

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Rytis Rudelis

**Paskirstytos sistemos projektavimas ir testavimo
metodikos tyrimas**

Magistro darbas

Darbo vadovas

prof. Eduardas Bareiša

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Rytis Rudelis

**Paskirstytos sistemos projektavimas ir testavimo
metodikos tyrimas**

Magistro darbas

Vadovas

Prof. Eduardas Bareiša
2008-05

Recenzentas

doc. dr. A. Lenkevičius

2006-05

Atliko

2008-05-23

IFM-2/2 gr. stud.
Rytis Rudelis

Kaunas, 2008

Turinys

1.	Įvadas.....	9
2.	Analitinė dalis.....	10
2.1.	Problema.....	10
2.2.	Panašios sistemos.....	10
2.2.1	Sistema TomTom Navigator.....	10
2.2.2	Sistema LocTracker.....	11
2.2.3	Sistema Locator.....	11
2.2.4	Egzistuojančių sistemų apibendrinimas.....	12
2.3.	Projektavimo metodika.....	12
2.3.1	Pozicionavimo technologijos.....	13
2.3.2	Architektūros pasirinkimas.....	14
2.3.3	Vartotojo pozicijos koordinačių nuskaitymas.....	15
2.3.4	Vartotojo pozicijos koordinačių transformavimas.....	16
2.3.5	Žemėlapių fragmentų koordinačių išskaičiavimas.....	16
2.3.6	REST panaudojimas.....	19
2.3.7	Apribojimai sprendimui.....	21
2.4.	Paskirstytų sistemų testavimas.....	22
2.5.	Užduotis projektavimui.....	24
2.6.	Analizės išvados.....	24
3.	Sistemos prototipas.....	25
3.1.	Sistemos reikalavimai.....	25
3.1.1	Sistemos tikslai.....	25
3.1.2	Vartotojai.....	25
3.1.3	Diegimo aplinka.....	26
3.1.4	Veiklos kontekstas.....	26
3.1.5	Sistemos panaudos atvejai.....	28
3.1.6	Funkciniai reikalavimai.....	33
3.1.7	Nefunkciniai reikalavimai.....	34
3.2.	Sistemos architektūra.....	35
3.2.1	Sistemos veikimo principas.....	35
3.2.2	Bendroji architektūra.....	36

3.2.3	Duomenų bazė	38
3.3.	Testavimas	41
3.3.1	Testavimo tikslai ir apimtis.....	41
3.3.2	Pagrindiniai apribojimai	41
3.3.3	Sąsajos testavimas.....	42
3.3.4	Testavimo strategijos	42
3.3.5	Vienetų testavimas	43
3.3.6	Integravimo testavimas	43
3.3.7	Priėmimo testavimas.....	45
3.3.8	Aukšto lygio testavimas.....	45
3.3.9	Testavimo rezultatų kaupimas	47
3.3.10	Testavimo rezultatai.....	47
3.4.	Kokybės analizė.....	51
3.4.1	Kokybės vertinimo tikslai.....	51
3.4.2	Kokybės vertinimo kriterijai.....	51
3.4.3	Kokybės užtikrinimo ir vertinimo procesas.....	52
3.4.4	Kokybės vertinimas	53
3.4.5	Kokybės vertinimo rezultatai.....	55
3.4.6	Išvados	56
4.	Tiriamoji eksperimentinė dalis	57
4.1.	Tikslas.....	57
4.2.	Tyrimo metodika.....	57
4.3.	Priemonės.....	58
4.4.	Rezultatai	58
4.5.	Išvados	61
5.	Išvados	62
6.	Literatūra.....	63
7.	Terminų žodynas.....	64
8.	Priedai	65

Paveikslai

1 pav. Bendrinė sistemos architektūros schema	14
2 pav. <i>Quadtree</i> žemėlapių struktūra	17
3 pav. Žemėlapių fragmentas	17
4 pav. Mercator projekcijos deformacijos	18
5 pav. Evoliucinis PĮ kūrimo modelis	23
6 pav. Veiklos konteksto diagrama	26
7 pav. Vartotojo panaudos atvejų diagrama (WEB srityje)	28
8 pav. Sistemos administratoriaus panaudos atvejų diagrama (WEB srityje)	30
9 pav. Vartotojo panaudos atvejų diagrama (klientinėje PĮ dalyje)	31
10 pav. Sistemos veikimo principas	36
11 pav. Bendroji sistemos architektūra	37
12 pav. Duomenų bazės schema	38
13 pav. WEB portalo integracinio testavimo schema	43
14 pav. Integracijos testavimo modulių sąryšių schema	44
15 pav. Klientinės PĮ integracinio testavimo schema	44
16 pav. Testavimo metu aptiktų ir išanalizuotų klaidų saugojimo failo pavyzdys	47

Lentelės

1 lentelė. GSM metodų tikslumai	13
2 lentelė. Veiklos padalinimas	27
3 lentelė. Duomenų bazės lentelės "USER" specifikacija	39
4 lentelė. Duomenų bazės lentelės "REQUESTING_USER" specifikacija	39
5 lentelė. Duomenų bazės lentelės "USER_ROUTE" specifikacija	40
6 lentelė. Duomenų bazės lentelės "USER_OBSERVER" specifikacija	40
7 lentelė. Testavimo rezultatų žurnalo pavyzdys	47
8 lentelė. Vienetų testavimo rezultatai	47
9 lentelė. REST serviso testavimo rezultatai	48
10 lentelė. Kokybės vertinimo rezultatai	55
11 lentelė. Aritmetinio algoritmo vykdymo eksperimento rezultatai	58
12 lentelė. Ciklo vykdymo eksperimento rezultatai	59
13 lentelė. Teksto apdorojimo eksperimento rezultatai	59
14 lentelė. Loginių operacijų vykdymo eksperimento rezultatų lentelė	60
15 lentelė. Failo nuskaitymo eksperimento rezultatai	60

Rudelis R. Distributed system design and testing methodology analysis: Master's work in science of informatics/ supervisor Prof. Dr. E. Bareiša; Department of Software Engineering, Faculty of Informatics, Kaunas University of Technology. – Kaunas, 2008. = 70 p

SUMMARY

Many different positioning systems for ordinary IT users were developed since GPS technology became available not only to US military. Those systems led to new kind of services – location-based services. Some of the widely used positioning systems are analyzed in this work and their main functionality is marked out, positioning technologies compared. Architecture of multi-user positioning system and its main implementations aspects are given. Also there is an analysis of distributed system development and testing issues that may lead to project failure or its cost rising and a software development and testing model suggested. This is a practical model which lets reduce the risk of failure.

Santrauka

GPS technologijai tapus prieinama ne tik karinei pramonei buvo pradėtos kurti įvairios paskirties pozicionavimo sistemos, skirtos eiliniams IT vartotojams. To pasėkoje atsirado naujo pobūdžio paslaugų rūšis – paslaugos pagrįstos vartotojo pozicijos nustatymu. Darbe išanalizuotos plačiai taikomos pozicionavimo sistemos ir išskirtas pagrindinis tokių sistemų funkcionalumas, palygintos pozicionavimo technologijos, jų privalumai ir trūkumai viena kitos atžvilgiu. Darbe aprašomas projektinis sprendimas, panaudotas kuriant daugiavartotojišką pozicionavimo sistemą, kurios architektūra ir svarbiausi realizacijos aspektai pateikiami šiame darbe. Išskirtos dažnai praktikoje pasitaikančios paskirstytų sistemų projektavimo, realizavimo ir testavimo klaidos, dėl kurių gali nepavykti sėkmingai ir laiku įgyvendinti projektą arba kritinės sistemos dalys gali būti netinkamos naudoti realiuose įrenginiuose. Pasiūlytas būdas, kaip išvengti tokių klaidų ir sumažinti riziką, kad projektas gali nepavykti. Eksperimentu metu įrodyta, jog paskirstytų sistemų klientų, skirtų mobiliems įrenginiams (mobiliems telefonams ir PDA), kūrimo ir testavimo procesas neturi apsiriboti vien tik emuliatoriais, kuriuose vykdomos programos elgesys ir charakteristikos gali labai skirtis nuo realiame įrenginyje vykdomos programos.

1. Įvadas

Pozicionavimo sistemos, kurios buvo kuriamos karinei pramonei, šiuo metu tampa vis plačiau naudojamos įvairiose kasdienėse gyvenimo srityse. Iš pradžių GPS paremtos sistemos buvo pradėtos naudoti laivyboje, aviacijoje, transporto logistikoje, o šiuo metu vis plačiau jos naudojamos ir privačiame sektoriuje. Vis labiau populiarėja automobilinės navigacijos sistemos, GPS naudojančios automobilinės signalizacijos bei kitos panašaus pobūdžio sistemos, kurios yra orientuotos į eilinių vartotoją.

Kita sparčiai populiarėjanti technologijų atmaina – mobilūs įrenginiai, tokie kaip telefonai, nešiojami kompiuteriai ir delniniai kompiuteriai (PDA). Didelę įtaką tam turi šių įrenginių spartus tobulėjimas ir jų funkcionalumo didėjimas. Nešiojami kompiuteriai jau nieko nesiskiria nuo standartinių stacionarių kompiuterių, telefonuose diegiamos operacinės sistemos, o delniniais kompiuteriais eiliniai vartotojai gali atlikti daugumą jiems būtinų funkcijų, kurias jie atlieka su stacionariais kompiuteriais. Tam nemažai įtakos turi ir vis plačiau prieinamos mobilaus interneto paslaugos. Tai leidžia praplėsti egzistuojančių sistemų galimybes ir padaryti jas prieinamas iš mobilių įrenginių bei kurti specializuotas paskirstytas sistemas, orientuotas į mobilius įrenginius.

Visa tai lėmė naujo pobūdžio paslaugas ir servisus teikiančių programų atsiradimą. Šios programos yra pagrįstos vartotojo pozicijos nustatymu (angl. „location-based services“ - LBS). Tokio pobūdžio programos apima pagalbos paslaugas, navigaciją, turistinių kelionių planavimus ir t.t. [12]

Šio darbo tikslas: sukurti daugiavartotojišką pozicionavimo sistemą, paremtą pozicionavimo paslaugų (LBS) principais ir realizuojančią pagrindines tokio pobūdžio sistemų funkcijas, bei paanalizuoti paskirstytų sistemų testavimo problematiką panaudojant sukurtą sistemą.

Šiame darbe pristatomas sprendimas yra sukurtas panaudojant paskirstytos sistemos principus, kliento serverio architektūrą ir realizuotas naudojant J2ME ir Java Servlet technologiją (Tapestry karkasas).

2. Analitinė dalis

2.1. Problema

Didžioji dalis eiliniams vartotojams prieinamų pozicionavimo sistemų dažniausiai yra orientuotos į vieną vartotoją. Jos leidžia vartotojui nustatyti savo buvimo vietą, atvaizduoti ją žemėlapyje, kurti ir saugoti maršrutus. Tokios sistemos dažniausiai naudojamos kelionėse kaip navigacijos priemonė.

Kita rūšis pozicionavimo sistemų leidžia nustatyti ne tik savo, bet ir kitų objektų pozicijas. Tokios sistemos dažniausiai yra orientuotos į logistikos įmones. Jos leidžia sekti daug objektų vienu metu. Tačiau šios sistemos dažnai reikalauja specialios įrangos naudojimo ir yra pakankamai brangios.

Šiame darbe aprašoma daugiavartotojiška pozicionavimo sistema yra tarpinis variantas. Ji skirta eiliniams vartotojams, tačiau leidžia stebėti ne tik savo, bet ir kitų vartotojų pozicijas. Sistema teikia įprastoms asmeninės navigacijos sistemoms būdingą standartinį funkcionalumą bei papildo jį logistikoje naudojamų sistemų galimybėmis. Tai leidžia vartotojams stebėti vienas kito poziciją realiu laiku ir keistis išsaugotais maršrutais.

2.2. Panašios sistemos

Rinkoje yra siūloma daug įvairių pozicionavimo sistemų. Šiame skyriuje bus apžvelgtos kelios tokio pobūdžio sistemos ir jų teikiamas funkcionalumas.

2.2.1 Sistema TomTom Navigator

Tai komercinė navigacijos sistema, sukurta ir palaikoma kompanijos TomTom. Vartotojo pozicijos identifikacijai sistema naudoja GPS technologiją. Sistema parduodama su specialia technine įranga arba gali būti instaliuojama į delninį kompiuterį.

Sistema vartotojui teikia daug įvairių funkcijų. Naudodamas šią sistemą vartotojas gali planuoti keliones, sudarinėti ir saugoti norimus maršrutus. Sistema leidžia atlikti kelionės tikslo paiešką pagal pašto kodą, miesto gatvę, sankryžą, namo numerį, lankytiną vietą ir t.t.¹ Už papildomą mokestį vartotojai gali užsisakyti ir papildomas paslaugas, tokias kaip informacija apie eismo srautus ir pan.

¹ <http://www.tomtom.com/products/features.php?ID=260&Category=2&Lid=1>

2.2.2 Sistema LocTracker

LocTracker² – tai objektų buvimo vietos ir judėjimo stebėjimo paslauga, leidžianti operatyviai ir tiksliai sužinoti transporto priemonių, mobiliųjų darbuotojų ar kitų judančių objektų buvimo vietą. Sistema naudoja vietos nustatymo įrenginių gamybos lyderio Trimble įrenginį TrimTrac.

Informaciją apie stebimo objekto buvimo vietą sistema pateikia internetinėje LocTracker svetainėje kas 1 minutę. Taip pat sistema leidžia peržiūrėti vietos nustatymų istoriją, gauti ataskaitas apie nuvažiuotą atstumą ir vidutinį važiavimo greitį. Sistemoje numatyta galimybė leisti partneriams stebėti tam tikrus įrenginius tam tikru metu³.

Sistema yra skirta transporto, statybos, techninių, kurjerinių, apsaugos ar projektavimo paslaugų bendrovėms bei įmonėms, turinčioms mobilius pardavimų vadybininkus, nuomojančioms ir pervežančioms vertingus daiktus.

2.2.3 Sistema Locator

Locator⁴ – tai vietos nustatymo GSM tinkle technologiją naudojanti sistema. Ji leidžia nustatyti mobilaus telefono buvimo vietą. Lietuvoje ir Latvijoje GSM/3G telefonų vietos nustatymo tikslumas miestuose – nuo 50 m patalpose iki 2 km lauke, mažesniuose miesteliuose – 5-10 km.

Sistema informaciją apie nustatytą abonentų buvimo vietą gali pateikti interneto svetainėje, WAP naršyklėje, MMS arba SMS žinute. Kai abonentas nepasiekiamas, sistema pateikia paskutinę žinomą jo buvimo vietą. Taip pat sistema leidžia periodinį telefono vietos nustatymą, kai vartotojas pasirenka vietos nustatymo periodą, dienas, valandas ir vietos nustatymo dažnumą. Vartotojai taip pat gali atlikti istorinių ir periodinių buvimo vietos nustatymų peržiūrą.

² <http://www.locator.lt/loctracker/>

³ <https://www.locator.lt/locator/?show>

⁴ https://www.locator.lt/locator/form/lt_bite_about_more.jsp

2.2.4 Egzistuojančių sistemų apibendrinimas

Išanalizavus plačiai naudojamą pozicionavimo sistemas, galima išskirti pagrindinį pozicionavimo sistemų teikiamą funkcionalumą:

- objekto pozicijos nustatymas,
- objekto pozicijos atvaizdavimas žemėlapyje,
- kelionės maršruto sudarymas,
- maršruto išsaugojimas,
- maršruto peržiūra,
- kitos funkcijos.

Tokių sistemos funkcionalumą galima realizuoti naudojant paprastą sistemos architektūrą. Tačiau plečiantis sistemos funkcionalumui, sistema neišvengiamai tampa paskirstyta. Šiuo atveju sistemoje atsiranda papildomi komponentai – serveriai, kurie leidžia realizuoti centralizuotą sistemos valdymą ir priežiūrą bei užtikrinti komunikaciją tarp sistemos vartotojų.

2.3. Projektavimo metodika

Kuriama daugiavartotojiška pozicionavimo sistema savyje turi apjungti į eilinį vartotoją orientuotų pozicionavimo sistemų funkcionalumą su logistikos įmonių naudojamų pozicionavimo sistemų funkcionalumu.

Išanalizavus tokio pobūdžio pozicionavimo sistemas, buvo išskirtas pagrindinis sistemos funkcionalumas, kuris apima:

- savo pozicijos nustatymą,
- pozicijos atvaizdavimą žemėlapyje,
- maršruto fiksavimą, išsaugojimą ir atvaizdavimą.
- galimybę stebėti kitų vartotojų poziciją,
- galimybę keisti išsaugotais maršrutais.

2.3.1 Pozicionavimo technologijos

Pozicijos nustatymui pozicionavimo sistemose dažniausiai yra naudojamos dviejų tipų technologijos: vietos nustatymas GSM tinkle ir globalinė pozicionavimo sistema - GPS.

Vietos nustatymas GSM tinkle yra naudojamas mobilaus ryšio operatorių perjungiant telefonus tarp aptarnaujančių stočių. Šio atveju gali būti naudojama keletas pozicijos nustatymo metodų:

- 1) artimiausios bazinės stoties nustatymas (Cell identification – Cell-ID),
- 2) signalo atėjimo kampo metodas (Angle of Arrival – AOA),
- 3) patobulintas laikų skirtumo metodas (Enhanced Observed Time Difference – E-OTD),
- 4) radijo signalo atėjimo laiko metodas (Time of Arrival – TAO) ir kt.

Pagrindinis šios technologijos privalumas yra tas, kad ji veikia ne tik lauko sąlygomis, bet ir pastatų viduje. Tačiau šios technologijos trūkumas – priklausomybė nuo GSM tinklo struktūros ir aptarnaujančių stočių išdėstymo tinklo. Tai lemia pakankamai netikslūs duomenis apie objekto poziciją. Metodų tikslumai ir vėlinimas pateikiami lentelėje žemiau [7]:

1 lentelė. GSM metodų tikslumai

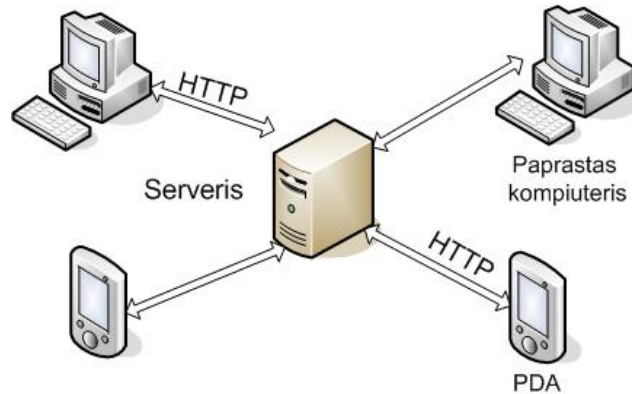
Technologija	Pastatuose	Mieste	Priemiesčiuose	Užmiestyje	Vėlinimas
Cell-ID	1-35 m	0,1-1 km	1-10 km	1-35 km	< 5 sek.
Cell-ID+TA	100-500 m	100-500 m	550 m	550 m	< 5 sek.
E-CGI	0,05-8 km	50-550 m	0,25-5 km	0,25-8 km	< 5 sek.
AOA	Netikslinga	>125 m	>125 m	>125 m	~ 10 sek.
U-TAO	25-100 m	85-100 m	30-75 m	25-75 m	~ 10 sek.
E-OTD	50-150 m	50-150 m	50-150 m	50-150 m	4-10 sek.

Globalinė pozicionavimo sistema (GPS) – tai palydovinė radijo navigacinė sistema. Šią sistemą sukūrė ir prižiūri JAV Gynybos Departamentas. Šiuo metu tai plačiausiai taikoma pozicionavimo technologija. Ji dengia visą Žemės paviršių, gali pateikti trimates koordinates, nustatyti objekto judėjimo kryptį ir greitį. Skirtingai nuo GSM technologijos GPS imtuvai veikia tik lauko sąlygomis. Tačiau ši technologija leidžia žymiai tiksliau nustatyti objekto pozicijos koordinates: standartinis GPS imtuvas gali pateikti pozicijos koordinates 3 – 15 m tikslumu [9].

Atsižvelgiant į sistemai keliamus reikalavimus ir abiejų technologijų privalumus ir trūkumus viena kitos atžvilgiu, buvo nuspręsta pozicijos nustatymui naudoti GPS technologija.

2.3.2 Architektūros pasirinkimas

Sistemoje numatyta galimybė vartotojams stebėti vieni kitus bei keisti maršrutais yra realizuojama serverio pagalba. Bendrinę sistemos architektūros schemą galima pavaizduoti taip:



1 pav. Bendrinė sistemos architektūros schema

Kaip matyti iš schemos, serveris yra centrinis sistemos mazgas. Jis leidžia:

- ✓ autentifikuoti ir autorizuoti vartotojus,
- ✓ vykdyti duomenų mainus tarp sistemos vartotojų,
- ✓ centralizuotai saugoti ir administruoti duomenis,
- ✓ centralizuotai valdyti ir administruoti sistemą.

Taikant tokią architektūrą, vartotojams yra prieinamos dviejų tipų prieigos:

- Iš paprasto kompiuterio – tai internetinis sistemos portalas, kurį vartotojas gali pasiekti naudodamas standartinės interneto naršykles. Naudodamasis šia prieiga, vartotojas turi galimybę užsiregistruoti sistemoje bei administruoti savo duomenis. Ši prieiga taip pat yra skirta sistemos administravimui.
- Iš delninio kompiuterio – tai instaliuojamas sistemos klientinis modulis (toliau – *klientas*). Naudodamasis šiuo moduliui, vartotojas gali ne tik administruoti savo duomenis, bet ir stebėti savo ir kitų vartotojų pozicijas realiu laiku bei registruoti savo maršrutus.

2.3.3 Vartotojo pozicijos koordinačių nuskaitymas

Savo ir kitų vartotojų pozicijai stebėti, vartotojas naudojami delniniu kompiuteriu, kuriame įdiegtas klientas. Šis klientas vartotojo pozicijos koordinates nuskaitymo iš kompiuteryje esančio GPS įrenginio per nuosekliosios sąsajos jungtį (COM port). Šis GPS įrenginys, kaip ir daugelis kitų, naudoja labiausiai paplitusį NMEA 0183 (<http://www.nmea.org/pub/0183/>) komunikacijos standartą. Taigi, vartotojui keičiant savo buvimo vietą atviroje aplinkoje, GPS įrenginys generuoja NMEA standarto apibrėžtus, specialios struktūros sakinius, kuriuose yra pateikiamos pozicijos koordinatės ir kita susijusi informacija. Papildoma informacija apima įvairius duomenis apie įrenginio aptiktus palydovus, aukštį virš jūros lygio ir pan.

Keletas NMEA sakinių pavyzdžių [3]:

- \$GPGLL,4916.45,N,12311.12,W,225444,A,*31
- \$GPGSV,2,1,08,01,40,083,46,02,17,308,41,12,07,344,39,14,22,228,45*75
- \$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A
- \$GPVTG,054.7,T,034.4,M,005.5,N,010.2,K*33
- \$GPAAM,A,A,0.10,N,WPTNME*43
- \$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47
- \$GPBOD,045.,T,023.,M,DEST,START*01

Iš tiesų, vidutiniškai tik nedidelis kiekis įrenginio sugeneruotu NMEA sakinių saugo pozicijos koordinates. Klientas šias koordinates paima tik iš trijų tipų sakinių, kurių pavyzdžiai pateikti žemiau:

- \$GPGLL,4916.45,N,12311.12,W,225444,A,*31
- \$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A
- \$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

Pabraukti sakiniuose esantys fragmentai yra reikiamos ilgumos ir platumos koordinatės.

Internete yra nemažai NMEA 0183 sakinių analizės atviro kodo programinių bibliotekų, tačiau šiuo atveju reikalingų sakinių kiekis yra labai mažas, todėl sistemos prototipe realizuotas efektyvus specializuotas šių sakinių programinis analizės modulis. Šis modulis išskiria reikiamas koordinates ir atiduoda jas kitiems moduliams, naudojantiems šias koordinates.

2.3.4 Vartotojo pozicijos koordinačių transformavimas

Iš GPS įrenginio nuskaitytos ir iš NMEA sakinių paimtos vartotojo pozicijos ilgumos ir platumos koordinatės priklauso geografinėi koordinačių sistemai, kuri yra sferinės koordinačių sistemos atmaina [8]. Ilgumos ir platumos koordinatės yra apriboti geometriniai kampai: ilgumos koordinatės kitimo režiai nuo -180° iki 180° , o platumos nuo -90° iki 90° . Sistemos prototipe šios koordinatės naudojamos reikiamam žemėlapių fragmentui išskaičiuoti ir ant šio fragmento atvaizduoti vartotojo pozicijos tašką. Naudojami žemėlapiai, pasiekiami internetinėje žemėlapių saugykloje, yra sudaryti cilindrinės Mercator [15] projekcijos pagrindu. Išskleidus šį cilindrą yra gaunamas viso pasaulio žemėlapis stačiakampėje koordinačių sistemoje. Natūraliai išskyla koordinačių sistemų nesuderinamumo problema, kurią galima išspręsti transformuojant geografines pozicijos koordinates į stačiakampę koordinačių sistemą. Tokią transformaciją galima atlikti panaudojant atvirkštinę Gudermano (*Gudermannian*) funkciją [16]:

$$y = gd^{-1}(\varphi) = \frac{1}{2} \ln \frac{1 + \sin(\varphi)}{1 - \sin(\varphi)}, \text{ kur } \varphi - \text{platumos koordinatė.}$$

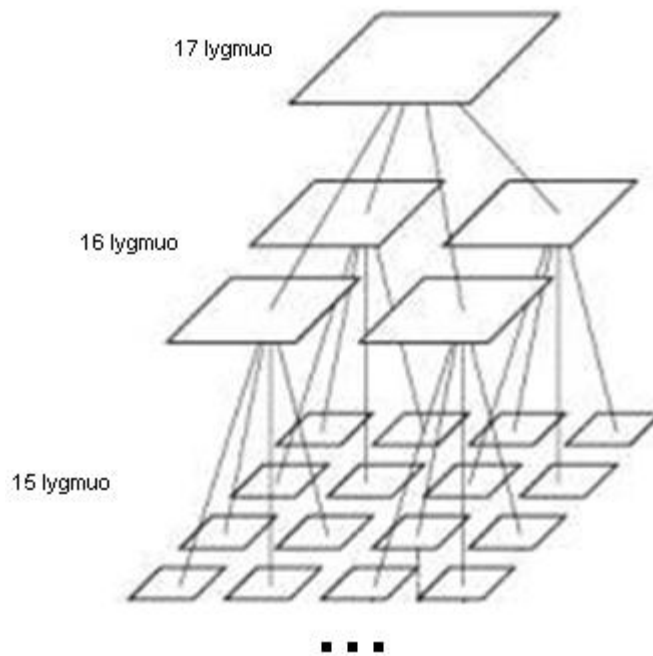
Kaip matyti, transformacija yra reikalinga tik platumos koordinatei. Po šios transformacijos pozicijos koordinatės yra suderinamos su žemėlapių koordinačių sistema.

2.3.5 Žemėlapių fragmentų koordinačių išskaičiavimas

Objekto pozicijos atvaizdavimui kliente reikalingi žemėlapiai. Turint objekto pozicijos koordinates, reikiamų žemėlapių fragmentų pirmiausiai ieškoma lokaliame žemėlapių keše. Jei reikiamo fragmento keše nėra, jis imamas iš internetu pasiekiamos žemėlapių saugyklos. Žemėlapių saugykla žemėlapių fragmentus pateikia pagal jų specifines koordinates, todėl klientas turi mokėti išskaičiuoti reikiamo fragmento koordinates.

Saugykloje žemėlapiai saugomi kvadratinių, 256×256 taškų dydžio, PNG (<http://www.libpng.org/pub/png/>) formato fragmentų pavidalu, o pilnas iš šių fragmentų sudarytas žemėlapis paremtas *Quadtree* struktūros pagrindu.

Quadtree struktūra yra medis, kurio kiekviena tėvinė viršūnė turi iki keturių žemesnio lygmens viršūnių. Ši struktūra dažniausiai taikoma rekursiniam dviejų dimensijų plokštumų sudalinimui į keturis kvadrantus, kas konkrečiu atveju yra pritaikyta naudojamiems žemėlapiams. Pilnas žemėlapis turi 18 išdidinimo lygmenų (0..17), kas atsispindi 2 pav. iliustruotoje *Quadtree* struktūroje. 17 lygmuo yra labiausiai nutolintas žemėlapių fragmentas ir šiuo atveju atitinka visą pasaulį.



2 pav. *Quadtree* žemėlapių struktūra

Pastebėsime, jog iliustracijoje pateikti fragmentai mažėja su kiekvienu lygmeniu gilyn, tačiau taip pavaizduota tik vaizdumo sumetimais ir realybėje visi jie yra vienodo – 256x256 pikselių dydžio.

Toliau trumpai nusakysime minėtos internetinės saugyklos žemėlapių fragmentų koordinatinių išskaičiavimą.

Išskaičiavimas pradedamas nuo pradinio fragmento X_0 , kuris atitinka visą pasaulį ir yra $z_0=17$ priartinimo lygmens. Šis fragmentas yra menamai dalinamas į keturias lygias dalis (3 pav.) ir, pagal turimas pozicijos koordinatas, parenkama viena reikiama dalis. Naujai parinktos dalies atžvilgiu veiksmai yra kartojami tol, kol pasiekiamas reikiamo priartinimo lygio reikiamas fragmentas. Kitaip tariant, einama gilyn per *Quadtree* struktūros lygmenis ir kaskart parenkamas reikiamas, vienas iš keturių, fragmentas.



3 pav. Žemėlapių fragmentas

Matematiškai pirmos iteracijos kiekvieno galimo naujai parenkamo fragmento išskaičiavimą galima apibrėžti taip:

$$\begin{aligned} X_1(x_1; y_1; z_1) &= X_0(2x_0; 2y_0; z_0 - 1) & X_3(x_3; y_3; z_3) &= X_0(2x_0; 2y_0 + 1; z_0 - 1) \\ X_2(x_2; y_2; z_2) &= X_0(2x_0 + 1; 2y_0; z_0 - 1) & X_4(x_4; y_4; z_4) &= X_0(2x_0 + 1; 2y_0 + 1; z_0 - 1) \end{aligned}$$

Kur $X_i(x_i; y_i; z_i)$, $i = \overline{1,4}$ yra 3 pav. pateiktas i -tasis fragmentas: x , y koordinatės ir z – priartinimo lygis.

Kaip ir buvo minėta, saugykloje saugomi žemėlapiai yra sudaryti Mercator projekcijos pagrindu. Ši projekcija yra plačiai taikoma ir gerai žinoma geografijos srities specialistams, tačiau turi esminį trūkumą – stipriai iškraipo žemėlapių mastelį tolstant nuo pusiaujo link ašigalių [2]. Kaip pavyzdį galima palyginti Afriką su Grenlandija, kurios Mercator projekcijoje yra beveik tokio paties dydžio, tačiau realybėje Grenlandija yra beveik 14 kartų mažesnė už Afriką.



4 pav. Mercator projekcijos deformacijos

(Paimta iš http://en.wikipedia.org/wiki/Mercator_projection)

Nepaisant šio trūkumo, ši projekcija yra tinkama interaktyvių internetinių žemėlapių sudarymui, kuriais patogiau manipuliuoti, todėl neretai yra taikoma geografinėse informacinėse sistemose (GIS) ir puikiai tinka realizuoto sistemos prototipo poreikiams.

2.3.6 REST panaudojimas

REST (*REpresentational State Transfer*) – yra tam tikra programinė architektūra, skirta paskirstytoms hiperžiniasklaidos sistemoms, tokioms kaip žiniatinklis. Terminas pirmą kartą paminėtas 2000 metais vieno iš pagrindinių HTTP (*Hypertext Transfer Protocol*) specifikacijos autorių Roy Fielding disertacijoje [4]. Yra suformuluoti pagrindiniai principai, kuriais turėtų remtis REST architektūrą įgyvendinančios programinės realizacijos [13][14].

Sparčiai plintančius REST pagrindu sudarytus servिसus galima laikyti tam tikra SOAP (*Simple Object Access Protocol*) Web servिसų alternatyva. Pastarieji laikui bėgant tapo sudėtingi ir neefektyvūs, o paprastumu ir efektyvumu pasižymintys REST servिसai vis labiau plinta [5]. Tačiau šiandien REST ideologija dar nėra apibrėžta kokio nors standarto ir neretai kyla debatai dėl šios architektūros ir termino panaudojimo. REST terminas dažnai yra naudojamas laisvesne prasme, kalbant apie paprastą sąsają, kuri skirta perduoti specifinius srities duomenis tiesiogiai panaudojant HTTP protokolą ir nenaudojant jokio papildomo pranešimų sluoksnio kaip SOAP. Internete yra daug šaltinių, kurie skelbiasi savo sistemose naudojančios REST architektūrą, tačiau daugelis iš jų yra tik tam tikros jos atmainos, įgyvendinančios tik dalį Roy Fielding suformuluotų REST principų.

Prototipo sistemos REST servिसas pagrįstas labai paprastu principu. Jis naudoja HTTP protokolą ir neturi jokio papildomo pranešimų perdavimo sluoksnio. Pranešimai yra perduodami tiesiog per HTTP parametrus. Kliento siunčiamas pranešimas, arba tiesiog HTTP užklausa, išskviečia tam tikrą REST serviso metodą, kuris gali turėti parametrus arba ne. Tokios užklaustos rezultatas yra HTTP atsakas – struktūrizuotas XML (*eXtensible Markup Language*) srautas. Komunikuojant klientui ir serveriui yra būtina identifikuoti, kuris vartotojas vykdo užklausa, todėl yra būtinas identifikacijos mechanizmas. Šiuo atveju jis realizuotas ad-hoc principu. Pirmą kartą jungiantis vartotojui yra vykdoma autentifikacija ir grąžinamas SHA-1 (<http://www.itl.nist.gov/fipspubs/fip180-1.htm>) algoritmu sugeneruotas kodas. Šis kodas naudojamas kaip sesijos identifikatorius prisijungusiam vartotojui identifikuoti ir yra pridėdamas prie užklaustos parametrų kiekvieną kartą klientui kviečiant kitus REST serviso metodus.

Prisijungimo užklaustos pavyzdys panaudojant HTTP GET metodą:

<http://myserver.com/RETSERVICE?method=login&username=myUsername&password=myPassword>

Tokios užklauskos galimas atsakas XML formatu atrodo taip:

```
<?xml version="1.0" encoding="utf-8" ?>
<root>
  <data>true</data>
  <sessionToken>465bf07ec6e9d588ecfdfff0e5e175b266f45a00</sessionToken>
</root>
```

Roy Fielding REST ideologijoje serveryje neturėtų būti saugoma jokia būsena įskaitant ir sesijos duomenis. Šiuo atveju autentifikacija yra būtina, todėl realizuotas REST servisas yra tik minėto autoriaus architektūros atmaina.

REST principų panaudojimas geografinių žemėlapių pateikimui internetu yra viena iš naujai besiformuojančių metodikų⁵. Realizuoto sistemos prototipo atveju, klientas žemėlapius nuskaito iš žemėlapių saugyklos, prie kurios prieiga teikiama per REST servisą. Kiekvienas žemėlapių fragmentas išrenkamas pagal jį apibrėžiančias x, y koordinatas ir z priartinimo lygį. Šis parametrų trejetas paprasčiausiai nurodomas nuorodoje kreipiantis į servisą. Tokios nuorodos pavyzdys:

<http://myserver.com/MAPService?x=1160&y=648&z=6>

Šios užklauskos rezultatas yra reikiamas žemėlapių fragmentas.

⁵ http://www.directionsmag.com/article.php?article_id=2515&trv=1

2.3.7 Apribojimai sprendimui

Viena pagrindinių šios sistemos teikiamų paslaugų yra vartotojų galimybė stebėti save bei kitus, sutikimą davusius, vartotojus. Vartotojų realaus laiko pozicija atvaizduojama tam skirtuose žemėlapiuose, kurie imami iš specialios internetinės žemėlapių saugyklos. Judant tam tikru maršrutu, norint stebėti savo padėtį, būtina turėti PDA ar kitą mobilųjį kompiuterį su integruotu GPS imtuvu savo padėties nustatymui ir atvaizdavimui. Bendruoju atveju, realizacija neribojama naudojamo mobilaus kompiuterio, tačiau šiame magistriniame darbe yra naudojamas konkretus, šiai sistemai skirtas PDA, todėl visi su klientine dalimi susiję tyrimai apribojami Fujitsu-Siemens N560 Pocket Loox delninio kompiuterio pritaikymu. Tikėtina, jog realizuota sistema taip pat sėkmingai funkcionuos ir su kitos klasės mobiliaisiais įrenginiais, kurie palaikys J2ME technologiją, nes būtent šią technologiją numatyta naudoti sistemos realizacijoje.

Kadangi numatyti naudoti žemėlapiai yra rastrinio formato, didelės priklausomybės nuo naudojamų žemėlapių nėra ir juos nesunkiai galima pakeisti kitais tokio tipo žemėlapiais.

Taip pat labai svarbu akcentuoti, jog mobilusis įrenginys būtinai turi turėti prieigą prie interneto, priešingu atveju bus neįmanoma atsisiųsti žemėlapių pozicijos atvaizdavimui, nebent šie bus patalpinti žemėlapių keše iš anksčiau. Be to, neturint interneto, negalima stebėti kitų vartotojų pozicijos bei prisijungti prie sistemos.

Apibendrinus, turime sekančius griežtus apribojimus, kurių netenkinant nepavyks pasinaudoti paslauga:

- Būtinai delninis kompiuteris palaikantis J2ME. Maksimaliam suderinamumui pasiekti, rekomenduojama naudoti *Fujitsu-Siemens N560 Pocket Loox* su integruotu GPS imtuvu.
- Būtinai pastovi prieiga prie interneto.
- Įrenginyje turi būti įdiegta J2ME (CLDC-1.1, MIDP-2.0) specifikaciją atitinkanti JVM (Java Virtual Machine).

Serveris realizuotas Java Servlet technologijos pagrindu, todėl gali funkcionuoti tiek Windows, tiek Linux operacinėse sistemose be griežtesnių apribojimų.

2.4. Paskirstytų sistemų testavimas

Programinės įrangos sistemos nuolat auga ir sudėtingėja. Dauguma sistemų pereina nuo paprastos architektūros prie paskirstytos. Paskirstytos architektūros sistema – tai tokia sistema, kurios programinė įranga yra vykdoma daugiau nei viename procesoriuje.

Dažniausiai paskirstytos architektūros sistemose naudojama kliento-serverio architektūra. Serveriai, kurie teikia tam tikrus servigus, yra kviečiami iš klientų ir traktuojami skirtingai negu patys klientai. Klientai dažniausiai žino apie serverius, o serveriai apie klientus – ne. Klientinei sistemos dalei gali būti naudojami dviejų tipų modeliai [11]:

- „Lengvo“ kliento modelis – jame visų programų vykdymas ir valdymas yra atliekamas serveryje, o klientas yra atsakingas tik už atvaizdavimą.
- „Sunkaus“ kliento modelis – šiuo atveju serveris yra atsakingas tik už duomenų valdymą, o kliento programinė įranga įgyvendina taikymus ir sąveiką su vartotoju.

Daugiavartotojiškoje pozicionavimo sistemoje yra realizuotas sunkaus kliento modelis, kadangi klientinė sistemos dalis yra ne tik atsakinga už atvaizdavimą, bet ir atlieka duomenų registravimą ir apdorojimą.

Kaip matyti iš sistemos apibrėžimo, paskirstytos sistemos skiriasi nuo įprastų sistemų tuo, kad jos dalys gali būti vykdomos ne tik skirtingų procesorių, bet ir skirtingose platformose. Tai reiškia, kad skirtingos sistemos dalys gali būti vykdomos skirtingose aplinkose, kurios skiriasi aparatine ir programine įranga, operacinėmis sistemomis ir t.t. Visa tai leidžia padidinti sistemos taikymų spektrą, tačiau tuo pačiu sukelia ir nemažai sunkumų.

Viena pagrindinių problemų paskirstytų sistemų kūrimo yra jų testavimas. Šiuo atveju ne visada pavyksta pritaikyti tradicinius sistemos kūrimo ir testavimo metodus ir būdus, ypač jei klientinės sistemos dalys yra vykdomos ribotų galimybių įrenginiuose, tokiuose kaip mobilieji telefonai ar PDA [1].

Iš dalies šias problemas galima sumažinti pasirenkant tinkamą programinės įrangos kūrimo procesą. Paprastai J2ME technologija paremtos programinės įrangos kūrimo procesas susideda iš tokių etapų:

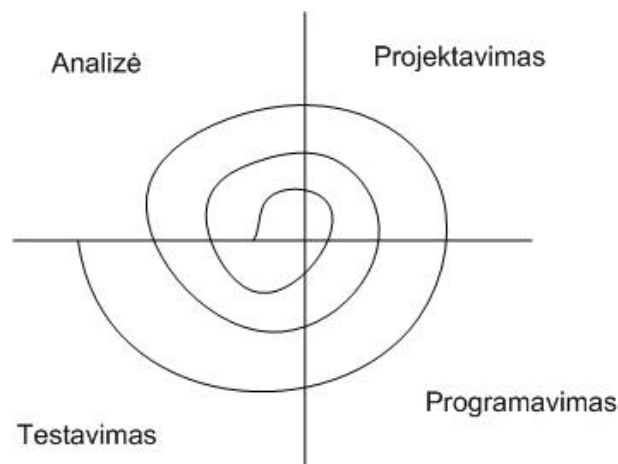
- projektavimo,
- realizavimo (kodo rašymo),
- testavimo naudojant emuliatorius,
- testavimo realiame įrenginyje.

Projektavimo etape apibrėžiama sistemos architektūra, realizacijos ypatumai, testavimo planas, testavimo atvejai ir t.t. Realizavimo etape atliekamas kodo rašymas. Testavimo etape atliekamas vienetų testavimas ir integracinis testavimas. Šiuo atveju dažniausiai sistema testuojama naudojant emuliatorius, o sistemos veikimo realioje aplinkoje testavimas atliekamas paskiausiai. Toks procesas dažniausiai yra taikomas dėl kelių priežasčių:

- sistema kuriama platformai, kuri dar nėra sukurta;
- sistemą kuria geografiškai nutolusios komandos;
- nėra tinkamų įrankių, leidžiančių efektyviai testuoti sistemos veikimą realiame įrenginyje;
- testavimas realiame įrenginyje reikalauja daug laiko.

Toks procesas yra tinkamas, jei klientinė sistemos dalis nėra didelė ir nereikalauja sąlyginai didelių kompiuterio resursų. Tačiau tais atvejais, kai sistemos klientinis modulis yra kuriamas naudojant „sunkaus“ kliento modelį, toks procesas nėra tinkamas. Šiuo atveju iškyla grėsmė, kad kuriant ir testuojant sistemą naudojami emuliatoriai ne visiškai tiksliai imituoja galutinės platformos veikimą. Tokiu atveju gali nutikti taip, kad sistema bus ištestuota ir be klaidų veiks naudojant emuliatorių, tačiau įdiegus ją realioje aplinkoje sistema bus nefunkcionalinė.

Kad to išvengtų, kuriant programinę įrangą ribotų galimybių įrenginiams, rekomenduojama taikyti spiralinį PĮ kūrimo procesą (5 pav.) [6]. Šiuo atveju rekomenduojama sistemos kūrimo pradžioje identifikuoti sistemos dalis, kurios potencialiai yra kritinės ir reikalauja daugiausiai resursų. Sistemos kūrimą pradėti nuo šių dalių realizavimo ir testavimo [10]. Be to, testavimui naudoti ne tik emuliatorius, bet ir testuoti sukurtas sistemos dalis realioje aplinkoje. Tai leis kuo anksčiau nustatyti galimas grėsmes sistemos realizacijai ir jų išvengti.



5 pav. Evoliucinis PĮ kūrimo modelis

2.5. Užduotis projektavimui

Suprojektuoti ir realizuoti paskirstytos architektūros daugiavartotojišką pozicionavimo sistemą. Sistema turi užtikrinti tokį funkcionalumą:

- savo pozicijos nustatymą ir atvaizdavimą žemėlapyje,
- maršruto fiksavimą, išsaugojimą ir atvaizdavimą,
- galimybę stebėti kitus sistemos vartotojus,
- galimybę keistis išsaugotais maršrutais.

2.6. Analizės išvados

- Analizuojant plačiai naudojamas pozicionavimo sistemas buvo išskirtas pagrindinis šių sistemų teikiamas funkcionalumas:
 - objekto pozicijos nustatymas,
 - objekto pozicijos pateikimas vartotojui priimtinu būdu,
 - duomenų apie objekto pozicijos kitimą (maršrutą) registravimas,
 - maršrutų saugojimas ir pateikimas vartotojams patogiu būdu,
 - kitos paslaugos.
- Atsižvelgiant į standartinių pozicionavimo sistemų funkcionalumą buvo apibrėžtas daugiavartotojiškos pozicionavimo sistemos funkcionalumas.
- Pasirinkta kuriamos sistemos architektūra, leidžianti realizuoti komunikaciją tarp sistemos vartotojų.
- Pasirinkta pozicionavimo sistema.
- Pasirinkti metodai pozicijos atvaizdavimui bei komunikacijai tarp serverio ir kliento.
- Apibrėžtos paskirstytų sistemų testavimo problemos.
- Suformuluota užduotis projektavimui.
- Pasirinktos priemonės sistemos kūrimui.

3. Sistemos prototipas

Šiame skyriuje pateikiamos esminės sukurto sistemos prototipo specifikacijos dalys: sistemai keliami reikalavimai, sistemos architektūra, su sistemos realizacija susijusių ypatumų aprašymai, sistemos testavimas ir diegimas.

3.1. Sistemos reikalavimai

3.1.1 Sistemos tikslai

Sistemos tikslas – registruotiems vartotojams leisti stebėti savo ir kitų, sutikimą davusių, vartotojų pozicijas realiu laiku. Taip pat, teikti maršrutų sudarymo, jų saugojimo ir peržiūros galimybes.

3.1.2 Vartotojai

Sistemoje numatomos dvi vartotojų grupės:

1) **Vartotojai** – asmenys, kurie naudojami sistemos teikiamomis galimybėmis. Tai įvairių amžiaus grupių vartotojai, turintys PDA su integruotu GPS imtuvu ir prieigą prie interneto bei mokantys naudotis IT technologijomis. Šie vartotojai naudojami sistema asmeniniais arba verslo tikslais.

2) **Administratoriai** – asmenys, administruojantys sistemą. Administratoriai prižiūri sistemos veikimą, administruoja vartotojus, sistemos programinę įrangą, prieigą prie sistemos. Tai turėtų būti patyrę darbuotojai, gerai nusimanantys apie informacines technologijas (toliau IT), sistemą, jos struktūrą, funkcijas ir galimybes. Šie žmonės turi gerai išmanyti GPS ir kitų sistemoje naudojamų technologijų veikimą. Taip pat turėtų žinoti apie galimus sistemos veikimo sutrikimus ir jų pašalinimo būdus.

Sistemos vartotojai, naudodamiesi sistema, gali:

- 1) registruotis sistemoje;
- 2) realiu laiku stebėti savo bei kitų vartotojų poziciją žemėlapyje;
- 3) peržiūrėti užfiksuotus savo ir kitų vartotojų judėjimo maršrutus;
- 4) įrašyti ir išsaugoti savo judėjimo maršrutą, kad jį vėliau galėtų peržiūrėti patys bei kiti sistemos vartotojai;
- 5) peržiūrėti savo bei kitų vartotojų išsaugotus maršrutus;
- 6) administruoti savo išsaugotus maršrutus.

Sistemą administruojantys vartotojai gali:

- 1) sukurti naujus vartotojus ir patvirtinti naujų, pageidaujančių tokiomis tapti, registraciją;
- 2) administruoti vartotojus ir su jais susijusią informaciją sistemoje;
- 3) administruoti vartotojų maršrutus.

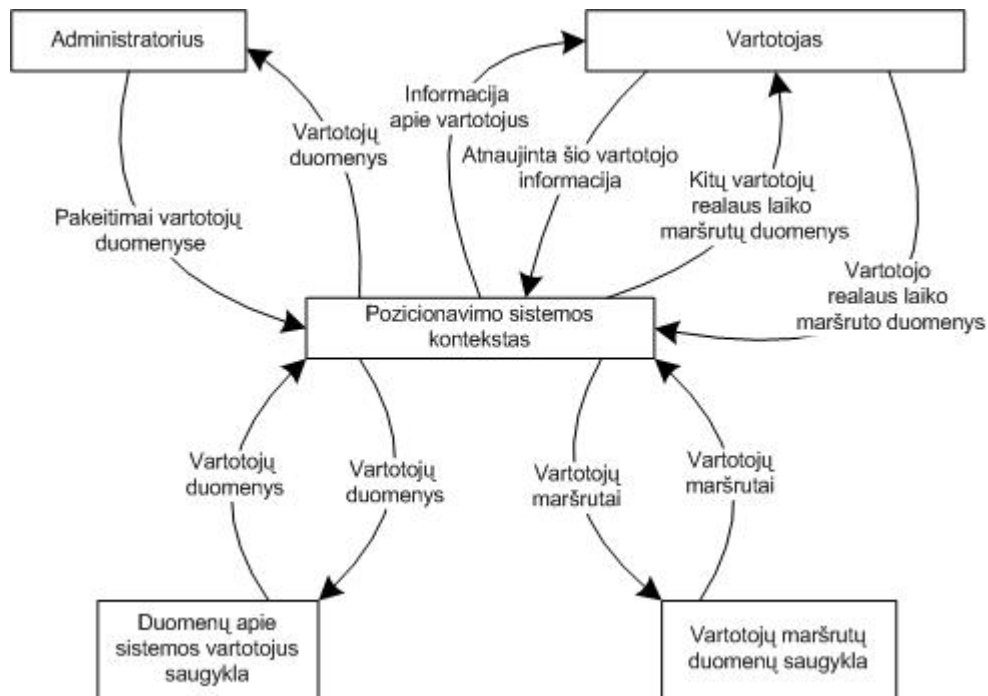
3.1.3 Diegimo aplinka

Kaip jau ir buvo minėta, viena pagrindinių sistemos dalių – klientinė, funkcionuos delniniame kompiuteryje *Fujitsu-Siemens N560 Pocket Loox* su integruotu GPS įrenginiu (*SirfStar3*; GPS protokolas – *NMEA*). Šis delninis kompiuteris turi įdiegtą *Windows Mobile 2005* operacinę sistemą. Naudojama *J2ME* specifikacijos realizacija yra *IBM J9 JVM (MIDP 2.0)*.

Serverio PĮ bus realizuojama panaudojant Java Servlet technologiją, Tapestry karkasas.

3.1.4 Veiklos kontekstas

Šiame skyriuje pateikiama pozicionavimo sistemos veiklos konteksto diagrama (6 pav.), vaizduojanti pozicionavimo sistemos sąveiką su vartotojais ir duomenų saugyklomis.



6 pav. Veiklos konteksto diagrama

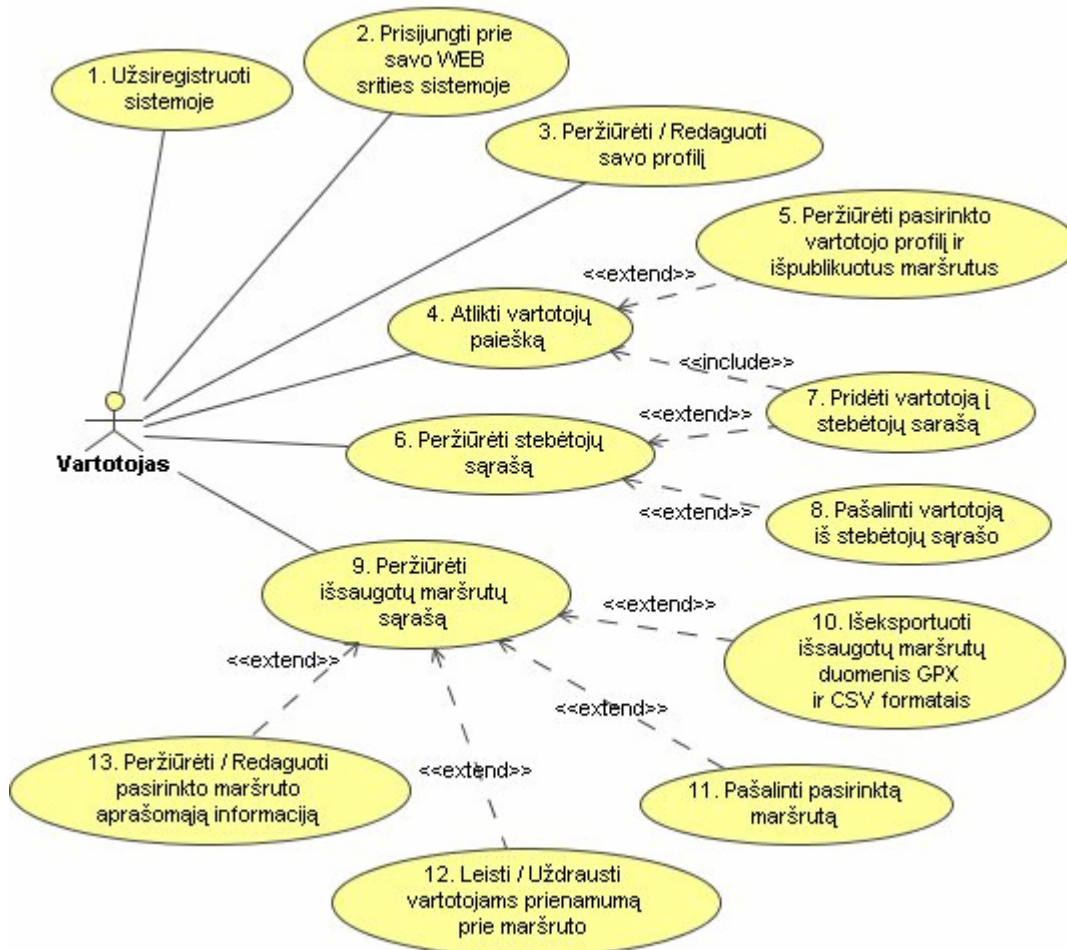
Sistemoje numatomi duomenų srautai pateikiami 2 lentelėje. Čia „in“ žymimas įeinantis srautas, o „out“ – išeinantis srautas.

2 lentelė. Veiklos padalinimas

Eil. Nr.	Įvykio pavadinimas	Įeinantys/Išeinantys informacijos srautai
1	Vartotojas pateikia pageidavimą naudotis sistema (registracija)	Vartotojo asmeniniai duomenys: vardas, pavarde, el. paštas ir kt. (in)
2	Vartotojas/Administratorius jungiasi prie WEB srities	Vartotojo prisijungimo duomenys: vartotojo vardas ir slaptažodis (in)
3	Vartotojo GPS įrenginys realiu laiku perduoda vartotojo buvimo vietą	Vartotojo realaus laiko GPS koordinatės (in)
4	Vartotojas stebi kitų sistemos vartotojų buvimo pozicijas realiu laiku	Pasirinkti norimi stebėti vartotojai (in). Vartotojų būvimo koordinatės (out)
5	Vartotojas keičia savo asmeninius duomenis	Atnaujinti vartotojo asmeniniai duomenys (in, out)
6	Vartotojas suteikia galimybę jį stebėti kitiems vartotojams	Sistemos vartotojų, kuriems ši galimybė suteikiama, identifikatoriai (in)
7	Vartotojas atlieka vartotojų paiešką	Ieškomų vartotojų daliniai preliminarūs duomenys (paieškos užklausa) (in). Atitikusių paieškos kriterijų vartotojų sąrašas bei informacija apie juos (out)
8	Vartotojas peržiūri savo ar kito vartotojo duomenis	Pasirinktas vartotojas (in). Vartotoją apibūdinantys duomenys (out)
9	Vartotojas išpublikuoja savo išsaugotus maršrutus	Publikuojamų maršrutų sąrašas (in, out)
10	Vartotojas išeksportuoja jam prieinamo pasirinkto maršruto duomenis	Pasirinktas prieinamas maršrutas (in). Maršruto duomenys pageidaujamo formato faile (out)
11	Vartotojas redaguoja maršrutą apibūdinančią informaciją	Maršrutą apibūdinantys duomenys (in, out)
12	Vartotojas peržiūri maršrutą	Pasirinktas maršruto identifikatorius (in). Pasirinktas maršrutas (out)
13	Vartotojas šalina iš sistemos paties užregistruotą maršrutą	Pasirinkto maršruto identifikatorius (in)
14	Administratorius patvirtina/atmeta vartotojo pageidavimą užsiregistruoti sistemoje	Vartotojas el. paštu gauna administratoriaus patvirtinimą/atmetimą (out)

3.1.5 Sistemos panaudos atvejai

Sistemos vartotojas prie sistemos gali prisijungti pasinaudodamas standartine interneto naršykle. Ši prieiga yra skirta registracijai sistemoje, savo profilio ir saugomų maršrutų administravimui. Detalesnis portalo funkcionalumas pateikiamas portalo vartotojo panaudos atvejų diagramoje (7 pav.).



7 pav. Vartotojo panaudos atvejų diagrama (WEB srityje)

Pakomentuosime kiekvieną iš panaudos atvejų.

1. **Užsiregistruoti sistemoje** - registracijos formos užpildymas ir pateikimas norint prisiregistruoti sistemoje, kad būtų galima naudotis sistemos teikiamomis paslaugomis.

2. **Prisijungti prie savo WEB srities sistemoje** – naudojantis standartine interneto naršykle, prisijungti prie sistemos ir naudotis portalo teikiamomis galimybėmis.

3. **Peržiūrėti / Redaguoti savo profilį** – vartotojas gali peržiūrėti jį apibūdinančią informaciją (profilį), kurią jis pateikė registracijos metu. Ši informacija apima tokius duomenis

apie vartotoją, kaip vardas, pavardė, el. pašto adresas, telefonas ir kt. Taip pat vartotojas gali šią informaciją koreguoti.

4. **Atlikti vartotojų paiešką** – vartotojas gali atlikti kitų sistemoje registruotų vartotojų paiešką. Gavęs paieškos rezultatus, gali peržiūrėti pasirinktų vartotojų duomenis, išpublikuotus maršrutus bei pridėti vartotoją į stebėtojų sąrašą, t.y. vartotojų, galinčių matyti vartotojo buvimo poziciją.

5. **Peržiūrėti pasirinkto vartotojo informaciją ir išpublikuotus maršrutus** – atlikęs paiešką, vartotojas gali peržiūrėti pasirinkto vartotojo informaciją ir jo išpublikuotų maršrutų sąrašą bei maršrutus apibūdinančią informaciją

6. **Peržiūrėti stebėtojų sąrašą** – vartotojas gali peržiūrėti sąrašą kitų sistemos vartotojų, kuriems yra suteikęs teisę matyti jo poziciją realiu laiku.

7. **Pridėti vartotoją į stebėtojų sąrašą** – vartotojas gali pridėti bet kurį kitą sistemos vartotoją į savo stebėtojų sąrašą, t.y. leisti bet kuriam sistemos vartotojui stebėti savo poziciją realiu laiku.

8. **Pašalinti vartotoją iš stebėtojų sąrašo** – vartotojas, peržiūradamas savo stebėtojų sąrašą, gali pašalinti bet kurį stebėtoją. Pašalinus vartotoją iš stebėtojų sąrašo, pašalintas vartotojas netenka teisės stebėti šio vartotojo (kuris šalina) poziciją realiu laiku.

9. **Peržiūrėti išsaugotų maršrutų sąrašą** – vartotojas gali peržiūrėti sistemoje išsaugotų savo maršrutų sąrašą.

10. **Išeksportuoti išsaugotų maršrutų duomenis GPX ir CSV formatais** – vartotojas, peržiūradamas maršrutų sąrašą, gali pasirinkti norimą maršrutą ir išeksportuoti šį maršrutą GPX⁶ ir/arba CVS⁷ formatais.

11. **Pašalinti pasirinktą maršrutą** – vartotojas gali pašalinti iš sistemos bet kurį savo maršrutą.

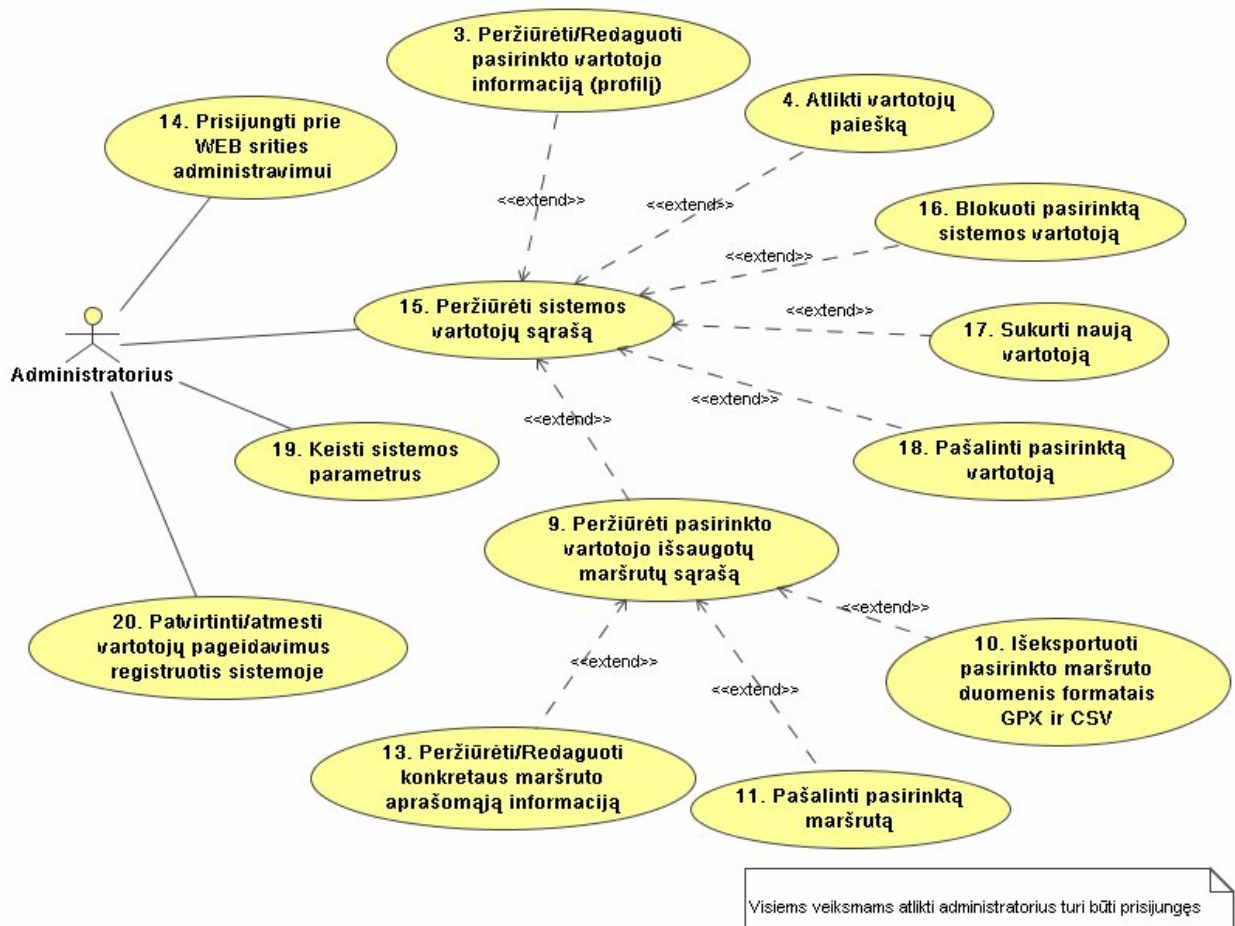
12. **Leisti / Uždrausti vartotojams prieinamumą prie maršruto** – vartotojas gali leisti kitiems sistemos vartotojams peržiūrėti pasirinktą maršrutą, t.y. išpublikuoti norimą maršrutą. Jei maršrutas išpublikuojamas, šį maršrutą gali peržiūrėti kiti sistemos vartotojai. Priešingu atveju maršrutas kitiems sistemos vartotojams yra neprieinamas.

⁶ <http://www.topografix.com/gpx.asp>

⁷ http://en.wikipedia.org/wiki/Comma-separated_values

13. **Peržiūrėti / Redaguoti pasirinkto maršruto aprašomąją informaciją** – vartotojas gali peržiūrėti ir pakeisti maršrutą apibūdinančią informaciją, tokią kaip pavadinimas, aprašymas ir kt.

Sistemos serveryje saugomų duomenų administravimas atliekamas per WEB portalo sąsają. Administratorius prie sistemos gali prisijungti su standartine interneto naršykle. Be specializuotų administravimo veiksmų, jis sistemoje gali atlikti didžiąją dalį tipiniam vartotojui prieinamų veiksmų. Administratoriui suteikiamas galimybes pateikiame administratoriaus panaudos atvejų diagramoje (8 pav.).



8 pav. Sistemos administratoriaus panaudos atvejų diagrama (WEB srityje)

14. **Prisijungti prie WEB srities administravimui** – naudodamasis standartine interneto naršykle gali prisijungti prie sistemos administravimo veiksams atlikti.

15. **Peržiūrėti sistemos vartotojų sąrašą** – administratorius gali peržiūrėti visų sistemoje registruotų vartotojų sąrašą. Naudodamasis šiuo vartotojų sąrašu administratorius gali peržiūrėti pasirinktų vartotojų informaciją, maršrutus ir atlikti vartotojų administravimo funkcijas.

16. **Blokuoti pasirinktą sistemos vartotoją** – administratorius gali užblokuoti pasirinktą sistemos vartotoją. Užblokuotas vartotojas negali naudotis sistemos teikiamomis paslaugomis.

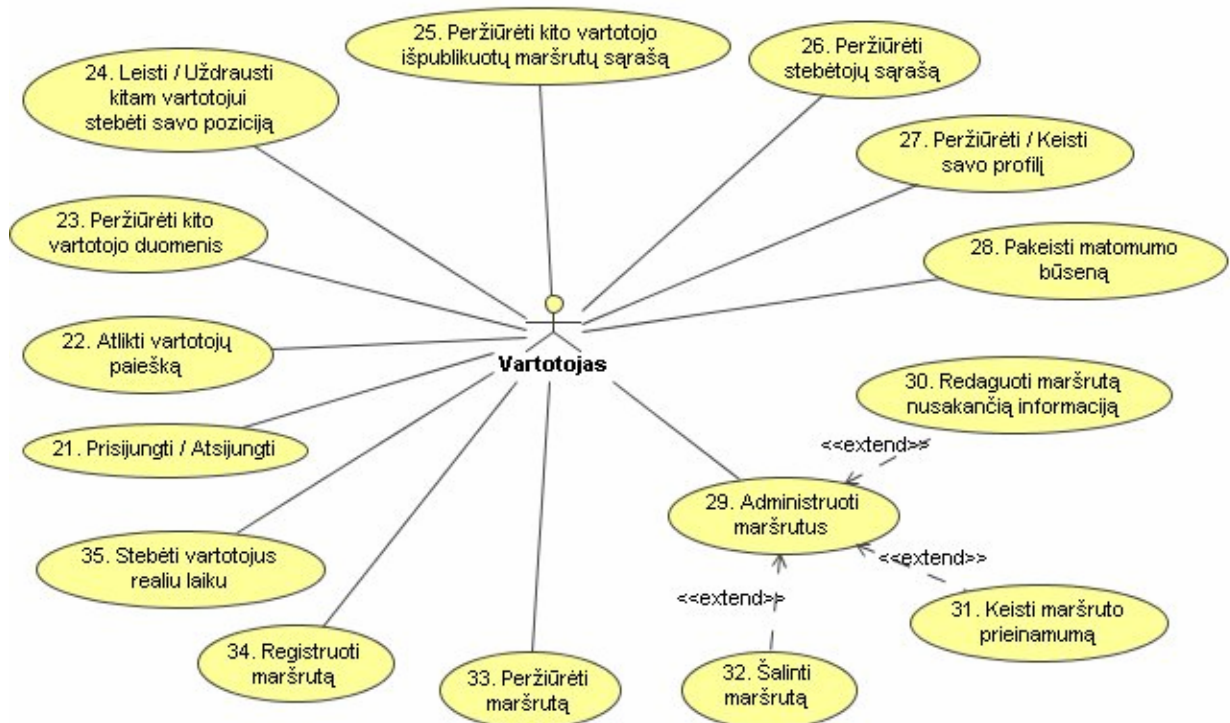
17. **Sukurti naują vartotoją** – administratorius gali pats sukurti sistemos vartotoją.

18. **Pašalinti pasirinktą vartotoją** – sistemos administratorius gali pašalinti sistemos vartotoją. Tokiu atveju vartotojo duomenys ištrinami iš sistemos.

19. **Keisti sistemos parametrus** – administratorius gali konfigūruoti sistemą nustatydamas tam tikrus sistemos veikimo parametrus.

20. **Patvirtinti / Atmesti vartotojų pageidavimus registruotis sistemoje** – sistemos vartotojams registruojantis sistemoje administratoriui pateikiami pranešimai apie vartotojus, pageidaujančius prisiregistruoti sistemoje ir naudotis jos teikiamomis paslaugomis. Administratorius gali patvirtinti arba atmesti vartotojo registraciją.

Pasinaudodamas klientu, įdiegtu delniniame kompiuteryje, sistemos vartotojas gali atlikti veiksmus, pateiktus sekančioje panaudos atvejų diagramoje (9 pav.).



9 pav. Vartotojo panaudos atvejų diagrama (klientinėje PĮ dalyje)

21. **Prisijungti / Atsijungti** – prisijungti prie sistemos norėdamas naudotis jos teikiamomis paslaugomis bei atsijungti nuo sistemos baigdamas darbą su sistema.

22. **Atlikti vartotojų paiešką** – vartotojas gali atlikti kitų sistemoje registruotų vartotojų paiešką. Gavęs paieškos rezultatus, gali peržiūrėti pasirinktų vartotojų duomenis, išpublikuotus maršrutus bei pridėti vartotoją į stebėtojų sąrašą.

23. **Peržiūrėti kito vartotojo duomenis** – vartotojas gali peržiūrėti kito sistemos vartotojo duomenis. Tai jis gali padaryti atlikęs vartotojų paiešką arba norėdamas peržiūrėti stebėtojo informaciją.

24. **Leisti / Uždrausti kitam vartotojui stebėti savo poziciją** – vartotojas gali pasirinktam vartotojui leisti/uždrausti stebėti savo poziciją realiu laiku įtraukdamas/pašalindamas vartotoją į/iš stebėtojų sąrašo.

25. **Peržiūrėti kito vartotojo išpublikuotų maršrutų sąrašą** – vartotojas gali peržiūrėti pasirinkto kito sistemos vartotojo išpublikuotų maršrutų sąrašą, atlikdamas jo duomenų peržiūrą.

26. **Peržiūrėti stebėtojų sąrašą** – vartotojas gali peržiūrėti sąrašą kitų sistemos vartotojų, kuriems yra suteikęs teisę matyti jo poziciją realiu laiku.

27. **Peržiūrėti / Keisti savo profilį** – vartotojas gali peržiūrėti ir keisti savo profilio duomenis, t.y. vardą, pavardę ir kitą informaciją.

28. **Pakeisti matomumo būseną** – vartotojas gali nustatyti ir keisti matomumo būseną. Jei vartotojas pasirenka būti nematomu, duomenys nėra siunčiami į serverį, net ir tada kai jis prisijungęs.

29. **Administruoti maršrutus** – vartotojas gali administruoti savo maršrutus peržiūrėdamas jų sąrašą.

30. **Redaguoti maršrutą nusakančią informaciją** – vartotojas gali redaguoti pasirinktą maršrutą nusakančius duomenis, tokius kaip maršruto pavadinimas, aprašymas ir kt.

31. **Keisti maršruto prieinamumą** – vartotojas gali keisti maršruto prieinamumą kitiems sistemos vartotojams. Maršrutai gali būti išpublikuoti arba neišpublikuoti.

32. **Šalinti maršrutą** – vartotojas gali pašalinti pasirinktą išsaugotą maršrutą iš serverio.

33. **Peržiūrėti maršrutą** – pasinaudodamas klientu, vartotojas gali peržiūrėti pasirinktą maršrutą. Peržiūra vyksta maršrutą atvaizduojant žemėlapyje.

34. **Registruoti maršrutą** – kaupti maršruto duomenis realiu laiku. Bet kuriuo metu galima nutraukti maršruto duomenų kaupimą ir išsaugoti jį lokaliame faile arba serveryje.

35. **Stebėti vartotojus realiu laiku** – vartotojas gali realiu laiku stebėti savo ir kitų, sutikimą davusių, sistemos vartotojų poziciją žemėlapyje.

3.1.6 Funkciniai reikalavimai

Funkciniai reikalavimai suformuluoti detalizuojant esamus panaudos atvejus. Kadangi tam tikros sistemos teikiamos funkcijos prieinamos tiek per portalą, tiek ir per klientą, tai dalis reikalavimų galioja abiem atvejams. Tačiau tam tikrais atvejais, dėl skirtingų platformų ir vartotojo sąsajos ypatumų, dubliuojamos funkcijos gali skirtis atvaizdavimo ir pateikiamos informacijos detalumu.

Toliau pateikiame tik esminius, sistemai taikomus, funkcinius reikalavimus:

- Vartotojo vardas sistemoje turi būti unikalus, todėl sistema turi patikrinti ir neleisti sukurti vartotojo su jau egzistuojančiu vartotojo vardu ir formuoti atitinkamą pranešimą.
- Registracijos ir profilio keitimo metu, vartotojo įvedama informaciją turi būti patikrinama, ar tenkina duomenų turinio ir apimties reikalavimus.
- Sistemos teikiamos galimybės turi būti prieinamos tik registruotiems ir prisijungusiems sistemos vartotojams. Bet kurie neprisijungę vartotojai, mėginami atlikti veiksmus, leistinus tik prisijungusiems vartotojams, turi būti automatiškai nukreipiami į prisijungimo formos langą.
- Jei sistemos vartotojas yra užblokuotas, tai jam jungiantis prie sistemos jis turi būti apie tai informuojamas atitinkamu pranešimu.
- Atlikęs paiešką, vartotojas turi turėti galimybę viename puslapyje peržiūrėti jo nurodytą pageidaujamą rezultatų kiekį, o sekančius rezultatų puslapius pasiekti nuspausdamas atitinkamą nuorodą.
- Kiekvienos paieškos užklausos metu, turėtų būti atliekama dalinė paiešką, o paieškos rezultatai nesaugomi serverio atmintyje.
- Vartotojui peržiūrint savo maršrutų arba jų galinčių stebėti vartotojų sąrašą, turi būti pateikiamas puslapiuojamas sąrašas su navigacija.
- Administratoriui patenkinus registracijos prašymą ir vartotojui patvirtinus savo el. pašto adresą, turi būti automatiškai sukuriamas aktyvus sistemos vartotojas.
- Vartotojo pašalinimo atveju, visi su juo susiję sistemoje saugomi duomenys turi būti pašalinami.

- Su klientu neprisijungęs prie serverio vartotojas turi turėti galimybę stebėti savo poziciją žemėlapyje ir išsaugoti maršrutą lokaliai, tačiau visos kitos sistemos teikiamos galimybės jam turi būti neprieinamos.
- Naudodamasis klientu, vartotojas turi turėti galimybę pasirinkti, kokius vartotojus jis nori stebėti žemėlapyje.

3.1.7 Nefunkciniai reikalavimai

Nefunkciniai sistemos reikalavimai apima reikalavimus vartotojo sąsajai, sistemos saugumui, panaudojamumui, vykdymo charakteristikoms, duomenims, sistemos palaikymui ir pan.

Toliau pateikiame tik esminius nefunkcinius sistemai taikomus reikalavimus:

- Vartotojo sąsaja turi būti neperkrauta, paprasto dizaino, meniu aiškus ir tinkamai išdėstytas.
- Sistema turi būti paprasta naudotis vartotojui, susipažinusiame su bazine naudojimosi instrukcija.
- Vartotojas neturi būti priverstas ilgai laukti, kol bus pradėti vykdyti jo nurodymai.
- Sistema neturi perduoti asmeninių vartotojo duomenų kitiems, su sistemos administravimu nesusijusiems, asmenims.
- Delninis kompiuteris, lyginant su stacionariu, turi nedidelį ekraną, todėl kliento sąsajos elementai turi būti pakankamo dydžio ir aiškiai įžiūrimi, nes smulkų tekstą sunku įskaityti, o smulkius mygtukus – paspausti.
- Vartotojo sąsajos elementai turi būti sugrupuoti pagal paskirtį ir, jei reikia, išdėlioti atskirose skiltyse. Sąsajos elementų grupavimas pagal paskirtį suteikia galimybę greičiau juos aptikti ir greičiau atlikti reikiamus pakeitimus.
- Sistemos kliento pilnas paleidimas turėtų būti įvykdomas paleidžiant vieną vykdomąjį failą, o ne atskirai paleidinėjant atskiras posistemas.

3.2. Sistemos architektūra

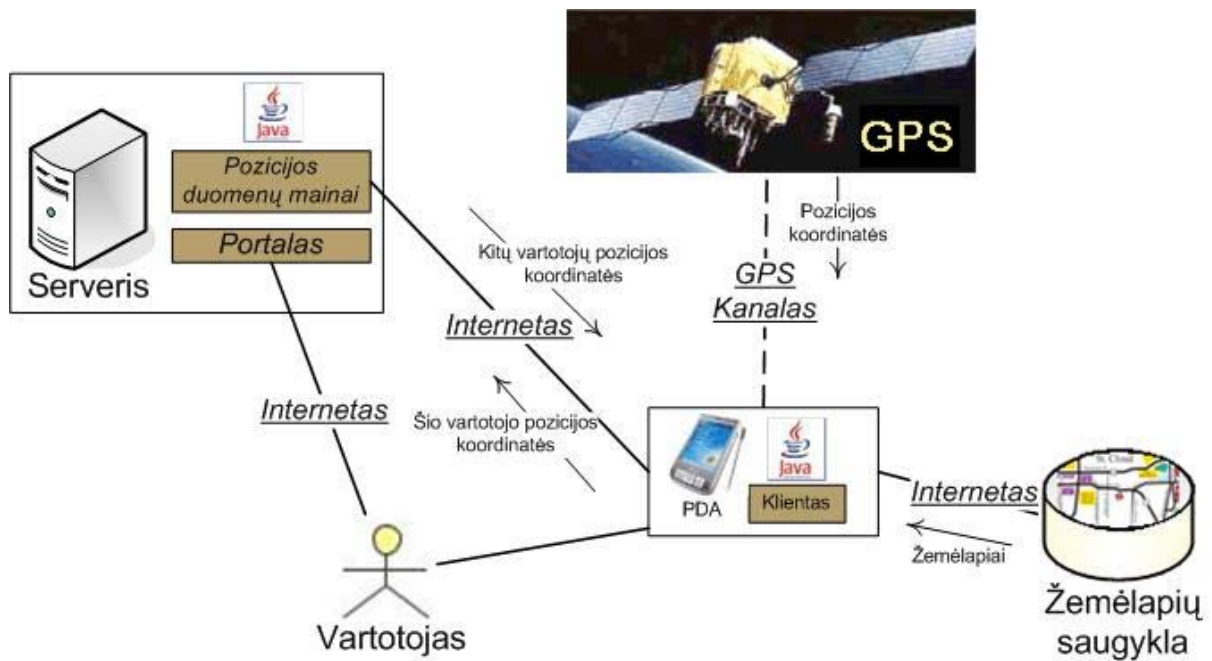
3.2.1 Sistemos veikimo principas

Sistemos prototipo veikimas yra paremtas kliento ir serverio principu. Sistemos vartotojams yra prieinamos dvi prieigos rūšys: internetinis portalas ir klientinė PĮ. Internetinis portalas yra skirtas sistemos administravimui, vartotojų registracijai sistemoje bei suteikia vartotojams jų profilio, maršrutų ir stebėtojų administravimo galimybes. Klientinė programinė įranga leidžia ne tik administruoti savo profilį, maršrutus ir stebėtojų sąrašą, bet ir stebėti savo ir kitų vartotojų pozicijas žemėlapyje realiu laiku. Klientinės PĮ pagalba vartotojas taip pat gali užregistruoti norimą maršrutą.

Maršruto registravimas vykdomas keliais etapais. Maršruto registravimą inicijuoja ir nutraukia pats vartotojas. Pradėjęs registravimą, sistema periodiškai nuskaito duomenis iš GPS imtuvo ir juos išsaugo. Kai vartotojas nusprendžia baigti maršruto registravimą, sistema jam leidžia išsaugoti nuskaitytą maršrutą. Išsaugojimo metu vartotojas gali nurodyti maršrutą apibūdinančią informaciją, tokią kaip pavadinimas ir aprašymas, ir pasirinkti maršruto saugojimo vietą. Maršrutą vartotojas gali išsaugoti į failą arba serveryje.

Kad vartotojas galėtų stebėti kitų vartotojų pozicijas realiu laiku, jis turi žinoti kitų vartotojų koordinates. Duomenų apie koordinates mainai yra vykdomi per serverį. Delninio kompiuterio GPS įrenginys vartotojo pozicijos duomenis gauna iš GPS palydovų ir perduoda klientui. Klientas šiuos duomenis siunčia į serverį, tokiu būdu sudarydamas galimybę juos pasiimti kitiems, tokiu pačiu klientu besinaudojantiems ir atitinkamą leidimą turintiems, vartotojams. Tuomet vartotojai kreipiasi į serverį, kuris patikrina jų užklausas ir grąžina stebimų vartotojų pozicijos koordinates. Toliau klientinė PĮ šias koordinates atvaizduoja skaitmeniniame žemėlapyje.

Vartotojų pozicijos atvaizdavimui naudojami žemėlapiai gali būti imami iš dviejų šaltinių: lokalaus kešo ir internetinės žemėlapių saugyklos. Pirmiausiai tikrinamas lokalus kešas. Jeigu jame yra randamas tinkamas žemėlapių fragmentas, tai jis ir yra naudojamas atvaizdavimui. Jei lokaliame keše nerandamas reikiamas žemėlapių fragmentas, tai sistema kreipiasi į internetinę žemėlapių saugyklą, iš kurios gauna reikiamą žemėlapių fragmentą. Iš internetinės žemėlapių saugyklos atsiųstas žemėlapių fragmentas yra įrašomas į lokalią kešą, kad jį būtų galima naudoti vėlesniems atvaizdavimams. Tai leidžia sumažinti duomenų srautų tarp serverio ir kliento apimtį.



10 pav. Sistemos veikimo principas

3.2.2 Bendroji architektūra

Kaip ir buvo minėta, šios sistemos architektūra yra paremta kliento ir serverio principu. Serverio dalis yra pilnai paremta tik *Java* technologija ir jos pagrindu sukurtomis *Tapestry Framework*, *Apache Axis* bei kitomis nemokamomis programinėmis bibliotekomis. Serveryje išskiriami 4 komponentai:

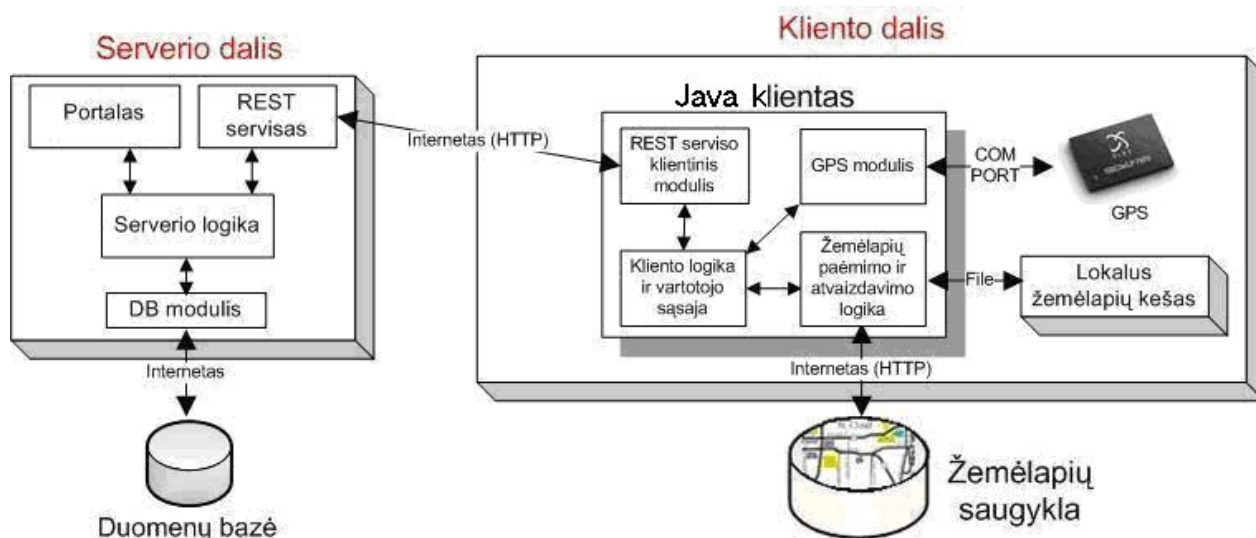
1) Portalas – tai vartotojams teikiama sistemos prieiga. Ji skirta sistemos administravimui ir vartotojų prieigai prie sistemos per standartinę interneto naršyklę. Portalas realizuotas naudojant atviro kodo *Java* programavimo kalba paremtą kūrimo šabloną *Tapestry Framework*, skirtą kurti sudėtingas *WEB* taikomas programas.

2) REST servisas – modulis, realizuojantis komunikaciją tarp serverio ir klientinės PĮ. Šis servisas naudojamas visiems duomenų mainams tarp serverio ir klientinės PĮ. Prototipo sistemos REST servisas pagrįstas labai paprastu principu. Jis naudoja HTTP protokolą ir neturi jokio papildomo pranešimų perdavimo sluoksnio. Pranešimai yra perduodami tiesiog per HTTP parametrus. Kliento siunčiamas pranešimas, arba tiesiog HTTP užklausa, iškviečia tam tikrą REST serviso metodą, kuris gali turėti parametrus arba ne. Tokios užklauskos rezultatas yra HTTP atsakas – struktūrizuotas XML (eXtensible Markup Language) srautas. Komunikuojant klientui ir serveriui yra būtina identifikuoti, kuris vartotojas vykdo užklausą, todėl yra būtinas identifikacijos mechanizmas. Šiuo atveju jis realizuotas ad-hoc principu. Pirmą kartą jungiantis vartotojui yra vykdoma autentifikacija ir gražinamas SHA-1 algoritmu sugeneruotas kodas. Šis

kodas naudojamas kaip sesijos identifikatorius prisijungusiam vartotojui identifikuoti ir yra pridamas prie užklauso parametru kiekvieną kartą klientui kviečiant kitus REST serviso metodus.

3) Serverio logikos komponentas – modulis, realizuojantis serverio logiką, apimantis veiklos taisykles ir galimus sistemos naudojimo scenarijus.

4) Duomenų bazės modulis – šis modulis naudojamas komunikacijai su duomenų baze.



11 pav. Bendroji sistemos architektūra

Kliento PĮ taip pat išskiriamos 4 dalys:

1) Kliento vidinės logikos ir vartotojo sąsajos modulis – šiuo atveju vartotojo sąsaja ir kliento logika, apimanti galimus veiklos scenarijus, yra glaudžiai susijusios, todėl apjungtos į vieną bendrą modulį. Tai pagrindinis klientinės PĮ modulis, atsakingas už vartotojo sąsają ir sistemos funkcionalumą.

2) GPS modulis – tai pagrindinio modulio naudojamas paketas, skirtas duomenų iš GPS įrenginio nuskaitymui. Jis nuskaity koordinatas iš GPS įrenginio per nuosekliają sąsajos jungtį (COM port).

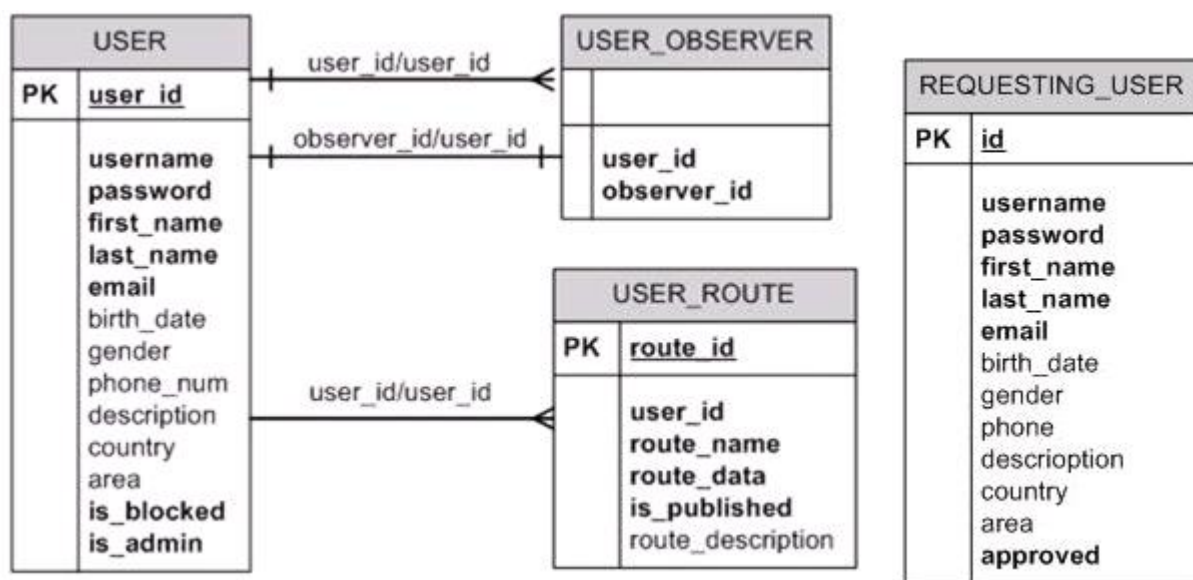
3) Žemėlapių paėmimo ir atvaizdavimo logikos modulis – skirtas žemėlapių nuskaitymui iš lokalaus kešo arba internetinės žemėlapių saugyklos. Šis modulis parenka ir suranda atvaizdavimui reikiamą žemėlapi. Parenkant žemėlapi apskaičiuojamas reikiamas žemėlapio fragmentas, o tada jis užkraunamas iš lokalaus kešo. Jei lokaliame keše reikiamo žemėlapio nėra, šis modulis jį atsiunčia iš internetinės žemėlapių saugyklos ir įrašo į lokalių kešą.

4) REST serviso klientinis modulis – naudojamas komunikavimui su serveriu. Per šį modulį vykdomi visi duomenų mainai su serveriu. Modulis atsakingas už sesijos iniciavimą ir valdymą, užklauso serveriui formavimą ir atsakymų apdorojimą.

Viena pagrindinių sistemos dalių – klientas, veikia delniniame kompiuteryje. Tokio kompiuterio našumas lyginant su paprasto stalinio kompiuterio našumu yra labai mažas. Prieigai prie interneto dažniausiai naudojamas bendras paketinis radijo ryšys (*General Packet Radio Service* – GPRS), kur apmokestinamas kiekvienas išsiųstas megabaitas. Būtent dėl šių priežasčių bendra sistemos architektūra yra sudaryta taupant resursus ir mažinant persiunčiamų bei naudojamų duomenų apimtį.

3.2.3 Duomenų bazė

Sistemoje duomenų saugojimui naudojama MySQL duomenų bazė. Duomenų bazės schema pateikta 12 pav.



12 pav. Duomenų bazės schema

Pastaba: Diagramoje naudojamas žymėjimas PK – pirminis raktas.

Duomenų bazės lentelių specifikacijos su lentelių laukais, duomenų tipais ir laukų aprašymais pateikiamos toliau.

Duomenų bazės lentelė „USER“ naudojama saugoti duomenims apie sistemos vartotojus.

3 lentelė. Duomenų bazės lentelės "USER" specifikacija

USER			
PK	Laukas	Tipas	Aprašymas
*	user_id	INTEGER	Vartotojo identifikatorius
	username	VARCHAR(32)	Vartotojo prisijungimo vardas
	password	CHAR(41)	Prisijungimo slaptažodis
	first_name	VARCHAR(40)	Vartotojo vardas
	last_name	VARCHAR(40)	Vartotojo pavardė
	email	VARCHAR(90)	El. pašto adresas
	birth_date	BIGINT(20)	Gimimo data
	gender	TINYINT(1)	Vartotojo lytis
	phone	VARCHAR(24)	Telefono numeris
	description	TEXT	Vartotojo aprašymas
	country	VARCHAR(90)	Vartotojo šalis
	area	VARCHAR(90)	Vartotojo miestas
	is_blocked	TINYINT(1)	Laukas, skirti pažymėti, kad vartotojas yra blokuojamas.
	is_admin	TINYINT(1)	Laukas, žymintis, kad vartotojas yra administratorius.

Pastaba: paryškinti laukai yra privalomi.

Duomenų bazės lentelėje „REQUESTING_USER“ saugomi prašymai registruotis sistemoje. Čia saugoma visa informacija, kuri naudojama naujo sistemos vartotojo sukūrimui, kai administratorius patvirtina prašymą registruotis, o vartotojas patvirtina registraciją pasinaudodamas į jo el. paštą siunčiamu pranešimu.

4 lentelė. Duomenų bazės lentelės "REQUESTING_USER" specifikacija

REQUESTING_USER			
PK	Laukas	Tipas	Aprašymas
*	id	INTEGER	Identifikatorius
	username	VARCHAR(32)	Vartotojo prisijungimo vardas
	password	CHAR(41)	Prisijungimo slaptažodis
	first_name	VARCHAR(40)	Vartotojo vardas
	last_name	VARCHAR(40)	Vartotojo pavardė
	email	VARCHAR(90)	El. pašto adresas
	birth_date	BIGINT(20)	Gimimo data
	gender	TINYINT(1)	Vartotojo lytis
	phone	VARCHAR(24)	Telefono numeris
	description	TEXT	Vartotojo aprašymas
	country	VARCHAR(90)	Vartotojo šalis
	area	VARCHAR(90)	Vartotojo miestas
	approved	TINYINT(1)	Laukas, žymintis, kad administratorius patvirtino registraciją.

Duomenų bazės lentelėje „USER_ROUTE“ yra saugomi vartotojo maršrutai ir su jais susiję metaduomenys, kuriuos jis išsaugo pasinaudodamas klientu.

5 lentelė. Duomenų bazės lentelės "USER_ROUTE" specifikacija

USER_ROUTE			
PK	Laukas	Tipas	Aprašymas
*	id	INTEGER	Identifikatorius
	user_id	INTEGER	Vartotojo identifikatorius, rodantis, kam priklauso maršrutas.
	route_name	VARCHAR(90)	Maršruto pavadinimas
	route_data	LONGTEXT	Maršruto koordinatės
	is_published	TINYINT(1)	Laukas, parodantis, ar maršrutas išpublikuotas.
	route_description	TEXT	Maršruto aprašymas

Duomenų bazės lentelė „USER_OBSERVER“ yra naudojama vartotojo stebėtojų sąrašui saugoti.

6 lentelė. Duomenų bazės lentelės "USER_OBSERVER" specifikacija

USER_OBSERVER			
PK	Laukas	Tipas	Aprašymas
	user_id	INTEGER	Vartotojo identifikatorius, rodantis, kuriam vartotojui priklauso stebėtojas.
	observer_id	INTEGER	Stebėtojo identifikatorius, nurodantis, kuris vartotojas yra stebėtojas.

3.3. Testavimas

3.3.1 Testavimo tikslai ir apimtis

Testavimo metu siekiama patikrinti sistemos atitikimą specifikacijai, aptikti galimas programavimo bei logikos klaidas, patikrinti atskirų sistemos dalių funkcionavimą bei kliento ir serverio dalių komunikavimą. Taip pat atliekama vartotojo sąsajos patikra ir įvertinimas.

Testavimui naudojamos kelių tipų strategijos. Vienetų testavimui taikoma „baltos dėžės“ (angl. „white box“) testavimo strategija, kuria siekiama patikrinti kodo padengimą bei aptikti galimas vidines logikos klaidas. Taip pat naudojama „juodos dėžės“ (angl. „black box“) testavimo strategija, kuria bandoma aptikti galimas funkcines klaidas. Toks vienetų testavimas taikomas abiemis sistemos dalims: tiek klientinei, tiek ir serverio daliai.

„Juodos dėžės“ testavimo strategija taikoma ir integracijos testavime, siekiant patikrinti apjungtų sistemos dalių funkcionavimą. Testavimo metu taip pat vertinama vartotojo sąsaja, jos patogumas, priimtinumai bei kitos savybės.

3.3.2 Pagrindiniai apribojimai

Šiame skyriuje pateikiami pagrindiniai veiksniai, galintys įtakoti sistemos testavimo pilnumą, t.y. kiek gerai galima ištestuoti sistemą.

1. Laiko trūkumas – sistemos testavimui ir klaidų ištaisymui yra skiriama palyginti nedaug laiko.
2. Įrangos trūkumas – turime tik vieną delninį kompiuterį, todėl pilnai iširti sistemos elgesį stresinėse situacijose praktiškai yra neįmanoma. Šis apribojimas taip pat įtakoja ir integracinį testavimą bei apriboja galimybes ištestuoti sistemą kitokiose aplinkose bei kitokiuose įrenginiuose.
3. Sistemos testavimą realiomis sąlygomis gali apsunkinti interneto ryšio pajungimas į delninį kompiuterį.
4. Saugumo testavimo metu galima bus remtis tik savo asmenine patirtimi.

3.3.3 Sąsajos testavimas

WEB portalas yra realizuotas Tapestry Framework pagrindu, todėl vartotojo sąsajos testavimas nėra paprastas procesas. Šis procesas priklauso nuo programavimo stiliaus, todėl programuojant taikoma sluoksninė metodika, kuri palengvina vartotojo sąsajos testavimą. Sluoksninės metodikos idėja yra realizuoti vartotojo sąsajos komponentus tokiu būdu, kad grafinės vartotojo sąsajos kodas būtų atskirtas nuo funkcionalumą realizuojančios dalies ir grafinėje dalyje paliktas tik labai plonas specifinis sluoksnis.

Automatizuotai testuoti grafinę vartotojo sąsają nėra lengva, ypač jei ji yra realizuota didelę specifiką įnešančių tarpinių technologijų (konkrečiu atveju Tapestry Framework) pagrindu. Todėl grafinių komponentų testavimas atliekamas rankiniu būdu – paprasčiausiai atvaizduojant komponentą ir stebint jį ekrane prie įvairių įmanomų vaizduojamų duomenų.

Atskirta grafinių komponentų funkcionalumo dalis yra patalpinta atskirose klasėse, todėl čia pritaikyta vienetų testavimo (unit test) strategija komponentų funkcionalumui testuoti.

Kadangi rankiniu būdu testuojant grafinę vartotojo sąsają sistemai paduodamos įvairios reikšmės ir stebimas sistemos elgesys, tai tuo pačiu kai kur dalinai testuojamas ir sistemos funkcionalumas. Tačiau čia pagrindinis akcentas yra vartotojo sąsajos komponentų languose pateikiamų klaidos pranešimų korektiškumas, nes funkcionalumo testavimas vykdomas atskirai.

Testuojant vartotojo sąsają klientinėje PĮ naudojamas analogiškas testavimas. Tačiau šiuo atveju vartotojo sąsaja yra labiau susieta su sistemos funkcionalumu, todėl testavimo metu testuojama ir vartotojo sąsaja, ir sistemos funkcionalumas.

3.3.4 Testavimo strategijos

Taikomos testavimo strategijos:

- **Vienetų testavimas** – strategija apimanti vienetus, kurie yra mažiausios sistemos dalys, testavimą. Konkrečiu atveju šie vienetai bus Java klasės ir jų metodai.
- **Integravimo testavimas** – strategija taikoma prieš bendrą sistemos testavimą. Konkrečiu atveju orientuota į susietų Java klasių/modulių tarpusavio sąveikos testavimą.
- **Priėmimo testavimas** – testavimo strategija, skirta įsitikinti, jog kuriama sistema atitinka užsakovo reikalavimus. Testavimas vykdomas dalyvaujant užsakovui ir/arba sistemos vartotojams.
- **Aukšto lygio testavimas** – užsakovo aplinkoje vykdomas testavimas, kurio metu siekiama nustatyti, ar korektiškai veikia sistema specifinėje užsakovo aplinkoje.

3.3.5 Vienetų testavimas

WEB portalas realizuotas Java Tapestry Framework technologijos pagrindu, todėl kaip vienetus čia galima traktuoti Tapestry komponentus. Tačiau šie komponentai iš principo yra grafinės sąsajos komponentai, apjungti su susijusiu funkcionalumu. Kaip jau buvo minėta, programavimas buvo vykdomas siekiant kuo labiau atskirti grafinę sąsają nuo susieto funkcionalumo, kad komponentų funkcijoms galima būtų taikyti vienetų testavimo strategiją.

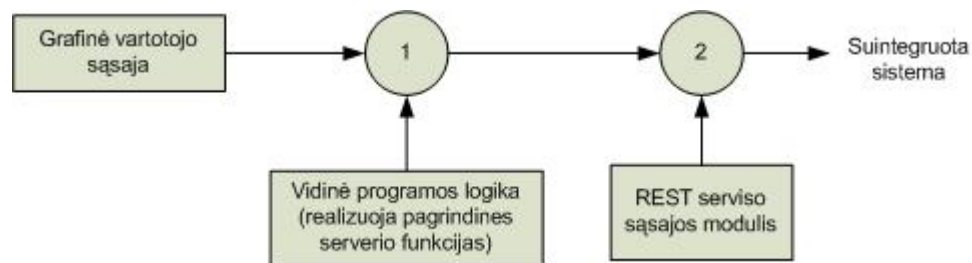
Klientinė programinė įranga yra realizuota naudojant *J2ME* technologiją. Šiuo atveju vienetais laikomos Java klasės.

Pagrindiniai testuojami vienetai yra statinės ir ne statinės klasės, bei atitinkamai jų funkcijos ir metodai. Programuojant buvo siekiama kur įmanoma ir prasminga bendrojo pobūdžio funkcionalumą realizuoti kaip statines funkcijas (iš kurių yra sudarytos statinės klasės), tokiu būdu supaprastinant sistemos struktūrą ir palengvinant testavimo procedūrą. Tokios klasės testuojamos paprasčiausiai iškviečiant jų funkcijas ir lyginant jų gražinamus rezultatus. Testuojant ne statines klases, sukuriama jų egzemplioriai, o vidiniams kintamiesiems suteikiamos įvairios reikšmės (tarp jų ir ribinės). Testuojant abiem atvejais perduodamiems per metodus ar funkcijas parametrus bus suteikiamos įvairios galimos reikšmės.

3.3.6 Integravimo testavimas

WEB portalo dalyje integravimo testavimas vykdomas grafinės vartotojo sąsajos ir REST serviso sąsajos klases apjungiant su bendrai naudojamomis vidinės logikos klasėmis. Ši vidinė logika realizuoja pagrindines funkcijas ir bus prieš tai ištestuota naudojant vienetų testavimo strategiją. Vartotojo sąsaja ir REST serviso sąsaja taip pat prieš integravimo testavimą ištestuojamos joms taikomomis testavimo metodikomis. Taigi integravimo testavimo metu siekiama nustatyti sistemos klaidas, atsiradusias apjungiant grafinę vartotojo sąsają ir REST serviso sąsają su vidine logika.

WEB portalo komponentų integravimo procesas schema pateikiama žemiau:



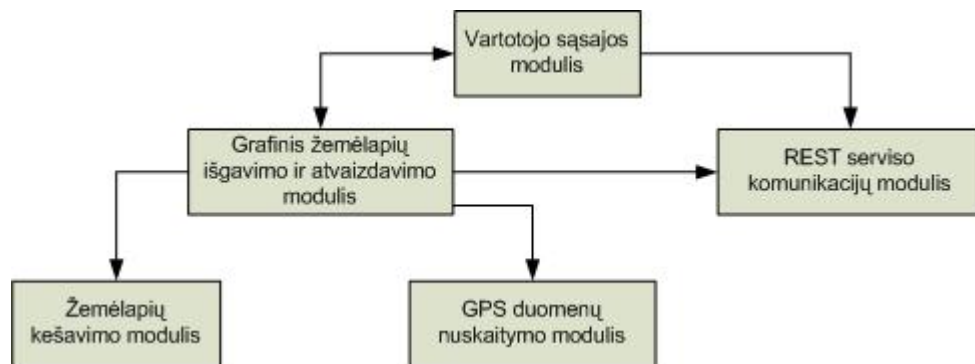
13 pav. WEB portalo integracinio testavimo schema

Grafike pavaizduoti sunumeruoti apskritimai atitinka atskirus integravimo testavimo etapus, kuomet palaipsniui yra suintegruojama sistema. Iš esmės integravimo tvarka šioje vietoje jokio skirtumo neturi, nes tiek grafinė WEB vartotojo sąsaja, tiek WEB serviso modulis naudoja tą pačią vidinę logiką, o tarpusavyje jie jokio sąryšio neturi.

Schemoje pateiktą grafinę vartotojo sąsają sudaro sekantys pagrindiniai komponentai:

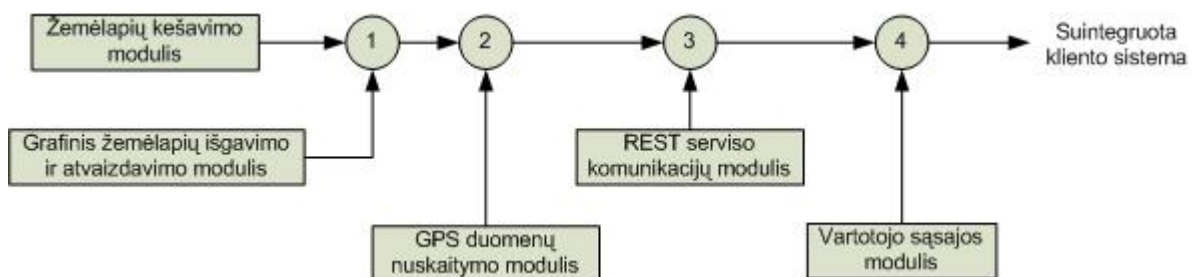
- Registracijos prie sistemos ir profilio redagavimo komponentas
- Prisijungimo prie sistemos komponentas
- Profilio peržiūros komponentas
- Profilio peržiūros/redagavimo komponentas
- Vartotojų paieškos vykdymo komponentas
- Galinčių stebėti vartotojų sąrašo komponentas
- Vartotojo išsaugotų maršrutų sąrašo komponentas
- Sistemos vartotojų sąrašo komponentas
- Besiregistruojančiųjų vartotojų sąrašo komponentas

Kliento dalyje integravimo testavimas taikomas apjungiant modulius, sudarytus iš Java klasių. Testuojamos sritys atspindi žemiau pateiktoje modulių sąryšių schemoje, kur kiekviena rodyklė atitinka atskirą testuojamą sritį:



14 pav. Integracijos testavimo modulių sąryšių schema

Kliento dalyje integravimo testavimas bus vykdomas apjungiant modulius sekančia tvarka:



15 pav. Klientinės PĮ integracinio testavimo schema

Iš tiesų, tokiam testavimui atlikti reikalingos specifinės sąlygos. Pavyzdžiui, kad iš GPS įrenginio nuskaityti duomenys reikia būti atviroje aplinkoje (lauke), nes GPS įrenginys uždaroje patalpose neveikia. REST serviso komunikacijų modulio testavimui reikalingas interneto ryšys bei veikiantis REST serviso paslaugą teikiantis serveris. Šie du moduliai komplikuoja testavimo procesą, todėl jų testavimas atliekamas atskirai, o integravimo testavimo metu, juos pakeičiame atitinkamais skeletais (angl. *stub*), kurie dirbtinai simuliuoja tikrųjų modulių veikimą.

3.3.7 Priėmimo testavimas

Šioje testavimo stadijoje atliekamas bendras visos sistemos testavimas ir tikrinamas sistemos atitikimas dokumentacijai. Testavimas atliekamas pagal tam tikrus scenarijus, parengtus atsižvelgiant į sistemai keliamus reikalavimus. Testavimo scenarijų tikslas – įvertinti sistemos ir jos charakteristikų atitikimą specifikacijai bei nustatyti sistemos tinkamumą užsakovui. Tam tikri testavimo scenarijai parengti ir tam, kad būtų galima įvertinti ir nefunkcinius sistemos reikalavimus, tokius kaip vartotojo sąsajos patogumas ir suprantamumas vartotojui.

Testavimas atliekamas kartu su sistemos užsakovu. Klientinė sistemos dalis įdiegiama į delninį kompiuterį *Fujitsu-Siemens N560 Pocket Loox*, o serverinė sistemos dalis – serveryje su *Linux OS*.

Šiame etape testuojamos abi sistemos dalys (serverio dalis ir kliento dalis). Testavimo metu tikrinamas vartotojo sąsajos bei sistemos funkcionalumo priimtinumai, sistemos dalių komunikavimo galimybės ir greičiai, vartotojo sąsajos patogumas ir suprantamumas.

Iškilus nesklandumams, visos problemos dokumentuojamos, o vėliau analizuojamos ir, atsižvelgiant į galimybes, bandomos spręsti.

3.3.8 Aukšto lygio testavimas

Aukšto lygio testavimo metu atliekami įvairaus pobūdžio sistemos testavimai, kurių metu bandoma įvertinti tokias sistemos charakteristikas kaip patikimumas, saugumas, greitaveika, galimybė atstatyti sistemą įvykus kritinei klaidai. Vertinamos charakteristikos:

✓ Patikimumas – tai sistemos gebėjimas funkcionuoti nustatytą laikotarpį, tolerantiškumas klaidoms ir pan. Projektuojant sistemą nebuvo pateikta reikalavimų dėl sistemos funkcionavimo nustatytą laikotarpį. Buvo numatyta, kad įvykus ne kritinei klaidai, sistema turi apie tai informuoti vartotoją. Specialus testavimas šiuo atveju nebus taikomas. Sistemos darbas bus stebimas kitų testavimų metu.

✓ Saugumas – šio testavimo metu testuojamas sistemos saugumas – tikrinamos galimybės prisijungti prie sistemos su klaidingais prisijungimo duomenimis bei prieiti ir valdyti resursus, kai tam neturima reikiamų teisių. Testavimo metu testuojamos vartotojų autorizacijos galimybės ir vartotojų galimybės prieiti prie tam tikrų sistemos dalių bei resursų. Testavimą galima suskirstyti į kelis etapus:

1. Pirmiausiai testuojamas prisijungimas prie sistemos. Šiuo atveju bandoma prisijungti prie sistemos pateikiant klaidingus prisijungimo duomenis (prisijungimo vardą ir slaptažodį).
2. Antrame etape bandoma patikrinti, ar vartotojai gali prieiti prie duomenų neturėdami tam tinkamų teisių, pavyzdžiui, prie kito vartotojo viešai neskelbiamo maršruto.
3. Trečiame etape bandoma patikrinti, ar vartotojai gali atlikti operacijas, kurioms jie neturi reikiamų teisių, pavyzdžiui, keisti duomenis apie kitą sistemos vartotoją arba koreguoti kito sistemos vartotojo maršrutą apibūdinančią informaciją.

✓ Greitaveika – tai sistemos veikimo ir duomenų perdavimo sparta. Testavimo metu tikrinamas duomenų perdavimo greitis tarp serverio ir kliento bei duomenų atvaizdavimas žemėlapiuose. Atliekami du testavimai, kurių metu tikrinama, ar sistema pakankamai greitai atlieka duomenų perdavimus tarp serverio ir kliento, ir ar pakankamai greitai vykdomas duomenų atvaizdavimas žemėlapyje.

✓ Atstatomumas – tai sistemos atstatymas po kritinės klaidos arba sistemos „lūžimo“. Kadangi reikalavimų šioms sistemos charakteristikoms nebuvo iškelta, tai specialus testavimas šiuo atveju nėra vykdomas. Sistemos darbas bus stebimas kitų testavimų metu.

✓ Stresinis testavimas – tai sistemos testavimas esant didelėms apkrovoms, pavyzdžiui, kai su sistema vienu metu dirba daug vartotojų. Tokio pobūdžio testavimo, kuris atspindėtų sistemos elgesį realiomis sąlygomis, praktiškai atlikti neįmanoma dėl resursų ir laiko stokos.

3.3.9 Testavimo rezultatų kaupimas

Testavimo metu momentiniai testavimo rezultatai bus pateikiami testuotojui ekrane bei registruojami testavimo pranešimų žurnale (angl. „log“). Tokio žurnalo įrašo pavyzdys pateikiamas žemiau (įrašas paimtas iš vieno testavimo):

7 lentelė. Testavimo rezultatų žurnalo pavyzdys

Data (YYYY-MM-DD)	Laikas (HH:MM:SS)	Testuojamo vieneto (klasės) pavadinimas	Testo pavadinimas	Sėkmingai įvykdytas	Testavimo pranešimas
2007-10-17	14:57	Database Access	DB prisijungimo testas	Taip/Ne	Testas įvykdytas sėkmingai

Testavimo metu aptiktos ir išanalizuotos klaidos bus saugomos Excel formato faile. Šio failo pavyzdys pateikiamas žemiau:

	A	B	C	D	E
1	Testuojamo vieneto pavadinimas	Testo pavadinimas	Testavimo atvejų kiekis	Atrasta klaidų	Testavimo pastabos
2	CoordinateTransformer	Ilgumos ir platumos koordinačių transformacija į planimetrinę koordinačių sistemą.	6	Ne	Testavimas vykdytas tikrinant leistinas paklaidas, susidariusias transformuojant tuos pačius koordinačių rinkinius į planimetrinę sistemą ir atgal.

16 pav. Testavimo metu aptiktų ir išanalizuotų klaidų saugojimo failo pavyzdys

3.3.10 Testavimo rezultatai

Šiame skyriuje pateikiami testavimo, vykdyto *JUnit* karkaso priemonėmis, rezultatai ir išvados. Pradinio testavimo metu buvo testuojamos 4 pagrindinės klasės. Jų testavimo rezultatai pateikiami žemiau.

8 lentelė. Vienių testavimo rezultatai

Testuojamo vieneto (klasės) pavadinimas	Testo pavadinimas	Testavimo atvejų kiekis	Atrasta klaidų	Testavimo pastabos
CoordinateTransformer	Ilgumos ir platumos koordinačių transformacija į planimetrinę koordinačių	6	Ne	Testavimas vykdytas tikrinant leistinas paklaidas, susidariusias transformuojant tuos pačius koordinačių rinkinius į planimetrinę sistemą ir atgal.

	sistema.			
Deserializer	Transportavimo struktūrų deserializacija.	8	Taip	Testavimas vykdytas paduodant pilnus ir nepilnus XML duomenis.
Serializer	Transportavimo struktūrų serializacija.	5	Ne	Testavimas vykdytas serializuojant objektus su priskirtomis ir nepriskirtomis skirtingų atributų reikšmėmis.
PhoneNumberValidator	Telefono numerio patikra.	13	Ne	Testavimo atvejai buvo parinkti pagal nustatytus patikros apribojimus, parenkant skirtingo ilgio ir formato testavimo duomenis.

Pagrindinio testavimo metu buvo sudaryti automatizuoti testai, testuojantys tiek atskirus vienetus, tiek ir jau integruotas posistemas. Viena iš tokių posistemių yra serverio pusėje veikiantis *REST* servisas, per kurį vyksta komunikacija su delninio kompiuterio klientais. Prototipo versijoje šis servisas teikia daugumą pagrindinių funkcijų, kurių testavimo rezultatai pateikti toliau:

9 lentelė. REST serviso testavimo rezultatai

Testuojamo vieneto (metodo) pavadinimas	Testo pavadinimas	Testavimo atvejų kiekis	Atrasta klaidų	Testavimo pastabos
login(username, password)	Prisijungimo metodo testavimas.	7	Taip	Bandyta prisijungti su tais pačiais, jau prisijungusio vartotojo, ir kitais prisijungimo duomenimis.
logout()	Atsijungimo metodo testavimas.	2	Ne	Testuota iškviečiant šį metodą prisijungus ir neprisijungus prie sistemos.
getMyProfile()	Einamojo vartotojo profilio paėmimo metodo testavimas.	2	Ne	-
getUserProfile(username)	Vartotojo profilio paėmimo metodo testavimas.	3	Ne	Bandyta su egzistuojančių ir neegzistuojančių vartotojų vardais.
updateMyProfile	Einamojo vartotojo	4	Taip	Testuota parenkant

Testuojamo vieneto (metodo) pavadinimas	Testo pavadinimas	Testavimo atvejų kiekis	Atrasta klaidų	Testavimo pastabos
(userProfileData)	profilio duomenų atnaujinimas			skirtingas reikšmes ir nepriskiriant būtinų laukų.
getMyObservers()	Einamojo vartotojo stebėtojų sąrašo paėmimo metodo testavimas.	2	Ne	-
getMyObservableUsers()	Einamajam vartotojui galimų stebėti vartotojų sąrašo paėmimo metodo testavimas.	2	Ne	-
addMyObserver(username)	Stebėtojo priskyrimo einamajam vartotojui metodo testavimas.	5	Ne	Bandyta su egzistuojančių ir neegzistuojančių vartotojų vardais bei jau priskirtais.
removeMyObserver(username)	Einamojo vartotojo stebėtojo pašalinimo metodo testavimas.	5	Taip	Bandyta su egzistuojančių ir neegzistuojančių vartotojo stebėtojų vartotojų vardais.
sendCoordinates(lon, lat)	Einamojo vartotojo pozicijos duomenų metodo testavimas.	2	Ne	-
getUsersCoordinates(usernames[])	Einamojo vartotojo stebimų vartotojų pozicijos duomenų paėmimo metodo testavimas.	5	Taip	Testuota paduodant vartotojui leistinų ir neleistinų stebėti vartotojų vardai.
searchUsers(query, from, Count)	Vartotojų paieškos metodo testavimas.	7	Ne	Testuota su įvairiomis paieškos užklausomis bei leistinomis ir neleistinomis parametru <i>from</i> ir <i>count</i> reikšmėmis.
saveMyRoute(routeData)	Einamojo vartotojo maršruto išsaugojimo metodo testavimas.	5	Taip	Testuota su maršrutais, turinčiais daugiau ir mažiau nei 2 poros maršruto koordinačių.
getMyRouteList()	Einamojo vartotojo maršrutų sąrašo	2	Ne	-

Testuojamo vieneto (metodo) pavadinimas	Testo pavadinimas	Testavimo atvejų kiekis	Atrasta klaidų	Testavimo pastabos
	paėmimo metodo testavimas.			
getMyRoute(routeId)	Einamojo vartotojo konkretaus maršruto paėmimo metodo testavimas.	4	Ne	Testuota su vartotojui priklausančių ir nepriklausančių maršrutų identifikatoriais.

Visi aukščiau pateikti REST serviso metodai taip pat buvo testuoti prisijungus ir neprisijungus prie sistemos. Tai atsispindi testavimo atvejų kiekyje – pridėtas papildomas testavimo atvejis.

Šios pagrindinės REST funkcijos taip pat nusako funkcionalumą, prieinamą portalo vartotojo srityje. Svarbu pastebėti, jog, testuojant REST servisą, taip pat dalinai buvo ištestuotas portale teikiamas funkcionalumas, nes tiek servisas, tiek ir portalas naudoja bendrą logiką.

Testavimo metu buvo tai pat buvo ištestuota grafinė vartotojo sąsaja tiek portalo, tiek ir delninio kompiuterio kliento. Šis testavimas dėl savo sudėtingumo buvo vykdomas neautomatizuotu būdu. Testavimo metu dauguma išaiškėjusių klaidų buvo ištaisyta. Tačiau šio testavimo metu taip pat paaiškėjo keletas sistemos trūkumų. Ypač tai susiję su portalo posisteme, kai, paspaudus naršyklės atnaujinimo (angl. „refresh“) mygtuką, tam tikrais atvejais yra atkartojama prieš tai vykdyta operacija. Dėl to atitinkamai duomenų bazėje yra gaunami duomenų dublikatai. Ši išaiškėjusį sistemos trūkumą numatyta ištaisyti galutinėje programos versijoje.

3.4. Kokybės analizė

Šioje dalyje pateikiami programų sistemos kokybės vertinimo tikslai ir kriterijai, kokybės užtikrinimo ir vertinimo procesas, vertinimo rezultatai ir išvados.

3.4.1 Kokybės vertinimo tikslai

Programų sistemos kokybės vertinimui keliami tikslai:

- ✓ užtikrinti, kad sistema atitiktų specifikaciją;
- ✓ sumažinti klaidų skaičių sistemoje;
- ✓ įvertinti sistemos priimtinumą vartotojui.
- ✓ įvertinti sistemos funkcionavimą įvairiomis sąlygomis;
- ✓ įvertinti sistemos vystymo ir palaikymo galimybes;

3.4.2 Kokybės vertinimo kriterijai

Norint įgyvendinti kokybės vertinimo tikslus bei įvertinti ir užtikrinti kuriamos programų sistemos kokybę, jos vertinimui parinkti tokie kriterijai:

- ***Korektiškumas*** – laipsnis, kuriuo programa atitinka specifikaciją. Ar yra visos numatytos funkcijos?
- ***Efektyvumas*** – kaip efektyviai programa panaudoja išteklius. Sistemos veikimas esant didelėms apkrovoms, kai yra didelis resursų naudojimas ar su sistema vienu metu dirbančių vartotojų skaičius.
- ***Lankstumas*** – kaip lengvai sistema yra pritaikoma konkreitiems poreikiams.
- ***Testuojamumas*** – kaip greitai galima nustatyti, ar sistema veikia teisingai ar ne.
- ***Pakartotinis panaudojimas*** – ar galima sistemos komponentus panaudoti kuriant kitas sistemas arba naujus sistemos modulius.
- ***Saugumas*** – kaip išpildyti sistemai keliami saugumo reikalavimai, tokie kaip autorizacija, resursų prieigos nustatymas ir pan.
- ***Patikimumas*** – laipsnis, kuriuo programa atlieka jai patikėtas funkcijas ir neatlieka jai nebūdingų funkcijų. Tai gali būti laiko tarpas tarp klaidų pasirodymų arba sistemos „lūžimų“.

- ***Sąveikos galimybės*** – kaip lengva programinę sistemą susieti su kitomis programinėmis sistemomis (čia įeina import, export ir t.t.).
- ***Panaudojamumas*** – ar lengva ir patogiu su programa dirbti, ar vartotojai lengvai ir greitai perpranta sistemą.
- ***Palaikomumas*** – pastangos, reikalingos aptikti ir ištaisyti programos defektus. Čia taip pat vertinamos pastangos, kurios reikalingos, norint įdiegti programos pakeitimus (atnaujinimas pagal kintančius reikalavimus).

3.4.3 Kokybės užtikrinimo ir vertinimo procesas

Kokybės užtikrinimo ir vertinimo procesą sudaro tam tikri etapai, kurie buvo atliekami vykdant projektą. Toliau pateikiami šių etapų aprašymai.

Sistemos specifikacijų peržiūros

Sistemos specifikacijų peržiūros metu buvo stengiamasi aptikti galimas sistemos logikos, funkcionalumo, architektūros bei kitas klaidas. Šios peržiūros metu buvo taisomi neatitikimai.

Šiame etape buvo peržiūrimos šios sistemos specifikacijos:

- ✓ reikalavimų specifikacija
- ✓ architektūros specifikacija
- ✓ detalios architektūros specifikacija
- ✓ vartotojo dokumentacijos
- ✓ testavimo plano peržiūra

Testavimas

Testavimas yra atskiras šios programų sistemos kūrimo etapas. Jis labai svarbus tiek vertinant, tiek ir užtikrinant sistemos kokybę. Testavimo metu ieškoma klaidų programų sistemos kode, sistemos funkcionavime. Testavimo metu taip pat tikrinamas sistemos funkcionavimas įvairiomis sąlygomis, pavyzdžiui esant įvairiam serverio apkrautumui ar resursų poreikiui, testuojama vartotojo sąsają bei atliekami kiti sistemos testavimai. Sistemos testavimui yra naudojami įvairūs sistemos testavimo metodai, pavyzdžiui, vienetų testavimas, integracinis testavimas ir kiti.

Taigi bendru atveju testavimas leidžia ne tik įvertinti įvairias sistemos kokybės charakteristikas, bet ir dalinai užtikrinti aukštesnę programų sistemos kokybę.

Klaidų taisymas

Kaip ir testavimas, šis etapas yra atskiras programų sistemos kūrimo etapas ir turi didelę įtaką sistemos kokybės užtikrinimui ir įvertinimui. Jis leidžia įvertinti sistemos palaikomumą, defektų pašalinimo sąnaudas, o taip pat gali įtakoti ir kitus sistemos kokybę nusakančius kriterijus.

3.4.4 Kokybės vertinimas

Vertinant sistemą pagal pasirinktus kriterijus, naudojama skaitinė įvertinimo išraiška. Galimas balų skaičius kiekvienam kriterijui – nuo 0 iki 10.

Korektiškumas

Sistemos atitikimo specifikacijai patikros metu buvo nustatytas jos atitikimo lygis. Buvo įsitikinta, jog realizuotos sistemos funkcijos atitinka specifikaciją. Sistema leidžia stebėti savo ir kitų vartotojų pozicijas, registruoti, saugoti ir peržiūrėti maršrutus. Įvertinimas – 10.

Efektyvumas

Testavimo metu buvo bandomas sistemos veikimas esant didesniems nei vidutiniai resursų poreikiams. Buvo bandomas dvidešimties ir daugiau vartotojų pozicijos stebėjimas realiu laiku, kas parodė sistemoje realizuoto REST serviso efektyvumo lygį sistemos komunikacijų su serveriu srityje. Atlikus testavimą, buvo įsitikinta, jog sistema veikia pakankamai efektyviai ir yra pilnai tinkama praktiniam jos naudojimui. Kadangi yra įmanoma dar labiau padidinti šios komunikacijų su serveriu dalies efektyvumą panaudojant tiesiogiai *socket* priemones ir apeinant HTTP protokolą, galutinis įvertinimas – 9.

Lankstumas

Kadangi sistema apima bendrąsias funkcijas ir tiksliai nenurodo jos taikymo srities, ji yra daugiau bendrojo pobūdžio. Taigi ši sistema nesunkiai gali būti pritaikyta specifinėje aplinkoje. Tačiau dėl šios abstrakcijos ji neapima daugelio kitų specifinių konkrečios srities poreikių dėl to juos tenka realizuoti patiems vartotojams savais metodais. Galutinis įvertinimas – 8.

Testuojamumas

Buvo vertinamos darbo ir laiko sąnaudos, kurių reikia norint ištestuoti sistemos veikimą ir įvertinti, ar sistema veikia taip kaip turėtų, ar nėra klaidų duomenų apdorojime ir kitose funkcijose. Sistemoje naudojamos technologijos (*Tapestry, J2ME*) ir jų prigimtis (ji sudaryta iš kelių tinklo mazgų ir veikia tinkle) šiek tiek komplikuoja automatizuotą jos testavimą. Nemažai testavimo darbų, tarp jų vartotojo sąsajos testavimą, tenka vykdyti rankomis, taigi testuojamumo įvertinimas – 7.

Pakartotinis panaudojimas

Buvo vertinamas sistemos modulių realizacijos sudėtingumas ir galimybė juos pritaikyti kuriant kitas sistemas arba modulius. Suprojektuota sistemos architektūra yra pakankamai lanksti, todėl ir atskiri moduliai yra prasmingi ir nesunkiai pakartotinai panaudojami. Be to sistema realizuota naudojant objektinę programavimą. Galutinis įvertinimas – 10.

Saugumas

Buvo tikrinamas sistemai keliamų saugumo reikalavimų realizavimas. Šiuo atveju buvo tikrinama sistemos autentifikacijos ir autorizacijos mechanizmo realizacija bei resursų prieigos ir kontrolės reikalavimų išpildymai. Kadangi saugomos informacijos pobūdis nėra reikalaujantis kritinės apsaugos, o ir testavimo metu nebuvo aptikta jokių saugumo spragų, galutinis įvertinimas – 10.

Patikimumas

Buvo tikrinama, ar sistema atlieka visas numatytas funkcijas bei neatlieka nebūdingų ir nepageidaujamų funkcijų. Neigiamą įtaką turi prototipo testavimo metu aptiktas nepageidautinas sistemos portalo elgesys tam tikrose vietose nuspaudus atnaujinimo („*Refresh*“) mygtuką naršyklėje, kuomet duomenų bazėje atsiranda duomenų dublikatai. Sistemos testavimo ir bandymo periodo metu, sistema nė karto nebuvo „nulūžusi“. Galutinis įvertinimas – 9.

Sąveikos galimybės

Vertinant sistemą pagal šį kriterijų, buvo tiriama galimybė susieti sistemą su kitomis programų sistemomis. Šiuo atveju buvo vertinama galimybė importuoti ir eksportuoti duomenis kitais formatais. Kadangi sistema realizuoja maršrutų eksportavimą CSV ir GPX formatais, galima nesunkiai įvykdyti maršrutų migravimą į kitas sistemas, palaikančias šiuos formatus.

Tačiau kiti sistemos saugomi duomenis gali būti eksportuojami tik tiesiai iš MySQL duomenų bazės, taigi gali būti reikalingi papildomas šių duomenų konvertavimas, todėl įvertinimas – 9.

Panaudojamumas

Buvo vertinama, kaip lengva naudotis sistema ir jos teikiamomis funkcijomis. Sistemos portalo dalyje realizuota labai paprasta meniu principu sudaryta vartotojo sąsaja. Vienu metu pateikiamos informacijos kiekis neužgožia vartotojo, o pavadinimai ir paaiškinimai yra trumpi ir lakoniški. Kliento pusėje taip pat kiek įmanoma buvo stengiamasi sudaryti patogią vartotojo sąsają, tačiau dėl J2ME technologijos ir delninio kompiuterio apribojimų tam tikri elementai negali būti realizuoti patogiausiu būdu, todėl šioje pusėje sistemos panaudojamumas yra šiek tiek labiau komplikuoatas. Nors kai kurios realizuojamos funkcijos yra sudėtingos, tačiau vartotojo akimis jų įvykdymas yra paprastas ir nereikalauja papildomų pastangų, todėl galutinis įvertinimas – 9.

Palaikomumas

Buvo vertinamos sąnaudos reikalingos ištaisyti sistemoje aptiktą defektą. Žinoma šios sąnaudos priklauso nuo defekto pobūdžio, tačiau dėl santykinio sistemos paprastumo, dažniausiai šie defektai yra nedideli ir nereikalauja esminių architektūrinių pakeitimų. Bendru atveju defektų šalinimas sistemoje yra pakankamai nesudėtingai realizuojamas, todėl įvertinimas – 9.

3.4.5 Kokybės vertinimo rezultatai

Kokybės vertinimo rezultatų lentelė:

10 lentelė. Kokybės vertinimo rezultatai

Nr.	Kriterijus	Įvertinimas
1	Korektiškumas	10
2	Efektyvumas	9
3	Lankstumas	8
4	Testuojamumas	7
5	Pakartotinis panaudojimas	10
6	Saugumas	10
7	Patikimumas	9
8	Sąveikos galimybės	9
9	Panaudojamumas	9
10	Palaikomumas	9

3.4.6 Išvados

Kokybės vertinimo metu buvo apibrėžti šiai sistemai taikytini vertinimo kriterijai ir pagal šiuos kriterijus, preliminariai buvo įvertinta jos kokybė. Apibendrinant šalia kiekvieno vertinimo kriterijaus pateiktas išvadas, matyti, jog sistemos prototipas veikia santykinai greitai ir nesukelia papildomų nesklandumų. Jis yra realizuotas išlaikant paprastumo principą ir juo yra labai paprasta naudotis, kas yra labai svarbu sistemos vartotojui, todėl bendroji kokybė yra pakankamai aukšta.

4. Tiriamoji eksperimentinė dalis

4.1. Tikslas

Eksperimento tikslas – išsiaiškinti, kiek tiksliai veikia J2ME emuliatorius ir palyginti sistemos veikimą naudojant emuliatorių su sistemos veikimu realiame įrenginyje.

4.2. Tyrimo metodika

Eksperimentui atlikti parenkami tokie eksperimentiniai atvejai:

- 1) Aritmetinis algoritmas – vykdomi matematiniai skaičiavimai pagal tam tikrą matematinį algoritmą.
- 2) Ciklo vykdymas – atliekamas ciklo, įrašancio duomenis į masyvą ir atliekančio masyvo rikiavimą, vykdymas.
- 3) Teksto apdorojimas – sukuriama tekstinė eilutė, kuri užpildoma tam tikru tekstu, o tada eilutėje ieškoma tam tikra teksto dalis.
- 4) Loginės operacijos – sukuriamas loginių verčių masyvas, kuris analizuojamas naudojant ciklus.
- 5) Failo nuskaitymas – nuskaitomi duomenys iš failo.

Eksperimentui atlikti sukuriamos specialios programos, kurios realizuoja atskirus eksperimentinius atvejus. Šios programos vykdymo metu fiksuoja sistemos veikimo greitį ir rezultatus.

Eksperimentinius atvejus atitinkančios programos yra vykdomos naudojant emuliatorių bei realų įrenginį. Eksperimento metu matuojamas laikas, per kurį yra įvykdomas kiekvienas eksperimentinis atvejis bei lyginami abiem atvejais rezultatai.

4.3. Priemonės

Eksperimento metu naudojamos priemonės:

- Delninis kompiuteris Fujitsu-Siemens N560 Pocket Loox:
 - Operacinė sistema – Windows Mobile 5.0.
 - JVM (Java Virtual Machine) - J2ME (CLDC-1.1, MIDP-2.0) specifikaciją atitinkanti IBM J9 JVM.
 - Procesorius – PXA270 - 624 MHz.
 - Operatyvinė atmintis – 50 MB.
- Standartinis kompiuteris:
 - Operacinė sistema – Windows XP.
 - Emuliatorius – Sun Java (TM) Wireless Toolkit 2.5.2 for CLDC.
 - Procesorius – AMD Athlon XP 1600+ (1.4 GHz).
 - Operatyvinė atmintis – 512 MB.

4.4. Rezultatai

Aritmetinis algoritmas

11 lentelė. Aritmetinio algoritmo vykdymo eksperimento rezultatai

Testas	Emuliatorius	PDA	Skirtumas, %
1	31	42	135,5
2	47	97	206,4
3	47	97	206,4
4	31	97	312,9
5	47	102	217,0
6	32	101	315,6
7	46	102	221,7
8	32	107	334,4
9	47	101	214,9
10	31	102	329,0
Vidutiniškai	39,1	94,8	249,4

Kaip matyti iš 11 lentelėje pateiktų rezultatų, aritmetinių funkcijų vykdymas realioje aplinkoje yra atliekamas vidutiniškai 2,5 karto ilgiau.

Ciklo vykdymas

12 lentelė. Ciklo vykdymo eksperimento rezultatai

Testas	Emulatorius	PDA	Skirtumas, %
1	31	48	154,8
2	31	49	158,1
3	32	49	153,1
4	31	50	161,3
5	31	48	154,8
6	31	62	200,0
7	32	48	150,0
8	31	49	158,1
9	31	45	145,2
10	31	39	125,8
Vidutiniškai	31,2	48,7	156,1

Kaip matyti iš 12 lentelėje pateiktų rezultatų, ciklų vykdymas realioje aplinkoje yra atliekamas vidutiniškai 1,5 karto ilgiau.

Teksto apdorojimas

13 lentelė. Teksto apdorojimo eksperimento rezultatai

Testas	Emulatorius	PDA	Skirtumas, %
1	125	35	357,1
2	125	95	131,6
3	141	35	402,9
4	156	35	445,7
5	125	35	357,1
6	141	35	402,9
7	156	82	190,2
8	156	34	458,8
9	125	35	357,1
10	141	35	440,6
Vidutiniškai	139,1	45,3	354,4

Kaip matyti iš 13 lentelėje pateiktų rezultatų, teksto apdorojimas realioje aplinkoje yra atliekamas vidutiniškai 3,5 karto greičiau.

Loginės operacijos

14 lentelė. Loginių operacijų vykdymo eksperimento rezultatų lentelė

Testas	Emuliatorius	PDA	Skirtumas, %
1	16	21	131,3
2	31	23	74,2
3	15	22	146,7
4	16	23	143,8
5	31	22	71,0
6	32	22	68,8
7	15	22	146,7
8	16	22	137,5
9	31	24	77,4
10	16	22	137,5
Vidutiniškai	21,9	22,3	113,5

Kaip matyti iš 14 lentelėje pateiktų rezultatų, loginių operacijų vykdymas realioje aplinkoje yra atliekamas tik šiek tiek lėčiau – vidutiniškai 13,5 %.

Failo nuskaitymas

15 lentelė. Failo nuskaitymo eksperimento rezultatai

Testas	Emuliatorius	PDA	Skirtumas, %
1	125	94	1413,8
2	125	78	1256,4
3	141	78	1184,6
4	156	94	1007,4
5	125	78	1237,2
6	141	110	890,0
7	156	78	1138,5
8	156	93	1123,7
9	125	94	979,8
10	141	78	1201,3
Vidutiniškai	139,1	87,5	1143,3

Kaip matyti iš 15 lentelėje pateiktų rezultatų, failo nuskaitymas realioje aplinkoje yra atliekamas vidutiniškai 11,5 karto lėčiau.

4.5. Išvados

Kaip matyti iš 4.4 skyriuje pateiktų eksperimento rezultatų, programos vykdymas naudojant emuliatorių tik vieninteliu atveju atitinka sistemos vykdymą realiame įrenginyje, t.y. loginių operacijų apdorojimo atveju. Visais kitais atvejais rezultatai rodo, jog programos vykdymas naudojant emuliatorių žymiai skiriasi nuo jos vykdymo realiame įrenginyje:

- aritmetinių operacijų vykdymo greitis skiriasi 2,5 karto;
- ciklų vykdymo greiti – 1,5 karto;
- teksto apdorojimo – 3,5 karto;
- failo nuskaitymo – net 11,5 karto.

Iš to galima daryti išvadą, jog sistemos kritinių dalių testavimui negalima apsiriboti vien emuliatoriumi, bet būtina sistemą testuoti realioje aplinkoje.

5. Išvados

- Darbe išanalizuotos plačiai taikomos pozicionavimo sistemos ir išskirtas pagrindinis tokių sistemų funkcionalumas.
- Palygintos pozicionavimo technologijos, jų privalumai ir trūkumai viena kitos atžvilgiu. Objekto pozicijos nustatymo GSM tinkle metodai labai praverčia stebimiems objektams esant uždaroje patalpose, tačiau didelį tikslumą turinti GPS technologija yra nepakeičiama lauko sąlygomis.
- Taupant kuklius delninio kompiuterio resursus ir mažinant persiunčiamų duomenų kiekį, tikslinga naudoti plintančią Web servisų atmainą – REST servisus, o naujai atsiųstus žemėlapių fragmentus – kešuoti.
- Pozicijos koordinatės gaunamos iš GPS įrenginio priklauso sferinei koordinačių sistemai, o žemėlapiai – stačiakampe, todėl, pozicijos taško atvaizdavimui ant žemėlapiro atlikti, būtina pozicijos koordinatės transformuoti į stačiakampę koordinačių sistemą panaudojant atvirkštinę Gudermano (Gudermannian) funkciją.
- Mercator projekcijos pagrindu sudaryto žemėlapiro mastelis yra stipriai iškraipytas, ypač tolstant nuo pusiaujo link ašigalių. Tačiau ši projekcija puikiai tinkama interaktyvių internetinių žemėlapių sudarymui, kuriais patogiu manipuliuoti, todėl neretai yra taikoma geografinėse informacinėse sistemose.
- Išskirtos dažnai praktikoje pasitaikančios paskirstytų sistemų projektavimo, realizavimo ir testavimo klaidos, dėl kurių gali nepavykti sėkmingai ir laiku įgyvendinti projektą arba kritinės sistemos dalys gali būti netinkamos naudoti realiuose įrenginiuose. Pasiūlytas būdas, kaip išvengti tokių klaidų ir sumažinti riziką, kad projektas gali nepavykti.
- Eksperimentiškai įrodyta, jog paskirstytų sistemų klientų, skirtų mobiliems įrenginiams (mobiliems telefonams ir PDA), kūrimo ir testavimo procesas neturi apsiriboti vien tik emuliatoriais, kuriuose vykdomos programos elgesys ir charakteristikos gali labai skirtis nuo realioje įrenginyje vykdomos programos.
- Suprojektuota ir realizuota daugiavartotojiška pozicionavimo sistema, leidžianti stebėti savo ir kitų sistemos vartotojų pozicijas realiu laiku, registruoti, saugoti ir peržiūrėti maršrutus bei keisti maršrutais tarp sistemos vartotojų.

6. Literatūra

- [1] Binder, R., V. Testing object-oriented systems. Addison-Wesley. 2003.
- [2] Brainerd, J.; Pang, A. Floating ring: a new tool for visualizing distortion in map projections. Computer Graphics International, 1998. Proceedings 22-26 June 1998 Page(s):466 – 480.
- [3] Dale DePriest, [Interaktyvus]. Prieiga per internetą: <http://gpsinformation.org/dale/nmea.htm>.
- [4] R.T. Fielding. Architectural Styles and the Design of Network-Based Software Architectures. doctoral dissertation, Dept. of Computer Science, Univ. of Calif., Irvine, 2000.
- [5] G. Goth. Critics Say Web Services Need a REST. IEEE Distributed systems online, IEEE Computing Society, Dec. 2004, Vol. 5(12).
- [6] Yamamichi, N.; Ozeki, T.; Yokochi, K.; Tanaka, T. The evaluation of new software developing process based on a spiral modeling. Global Telecommunications Conference, 1996. GLOBECOM '96. 'Communications: The Key to Global Prosperity, 18-22 Nov. 1996, Vol. 3, p.2007 - 2012.
- [7] Liutkauskas, V. Mobiliųjų technologijų įvadas. Telekomunikacijų tinklai: paskaitos pranešimo medžiaga. Kaunas, 2005, p. 42.
- [8] M. Kennedy, S Kopp - Understanding Map Projections, 2000.
- [9] Petkevičius Evaldas. GPS technologija ir panaudojimas. Iš Elektronika [interaktyvus]. 2004, rugsėjis [žiūrėta 2006-10-15]. Prieiga per internetą: <http://www.elektronika.lt/>.
- [10] Ramamoorthy, C.V. Distributed techniques in software systems integration. Distributed Computing Systems, 28-30 Aug. 1995. p. 252-256.
- [11] Sommerville, I. Software Engineering. 6th edition. Addison-Wesley. 2000.
- [12] Voisard, A; Schiller, J., H. Location-based Services. Morgan Kaufmann Pub. 2004.
- [13] S. Vinoski. Demystifying RESTful Data Coupling. IEEE Internet Computing online, IEEE Computing Society, Mar.-Apr. 2008, Vol. 12(2), p. 87-90.
- [14] S. Vinoski. REST eye for the SOA Guy. IEEE Internet Computing online, IEEE Computing Society, Jan.-Feb. 2007, Vol. 11(1), p. 82-84
- [15] Weisstein, Eric W. "Mercator Projection." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/MercatorProjection.html>
- [16] Weisstein, Eric W. "Inverse Gudermannian." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/InverseGudermannian.html>.

7. Terminų žodynas

Ad-hoc – terminas, nusakantis specifinį problemos sprendimą, sudarytą konkrečiam uždaviniui spręsti.

Com port – Nuosekloji kompiuterio sąsajos jungtis, skirta duomenų mainams su išoriniais įrenginiais realizuoti.

JVM (Java Virtual Machine) – Java kalbos virtualioji mašina, kuri interpretuoja ir vykdo Java kompiliatoriumi sukompiliuota „Java baitų kodą“. Šiuo atveju kalba eina apie *J2ME JVM*.

J2ME (Java Platform, Micro Edition) – tai specifikacija, apibrėžianti programavimo priemonių rinkinį, paremtą Java programavimo kalba ir skirtą programinės įrangos, orientuotos į mobiliuosius įrenginius, kūrimui. Ji apima *CLDC (Connected Limited Device Configuration)* ir *MIDP (Mobile Information Device Profile)* specifikacijas.

GPRS (General Packet Radio Service) – Specifinis mobiliojo interneto ryšio tipas mobiliajame įrenginyje.

GPS (Global Positioning System) – šiuo metu tai vienintelė pilnai funkcionuojanti globali palydovinė navigacijos sistema.

GSM (Global System for Mobile communications) – Globalus mobilių telefonų ryšio standartas. Yra labiausiai paplitęs mobiliųjų telefonų ryšio standartas pasaulyje.

LBS (Location-Based Service) – pozicionavimu pagrįstos paslaugos. Šis terminas apima sistemas, kurios apjungia tradicines paslaugas su vartotojo pozicijos nustatymu.

NMEA 0183 – GPS įrenginiuose naudojamas komunikavimo standartas – specifinis komunikavimo protokolas.

PDA (Personal digital assistant), Pocket PC, „Delninkas“ – tai rankinis, dažniausiai dėlne telpantis, kompiuteris.

Tapestry Framework – atviro kodo Java programavimo kalba paremtas kūrimo šablonas, skirtas kurti sudėtingas *WEB* taikomąsias programas.

XML – išplėstinė žymių kalba – duomenų formatas struktūrizuotų dokumentų keitimuisi tinkle (*eXtensible Markup Language*).

8. Priedai

Daugiavartotojiška pozicionavimo sistema

Andrius Blažinskas, Rytis Rudelis

KTU Informatikos fakultetas, Studentų g. 50, LT-51368 Kaunas, Lietuva

Šiame straipsnyje aptarsime realizuotą daugiavartotojišką pozicionavimo sistemą, naudojančią GPS technologiją.

Detaliau paaiškinsime jos veikimo principus ir architektūrą. Išdėstysime pozicijos koordinatžių nuskaitymo iš GPS įrenginio ir žemėlapių fragmentų nuskaitymo iš internetinės žemėlapių saugyklos metodus. Aptarsime naudojamą WEB servisų alternatyvą – REST servisu. Taip pat paminėsime keletą kitų panašių programinės įrangos sistemų ir jų naudojamų GPS ir GSM technologijų privalumus ir trūkumus.

1. Įvadas

Pozicijos nustatymo problema jau nuo senų laikų žmonėms buvo labai aktuali. Ji pakankamai efektyviai išspręsta tik XX a. pabaigoje. Tą sėkmingiausiai padarė JAV Gynybos Departamentas, sukūręs palydovinę radijo navigacinę sistemą – GPS (*Global Positioning System*).

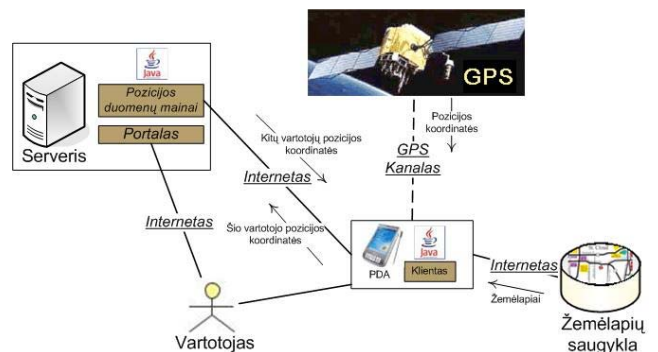
Kita pozicijos nustatymo technologija atsirado sparčiai išpopuliarėjus mobiliesiems telefonams. Ji remiasi mobilaus telefono pozicijos nustatymu GSM (*Global System for Mobile Communications*) tinkle, panaudojant tokius metodus kaip artimiausios bazinės stoties nustatymas (*Cell identification* – Cell-ID), signalo atėjimo kampo metodas (*Angle of Arrival* – AOA), radijo signalo atėjimo laiko metodas (*Time of Arrival* – TAO) ar patobulintas laikų skirtumo metodas (*Enhanced Observed Time Difference* – E-OTD) [4][5].

GPS tapus visuotinai prieinamai ir atpigus GSM paslaugoms, buvo pradėtos kurti įvairios programinės įrangos (toliau - *PI*) sistemos, naudojančios šias technologijas ir skirtos eiliniams vartotojams [1]. Pavyzdžiui GPS naudojančios *TomTom*, *Frotcom*, *GPSPursuit* ir kt., GSM naudojanči *Locator* ir kt. Tokio pobūdžio sistemų taikymo sritis nuolat plečiasi ir jos plačiai pradedamos naudoti kasdieniniame gyvenime. Būtent paprastam vartotojui yra skirta *Daugiavartotojiška pozicionavimo sistema*, kurios prototipą aptarsime toliau.

2. Sistemos prototipas

2.1 Veikimo principas

Sistemos prototipo veikimas yra paremtas kliento ir serverio principu. Vartotojas naudodamasis delniniame kompiuteryje įdiegta programine įranga (toliau - *klientas*) gali stebėti savo ir kitų vartotojų poziciją. Delninio kompiuterio GPS įrenginys vartotojo pozicijos duomenis gauna iš GPS palydovų ir perduoda klientui. Klientas šiuos duomenis siunčia į serverį, tokiu būdu sudarydamas galimybę juos pasiimti kitiems, tokiu pačiu klientu besinaudojantiems ir atitinkamą leidimą turintiems, vartotojams. Žemėlapius klientas tiesiogiai nuskaity iš lokalaus kešo, arba, jei reikiamo fragmento ten



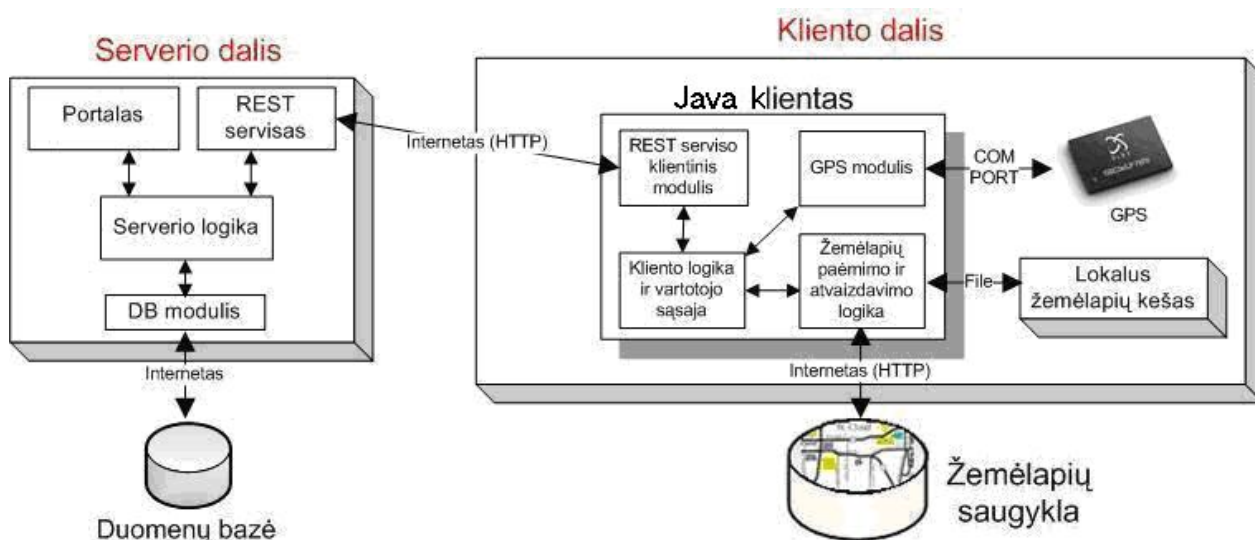
1 pav. – veikimo principas

nėra, iš internetinės žemėlapių saugyklos. Serverio dalyje veikiantis portalas suteikia vartotojui galimybę tvarkyti savo duomenis, įskaitant ir maršrutus, kuriuos jis gali išsaugoti serveryje pasinaudodamas klientu.

Renkantis pozicionavimo technologiją buvo nuspręsta naudoti GPS, kadangi ji dengia visą Žemės paviršių, gali pateikti trimates koordinatas, nustatyti objekto judėjimo kryptį ir greitį. Nors dauguma GPS imtuvų veikia tik lauko sąlygomis, tačiau yra ganetinais tikslūs: standartinis GPS imtuvas gali pateikti pozicijos koordinatas 3 – 15 m tikslumu, o aukštesnio lygio – 1 m tikslumu. GSM pozicionavimo technologija veikia ne tik lauke, bet ir pastatų patalpose, tačiau ji pateikia tik dvimates koordinatas. Šių koordinatų tikslumas yra žymiai mažesnis ir priklauso nuo GSM tinklo struktūros, pozicijos nustatymui naudojamo metodo ir vietovės. GSM atveju paklaida svyruoja nuo 50 m iki 30 km.

2.2 Bendroji architektūra

Kaip ir buvo minėta, šios sistemos architektūra yra paremta kliento ir serverio principu. Serveris vartotojams teikia prieigą prie portalo ir prie REST (žiūrėti 3 skyrių „**REST panaudojimas**“) serviso, naudodamas bendrą serverio logikos modulį, kuris komunikuoja su duomenų baze.



2 pav. – bendroji architektūra

Kliento dalyje pagrindinis vidinės logikos modulis yra glaudžiai susijęs su vartoto sąsąja. Jis naudoja GPS modulį duomenų iš GPS įrenginio nuskaitymui, žemėlapių nuskaitymo ir atvaizdavimo logikos modulį. Su serveriu komunikuoja naudodamas paprastą, bet efektyvų, REST serviso klientinį modulį. GPS modulis koordinatas nuskaitymo iš GPS įrenginio per nuosekliąją sąsajos jungtį (COM port). Žemėlapių atvaizdavimo ir valdymo modulis reikalingus žemėlapių fragmentus nuskaitymo iš failų sistemoje esančio kešo arba iš internetinės žemėlapių saugyklos. Vartotojui nurodžius, naujai parsisiūsti žemėlapių fragmentai gali būti kešuojami.

Viena pagrindinių sistemos dalių – klientas, veikia delniniame kompiuteryje. Tokio kompiuterio našumas lyginant su paprasto stalinio kompiuterio našumu yra labai mažas. Prieigai prie interneto dažniausiai naudojamas bendras paketinis radijo ryšys (*General Packet Radio Service – GPRS*), kur apmokestinamas kiekvienas išsiustas megabaitas. Būtent dėl šių priežasčių bendra sistemos architektūra yra sudaryta taupant resursus ir mažinant persiunčiamų bei naudojamų duomenų apimtį.

2.3 Objekto pozicijos koordinatės ir jų transformacija

Objekto pozicijos duomenis klientas nuskaito iš delninio kompiuterio GPS įrenginio. Šis įrenginys, kaip ir daugelis kitų, naudoja labiausiai paplitusį NMEA 0183 (<http://www.nmea.org/pub/0183/>) komunikavimo standartą. Šis standartas apibrėžia įrenginio per nuoseklaus duomenų perdavimo jungti siunčiamus sakinius ir jų struktūrą. GPS įrenginys formuoja įvairių tipų NMEA 0183 sakinius, tačiau tik dalis iš jų saugo informacija apie objekto poziciją. Tokių pozicijos koordinatės saugančių sakinių pavyzdžiai:

- \$GPGLL, 4916.45, N, 12311.12, W, 225444, A, *31
- \$GPRMC, 123519, A, 4807.038, N, 01131.000, E, 022.4, 084.4, 230394, 003.1, W*6A
- \$GPGGA, 123519, 4807.038, N, 01131.000, E, 1, 08, 0.9, 545.4, M, 46.9, M, , *47

Internetu yra nemažai NMEA 0183 sakinių analizės atvirojo kodo bibliotekų, tačiau šiuo atveju reikalingų sakinių kiekis yra labai mažas, todėl realizuotas efektyvus specializuotas šių sakinių analizės modulis.

NMEA 0183 sakiniuose, kartu su visa kita informacija, yra pateikiamos geografinės ilgumos ir platumos koordinatės, kurios priklauso sferinei koordinatėms sistemai. Naudojami žemėlapių, pasiekiami internetinėje žemėlapių saugykloje, yra sudaryti cilindrinės Mercator projekcijos (<http://mathworld.wolfram.com/MercatorProjection.html>) pagrindu. Taigi, naudojamos skirtingos koordinatėms sistemos sukelia problemą išrenkant reikiamą žemėlapių fragmentą ir atvaizduojant pozicijos tašką žemėlapyje. Šiai problemai išspręsti reikalingas koordinatėms pervedimas iš sferinės koordinatėms sistemos į stačiakampę. Tokį pervedimą galima atlikti panaudojant atvirkštinę Gudermano (Gudermannian) funkciją:

$$y = gd^{-1}(\varphi) = \frac{1}{2} \ln \frac{1 + \sin(\varphi)}{1 - \sin(\varphi)}, \text{ kur } \varphi - \text{platumos koordinatė.}$$

Kaip matyti, transformacija yra reikalinga tik platumos koordinatei. Po šios transformacijos pozicijos koordinatės yra suderinamos su žemėlapių koordinatėms sistema ir gali būti panaudojamos tiek reikiamo žemėlapių fragmento koordinatėms išskaičiavimui, tiek ir paties pozicijos taško žemėlapyje atvaizdavimui.

2.4 Žemėlapiai

Objekto pozicijos atvaizdavimui kliente reikalingi žemėlapiai. Turint objekto pozicijos koordinatės, reikiamų žemėlapių fragmentų pirmiausiai ieškoma lokaliame žemėlapių keše. Jei reikiamo fragmento keše nėra, jis imamas iš internetu pasiekiamos žemėlapių saugyklos. Žemėlapių saugykla žemėlapių fragmentus pateikia pagal jų specifines koordinatės, todėl klientas turi mokėti išskaičiuoti reikiamo fragmento koordinatės.

Saugykloje žemėlapiai saugomi kvadratinių, 256x256 taškų dydžio, fragmentų pavidalu, o pilnas iš šių fragmentų sudarytas žemėlapis paremtas Quadtree struktūros pagrindu. Trumpai nusakysime tokios saugyklos žemėlapių fragmentų koordinatėms išskaičiavimą.

Išskaičiavimas pradedamas nuo pradinio fragmento X_0 , kuris paprastai atitinka visą pasaulį. Šis fragmentas yra dalinamas į keturias lygias dalis (3 pav.) ir parenkama viena reikiama dalis. Naujai parinktos dalies atžvilgiu veiksmai yra kartojami tol, kol pasiekiamas reikiamo priartinimo lygio reikiamas fragmentas.



3 pav. – žemėlapių fragmentas

Matematiškai kiekvieno fragmento išskaičiavimą galima apibrėžti taip:

$$\begin{aligned} X_1(x_1; y_1; z_1) &= X_0(2x_0; 2y_0; z_0 - 1) & X_3(x_3; y_3; z_3) &= X_0(2x_0; 2y_0 + 1; z_0 - 1) \\ X_2(x_2; y_2; z_2) &= X_0(2x_0 + 1; 2y_0; z_0 - 1) & X_4(x_4; y_4; z_4) &= X_0(2x_0 + 1; 2y_0 + 1; z_0 - 1) \end{aligned}$$

Kur $X_i(x_i; y_i; z_i)$, $i = \overline{1,4}$ yra 3 pav. pateiktas i -tasis fragmentas: x , y koordinatės ir z – priartinimo lygis.

Kaip ir buvo minėta, saugykloje saugomi žemėlapiai yra sudaryti Mercator projekcijos pagrindu. Ši projekcija yra plačiai taikoma ir gerai žinoma geografijos srities specialistams, tačiau turi esminį trūkumą – stipriai iškraipo žemėlapių mastelį tostant nuo pusiaujo link ašigalių. Kaip pavyzdį galima palyginti Afrika su Grenlandija, kurios Mercator projekcijoje yra beveik tokio paties dydžio, tačiau realybėje Grenlandija yra beveik 14 kartų mažesnė už Afriką. Nepaisant šio trūkumo, ši projekcija yra tinkama interaktyvių internetinių žemėlapių sudarymui, kuriais patogu manipuliuoti, todėl neretai yra taikoma geografinėse informacinėse sistemose (GIS).

3. REST panaudojimas

REST (*REpresentational State Transfer*) – yra tam tikra programinė architektūra, skirta paskirstytoms hiperžiniasklaidos sistemoms, tokioms kaip žiniatinklis. Terminas pirmą kartą paminėtas 2000 metais vieno iš pagrindinių HTTP (*Hypertext Transfer Protocol*) specifikacijos autorių Roy Fielding disertacijoje [2]. Yra suformuluoti pagrindiniai principai, kuriais turėtų remtis REST architektūrą įgyvendinančios programinės realizacijos [6][7].

Sparčiai plintančius REST pagrindu sudarytus servigus galima laikyti tam tikra SOAP (*Simple Object Access Protocol*) Web servisų alternatyva. Pastarieji laikui bėgant tapo sudėtingi ir neefektyvūs, o paprastumu ir efektyvumu pasižymintys REST servigai vis labiau plinta [3]. Tačiau šiandien REST ideologija dar nėra apibrėžta kokio nors standarto ir neretai kyla debatai dėl šios architektūros ir termino panaudojimo. REST terminas dažnai yra naudojamas laisvesne prasme, kalbant apie paprastą sąsają, kuri skirta perduoti specifinius srities duomenis tiesiogiai panaudojant HTTP protokolą ir nenaudojant jokio papildomo pranešimų sluoksnio kaip SOAP. Internete yra daug šaltinių, kurie skelbiasi savo sistemose naudojančius REST architektūrą, tačiau daugelis iš jų yra tik tam tikros jos atmainos, įgyvendinančios tik dalį Roy Fielding suformuluotų REST principų.

Prototipo sistemos REST servigas pagrįstas labai paprastu principu. Jis naudoja HTTP protokolą ir neturi jokio papildomo pranešimų perdavimo sluoksnio. Pranešimai yra perduodami tiesiog per HTTP parametrus. Kliento siunčiamas pranešimas, arba tiesiog HTTP užklausa, iškviečia tam tikrą REST servigo metodą, kuris gali turėti parametrus arba ne. Tokios užklauskos rezultatas yra HTTP atsakas – struktūrizuotas XML (*eXtensible Markup Language*) srautas. Komunikuojant klientui ir serveriui yra būtina identifikuoti, kuris vartotojas vykdo užklauską, todėl yra būtinas identifikacijos mechanizmas. Šiuo atveju jis realizuotas ad-hoc principu. Pirmą kartą jungiantis vartotojui yra vykdoma autentifikacija ir grąžinamas SHA-1 (<http://www.itl.nist.gov/fipspubs/fip180-1.htm>) algoritmu sugeneruotas kodas. Šis kodas naudojamas kaip sesijos identifikatorius prisijungusiam vartotojui identifikuoti ir yra pridodamas prie užklauskos parametrų kiekvieną kartą klientui kviečiant kitus REST servigo metodus.

Prisijungimo užklauskos pavyzdys panaudojant HTTP GET metodą:

- <http://myserver.com/RETSERVICE?method=login&username=myUsername&password=myPassword>

Tokios užklauskos galimas atsakas XML formatu atrodo taip:

```
<?xml version="1.0" encoding="utf-8" ?>
<root>
  <data>true</data>
  <sessionToken>465bf07ec6e9d588ecfdfff0e5e175b266f45a00</sessionToken>
</root>
```

Roy Fielding REST ideologijoje serveryje neturėtų būti saugoma jokia būsena įskaitant ir sesijos duomenis. Šiuo atveju autentifikacija yra būtina, todėl realizuotas REST servisas yra tik minėto autoriaus architektūros atmaina.

REST principų panaudojimas geografinių žemėlapių pateikimui internetu yra viena iš naujai besiformuojančių metodikų (http://www.directionsmag.com/article.php?article_id=2515&trv=1). Realizuoto sistemos prototipo atveju, klientas žemėlapius nuskaito iš žemėlapių saugyklos, prie kurios prieiga teikiama per REST servisą. Kiekvienas žemėlapių fragmentas išrenkamas pagal jį apibrėžiančias x, y koordinates ir z priartinimo lygį. Šis parametru trejetas paprasčiausiai nurodomas nuorodoje kreipiantis į servisą. Tokios nuorodos pavyzdys:

- <http://myserver.com/MAPService?x=1160&y=648&z=6>

Šios užklauskos rezultatas yra reikiamas žemėlapių fragmentas.

4. Išvados

- GSM pozicijos nustatymo metodai labai praverčia stebimiems objektams esant uždaroje patalpose, tačiau didelį tikslumą turinti GPS technologija yra nepakeičiama lauko sąlygomis.
- Taupant kuklius delninio kompiuterio resursus ir mažinant persiunčiamų duomenų kiekį, tikslinga naudoti plintančią Web servisų atmainą – REST servisas, o naujai atsiųstus žemėlapių fragmentus – kešuoti.
- Pozicijos koordinatės gaunamos iš GPS įrenginio priklauso sferinei koordinatinių sistemai, o žemėlapiai – stačiakampe, todėl, pozicijos taško atvaizdavimui ant žemėlapių atlikti, būtina pozicijos koordinatės transformuoti į stačiakampę koordinatinių sistemą panaudojant atvirkštinę Gudermano (Gudermannian) funkciją.
- Mercator projekcijos pagrindu sudaryto žemėlapių mastelis yra stipriai iškraipytas, ypač tolstant nuo pusiaujo link ašigalių. Tačiau ši projekcija puikiai tinkama interaktyvių internetinių žemėlapių sudarymui, kuriais patogiu manipuliuoti, todėl neretai yra taikoma geografinėse informacinėse sistemose.

Literatūros sąrašas

- [1] **D. Dao, C. Rizos, G. Wang.** Location-based service: Technical and business issues. *GPS solutions*, 2002, Vol. 6(3), p. 169-178.
- [2] **R.T. Fielding.** Architectural Styles and the Design of Network-Based Software Architectures. doctoral dissertation, Dept. of Computer Science, Univ. of Calif., Irvine, 2000.
- [3] **G. Goth.** Critics Say Web Services Need a REST. *IEEE Distributed systems online, IEEE Computing Society*, Dec. 2004, Vol. 5(12).
- [4] **V. Liutkauskas, I. Lagzdinytė.** GSM tinklo mobilios vietos patikslinimo būdai. *Elektronika ir elektrotechnika, KTU*, 2004, Nr. 5, p. 141-146.
- [5] **V. Palilionis.** GIS ir mobiliųjų technologijų integravimo lokalizuotųjų paslaugų sistemose. *Geodezija ir kartografija, VGTU*, 2004, XXX t., Nr. 4, p. 117-122.
- [6] **S. Vinoski.** Demystifying RESTful Data Coupling. *IEEE Internet Computing online, IEEE Computing Society*, Mar.-Apr. 2008, Vol. 12(2), p 87-90.
- [7] **S. Vinoski.** REST eye for the SOA Guy. *IEEE Internet Computing online, IEEE Computing Society*, Jan.-Feb. 2007, Vol. 11(1), p. 82-84.

Multi-user position tracking system

Different kinds of positioning systems are being used around the world. Very often they are specialized for a particular problem domain and named differently, but usually based on the same well known technologies. Most of these systems can be classified as GPS or GSM based systems, although some of them combine these two technologies to achieve optimal results. In most of these systems it is very important their operational range, position data accuracy and timing. Prevailing systems use mostly GPS technology. However both GPS and GSM based positioning systems have their advantages and drawbacks. Because mobile internet connection is improving quickly and becoming widely available, it opens wider possibilities of using such systems to ordinary consumers in every day life. In this article we briefly describe concept and main functional principles of implemented multi-user positioning system prototype based on GPS technology.