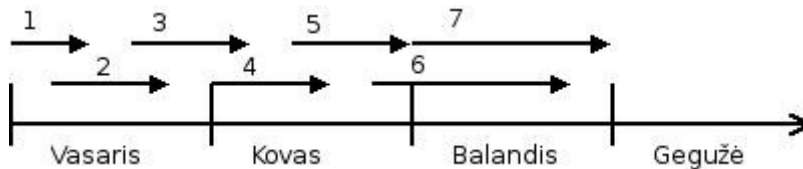
3.3.8 Apibendrinimas

Pagal pateiktą projektą buvo pradėta rašyti programa panaudojus pasirinktas priemones ir technologijas. Sukurta duomenų bazės struktūra, užpildyta duomenų bazė testiniais duomenimis. Sukonfigūruoti HTTP ir DBVS serveriai.

IV. Darbų eigos aprašymas

1) Darbų eigos grafas

Laiko ašyje TKAS analizė, projektavimas ir realizavimas pateiktas darbų eigos grafe (11 pav.):



Pav. 11: Darbų eigos grafas

1. Temos analizė.
2. CRM sistemų apžvalga.
3. CRM sistemose naudojamų technologijų ir realizavimo metodų analizė.
4. TKAS projektavimas.
5. Priemonių ir technologijų pasirinkimas.
6. Programavimas.
7. Testavimas ir problemų sprendimas.

Visi etapai aprašyti teorinėje ir projektinėje dalyse. Jie įvykdyti sėkmingai, laikantis grafiko. Liko atlikti ir aprašyti testavimą ir jo rezultatus.

2) Testavimas

Norint įvertinti sukurtos sistemos veiklos korektiškumą, galimybes ir spartą buvo atliktas visapusiškas testavimas. Testavimai buvo atliekami kiekvienu kūrimo etapu. Galutinį testavimą atliko 9 klientų aptarnavimo srityje dirbantys asmenys.

Testavimo metu buvo stebimi pagrindiniai procesai:

- Autorizacija.
- Duomenų įvedimas.
- Paieška.
- Komentavimas.
- Paskyrų suteikimas.

Testas 1: prisijungimas

Visiems bandantiems prisijungti prie sistemos, pateikiama prisijungimo forma, kurioje prašoma įvesti vartotojo vardą ir slaptažodį.



The image shows a login form with a light green background. It contains the following elements: a label 'Vartotojo vardas:' followed by a text input field containing 'testuotojas1'; a label 'Slaptažodis:' followed by a password input field containing '*****'; and a button labeled 'Prisijungti'.

Pav. 12: Prisijungimo forma

Problema

Kai kurie vartotojai negali prisijungti prie sistemos.

Sprendimas

Ne visose DB esančiose lentelėse suteiktos reikalingos privilegijos. Klaida ištaisyta.

Testas 2: paieška ir rezultatų pateikimas

Vartotojai turėjo atlikti paiešką pagal pasirinktus kriterijus.

Problema

Esant nedideliam duomenų kiekiui (apie 100 įrašų) tiesioginė paieška iš lentelių buvo atliekama sparčiai, nesijautė delsimo. Tačiau, kai buvo sugeneruota apie 40 000 įrašų, darbas nebegalėjo būti tęsiamas, rezultatų pateikimas užtrukdavo apie 10 sekundžių. Taip pat paieškos užklausa (angl. *query*) labai ilga ir sunkiai kaitoma.

Sprendimas

Paieškos pagreitinimui buvo sukurta papildoma lentelė „t_srch“ (žiūrėti pav. 9) su dviem stulpeliais: „tc_id“ – registruotos užduoties numeris ir „raktiniai“ – visų pagrindinių parametru rinkinys (adresas, telefono numeris, laikas, komentarai ir kita informacija), kuris įkeliamas užduoties registravimo, koregavimo ar tranzakcijų atlikimo metu.

Po pataisos paieškos užklausa tapo tokio pavidalo:

```
select * from t_srch s
      inner join tickets t on t.id = s.tc_id
where raktiniai ~* _raktas;
```

Rezultatas

Paieška atliekama sparčiai, užklausa paprastesnė. Nesudėtinga tapo pildyti paieškos lentelę raktiniais žodžiais.

Testas 3: duomenų įvedimas

Testas atliktas sėkmingai, šis etapas buvo išnagrinėtas programavimo metu.

Testas 4: komentavimas

Testo tikslas – įvertinti komentarų rašymo formą, jos patogumą ir tikslumą.

Problema

Sistemos testuotojams nepatogumą kėlė legendos nebuvimas.

Sprendimas

Parašyta keletas funkcijų registruotos užduoties legendai pateikti.

Rezultatas

Legendoje matyti tokia informacija: adresas, telefono numeris, registracijos data, vykdymo data, dispečerio vardas, komentaras, paslaugos tipas, visi darbuotojų komentarai (veiksmai).

Testas 5: paskyrų suteikimas

Testo tikslas įvertinti paskyrų mechanizmą.

Problema

Testuotojams nepatogu buvo paskyras teikti kiekvienai užduočiai atskirai. Buvo pareikalauta paspartinti šį procesą.

Sprendimas

Panaudotos „checkbox“ formos, papildyta sistema funkcijomis, kuriomis tapo įmanoma pažymėti varnelėmis reikiamas arba visas užduotis ir suteikti paskyrą, atspausdinti.

Rezultatas

Paskyrų skirstymas tapo ženkliai efektyvesnis.

3) Galutinio projekto būvio aprašymas

Pagal pateiktus reikalavimus buvo sukurta klientų aptarnavimo sistema, kurią galima pritaikyti bet kuriai veiklos sričiai.

Atlikus pagrindinių funkcijų testavimą ir iškilusių problemų sprendimą, galima teigti, jog projektas yra išbaigtas ir funkcionuojantis. Sistema atlieka visus funkcinis reikalavimus išvardintus projektinėje dalyje. Veikia korektiškai ir sparčiai.

4) Darbo rezultatų analizė

Nagrinętą TKAS projektą galima naudoti kuriant CRM sistemas, ji gali būti šablonas ar modulis tolimesniuose projektuose. Atlieka pagrindines funkcijas grupinio darbo organizavimui klientų aptarnavimo srityje. Šis projektas gali būti papildytas įvairiomis papildomomis funkcijomis, tokiomis kaip:

- privačių žinučių sistema;
- galimybė klientui pačiam užregistruoti problemą;
- automatinis laiškų siuntimas;
- ir kitomis.

Išvardintos funkcijos būtų naudingos, jei TKAS veiktų autonomiškai (atskirai nuo CRM sistemos). Išvardintas savybes paprastai turi CRM sistemos, todėl šios funkcijos nebuvo nagrinėjamas ir projektuojamos.

Igyvendinus projektą ir atlikus testavimą buvo pasiekti projekte aprašyti tikslai, įgyta žinių klientų aptarnavimo srityje, DBVS bei WEB technologijose.

V. Išvados

1. Darbe išanalizuotos ryšiams su klientams valdyti skirtos populiariausios programos. Nustatyta, jog didžiausias dėmesys jose skiriamas tiesioginėms verslo šakoms: rinkodarai, pardavimams, kontaktams, žinių gavybai.
2. Įvertinus naudojamas technologijas galima daryti išvadą, jog CRM sistemų gamintojai naudoja tik populiariausias duomenų bazių valdymo sistemas: MS SQL, ORACLE, DB2. Visos jos užtikrina stabilumą, spartą, saugumą, lankstumą. Tik keletas CRM sistemų gali veikti MySQL ar PostgreSQL DBVS pagrindu, nors pastarosios gali konkuruoti vidutinio verslo įmonėse.
3. Didelis dėmesys nagrinėjamose sistemose yra skiriamas duomenų (žinių) gavybos technologijoms. Tokios technologijos, kaip OLAP ir duomenų sandėliai yra visose CRM sistemose. Duomenų gavybos įrankius turi tik komercinės DBVS (MS SQL, ORACLE, DB2).
4. Komerciniai ir nekomerciniai produktai pastaruoju metu sparčiai populiarėjančias JAVA technologijas (J2EE, J2ME, J2SE). Taip pat naudojamos ASP ir kitos technologijos.
5. Patogių tikslinių priemonių, skirtų aptarnauti klientams (skundų, darbų broko, užduočių registravimas) turi tik keletas 1 lentelėje paminėtų programų.
6. Sukurta programa, kurią galima integruoti į jau esamą klientų apskaitos, ryšiams su klientais valdyti skirtą sistemą.
7. Projektinėje dalyje pateiktas išsamus TKAS projektas, kuriame yra duomenų srautų diagramos, sistemos dinaminis vaizdas ir panaudojimo atvejai, duomenų struktūra. Panaudojus šį projektą, galima sukurti TKAS modulį CRM sistemai.
8. Atlikus sukurtos sistemos bandymus, buvo pastebėta, jog tokios struktūros sistema gali veikti bet kurios veiklos srities įmonėje. Šia sistema gali naudotis net skirtingas paslaugas teikiančios įmonės.

VI. Literatūra

1. Olovía Paar Rud. Data Mining Cookbook. Modeling Data for Marketing, Risk and Customer Relationship Management. 2001
2. CRM Software Selection Kit 2006 08. Prieiga per internetą [Paskutinį kartą žiūrėta: 2007-04-03]: <<http://www.ctsguides.com/crm.asp>>
3. The J2EE 1.4 Tutorial. Interneto prieiga [Paskutinį kartą žiūrėta: 2007-04-03]: <<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/>>
4. PostgreSQL 8.2.4 Documentation. Prieiga per internetą [Paskutinį kartą žiūrėta: 2007-04-03]: <<http://www.postgresql.org/docs/8.2/interactive/index.html>>
5. Oracle Corporation. Customer Relationship Management Call Center Self Study Pack-Part 2. 1999
6. CRM architecture for enterprise relationship marketing in the new millenium. Prieiga per internetą [Paskutinį kartą žiūrėta: 2007-04-03]. <http://h71028.www7.hp.com/enterprise/downloads/CRMArchitecture_Whitepaper_HPC.pdf>
7. Informacinės Technologijos. 2001 Pavasaris (12). Prieiga per internetą [Paskutinį kartą žiūrėta: 2007-04-03]: <[http://www.it.lt/itweb/it3.nsf/0/F6CE9E752F2DA2A942256CAB00666C2F/\\$FILE/informacines_technologijos%20_12.pdf](http://www.it.lt/itweb/it3.nsf/0/F6CE9E752F2DA2A942256CAB00666C2F/$FILE/informacines_technologijos%20_12.pdf)>
8. R. Baronas. Duomenų bazių valdymo sistemos. 2005 m.
9. Mokslinė elektroninė biblioteka. Prieiga per internetą [Paskutinį kartą žiūrėta: 2007-04-03]: <<http://www.elibrary.lt>>
10. EBSCO Mokslinė elektroninė biblioteka. Prieiga per internetą [Paskutinį kartą žiūrėta: 2007-04-03]: <<http://search.ebscohost.com/>>
11. Surajit Chaudhuri, Umeshwar Dayal. An Overview of Data Warehousing and OLAP Technology.
12. IBM Redbooks. Tivoli Management Services Warehouse and Reporting 2007. [Paskutinį kartą žiūrėta: 2007-04-03] Prieiga per internetą: <<http://www.redbooks.ibm.com/abstracts/sg247290.html?Open>>
13. Chris Todman. Designing a Data Warehouse: Supporting Customer Relationship Management. 2000.

VII. Anotacija

Adomas Buškus

Klientų aptarnavimo sistemų tyrimas

Darbe nagrinėjamos ryšiams su klientams valdyti skirtų sistemų raida, funkciniai reikalavimai, technologijos. Aptariami klientų aptarnavimo sistemų įrankiai. Pateikiama tikslinė klientų aptarnavimo sistema, kurios savybės leidžia sistemą taikyti bet kurios veiklos srities įmonėse.

Adomas Buškus

Customer support systems research

In this thesis we look into the customer relationship management systems evolution, functions and technologies. Discuss general customer support systems tools. We present objective customer support system, which give possibility to apply this system to the different business activities.