

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

TVIRTINU
Katedros vedėjas
dr. Eduardas Bareiša
2005 05 23

AUTOMOBILIŲ STOVĖJIMO AIKŠTELIŲ
DAUGIAFUNKCINĖ VALDYMO SISTEMA (APS)

Informatikos mokslų magistro baigiamasis darbas

Kalbos konsultante
Lietuvių k. katedros lektorė
dr. J. Mikelionienė
2005 05 23

Vadovas
prof. dr. E. Kazanavičius
2005 05 23

Recenzentas
Dr. Eugenijus Toldinas

Atliko
IFM-9/2 gr. stud.
Erikas Pliaukšta

KAUNAS
2005

1	ĮVADAS	4
1.1	Dokumento paskirtis	5
1.2	Santrauka	5
2	SISTEMOS ANALITINĖ APŽVALGA	8
2.1	Panašių sistemų analizė	8
2.1.2	Siemens sisteminiai automobilių stovėjimo aikštelių sprendimai	8
2.1.3	Kompanijos Ascom automobilių statymo sistema „E-parking“	9
2.1.4	Webraska ir Schlumberger automobilių stovėjimo sistema.....	10
2.1.5	Situacijos Lietuvoje įvertinimas	12
2.2	Tinklo servिसai.....	14
2.2.1	Apžvalga	14
2.2.2	Tinklo servिसų standartai	15
2.2.3	SOAP	16
2.2.4	DIME	17
2.3	Sistemos specifikacija	18
2.3.1	Projektuojamas objektas	18
2.3.2	Projektuojamo objekto paskirtis	19
2.3.3	Projektuojamo objekto veiklos procesų analizė.....	19
2.3.4	Projektuojamo objekto sistemos funkciniai reikalavimai	22
2.3.5	Reikalavimai APS sisteminiai įrangai.....	23
2.3.6	Reikalavimai serverio techninei įrangai.....	23
2.3.7	Reikalavimų serverio programinei įrangai specifikacija	23
2.3.8	Reikalavimai informacijos posistemei	24
2.3.9	Sistemos vartotojai.....	24
2.3.10	Reikalavimai vartotojo sąsajai	24
2.3.11	Eksplotavimo aplinka.....	26
2.3.12	Reikalavimai projekto dokumentacijai	26
2.3.13	Sistemos apribojimai.....	26
2.3.14	Reikalavimai eksperimentui.....	26
3	SISTEMOS PROJEKTINĖ DALIS.....	27
3.1	Sistemos duomenų bazė.....	27
3.2	Klasių diagrama	34
3.3	Būsenų diagrama.....	37
4	TYRIMO DALIS	38
4.1	Sąsajos testavimas.....	38
4.1.1	Testavimo strategija.....	39
4.1.1.1	Vienetų testavimas	39
4.1.1.2	Integracinis testavimas	39
4.1.1.3	Patvirtinamasis testavimas	39
4.1.1.4	Aukštesnio lygio testavimas	40
4.1.2	Pasirinktas testavimo metodas	41
4.2	Microsoft SOAP Toolkit.....	42
4.2.1	DIME užklauskos struktūra	42
4.2.2	DIME palaikymas SOAP Toolkite	43
4.3	ATL 7.0.....	44
4.3.1	ATL Server bibliotekos struktūra	44
4.3.2	ATL Server aplikacijų architektūra.	46
4.3.3	DIME ATL Server aplikacijose	47
5	EKSPERIMENTINĖ DALIS.....	47
5.2	Sistemos naudojimosi eksperimentas	47
5.2.1	Prisijungimas.....	47

5.2.2	Rezervavimas.....	47
5.2.3	Informacijos įvedimas.....	48
5.2.4	Prisijungimas naudojant WAP technologiją.....	49
5.2.5	Automobilio statymo procesas.....	50
5.2.6	Vartotojo išvykimas.....	50
5.3	Našumo palyginimas perduodant binarinius duomenys skirtingais metodais.	51
6	IŠVADOS.....	55
7	LITERATŪRA.....	57
7.1	Knygos.....	57
7.2	Prieigos per internetą.....	57
8	TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	58
9	PRIEDAI.....	60
9.1	SOAP realizacijos:.....	60
9.2	DIME užklauso antraštė.....	61

1 ĮVADAS

Dabar automobilis tampa būtina susisiekimo priemone, užtikrinančia geresnę gyvenimo kokybę. Dėl šios priežasties automobilių skaičius didėja.

Dauguma firmų, įstaigų, parduotuvių, pramogų centrų yra sutelkta miesto centre. Todėl miesto centre susitelkia didelis automobilių kiekis ir iškyla automobilių stovėjimo problema, didėja triukšmas, užterštumas, nepatogumas automobilių vairuotojams ir aplinkui gyvenantiems žmonėms.

Iki šiol taikyti automobilių stovėjimo problemų sprendimo būdai sprendžia tik stovėjimo paslaugos apmokestinimo problemą.

Automobilių stovėjimo problema aktuali ir yra sprendžiama daugelyje valstybių. Šią problemą Lietuvoje bandoma spręsti apmokestinant automobilių stovėjimo aikšteles miesto centre. Apmokestinimas vykdomas keliais būdais: mokėjimo automatas, apmokamas darbuotojas ir naujusias būdas – SMS trumpąja žinute. Visi šie trys apmokėjimo būdai turi savų privalumų ir trūkumų.

Mokėjimo automato kortelės patogios tuo, kad ją galima naudoti keletą kartų, ją nėra sunku gauti, neturi galiojimo limito laiko atžvilgiu. Pagrindiniai mokėjimo kortelių trūkumai, kad jas galima įsigyti tik nustatyto nominalo, jos negalioja skirtinguose miestuose ir vis tikrai reikia eiti į artimiausia prekybos vietą kurioje galima šią kortelę įsigyti.

Apmokamas darbuotojas kažkiek palengvina apmokėjimo procedūrą, t.y. nebūtina turėti apmokėjimo kortelę (galima sumokėti pačiam darbuotojui), tačiau jis ne visą laiką būna savo darbo vietoje (dėl to gaištamas laikas), jam reikia mokėti atlyginimą SMS trumpojo žinutė yra bene pats patogiausias apmokėjimo būdas, nes nesvarbu, kokiam mieste esi, nereikia turėti su savimi smulkių grynųjų pinigų (mokant darbuotojui) arba mokėjimo kortelės. Tinka visi mobilūs įrenginiai palaikantys SMS žinutes. Trūkumai: reikia turėti specialų lipduką, siunčiant SMS žinutę reikia nurodyti daug informacijos. Vienas bendras visų apmokėjimo rūšių trūkumas, kad mažiausias apmokėjimo intervalas yra 30 min. ir už neišnaudotą laiką pinigai imami neteisėtai.

Įdiegus produktą sumažėja triukšmas, aplinkos užterštumas, taupomas vairuotojų laikas ir aplinkinių gyventojų sveikata. Sistema žymiai palengvina apmokėjimą už stovėjimą, nereikalauja darbuotojų aikštelėse. Vietos rezervavimo funkcija užtikrina, vairuotojui vietą mokamoje stovėjimo aikštelėje. Įvairūs komunikavimo su sistema būdai ir priemonės pritraukia daugiau vartotojų, nes tai patogiu, nėra gaištamasis laikas. Sistema taip pat pateikia išsamią informaciją apie padėtį aikštelėse, vartotojas nukreipiamas iki laisvų vietų.

Darbą atlikinėjome dviese. Danielius Comik ir Erikas Pliaukšta. Danielius Comik atsakingas už automobilių valdymo sistemos projektavimą, Erikas Pliaukšta - programavimą. Atlikome mokslinį tyrimą. Danielius Comik – projektavimo metodų parinkimas ir palyginimas bei domenų perdavimas ir gavimas nuo kliento - serveriui, metodu, kuris buvo sukurtas Toolkit įrankio pagalba, Erikas Pliaukšta - domenų perdavimo ir gavimo nuo kliento - serveriui metodų realizavimas. Domenų perdavimas ir gavimas nuo kliento - serveriui, ATL 7.0 bibliotekos klasėmis, praplečiant jas klasėmis, skirtomis naudotis DIME formtu. Atlikome šių metodų našumo palyginamąją analizę siunčiant binarinius duomenys. Taip pat lyginome su realizacija, kuri buvo atlikta naudojant ATL bibliotekos klases duomenų perdavimui ir gavimui nuo kliento - serveriui nenaudojant DIME formatą.

1.1 Dokumento paskirtis

Dokumentas skirtas projekto reikalavimų suderinimui tarp projekto užsakovo, projekto vadovo ir projekto vykdytojų. Kartu dokumentas yra Kauno technologijos universiteto Informatikos fakulteto Programų inžinerijos katedros magistro baigiamojo darbo ataskaita. Taip pat dokumentas gali būti naudingas susipažinimui su naujausiomis programų sistemomis konkrečioje taikymo srityje.

1.2 Santrauka

Trumpas projekto apibūdinimas

Šiame darbe projektuojama mobili transporto priemonių stovėjimo valdymo sistema. Ši sistema skirta supaprastinti automobilių stovėjimo apmokestinimą, sutaupyti

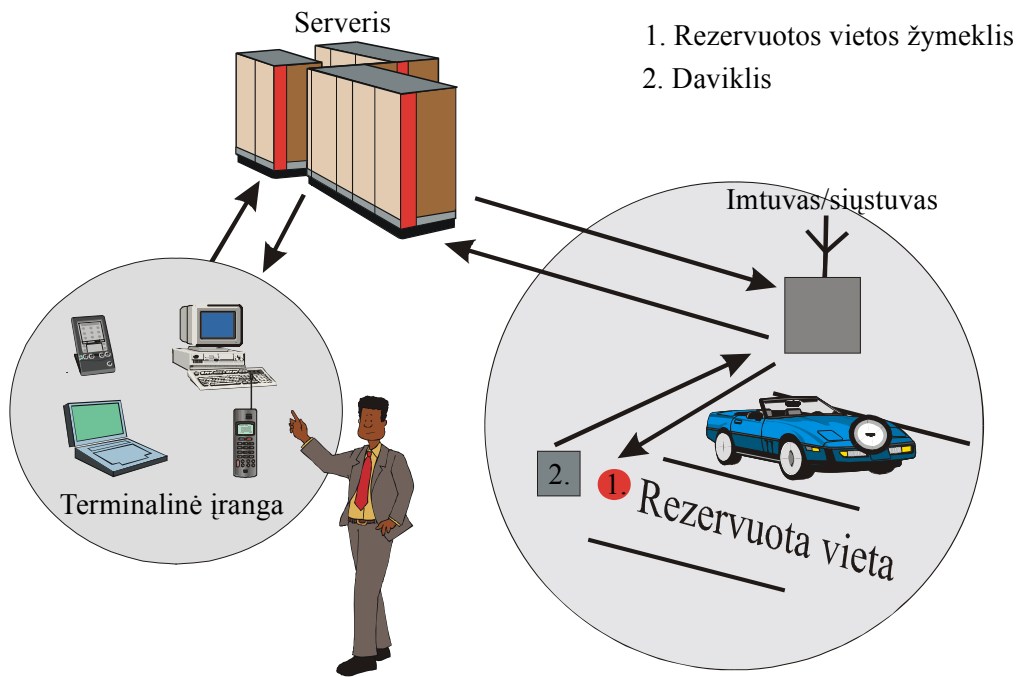
klientų brangų laiką ir pinigus, sumažinti kamščius ir taršą miestų centruose. Įdiegus mūsų stovėjimo sistemą mes išvengsime šių problemų.

Mūsų klientais bus visi vartotojai turintys automobilius ir terminalinę įrangą (asmeninį kompiuterį su priejimu prie internetinio tinklo, WAP ar GPRS technologijas palaikantį mobilųjį telefoną, kišeninį kompiuterį). Apmokėjimas bus labai paprastas, jei jums bus rezervuota vieta, nuo to momento pinigai pradedami skaičiuoti iš jūsų sąskaitos. Jei įmanoma, jums bus atsiusta tiksli informacija kaip nusigauti iki jūsų rezervuotos vietos.

Projekto tikslas

Šiame projekte siūlomas automobilių stovėjimo mokamose stovėjimo aikštelėse problemos sprendimo būdas: sistema, kuri leidžia vartotojui:

- gauti:
 - a) mokamų stovėjimo aikštelių planą;
 - b) laisvų vietų skaičių mokamose stovėjimo aikštelėse;
 - c) maršrutą iki pasirinktos mokamos stovėjimo aikštelės;
- rezervuoti vietą:
 - a) norimam laiko intervalui;
 - b) neribotam laiko intervalui;
- naudotis sistema iš:
 - a) mobilaus telefono turinčio WAP arba GPRS funkcijas;
 - b) asmeninio kompiuterio prijungta prie interneto;
 - c) nešiojamo kompiuterio turinčio “BlueTooth” technologiją;
 - d) kišeninio kompiuterio, palaikančio WAP protokolą arba turinčio “BlueTooth” technologiją.



1 pav. Automobilių stovėjimo aikštelių daugiafunkcinės valdymo sistemos modelis

Projektas yra labai aktualus dėl automobilių didėjimo skaičiaus ir atsirandančios tikrai labai didelės problemos su transporto priemonių stovėjimo. Projektas yra pagrįstas naujomis technologijomis ir yra naujovė mūsų butyje.

Vartotojai – mūsų klientais bus visi vartotojai turintys automobilius ir terminalinę įrangą (asmeninį kompiuterį su priėjimu prie interneto, WAP ar GPRS technologijas palaikantį mobilųjį telefoną, kišeninį kompiuterį).

Užsakovas - Prof. Dr. Egidijus Kazanavičius

Kompiuterių katedra 214c, Tel. 300386

e.paštas: ekaza@ifko.ktu.lt

Sistemos kūrėjai – D.Comik, E. Pliaukšta

Projektavimo planas – 3 semestrai (pusantrų metų)

Projekto įgyvendinimo terminas – 2 metai

2 SISTEMOS ANALITINĖ APŽVALGA

2.1 Panašių sistemų analizė

Norint suprasti kokios užduotys bei reikalavimai yra keliami mūsų projektuojamai sistemai reikia išanalizuoti jau šiuo metu rinkoje esančius produktus, artimus mūsų mobiliai stovėjimo sistemai. Atsiranda vis daugiau firmų, kurios užsiima intelektualių stovėjimo sistemų projektavimu ir gamyba. Vienos firmos dar tik pradedančios ir yra bandomųjų projektų stadijoje, kitos jau pažengusios šioje srityje ir savo projektus pritaikytus konkrečiuose objektuose skaičiuoja šimtais. Taip pat yra keletas universitetinių įstaigų, kurios kuria APS sistemas savo reikmėms, kad optimizuotų automobilių stovėjimo aikšteles savo teritorijose. Daugelis rastų panašių projektų daugiau orientuoti į daugiaaukščius stovėjimo garažus, kuriuose yra įdiegta mobili stovėjimo sistema. Čia pateiksime keletą mūsų apžvelgtų protingų stovėjimo sistemų sąrašą

2.1.2 Siemens sisteminiai automobilių stovėjimo aikštelių sprendimai

Kai žmonės kalba apie automobilių kamščius miestuose, jie įsivaizduoja daug automobilių vairuotojų, norinčių patekti į jiems reikalingą vietą. Realybė yra kitokia. Kaip parodė tyrimai – beveik 40% vairuotojų sukiojasi ratais ieškodami laisvų stovėjimo vietų, todėl susidaro eismo kamščiai [4]. Vienas iš šios problemos sprendimo būdų – padėti vairuotojams rasti laisvas stovėjimo vietas. Kaip rodo patirtis, eismo kompleksiskumas reikalauja rimtų ir apgalvotų sprendimų, atsižvelgiant į visus statistinius kriterijus. Net ir geriausios stovėjimo vietų informacijos sistemos negalės padėti, jei prie daugiaaukščių stovėjimo aikštelių susidarys automobilių eilės. Automobilių stovėjimo valdymo sistema turi veikti taip vieningai, kaip įranga įrengta periferijoje ir turi būti visiškai integruota. Visa tai siejama su centrine darbo stotimi bei jai priklausančia technine bei programine įranga, tame tarpe ir užimtumo sensoriais, įvažiavimo ir išvažiavimo sistemomis, bilietų išdavimo įtaisais ir apmokėjimo aparatais. Tik susieta sistema gali koordinuoti :

- laisvų stovėjimo vietų stebėjimą
- nukreipimą į stovėjimo vietas
- stovėjimo apmokestinimą

- duomenų perdavimą
- klaidų valdymą

Siemens automobilių stovėjimo vietų valdymo sistema parodo optimalų visų šių reikalavimų įgyvendinimą. Integruotos sistemos apimančios pilną paslaugų rinkinį filosofija pasirodė esanti tinkamiausia žiūrint į ateitį.

Tokie paminėti sprendimai jau veikia Vokietijos didmiesčių priemiesčiuose, kur dinaminės automobilių stovėjimo valdymo sistemos nukreipia vairuotojus į laisvas stovėjimo vietas centre arba artimiausias stovėjimo aikšteles.

Stovėjimo vietos daugiaaukštėse aikštelėse nuolat stebimos ultragarsiniais sensoriais, kurie į kontrolės centrą grąžina vieną reikšmę iš dviejų – laisva arba užimta.

Pačiose stovėjimo aikštelėse vairuotojas sekdamas aiškias nuorodas ekranuose nesunkiai ir greitai randa laisvą vietą.

Ir netgi toks kasdieninis darbas, kaip apmokestinimas už stovėjimą gali būti supaprastintas naudojant naujoviškas technologijas.

Nauji displėjiniai apmokėjimo aparatai, tokie kaip Siemens (SITY ir PRISMA), leidžia nenaudojant grynųjų pinigų apmokėti už stovėjimą pasitelkiant mobilųjį radijo ryšį.

Kyla klausimas ar tokios aukštos kokybės technologijos, kurių nauda eismo reguliavimui neginčijama, bus prieinamos tik pagrindiniams šalies miestams ar ir mažesniesiems.

2.1.3 Kompanijos Ascom automobilių statymo sistema „E-parking“.

Ši sistema skirta stovėjimo aikštelių išankstiniam, mobiliam rezervavimui. Jų duomenimis Europos Sąjungoje nuo 20 iki 30 procentų susidarančių kamščių miestuose kyla dėl to, kad vairuotojai ieško vietos, kur pastatyti savo automobilį. Buvo išbandyta daug variantų, laisvų vietų informavimo sistema, supaprastinta automobilių statymo sistema, bet laisvų vietų trūkumas ir toliau buvo jaučiamas.

Projekto esmė yra internetas kaip priemonė rezervuoti vietą ir sumokėti tam tikrą mokestį [1]. Ši sistema gali būti pritaikyta aerouostų lankytojams, žmonėms einantiems į koncertus, sporto aistruoliams, kino filmų žiūrėtojams ir kiekvienam, kas ieško laikinos

automobilio pastatymo vietos. Tuomet jie galėtų tiesiog atvažiuoti, pastatyti automobilį ir eiti kur jiems reikia, o ne sėdėti eilėje ir važinėti aplinkui.

Žmogus ieškantis laisvos vietos gali ją rezervuoti per internetą. Tada visa informacija perduodama į automobilių statymo sistemą, kad įsitikinti, ar ta vieta bus laisva kai vairuotojas atvyks. Vairuotojas bus identifikuojamas pagal mašinos numerį arba pagal kreditinės kortelės numerį. Šis naujoviškas projektas suderina Ascom patirtį automobilių statymo sistemose (daugiau nei 700 sistemų įrengtų aerouostuose, prie prekybos centrų, viešbučių ir ligoninių). Automatinė numerio atpažinimo sistema, naudojanti video kameras, nuskenuoja automobilio numerį ir palygina jį su numeriu esančiu registracijoje ir automatiškai nustatanti ar galima tam automobiliui įvažiuoti į aikštelę. Išvažiuoti gali tik tie automobiliai kurių numeriai yra užregistruoti duomenų bazėje.

Kad dar padidinti elektroninio mokėjimo funkcijas, yra integruojamos mobiliosios komunikacijos. Pavyzdžiui, vairuotojas gali rezervuoti vietą ne tik per asmeninį kompiuterį, bet ir per mobilųjį telefoną, kuris suderintas su GPRS arba WAP, kol jis yra kelyje. Tada rezervacija bus patvirtinta į mobilųjį telefoną, pavyzdžiui, atsiunčiant patvirtinimo kodą, su kuriuo jis gali įvažiuoti į stovėjimo aikštelę, tai būtų virtualus įvažiavimo bilietas.

„E-parking“ garantuoja vartotoją ir užtikrina laisvą vietą jo automobiliui. Taip pat „E-parking“ siūlo idealų sprendimą didelių aikštelių priežiūrai tiesiogiai per internetą. Šis sprendimas leidžia operatoriui matyti dienos ataskaitą ir statistiką automatiškai kompiuterio ekrane.

2.1.4 Webraska ir Schlumberger automobilių stovėjimo sistema

2000-ųjų Kovo 9 diena, Paryžiuje mobiliųjų technologijų bendrovės Webraska ir Schlumberger praneša apie bendradarbiavimą pademonstruojant naują, padedančią sutaupyti daug brangaus laiko, automobilių vairuotojams paslaugą [5]. Analizuojant duomenis apie stovėjimo vietų užimtumą, ar tai būtų gatvė ar automobilių stovėjimo aikštelė, ir kombinuojant juos su duomenimis apie realią eismo situaciją mieste, bei atsižvelgiant į žemėlapius ir maršrutą – automobilineiškai naudodamiesi mobiliaisiais telefonais su WAP palaikymu, galės gauti pačią naujausią informaciją apie mašinos statymo galimybę atvykimo vietoje.

Su naująja paslauga, WAP mobiliųjų telefonų vartotojai galės suderinti automobilio pastatymo vietą toje apylinkėje į kurią jie vyksta. Tai bus įmanoma naujoviškos Schlumberger ir Webraskos internetu pagrįstos išskirstytos navigacijos paslaugos dėka. Ši sistema leidžia vartotojui gauti informaciją apie laisvas stovėjimo vietas aplinkinėse gatvėse ir mašinų aikštelėse realiaame laike, taip pat galima sužinoti informaciją apie eismo būklę toje vietoje, į kurią jūs vykstate, bei padeda pasirinkti optimaliausią kelią į jūsų pasirinktą stovėjimo vietą

Schlumberger automobilių stovėjimo terminalai ir valdymo sistemos persiūs esamos padėties informaciją per GSM ar Interneto tinklus į atviros sąsajos Automobilių Stovėjimo Informacijos Darbo stotį. Darbo stotis skaičiuoja konkretaus rajono užimtumo lygius bei dar keletą dinaminių charakteristikų, tokių kaip, ar laisvų stovėjimo vietų skaičius didėja ar mažėja. Šios charakteristikos leidžia vairuotojui pasirinkti optimalų automobilio statumą, taip sutaupant daug laiko. Tada Webraska darbo stotis apskaičiuos geriausią maršrutą ir padės telefono naudotojui nusigauti į pasirinktą vietą.

Virš 95% Schlumberger automobilių stovėjimo terminalų šiuo metu įrengtų įvairiose pasaulio vietose gali būti lengvai atnaujinti, kad galėtų generuoti būtinus duomenis, kurie suteiktų žymią galimybę vietinei ir miesto valdžiai pagerinti jų paslaugų įvaizdį vairuotojams.

Ši partnerystė su Schlumberger yra logiškas Webraska žingsnis papildant svarbų miesto mobilumo aspektą – kur palikti savo automobilį atvykus – iki dabartinės realaus laiko paslaugos, kuri padeda mobiliųjų telefonų naudotojams pasiekti savo tikslą be didelių pastangų. Šis susitarimas taip pat įgalina tiekti mobiliesiems operatoriams ir jų klientams papildomos vertės paslaugas.

Webraska – pirmoji pasaulyje pradėjo tiekti pridėtosios vertės realaus laiko informaciją, ir navigacijos paslaugas mobiliuosiuose telefonuose, paversdama juos į pagalbinius. Webraska tikslas pateikti paslankią ir išbaigtą alternatyvą kompaktinių diskų grotuvų pavidalu egzistuojančioms automobilių navigacijos sistemoms, naudojant jos pačios užpatentuotą internetu pagrįstą paskirstytą navigaciją. Šios kompanijos klientai – telekomo operatoriai norintys uždirbti naują pelną ir sukurti naują paslaugų rūšį, atitinkančią pagrindines jų vartotojų reikšmes. Pradžioje sukurtos kartu su pirmaujančia Prancūzijos telefonijos operatore SFR, Webraska paslaugos yra sukurtos ant nešėjo – laipsnio kokybės skaitmeninių žemėlapių darbo stočių pritaikytų aptarnauti milijonus vartotojų. Į paslaugas įeina WAP priėjimas, JAVA pagrįstos skambučių centro sąsajos, XML užklauso IVR sistemoms, trumpųjų žinučių prisijungimo galimybė, interneto

palaikymas ir vizualizacija. Webraska paslaugos prieinamos SFR abonentams Prancūzijoje ir KPN abonentams Nyderlanduose.

Schlumberger sprendimai savivaldybėms supažindina miesto tarybas ir vietines valdžias su susietais protingomis kortelėmis pagrįstais sprendimais, tranzitui, automobilių statymu ir telekomunikacijoms.

„Schlumberger Test & Transactions“ pateikia protingosiomis kortelėmis pagrįstus sprendimus; puslaidininkinę įrangą ir bendrus IP ir tinklų sprendimus klientams visame pasaulyje.

2.1.5 Situacijos Lietuvoje įvertinimas

Automobilių stovėjimo problema aktuali ir yra sprendžiama daugelyje valstybių. Šią problemą Lietuvoje bandoma spręsti apmokestinant automobilių aikšteles miesto centre. Dažniausiai apmokestinimai vykdomi šiais būdais: sumokama mokėjimo automatu, apmokamam darbuotojui arba trumpąja žinute. Visi šie trys apmokėjimo būdai turi savų privalumų ir trūkumų.

Mokėjimo automato kortelės patogios tuo, kad ją galima naudoti keletą kartų, ją nesunku gauti, neturi galiojimo limitu laiko atžvilgiu. Pagrindiniai mokėjimo kortelių trūkumai, kad jas galima įsigyti tik nustatyto nominalo, jos negalioja skirtinguose miestuose ir vis tikrai reikia eiti į artimiausią prekybos vietą kurioje galima šią kortelę įsigyti.

Kai kuriuose miestuose galima sumokėti apmokamam darbuotojui kai automobilis pastatomas į stovėjimo vietą. Be to jis tikrina pažeidimus aikštelėse, tačiau jį ne visą laiką galima rasti (dėl to gaišamas laikas), jam reikia mokėti atlyginimą.

Kitas būdas - mokėti trumpąja žinute už stovėjimą. Tačiau norint tokiu mokėjimu pasinaudoti reikia Omnitel arba Bitė prekybos salone įsigyti lipduką, o taip pat stovėjimo aikštelės kontrolieriaus galima paprašyti (Vilniuje - su brūkšniniu kodu) ir užklijuoti jį matomoje vietoje, ant priekinio automobilio stiklo. Trūkumai - negalima užsisakyti stovėjimo vietos ir laiko, paslauga gali naudotis ne visų mobiliojo ryšio tinklų klientai.

Remiantis daugelio pasaulio valstybių patirtimi automobilių stovėjimo aikštelių srityje, 1997 metų spalio 14-tą Kauno mieste buvo įdiegta automatizuota mokama automobilių stovėjimo sistema. Kauniečiai buvo pirmieji Lietuvoje įdiegę tokią sistemą. Vairuotojas gali atvažiuoti ir sumokėti už stovėjimą. Pastatyti automobilį jis gali tik tada kai yra tuščia stovėjimo vieta. Jis negali išlanksto užsisakyti vietą, ką mes ir bandome

padaryti. Mokėti už automobilių stovėjimą Kauno miesto centrinėje dalyje galima elektronine kortele į automobilių stovėjimo mokėjimo automata . Tai nėra labai patogu. Neturintis kortelės vairuotojas turi laukti kol ateis aptarnaujantis žmogus ir duos jam mokėjimo čekį. Prarandamas brangus laikas. Dabar galima pirkti stovėjimo bilietus „Kauno spauda“ kioskuose, bet vėlgi iki jų reikia eiti ir pirkti.

Tokie yra žinomi automobilių statymo būdai Lietuvoje.

Žemiau pateikta šių apžvelgtų sistemų palyginimų lentelė (1 lentelė).

1 lentelė. Sistemų palyginimas

Sistema	Reikalingas interneto ryšys	Reikalingas mobilusis telefonas	Būtinasis susisiekimasis su firma dėl papildomos įrangos	Paprastumas naudotis
Automobilių aikštelių sprendimai Lietuvoje	Ne	Nebūtinasis	Nebūtinasis	Ne
Ascom – „E-parking“	Taip	Nebūtinasis	Ne	Taip
EMT – „Mobile Parking“	Ne	Taip	Ne	Taip
AirClic – „Parking Services“	Ne	Taip	Taip	Taip
„IPS“	Taip	Taip	Ne	Reikia įvedinėti slapta kodą
Webraska ir Schlumberger automobilių stovėjimo sistema	Taip	Taip	Ne	Taip

2.2 Tinklo servisas

2.2.1 Apžvalga

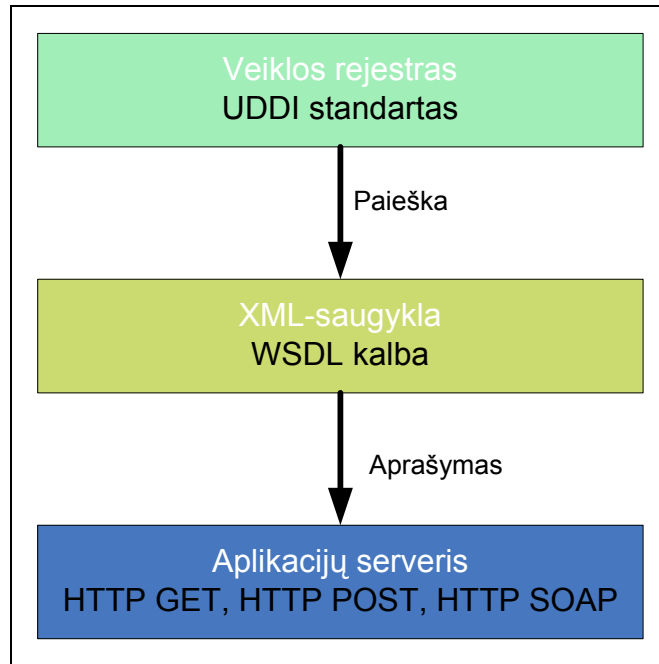
Pradžioje Worl Wide Web buvo dokumentų tinklas. Serveriai bendravo su klientais per HTTP protokolą ir persiūsdavo informaciją naudojant hipertekstinių dokumentų pagalba, kurie buvo sukurti HTML kalbos priemonėmis. Tokie dokumentai buvo atvaizduojami naršyklėse ir turėjo nuorodas į kitus dokumentus. To visiškai pakako tinklo kūrėjams informacijos apsikeitimui. Atsirado komerciniai puslapiai, kurie reklamavo įvairių kompanijų produktus. Tokie puslapiai sudaro daugumą ir dabartiniame Internete. Vėliau atsirado poreikis palaikyti pastovias transakcijas su klientais. Tokie puslapiai tapo elektroninės komercijos saitais. Dauguma komercinių saitų kuriami ne paprastų Web-serverių pagrindu, o naudojant daugiafunkcinę architektūrą, kuri leidžia naudoti serverio aplikacijas. Pagrindinis skirtumas tarp paprastų serverių ir serverių aplikacijų yra tas, kad aplikacija ne tik gražina klientui dokumentą, bet gali apdoroti kliento užklausą, o taip pat į ją įeina programinis kodas, kuriame realizuota veiklos logika. Dažniausiai serverio aplikacijos generuoja gražinamą dokumentą dinamiškai pagal kliento parametrus. Taip pat reikia pažymėti, kad serverio aplikacijos leidžia sukurti sprendimus leidžiančius aptarnauti daug vartotoju vienu metu.

Elektroninė komercija negali atsiriboti tik paprastu kliento užklausoje apdorojimu. Sekantis Interneto vystymosi žingsnis tapo skirtingų kompanijų veiklos procesų integracija. Atsirado orientuotas į servisą Internetas. Jo pagrindą sudaro dvi technologijos – SOAP ir XML. Pagal šį scenarijų tinklą sudaro serverių aplikacijų rinkinys. Šios aplikacijos keičiasi tarpusavy informacija XML formatu per SOAP protokolą. Orientuoto į servisą tinklo pagrindas yra Tinklo servisas – logiškai surištų funkcijų rinkinys, kurios gali būti iškviečiamos per Internetą. Informaciją apie galimas tinklo serviso funkcijas yra laikoma WSDL dokumente, o egzistuojančių tinklo servisų paieškai siūloma naudoti specialius rejestrus, kurie specifikuoti pagal UDDI.

2.2.2 Tinklo servisų standartai

Tinklo servisi remiasi trejais pagrindiniais standartais:

- SOAP – protokolas pranešimų siuntimui per HTTP ir kitus Interneto protokolus;
- WSDL – tinklo servisų programinių vienetų aprašymo kalba;
- UDDI – tinklo servisų indeksacijos standartas.



2 pav. Tinklo servisų standartų sąryšis

Aplikacijų serveris yra tinklo servisų saugykla. Jie yra prieinami per protokolus HTTP GET, HTTP POST arba HTTP SOAP.

Egzistuojantys tinklo servisi aprašomi WSDL dokumentuose, kurie laikomi aplikacijų serveryje arba specialiuose XML saugyklose. WSDL dokumentai gali kreiptis į kitus WSDL dokumentus arba į XSD dokumentus, kuriose aprašomi duomenų tipai, kurios naudoja tinklo servisi. XML saugyklos naudojamos WSDL dokumentų valdymui. WSDL dokumento viduje yra tinklo serviso adresas (URL). Tinklo servisi yra aprašyti ir indeksuoti veiklos rejestre, kuriame yra adresai(URL) į WSDL dokumentus.

2.2.3 SOAP

SOAP – tai protokolas skirtas informacijos apsikeitimui tarp sistemų komponentų. SOAP nustato bendravimo tarp kliento ir serverio standartą, kaip turi būti atliktas iškvietimas, perdavinėjami parametrai ir grąžinamos reikšmės. Bet kokios informacijos, kuri perduodama tarp klientinės ir serverinės dalies, atvaizdavimui naudojamas XML. SOAP neribuoja transporto protokolo naudojimą. Duomenų perdavimui gali būti panaudoti bet kokie protokolai ir produktai. Pavyzdžiui HTTP, HTTPS, SMTP. Tačiau dažniausiai naudojamas protokolas HTTP.

Persikirstymo sistemose SOAP naudojamas įvairių jos komponentų tarpusavio bendravimui. Šio metu egzistuoja daug technologijų ir protokolų, kurie leistų nesunkiai išspręsti šią užduotį. Viena iš jų yra DCOM, kuri leidžia efektyviai realizuoti RPC kreipinius, perduoti ir priimti duomenys, persikirstyti apkrovimą tarp kelių serverių. Tačiau, sistemose, naudojančiose DCOM yra labai žymus trukumas, kuris apsunkina skirtingų sistemos lygmenų bendravimą per Internetą. Nors DCOM aplikacijos gali naudoti TCP/IP protokolą RPC iškvietimų perdavimui, dauguma šiuolaikinių tinklinių mazgų uždraustų tokių paketų perdavimą dėl saugumo sumetimų. Galima pasakyti, kad DCOM yra dominuojanti technologija informacijos apsikeitimui vidinio tinklo ribose, bet kai reikia perduoti informaciją už vidinio tinklo ribų, DCOM yra neefektyvi.

DCOM alternatyva kuriant persikirstymo sistemas gali būti Web-sąsaja ASP pagrindu. Tokiais atvejais klientinėje dalyje turi būti tik naršyklė, kuri gali prisijungti prie Web-serverio. Perduodant užklausą iš kliento iki serverio galima panaudoti POST metodą, o užklausos apdorojimas vykėtų serverinėje dalyje. Nežiūrint į tokio metodo populiarumą, jis turi labai didelį trukumą. Perduodamos informacijos sudėtis, o svarbiausia, jos perdavimo būdas yra specifikuojamas konkrečiai aplikacijai. O jeigu atvaizdavimo lygyje reikalinga pilnavertė Windows aplikacijos vartotojo sąsaja, kai kuriais atvejais tai gali būti labai sunku realizuoti.

Kaip gi šio atveju galima sukurti bendravimą tarp kliento ir serverio? Ir kaip atlikti šį bendravimą, jeigu būtina užtikrinti palaikymą ne tik HTTP, bet ir SMTP arba MSMQ? Šias užduotys galima išspręsti tokiu būdu:

- Performuoti iškvietimą į tekstinį pavidalą išsaugojant informaciją apie visus parametrus ir jų reikšmes
- Perduoti šį tekstą serveriui per HTTP, SMTP arba kitą norimą protokolą.

- Interpretuoti tekstinį pranešimą serveryje ir atlikti reikiamo metodo iškvietimą
- Suformuoti išeinamus parametrus arba informaciją apie klaidą, kurie išsaugomi teksto pavidalu.
- Perduoti šį tekstą klientui per HTTP, SMTP arba kitą norimą protokolą.
- Interpretuoti rezultata.

Realizuojant aprašytą scenarijų, galima suprojektuoti savo parametrų ir iškvietimo informacijos perdavimo formatą arba naudoti SOAP, tuo sumažinant išlaidas realizavimui, dokumentavimui ir palaikymui.

Taip pat labai didelis SOAP privalumas yra jo neutralumas platformai, t.y. neapribuoja kokias platformas naudoja klientas ir serveris. Kliento, kuris dirba Windows šeimos sistemoje, užklausa gali būti sėkmingai apdorota Unix serveryje.

2.2.4 DIME

Neseniai SOAP aplikacijų ir XML tinklo servisų pasaulyje atsirado terminas DIME (Direct Internet Message Encapsulation). DIME – tai skirtingos kilmės duomenų, tokių kaip XML, binariniai srautai, atvaizdai ir video, prisegimo prie vieno pranešimo specifikacija. Nors DIME yra bendros paskirties formatas, tačiau jis buvo kuriamas efektyviam duomenų perdavimui naudojant SOAP protokolą.

Pagrindinis SOAP protokolo privalumas (platformos neutralumas, lankstumas ir plėtrumas), o tuo pačiu metu jo trukumas (žiūrint iš efektyvumo pusės) – tai XML naudojimas užklausų aprašymui. XML užtikrina struktūrizuotą ir tvarkingą darbą su SOAP užklausomis. Bet XML yra tekstinis formatas ir jis negali efektyviai aprašyti binarinių duomenų srautus. Tam, kad persiusti tokius duomenys XML dokumente jos reikia reformuoti į base64. Toks reformavimas turi kelis trukumus:

- Tiesioginiam ir atgaliniam reformavimui gaištami procesoriaus resursai kliento ir serverio dalyse.
- Reformuojant duomenis į base64 duomenų apimtis padidėja 30%, o tai padidina XML dokumento dydį, o kai kurie plačiai naudojami XML skaitytuvai užkrauna visą dokumentą į atmintį.

Problemos iškyla ne tik perduodant binarinį srautą per XML, bet taip pat perduodant kitų XML dokumentų fragmentus. Tai atsitinka dėl to, kad skirtingi XML dokumentai gali būti skirtingai koduojami, turėti skirtingas struktūras.

Formatas DIME buvo sukurtas, kad leisti organizuoti perdavimą skirtingos kilmės duomenys vienoje XML SOAP užklausoje. Šis formatas aprašo tokių duomenų apjungimo metodus į vieną užklausą. Pats DIME yra bendros paskirties formatas, todėl jis neriboja transporto protokolą (tai gali būti HTTP arba TCP/IP) ir duomenų struktūros.

DIME yra ne pirmas formatas skirtingų duomenų formavimui į vieną užklausą. Elektroninio pašto žinutėse naudojamas MIME formatas. Šis formatas operuoja metaduomenimis ir yra labai lankstus, tačiau dėl šio lankstumo buvo paaukotas efektyvumas. Pavyzdžiui, tam, kad suskaičiuoti prisegtų objektų skaičių reikia peržiūrėti visą duomenų srautą. Dėl šios priežasties tinklo servisuose skirtingos kilmės duomenų formavimui į vieną žinutę efektyviau naudoti DIME formatą, kuris buvo suprojektuotas taip, kad kiek galima palengvinti užklausų formavimą ir skaitymą. Pavyzdžiui, jeigu DIME užklausoje yra penki prisegti objektai, tai galima lengvai perskaityti penktą elementą, neanalizuojant ir neapdorojant prieš tai esančius užklaunos elementus.

DIME gali būti naudojamas bet kokiose sistemose, kai reikia perduoti duomenų visumą viename loginiame vienete. XML tinklo servisuose DIME naudojamas prisegant prie XML užklaunos bet kokio kiekio papildomų duomenų.

2.3 Sistemos specifikacija

2.3.1 Projektuojamas objektas

Šiame darbe yra projektuojama automobilių stovėjimo aikštelių daugiafunkcinė valdymo sistema (APS). Ši sistema skirta supaprastinti automobilių stovėjimo apmokestinimą, sutaupyti klientų brangų laiką ir pinigus, sumažinti kamščius ir taršą miestų centruose, išlaisvinti darbo jėgą; pagerinti apmokėjimo kokybę; padidinti apskaitos efektyvumą;

Mūsų klientais bus visi vartotojai turintys automobilius ir terminalinę įrangą (asmeninį kompiuterį su priėjimu prie interneto, WAP ar GPRS technologijas palaikantį mobilųjį telefoną, kišeninį kompiuterį).

2.3.2 Projektuojamo objekto paskirtis

Suteikti vartotojui galimybę rezervuoti stovėjimo vietą iš anksto, gauti aikštelių planus ir kitą informaciją, susijusią su stovėjimo vietomis.

2.3.3 Projektuojamo objekto veiklos procesų analizė

Šiame etape analizuojama mobiliosios transporto priemonių stovėjimo valdymo sistemos veiklos procesas.

Vartotojo funkciniai reikalavimai

Ši sistema yra skirta klientų patogumui, todėl projektuojant šią sistemą remiamasi į vartotojo poreikius. Šios sistemos sėkmė priklauso nuo jos pasisekimo rinkoje, o tai sąlygoja jos paprastas įdiegimas ir elementarus naudojimas. Klientui svarbiausia yra jo sutaupyta laikas ir gautas rezultatas. Šiais faktoriais ir yra paremta kuriama sistema.

Mūsų klientai - asmenys dažnai turintys reikalų miestų centruose ir naudojančys savo asmenines transporto priemones.

Mūsų klientai – asmenys, naudojančys kompiuterines (asmeninį kompiuterį su prieiga prie interneto, kišeninį kompiuterį) bei ryšių technologijas (WAP protokolą, GPRS ar “BlueTooth” technologiją palaikančią mobilųjį telefoną ar kitą terminalinę įrangą turinčią šias technologijas).

Mūsų klientai – asmenys, norintys sutaupyti laiką ir pinigus.

Automobilių stovėjimo aikštelių daugiavfunkcinė valdymo sistema suteikia vartotojui:

- gauti:
 - mokamų stovėjimo aikštelių planą;
 - laisvų vietų skaičių mokamose stovėjimo aikštelėse;
 - maršrutą iki pasirinktos mokamos stovėjimo aikštelės;
- rezervuoti vietą:
 - norimam laiko intervalui;
 - neribotam laiko intervalui;
- naudotis sistema iš:
 - mobilaus telefono turinčio WAP naršyklę;
 - asmeninio kompiuterio prijungta prie interneto;

nešiojamo kompiuterio turinčio “BlueTooth” technologiją;

Reikalinga taikomoji sisteminė įranga

Pilnam sistemos funkcionavimui užtikrinti reikalinga: techninė įranga; programinė įranga; ryšių priemonės.

Sistemos funkcionalumą užtikrins priemonės teikiamos mobilaus ryšio operatoriaus arba prieiga prie interneto.

Reikalinga:

- mobilus telefonas, palaikantis WAP protokolą, ar GPRS paketinio duomenų perdavimo technologija;
- asmeninis kompiuteris su prieiga prie internetu;
- nešiojamas kompiuteris palaikantis “BlueTooth” technologiją;
- kišeninis kompiuteris, palaikantis WAP protokolą ir turintis “BlueTooth” technologiją;

APS sistemai reikalinga :

- vieta mobilaus ryšio operatoriaus serveryje;
- apmokėjimo sistema, e. apmokėjimo sistema
- teisiniai aktai su miesto savivaldybe, dėl:
 - sistemos įdiegimo;
 - nedrausmintų vartotojų atsakomybės;
- programinė įranga
- duomenų bazė

Automobilių stovėjimo aikštei reikalinga:

- techninė įranga
 - davikliai-sensoriai, tikrinantys ir siunčiantys informaciją apie stovėjimo vietos užimtumą;
 - rezervuotos vietos žymeklis;
 - siūstuvai/imtuvai, surenkantis duomenis iš daviklių stovėjimo vietoje, komunikuojantis su serveriu ir siunčiantis signalus į rezervuotos vietos žymeklį.
 - Skaitliukas

Ši sistema yra maksimaliai priklausoma nuo terminalinės įrangos, kuria vartotojas naudojami teikiama paslauga (mobilus telefonas, nešiojamas kompiuteris, stalinis kompiuteris, kišeninis kompiuteris ir kitais prietaisais, kuriais galima naudotis internetu).

Funkcionalumas

Čia pateiktas sistemos funkcionalumo veiklos procesas : nuo momento kaip vartotas užsisako paslaugą iki momento kai palieka stovėjimo vietą.

Mobilios transporto stovėjimo sistemos valdymo tikslas: padėti vartotojui greitai ir patogiai susirasti ir rezervuoti vietą mokamoje automobilių stovėjimo aikštelėje.

I etapas: vartotojui suteikiama informacija apie esama padėtį automobilių stovėjimo aikštelėje.

- Vietos sensorius, sujungtas su tai gatvei arba tam tikram stovėjimo vietų zonai pastatytu aparatu (vietinė duomenų bazė), nuolat siunčia informaciją apie konkrečią stovėjimo vietą, o šie persiunčia ją į pagrindinę duomenų bazę.
- vartotojas pasirinktu būdu (Wap, Sms ar GPRS, internetu) siunčia užklausa į sistemos duomenų bazę ar aikštelėse yra laisvų stovėjimo vietų, prieš tai identifikuojant aikštelę (pvz. Gedimino gatvė – 2). Ši informacija yra nuolat atnaujinama. Svarbu tai jog norint gauti detalų nuvykimo planą iki stovėjimo vietos reikia prieš siunčiant užklausa tai nurodyti(dar nenuvirta koks bus išskirtinis ženklas)
- Sistema informuoja klientą apie laisvas vietas(vietos identifikuojamos) ir atsiunčiamas i ekraną nuvykimo planas jei buvo pageidauta.

II etapas: vartotojas rezervuoja automobilio stovėjimo vietą

- vartotojas pasirinktu būdu (Wap protokolu, sms, GPRS technologija, internetu) siunčia užklausa skirtą vietos rezervavimui nurodant vietos identifikatorių, laiką (iki kada rezervuojama konkreti vieta) į sistemos duomenų bazę.
- gaunamas patvirtinimas apie rezervuotą norimą stovėjimo vietą ir nuo to momento skaitliukas (*tai mechaninis įrenginys kaupiantis informaciją apie laisvas ir užimtas stovėjimo vietas, taip pat atliekantis laikmačio funkciją*) pradeda skaičiuoti laiką.

III etapas: atvyksta vartotojas į automobilio stovėjimo vietą aikštelėje.

- atvyksta klientas iš anksto rezervavęs vietą, ir vietos sensorius parodo jo teisėtą buvimą šioj vietoj.
- atvyksta klientas iš anksto nerezervavęs vietos ir jeigu yra laisvų vietų jis užsiparkuoja. Nuo to momento skaitliukas pradeda skaičiuoti laiką už kurį klientas yra sumokėjas.

IV etapas: išvyksta iš automobilių stovėjimo aikštelės.

- stovėjimo vietą palieka vartotojo automobilis. Sensorius reaguoja ir praneša į serverį, apie laisvą vietą. Sistemos serverio skaitliukas atnaujina informaciją apie laisvas vietas duomenų bazėje. Duodamas signalas vietos sensoriui, uždegamas žalias indikatorius, t.y. vieta laisva.
- Apmokestinimo sistema pagal prastovėtą kliento laiką ir nustatytus tarifus nustatyto užmokesčio dydį ir jį nuskaito nuo vartotojo abonentinės sąskaitos, kurią jis turi šias paslaugas teikiančioje firmoje. Visi pinigai sumokėti kliento už mūsų stovėjimo ir rezervavimo paslaugą bus įtraukiami į aktoriaus asmeninę telefono sąskaita. Jis mokės už tai kaip už ryšio paslaugas. Gale mėnesio jam bus įskaičiuojama į jo sąskaitą visos jo stovėjimo ir rezervavimo operacijos.

2.3.4 Projektuojamo objekto sistemos funkciniai reikalavimai

Mūsų projektuojama sistema atliks šias funkcijas ir leis:

- vartotojui greitai ir patogiai gauti informaciją apie mokamas stovėjimo aikšteles:
 - mokamų stovėjimo aikštelių planą;
 - laisvų vietų skaičių mokamose stovėjimo aikštelėse;
 - maršrutą iki pasirinktos mokamos stovėjimo aikštelės;
- rezervuoti vietą:
 - norimam laiko intervalui;
 - neribotam laiko intervalui;
- išlaisvinti darbo jėgą;
- pagerinti apmokėjimo kokybę;
- padidinti apskaitos efektyvumą;
- sumažinti aplinkos taršą ir triukšmą;

2.3.5 Reikalavimai APS sisteminei įrangai

Sistema sudaryta iš trijų dalių: serverinės, duomenų surinkimo ir siuntimo. Reikalavimai posistemėms yra išdėstyti žemiau esančiuose punktuose.

2.3.6 Reikalavimai serverio techninei įrangai

Reikalavimai serverinei daliai:

- kompiuteris su Intel Pentium III 1000Mhz procesoriumi,
- 512 MB operatyviosios atmintinės,
- nuolatinio interneto bei mobiliųjų paslaugų tiekėjo ryšio.

Duomenų surinkimo ir siuntimo daliai:

- vartotojo terminalinės įrangos,
- mobilaus ryšio operatoriaus serverio,
- siųstuvo/įmtuvo, daviklio ir žymeklio.

2.3.7 Reikalavimų serverio programinei įrangai specifikacija

Operacinė sistema

Microsoft Windows 95/98/ME/2000/XP

Microsoft Windows NT 4.0

Projektui realizuoti buvo reikalinga ši programinė įranga:

Programinė įranga:	Gamintojas	Licenzijos kaina
Java <u>J2SE 1.4.1 SDK</u> programavimo kalba	2000 © Sun Microsystems	nemokama
MySQL duomenų bazių kūrimo kalba	MySQL AB	400\$
jakarta-tomcat-4.1.24	© 2002 The Apache Software Foundation	nemokama
ArgoUML projektavimo kalba	Open Source Software Engineering	nemokama
BPWin	“Computer Associates	2133 \$
Sparx Enterprise Architect programa	Sparx Enterprise Co	70\$

Projektui realizuoti buvo panaudota ši programinė įranga, kadangi šios programinės įrangos licenzijų kaina yra mažiausia arba produktas yra visiškai nemokamas.

2.3.8 Reikalavimai informacijos posistemei

Serverinei daliai:

- Duomenų bazė, sukurta su MySQL‘u

Duomenų surinkimo daliai:

- Java virtuali mašina (JVM), įdiegta į mobilų telefoną, kišeninį kompiuterį.

2.3.9 Sistemos vartotojai

APS vartotojai - tai motorinių transporto priemonių vairuotojai, nenorintys gaišti laiko ieškant kur pasistatyti savo transporto priemonę miesto centre.

2.3.10 Reikalavimai vartotojo sąsajai

Grafinės vartotojo sąsajos reikalavimai (jungiantis per asmeninį kompiuterį):

- Windows 98/ME/2000 Pro/XP operacinė sistema;
- Lengva navigacija;
- “Drag - and - drop” modelių redaktorius;
- Jos veikimas realiuoju laiku;
- Paprastas prisijungimas;
- Turi būti intuityvaus pobūdžio;
-

Grafinės vartotojo sąsajos reikalavimai (jungiantis per mobilųjį telefoną):

- Vartotojo mobilusis telefonas turi palaikyti WAP 2.0 (WML2 / XHTML) standartą;
- Java technologijos palaikymas;
- Paprasta navigacija;
- Grafinės vartotojo sąsajos veikimas realiuoju laiku;
- Dialoginė vartotojo sąsaja;
- Vartotojo sąsajos meniu turi būti lietuvių kalba;
- Mobiliojo ekrano minimali rezoliucija turi būti 57 x 80 taškų;

Grafinės vartotojo sąsajos reikalavimai (jungiantis per delninį kompiuterį (Palm)):

- Palm OS 4.1 operacinė sistema;
- Vartotojo delninis kompiuteris turi palaikyti WAP 2.0 (WML2 / XHTML) standartą;
- Paketinio duomenų perdavimo technologija GPRS;
- Java technologijos palaikymas;
- Paprasta navigacija;
- Grafinės vartotojo sąsajos veikimas realiuoju laiku;
- Dialoginė vartotojo sąsaja;
- Skystų kristalų vaizduoklis LCD: STN pasyvioji matrica, 58,621 spalvų ,integuotas;
- „Touch Screen“ – jautrus lietimui ekranas

2.3.11 Eksploatavimo aplinka

- Aparatūra stovėjimo vietoje turi atitikti esamas gamtines sąlygas;
- Aparatūra turi veikti be sutrikimų (nuo -30°C iki 55°C) temperatūroje;
- Aparatūra turi būti apsaugota nuo vandalų;

2.3.12 Reikalavimai projekto dokumentacijai

Projekto dokumentacija turi atitikti Lietuvių kalbos rašybos ir skyrybos reikalavimus bei taisykles pateiktas V. Limanauskienės " BENDRIEJI NURODYMAI BAIGIAMIESIEMS DARBAMS" vadove.

2.3.13 Sistemos apribojimai

Duomenų bazei:

Maksimalus bazių skaičius Neribojamas

Maksimalus duomenų bazės įrašų skaičius 100 000 .

Maksimalus įrašo ilgis 32 simboliai

Maksimalus lauko ilgis 32 simboliai

Techninė įranga:

Sisteminės įrangos veikimo temperatūros diapazonas: nuo -20°C iki $+40^{\circ}\text{C}$

2.3.14 Reikalavimai eksperimentui

Operacinė sistema: Windows XP

Procesorius: Intel Pentium IV 1500 Mhz

Operatyvinė atmintis: 512 DDRAM

Programinė įranga: JVM – Java Virtual Machine, MySQL duomenų bazė, „Apache“, PHP;

3 SISTEMOS PROJEK TINĖ DALIS

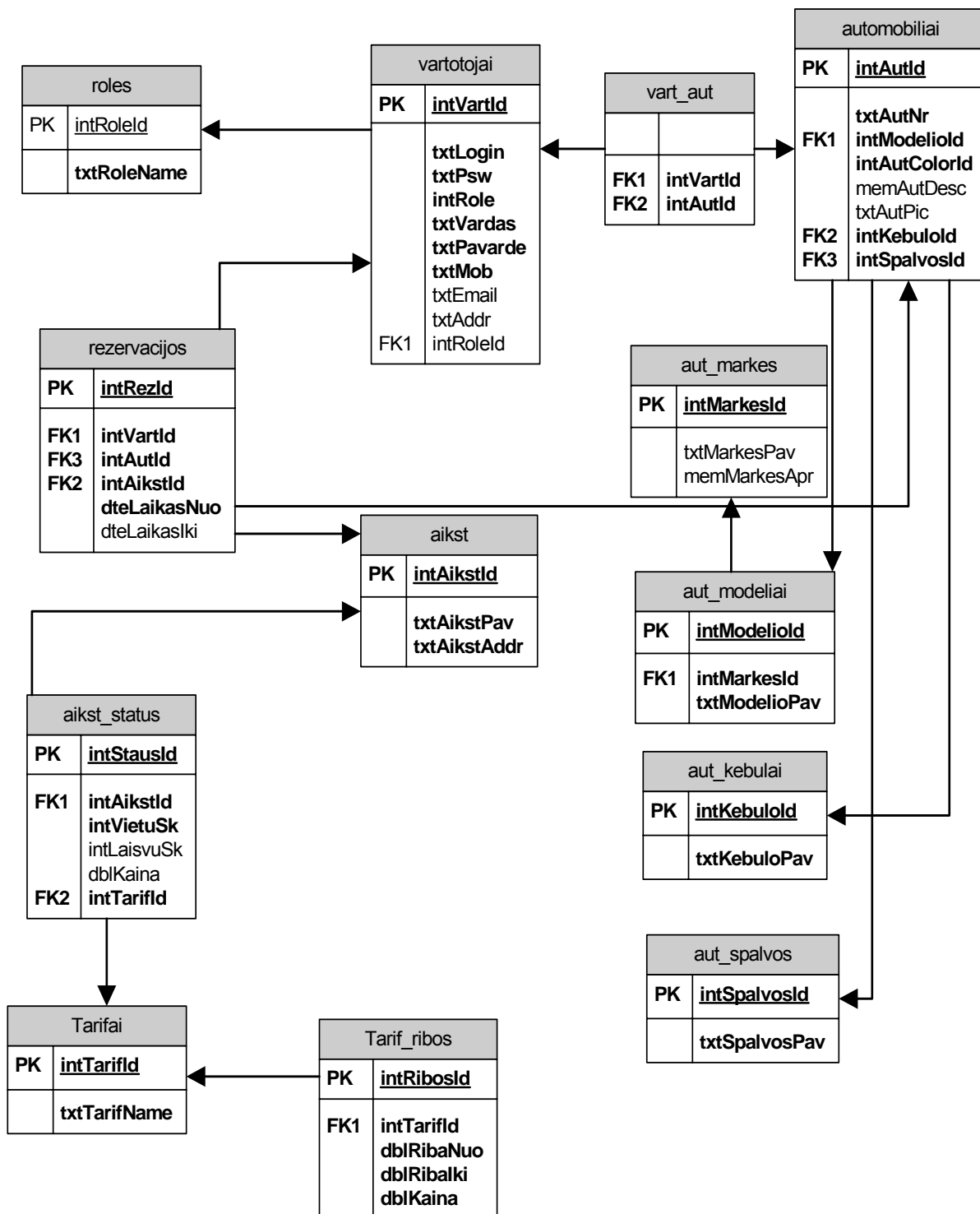
3.1 Sistemos duomenų bazė

Projektui įgyvendinti naudojome MySQL duomenų bazę. Pasirinkimas buvo toks, nes MySQL derinti galima su kitomis informacinėmis posistemėmis. MySQL leidžia jungtis prie šios duomenų bazės nutolus, nors konkrečiai mūsų projektui tai nėra labai aktualu, tačiau tai gali praversti, kai reikės įvesti pataisymus labai skubiai. Pagrindiniai šios duomenų bazės pasirinkimo argumentai:

- Pakankama ir greita
- Darbas paremtas Client/Server architektūra.

Sistemos duomenų bazę sudaro tokios lentelės:

1. vartotojai
2. roles
3. automobiliai
4. vart_aut
5. aut_markes
6. aut_modeliai
7. aut_kebulai
8. aut_spalvos
9. aikst
10. aikst_status
11. tarifai
12. tarif_ribos
13. rezervacijos



3. pav. Sistemos APS duomenų bazė

Vartotojai

3 lentelė. Vartotojai

Laukas	Aprašymas	Tipas
intVartId	Vartotojo identifikacijos numeris	Sveikas skaičius iki 12 skaitmenų
txtLogin	Prisijungimo vardas	Tekstinė eilutė iki 30 simbolių
txtPsw	Prisijungimo slaptažodis	Tekstinė eilutė iki 32 simbolių
intRole	Rolės identifikatorius iš lentos Roles.	Sveikas skaičius iki 3 skaitmenų.
txtVardas	Vartotojo vardas	Tekstinė eilutė iki 30 simbolių
txtPavarde	Vartotojo pavardė	Tekstinė eilutė iki 50 simbolių
txtMob	Vartotojo mobilaus telefono numeris	Tekstinė eilutė iki 20 simbolių
txtEmail	Vartotojo elektroninio pašto adresas	Tekstinė eilutė iki 50 simbolių
txtAddr	Vartotojo adresas	Tekstinė eilutė

Lentelė skirta informacijai apie vartotojus laikymui. Laukai *txtEmail* ir *txtAddr* gali būti neužpildyti.

Roles

4 lentelė. Roles

Laukas	Aprašymas	Tipas
intRoleId	Vartotojų teisių rolės identifikacijos numeris	Sveikas skaičius iki 3 skaitmenų
txtRoleName	Vartotojų teisių rolės pavadinimas	Tekstinė eilutė iki 30 simbolių

Vartotojų teisių sistemoje nusakanti lentelė. *intRoleId* teisių identifikacijos numeris, kuris įrašomas į vartotojų lentelę. Pagal šį identifikatorių kiekvienam vartotojui leidžiama atlikti skirtingus veiksmus, o kai kurie veiksmai draudžiami tam tikroms vartotojų grupėms.

Lentelė Automobiliai

5 lentelė. Automobiliai

Laukas	Aprašymas	Tipas
intAutId	Automobilio identifikacijos numeris	Sveikas skaičius iki 12 skaitmenų
txtAutNr	Automobilio numeris	Tekstinė eilutė iki 15 simbolių
intModelioId	Automobilio modelio identifikatorius iš lentelės aut_modeliai	Sveikas skaičius iki 3 skaitmenų
memAutDesc	Automobilio trumpas aprašymas	Tekstas
txtAutPic	Nuoroda į nuotrauka su automobiliu	Tekstinė eilutė iki 50 simbolių
intKebuloId	Automobilio kėbulo identifikatorius iš lentelės aut_kebulai	Sveikas skaičius iki 3 skaitmenų
intSpalvosId	Automobilio spalvos identifikatorius iš lentelės aut_spalvos	Sveikas skaičius iki 3 skaitmenų

Informaciją apie vartotojų automobilius sauganti lentelė. Laukai *memAutDesc* ir *txtAutPic* gali būti neužpildyti.

Lentelė Vart_aut

6 lentelė. Vart_aut

Laukas	Aprašymas	Tipas
intVartId	Vartotojo identifikacijos numeris	Sveikas skaičius iki 12 skaitmenų
intAutId	Automobilio identifikacijos numeris	Sveikas skaičius iki 12 skaitmenų

Kadangi kiekvienas vartotojas gali turėti kelis automobilius, todėl būtina surišti su kiekvienu vartotoju jo automobilius per tarpinę lentelę. Vartotojui, kurio identifikatorius lentelėje *Vartotojai* yra *intVartId* atitinka automobilis, kurio identifikatorius lentelėje

Automobiliai lygus *intAutId*.

Lentelė aut_markes

7 lentelė. Aut_markes

Laukas	Aprašymas	Tipas
intMarkesId	Automobilio markės identifikatorius	Sveikas skaičius iki 12 skaitmenų
txtMarkesPav	Automobilio markės pavadinimas	Tekstinė eilutė iki 50 simbolių.
memMarkesApr	Automobilio markės aprašymas	Tekstas

Informaciją apie automobilių markės sauganti lentelė. Laukas *memMarkesApr* gali būti neužpildytas.

Lentelė aut_modeliai

8 lentelė. Aut_modeliai

Laukas	Aprašymas	Tipas
intModelioId	Automobilio modelio identifikatorius	Sveikas skaičius iki 12 skaitmenų
intMarkesId	Automobilio markės identifikatorius iš lentelės aut_markes	Sveikas skaičius iki 12 skaitmenų
txtModelioPav	Automobilio modelio pavadinimas	Tekstinė eilutė iki 50 simbolių.

Lentelė aut_kebulai

9 lentelė. Aut_kebulai

Laukas	Aprašymas	Tipas
intKebuloId	Automobilio kebulio identifikatorius	Sveikas skaičius iki 12 skaitmenų
txtKebuloPav	Automobilio kebulio pavadinimas	Tekstinė eilutė iki 50 simbolių.

Lentelė aut_spalvos

10 Lentelė. Aut_spalvos

Laukas	Aprašymas	Tipas
intSpalvosId	Automobilio spalvos identifikatorius	Sveikas skaičius iki 12 skaitmenų
txtSpalvosPav	Automobilio spalvos pavadinimas	Tekstinė eilutė iki 50 simbolių.

Lentelė aikst

11 Lentelė. Aikst

Laukas	Aprašymas	Tipas
intAikstId	Stovėjimo aikštelės identifikatorius	Sveikas skaičius iki 12 skaitmenų
txtAikstPav	Stovėjimo aikštelės pavadinimas	Tekstinė eilutė iki 50 simbolių.
txtAikstAddr	Stovėjimo aikštelės adresas	Tekstinė eilutė iki 50 simbolių.

Lentelė aikst_status

12 lentelė. Aikst_status

Laukas	Aprašymas	Tipas
intStatusId	Aikštelės būsenos identifikatorius	Sveikas skaičius iki 12 skaitmenų
intAikstId	Stovėjimo aikštelės identifikatorius iš lentelės <i>aikst</i>	Sveikas skaičius iki 12 skaitmenų
intVietuSk	Kiek aikštelėje yra vietų	Sveikas skaičius iki 6 skaitmenų
intLaisvuSk	Kiek aikštelėje yra laisvų vietų	Sveikas skaičius iki 6 skaitmenų
intTarifId	Tarifo identifikatorius iš lentelės <i>tarifai</i>	Sveikas skaičius iki 12 skaitmenų

Lentelė tarifai

13 Lentelė. Tarifai

Laukas	Aprašymas	Tipas
intTarifId	Mokėjimo tarifo identifikatorius	Sveikas skaičius iki 12 skaitmenų
txtTarifName	Mokėjimo tarifo pavadinimas	Tekstinė eilutė iki 50 simbolių.

Lentelė tarif_ribos

14 Lentelė. Tarif_ribos

Laukas	Aprašymas	Tipas
intRibosId	Mokėjimo tarifo identifikatorius	Sveikas skaičius iki 12 skaitmenų
intTarifId	Mokėjimo tarifo pavadinimas	Tekstinė eilutė iki 50 simbolių.
dblRibaNuo	Suma, nuo kurios aktyvuojamas tarifas	Sveikas skaičius
dblRibaIki	Suma, iki kurios aktyvuojamas tarifas	Sveikas skaičius
dblKaina	Tarifo dydis	Sveikas skaičius

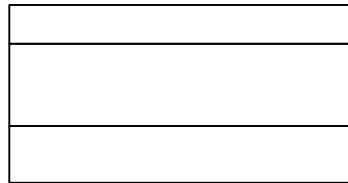
Lentelė rezervacijos

15 Lentelė. Realizacijos

Laukas	Aprašymas	Tipas
intRezId	Rezervacijos identifikatorius	Sveikas skaičius iki 12 skaitmenų
intVartId	Vartotojo identifikatorius iš lentelės <i>vartotojai</i> .	Sveikas skaičius iki 12 skaitmenų
intAutId	Automobilio identifikatorius iš lentelės <i>automobiliai</i>	Sveikas skaičius iki 12 skaitmenų
intAikstId	Aikštelės identifikatorius iš lentelės <i>aiskt</i>	Sveikas skaičius iki 12 skaitmenų
dteLaikasNuo	Rezervuoto laiko pradžia	Iš 14 skaitmenų sudaryta datos ir laiko žymeklis
dteLaikasIki	Rezervuoto laiko pabaiga	Iš 14 skaitmenų sudaryta datos ir laiko žymeklis

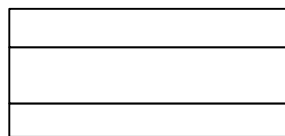
3.2 Klasių diagrama

VAIRUOTOJAS – ši klasė skirta vairuotojo registravimui sistemoje ir stovėjimo vietos rezervavimui. Vairuotojas siunčia trumpąją žinutę su užklausa, kurioje nurodo automobilio numerį, aikštelės numerį ir pageidaujama stovėjimo laiko.



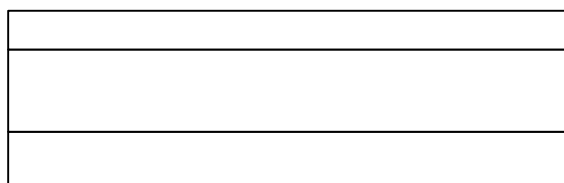
4 pav. Vairuotojų klasė

TIKRINTOJAS – ši klasė skirta vairuotojo registravimo duomenų patikrinimui. Tikrintojas įveda savo duomenys (vardą ir pavardę), prisijungia prie sistemos duomenų bazės ir atlieka rezervavimo duomenų patikrinimą.



5 pav. Tikrintojų klasė

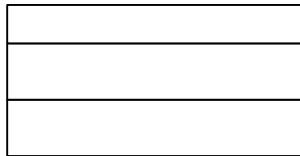
TERMINALINIS SERVERIS – klasė atlieka užsakymų priėmimo ir užsakymų saugojimo veiksmus. Galimi tokie veiksmai kaip: tekstinio pranešimo, automobilio numerio ir stovėjimo laiko išsiuntimas (kaip vairuotojui, taip ir tikrintojui).



6 pav. Terminalinio serverio klasė

Va
-masinos_num
-aiksteles_num
-laikas
+sms_issiuntir
+()

GRAFINIS ATVAIZDAVIMAS – atlieka informacijos atvaizdavimą sistemoje, o taip pat vairuotojams arba tikrintojui užklauius informacija atvaizduojama į jų terminalinę įrangą (gaunama informacija). Atvaizduojama informacija: rezervavimo laikas ir aikštelės numeris.



7 pav. Grafinio atvaizdavimo klasė

DUOMENŲ BAZĖ – duomenų saugojimo komponentė, kurioje saugojami duomenys: vairuotojų rezervuotas laikas, jų automobilių valstybiniai numeriai ir stovėjimų aikštelių numeriai. Taip pat formuojamas duomenų archyvas. Su duomenų baze atliekami tokie veiksmai: duomenų saugojimas, duomenų apdorojimas, duomenų ištrynimasis, duomenų ištraukimas, duomenų atnaujinimas.

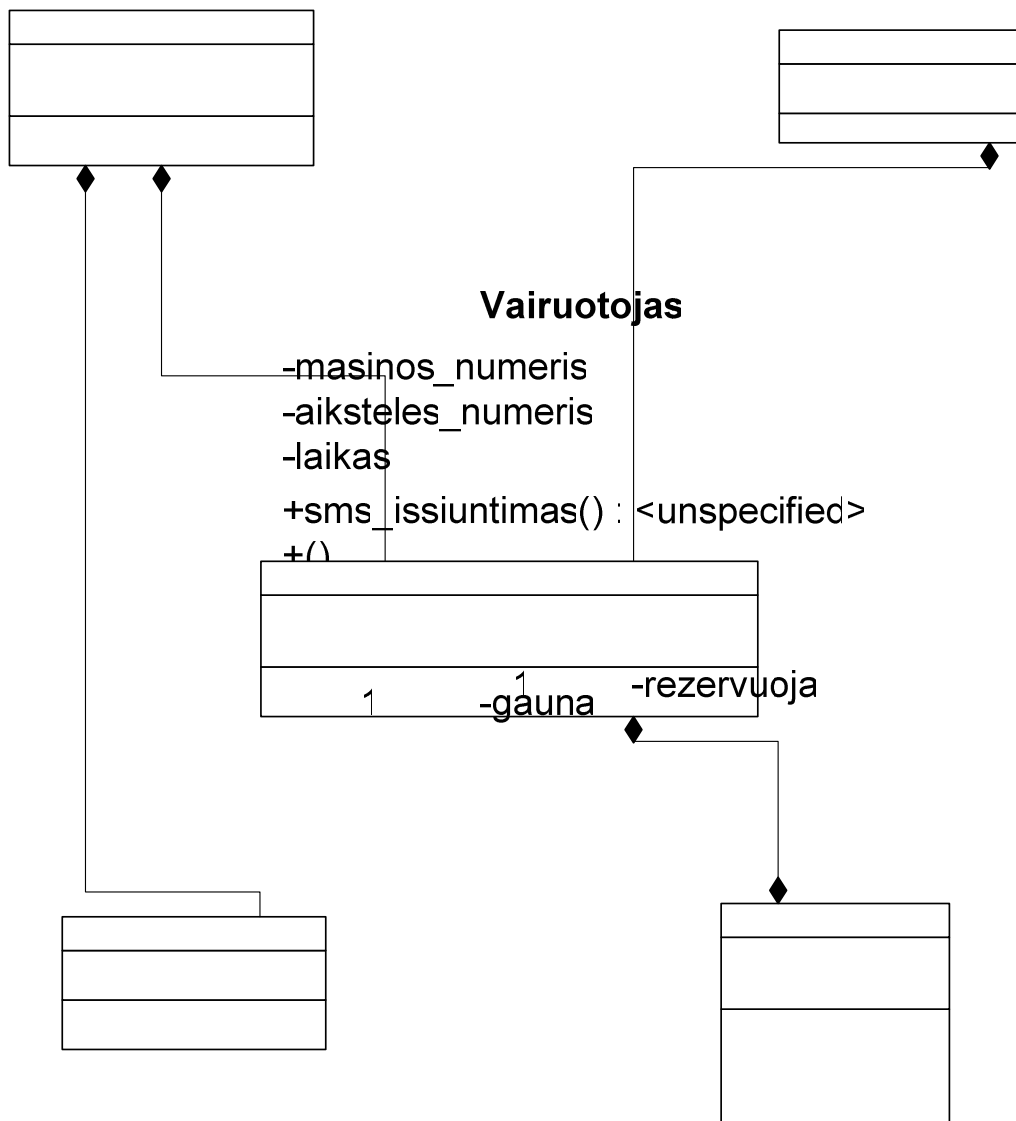


8 pav. Duomenų bazės klasė

Grafinis

- rezervuotas
- aiksteles_n
- +informacijo
- +informacijo

Klasių diagrama



9 pav. klasių diagrama

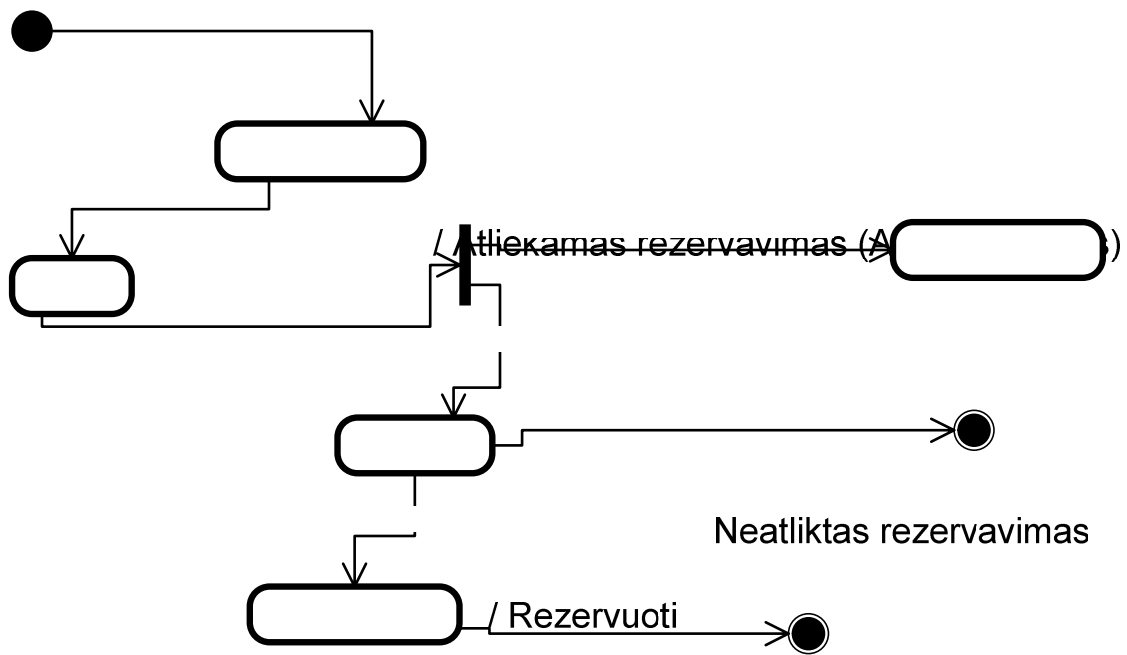
* -rezervavimas a

Terminalinis s

-tekstinis_pranesimas
 -masinos_numeris
 -stovėjimo_laikas
 +uzsakymu_priemimas()
 +uzsakymu_saugojimas()

* -yra gaunamas

3.3 Būsenų diagrama



10 pav. būsenų diagrama

Rezervuota
 Būsenų diagramoj galime matyti vartotojo būsenas. Pradedame nuo užklauso rezervuoti vietą ir baigiame rezervavimo užsakymu. Galutines būsenas atvaizduoja juodi taškai. Tai reiškia kad veiksmas baigtas.

/ Teisingai sur

Leidžiama stovėti

/ Uzsakomas rezervavimas

Uzsakytas rezervavimas

4 TYRIMO DALIS

Norėdami, kad produktas būtų be klaidų, o taip pat būtume užtikrinti, kad nėra defektų, testavimui skyrėme daug laiko.

4.1 Sąsajos testavimas

Registravimo langas

Prie sistemos bus galima prisijungti vartotojams skirtingais variantais, kaip vartotojui norinčiam rezervuoti vietą ir kaip tikrintojui. Mes testosime prisijungimo langą. Taip pat testosime OK ir CANCEL mygtukus. Du prisijungimo variantus.

Rezervavimas:

Rezervuoti vietą iš anksto

Stovėjimas:

Pastatyti automobilį

Tikrintojas:

Galimybė patikrinti rezervavimą ar parakavimą.

.

Pranešimai:

Atvaizduoti rezervavimą:

Kai mes pasirenkam šį variantą, mums bus atvaizduojamas pranešimas lange.

Atvaizduoti kiek laiko liko stovėti vartotojui:

Kai mes pasirenkam šį variantą, mums bus atvaizduojamas pranešimas lange.

Palaikymas:

Tikrinimo sąrašas:

Kai mes pasirenkam šį variantą, mums bus atvaizduojamas pranešimas lange.

Mes galėsime peržiūrėti sąrašą naudojant lango informaciją. Tai tikrintojo funkcija.

Vartotojai:

Kai mes pasirenkam šį variantą, mums bus atvaizduotas vartotojui skirtas meniu .

Pagalba

Turinys:

Standartinis pagalbos meniu bus šiame skyriuje.

Apie:

Pagalbos meniu padės vartotojui suprasti apie rezervavimo ar stovėjimo taisykles, bei paaiškins kaip naudotis sistema.

4.1.1 Testavimo strategija

Šiame skyriuje aptarsime testavimo strategiją. Mes naudojome keturis skirtingus testavimo metodus.

4.1.1.1 Vienetų testavimas

Vienetų testavime mes bandysime skirtingus programinės įrangos modulius. Atliksime baltos dėžės testavimą, kur kiekvienas komponentas testojamas individualiai. Komponentės bus testuojamos praleidžiant duomenis per juos, tuo pačiu metu stebint duomenys ieškant klaidų.

Testas atliekamas programuotojo, kuris suprojektavo ir įgyvendino modulį. Pagrindinis testuotojas įvykdys komponentių testavimą užbaigiant testavimo procesą.

4.1.1.2 Integracinis testavimas

Programinė įranga bus išbandyta vartotojo aplinkoje, naudojant varotojo techniką. Bus ieškoma bet kokių klaidų tarp mūsų programinės įrangos ir varotojo įrenginių.

Programinė įranga bus įdiegta ir paleista naudojimo vietoje. Mes turėsime įsitikinti, kad duomenys apdorojami teisingai, nėra jokių duomenų praradimų ir duomenų bazės netikslumų.

4.1.1.3 Patvirtinamasis testavimas

Šio testavimo metu bus dirbama su žmonėmis, kurie naudosis sistema. Mes norime būti užtikrinti, kad vartotojai gaus tai ko jiems reikia. Bus žiūrima į programinės įrangos specifikacijas, kai reikės spręsti netikslumus susijusius su vartotojo reikalavimais.

Bus atliktas juodosios dėžės testavimas. Naudosime kelėta įėjimo duomenų, pagal kurios bus apskaičiuoti rezultatai. Šie įėjimo duomenys bus įterpti į mūsų programinę įrangą, o gauti rezultatai bus lyginami su apskaičiuotais.

Jeigu atliekant testą atsiranda problemos, bus sukurtas netikslumų sąrašas, kur bus įrašytos visos iškilusios problemos. Atliekant patvirtinamąjį testavimą mes testuosime visus programos komponentus.

4.1.1.4 Aukštesnio lygio testavimas

Bus sujungti keli skirtingi testavimo metodai. Bus bandomos keturios skirtingos sąlygos.

- Atstatomumo testavimas
Čia žiūrima kaip programinė įranga gali atstatyti prarastus duomenys. Mes turime įsitikinti, kad programa nepraranda visų duomenų nenumatytais atvejais.
- Saugumo testas
Ši testavimo dalis skirta įsitikinti, kad visi saugumo patikrinimai veikia ir niekas negali įsibrauti ir pažeisti duomenų.
- Kritinis testavimas
Bandoma užkrauti sistemą iki jos galimybių ribos. Žiūrima kaip sistema veiks esant dideliai apkrovai. Norima įsitikinti, kad sistema neužstrigs, kai visu pajėgumu naudojasi.
- Vykdyto testavimas

Vykdyto reikalavimai yra nustatomi projektuojant sistemą. Šie reikalavimai padeda nustatyti sistemos efektyvumą.

4.1.2 Pasirinktas testavimo metodas

Išanalizavus mūsų pasirinktus 4 testavimo metodus, padarėm išvada kad integracinis testavimo metodas labiausiai tinka mūsų sistemos testavimui. Būtent dėl to ji ir pasirinkome.

Šio testavimo metu sistema buvo įdiegta ir paliesta naudojimo vietoje. Taigi produktas buvo bandomas naudojant vartojimo įrangą. Buvo naudojamas „Iš viršaus į apačią“ testavimo modelis.

Testavimo metu buvo naudojamos funkcijos (Stubs). Šios funkcijos naudojamos atskirtų modulių testavimui. Kiekvienas testuojamas modulis turėjo savo nutolusią funkciją.

Taip pat buvo stebima ar nėra bet kokių nesuderinamumų tarp sistemos ir naudojimo įrangos. Būtina įsitikinti, kad neatsiranda problemų, kai atskiri moduliai dirba vienu metu.

Kiekviena forma buvo testuojama bet kokiais logiškai įmanomai klaidai rasti.

Sistema buvo įdiegta ir paleista vartotojo aplinkoje. Tuo pat metu dirbo ir kitos aplikacijos. Buvo užtikrinta, kad duomenys išsaugoti teisingai, nėra duomenų praradimų ir duomenų bazės problemų.

Testavimo pabaigoje visi rezultatai būti teigiami. Visi sistemos komponentai privalo dirbti teisingai. Jeigu testavimo metu atsiranda klaidos, mes užregistruojame jas klaidų dokumente. Vėliau šios klaidos bus taisomos.

4.2 Microsoft SOAP Toolkit

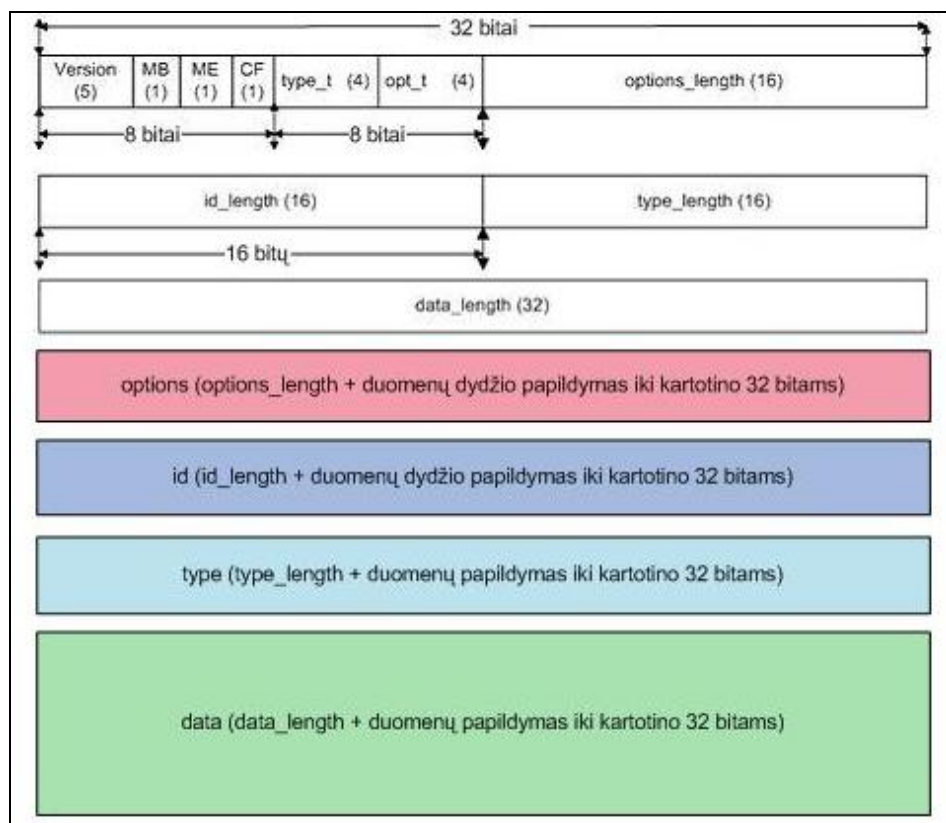
Persikirstymo sistemų, kurios naudoja protokolą SOAP, realizavimą nebūtina pradėti nuo nulio. Egzistuoja daug komponentų, kurie palengvina kūrimo procesą ir perima įvairius užklausų išskvietimo ir rezultato gražinimo etapus. Microsoft SOAP Toolkit – tai įrankių rinkinys, kuris palengvina šiuolaikinių SOAP komponentų realizaciją, kurie užtikrina užklausų priėmimą ir apdorojimą.

Sudėtis:

- SOAP Listener – serveriniai komponentai, skirti įeinamų užklausų apdorojimui. Būna dviejų tipų: ASP puslapis ir ISAPI išlietimas.
- Proxy – klientinis komponentas, leidžiantis reformuoti metodo išskvietimą į SOAP užklausą.
- SOAP užklausų trasavimo įrankis, leidžiantis perimti užklausą į serverį ir analizuoti užklausą/atsakymą iš serverių.
- WSDL-generatorius – pagal komponentų tipų biblioteką generuoja failus, kurie reikalingi SOAP Listener ir Proxy darbui. Šiose failuose yra metodų, parametrų, tipų aprašymai.
- Žemo lygio komponentų rinkinys, skirtas SOAP užklausų kūrimui, įvairių duomenų tipų reformavimui į tekstinį formatą, paketų priėmimo ir perdavimo per HTTP protokolą ir serverio atsakymo skaitymui.
- Dokumentaciją ir pavyzdžiai.

4.2.1 DIME užklausos struktūra

Bet kokia DIME užklausa yra sudaryta iš įrašų visumos. Įrašus sudaro kaip duomenų charakteristikos užklausos antraštėje – tipas, ilgis, identifikatorius, taip ir patys duomenys. Pirmame DIME užklausos įrašė uždedamas įrašo pradžios pažymėjimas „MB“ (Message Begin), o paskutiniame įrašė uždedamas pažymėjimas „ME“ (Message End). Užklausą gali sudaryti tik vienas įrašas. Tokiu atveju abu pažymėjimai bus uždėti viename įrašė. Taip pat galima pridėti vieną DIME užklausą į kitą. Tada pilna antra užklausa bus įdėta į prima.



pav. 11. DIME įrašas.

4.2.2 DIME palaikymas SOAP Toolkite

Prisegtu prie užklauso duomenų perdavimui SOAP Toolkite yra keli COM objektai, kiekvienas iš jų skirtas konkrečiam duomenų tipui.

Pavadinimas	Aprašymas
FileAttachment30	Skirtas bylų perdavimui (IFileAttachment)
StringAttachment30	Skirtas eilutės tipo duomenų perdavimui (IStringAttachment)
ByteArrayAttachment30	Skirtas baitų masyvų perdavimui (IByteArrayAttachment)
StreamAttachment30	Perduoda duomenys į IStream (IStreamAttachment)

Tam, kad automatiškai įjungti DIME palaikymą, užtenka aprašyti objekto, kuris perduoda prisegimą kaip vieną iš aprašytų sąsajų, metodo parametą. Kai WSDL generatorius tipų bibliotekoje sutiks tokį parametą, jis pridės prie WSDL dokumento atitinkamą dalį su tipo aprašymu. Priėmimo pusėje parametras turės jau kitą tipą –

IreceivedAttachment. Šios sąsajos pagalba gautą prisegimą galima išsaugoti byloje arba atlikti tam tikrą apdorojimą.

DIME palaikymą galima naudoti ne tik šių keturių tipų perdavimui. Tam, kad perduoti savo tipo duomenis, reikia naudoti specialius COM objektus, kurie nusako tiesioginį ir atgalinį sudėtingų duomenų reformavimą į XML. Šio atveju sudėtingi duomenis – tai COM objektai ir UDT (User Defined Type) struktūros.

4.3 ATL 7.0

Septintoje ATL klasių bibliotekos versijoje atsirado žymūs pakeitimai. Dabar ją sudaro dvi pagrindinės dalys:

- ATL – biblioteka skirta COM objektų parengimui. Ši bibliotekos dalis palyginus su ATL 3.0 pasikeitė nežinia.
- ATL Server – tai C++ klasių rinkinys, skirtas tinklo aplikacijų, tinklo servisų, ISAPI praplėtimų ir kitų serverinių aplikacijų kurimui. Tai visiškai nauja ATL bibliotekos dalis.

Bibliotekoje ATL Server yra daug klasių, kurios sprendžia tipines serverinių aplikacijų užduotys – keršavimas, šifravimas, MIME ir SMTP palaikymas, HTML generavimas, srautų saugyklų realizacijos, įvairių kodavimų pritaikymas. Be viso to, ATL 7.0 atsirado SOAP protokolo palaikymas. Šį palaikymą sudaro serverių klasių rinkinys, reformuojantis SOAP užklausą į metodų iškvietimą, ir klientinės klasės leidžiančios formuoti užklausas į serverį ir gauti atsakymą.

4.3.1 ATL Server bibliotekos struktūra

ATL Server biblioteką sudaro daug klasių. Šias klases galima suskirstyti į kelias kategorijas:

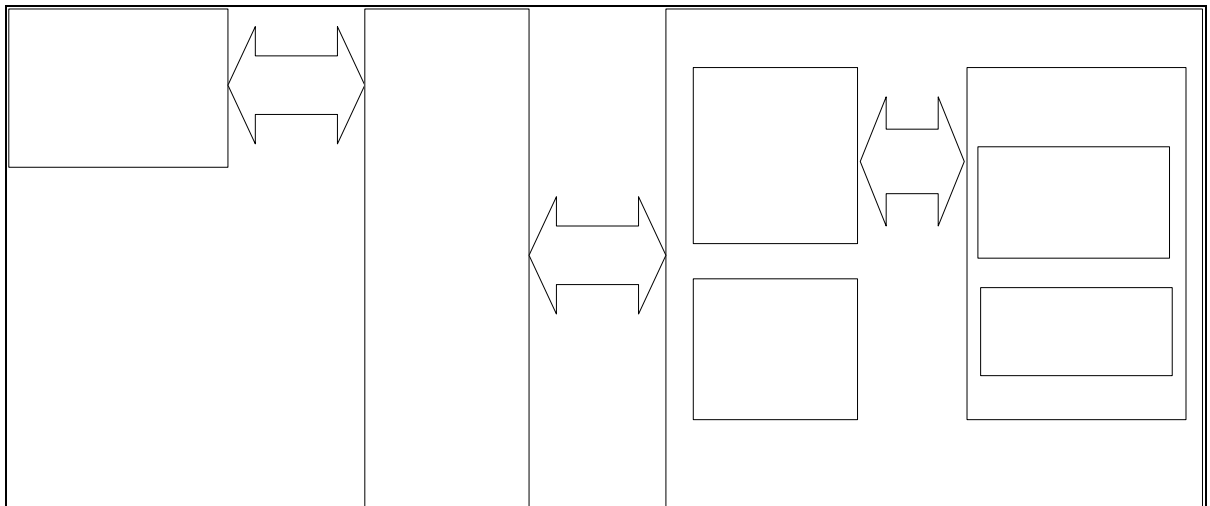
Kategorija	Aprašymas
ATL Server Framework	Pagrindinė ATL Server infrastruktūra
Užklausų apdorojimas	Klasės ir atributai skirti HTTP užklausų apdorojimui
Užklausų statistikos apdorojimas	Klasių rinkinys skirtas užklausų apdorojimo statistikos surinkimui
Išplėtimai išoriniam tinklo aplikacijų valdymui	Klasės leidžiančios valdyti tinklo aplikaciją iš išorės per SOAP protokolą
Shared Services	Klasių rinkinys įgalinantis servisų kūrimą, kurie būtų prienami tinklo aplikacijoms.
Session State	Klasių rinkinys leidžiantis išsaugoti prisijungimo sesijos duomenys atmintyje arba duomenų bazėje.
SOAP	Serverio ir kliento klasės SOAP protokolo palaikymui.
Darbas su kešu	Klasės ir sąsajos kešo išvalymui, kuris remiasi prieinamumo prie elementų laiku.
Šifravimas	Klasių rinkinys darbui su kriptavimo servisu, raktais, paplitusių algoritmų SHA, MD5 palaikymas.
Kuoduočių palaikymas	Klasės skirtos duomenų formavimui į paplitusias kuoduotės (base64, uuencode, utf8)
HTML generacija	HTML generavimo klasės.
HTML klientas	Klasių rinkinys HTTP kliento daliai realizavimui, leidžia formuoti užklausas ir gauti serverio atsakymus nenaudojant papildomų priemonių.
MIME ir SMTP	Klasės skirtos darbui su SMTP Protokolu.

Šio metu ATL biblioteka palaiko dauguma protokolu, kurie naudojami Internetė. Pagrindinė ATL klasių savybe yra tai, kad naudojant jas sumažėja reikalavimai resursams, padidėja sparta ir sumažėja priklausomybė nuo išoriniu modulių ir komponentų.

4.3.2 ATL Server aplikacijų architektūra.

Bet kokios ATL Server aplikacijos architektūra sudaroma iš keturių pagrindinių elementų:

- Internet Informatikon Servines (IIS) – tinklo serveris, priimančias klientų užklausas ir perduodantis juos į ISAPI išplėtimą.
- ISAPI išplėtimas priima užklausas ir perduoda juos apdorojimui atitinkamam moduliui (Web Application Dll), suteikia bendrus servisus tinklo aplikacijoms (binarinių duomenų saugyklą atmintyje). ISAPI išplėtimas apdoroja užklausas į failus su išplėtimais .dll ir .srf.
- Web Application Dll – tinklo aplikacijos modulis, kuris realizuoja tinklo aplikacijos logiką ir suteikia užklausų apdorojimo elementą (request handler) – modulio įėjimo taškai, kurių pagalba vyksta darbas su ISAPI išplėtimu. Kiekvienas modulis turi vieną arba kelis apdorojimo elementus, su kuriais sąveikauja ISAPI išplėtimas.
- Server Response Files (srf) – tai tekstiniai šablonai, kuriuose yra statistinė serverio atsakymo dalis.



Pav. 12 XML Tinklo serviso architektūra.

Kliento užklausa patenka į ISAPI išplėtimą, kuris išskiria iš užklausa reikiamo modulio vardą ir perduoda jam valdymą. Tinklo aplikacijos modulyje iš SOAP užklausa išskiriami parametrai, kurie iš tekstinio pavidalo paverčiami į binarinį. Toliau veiklą pradeda tinklo aplikacijos vartotojo klasė, kuri atlieka norimus veiksmus ir grąžina rezultatą, kuris

transformuojamas į tekstinį pavidalą ir formuojamas serverio XML atsakymas. Šis atsakymas siunčiamas klientui per ISAPI išplėtimą ir tinklo serverį.

4.3.3 DIME ATL Server aplikacijose

Kuriant XML tinklo servisu, naudojant biblioteką ATL 7.0 ir jos dalies žinomos kaip ATL Server, DIME palaikymo nėra. Tačiau, pasinaudojus DIME aprašymu, DIME užklauso analizės įrankiais, ATL bibliotekos kodo išėities tekstu ir galimybe sukurti savo norimas klases taip praplečiant šios klasės funkcionalumą galima įdiegti DIME palaikymą į ATL Server biblioteką. Tam reikia pakeisti WSDL failo generavimo mechanizmą taip, kad būtų įdiegti atitinkami elementai operacijų aprašymui ir IStream duomenų tipo aprašymui, kurie reikalingi perduodant DIME užklausa. Taip pat reikia praplėsti ATL klases naujomis, kurios dirbtų su DIME įrašais ir apdorotų duomenys.

5 EKSPERIMENTINĖ DALIS

5.2 Sistemos naudojimosi eksperimentas

5.2.1 Prisijungimas

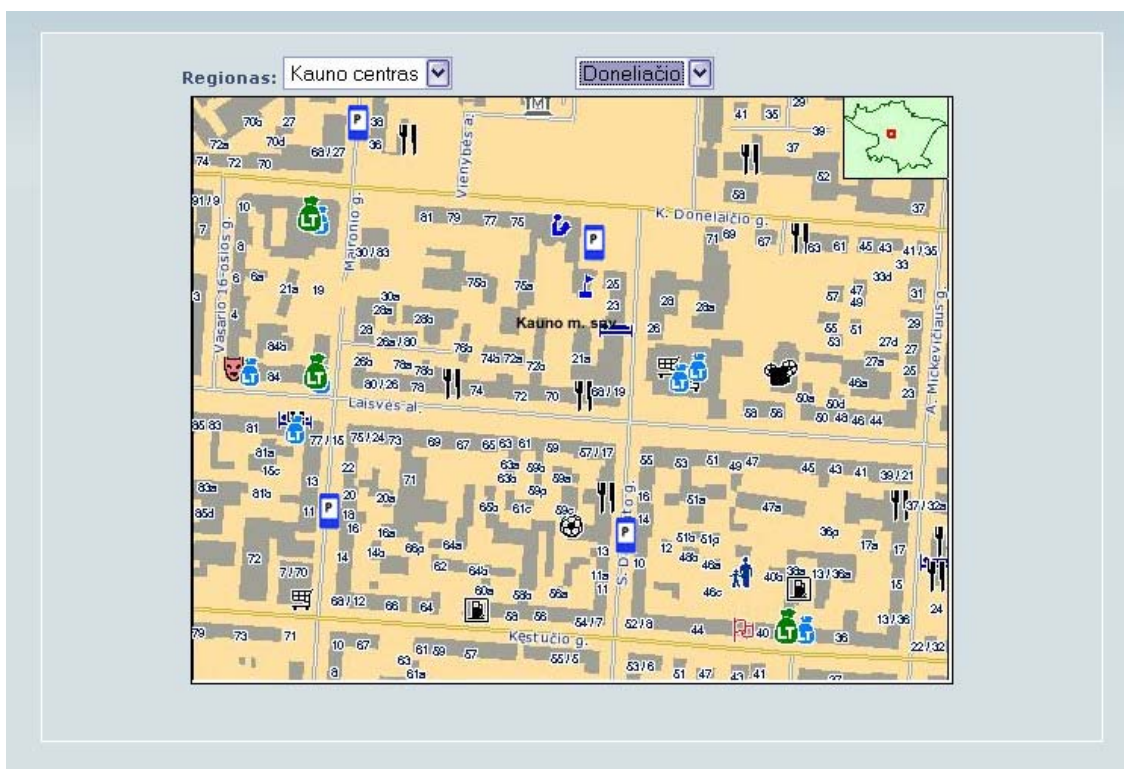
Vartotojas pasirinktu būdu (per WAP arba internetą) siunčia užklausa į sistemą ar aikštelėse yra laisvų stovėjimo vietų, prieš tai identifikuojant aikštelę (pvz. Gedimino gatvė –2). Ši informacija yra nuolat atnaujinama. Svarbu tai, jog norint gauti detalų nuvykimo planą iki stovėjimo vietos reikia prieš siunčiant užklausa tai nurodyti (dar nenuorta koks bus išskirtinis ženklas).

Sistema informuoja klientą apie laisvas vietas (vietos identifikuojamos) ir atsiunčiamas į ekraną nuvykimo planas jei buvo pageidauta.

Vartotojas pasirinktu būdu (per Wap arba internetą) siunčia užklausa skirtą vietos rezervavimui nurodant vietos identifikatorių, laiką (iki kada rezervuojama konkreti vieta) į sistemą.

5.2.2 Rezervavimas

Rezervuojant stovėjimo vietą internetu, pirmiausia žemėlapyje pasirenkama aikštelė (Pav. 10). Jeigu aikštelė yra kitame regione reikia pakeisti ją ir atsiradusiame regione žemėlapyje pasirinkti norimą aikštelę.



13 pav. aikštelių planas

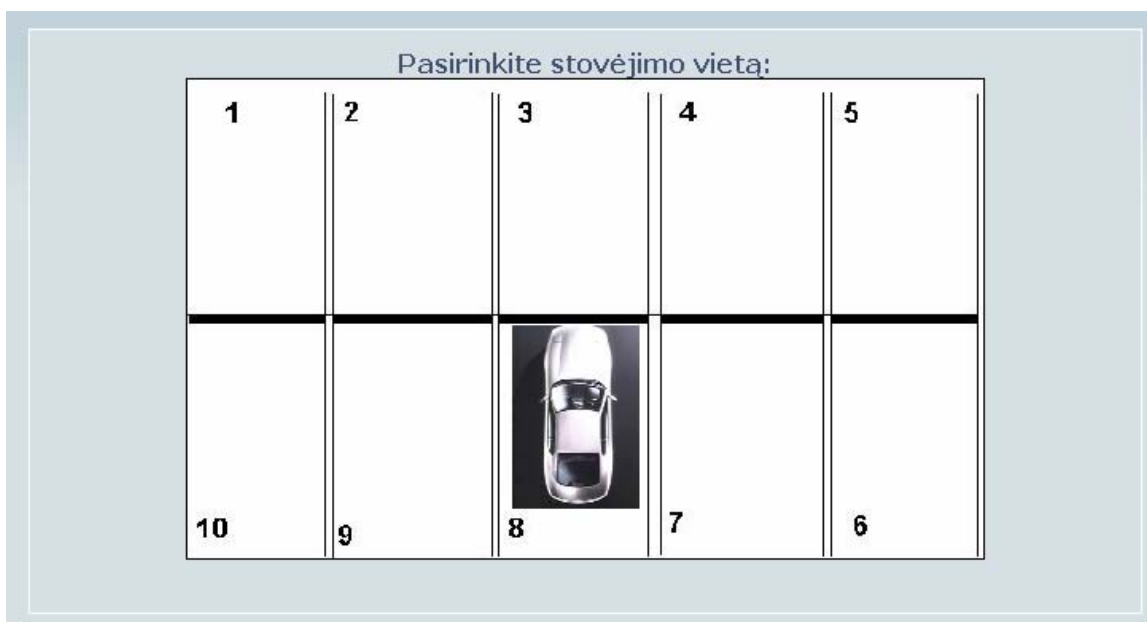
5.2.3 Informacijos įvedimas

Toliau įvedami stovėjimo laiko pradžia, pabaiga ir automobilio numeris:

Regionas:	Kauno centras	▼
Aikštelė :	Maironio-1	▼
Automobilio nr. :	Irs444	▼
Laikas nuo :	2005.05.05 13:00	
Laikas iki :	2005.05.05 14:00	
Patvirtinti		

14 pav. duomenų įvedimas

Stovėjimo aikštelės schemoje pasirenkama laisva vieta. Visos pasirinktu laikotarpiu užimtos vietos pažymėtos stovinčiu automobiliu.



15 pav. stovėjimo vietos pasirinkimas

Pasirinkus stovėjimo vietą gaunama SMS žinutė su patvirtinimu.

5.2.4 Prisijungimas naudojant WAP technologiją

Prisijungus prie WAP puslapio naudojant mobilaus telefono WAP naršyklę, reikia pasirinkti rezervacijos punktą ir atsiradusiame lange įvesti aikštelę, stovėjimo laiką ir automobilio valstybinį numerį:



16 pav. WAP puslapis

Gavęs patvirtinimą apie norimą stovėjimo vietą, serveris perduoda duomenys apie užrezervavusį stovėjimo vietą asmenį į stovėjimo aikštelėje esančią sistemą.

5.2.5 Automobilio statymo procesas

Atvyksta klientas iš anksto rezervavęs vietą, ir vietos sensorius (Zigbee) nuskenuoja kliento automobilio numerį ir perduoda jį į stovėjimo aikštelės sistemą. Ši sistema patikrina ar atvažiavęs klientas rezervavo stovėjimo vietą. Jeigu šis klientas iš anksto rezervavo stovėjimo vietą, tai sistema siunčia sensoriui patvirtinimą ir sensorius uždega žalią šviesą, leisdamas klientui statyti automobilį duotoje vietoje. Jeigu klientas iš anksto neužsisakė stovėjimo vietos, tai stovėjimo aikštelės sistema gavus iš sensoriaus informaciją apie atvykusį klientą, patikrina ar ši vieta nėra jau rezervuota. Jeigu vieta laisva, tai sistema siunčia signalą sensoriui uždegti žalią šviesą ir pradeda skaičiuoti stovėjimo laiką. Jeigu vieta bus užimta 30 minučių bėgyje, tai uždegama raudona šviesa ir klientui neleidžiama statyti automobilio toje vietoje.

„Zigbee“ naudoja IEEE 802.15.4 specifikacijos standartą. Duomenys perduoda 2,4 GHz dažnyje nuo vieno įrenginio prie kito, kol nepasiekia adresato. Atstumas tarp dviejų įrenginių negali viršyti 70-100 metrų. Didžiąją savo darbo dalį įrenginys būna budėjimo režime. Galima nustatyti laiką, kada įrenginys trumpam įsijungia ir patikrina ar nėra kreipinių į jį. Tai ženkliai sumažina energijos naudojimą.

5.2.6 Vartotojo išvykimas

Stovėjimo vietą palieka vartotojo automobilis. Sensorius reaguoja ir praneša į serverį, apie laisvą vietą. Sistemos serverio skaitliukas atnaujina informaciją apie laisvas vietas duomenų bazėje. Duodamas signalas vietos sensoriui, uždegamas žalias indikatorius, t.y. vieta laisva.

Apmokestinimo sistema pagal prastovėtą kliento laiką ir nustatytus tarifus nustatyto užmokesčio dydį ir jį nuskaito nuo vartotojo abonentinės sąskaitos, kurią jis turi šias paslaugas teikiančioje firmoje. Visi pinigai sumokėti kliento už mūsų stovėjimo ir rezervavimo paslaugą bus įtraukiami į aktošiaus asmeninę telefono sąskaitą. Jis mokės už tai kaip už ryšio paslaugas. Gale mėnesio jam bus įskaičiuojama į jo sąskaitą visos jo stovėjimo ir rezervavimo operacijos.

5.3 Našumo palyginimas perduodant binarinius duomenys skirtingais metodais.

Šiame darbe atliktas našumo palyginimas perduodant binarinius duomenys, kai klientas, siunčiantis arba gaunantis duomenys, ir serveris, apdorojantis kliento užklausas, yra skirtingai konfigūruojami. Pateikiamos tris serverio ir kliento konfigūracijos:

1. Serveris ir klientas ATL 7.0 be DIME palaikymo (t.y. sukurti standartinėm ATL klasėm)
2. Serveris ir klientas sukurti naudojant SOAP Toolkit 3.0 su DIME palaikymu.
3. Serveris ir klientas ATL 7.0 su papildomom klasėm, realizuojančiom DIME palaikymą.

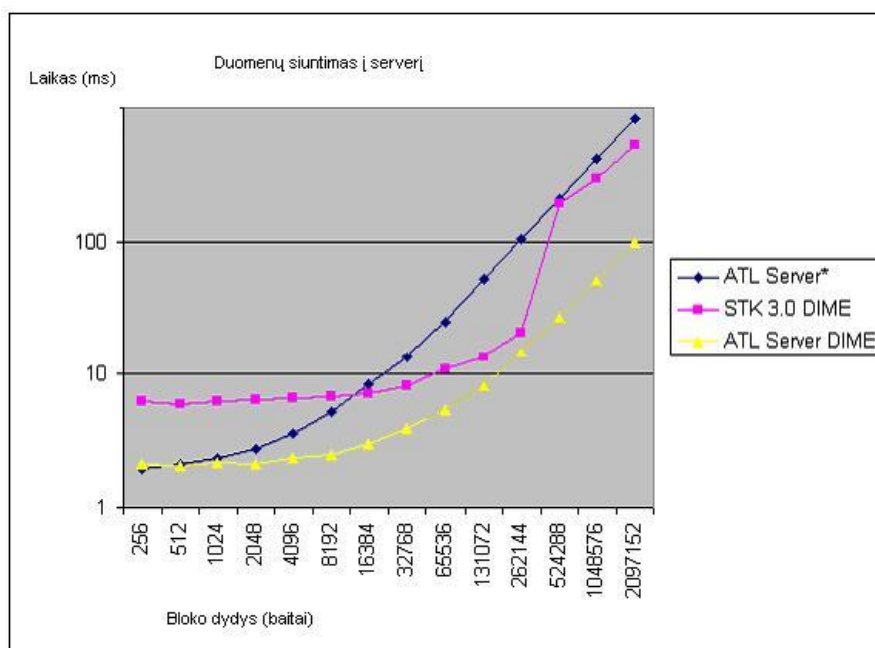
Serverio komponentas SOAP Toolkit, taip pat, kaip ir serverio komponentas ATL Server, perduoda ir gauna iš kliento atitinkamo dydžio duomenų bloką. Taip pat kliento dalis matuoja duomenų išsiuntimo/gavimo iš serverio laiką.

Klientas ir serveris pirmosios konfigūracijos yra identiškias klientui ir serveriui trečiosios išskyrus tai, kad trečioji konfigūracija naudoja papildomas klases skirtos darbui su DIME.

ISAPI išplėtimas SOAP Toolkit 3.0 pagal nutylėjimą neleidžia klientui perduoti paketus didesnius negu 100 kb. Tai padaryta tam, kad nebūtų įmanoma piktavališkai paveikti serverio darbą, perduodant didelius duomenų paketus. Mūsų bandyme šis apribojimas trukdytų perduoti didelius paketus serveriui, todėl būtina pakeisti maksimalų duomenų perdavimo serveriui kiekį serverio registre.

17. Lentelė. Duomenų siuntimo į serverį rezultatai

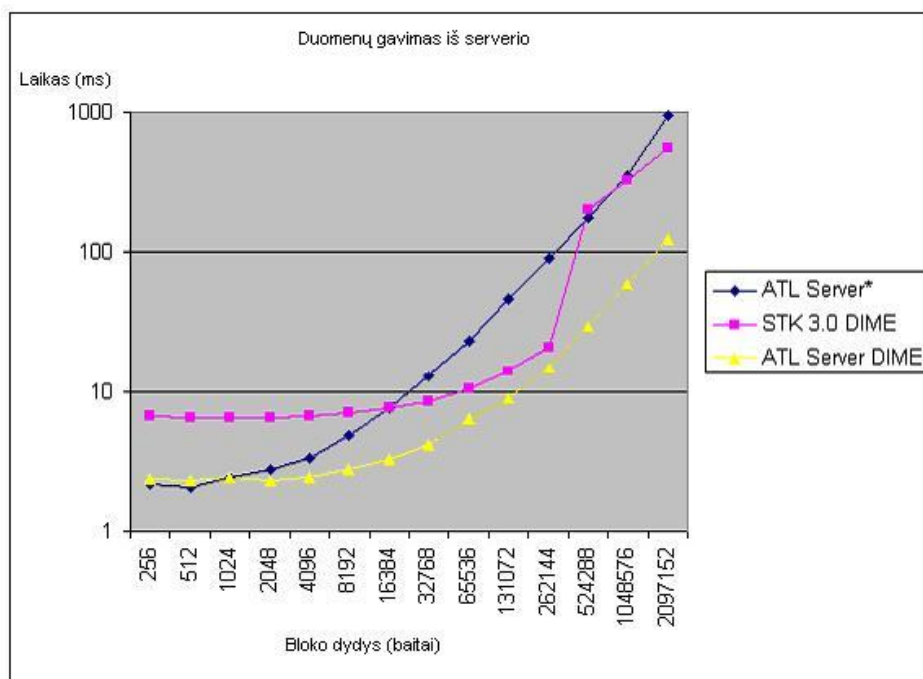
Bloko dydis	1 konfigūracija	2 konfigūracija	3 konfigūracija
256	2,185	6,57	2,345
512	2,03	6,4	2,265
1024	2,425	6,41	2,42
2048	2,735	6,41	2,265
4096	3,355	6,71	2,425
8192	4,845	7,04	2,73
16384	7,58	7,5	3,205
32768	12,81	8,43	4,14
65536	23,36	10,47	6,25
131072	46,485	14,06	8,905
262144	91,405	21,1	14,845
524288	177,89	204,22	29,925
1048576	351,8	325,15	59,765
2097152	958,985	560	124,375



17.pav Duomenų siuntimas į serverį

18.Llentelė. Duomenų gavimo iš serverio rezultatai

Bloko dydis	1 konfigūracija	2 konfigūracija	3 konfigūracija
256	1,95	6,25	2,11
512	2,11	5,94	2,03
1024	2,345	6,25	2,185
2048	2,735	6,41	2,11
4096	3,595	6,56	2,345
8192	5,155	6,72	2,5
16384	8,36	7,19	2,97
32768	13,59	8,12	3,905
65536	24,77	10,94	5,39
131072	51,715	13,6	8,205
262144	104,14	20,63	14,53
524288	211,095	190,78	26,955
1048576	411,565	291,41	50,935
2097152	832,03	527,97	98,905



18. pav. Duomenų gavimo iš serverio rezultatai

SOAP Toolkit 3.0 lėčiausiai apdoroja mažus pakėtus (dėl wsdl ir wsml failų išskirimo būtinumo). Pradedant nuo 16 kb. šiuo įrankių sukurtas servisas žymiai lenkia pirmos konfigūracijos ATL Server servisa, bet dirbant su paketais nuo 512 kb. apdorojimo laikas žymiai padidėja. Tai atsitinka dėl to, kad STK 3.0 nebando didelių blokų įrašyti į atmintį, o automatiškai išsaugoja jos laikinoje laikmenoje %TEMP%. Diskinės operacijos pareikalauja papildomo laiko užklausų apdorojimui. Vis tik STK 3.0 lenkia paprastą ATL realizaciją, o didėjant duomenų srautui šis skirtumas tik didėja.

Pažymėtina, kad paprasta ATL realizacija (konfigūracija 1) paketo, kurio dydis 2 MB apdorojimui, reikalauja apie 1 sekundę.

ATL Server realizacija su DIME palaikymu pirmauja prieš konkurentus visose paketų dydžiuose. Didelis skirtumas tarp STK 3.0 ir ATL Server DIME yra tas, kad ATL realizacija visus duomenys perduoda per atmintį ir nenaudoja laikinų laikmenų didelių paketų saugojimui. O tai gali negatyviai atsiliepti perduodant labai didelius paketus. (Pvz. 10 - 100 MB). Tačiau, didelių paketų perdavimas dažniausiai parodo projektavimo trūkumus. Žymiai efektyviau skaidyti pakėtus į mažesnius ir siusti jos atskiromis porcijomis. Tai leidžia valdyti pakėto siuntimą ir suteikti papildomų funkcijų vartotojui (Pvz. nutraukti perdavimą ir vėliau jį pratęsti).

6 IŠVADOS

Kas padaryta šiame darbe?

Šiame darbe suprojektuota mobili transporto priemonių parkavimo valdymo sistema. Ši sistema skirta supaprastinti automobilių parkavimo apmokestinimą, sutaupyti klientų brangų laiką ir pinigus, sumažinti kamščius ir taršą miestų centruose. buvo

Darbo eigoje buvo išanalizuoti 3 projektavimo metodai ir išrinktas tinkamiausias mūsų valdymo sistemai projektuoti RUP modelis.

Taip pat buvo išanalizuoti 3 duomenų perdavimo ir gavimo iš kliento į serverį metodai, kurie buvo tarpusavyje palyginti. Plačiau buvo analizuojamas metodas sukurtas su TOOLKIT įrankio pagalba.

Antras sistemos kūrėjas, Danielius Comik analizavo metodą, sukurtą su Toolkit įrankio pagalba.

Išanalizuoti 4 testavimo metodai ir išrinktas vienas tinkamiausias mūsų valdymo sistemai testuoti. Sėkmingai atliktas automobilių stovėjimo aikštelių daugiafunkcinės valdymo sistemos testavimas, kuriuo metų buvo testuojama pagal išrinktą testavimo metodą.

Kaip tai padaryta?

Pažintis su problemine sritimi

Projekto metu išsinagrinėtos panašios sistemos, jų veikimo principai ir architektūra. Palygintos jų teikiamos paslaugos ir funkcijos su šio projekto siūloma automobilių stovėjimo valdymo daugiafunkcine sistema. Atlikta šiuo metu egzistuojančių sprendimų ir metodų apžvalga. Išanalizavus situaciją galime teigti, jog poreikis automobilių valdymo sistemoms tikrai yra ir technologijos sparčiai vystomos šitoje srityje.

Projektui realizuoti pasirinkome PHP programavimo kalbą. Šiam pasirinkimui įtakos turėjo tai, jog vartotojai pageidavo, kad galima būtų naudotis tiek Windows, tiek Linux platformose. Projekte naudojama MySQL duomenų bazė. Pasirinkimas buvo toks, nes MySQL derinti galima su kitomis informacinėmis posistemėmis. MySQL leidžia jungtis prie šios duomenų bazės nutolus, nors konkrečiai mūsų projektui tai nėra labai aktualu, tačiau tai gali praversti, kai reikės įvesti pataisymus labai skubiai. Pagrindiniai šios

duomenų bazės pasirinkimo argumentai:

- Pakankama ir greita
- Darbas paremtas Client/Server architektūra.
- MySQL yra nemokama

Išnagrinėti projektavimo metodai, jų sprendimo būdai bei principai. Palyginta tarpusavyje ir išrinkta kas yra geriau, t.y. kas buvo panaudota mūsų valdymo sistemos projektavime.

Realizuoti ir išanalizuoti duomenų perdavimo ir gavimo iš kliento į serverį metodai, kurie buvo palyginti tarpusavyje.

Realizuotas ir išanalizuotas testavimo metodas, kurio pagalba buvo testuojama sistema.

Darbe buvo pritaikytas ir panaudotas Kauno miesto automobilių stovėjimo aikštelių planas bei Kauno miesto žemėlapis. Jo pagalba sudarytas mūsų valdymo sistemos aikštelių planas.

Išsiaiškinome kaip sudarytas DIME formatas ir kaip formuojami pranešimai šitame formate, o taip pat realizavome DIME palaikymą ATL 7.0 klasėse. Ši realizacija nepretenduoja pramoniniam naudojimui, tačiau šios realizacijos pagalba mums pavyko daugiau sužinoti apie formatą DIME, vidinę ATL server sandarą.

Serverio našumas ir binarinių duomenų perdavimo tarp kliento ir serverio greitis priklauso nuo daugelio faktorių. Bet visiškai tiksliai galima pasakyti, kad DIME formato naudojimas leidžia sumažinti tinklo srautą, išvengiant duomenų kodavimo į base64 ir sutaupyti serverio resursus.

7 LITERATŪRA

7.1 Knygos

1. Patrick Naughton, Herbert Schildt; Java™ 2: The Complete Reference, Third Edition;- „BHV – Sankt Peterburgh“ – 2000 m. – 1050 p.
2. M. Holl; Servletai ir JavaServer Pages; „SPB – Sankt Peterburgh“ – 2001 m. -496 p
3. Butkienė R., Butleris R. The Approach for User Requirements Specification // 5th East-European conference ADBIS'2001, Research Communications, Ed. by A Čaplinskas, J.Eder, Vilnius, 2001, p. 225-240.

7.2 Prieigos per internetą

1. UAB „Bitė“, telekomunikacijų bendrovė. URL: <http://www.bite.lt/> (žr. 2003 12 11).
2. VeriSign, Inc., interneto ir telekomunikacijų tinklų saugumo, patikimumo technologijos. URL: <http://www.verisign.com/> (žr. 2003 12 11)
3. Ascom, Inc., e-parkingo sistema. URL: <http://www.ascom.com/> (žr. 2003 04 01)
4. Ericsson Enetrprise, e-parkingas. URL: <http://www.ericsson.com/uk/eparking/> (žr. 2003 05 11)
5. Ericsson Enterprise, stovėjimas naudojantis wap, sms. URL: http://www.ericsson.com/mobilityworld/sub/articles/success_stories/parking_wap_sms (žr. 2003 05 11)
6. Sun Microsystems, Inc., Java technologijos. URL: <http://java.sun.com/downloads> (žr. 2003 04 01)
7. EMT (Estonian Mobile Telephone), telekomunikacijų bendrovės m-parkingo projektas. URL: <http://www.m-parking.emt.ee> (žr. 2003 04 01)
8. Airclit, Inc., parkingo sistema. URL: <http://www.airclit.com> (žr. 2003 04 01)
9. Open Mobile Alliance Ltd., informacija apie WAP. URL: <http://www.wapforum.com> (žr. 2003 04 01)
10. Rational Software Corporation, komercinis modeliavimo produktas RationalRose. URL: <http://www.rational.com/products/rup> (žr. 2003 10 11).

8 TERMINŲ IR SANTRUMPŲ ŽODYNAS

SMS - trumpųjų žinučių paslauga (Short Message Service)

WAP – specialus duomenų perdavimo protokolas bevielams įrenginiams (Wireless Application Protocol)

LAN – vietinis tinklas (local area network)

GPRS – paketinio duomenų perdavimo technologija (General Packet Radio Service)

GSM - tai sparčiai plečiama mobiliojo korinio ryšio sistema pasaulyje. (Global System for Mobile communications)

WML – tai programavimo XML kalbos atmaina (Wireless Markup Language).

XML - išplečiama žymių kalba (Extensible Markup Language)

BlueTooth - tai nauja bevielė technologija, kuri ateityje turėtų pakeisti infraraudonųjų spindulių jungtį.

JSP – programavimo kalba (Java Server Pages)

JVM – Java virtual machine

ASP- Active Server Pages

UML – modeliavimo kalba (unified modeling language)

PHP- programavimo kalba (hypertext preprocessor)

WEB – tinklas

IIS - galingas WEB serveris (internet information server)

HTML- programavimo kalba (HyperText Markup)

MySQL – duomenų bazėms kurti programavimo kalba

GUI – grafinė sąsaja (graphical user interface)

PIM - Personal Information manager

VPN - (Virtual Private Network) technologija

PDA – personal digital assistant

TCP/IP – transmission control protocol / internet protocol.

SSL – kodavimo technologija (secure sockets layer)

HTTP – internetinis protokolas (Hyper Text Transfer Protocol)

KTU – Kauno Technologijos Universitetas

SOAP – protokolas skirtas informacijos apsikeitimui tarp sistemos komponentų (simple object access protocol)

WSDL – tinklo serviso aprašymo kalba (web service description language)

UDDI –

DCOM – komponentų modelis (distributed component object model)

DIME – objektų prisegimo prie vieno pranešimo specifikacija (direct internet message encapsulation)

MIME – objektų prisegimo prie vieno pranešimo specifikacija (internet message encapsulation)

ATL – klasių rinkinys

9 PRIEDAI

9.1 SOAP realizacijos:

19 Lentelė. Atviro kodo/nemokamos realizacijos

Produktas	Kompanija/Organizacija/Autorius	Aprašymas
Apache SOAP	Apache Software Foundation	SOAP 1.2 realizacija (Java)
Apache Axis	Apache Software Foundation	SOAP 1.2 realizacija (Java,C++)
Wingfoot SOAP	Wingfoot Software	J2ME realizacija SOAP(Java)
kSOAP	ObjectWeb Open Source Consortium	J2ME realizacija SOAP(Java)
PocketSOAP	Simon Fell	SOAP realizacija kaip COM objektai (C++)
SOAP::Lite	Pavel Kulchenko	SOAP realizacija (Perl)
PHP SOAP	Shane Caraveo, Arnaud Limbourg	SOAP realizacija (PHP)
gSOAP	Robert A. van Engelen	SOAP realizacija (C++)

20 Lentelė. Komercinės realizacijos

Produktas	Kompanija/Organizacija/Autorius	Aprašymas
IBM Websphere	IBM	Tinklo serviso realizacija Websphere produkte.
Web Services Developer Pack (WSDP)	Sun Microsystems	JAX* API, saugumo realizacijos, WSDP Registry Server, kiti įrankiai
WASP Server	Systinet	Tinklo serviso realizacija (Java,C++)
Microsoft .NET Framework	Microsoft	Tinklo serviso realizacija
Oracle Application Server	Oracle	Tinklo serviso realizacija
BEA Weblogic	BEA	Tinklo serviso realizacija (Weblogic 8.x)
webMethods	webMethods	Tinklo serviso realizacija

GLUE		(Java)
Artix Enterprise Web Services	IONA	Tinklo serviso integracijos produktas

9.2 DIME užklauso antraštė

21 Lentelė. DIME užklauso antraštė

Laukas	Aprašymas
VERSION	Užduodama DIME formato versija. Visi įrašai vienoje DIME užklausoje turi būti vienodos versijos. Jeigu DIME skaitiklis nepalaiko įrašė nurodytos versijos, tai jis turi atmesti visą užklausą.
MB	Pirmo įrašo pažymėjimas (1 bitas). Pirmas užklauso įrašas privalo būti pažymėtas.
ME	Paskutinio užklauso įrašo pažymėjimas (1 bitas). Paskutinis įrašas privalo būti pažymėtas.
CF	Pažymėtas parodo, kad perduodamas duomenų fragmentas. Duomenys pilnai aprašomi įrašų seka su nustatytu bitu CF, paskutinis įrašas sekoje prilygina CF bitą nuliui. Pilnas duomenų bloko ilgis nustatomas kaip suma kiekvieno sekos įrašo fragmento.
TYPE_T	Trijų bitų laukas, aprašantis lauko TYPE turinį. Dažniausiai sutinkama TYPE_T lauko reikšmė lygi 1 arba 2. Kai ši reikšmė lygi 1, tai TYPE laukas yra MIME tipo, kai 2, tai URI tipo. Jeigu reikšmė lygi 0, tai reiškia, kad perduodami duomenys išskaidyti į kelis įrašus (Unchanged). Kai reikšmė lygi 3 tai reiškia, kad įrašas neturi jokios informacijos.
OPTIONS_LENGTH	16 bitų. Užduoda lauko OPTIONS dydį.
ID_LENGTH	16 bitų. Užduoda lauko ID dydį.
TYPE_LENGTH	16 bitų. Užduoda lauko TYPE dydį.
DATA_LENGTH	32 bitai. Duomenų bloko dydis.
OPTIONS	Skirtas ateities DIME praplietimams.
ID	Užduodamas įrašo identifikatorius. Šis identifikatorius leidžia kreiptis į įrašą kitiems įrašams.
TYPE	Įrašo duomenų tipas. Gali būti MIME tipo arba URI.
DATA	Duomenų blokas, kurio dydis gali siekti 4 GB.

