

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.
Danielius Comik

KAUNAS
2005

TURINYS

1	ĮVADAS	4
1.1	Dokumento paskirtis	5
1.2	Santrauka.....	5
2	SISTEMOS ANALITINĖ APŽVALGA	8
2.1	Panašių sistemų analizė.....	8
2.1.1	Estonian Mobile Telephone firmos projektas „Mobile Parking“	8
2.1.2	AirClic firmos „Parking Servines“ sistema	9
2.1.3	Intelektualios automobilių statymo sistemos diegimas Ostino Universitete	11
2.1.4	Protinga automobilių stovėjimo aikštelė „IPS“	12
2.1.5	Situacijos Lietuvoje įvertinimas	13
2.2	Užduoties sprendimo algoritmai ir metodai.....	14
2.2.1	Metodų ir priemonių parinkimas	17
2.2.2	Objektinio modeliavimo modelis (OMT)	17
2.2.3	OPEN metodas ir OML	18
2.2.4	UML pagal RUP	18
2.3	Sistemos specifikacija.....	19
2.3.1	Projektuojamas objektas	19
2.3.2	Projektuojamo objekto paskirtis	19
2.3.3	Projektuojamo objekto veiklos procesų analizė.....	19
2.3.4	Projektuojamo objekto sistemos funkciniai reikalavimai	23
2.3.5	Reikalavimai APS sisteminei įrangai.....	24
2.3.6	Reikalavimai serverio techninei įrangai.....	24
2.3.7	Reikalavimų serverio programinei įrangai specifikacija	24
2.3.8	Reikalavimai informacijos posistemei	25
2.3.9	Sistemos vartotojai.....	25
2.3.10	Reikalavimai vartotojo sąsajai	25
2.3.11	Eksploatavimo aplinka.....	26
2.3.12	Reikalavimai projekto dokumentacijai	26
2.3.13	Sistemos apribojimai.....	27
2.3.14	Reikalavimai eksperimentui.....	27
3.	SISTEMOS PROJEKTINĖ DALIS.....	28
3.1	Sistemos duomenų bazė.....	28
3.2	Sistemos kontekstas	35
3.3	Diagramos	36
3.3.1	Panaudojimo atvejų diagramos	36
3.3.2	Bendradarbiavimo diagrama.....	41
3.3.3	Sekos diagrama	43
3.4	Sistemos architektūra	45
3.5	Sistemos lygmenys.....	46
3.6	Organizacijos modelis.....	47
4.	TYRIMO DALIS	48
4.1	Sąsajos testavimas.....	48
4.1.1	Testavimo strategija	49
4.1.2	Vienetų testavimas	49
4.1.3	Integracinis testavimas.....	49
4.1.4	Patvirtinamasis testavimas	49
4.1.5	Aukštesnio lygio testavimas	50
4.1.6	Pasirinktas testavimo metodas	51
4.2	Microsoft SOAP Toolkit įrankis.....	51
5	EKSPERIMENTINĖ DALIS	54

5.1	Sistemos naudojimas.....	54
5.1.1	Prisijungimas.....	54
5.1.2	Rezervavimas.....	54
5.1.3	Informacijos įvedimas.....	55
5.1.4	Prisijungimas naudojant WAP technologiją.....	56
5.1.5	Automobilio statymo procesas.....	57
5.1.6	Vartotojo išvykimas.....	57
5.2	Našumo palyginimas perduodant binarinius duomenys skirtingais metodais ...	58
6	IŠVADOS	62
7	LITERATŪRA	64
7.1	Knygos	64
7.2	Prieigos per internetą.....	64
8	TERMINŲ IR SANTRUMPŲ ŽODYNAS	65
9	PRIEDAI.....	67

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 naujausias 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 artimiausią 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 trumpoji žinutė yra bene pats patogiausias apmokėjimo būdas, nes nesvarbu, kokiame 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 gaišamas 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 baigiamojo magistro 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

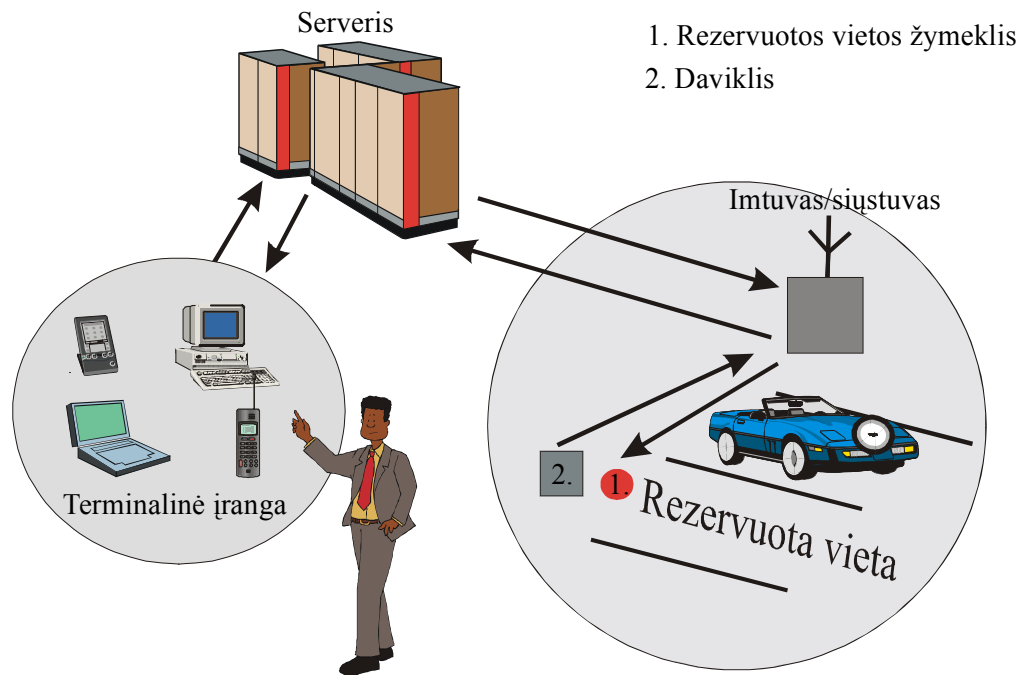
klientų brangų laiką ir pinigus, sumažinti kamščius ir taršą miestų centruose. Įdiegus mūsų stovėjimo sistemą mes išvengsime šių problemų.

Mūsų klientais bus visi vartotojai turintys automobilius ir terminalinę įrangą (asmeninį kompiuterį su priėjimu prie interneto, WAP ar GPRS technologijas palaikantį mobilųjį telefoną, kišeninį kompiuterį). Apmokėjimas bus labai paprastas, jei jums bus rezervuota vieta, nuo to momento pinigai pradkami 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 daugiavilniškės valdymo sistemos modelis

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

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. 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 projektavimų 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

2.1.1 Estonian Mobile Telephone firmos projektas „Mobile Parking“

EMT firmos sistema „Mobile Parking“ [2] yra unikalus sprendimas visiems aikštelių savininkams, sujungiantis senas ir naujas technologijas. EMT suteikia galimybę pertvarkyti aikšteles į lengvai naudojamą pritaikant elektroninę komerciją. Ši sistema suteiktų papildomų pajamų GSM tinklo operatoriams ir sutaupytų pinigų aikštelių savininkams.

EMT firmos „Mobile Parking“ sistema buvo pristatyta 2000 metų vasarą Estijoje. Nuo 2000 metų birželio mėnesio kamščių Estijos sostinėje Taline sumažėjo 25 procentais. Mokesčių rinkėjai surinko 12 procentų daugiau mokesčių už automobilių statymą, tai parodo, kad žmonės yra linkę tvarkingai mokėti, jei tik aptarnavimo lygis yra aukštas.

Žiūrint iš Talino savivaldybės pusės, jie eliminavo išsiskolinimų rinkimo riziką, sumažino nereikalingą biurokratiją, taip pat naudojant naujas technologijas sumažėjo administraciniai mokesčiai. Atvykstantiems užsienio svečiams į Taliną buvo daugybė problemų dėl automobilio statymo. Miesto savivaldybė bandė daugybę būdų, panaikino bilietus, grynųjų pinigų mašinas, bet kiekviena sistema turėjo daugybę problemų su

mokesčių surinkimu. EMT sukūrė labai efektyvią sistemą nepadidinant mokesčių vartotojams ir investavo daugiau nei 2 milijonus Estijos kronų į šį projektą. Praėjus trims mėnesiams po šio projekto testinės versijos paleidimo, mokesčių surinkimas padidėjo 12 procentų. EMT surado tinkamiausią būdą kaip pagerinti automobilių statymą, nes dauguma vairuotojų taip pat yra mobiliųjų telefonų vartotojai.

Pranašumai:

1. Eliminuoja grynujų pinigų mašinas ir asmenis, kurie operuotų grynaisiais pinigais.
2. Žymiai padidina mokesčių surinkimą.
3. Ženkliai sumažina aikštelių operatorių problemas susijusias su mokesčių surinkimu.
4. „Mobile Parking“ siūlo mobiliosios komercijos galimybę GSM tinklo operatoriams.
5. Lengva naudotis.

Automobilių savininkams „M-Parking“ suteikia galimybę atsikratyti monetų, kupiūrų ir kortelių. Jie užmoka už automobilio statymą kartu su savo mėnesine mobilaus telefono sąskaita ir jie moka būtent už tą laiką kurį jie prastovėjo.

Vartotojas paskambina trumpuoju numeriu ir atidaro savo virtualią sąskaitą. Jis gali pasirinkti tarp skirtingų numerių, kurie suteikia skirtingą pinigų limitą. Tai nepriklauso nuo stovėjimo trukmės ar vietos.

Norėdamas pastatyti automobilį vartotojas nusiunčia trumpąją žinutę, su savo automobilio numeriu, trumpuoju numeriu. Taip užfiksuoja stovėjimo pradžią. Taip pat sesija gali būti pradėta ir per WAP.

Norint užbaigti stovėjimo sesiją, vartotojas paskambina trumpuoju numeriu, taip yra užfiksuojamas laikas, kurį jis prastovėjo. Jei automobiliui dar stovint baigiasi pinigai iš virtualios sąskaitos, dešimt minučių prieš pasibaigiant yra išsiunčiama įspėjamoji trumpoji žinutė.

Vartotojas už stovėjimą turės sumokėti kartu su savo mobiliojo telefono sąskaita. Taip pat vartotojas turės susimokėti už trumpąsias žinutes. Daugiau jokių papildomų mokesčių.

2.1.2 AirClic firmos „Parking Services“ sistema.

Greičiausias, lengviausias ir naudingiausias būdas kontroliuoti automobilių stovėjimą [3].

Firma AirClic su savo unikalia „Mobile Information Platform“ ir „SmartCodes“ leidžia automobilių stovėjimo aikštelių kompanijoms teikti jų klientam pagerintą servisą,

supaprastintą apmokėjimą, tuo pačiu aprūpinti savo darbuotojus įrankiais, su kuriais jie galėtų kontroliuoti automobilių stovėjimo vietas. AirClic suteikia klaidų sumažinimą iki minimumo, realaus laiko mobilią informaciją ir stovėjimo mokesčių surinkimą taip pat gerai, kaip ir realiam laike.

AirClic pristato mobilius automobilių stovėjimo sprendimus:

- Žemomis kainomis
- Minimaliomis turto išlaidomis
- Ribotais pakeitimais esamai sistemai

Automobilių aikštelių kompanijos nori tiekti savo klientam paslaugą, kuria būtų lengva prisiregistruoti ir išsiregistruoti, bei apmokėti už automobilio stovėjimą. Taip pat jos nori, kad stovėjimo vietų prižiūrėtojai galėtų lengvai kontroliuoti aikšteles, patikrinti ar tam tikra mašina turi teisę stovėti toje vietoje.

Automobilių aikštelių kompanijos pasiekė tai savo klientam su AirClic suteiktu prisiregistravimo/išsiregistravimo ir apmokėjimo sprendimu, kuris veikia su bet koku mobiliu įrenginiu, kurį turi vartotojas.

- Vartotojas susisiekiama su automobilių statymo servisu pateikdamas jam tokią informaciją
 - Savo mobiliojo telefono numerį
 - Savo transporto priemonės numerį
 - Atsiskaitymo informaciją
- Automobilių aikštelių kompanijos išduoda vartotojui prie lango klijuojamą lipduką, kuriame yra brūkšninis kodas, kurį vartotojas prisiklijuoja prie savo automobilio stiklo.
- Kai vartotojas atvyksta į automobilių statymo vietą, jis paprasčiausiai informuoja kompaniją atsakingą už automobilių stovėjimą
 - Pranešdamas aikštelę, zoną ir vietos numerį
 - Pasirenka arba tam tikrą laiką, arba vėl pranešti kada jis paliks tą vietą
- Vartotojas pateikia šią informaciją per WAP, arba trumpąja žinute, arba per garsinę sistemą, priklausomai nuo jo pasirinkimo. Automobilių aikštelių kompanija turėdama vartotojo telefono numerį gali jį apmokestinti
- Jei vartotojas pasirenka tam tikro laiko stovėjimą, 15 minučių prieš pasibaigiant laiką jis gaus išpėjamąjį skambutį arba trumpąją žinutę, tačiau taip pat jis galės ir pratęsti stovėjimą

- Paliekant stovėjimo aikštelę, vartotojas tiesiog išvažiuoja, jei jis buvo pasirinkęs „kai išvažiuosiu tada pranešiu“ opciją, jis informuoja kompaniją ir ji žino tikslų laiką, kiek vartotojas stovėjo.

Automobilių stovėjimo aikštelių prižiūrėtojai yra aprūpinti skeneriais suderintais su mobiliuoju telefonu. Paprasčiausiai jie praeidami pro automobilį nuskenuoja brūkšninį kodą ir mato savo mobiliojo telefono ekrane informaciją apie tą automobilį, ar sumokėtą už jo stovėjimą.

Šiuo metu yra įdiegta keletas AirClic kompanijos sistemų, iš kurių kiekviena turi savitų savybių.

2.1.3 Intelektualios automobilių statymo sistemos diegimas Ostino Universitete

Dėl didėjančio, vairuotojų susidomėjimo realaus laiko informacijos priėjimu keliaujant į konkrečią vietą, įgalina protingą transportavimo sistemą (Intelligent Transportation Systems („ITS“)) susikoncentruoti skleisti realaus laiko informaciją. Kadangi centrinės biznio sritys, oro uostai, tranzito stotys ir prekybos centrai ir toliau pastoviai perpildyti piko metu, tai vis dažniau iškyla poreikis gauti realaus laiko automobilių stovėjimo informacijos. Universiteto aplinka, ne išimtis šioje situacijoje. Mažėjant stovėjimo vietų skaičiui ir didėjant priėmimams į universitetus ir valdžios bei fakultetų skaičiui, pastarieji pagaliau supranta teisingo, prieinamo automobilių statymo išdėstymo svarbą.

Intelektuali automobilių stovėjimo aikštelių sistema „IPS“ gali pateikti gerą maršrutą universiteto valdžiai, kad saugiai ir greitai surasti laisva stovėjimo vietą.

Texas'o universitetas (JAV) Ostine – puikus pavyzdys. Čia galime rasti pagrindines automobilių statymo problemas su kuriomis susiduria šis universitetas. Kadangi šis universitetas randasi Ostino biznio centro srityje, jis privalo teikti atitinkamą stovėjimo paslaugą studentams, fakultetų vadovybei ir valdžiai, bei lankytojams, apkrautame augančiame mieste. Mažėjantis stovėjimo vietų skaičius ir universiteto narių skaičiaus didėjimas jau iškreipė apkrautą infrastruktūrą.

Intelektuali automobilių stovėjimo sistema gali padėti universitetui iš naujo perskirstyti stovėjimo vietas ir sumažinti grūsčių bei nelegalių automobilių pastatymų skaičių. Taip pat universitetas yra numatęs sistemą, kuri padėtų rasti reikiamą kelionės tikslą universiteto teritorijoje, ši sistema papildytų „IPS“ paketą. Krypties ir stovėjimo

padėties informacija gali būti rodoma naudojantis kintančiais pranešimų ženklais – „VMS“. Kita vertus ši priemonė yra brangi ir užgriozdins universitetų gatves.

„APMS“ programinė įranga leistų surinkti visą informaciją apie galimas stovėjimo vietas visuose universiteto garažuose centrinėje dalyje. Kai informacija būtų surinkta, koku metodu ją panaudoti galėtų pasirinkti pats universitetas.

Internetinės ir mobilios aplikacijos atrodo idealios alternatyvos nepatogiems ir brangiems „VMS“ ženklams. Padidėjęs bevielių technologijų naudojimo lygis kartu su bevielių aplikacijų pasiekimais transporto srityje, gali pasiūlyti, kad skaitmeninės informacijos skleidimas gali būti atsakymas į vežimo kaštų informacijos spragas.

Tinkamiausia universiteto „IPS“ aplikacija turėtų teikti realaus laiko stovėjimo informaciją, sumažinti grūstis, ir perskirstyti stovėjimo išteklius. Kad atitiktų visus šiuos reikalavimus Intelektuali automobilių stovėjimo sistema turi būti naudojama pačių universiteto šeiminkų, jei ne – tada „IPS“ praktiškai neturės jokios įtakos universiteto automobilių stovėjimo problemai.

2.1.4 Protinga automobilių stovėjimo aikštelė „IPS“

„IPS“ yra sistema, kuri naudoja pažangiausią informacinių komunikacijų technologiją, kad realizuotų naują kelių eismo ir transporto aptarnavimą. Kad būtų sukonstruota kaip transporto priemonių, kelių ir žmonių integruota sistema staigiai didėjančio saugumo, transporto efektyvumo ir komforto tikslui, pastangos yra daromos sudominti „IPS“ nacionaliniu lygiu. Kaip dalis šio pasiūlymo, „IPS“ (protingos automobilių statymo sistemos) prototipas buvo suprojektuotas įtraukiant rezervavimo valdymo sistemą, pritaikytą automobilių statymo verslui ir rezervacijos autentifikavimo mechanizmas. Tikrinimas testavimu parodė sistemos palankius rezultatus.

„IPS“ naudoja internetą nustatyti rezervavimo vietą su automobilių aikštelių rezervacijos serveriu per asmeninį kompiuterį arba mobilųjį terminalinį įrenginį. Apmokėjimas yra padaromas per internetą. Kai tik rezervacija yra padaryta, vartotojas gali įvažiuoti į aikštelę naudojantis radijo žymekliu, įveddamas savo slaptą kodą į klaviatūrą, prijungtą prie įvažiavimo sekcijos Rezervavimai per mobilų terminalinį įrenginį yra autentifikuojami naudojant telefono numerį, iš kurio rezervacija buvo padaryta.

Kad garantuotumėme rezervacijos laikus, „IPS“ siūlo vartotojams stovėjimo vietos rezervacijos pasirinkimą nuo rezervavimo laiko arba tarp tiksliai apibrėžtų įvažiavimo ir išvykimo laikų. Taip pat valdomas reguliarus transporto priemonių sambūvis su tais, kurie

rezervavo stovėjimo vietas, tuo būdu palengvinant efektyvias automobilių statymo operacijas.

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 tiksliai 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	Būtinus susisiekimą su firma dėl papildomos įrangos	Paprastumas naudotis
Automobilių aikštelių sprendimai Lietuvoje	Ne	Nebūtinus	Nebūtinus	Ne
Ascom – „E-parking“	Taip	Nebūtinus	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 Užduoties sprendimo algoritmai ir metodai

Ši sistema skirta tiek klientui, tiek aptarnaujančiam personalui, kuris tikrina aikštes. Pateikiami kliento ir tikrintojo algoritmai.

2 lentelė „APS“ kliento algoritmas

Žingsnis

Veiksmas

- 1 Prisijungimas:
 1. naujo vartotojo registracija
 2. esamo vartotojo prisijungimas
- 2 Meniu punkto „klientas“ pasirinkimas ir duomenų įvedimas:

1. vartotojas prisijungia prie sistemos
2. vartotojas pateikia duomenis
3. vartotojas nurodo kur yra laisvos vietos
4. duomenis yra siunčiami į DB
- 3 Užklausos siuntimas iš mob. telefono
 1. vartotojas iš mob. telefono siunčia užklausą
 2. vartotojas gauna atsakymą į ekraną
 3. duomenis patalpinami į DB
- 4 Rekomendacijos ir papildomos informacijos surinkimas
 1. pasirenkama už ką bus mokama
 2. duomenys siunčiami į duomenų bazę
 3. vartotojas pasirenka norimą laiką ir vietą
- 5 Galutinio sprendimo priėmimas

Vartotojas sistemoje gali atlikti tokius veiksmus:

- Redaguoti asmeninius duomenis
- Redaguoti prisijungimo duomenis
- Rezervuoti vietą
- Naudotis pagalbos sistema

Kiekviename sistemos žingsnyje yra grįžimo procedūra, jeigu vartotojas ką nors pamiršo ir nori pakeisti, ar papildyti duomenis. Jeigu jam reikia pradėti viską iš pradžių, tai tam yra skirtas punktas “Į pradžių” ir jis viską gali pradėti iš naujo.

Vartotojo algoritmas plačiau:

Visų pirma sistema gauna iš vartotojo tam tikrą rinkinį duomenų, remiantis kuriais ji turi atlikti tam tikros vietos rezervaciją. Tam naudojamas sekantis algoritmas.

- *Prisijungimas.*

Taigi pirmas žingsnis. Paleidęs programą vartotojas mato priešais save prisijungimo langą. Jei tai naujas vartotojas, jis turi užsiregistruoti ir pateikti savo asmeninius duomenis. Esamas vartotojas suveda savo prisijungimo duomenis. Toliau sistema pereina prie sekančio punkto.

- *Meniu punkto „klientas“ pasirinkimas ir duomenų įvedimas.*

Po to kai vartotojas įveda prisijungimo duomenis, sistema prisijungia. Tuomet vartotojas gali pateikti duomenis apie jį dominčią vietą.

1. vartotojas prisijungia prie sistemos
2. vartotojas pateikia duomenis
3. vartotojas nurodo kur yra laisvos vietos
4. duomenis yra siunčiami į DB

- *Užklausos siuntimas iš mob. telefono.*

Vartotojas gali prisijungti prie sistemos per mobilųjį telefoną.

1. vartotojas iš mob. telefono siunčia užklausą
2. vartotojas gauna atsakymą į ekraną
3. duomenis patalpinami į DB

- *Rekomendacijos ir papildomos informacijos surinkimas.*

Reikia pažymėti, kad klientą dominantį vietą gali būti užimta, todėl norėdama patikslinti kai kurias detales, sistema vartotojui uždavinėja papildomus klausimus. Realu, kad į kai kuriuos klausimus vartotojas negalės atsakyti nieko. Todėl sistema turėtų duoti patarimus kaip jam apsispręsti. Pasirinkti kitą aikštelę, gauti maršrutą iki kitos aikštelės ir t.t.

- *Galutinio sprendimo priėmimas*

Po šių žingsnių sistemai lieka rezervuoti vietą. Vartotojas atsijungia nuo sistemos. Klientui rezervavus arba atvykus į stovėjimo vietą, pradedamas skaičiuoti laikas. Transporto priemonei išvažiuojant, fiksuojama bendra stovėjimo trukmė, šie duomenys siunčiami į duomenų bazę. Iš ten vartotojui atsiunčiama apmokestinimo ataskaita: prastovėtas laikas ir suma.

Yra labai svarbu užtikrinti normalų sistemos funkcionalumą, ką padeda padaryti aptarnaujantys asmenys, tikrintojai. Jie taip pat gali naudotis šia sistema.

Žingsnis	Veiksmas
1	Prisijungimas
2	Tikrintojas žiūri ar viskas yra tvarkingai sumokėta
3	tikrinama rezervavimas konkretaus vartotojo rezervacija
4	siunčiama užklausa
5	gaunama užklausa atgal
3	Atsijungimas

- *Tikrintojas žiūri ar viskas yra tvarkingai sumokėta*

Prisijungus prie sistemos kaip tikrintojui galima patikrinti konkretaus vartotojo rezervaciją. Tikrintojas mato už kiek laiko baigsis rezervacija ir ar išviso jis turi teisę ten stovėti.

2.2.1 Metodų ir priemonių parinkimas

Yra daug metodų ir priemonių sistemos analizei atlikti. Žemiau pateikti keli iš jų:

2.2.2 Objektinio modeliavimo modelis (OMT)

OMT [12]– objektinio modeliavimo metodas, kuris apima trijų tipų modelius: objektų, dinaminį ir funkcinį. Objektų modelis aprašo objektus, klase bei jų ryšius, dinaminis modelis – sąveikas tarp objektų klasių, funkcinis modelis – sistemoje vykstančias duomenų transformacijas. Kiekvienas iš šių trijų modelių susideda iš kelių diagramų, tai leidžia smulkiai išskaidyti problemą ir ją korektiškai atvaizduoti diagramoje. Šiame metode visi naudojami objektai yra identifikuojami – turi vardą, juos galima atskirti vieną nuo kito. Objektai jungiami į klases, turinčias savo atributus, savybes, operacijas. Šis metodas dar ypatingas tuo, kad galima objektus rūšiuoti, įgyvendinti klasių savybių paveldėjimą.

2.2.3 OPEN metodas ir OML

Tai taip pat objektinio modeliavimo krypties metodas. OPEN – Object-oriented Process, Environment and Notation [13]. OPEN metodas apima visus būtinus metodui atributus, ne tik modeliavimo kalbą ir metamodelį, kaip UML. OPEN metodo esmė yra ryšiai tarp užduočių (Tasks) ir gyvavimo ciklo veiklų (Activities). Šiame metode apibrėžtos veiklos (activities), suskirstytos į etapus:

- 1) Biznio organizavimo etapas (business build stage);
- 2) Realizavimo etapas (build phase);
- 3) Įdiegimo etapas (delivery phase).

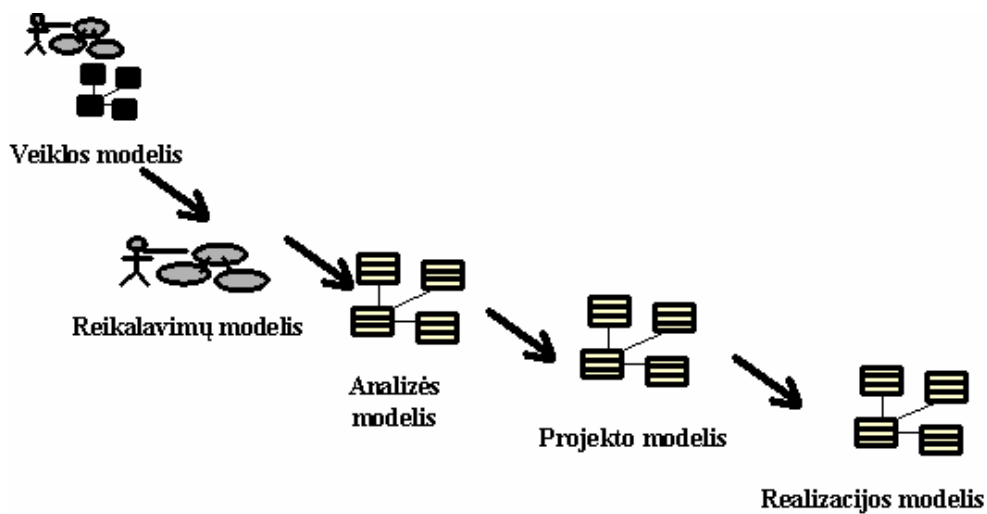
UML yra “data –driven” – duomenimis grindžiamas objektinis IS kūrimo metodas.

OML yra “responsibility-driven” – pareigomis (atsakomybėmis) grindžiamas IS kūrimo metodas.

2.2.4 UML pagal RUP

UML yra tikrai modeliavimo kalba, projektavimo metodas apima kalbą, kūrimo metodą ir procesą. Kartu su UML taikomas universalus kūrimo procesas Rational Unified Process (RUP). RUP turi ir metodo, ir proceso elementų, tačiau tai daugiau procesas. UML galima naudoti su daugeliu objektinio projektavimo metodų.

Remiantis RUP, projektuojant sistemą sudaroma eilė modelių, kuri (2 pav.):



2 pav. RUP modeliai

Šiame magistriniame darbe sistema APS kuriama naudojant RUP metodiką. Pasirinkome šį modelį, nes jis pilnai atitinka mūsų darbo struktūrą bei reikalavimus. Modelis yra plačiai naudojamas ir jį galima naudoti su daugeliu objektinio projektavimo metodų.

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 priejimu 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 sutaupytas laikas ir gautas rezultatas. Šiais faktoriais ir yra paremta kuriama sistema.

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

Mūsų klientai – asmenys, naudojantys kompiuterines(asmeninį kompiuterį su prieiga prie interneto, kišeninį kompiuterį) bei ryšių technologijas(WAP protokolą, GPRS ar “BlueTooth” technologiją palaikantį 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;
 - siųstuvas/įmtuvas, 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*)pradedą skaičiuoti laiką.

III etapas: atvyksta vartotojas i automobilio stovėjimo vietą aikštelėje.

- atvyksta klientas iš anksto rezervavęs vietą, ir vietos sensorius parodo jo teisėta 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 sąskaitą. 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/imtuvo, 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 J2SE 1.4.1 SDK 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, integruotas;
„Touch Screen“ – jautrus lietimui ekranas

2.3.11 Eksploatavimo aplinka

- Aparatūra stovėjimo vietoje turi atitikti esamas gamtines sąlygas;
- Aparatūra turi veikti be sutrikimų (nuo -30⁰C iki 55⁰C) 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. Limanuskienės " 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 $+40^{\circ}$ 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 PROJEKTINĖ DALIS

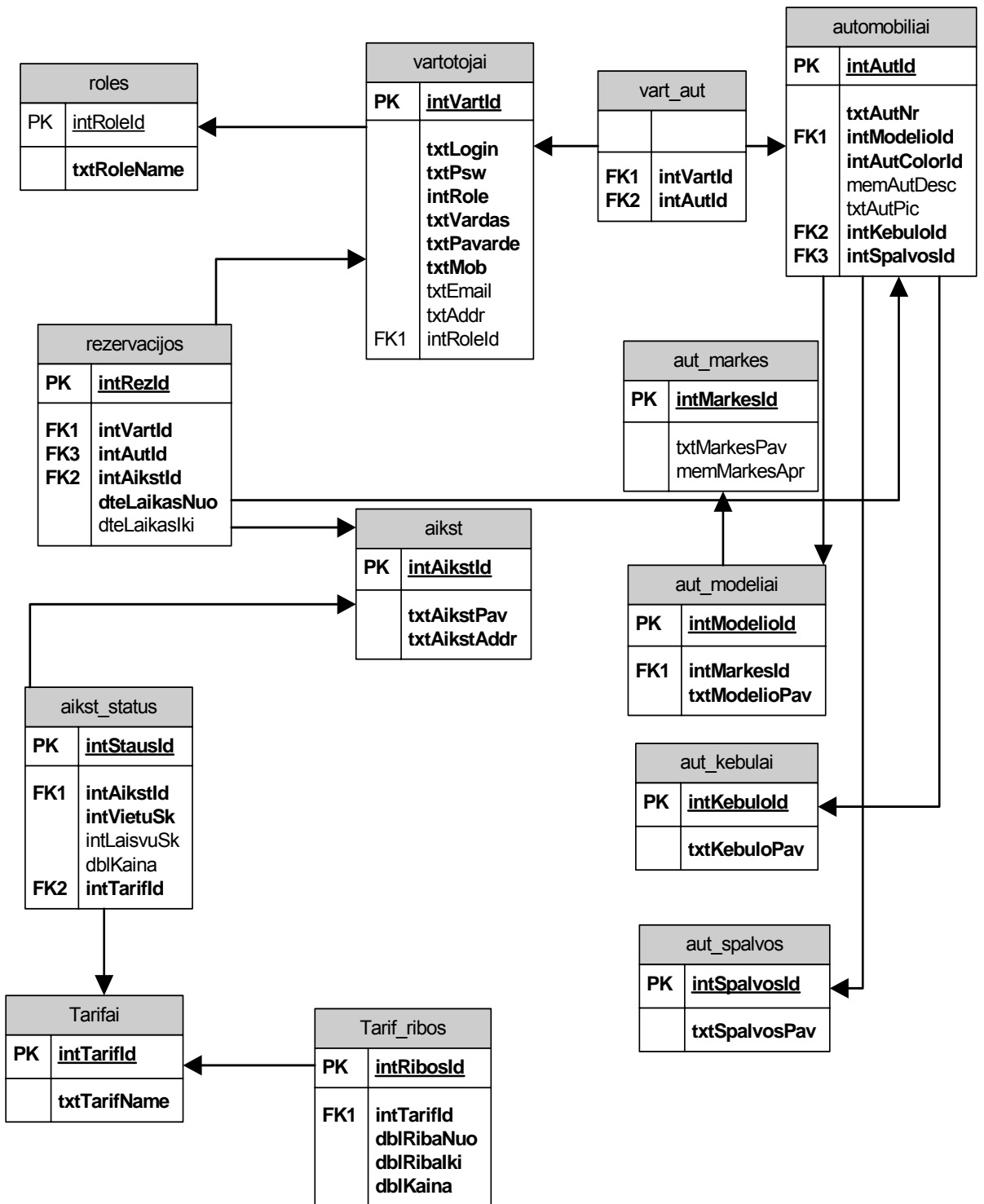
3.1 Sistemos duomenų bazė

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

4 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

5 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

6 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

7 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

8 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

9 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

10 lentelė. Aut_kebulai

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

Lentelė aut_spalvos

11 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

12 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

13 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 aikšteleje yra vietų	Sveikas skaičius iki 6 skaitmenų
intLaisvuSk	Kiek aikšteleje yra laisvų vietų	Sveikas skaičius iki 6 skaitmenų
intTarifId	Tarifo identifikatorius iš lentelės <i>tarifai</i>	Sveikas skaičius iki 12 skaitmenų

Lentelė tarifai

14 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

15 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
dblRibaki	Suma, iki kurios aktyvuojamas tarifas	Sveikas skaičius
dblKaina	Tarifo dydis	Sveikas skaičius

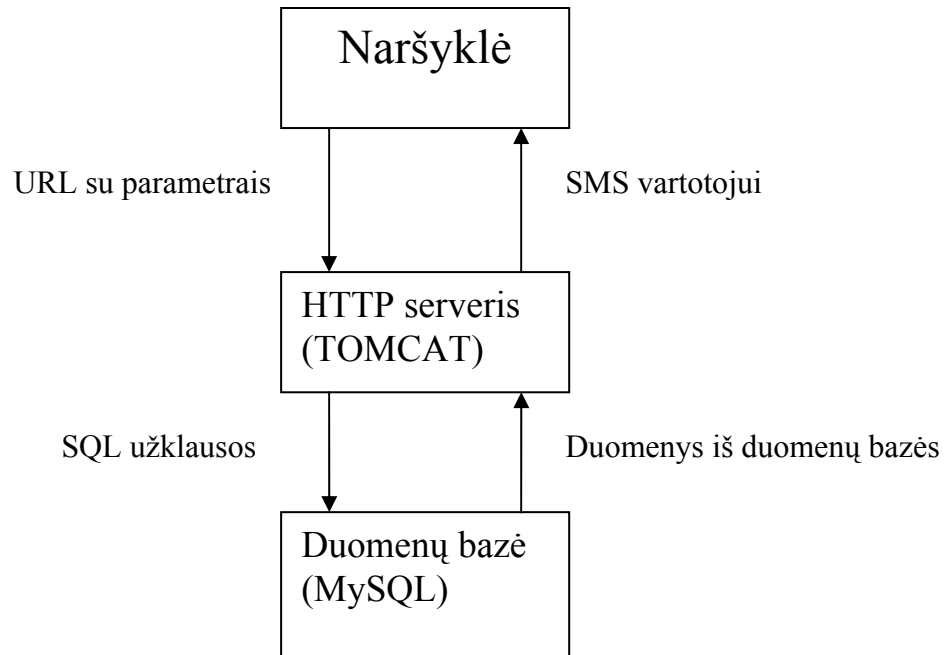
Lentelė rezervacijos

16 Lentelė. Realizacijos

Laukas	Aprašymas	Tipas
intRezId	Rezervacijos identifikatorius	Sveikas skaičius iki 12 skaitmenų
intVartId	Vartotojo identifikatorius iš lentelės <i>virtotojai</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 Sistemos kontekstas

Iš vartotojo įvestų duomenų apie stovėjimo aikštelės rezervavimą yra suformuojamas URL su parametrais, kuriuos įvedė vartotojas ir jie paduodami į HTTP serverį. Iš HTTP serverio SQL užklausų pagalba surašomi duomenis į duomenų bazę. Toliau duomenys iš duomenų bazės patenka į HTTP serverį. Iš serverio į telefoną ateina SMS vartotojui:



4 pav. APS kontekstas

3.3 Diagramos

3.3.1 Panaudojimo atvejų diagramos

Panaudojimo atvejų ir aktorių sąrašai

Aktorių sąrašas:

- administratorius
- vartotojas 1 (klientas)
- vartotojas 2 (tikrintojas)
- sistema

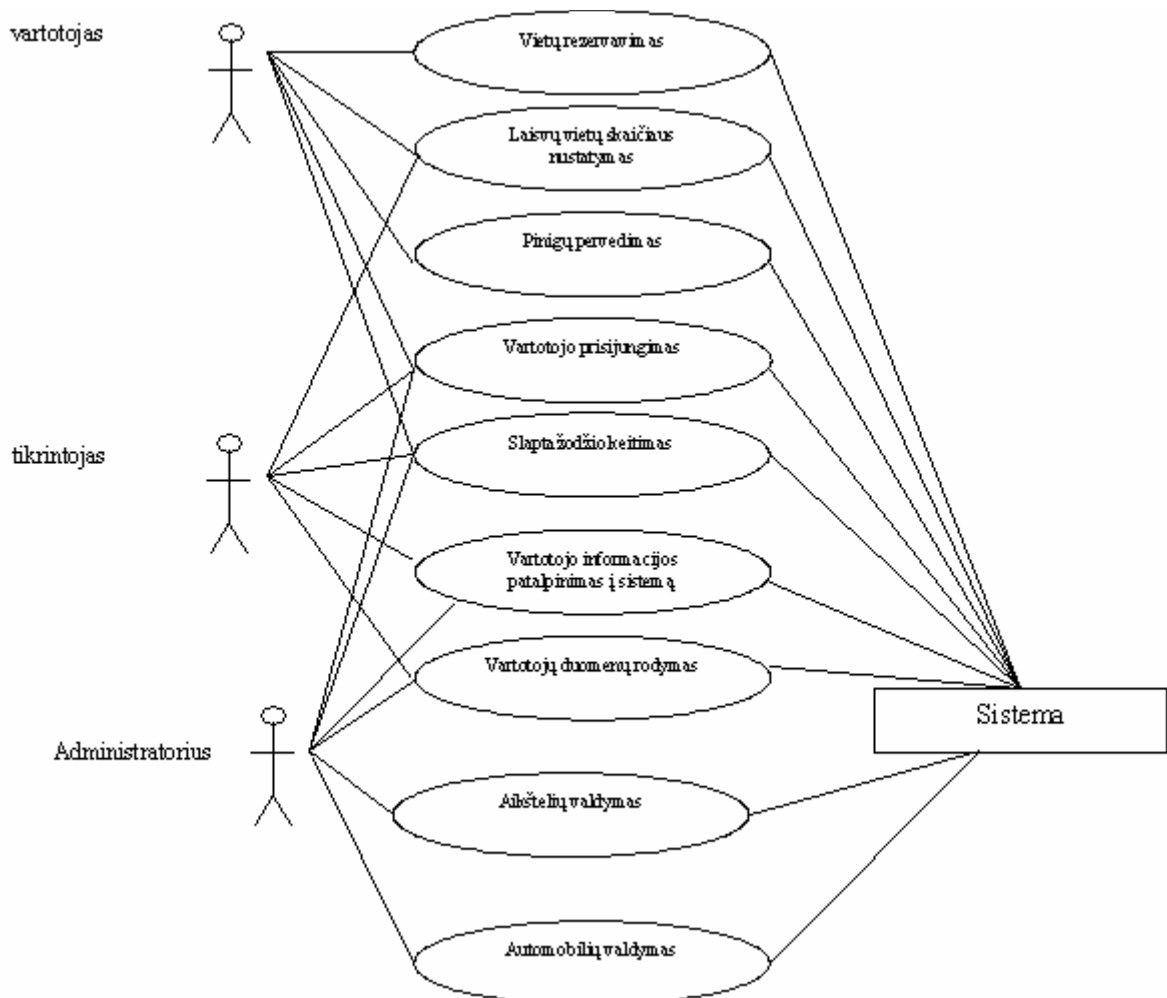
Panaudojimo atvejai:

- Vietų rezervavimas
- Laisvų vietų skaičiaus nustatymas
- Pinigų pervedimas
- Vartotojo prisijungimas
- Stovėjimo laiko priminimas
- Vartotojo informacijos patalpinimas į sistemą
- Vartotojų duomenų rodymas

PAM diagrama

Aktorių funkcijos

- Administratorius- atsakingas už sistemos priežiūrą. Periodiškai patikrina ar sistemoje nėra problemų ar ji gerai veikia ar vartotojai nenaudoja sistemos ne pagal paskirtį. Jei aptinka piktybiškų vartotojų, jiems gali uždrausti prisijungimą prie sistemos, o jei aptinka klaidas, ištaiso jas.
- Vartotojas 1 - aktyvus sistemos naudotojas. Galintis rezervuoti vietas ir gauti informaciją apie stovėjimo vietas.
- Vartotojas 2 – tai tikrintojas, kuris gali patikrinti kaip rezervuota vieta, ar nėra klaidingų rezervavimų, ar vartotojas užmokėjo už rezervavimą. Kaip „žaliukas“ dabartiniu metu miestuose.
- Sistema- aptarnauja vartotojus ir administratorių.



5 pav. Panaudojimo atvejų diagrama

Panaudojimo atvejų specifikacijos

Vartotojo prisijungimas

Tikslas: prisijungti prie sistemos ir rezervuoti bei gauti informacijos apie vietas.

Aktoriai: vartotojas, sistema.

Ryšiai su kitais PA: nėra.

Nefunkciniai reikalavimai: nėra

Prieš-sąlygos:

3. pasirinkta interfeiso kalba
4. vartotojas dirbantis su sistema yra autorizuotas

Sužadinimo sąlyga: iš vartoto sąrašo meniu punkto „klientas“ pasirinkimas ir duomenų įvedimas

Po-sąlyga: duomenis yra patalpinti į sistemą

Pagrindinis scenarijus:

1. vartotojas prisijungia prie sistemos
2. vartotojas pateikia duomenis apie vietas
3. vartotojas nurodo kur yra laisvos vietos
4. duomenis yra siunčiami į DB

Alternatyvus scenarijai: nerezervuoja vietą.

Vartotojo informacijos patalpinimas į sistemą

Tikslas: patalpinti vartotojo duomenis į DB

Aktoriai: vartotojas 1, vartotojas 2, sistema

Ryšiai su kitais PA: vartotojo prisijungimas

Nefunkciniai reikalavimai: turi stabiliai veikti

Prieš-sąlygos:

1. sutvarkyta DB

Sužadinimo sąlyga: vartotojas iš mob. telefono siunčia užklausą

Po-sąlyga: nėra

Pagrindinis scenarijus:

4. vartotojas iš mob. telefono siunčia užklausą
5. vartotojas gauna atsakymą į ekraną

6. duomenis patalpinami į DB

Alternatyvus scenarijai: vartotojas palieka sistemą

Vartotojų duomenų rodymas

Tikslas: vartotojas 2 galėtų pažiūrėti ar viskas yra tvarkingai sumokėta.

Aktoriai: vartotojas 2, sistema

Ryšiai su kitais PA: nėra

Nefunkciniai reikalavimai: sistema turi veikti korektiškai

Prieš-sąlygos:

- pasirinktas vartotojo rezervavimas
- tikrinamas rezervavimas konkretaus vartotojo
- siunčiama užklausa vartotojo 2 į sistemą
- gaunama užklausa atgal

Sužadinimo sąlyga: per vartoto 2 interfeisą pasirinktas punktas 'tikrintojas'

Po-sąlyga: nėra

Pagrindinis scenarijus:

1. užklausa keliauja į sistemą
2. gaunamas atsakymas atgal

Alternatyvus scenarijai: netikrina

Vietų rezervavimas

Tikslas: – tai yra procesas, kurio pagalba aktorius rezervuos sau laiką ir vietą stovėjimo aikštelėse. Jam reikės jo telefonu nusiusti žinutę arba pasinaudojus WAP ir GPRS paslaugomis užsisakyti sau vietą. Nusiuntęs užklausa į mūsų serverį jis gaus patvirtinimą ir galės statyti mašiną į tą vietą, kuria jis užsakė.

Aktoriai: vartotojas

Ryšiai su kitais PA: nėra

Nefunkciniai reikalavimai: vartotojo prisijungimo prie sistemos įrankis-Interneto naršyklė ir mobilusis telefonas, tam tikras charakteristikas

Prieš-sąlygos: vartotojas prisijungęs prie sistemos

Sužadinimo sąlyga: vartotojo interfeiso meniu punkto išsirinkimas

Po-sąlyga: duomenys siunčiami i duomenų bazę

Pagrindinis scenarijus:

1. vartotojas prisijungia prie sistemos
2. vartotojas pateikia duomenis (automobilio numeris, stovėjimo vieta, laikas, kas jįsai yra (klientas ar tikrintojas)

Alternatyvus scenarijai: vartotojas palieka sistemą

Laisvų vietų skaičių nustatymas

Tikslas: - tai informacija apie laisvas vietas esančias aikštelėje. Kuri bus atvaizduojama displėjuje. Jeigu vieta bus laisva, tai aktorius matys tam tikru būdu mes atvaizduosime tai displėjuje.

Aktoriai: displėjus

Ryšiai su kitais PA: laisvų ir užimtų vietų skaičius

Nefunkciniai reikalavimai: duomenų bazėje turi būti duomenys

Prieš-sąlygos:

1. pasirinkti duomenys
2. duomenų bazėje yra prieš tai pasirinkti duomenys

Sužadavimo sąlyga: vartotojas duoda užklausą

Po-sąlyga: nėra

Pagrindinis scenarijus:

1. vartotojas iš mobilaus telefono ar interneto duoda užklausą
2. vartotojas gauna duomenis
3. į ekraną išvedama informacija apie vartotojo pasirinktą užklausą

Alternatyvus scenarijai: vartotojas palieka sistemą

Pinigu pervedimas

Tikslas: - tai pinigų kelias, kurio pagalba aktoriaus pinigai bus pervedami į mūsų sąskaitas. Tai reikalinga tam , kad mes gautume tos pinigus. Aktorius sumokės, sistema nuskaičiuos nuo jo telefono sąskaitos. Toliau pinigai bus pervedami į mūsų sąskaitas.

Aktoriai: apmokestinimo sistema

Ryšiai su kitais PA: nėra

Nefunkciniai reikalavimai: pervedimas (sistema) turi veikti korektiškai

Prieš-sąlygos:

4. pasirinkta už ką bus mokama
5. duomenys nusiusti į duomenų bazę
6. nurodyti visi duomenys, reikalingi korektiškam pervedimui

Sužadinimo sąlyga: apmokama už stovėjimą

Po-sąlyga: nėra

Pagrindinis scenarijus:

1. duomenys siunčiami į duomenų bazę
2. vartotojas pasirenka norimą laiką ir vietą

Alternatyvus scenarijai: nėra

3.3.2 Bendradarbiavimo diagrama

„Main“: Stovėjimo sistema

Tai mūsų pagrindinė sistema, apjungianti visas dalis i visuma.

[važiavimas: Masinos sensorius

Tai elementas fiksuojantis apie esamą padėti automobilio stovėjimo vietoje ir siunčiantis šią informaciją į duomenų bazę. Šis elementas gali būti realizuotas keliais būdais: šviesos diodu (žalia šviesa-vieta neužimta; raudona-vieta užimta), kuoliukas (nuleistas-vieta neužimta, nuleistas-vieta užimta), skaitinė-žodinė (vieta neužimta, vieta užimta). Šis sensorius bus sujungtas su tai gatvei arba tam tikram stovėjimo vietų zonai pastatytu aparatu (vietinė duomenų bazė) ir nuolat siūs informaciją apie konkrečią stovėjimo vietą.

Užklausa: GPRS

Užklausa-tai vartotojo siunčiamas pranešimas į pagrindinę duomenų bazę apie jį dominančią informaciją. GPRS - priemonė skirta perduoti vartotojo užklausiai į pagrindinę duomenų bazę (Main). Tai nauja technologija kuri yra prieinama iš kiekvieno terminalinio įrenginio palaikančio GPRS. Ja gali būti siunčiama tiek žodinė – skaitinė, tiek vaizdinė informacija. Šios technologijos pagalba galima pateikti reikiamą informaciją ir taip pat sėkmingai atlikti apmokestinimą už automobilio stovėjimą norimoj vietoj.

Užklausa: SMS

SMS- priemonė skirta perduoti vartotojo užklausiai į pagrindinę duomenų bazę (/Main). Tai technologija kuri yra prieinama iš kiekvieno mobiliojo ryšio aparato. Vartotojas išsiuntęs SMS-užklausa, netrukus gaus atsakymą apie jį dominančią stovėjimo dislokaciją SMS tekstu. SMS pagalba galima gauti informaciją ir taip pat atlikti apmokestinimą.

Užklausa: WAP

WAP- tai specialus duomenų perdavimo protokolas bevielams įrenginiams, jungiantis dvi-mobiliojo ryšio ir interneto technologijas. Mobiliojo ryšio priemonė siunčia užklausa į tokį vertimą atliekantį serverį. Iš šio užklausa keliauja į įprastinį WEB serverį, o gauta informacija tuo pačiu keliu pasiekia mobilaus ryšio priemonės vartotoją.

Kliento transporto priemonė:

Tai kiekvieno aktoiaus vairuojamas automobilis, kuriam jis nori rezervuoti vietą mūsų stovėjimo sistemos pagalba.

Vietos sensorius

Tai elementas fiksuojantis būseną transporto priemonei palikus stovėjimo vietą ir siunčiantis šią informaciją į duomenų bazę. Šis elementas gali būti realizuotas keliais būdais: šviesos diodu (žalia šviesa-vieta neužimta), kuoliukas (nuleistas-vieta neužimta), skaitinė-žodinė (vieta neužimta). Šis sensorius bus sujungtas su tai gatvei arba tam tikram stovėjimo vietų zonai pastatytu aparatu (vietinė duomenų bazė) ir siūs informaciją į ją.

Išvažiavimas:

Kliento transporto priemonė palieka stovėjimo vietą ir fiksuojamas prastovėtas laikas.

Apmokestinimo sistema

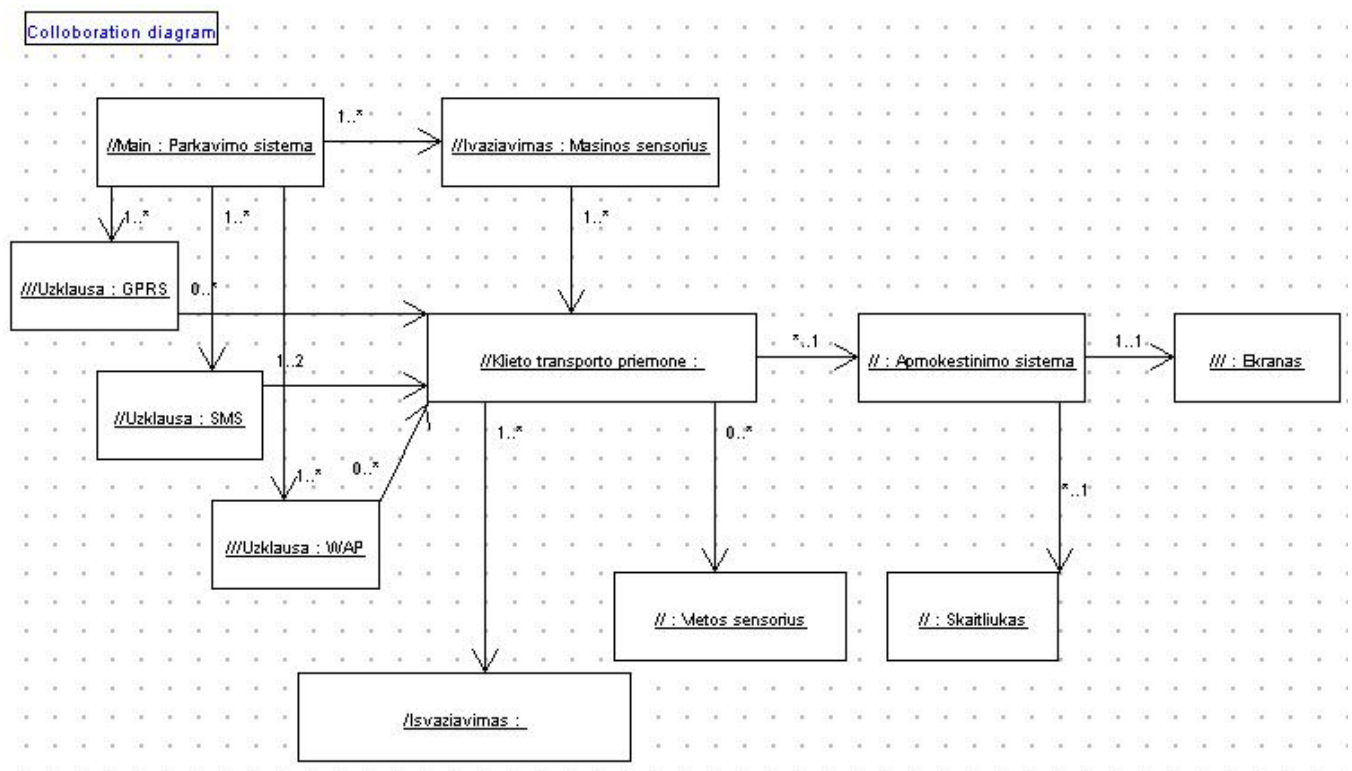
Kliento atsiskaitymo už paslaugas būdas. Klientui rezervavus arba atvykus į stovėjimo vietą, pradedamas skaičiuoti laikas. Transporto priemonei išvažiavus, fiksuojama bendra stovėjimo trukmė, šie duomenys siunčiami į duomenų bazę. Iš ten vartotojui atsiunčiama apmokestinimo ataskaita: prastovėtas laikas ir suma. Apmokestinama bus iš vartotojo abonentinės sąskaitos, kurią jis turi šias paslaugas teikiančioje firmoje.

Skaitliukas

Tai mechaninis įrenginys kaupiantis informaciją apie laisvas ir užimtas stovėjimo vietas, taip pat atliekantis laikmačio funkciją, t.y. klientui rezervavus arba atvykus į stovėjimo vietą, pradeda skaičiuoti laiką.

Ekranas

Vartotojo terminalinio įrenginio ekrane, priklausomai nuo jo užklauso tipo, bus pateikiama informacija apie laisvas ir stovėjimo vietas, detali apmokestinimo tvarka, detalus planas kaip patekti į norimą stovėjimo vietą.



6 pav. Bendradarbiavimo diagrama

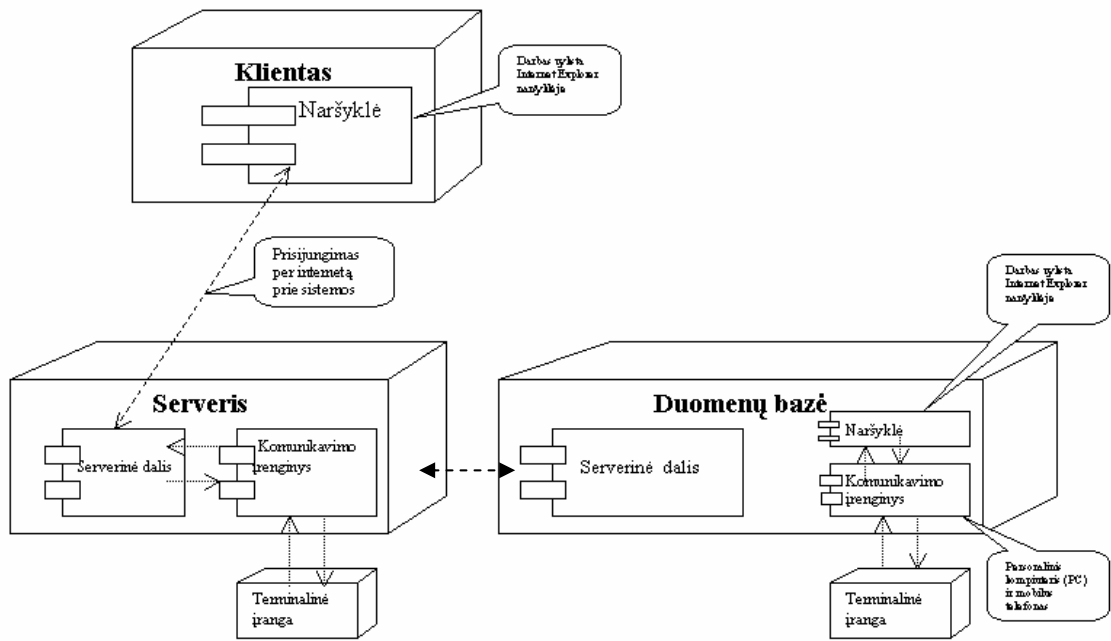
3.3.3 Sekos diagrama

Tam, kad pavaizduoti objektų sąveikas laike naudojame *sekos* diagramą. Ji parodo kaip vartotojai ar aktoriai ir objektai, ir komponentai sąveikauja vykdant *use case* modelį. *sekos* diagramas vėliau naudosime sistemos projektavime. *sekos* diagrama parodo žinučių srautą ir sąryšį nuo vieno objekto prie kito, ir kaip tai derinasi su metodais ar įvykiais palaikomais klasės/objekto.

- Iš schemos matyti, kad pirmame etape vartotojas pasirinktu būdu (WAP, SMS ar GPRS) užklausia sistemą ar aikštelėse yra laisvų stovėjimo vietų.
- Sistema informuoja klientą apie laisvą vietą ir jos numerį.
- Jau po to, kai gautas patvirtinimas apie rezervuotą stovėjimo vietą, vairuotojas gali statyti transporto priemonę ar atvažiuoti vėliau.
- Kliento rezervuota vieta laikoma iki jam pageidaujamo momento.
- Sistema nuskaičiuoja pinigus.
- Jei klientas atlaisvina vietą, sensoriai reaguoja ir praneša į serverį, apie laisvą vietą.
- Sistemos serverio skaitliukas nuolat atnaujina informaciją apie laisvas vietas.
- Visa tai atvaizduojama ekrane.

3.4 Sistemos architektūra

- Naršyklė- tai Internet Explorer programa skirta vartotojui užrezervuoti automobilio stovėjimo vietą.
- Serverinė dalis - tai vartotojo sąsajos neturinti programa, paleista ant kompiuterio, kuri gauna URL užklausas su parametrais, apdoroja jas ir perduoda virtualiai Java mašinai Java klases. Vėliau Java virtuali mašina gražina XML kodą serveriui, kuris perduoda tą kodą vartotojui.
- Komunikavimo įrenginys – Internet Explorer jei rezervuojama vieta iš personalinio kompiuterio; WAP protokolas ar GPRS technologija jei rezervuojama iš telefono ir kišeninio kompiuterio.
- Terminalinė įranga – personalinis kompiuteris, nešiojamas kompiuteris, kišeninis kompiuteris, mobilusis telefonas.



8 pav. Sistemos Architektūra

3.5 Sistemos lygmenys

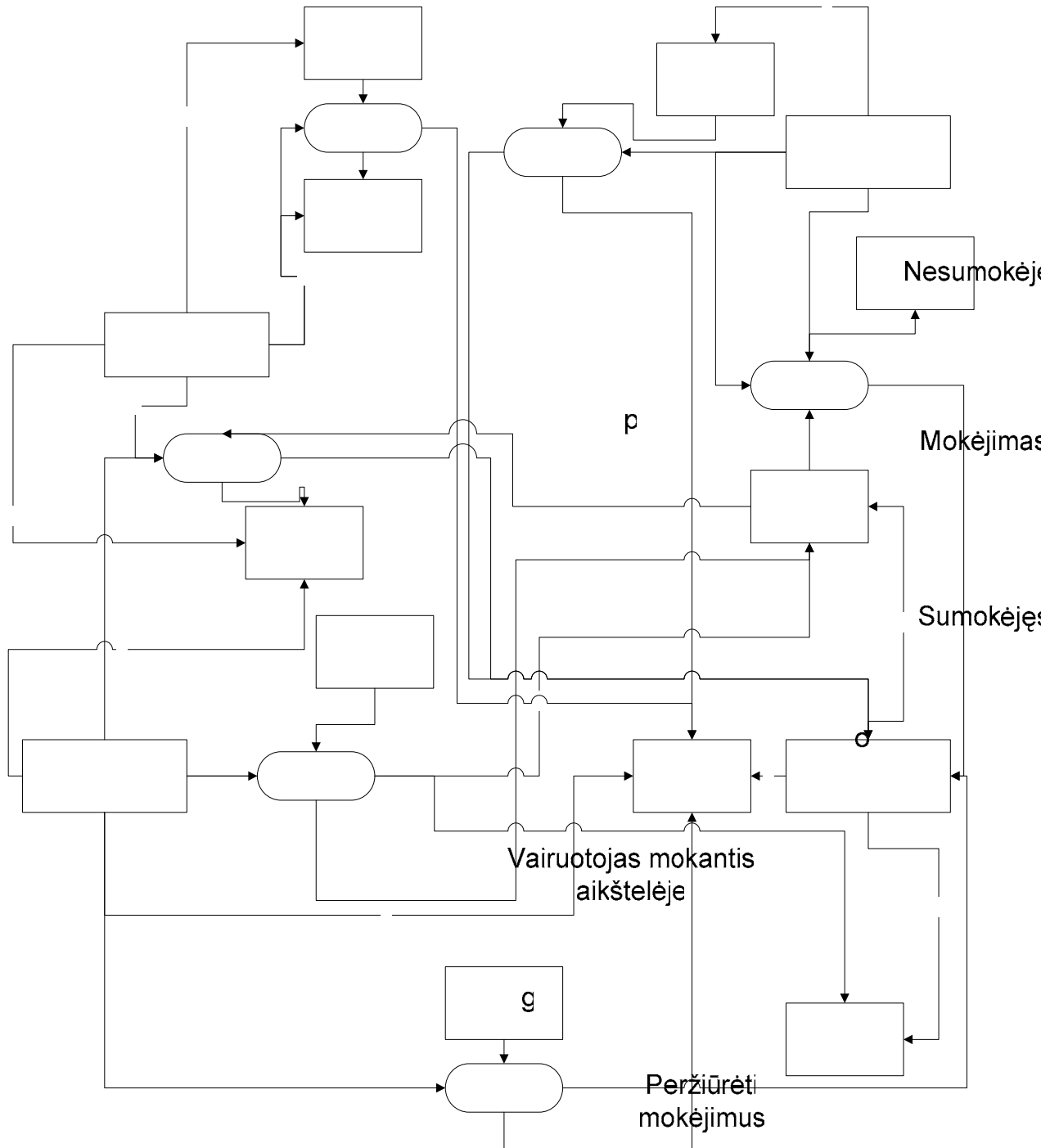
APS projektuojama sistema susideda iš 3 lygmenų:

- Vartotojo/Programinės įrangos lygmuo
- OS lygmuo
- Fizinis lygmuo/ Techninė įranga

Vartotojo/Programinės įrangos lygmuo	OS lygmuo	Fizinis lygmuo/ Techninė įranga
--------------------------------------	-----------	------------------------------------

9 pav. Projektuojamos sistemos lygmenys

3.6 Organizacijos modelis



10 pav. Organizacijos modelis

g

Mūsų sistemos organizacijos modelis, parodo kaip organizuojami procesai vykstantis sistemoje. Vartotojas kreipdamasis į sistemą dalyvauja procese. Jis patenka į organizuotus vartotojus, kur daro rezervaciją. Jis yra tikrinamas ir gauna duomenis, jeigu davė užklausą. Tikrinamas rezervacijos procesas, ar gerai atliko ar atvyko į rezervavimo vietą. Stebimas apmokėjimo procesas. Interneto ar vietoje mokėjimo galimybės. Tikrintojas tikrina vartotojus. Tikrina mokėjimus, rezervacijos procesus, atvaikymą ir t.t.

Susimokėti už stovėjimą

Neatliktas 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.

4.1.2 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.

4.1.3 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.

4.1.4 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.

4.1.5 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 išibrauti 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.

4.1.6 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 kokiai 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 įrankis

Persikirstymo sistemų, kurios naudoja protokolą SOAP, realizavimą nebūtina pradėti nuo nulio. Egzistuoja daug komponentų, kurie palengvina kūrimo procesą ir perima įvairius užklausų iškvietimo ir rezultato grąžinimo 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 iškvietimą 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.

DIME palaikymas SOAP Toolkite

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

17 Lentelė. COM objektai

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 parametras. Kai WSDL generatorius tipų bibliotekoje sutiks tokį parametras, 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.

5 EKSPERIMENTINĖ DALIS

5.1 Sistemos naudojimas

5.1.1 Prisijungimas

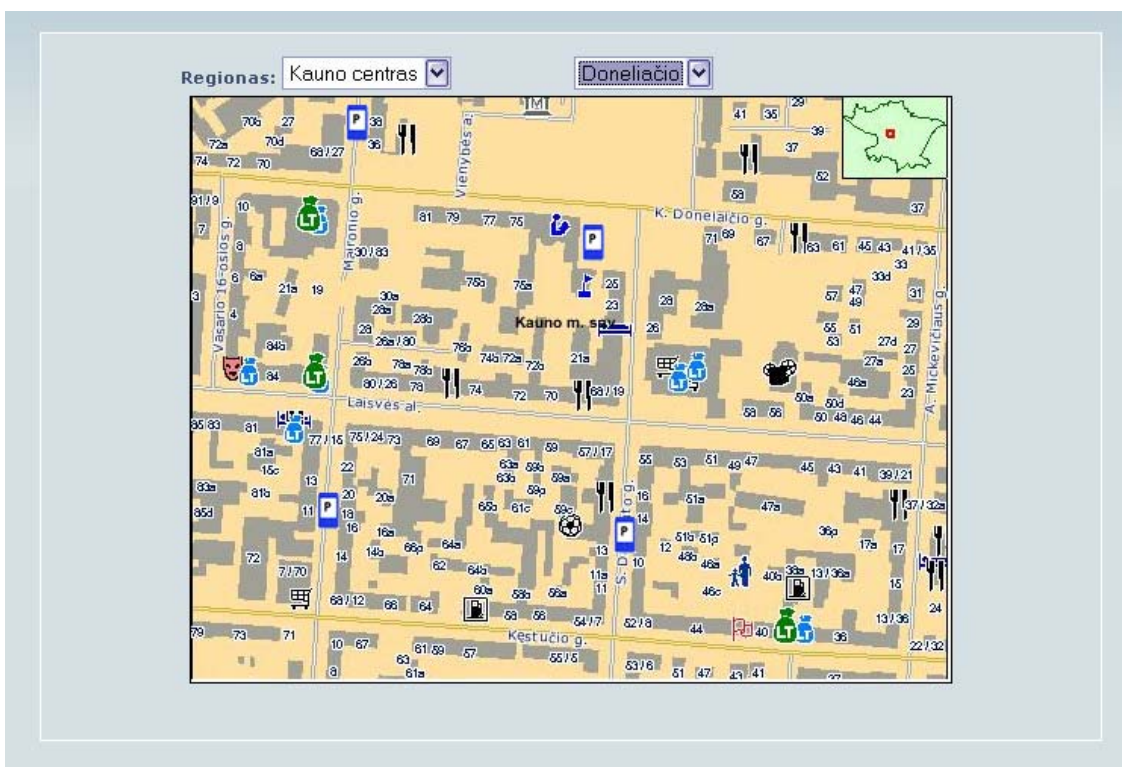
Vartotojas pasirinktu būdu (per WAP arba internetą) siunčia užklausa į sistemą ar aikštelėse yra laisvų stovėjimo vietų, prieš tai identifikuojant aikštelę (pvz. Gedimino gatvė –2). Ši informacija yra nuolat atnaujinama. Svarbu tai, jog norint gauti detalų nuvykimo planą iki stovėjimo vietos reikia prieš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

5.1.2 Rezervavimas

Rezervuojant stovėjimo vietą internetu, pirmiausia žemėlapyje pasirenkama aikštelė (Pav. 11). Jeigu aikštelė yra kitame regione reikia pakeisti jį ir atsiradusiame regione žemėlapyje pasirinkti norimą aikštelę.



11 pav. aikštelių planas

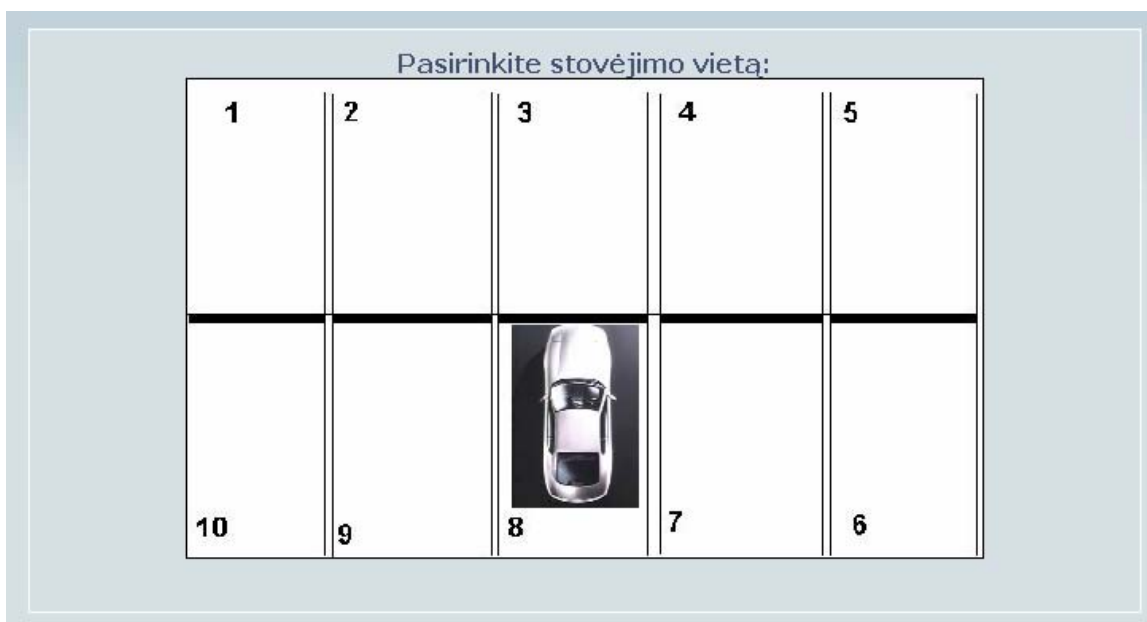
5.1.3 Informacijos įvedimas

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		

12 pav. duomenų įvedimas

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



13 pav. stovėjimo vietos pasirinkimas

Pasirinkus stovėjimo vietą gaunama SMS žinutė su patvirtinimu.

5.1.4 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:



14 pav. WAP puslapis

Gavęs patvirtinimą apie norimą stovėjimo vietą, serveris perduoda duomenys apie užrezervavusią stovėjimo vietą asmeni į stovėjimo aikštelėje esančia sistema.

5.1.5 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 gaus 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.

5.1.6 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 aktoriaus 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.

5.2 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.

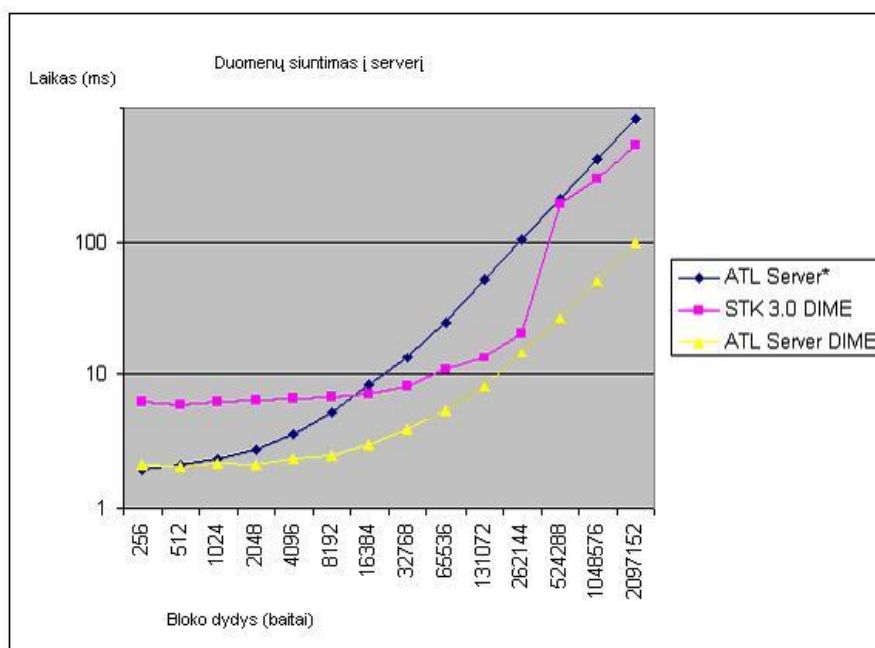
Serverio komponentas SOAP Toolkit, taip pat, kaip ir serverio komponentas ATL Server, perduoda ir gauna iš kliento atitinkamo dydžio duomenų bloką. Taip pat kliento dalis matuoja duomenų išsiuntimo/gavimo iš serverio laiką.

Klientas ir serveris pirmosios konfigūracijos yra identiškąs 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.

18. Lentelė. Duomenų siuntimo į serverį rezultatai

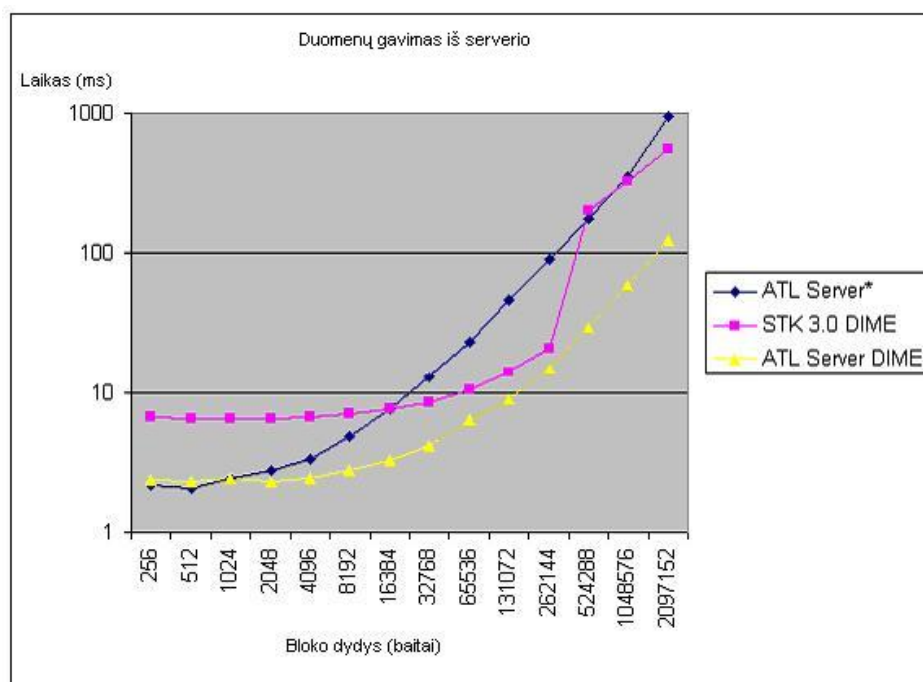
Bloko dydis	1 konfigūracija	2 konfigūracija	3 konfigūracija
256	2,185	6,57	2,345
512	2,03	6,4	2,265
1024	2,425	6,41	2,42
2048	2,735	6,41	2,265
4096	3,355	6,71	2,425
8192	4,845	7,04	2,73
16384	7,58	7,5	3,205
32768	12,81	8,43	4,14
65536	23,36	10,47	6,25
131072	46,485	14,06	8,905
262144	91,405	21,1	14,845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



15pav. Duomenų siuntimas į serverį

19. lentelė. 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



16. 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 servisą, 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 pirmuoja 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).

Išvados

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.

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, Erikas Pliaukšta realizavo ir analizavo metodą, sukurtą su ATL 7.0 biblioteka su įdegtų DIME formato palaikymu.

Išanalizuoti 4 testavimo metodai ir išrinktas vienas tinkamiausias mūsų valdymo sistemai testuoti. Sėkmingai atliktas automobilių stovėjimo aikštelių daugiafunkcinės valdymo sistemos testavimas, kuriuo metu 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 šioje 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 naudojam 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.
- 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.

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. Airclic, Inc., parkingo sistema. URL: <http://www.aircllic.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)

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

Kelios SOAP realizacijos:

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++)

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