



LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

MOBILĀ LIETOJUMPROGRAMMATŪRA NAVIGĀCIJAI  
RĪGĀ

KVALIFIKĀCIJAS DARBS

Autors: **Jevgēnijs Cibankovs**

Stud. apl. **jc09076**

Darba vadītājs: **Ēriks Klotiņš**

RĪGA 2011

## ANOTĀCIJA

Mobilā lietojumprogrammatūra navigācijai Rīgā ir tīmekļa lietojumprogrammatūra, kas paredzēta viedtālrunu lietotājiem, kuri vēlas atrast kādu no sistēmas piedāvātiem objektiem savā tuvumā. Sistēma nodrošina lietotājam iespēju ātri apskatīt uz kartes savu atrašanās vietu un objektus savā tuvumā, ka arī uzzināt maršrutu sarakstu tuvākajās sabiedriskā transporta pieturās. Ar sistēmas palīdzību, lietotājs var dalīties savās domās par konkrēto vietu ar saviem sekotājiem tviterī.

Sistēma izmanto SIA “Rīgas satiksme”, OpenStreetMap un Google Maps sniegto informāciju, ka arī publicē lietotāja komentārus tviterī.

Atslēgas vārdi: Rīga, karte, Twitter, SIA “Rīgas satiksme”, OpenStreetMap, Google Maps, AJAX

## ABSTRACT

Mobile application for navigation in Riga is an application provided for smart phone users, who want to find certain system provided objects in their close range. This system provides it's user with capability to quickly locate his or her location on map and objects close to him/her. It also allows observing timetables on closest public transport stations. With this system user can share opinions about his/ her current location with followers on Twitter.

System uses information from OpenStreetMap and Google Maps provided from SIA "Rīgas Satiksme"( Ltd. Riga Public Transportation), and posts it's users tweets in Twitter.

Key words: Riga, map, Twitter, SIA "Rīgas satiksme", OpenStreetMap, Google Maps, AJAX

# SATURS

<b>IEVADS.....</b>	<b>1</b>
<b>PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA.....</b>	<b>2</b>
<b>1. IEVADS.....</b>	<b>2</b>
1.1. Nolūks.....	2
1.2. Darbības sfēra.....	2
1.3. Definīcijas un saīsinājumi.....	2
1.3.1. Jēdzieni.....	2
1.3.2. Pieņemtie datu tipi.....	4
1.4. Saistība ar citiem dokumentiem .....	5
1.5. Dokumenta pārskats.....	5
<b>2. VISPĀRĒJS APRAKSTS.....</b>	<b>6</b>
2.1. Produkta perspektīva.....	6
2.2. Biznesa prasības.....	7
2.3. Sistēmas lietotāju grupas .....	7
2.4. Lietotāja raksturiezīmes .....	7
2.5. Produkta funkcijas.....	7
2.6. Vispārēji ierobežojumi .....	9
2.7. Pieņēmumi un atkarības.....	10
<b>3. FUNKCIONĀLĀS PRASĪBAS.....</b>	<b>11</b>
3.1. Lietotāja funkcijas.....	11
3.1.1. Rādīt programmas ielādes logu .....	11
3.1.2. Rādīt sākulapu .....	12
3.1.3. Izvēlēties meklējamo objektu kategorijas .....	13
3.1.4. Parādīt tuvumā esošos objektus .....	14
3.1.5. Skatīt objekta informāciju .....	16
3.1.6. Izvēlēties dienas tipu .....	17
3.1.7. Skatīt pieturas transportu sarakstu .....	18
3.1.8. Skatīt transportu .....	19
3.1.9. Skatīt pienākšanas laikus nākamajās pieturās .....	20
3.1.10. Skatīt maršrutu kartē.....	21
3.1.11. Ieteikt trūkstošo objektu.....	22
3.1.12. Mainīt valodu .....	23
3.1.13. Skatīties GeoTviteri .....	24
3.1.14. Autorizēties ar tviteri.....	25

3.1.15.	<i>Atstāt komentāru.....</i>	26
3.1.16.	<i>Lasīt komentārus .....</i>	27
3.1.17.	<i>Atteikties .....</i>	28
3.2.	<i>Funkcijas sistēmai .....</i>	29
3.2.1.	<i>Atjaunot objektu informāciju.....</i>	29
3.2.2.	<i>Atjaunot Rīgas sabiedriska transporta informāciju .....</i>	30
<b>4.</b>	<b>ĀRĒJĀ SASKARNE.....</b>	<b>31</b>
4.1.	<i>Lietotāja saskarne .....</i>	31
4.2.	<i>Aparatūras saskarne .....</i>	31
4.3.	<i>Programmatūras saskarne .....</i>	31
<b>5.</b>	<b>NEFUNKCIONĀLĀS PRASĪBAS .....</b>	<b>32</b>
5.1.	<i>Veiktspējas prasības .....</i>	32
5.2.	<i>Pieejamība.....</i>	32
5.3.	<i>Lietojamība.....</i>	32
5.4.	<i>Datu drošība.....</i>	33
5.5.	<i>Uzticamība .....</i>	33
	<b>PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS .....</b>	<b>34</b>
1.	<b>IEVADS .....</b>	<b>34</b>
1.1.	<i>Nolūks.....</i>	34
1.2.	<i>Darbības sfēra.....</i>	34
1.3.	<i>Definīcijas .....</i>	34
1.3.1.	<i>Izmatotie saīsinājumi.....</i>	34
1.3.2.	<i>Izmantotie jēdzieni.....</i>	35
1.4.	<i>Saistība ar citiem dokumentiem .....</i>	35
2.	<b>DEKOMPOZĪCIJAS APRAKSTS .....</b>	<b>36</b>
2.1.	<i>Moduļu dekompozīcija .....</i>	36
2.1.1.	<i>Galvenais kontrolieris .....</i>	38
2.1.2.	<i>AJAX pieprasījums .....</i>	38
2.1.3.	<i>Procesa apstrādes bilde .....</i>	38
2.1.4.	<i>Ekranformas attīrīšana.....</i>	38
2.1.5.	<i>Atrašanas vietas noteikšana .....</i>	38
2.1.6.	<i>DOM operācijas .....</i>	39
2.1.7.	<i>Kļūdas un paziņojumi.....</i>	39
2.1.8.	<i>Valodas apstrāde.....</i>	39
2.1.9.	<i>Marķieru pārvaldīšanas modulis.....</i>	39
2.1.10.	<i>Iezīmju operācijas .....</i>	40
2.1.11.	<i>Taimeris.....</i>	40

2.1.12.	<i>Programmas uzsākšanas sadaļa</i>	40
2.1.13.	<i>Sākulmapas sadaļa</i>	40
2.1.14.	<i>Objektu ieteikšanas sadaļa</i>	41
2.1.15.	<i>Kategoriju sadaļa</i>	41
2.1.16.	<i>„Objekti apkārt man” sadaļa</i>	41
2.1.17.	<i>Dienas izvēles sadaļa</i>	41
2.1.18.	<i>Pieturas saraksta sadaļa</i>	42
2.1.19.	<i>Transporta atiešanas laiku sadaļa</i>	42
2.1.20.	<i>Maršruta laiku pieturās sadaļa</i>	42
2.1.21.	<i>Maršruta uz kartes sadaļa</i>	42
2.1.22.	<i>Valodas maiņas sadaļa</i>	42
2.1.23.	<i>ĢeoTvitera sadaļa</i>	43
2.1.24.	<i>Komentāru atstāšanas sadaļa</i>	43
2.1.25.	<i>Komentāru attēlošanas sadaļa</i>	43
2.1.26.	<i>Ieteikumu kontrolieris</i>	43
2.1.27.	<i>Objektu kontrolieris</i>	44
2.1.28.	<i>Rīgas sabiedriska transporta kontrolieris</i>	44
2.1.29.	<i>Lietotāju kontrolieris</i>	44
2.1.30.	<i>Objektu modelis</i>	44
2.1.31.	<i>Rīgas sabiedriska transporta modelis</i>	45
2.1.32.	<i>Ieteikumu modelis</i>	45
2.1.33.	<i>Tvitera modelis</i>	45
2.1.34.	<i>Datu modelis</i>	45
2.1.35.	<i>Rīgas Sabiedriska transporta informācijas ielāde</i>	46
2.1.36.	<i>RST laiku pieturās lejuplādēšana</i>	46
2.1.37.	<i>RST laiku pieturās ielāde</i>	46
2.1.38.	<i>OpenStreetMap objektu informācijas ielāde</i>	46
2.1.39.	<i>Notikumi</i>	47
2.2.	<i>Datu dekompozīcija</i>	47
3.	<b>ATKARĪBAS APRAKSTS</b>	48
3.1.	<i>Starpmoduļu atkarības</i>	48
3.2.	<i>Datu atkarības</i>	52
3.2.1.	<i>Konceptuālais ER modelis</i>	52
3.2.2.	<i>Realizācijas ER modelis</i>	53
4.	<b>DETALIZĒTS PROJEKTĒJUMS</b>	54
4.1.	<i>Datu detalizēts projektējums</i>	54
4.1.1.	<i>Tabula „Objects”(Obj)</i>	54
4.1.2.	<i>Tabula „ObjectCategories”(ObjCat)</i>	54

4.1.3.	Tabula „Stops”(Stops) .....	55
4.1.4.	Tabula „Trip”(Trip).....	55
4.1.5.	Tabula „TransportType”(TrtID).....	56
4.1.6.	Tabula „TripOnStops”(TOS) .....	56
4.1.7.	Tabula „DepartureTime”(DepT) .....	56
4.1.8.	Tabula „Advices”(Advic).....	57
4.1.9.	Tabula „Timing”(Tim) .....	57
4.1.10.	Tabula „Users”(Users).....	57
4.1.11.	Tabula „Tweets”(Tweets) .....	58
4.2.	Moduļu detalizēts projektējums.....	59
4.2.1.	Galvenā kontroliera moduļa algoritma projektējums .....	59
4.2.2.	Atrašanas vieta noteikšanas moduļa algoritma projektējums.....	60
4.2.3.	Marķieru pārvaldības moduļa algoritma projektējums .....	61
4.2.3.1.	Marķieru grupēšanas funkcijas algoritma projektējums.....	61
4.2.4.	„Objekti apkārt man” moduļa algoritma projektējums .....	62
4.2.5.	Objektu modeļa moduļa algoritma projektējums .....	63
4.2.5.1.	Metodes „Cluster” algoritma projektējums .....	63
4.2.6.	OpenStreetMap objektu informācijas ielādes moduļa algoritma projektējums .....	64
4.2.7.	Rīgas Sabiedriska transporta informācijas ielādes moduļa algoritma projektējums .....	65
4.2.8.	RST laiku pieturās ielādes moduļa algoritma projektējums .....	67
<b>TESTĒŠANAS DOKUMENTĀCIJA .....</b>		<b>68</b>
<b>1. IEVADS.....</b>		<b>68</b>
<b>2. VIENIBTESTĒŠANAS REZULTĀTI .....</b>		<b>68</b>
2.1.	Lietotāja pieslēgšanās sistēmai.....	69
2.2.	Sākumlapa .....	69
2.3.	Ieteikumu sadaļa .....	69
Sadaļas forma atbilst pielikumā aprakstītaj Ieteikumu.....		69
2.4.	Kategoriju izvēles sadaļa .....	73
2.5.	„Objekti apkārt man” sadaļa.....	74
2.6.	Dienas izvēles sadaļa .....	78
2.7.	Pieturas saraksta sadaļa .....	78
2.8.	Maršruta pienākšanas laiku sadaļa .....	79
2.9.	Maršruta laiku pieturās sadaļa .....	80
2.10.	Maršruta uz kartes sadaļa.....	81
2.11.	GeoTvitera sadaļa .....	82
2.12.	Komentāru atstāšanas sadaļa .....	83
2.13.	Komentāru attēlošanas sadaļa .....	85

2.14. Valodas maiņa un valodas maiņas sadaļa .....	87
<b>PROJEKTA ORGANIZĀCIJA.....</b>	<b>88</b>
<b>KVALITĀTE NODROŠINĀŠANA .....</b>	<b>89</b>
<b>KONFIGURĀCIJU PARVALDĪBA.....</b>	<b>90</b>
<b>DARBIETILPĪBAS NOVĒRTĒJUMS.....</b>	<b>91</b>
<b>IZMANTOTĀS TEHNOLOĢIJAS .....</b>	<b>93</b>
<b>PROGRAMMATŪRAS PIRMKODS .....</b>	<b>94</b>
1. Galvenā kontroliera saturs .....	94
2. Rīgas sabiedriska transporta kontroliera saturs .....	98
3. Rīgas sabiedriska transporta modeļa saturs .....	100
4. Rīgas sabiedriska transporta informācijas ielādes moduļa saturs.....	103
5. Komentāru attēlošanas sadaļas moduļa saturs .....	113
6. Atrašanas vietas noteikšanas moduļa saturs .....	118
<b>SECINĀJUMI .....</b>	<b>119</b>
<b>IZMANTOTĀ LITERATŪRA.....</b>	<b>120</b>
<b>PIELIKUMI .....</b>	<b>121</b>
1. EKTRANFORMU PIEMĒRI .....	121
1.1. Sākumlapas forma .....	121
1.2. Ieteikumu sadaļas forma .....	122
1.3. Kategoriju izvēles sadaļas forma .....	123
1.4. „Objekti apkārt man” sadaļas forma .....	124
1.5. Dienas izvēles sadaļas forma .....	125
1.6. Pieturas saraksta sadaļas forma .....	126
1.7. Maršruta pienākšanas laiku sadaļas forma .....	127
1.8. Maršruta laiku pieturās sadaļas forma.....	128
1.9. Maršruta uz kartes sadaļas forma.....	129
1.10. GeoTvitera sadaļas forma.....	130
1.11. Komentāru atstāšanas sadaļas forma .....	131
1.12. Komentāru attēlošanas sadaļas forma .....	132
1.13. Valodas maiņas sadaļas forma .....	133
2. PAZIŅOJUMI .....	134



## IEVADS

Ar katru dienu viedai tālrunu lietotāju skaits palielinās, un šī tendence labi parāda mūsdienu pasaules cilvēka vēlmes – ātri, viegli, kompakti. Cilvēki grib iegūt ātri viņiem nepieciešamo informāciju, atrodoties jebkurā pasaules malā. Reducējot visu pasauli uz Rīgu, savā kvalifikācijas darbā mēģināju izstrādāt sistēmu, kas sniegs lietotāja viņam nepieciešamo informāciju, izdarot tikai divus „klikšķus”.

Darba mērķis bija izstrādāt "Mobilā lietojumprogrammatūra navigācijai Rīgā", kas viņas lietotājiem sniegtu iespēju ātri un viegli iegūt viņiem nepieciešamo informāciju par objektiem un sabiedrisko transportu Rīgā, ka arī, atstāt komentāru sava atrašana vietā un šie komentāri būtu redzami viņu draugiem tviterī.

Izstrādāta sistēma ir tīmekļa lietojumprogrammatūra, kurā ir domāta viedai tālruniem. Sistēma sniedz lietotājiem iespējas izvēlēties objektus, kurus viņi vēlas meklēt savā tuvumā un ieraudzīt tos uz kartes, ka arī redzēt savu atrašanas vietu. Lietotājs var ātri un viegli atrast tādus objektus, ka bankomātus, kafejnīcas, stāvvietas, ka arī Rīgas sabiedriska transporta pieturās. Pateicoties tam, lietotājs var ātri uzzināt sabiedriskā transporta sarakstu pieturās, transportu pienākšanas laikus un apskatīt maršrutu uz kartes, lai pārliecināties, ka šis transports nogadās viņu vēlamajā vietā. Sistēma tika izstrādāta tā, lai tās saskarne būtu pēc iespējas vienkāršāka un viegli izmantojama.

Kvalifikācijas darba ietvaros tika izstrādāta programmatūras prasību specififikācijas, projektējuma apraksta, testēšanas dokumentācijas, projekta organizācijas, kvalitātes nodrošināšanas, konfigurāciju pārvaldības, dots darbietilpības novērtējuma, aprakstītas izmantotas tehnoloģijas un veikti secinājumi par kvalifikācijas darbā paveikto. Pievienots arī programmatūras pirmkods un pielikumi.

# PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

## 1. Ievads

### 1.1. Nolūks

Dokumenta nolūks ir apkopot un aprakstīt sistēmas "Mobilā lietojumprogrammatūra navigācijai Rīgā" prasības – gan funkcionālās, gan nefunkcionālās.

Balstoties uz šo dokumentu turpmāk tiks izstrādāta cita nepieciešamā dokumentācija, kā arī pati sistēma.

### 1.2. Darbības sfēra

Šī sistēma ir paredzēta lietotājiem, kuriem ir vajadzīga palīdzība navigācijai Rīgā. Tā palīdzēs atrast dažāda tipu objektus, ka arī parādīs sabiedriska transporta pieturu atrašanās vietas un maršrutu sarakstus tajās.

### 1.3. Definīcijas un saīsinājumi

#### 1.3.1. Jēdzieni

Jēdziens	Skaidrojums
Viedais tālrunis	Mobilā telefona aparāts, kas nodrošina ne tikai balss pakalpojumu, bet arī e-pasta un teksta ziņojumu sūtīšanu, piekļuvi tīmeklim un balss atpazīšanu. Angliskais termins - smart phone.[7.]
Pieskarties	Pirksta nolaišana uz ekrāna.
OpenStreetMap	Karte, kuru var izmantot jebkurš (cilvēks, organizācija, uzņēmums, valsts iestāde) jebkuriem mērķiem un kuru var papildināt jebkurš lietotājs ar sevis savāktiem vai trešās puses datiem, ja tos ir atļauja publicēt par brīvu, izmantojot atsauci.[11.]
Pinch-to-zoom	Žests, kas ļauj lietotājam veikt tālummaiņu, virzot divus pirkstus tālāk vienu no otra, vai tuvāk, bet pieskaroties displeja.

<b>Jēdziens</b>	<b>Skaidrojums</b>
Skārienekrāns	Displeja ekrāns, kas pārklāts ar caurspīdīgu skārienjūtīgu pārklājumu. Skārienekrāns pārveido lietotāja pirksta pieskārienu kādā noteiktā ekrāna vietā informācijā, kas tiek nodota programmatūrai. Tas dod iespēju skārienekrānu izmantot peles, gaismas zīmuļa vai tml. rādītājierīču aizstāšanai.[7.]
Lietotājs	Persona, kas izmanto lietojumprogrammu savu uzdevumu risināšanai.
Pārlūkprogramma	Lietojumprogramma, kas paredzēta datu bāzu, datņu sarakstu, kā arī globālā tīmekļa dokumentu izskatīšanai, lai atrastu lietotājam vajadzīgo informāciju.[7.]
HTML	Valoda, kas, izmantojot speciālus kodus, nosaka hiperteksta dokumenta atveidojumu displeja ekrānā gadījumā, ja tiek lietotas tīkla Internet globālā tīmekļa lappuses.[7.]
CSS	Īpaša stila lapas valoda, ko lieto, lai aprakstītu izskatu iezīmēšanas valodā veidotiem dokumentiem. [13.]
Datubāze	Savstarpēji saistītu informacionālu objektu tematisks kopums, kas ar speciālas pārvaldības sistēmas starpniecību organizēts tā, lai nodrošinātu ērtu informācijas izguvi, izdarītu tās atlasīšanu un kārtošanu. Informācija datu bāzē parasti ir sadalīta ierakstos (tabulās), no kuriem katram var būt viens vai vairāki lauki (kolonnas).[7.]
JavaScript	Firmas Netscape izveidota valoda, kas ļauj globālā tīmekļa izstrādātājiem veidot interaktīvas vietnes. JavaScript var sadarboties ar valodas HTML pirmkoda programmām, tādējādi ļaujot globālā tīmekļa izstrādātājiem papildīt šīs vietnes ar dinamisku saturu.[7.]
AGPS	Tehnoloģija, kas paātrina "aukstās palaišanas" GPS-uztvērēju. Paātrinājums nodrošina, piemēram, Wi-Fi vai mobilo sakaru torņi. Bieži lieto mobilos telefonos, kas satur GPS uztvērējus. [8.]
Pietuvinājums	Šī dokumenta konteksta, pietuvinājums nozīmē to – kartes

<b>Jēdziens</b>	<b>Skaidrojums</b>
	detalizācijas pakāpi, sākot no visas pilsētas, līdz ielai.
Ģeogrāfiskais platums	Leņķis starp ekvatoru un kādu no poliem uz globusa. Papildus precizitātei tiek izmantotās minūtes un sekundes [9.]
Ģeogrāfiskais garums	Leņķis starp Griničas meridiānu un to meridiānu, kas iet caur kādu konkrētu punktu. Garums var būt no 0° (Griničas meridiāns) līdz 180° (datumu maiņas līnija). Papildus precizitātei tiek izmantotās minūtes un sekundes. [10.]
UTF-8	Mainīga platuma kodējums. Tas var attēlot jebkuru simbolu unikoda standartā, un vienlaikus ir savietojams ar ASCII. Tas ir guvis plašu izplatību e-pastā, globālajā tīmeklī un citos teksta glabāšanas vai pārsūtīšanas lietojumos. [12.]
Tviteris	Virtuālas komunikācijas un mikroblogošanas tīkls, kurā cilvēki var sūtīt un lasīt savas un citu rakstītas īsas ziņas, kuras sauc par tvītiem (angļu: tweets).[14.]

### 1.3.2. Pieņemtie datu tipi

<b>Datu tips</b>	<b>Skaidrojums</b>
Izvēle no saraksta	Viens no saraksta elementiem. Tiek izmantots, kad lietotājam vajag sniegt izvēli no kaut kā. Datus iegūst, kad lietotājs pieskārās kādam no saraksta elementiem.
Koordināta	Ģeogrāfisko koordināšu vērtība decimālajā pierakstā. Platuma koordināta ir robežā no -90 līdz 90, un garuma koordināta – no -180 līdz 180.  Piemērs punkta koordinātam – (56.951433, 24.113241), kur pirmā koordināte ir platums, bet otra – garums.
Teksts	UTF-8 simbolu virkne.

## **1.4. Saistība ar citiem dokumentiem**

Šis dokuments tika izstrādāts pēc LVS 68:1996 "Programmatūras Prasību Specifikācijas Ceļvedis" un ir lietojams kopā ar sekojošiem dokumentiem - „Programmatūras projektējuma apraksts”, „Testēšanas dokumentācija”, „Konfigurāciju pārvaldība”, „Kvalitātes nodrošināšana”, „Projekta organizācija”, „Darbietilpības novērtējums”.

## **1.5. Dokumenta pārskats**

Dokuments sastāv no 5 nodaļām: ievads, vispārējs apraksts, funkcionālās prasības, ārējā saskarne, nefunkcionālās prasības.

Ievadā aprakstīti dokumenta mērķi, nolūks, izmantotie jēdzieni, kas sastopami dokumentā.

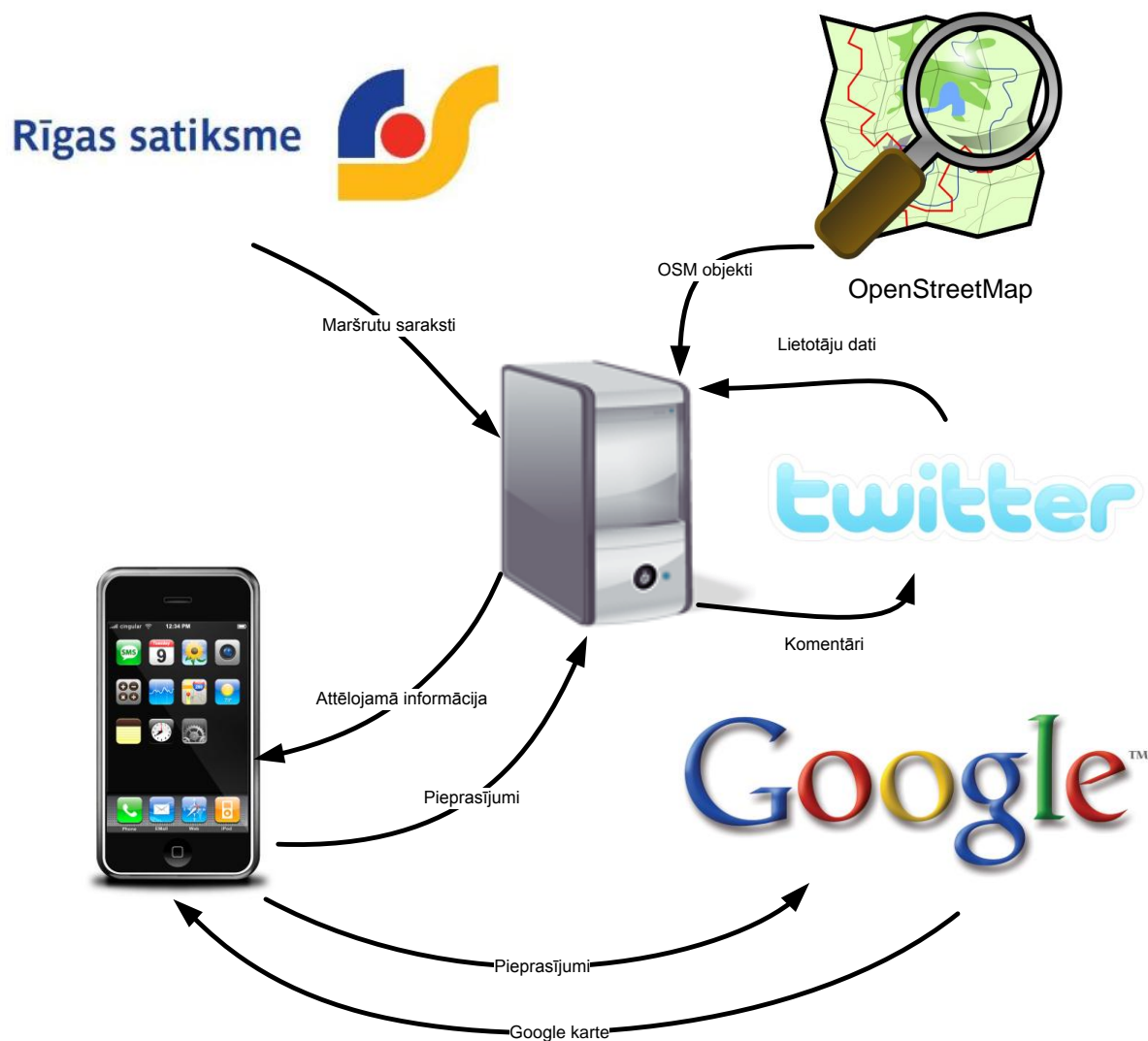
Vispārējais apraksts tiek sniegts pārskats par funkcijām, kas sistēmai ir jāpilda, kā arī aprakstītas lietotāja raksturiezīmes. Šajā nodaļā ir aprakstīti ierobežojumi un atkarības, kas var ietekmēt sistēmas izstrādi.

Funkcionālo prasību nodaļā tiek aprakstītas sistēmā iekļautās funkcijas, to ieejas dati, šo datu apstrādes darbības, izejas dati, funkciju atkarības.

## 2. Vispārējs apraksts

### 2.1. Produkta perspektīva

Sistēma izmanto OpenStreetMap un Rīgas satiksmes piedāvātos resursus un informāciju, ka arī publicē lietotāja komentārus twitter.com.



2.1. att. Sistēmas konceptuālais modelis

## 2.2. Biznesa prasības

Prasība	Identifikators
Jāparada cilvēkam objektus viņa apkārtnē, tādus ka restorāni, kafējnīcas, hoteļus, bankas, pieturas.	B.1
Jāsniedz lietotājam informāciju par tuvākām pieturām un maršrutu sarakstiem tajās.	B.2
Jāsniedz lietotājam iespēju atstāt komentārus vietā, kur viņš atrodas, un lasīt citu lietotāju komentārus.	B.3

## 2.3. Sistēmas lietotāju grupas

Sistēmas lietošanā ir paredzēts viena sistēmas lietotāju grupa – lietotājs. Lietotājs ir cilvēks, kurš izmanto mobilo aplikāciju un var darboties ar tādām funkcijām, ka objektu meklēšana apkārt sev.

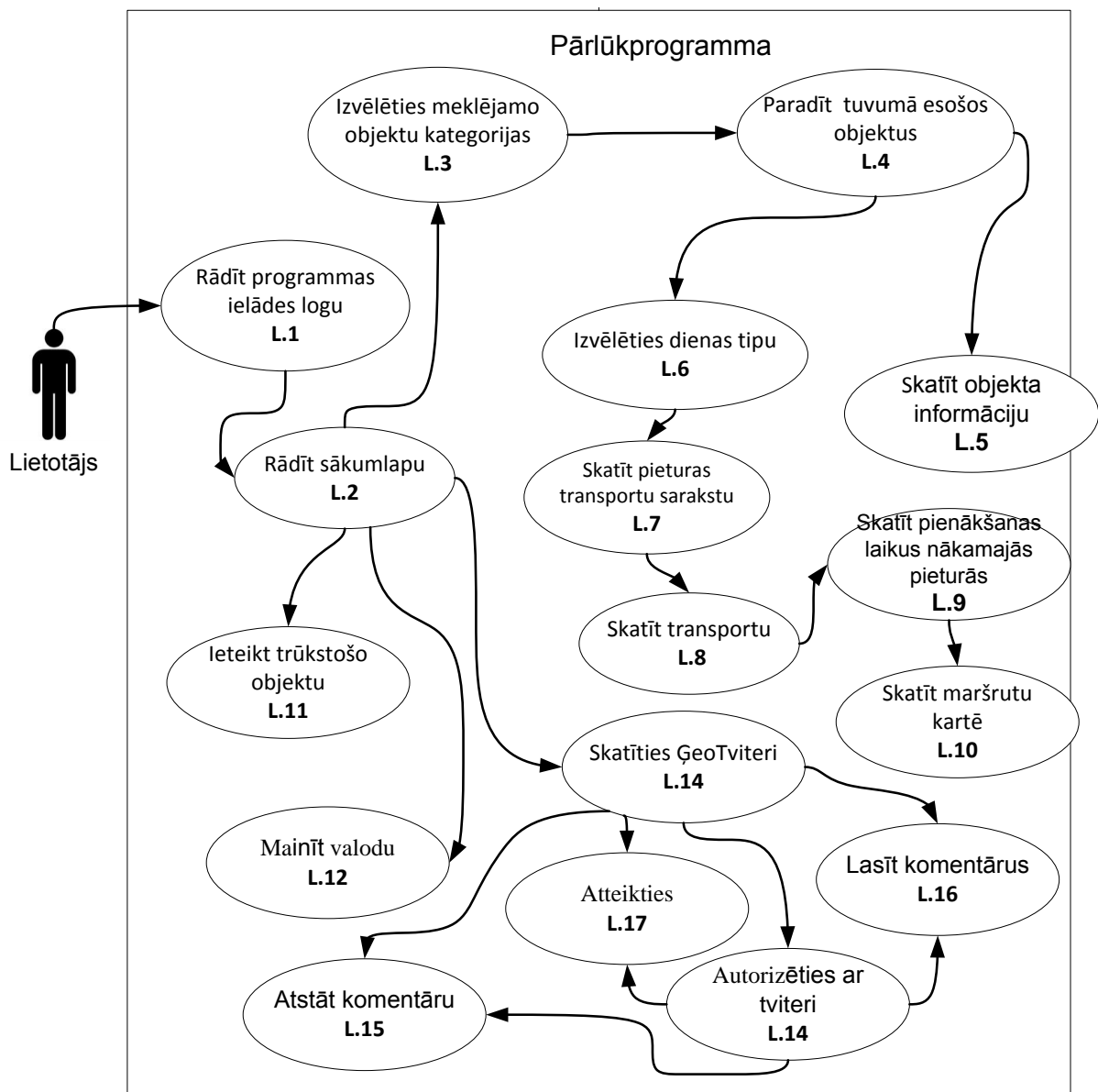
Papildus lietotājam, uz servera ir dažas funkcijas, kuras darbojas patstāvīgi, bet tas iedarbinās cilvēks, kas apkalpo serveri.

## 2.4. Lietotāja raksturiezīmes

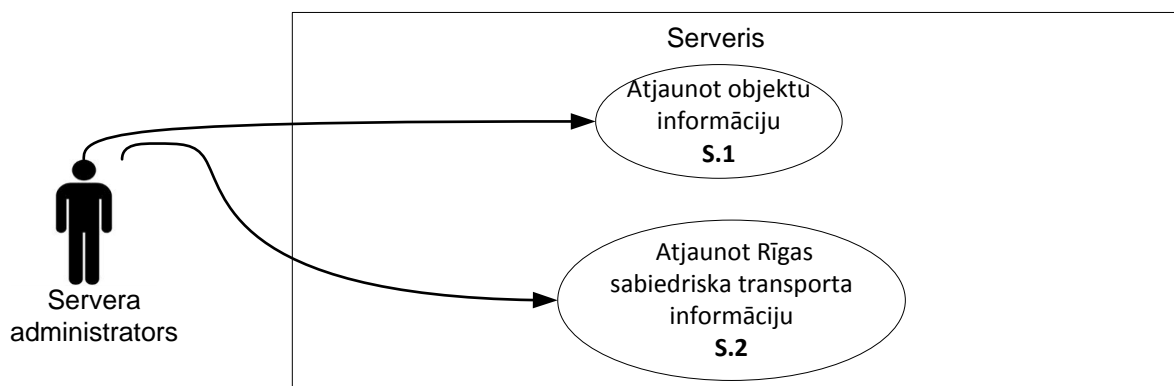
Lietotājs, ir cilvēks, kurš izmanto mobilas lietojumprogrammatūras, un kurām ir pieredze darbā ar tām.

## 2.5. Produkta funkcijas

Funkcijas tiek grupētas divās grupās - lietotāja funkcijas un servera funkcijas.



2.2. att. Lietošanas diagramma „Lietotājs”



2.3. att. Lietošanas diagramma „Sistēma”



Katrai funkcijai ir savs identifikators. Identifikatora sastāv no divām vērtībām A.B.

- A – pirmais burts atkarība no funkcijas izpildītāja:
  - L – lietotājs;
  - S – serveris.
- B – kārtas numurs funkcijai.

Funkcijas lietotājam:

- Rādīt programmas ielādes logu – L.1;
- Rādīt sākumlapu – L.2
- Izvēlēties meklējamo objektu kategorijas – L.3;
- Parādīt tuvumā esošos objektus – L.4;
- Skatīt objekta informāciju – L.5;
- Izvēlēties dienas tipu – L.6;
- Skatīt pieturas transportu sarakstu – L.7;
- Skatīt transportu – L.8;
- Skatīt pienākšanas laikus nākamajās pieturās – L.9;
- Skatīt maršrutu kartē – L.10;
- Ieteikt trūkstošo objektu – L.11;
- Mainīt valodu – L.12;
- Skatīties GeoTviteri – L.13;
- Autorizēties ar tviteri – L.14;
- Atstāt komentāru – L.15;
- Lasīt komentārus – L.16;
- Atteikties – L.17.

Funkcijas sistēmai:

- Atjaunot objektu informāciju - S.1;
- Atjaunot Rīgas sabiedriska transporta informāciju - S.2.

## **2.6. Vispārēji ierobežojumi**

Sistēmas lietošanai nepieciešams specifiska iekārta – viedais tālrunis(smart phone) ar W3 standartiem atbilstošo pārlūku, kurš atbalsta HTML 5, CSS 3 un JavaScript.

## **2.7. Pieņēmumi un atkarības**

Šī sistēma ir pieejama lietotājiem tikai interneta vidē, tāpēc sistēma vienmēr būs atkarīga no interneta pieejamības un ātrdarbības.

Sistēmas datu korektums un pilnība ir atkarīgs no OpenStreetMap un Rīgas satiksmes sniegtajiem datiem.

Sistēma arī ir atkarīga no Google maps kartes nodošanas lietotājam, ātruma.

Sistēmas komentāru lasīšanas un atstāšanas iespējas ir atkarīgas no twitter.com darbības.

Sistēmas darbība un tas darbības rezultātu precizitāte būs atkarīga no mobilas ierīces AGPS precizitātes.

### 3. Funkcionālās prasības

Šajā nodaļā tiek detalizēti aprakstītas sistēmas funkcionālās prasības. Funkcijas izmantotie kļūdu identifikatori ir aprakstīti pielikumā „Paziņojumi”. Parasto paziņojumu identifikatori sākas ar MSG, bet kļūdu – ERR.

Kartes pietuvinājumā līmeņi ir 3. 1. – mazs, 2. – vidējais, 3. – liels. Mazā pietuvinājuma lietotājs var redzēt visu pilsētu kopumā. Vidējā pietuvinājumā lietotājs var redzēt pilsētas kvartālus. Lielajā pietuvinājumā, lietotājs var apskatīt ielu, un tas ir maksimālai Google maps piedāvātais pietuvinājums.

Ja lietotājs pēc pieprasījuma nosūtīšanas uz serveri, neiegūst atbildi 14 sekunžu laikā, viņam tiek parādīts paziņojums ERR\_TIME14, un viņš tiek atgriezts uz iepriekšējo darbību.

#### 3.1. Lietotāja funkcijas

##### 3.1.1. Rādīt programmas ielādes logu

ID	L.1
Mērķis	Šī funkcija automātiski tiek iedarbināta, kad lietojumprogramma tiek palaista. Tā attēlo OpenStreetMaps, Google, Rīgas satiksmes un Tvitera logotipus un pašas lietojumprogrammas logotipu.
Pieejas tiesības	Visi
Atkarība	-
Dod pieeju pie	L.2: Rādīt sākumlapu.
Ievaddati	
-	
Apstrāde	
-	
Izvaddati	

Ar slīdrādi tiek parādīt bilde ar OpenStreetMap, Rīgas satiksmes, Google, Tvitera logotipu 2 sekundžu laikā, un tad parādās lietojumprogrammas logotops uz 2 sekundēm. Pēc tam tiek rādīta ielādes bilde, līdz lietojumprogramma tiek ielādēta.

Kļūdu apstrāde:

Ja lietotājam rodas problēmas ar pieslēgšanos datubāzei, tad parāda paziņojumu ERR\_DB un pāradresē uz sākumlapu (pārējas funkcijās līdzīgas kļūdas rašanas gadījuma, darbība ir identiska).

### 3.1.2. Rādīt sākumlapu

ID	L.2
Mērķis	Šī funkcija nodrošina lietotājam izvēlēties kādas darbības tas vēlas veikt sistēmā.
Pieejas tiesības	Visi
Atkarība	L.1: Rādīt programmas ielādes logu
Dod pieeju pie	L.3: Izvēlēties meklējamo objektu kategorijas, L.11: Ieteikt trūkstošo objektu, L.12:Mainīt valodu, L.13: Skatīties ĢeoTviteri.

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Sadaļas nosaukums	Izvēle no saraksta	-	Jā	

#### Apstrāde

Ja lietotājs izvēlas sadaļu L.3 funkcijas sadaļu, sistēma atlasa objektu kategorijas, kuras piedāvās lietotājam izvēlēties.

#### Izvaddati

Parāda lietotājam izvēlēto sadaļu( L.3, L.11, L.12 vai L.13).

#### Kļūdu apstrāde:

Ja lietotājam izvēloties funkcijas L.3 sadaļu netiek piedāvāta neviena no izvēles kategorijām, parāda paziņojums ERR\_NCF un atgriež uz funkciju L.2.

### 3.1.3. Izvēlēties meklējamo objektu kategorijas

ID	L.3
Mērķis	Šī funkcija nodrošina lietotājam iespēju izvēlēties objektus, kurus viņš gribēs redzēt uz kartes.
Pieejas tiesības	Visi
Atkarība	L.2: Rādīt sākumlapu.
Dod pieeju pie	L.4: Parādīt tuvumā esošos objektus.

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Kategorijas nosaukums	Izvēle no saraksta	-	Jā	

#### Apstrāde

Izvēlēta kategorija tiek padota L.4 funkcijai.

#### Izvaddati

Tiek atvērta L.4 funkcijas sadaļa.

### 3.1.4. Parādīt tuvumā esošos objektus

ID	L.4
Mērķis	Šī funkcija nodrošina lietotājam iespēju apskatīt kartē savu atrašanas vietu un objektus apkārt sev, izvēloties dažādus attālumus(200m, 500m, 1km, 2km, 3km), vai arī apskatīt visus objektus, kas ir pieejami noteiktajā kategorijā. Šī funkcija attēlo tos objektus, kuru kategorija tika izvēlēta funkcijā L.3.
Pieejas tiesības	Visi
Atkarība	L.3: Izvēlēties meklējamo objektu kategorijas.
Dod pieeju pie	L.5: Skatīt objekta informāciju, L.6: Skatīt pieturas transportu sarakstu.

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Meklēšanas apgabals	Izvēle no saraksta	-	Jā	Meklēšanas apgabali: 200m, 500m, 1km, 2km, 3km vai visi objekti.  Kad karte tiek ielādēta pirmo reizi, noklusētais meklēšanas apgabals ir 500m.

#### Apstrāde

Tiek noteikta lietotāja atrašanas vieta. Sistēmā sameklē objektus, kuri ir noteiktajā meklēšanas apgabalā apkārt lietotājam. Tiek noteikta objektu atrašanas vieta uz kartes un tie tiek izvietoti uz tās. Katram objektam tiek pievienots apraksts par to, kuru var apskatīt pieskaroties objekta ikonai, vai pieturām tiek pievienotas funkcijas, kuras attēlos pieturu informāciju. Tiek noteikta lietotāja atrašanas vieta un tuvākais objekts.

#### Izvaddati

Parāda karti ar objektiem un lietotāja atrašanas pozīciju kartē. Lietotāja atrašanas marķieris atrodas ekrāna centrā. Noklusētais pietuvinājums ir vidējais. Lietotājam mainot pietuvinājumus starp vidējo un mazo, objektu ikonas(marķieri) savienojās grupās, lai izvairīties no pārklāšanas. Virs marķieriem atrodas lauki, kuros ir ierakstīts objekta nosaukums. Lietotāja marķieri un tuvāka objekta marķieri savieno zilā līnija, kurai pieskaroties, par kartes centru kļūst tuvākais objekts. Līnija nav redzama, kad marķieri apvienojas grupās.

Kļūdu apstrāde:

Ja neviens objekts netika atrasts, parāda paziņojumu ERR\_NOF. Ja nav objektu, kas ir meklēšanas apgabalā- parāda paziņojumu MSG\_NOINR.

Visos neveiksmīgos atrašanas vietas noteikšanas gadījumos, jāparada paziņojums ERR\_GUNDEF un par lietotāja noklusēto atrašanas vietu jāpieņem pozīcija ar ģeogrāfiskam koordinātam (platums:'56.946238', garums:'24.104905').

Ja lietotais atrodas ārpus Rīgas, viņu pabrīdina ar paziņojumu MSG\_NIRIGA. Lietotājs kurš atrodas Rīga un mēģina pārvietoties ārpus vietas par kuru sistēmai ir informācija(nosacīta Rīgas robeža), lietotājs tiek pabrīdināts ar paziņojumu MSG\_OUTRIGA un viņa pārvietošanās pa karti tiek apturēta.

Ja radās problēmas ielādējot karti, parāda paziņojumu - ERR\_IMAP un atgriež uz kategoriju izvēles sadaļu

---

### 3.1.5. Skatīt objekta informāciju

ID	L.5
Mērķis	Šī funkcija nodrošina lietotājam izvēlēties kādas darbības tas vēlēš veikt sistēmā.
Pieejas tiesības	Visi
Atkarība	L.4: Parādīt tuvumā esošos objektus.
Dod pieeju pie	-

#### Ievaddati

-

#### Apstrāde

-

#### Izvaddati

Virš objekta parādās logs, kurā tiek attēlota informācija par objektu. Loga augšējā labajā stūrī ir poga ar kuras palīdzību var aizvērt logu. Logu var aizvērt pieskaroties brīvā vietā uz kartes. Ja informācija par objektu nav pieejama, logs netiek atvērts.



### 3.1.6. Izvēlēties dienas tipu

ID	L.6
Mērķis	Šī funkcija nodrošina lietotājam iespēju izvēlēties kādas dienās meklēt transporta līdzekļa pienākšanas laikus pieturā.
Pieejas tiesības	Visi
Atkarība	L.5: Parādīt tuvumā esošos objektus.
Dod pieeju pie	L.7: Skatīt pieturas transportu sarakstu.

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Dienas tips	Izvēle no saraksta	-	Jā	Ir trīs dienu tipi: darbdiena, brīvdiena, tagad. 'Tagad' ir visi tie laiki, kas ir pēc tekoša laika līdz dienas beigām

#### Apstrāde

Dienas tips tiek padots funkcijai L.7.

#### Izvaddati

Tiek atvērta L.7 funkcijas sadaļa.

### 3.1.7. Skatīt pieturas transportu sarakstu

ID	L.7
Mērķis	Šī funkcija nodrošina lietotājam iespēju apskatīt sabiedriska transporta pieturas transportu sarakstu.
Pieejas tiesības	Visi
Atkarība	L.6: Izvēlēties dienas tipu.
Dod pieeju pie	L.8: Skatīt transportu.

#### Ievaddati

Saņemtais dienas tips no funkcijas L.6.

#### Apstrāde

Sistēmā tiek atlasīta informācija par maršrutiem šajā pieturā tāda ka - maršruta nosaukums, maršruta numurs, transporta tips, maršruta numurs.

#### Izvaddati

Lietotājam tiek parādīti saraksti ar transportiem, kas pietur dotajā pieturā. Maršruti tiek grupēti trijās kategorijās – autobusi, trolejbusi, tramvaji. Katrai kategorijai savs saraksts. Ja kāda transporta veida nav, tad tā veida sarakstu nerāda.

#### Kļūdu apstrāde:

Ja neviens transports netika atrasts, parāda paziņojumu ERR\_NTF un atgriež uz dienu izvēles sadaļu(L.6).

### 3.1.8. Skatīt transportu

ID	L.8
Mērķis	Šī funkcija nodrošina lietotājam iespēju apskatīt noteikta maršruta transporta pienākšanas laikus pieturā.
Pieejas tiesības	Visi
Atkarība	L.7: Skatīt pieturas transportu sarakstu.
Dod pieeju pie	L.9: Skatīt pienākšanas laikus nākamajās pieturās.

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Maršruts	Izvēle no saraksta	-	Jā	Izvēle no L.7 funkcijas attēlotā maršrutu saraksta.

#### Apstrāde

Sistēmā tiek sameklēti laiki, kādos izvēlētais maršruts pienāk pieturā.

#### Izvaddati

Parāda sarakstu, kurā augšdaļa ir maršruta numurs un nosaukums. Saraksts sastāv no laikiem, kuros transportlīdzeklis pienāk pieturā.

Kļūdu apstrāde:

Ja pienākšanas laikus neizdevās atrast, parāda paziņojumu ERR\_NDTF.

### 3.1.9. Skatīt pienākšanas laikus nākamajās pieturās

ID	L.9
Mērķis	Šī funkcija nodrošina lietotājam iespēju apskatīt maršruta pienākšanas laikus nākamajās pieturās, atkarībā no tā pienākšanas laika izvēlētāja pieturā.
Pieejas tiesības	Visi
Atkarība	L.8: Skatīt transportu.
Dod pieeju pie	L.10: Skatīt maršrutu kartē.

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Pienākšanas laiks	Izvēle no saraksta	-	Jā	Izvēle no L.8 funkcijas attēlotā pienākšanas laiku saraksta

#### Apstrāde

Sistēma tiek sameklēti maršruta pienākšanas laiki pieturās, un pašu pieturu informāciju. Atlasa no iegūtām pieturām tikai tās, kuras ir maršrutā pēc izvēlētas pieturas(pietura, kuras informāciju skatās lietotājs). Izmanto L.8 funkcijas informāciju par maršrutu.

#### Izvaddati

Parāda sarakstu, kurā augšdaļa ir maršruta numurs un maršruta nosaukums. Saraksta pirmais elements ir izvēlēta pietura un pienākšanas laiks tajā. Pārējais saraksts sastāv no sekojošam maršruta pieturām un pienākšanas laikiem tajās.

#### Kļūdu apstrāde:

Ja neviena pietura izņemot izvēlēto netiek atrasta, parāda paziņojumu ERR\_NOSTOPS un atgriež uz funkciju L.8.

### 3.1.10. Skatīt maršrutu kartē

ID	L.10
Mērķis	Šī funkcija nodrošina lietotājam iespēju skatīties kartē, kur atrodas pieturās kurās pienāk dotais maršruts, ka arī savu atrašanas vietu.
Pieejas tiesības	Visi
Atkarība	L.9: Skatīt pienākšanas laikus nākamajās pieturās.
Dod pieeju pie	-

#### Ievaddati

Saņem maršruta pieturas, pienākšanas laikus tajās, informāciju par izvēlēto pieturu, maršrutu un tā pienākšanas laiku pieturā. Lietotāja noteiktas atrašanas vietas ģeogrāfiskas koordinātes no funkcijas L.4.

#### Apstrāde

Pieturam tiek noteikts, kura pietura ir maršruta sākuma pietura un kura no pieturām ir maršruta galapunkts. Pieturas maršrutā pirms izvēlētas pieturas un pieturas pēc tās. Katrai pieturai tiek pievienots tas nosaukums, iela un maršruta pienākšanas laiks tajā. Lietotāja atrašanas vietai tiek izmantotas tas pašas koordinātes, kas funkcijā L.4.

#### Izvaddati

Lietotājam tiek parādīta karte ar pieturām un viņa atrašanas vieta. Kartei noklusētam centram un pietuvinājumam jābūt tādām, lai ietvertu gan lietotāja atrašanas vietas marķieri, gan pieturu marķierus uz kartes. Katra veida pieturu (aprakstīti apstrādē) marķieri ir atšķirīgā krāsā. Pieskaroties pieturai parāda informāciju par to – nosaukums, iela un maršruta pienākšanas laiks tajā. Virs katras pieturas marķiera, lielajā un vidējā pietuvinājumā parādās lauks, kurā ir ierakstīts pienākšanās laiks dotajā pieturā un pieturas nosaukums.

#### Kļūdu apstrāde:

Ja radās problēmas ielādējot karti, parāda paziņojumu - ERR\_IMAP un atgriež uz funkciju L.9.

### 3.1.11. Ieteikt trūkstošo objektu

ID	L.11
Mērķis	Šī funkcija nodrošina lietotājam iespēju saglabāt sistēmā ieteikumu par objektu.
Pieejas tiesības	Visi
Atkarība	L.2: Rādīt sākumlapu.
Dod pieeju pie	-

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Platuma koordināta	Koordināta	-	Jā	Koordinātes tiek iegūtas no punkta, kur pieskārs lietotājs kartei.
Garuma koordināta	Koordināta	-	Jā	
Ieteikta objekta apraksts	Teksts	500	Jā	

#### Apstrāde

Kad lietotājs ievada ieteikta objekta aprakstu, tiek pārbaudīts jau ievadīto simbolu skaits, un ja lietotājs mēģina ievadīt 501 simbolu, tas netiek ievadīts. Arī tiek pārbaudīts vai teksta ievades lauka, nav tikai tukšumi. Kamēr koordinātas un ieteikta objekta apraksts nav ievadīti, lietotājam nevar viņus nosūtīt, nospiežot pogu.

Pēc nosūtīšanas tiek pārbaudīts vai koordinātes un apraksts ir saņemti, un vai tie atbilst saviem datu tipiem. Tiek pārbaudīts, vai ieteikta objekta apraksts nesastāv tikai no atstarpēm.

Ja vis izpildījās veiksmīgi, ieteikums tiek saglabāts sistēmā.

#### Izvaddati

Ja dati ir korekti un ieteikums tika saņemts, tiek parādīts paziņojums MSG\_THX un lietotājs tiek atvērta sākumlapa(L.2).

#### Kļūdu apstrāde:

Ja teksta garums nav robežas no 1 līdz 500 simboliem, parāda paziņojumu ERR\_TXT500. Ja koordinātes tika nosūtītas, bet notika kļūda to apstrāde(neatbilst datu tipam), parāda paziņojumu ERR\_CORD. Ja koordinātes netika norādītas, parāda paziņojumu ERR\_NCORD, bet ja netika norādīts ieteikuma teksts - ERR\_NADTXT.

Ja notiek kāda no augstāk minētajām kļūdām, lietotājam turpina rādīt ieteikumu sadaļu.

### 3.1.12. Mainīt valodu

ID	L.12
Mērķis	Šī funkcija nodrošina lietotājam iespēju nomainīt saskarnes valodu.
Pieejas tiesības	Visi
Atkarība	L.2:Rādīt sāukmlapu.
Dod pieeju pie	-

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Valoda	Izvēle no saraksta	-	Jā	Latviešu, angļu un krievu valoda.

#### Apstrāde

-

#### Izvaddati

Atgriež uz sāukmlapu, un saskarnes valoda atbilsts izvēlētajai.

### 3.1.13. Skatīties ĢeoTviteri

ID	L.13
Mērķis	Šī funkcija nodrošina lietotājam iespēju redzēt iespējamās darbības sistēmas ĢeoTviterī.
Pieejas tiesības	Visi
Atkarība	L.2:Rādīt sāukmlapu.
Dod pieeju pie	L.14: Autorizēties ar tviteri, L.15: Atstāt komentāru, L.16: Lasīt komentārus, L.17: Atteikties.

#### Ievaddati

-

#### Apstrāde

-

#### Izvaddati

Rāda sarakstu ar lietotāja iespējamā darbībām ĢeoTviteri. Ja lietotājs ir autorizējies sistēmā, piedāvā atstāt komentārus(L.15), lasīt savus, draugu, visu lietotāju komentārus(L.16), vai atteikties(L.17). Ja lietotājs nav autorizējies, piedāvā lasīt visus komentārus(L.16) vai autorizēties ar tviteri(L.14).



### 3.1.14. Autorizēties ar tviteri

ID	L.14
Mērķis	Šī funkcija nodrošina lietotājam iespēju pieteikties sistēma ar tvitera kontu.
Pieejas tiesības	Visi
Atkarība	L.13:Skatīties ĢeoTviteri.
Dod pieeju pie	L.14: Atstāt komentāru, L.15: Lasīt komentārus, L.16: Atteikties.

#### Ievaddati

Pašā sistēma lietotājs neko neievada, bet tiek pāradresēts uz tvitera autorizāciju.

#### Apstrāde

Pēc tvitera atgrieztas informācijas, tiek noidentificēts lietotājs un tiek noteikti viņa draugi.

#### Izvaddati

Rāda sarakstu ar L.14, L.15, L.16 iespējamām darbībām, un saraksta galvenē, tiek rādīts lietotāja vārds.

#### Kļūdu apstrāde:

Ja radās kļūdas lietotāja autorizēšanas procesā, mēģina nosūtīt lietotāju vēlreiz uz autorizāciju, ja kļūda atkārtojas parāda paziņojumu ERR\_FAUTH. Rāda ĢeoTvitera sadaļu(L.13).

### 3.1.15. Atstāt komentāru

ID	L.14
Mērķis	Šī funkcija nodrošina lietotājam iespēju atstāt komentāru vietā, kur viņš atrodas. Komentārs tiek publicēts arī lietotāja tvitera lapā.
Pieejas tiesības	Autorizēts lietotājs
Atkarība	L.13:Skatīties ĢeoTviteri, L.14: Autorizēties ar tviteri.
Dod pieeju pie	-

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Platuma koordināta	Koordināta	-	Jā	Koordinātes tiek iegūtās no lietotājas mobilas ierīces atrašanas vietas.
Garuma koordināta	Koordināta	-	Jā	
Komentārs	Teksts	140	Jā	

#### Apstrāde

Tiek noteiktas lietotāja koordinātes. Kad lietotājs ievada komentāru, tiek pārbaudīts jau ievadīto simbolu skaits, un ja lietotājs mēģina ievadīt 141 simbolu, tas netiek ievadīts. Arī tiek pārbaudīts vai teksta ievades lauka, nav tikai tukšumi. Kamēr koordinātas un komentārs nav ievadīti, lietotājs nevar viņus nosūtīt, nospiežot pogu.

Pēc nosūtīšanas tiek pārbaudīts vai koordinātes un komentārs ir saņemti, un vai tie atbilst saviem datu tipam. Tiek pārbaudīts vai komentārs nesastāv tikai no atstarpēm.

Sistēma izveido komentāru ar saiti uz sistēmu, no komentāra izgriežot tādu skaitu simbolu, lai kopā ar saiti būtu 140 simboli. Publicē izveidoto komentāru tviterī, lietotāja kontā. Ja vis izpildījās veiksmīgi, komentārs tiek saglabāts sistēmā.

#### Izvaddati

Ja dati ir korekti un komentārs tika saņemts, tiek parādīts paziņojums MSG\_THX un sadaļa kur var atkal ievadīt komentāru vai skatīties komentārus.

#### Kļūdu apstrāde:

Ja teksta garums nav robežas no 1 līdz 140 simboliem, parāda paziņojumu ERR\_TXT140. Ja koordinātes tika nosūtītas, bet notika kļūda to apstrāde(neatbilst datu tipam), parāda paziņojumu ERR\_CORD. Ja netika norādītas koordinātes vai komentārs parāda paziņojumu - ERR\_UNDEF. Ja notiek kāda no augstāk minētajām kļūdām, lietotāju atgriež uz ĢeoTvitera sadalu(L.13).

Ja lietotāja koordinātes neizdodas noteikt 14 sekunžu laikā, parāda paziņojumu ERR\_TIME14 un lietotājs tiek atgriezts uz ĢeoTvitera sadalu(L.13).

### 3.1.16. Lasīt komentārus

ID	L.16
Mērķis	Šī funkcija nodrošina lietotājam iespēju lasīt savus, savu draugu vai visus sistēma atstātos komentārus, vai kādu noteiktu komentāru.
Pieejas tiesības	Autorizēts lietotājs un visi.
Atkarība	L.13:Skatīties ĢeoTviteri, L.14:Autorizēties ar tviteri.
Dod pieeju pie	-

#### Ievaddati

Lauks	Tips	Garums	Obligāts	Komentāri
Komentāru kategorija	Izvēle no saraksta	-	Jā	Komentāru veidi – mani, draugu, visu lietotāju.  Visi sistēmas lietotāji var lasīt tikai visus komentārus sistēma, bet autorizēts lietotājs var lasīt savus un draugu komentārus. Kategorijas, ka attēlojas ĢeoTvitera sadaļas darbību sarakstā(L.13).
Komentāru skaits	Izvēle no saraksta	-	Jā	Pēdējie 10, 20, 40 vai 50 komentāri. Noklusētā vērtība - 10.

Papildus izvēlei no saraksta, kādam konkrētam komentāram var piekļūt caur tviteri publicētu saiti.

#### Apstrāde

Sistēmā tiek atlasīts jaunākie komentāri, kuru skaits atbilst norādītām. Tiek atlasītās koordinātes, kurās viņi tika publicēti atkarība no izvēlētajās kategorijas. Ja izvēlas skatīties draugu komentārus, tad atlasa to lietotāju komentārus, kuriem lietotais seko tviterī. Ja pieprasa kāda lietotāja konkrēto komentāru tad atlasa tikai to.

#### Izvaddati

Parāda karti uz kuras tiek attēloti marķieri komentārus atrašanas vietās. Noklusētais pietuvinājums ir vidējais un kartes centrs atrodas ģeogrāfiskajās koordinātes ar vērtībām - platums:'56.946238', garums:'24.104905'. Pieskaroties marķierim tiek atvērts komentāra teksts, papildus kurām tiek norādīt autora vārds un publicēšanas datums. Tiek parādītas divas pogas, ar kuru palīdzību var pārvietoties starp komentāriem. Parāda pogu uz kuru nospiežot parādās saraksts, kurā var mainīt, attēloto komentāru skaitu.

Kļūdu apstrāde:

Ja radās problēmas ielādējot karti, parāda paziņojumu - ERR\_IMAP komentāru

kategorijas izvēles sadaļu.

Ja neviens komentārs netiek atrasts, parāda paziņojumu ERR\_NFTF un atgriež uz komentāru kategorijas izvēli.

### 3.1.17. Atteikties

ID	L.17
Mērķis	Šī funkcija nodrošina lietotājam iespēju atteikties no sistēmas.
Pieejas tiesības	Autorizēts lietotājs
Atkarība	L.13:Skatīties ĢeoTviteri, L.14:Autorizēties ar tviteri.
Dod pieeju pie	-
Ievaddati	
Lietotājs nospiež atteikšanas pogu.	
Apstrāde	
Lietotājs tiek atslēgts no sistēmas.	
Izvaddati	
Lietotajam pazūd pieeja pie autorizēta lietotāja funkcijām.	

## 3.2. Funkcijas sistēmai

### 3.2.1. Atjaunot objektu informāciju

ID	S.1
Mērķis	Šī funkcija nodrošina iespēju atjaunot informāciju par objektiem
Pieejas tiesības	Visi
Atkarība	-
Dod pieeju pie	-
Ievaddati	
-	
Apstrāde	
Funkcija nolasa no OpenStreetMap datus saistītos ar Rīgu, atlasa tādus objektus, kā ātrās ēdināšanas vietas, restorānus, kafejnīcas, bankas, teātrus, bārus, muzejus, kinoteātrus, bankomātus, viesnīcas, stāvvietas, veikalu, un saglabā iegūto informāciju datubāzē.	
Izvaddati	
-	

### 3.2.2. Atjaunot Rīgas sabiedriska transporta informāciju

ID	S.1
Mērķis	Šī funkcija nodrošina iespēju atjaunot informāciju saistīto ar Rīgas sabiedrisko transportu.
Pieejas tiesības	Visi
Atkarība	-
Dod pieeju pie	-
Ievaddati	
-	
Apstrāde	
Funkcija nolasa no <a href="http://www.rigassatiskme.lv">www.rigassatiskme.lv</a> informāciju saistīto ar Rīgas sabiedrisko transportu tādu, kā maršruti, viņu pienākšanas laiki pieturās un saglabā iegūto informāciju datubāzē.	
Izvaddati	
-	

## 4. Ārējā saskarne

### 4.1. Lietotāja saskarne

Sistēmai jāatbalsta skārienekrāniem pieejamos pirkstu žestus, kuri izpilda pogu uzspiešanu, ritināšanu, tālummaiņu.

Ja darbība prasa informācijas ielādēšanu, iegūšanu no ārpusē, lietotājam jāparada ielādes logs.

Ja operācijas rodas kļūdas, lietotājam jāparada paziņojums, kas informē par to. Paziņojumi izmantos noklusēto mobila tālruņa pārlūka dizainu (*skat.4.1.att.*).



4.1. att. Paraugs paziņojumam

Ekranformas augšdaļā jāatrodas josla 40px augstumā un 100% platumā, turpmāk šī josla sauksies par galveni. Uz galvenes atradīsies poga [Atpakaļ] . Poga [Atpakaļ] atgriez lietotāju uz iepriekšēju sadaļu. Galvenes vidū atrodas sadaļas nosaukums.

Ekranformu piemērus skatīt pielikumā „Lietotnes formu piemēri”.

### 4.2. Aparatūras saskarne

Lai piekļūtu sistēmai, klientam jāizmanto kāda no pārlūkprogrammām, kuras minētas "Programmatūras saskarnes" aprakstā. Tāpēc minimālas prasības, kuras nepieciešamas lietotāja mobilai ierīcei, atkarīgas no pārlūkprogrammas minimālajām prasībām.

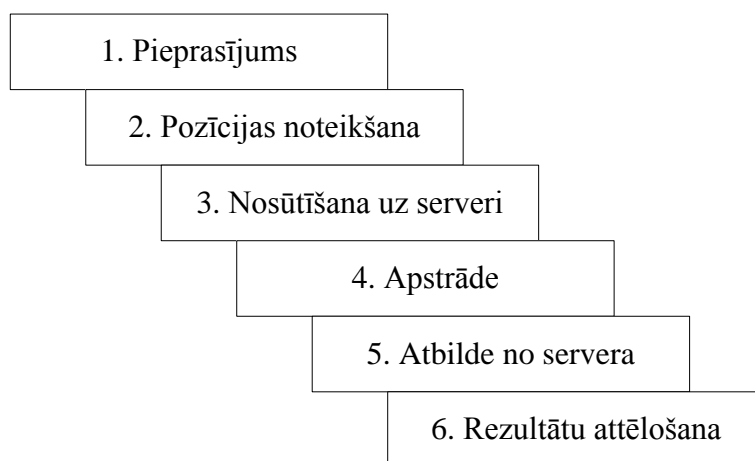
### 4.3. Programmatūras saskarne

Sistēma pieejama caur pārlūkprogrammu. Ieteicams izmantot kādu no šīm pārlūkprogrammām: Safari, Chromium. Programmai korekti jāstrādā uz sekojošām operētājsistēmām: iOS 4.x, Android 2.1. Izņemums tas, ka Android 2.1. neatbalsta Google maps pietuvinājumu ar pirkstiem(pinch-to-zoom).

## 5. Nefunkcionālās prasības

### 5.1. Veiktspējas prasības

Sistēmai jānodrošina 500 lietotāju vienlaicīgu sistēmas lietošanu. Sistēmas ātrdarbība ir atkarīga no interneta pieslēguma serverim un lietotāja mobilai ierīcei, ka arī no mobilas ierīces parametriem un ātrdarbības. Takā šos lielumus sistēma nevar ietekmēt, vienīgais posms, ko var nodrošināt pati sistēma, ir datu apstrāde uz servera (skat.Att:6.). Datu apstrādi jāveic 1-14 sekunžu laikā. Ja atbilde netiek saņemta 14 sekunžu laikā, jāpārtrauc pieprasījums. Ja tiek nodrošināts 2MB/s interneta pieslēgums mobilai ierīcei, visu posmu jāveic 15 sekunžu laikā pēc pirmā pieprasījuma (kad cilvēks pieslēdzās sistēmai), un 1-14 sekundēs turpmākajos pieprasījumos.



5.1. att. Sistēmas un lietotāja sadarbības posmi

### 5.2. Pieejamība

Sistēmai jābūt nepārtraukti pieejamai 18 stundas dienā laika posmā no 6:00 līdz 24:00.

### 5.3. Lietojamība

Sistēma ir paredzēta lietotājiem, kuriem ir pieredze darbā ar viedai tālruni (smart phone) un skārienjūtīgo ekrānu, interneta pārlūkprogrammām. Lietotāji, kuriem ir augstāk mīnēta pieredze, varēs strādāt ar sistēmu, bez papildus zināšanu iegūšanas.

Sistēmas saskarne ir pielāgota tā, ka teksta lasīšana un pogu nospiešana nerāda grūtības, un lietotājs viegli un ātri var aptvert visu viņam sniegto informāciju.

Sistēmas saskarne nav speciāli piemērota invalīdiem.



## **5.4. Datu drošība**

Katru reizi pirms palaist funkcijas, kas uzlabo, maina datus datubāzē, datubāzi jānoeksportē. Serverim saņemot datus, jāpārbauda vai ir padoti nepieciešamie dati, jāpārbauda to garumi un tipi. Serverim jābūt aizsargātam arī no gadījumiem, kad lietotais izmanto kādu JavaScript rīku, lai izsaukt kādas funkcijas, un mēģina nosūtīt nekorektos datus. Veicot saglabāšanu datubāzē, vajag aizsargāt to no injekcijām.

## **5.5. Uzticamība**

Ja lietotājs ievada kļūdainus datus, vai veic darbības neatbilstošas sagaidāmajām, viņam jābūt informētam ar paziņojumu, kurš izskaidro viņa kļūdu. Ja lietotais veica darbību korekti, tad, ja nav paredzēta nepārprotama tālāka darbības attīstība, kas nepārprotami liecina par veiksmīgu darbības izpildi, tad lietotājs obligāti ir jāinformē par veiksmīgu darbības paveikšanu ar paziņojumu.

Lietotāja dati, parole glabājas tviterī un sistēmā glabājas tikai lietotāja lietotājvārds.

# PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

## 1. Ievads

Šajā nodaļā tiek aprakstīts dotais dokuments, tiek definēta mērķauditorija, tiek aprakstīta saistība ar citiem dokumentiem, aprakstīts dokumenta nolūks.

### 1.1. Nolūks

Šī dokumenta nolūks ir aprakstīt "Mobilā lietojumprogrammatūra navigācijai Rīgā" sistēmas uzbūvi. Dokuments paredzēts sistēmas izstrādātājiem. Dokumenta nolūks ir palīdzēt sistēmas analīzē, plānošanā un implementēšanā.

### 1.2. Darbības sfēra

Šī sistēma ir paredzēta lietotājiem, kuriem ir vajadzīga palīdzība navigācijai Rīgā. Tā palīdzēs atrast dažāda tipu objektus, ka arī parādīs sabiedriska transporta pieturu atrašanas vietas un maršrutu sarakstus tajās.

### 1.3. Definīcijas

#### 1.3.1. Izmatotie saīsinājumi

Saīsinājums	Skaidrojums
DPD	Datu plūsmu diagramma - diagramma, ar kuras palīdzību attēlo funkciju savstarpējās atkarības.
ER	Modelis attēlo datubāzes entītiju savstarpējās atkarības.
PPS	Programmatūras prasību specifikācija.
PK	Primārā atslēga- relāciju datu bāzes pārvaldības sistēmā — unikāls lauks, kas identificē datu bāzes tabulas rindiņu.[7.]
FK	Ārējā atslēga – saites ierobežojums, norāda, ka lauks satur norādi uz citas tabulas ierakstu.[7.]
UK	Unikāla atslēga – unikāls lauks, kas nedrīkst atkārtoties datubāzes tabulā.
ID	Identifikators.

Saīsinājums	Skaidrojums
RST	Rīgas sabiedriskais transports

### 1.3.2. Izmantotie jēdzieni

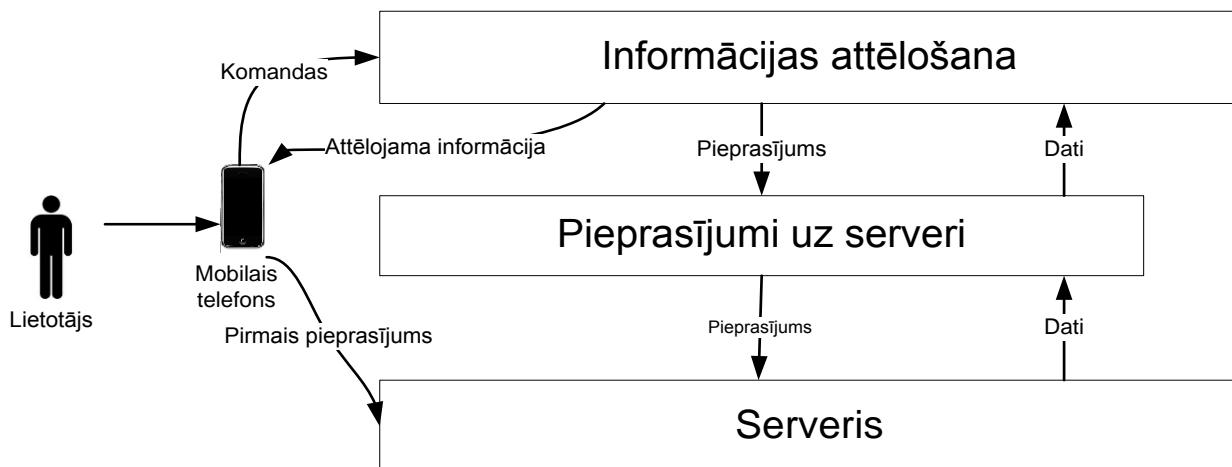
Jēdziens	Skaidrojums
OpenStreetMap	Karte, kuru var izmantot jebkurš (cilvēks, organizācija, uzņēmums, valsts iestāde) jebkuriem mērķiem un kuru var papildināt jebkurš lietotājs ar sevis savāktiem vai trešās puses datiem, ja tos ir atļauja publicēt par brīvu, izmantojot atsauci.[11.]
AJAX	Ir grupa ar savstarpēji saistītiem metodēm, kas tiek izmantotas klienta pusē, lai padarīt lietojumprogrammatūru interaktīvu.
Lietotājs	Persona, kas izmanto lietojumprogrammu savu uzdevumu risināšanai.
Tviteris	Virtuālas komunikācijas un mikroblogošanas tīkls, kurā cilvēki var sūtīt un lasīt savas un citu rakstītas īsas ziņas, kuras sauc par tvītiem (angļu: tweets).[14.]
JavaScript	Firmas Netscape izveidota valoda, kas ļauj globālā tīmekļa izstrādātājiem veidot interaktīvas vietnes. JavaScript var sadarboties ar valodas HTML pirmkoda programmām, tādējādi ļaujot globālā tīmekļa izstrādātājiem piepildīt šīs vietnes ar dinamisku saturu.[7.]

## 1.4. Saistība ar citiem dokumentiem

Šis dokuments tika izstrādāts pēc LVS 72:1996 "Ieteicamā prakse programmatūras projektējuma aprakstīšanai" standarta. Dokuments tika izveidots balstoties uz agrāk izstrādāto PPS. Dokumentu nepieciešams lietot kopā ar PPS.

## 2. Dekompozīcijas apraksts

### 2.1. Moduļu dekompozīcija

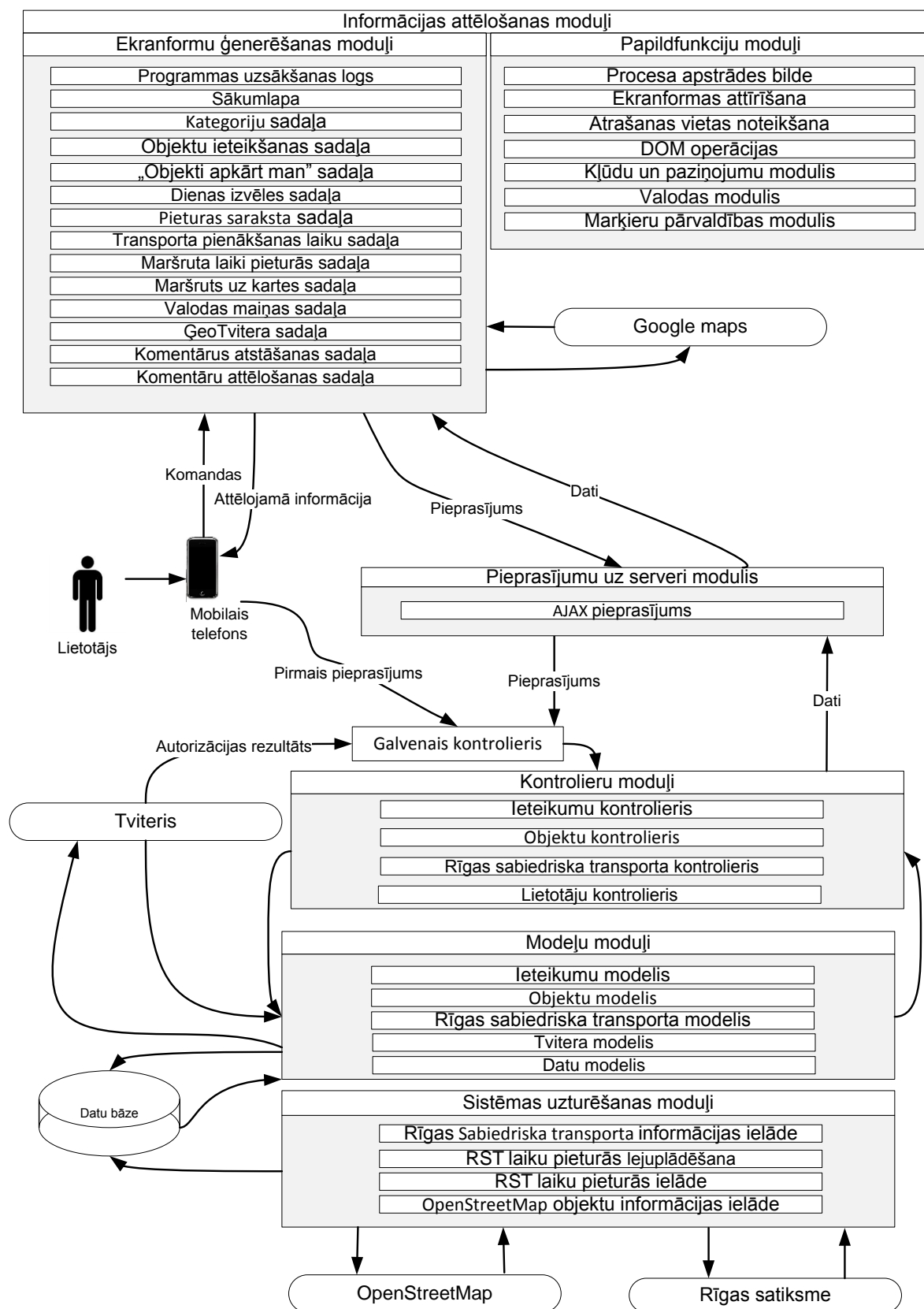


2.1. att. Sistēmas konceptuāla uzbūve

Moduļi ir sadalīti septiņu veidu moduļos:

- Kontrolieris – klase, kas satur operāciju loģisko secību. Tajā atrodas pārsvarā tikai loģiskas darbības, un tā izmanto modeļu moduļus, lai veikt tehniskas darbības tādas, kā datu atlasīšana. Atgriež rezultātu lietotāja pieprasījumiem. Turpmāk tekstā apzīmēsies kā *Controler*. Glabājas mapē *controllers*. Realizācijas valoda – php.
- Modelis – klase, kas satur metodes, kurās paredzētas datu iegūšanai no datubāzes, vai no citās sistēmas, vai aprēķinu veikšanai un citām līdzīgam darbībām. Glabājas mapē *models*. Realizācijas valoda – php.
- Galvenais kontrolieris – ir tikai viens tāds kontrolieris, atrodas saknes direktorija mapē, un saucās *index.php*.
- Skats, jeb sadaļa (ekranformas ģenerētais) – atbild par lietotāja ekranformās izmaiņšanu, aizpildīšanu, izveidošanu. No skatiem tiek veikti AJAX pieprasījumi uz galveno kontrolieri. Šeit tiek saņemti pieprasījumu rezultāti no AJAX pieprasījumiem, un tie tiek apstrādāti un attēloti lietotājam. Glabājas mapē *view*. Realizācijas valoda – javascript, php.
- Papildfunkcija – funkcija, vai funkciju kopa, kura var tikt izmantota citos moduļos. Glabājas mapē *view/library*. Realizācijas valoda – javascript.
- Pieprasījumi uz serveri – atbild par pieprasījuma izveidošanu uz galveno kontrolieri, un atbildes saņemšanu un atgriešanu skatam. Glabājas mapē *view/library*. Realizācijas valoda – javascript.
- Sistēmas uzturēšana – veic datu iegūšanu no ārpusē, saglabā tos lokālajā datubāzē, filtrē datus. Apgāda sistēmu ar datiem. Šos moduļus parastais lietotājs nevar izmantot,

tie tiek palaisti uz paša servera. Glabājas mapē – dbfunctions. Realizācijas valoda – php.



2.2. att. Sistēmas moduļu sadalījums

Faila nosaukums kalpo arī par identifikatoru modulim.

### **2.1.1. Galvenais kontrolieris**

Faili

index.php

Apraksts

Uzsāk lietotāja un sistēmas sadarbību. Nosaka kādu informāciju vēlas iegūt lietotais. Šeit nonāk visi AJAX pieprasījumi no lietotāja mobila telefona. Modulī tiek pārbaudīts vai lietotais uzsāk darbību ar sistēmu, vai turpina jau uzsākto darbību. Tiek pieslēgts programmas uzsākšanas skats ja lietotais veic pirmo pieprasījumu uz sistēmu. Modulis uzsāk sesiju un pieslēdz datubāzi.

---

### **2.1.2. AJAX pieprasījums**

Faili

HTTPPostRequest.js

Apraksts

Izveido AJAX pieprasījumu uz serveri un atgriež skatiem atbildes no kontrolieriem.

---

### **2.1.3. Procesa apstrādes bilde**

Faili

LoadingScreen.js

Apraksts

Rada lietotājam ielādes, jeb procesa apstrādes bildi.

---

### **2.1.4. Ekranformas attīrīšana**

Faili

ClearScreen.js

Apraksts

Atīra ekranformu un sagatavot to jaunas informācijas attēlošanai.

---

### **2.1.5. Atrāšanas vietas noteikšana**

Faili

GeoLocation.js

Apraksts

Nosaka mobila tālruņa atrāšanas vietu .

---

### 2.1.6. DOM operācijas

Fails

DOM.js

Apraksts

Veic darbības ar HTML DOM koku, pievieno un dzēš objektus, pievienot un noņemt objektiem notikumus uz funkcijas kuras jāizsauc ja notikums ir iestājies.

---

### 2.1.7. Kļūdas un paziņojumi

Fails

ErrorAndMessageControl.js

Apraksts

Saņem kļūdas vai paziņojuma identifikatoru, un parada to.

---

### 2.1.8. Valodas apstrāde

Fails

Language.js

Apraksts

Izmantojot saņemto teksta identifikatoru, atgriež attiecīgo tekstu atkarība no valodas.

---

### 2.1.9. Marķieru pārvaldīšanas modulis

Fails

Markers.js

Apraksts

Modulis satur dažādas funkcijas darbam ar marķieriem un karti. Vienā no funkcijām ir marķieru grupēšana. Saņemot masīvu ar marķieru skaitu noteiktajā apgabalā un to koordinātam, atgriež marķierus, kas attēlo saņemtu marķieru grupas. Katrai tādai grupai, atkarībā no elementu skaita, nosaka marķiera bildi. Papildus šai funkcijai, satur tādas funkcijas, kas attīra karti no marķieriem, atrod tuvāko marķieri punktam.

---

### **2.1.10. Iezīmju operācijas**

Fails

Labels.js

Apraksts

Modulis satur funkcijas darbam ar iezīmēm uz Google map. Satur funkcijas, kas maina iezīmju redzamību, izveido iezīmes, izdzēš iezīmes.

---

### **2.1.11. Taimeris**

Fails

Timeout.js

Apraksts

Satur funkcijas, kas uzsāk laika atskaiti, izsauc saņemto funkciju ja laiks ir sasniedzis limitu(14 sekundes), vai aptur laika skaitīšanu. Tiek izmantots lai noteikt cik ilgi bija veikts pieprasījums uz serveri, un lai apturēt operāciju ja atbilde nav saņemta 14 sekunžu laikā.

---

### **2.1.12. Programmas uzsākšanas sadaļa**

Fails

startup.js un startup.php

Apraksts

Attēlo lietotājam slīdrādi ar lietojumprogrammas logo vai citam bildēm. Tiek izveidota pirmatnēja HTML struktūra un pieslēdz visus nepieciešamos skatus. Caur startup.php var tik izsauktā arī uzreiz nepieciešama sadaļa, gadījumos ja atgriežas no tvitera autorizācijas, vai lietotājs caur hipersaiti mēģina apskatīt uz kartes noteiktu tvitera komentāru.

---

### **2.1.13. Sākumlapas sadaļa**

Fails

Menu.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var izvēlēties sistēmas sadaļas.

---



### **2.1.14. Objektu ieteikšanas sadaļa**

Fails

AdvicesLog.js

Apraksts

Attēlo lietotājam ekranformu, kurā viņš var ievadīt ieteicamā objekta aprakstu un norādīt objekta atrašanas vietu kartē.

---

### **2.1.15. Kategoriju sadaļa**

Fails

ObjectCategories.js

Apraksts

Attēlo lietotājam sadaļu, kurā lietotājs var izvēlēties objektu kategoriju, kurus viņš vēlas redzēt uz kartes.

---

### **2.1.16. „Objekti apkārt man” sadaļa**

Fails

ObjectsAround.js

Apraksts

Attēlo lietotājam ekranformu ar karti uz kuras parādīti izvēlētas kategorijas objekti viņam apkārt. Lietotājs var mainīt objektu meklēšanas apgabalu. Nospiežot uz kāda no objektiem, parādās tā apraksts. Ja tas ir pietura, tiek atvērta dienas izvēles sadaļa. Objekti tiek grupēti ja pietuvinājums ir mazāks vai vienāds ar 15.

---

### **2.1.17. Dienas izvēles sadaļa**

Fails

ChDayType.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var izvēlēties, kurā dienā viņš vēlas skatīties transportu uz pieturas.

---

### **2.1.18. Pieturas saraksta sadaļa**

Fails

StopsSchedulBoard.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var redzēt pieturā pienākošo transportu sarakstu un izvēlēties, kura transporta pienākšanas laikus viņš vēlas apskatīt.

---

### **2.1.19. Transporta atiešanas laiku sadaļa**

Fails

TransportDepartureTime.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var redzēt transporta pienākšanas laikus pieturā. Nodrošina lietotājam iespēju izvēlēties laiku, kurām viņš gribēs redzēt maršruta pienākšanas laikus citās pieturās.

---

### **2.1.20. Maršruta laiku pieturās sadaļa**

Fails

TripOnStops.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var redzēt maršruta pienākšanas laikus pieturās.

---

### **2.1.21. Maršruta uz kartes sadaļa**

Fails

TripOnMap.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var redzēt karti ar maršruta pieturām. Nodrošina lietotājam iespēju apskatīt katras pieturas nosaukumu un pienākšanas laikus tajās.

---

### **2.1.22. Valodas maiņas sadaļa**

Fails

LanguageChange.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var mainīt sistēmas saskarnes valodu.

---

### 2.1.23. ĢeoTvitera sadaļa

Fails

GeoTwitter.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var redzēt iespējamās darbības saistība ar ĢeoTviteri, tādas kā autentificēšanas, komentāru atstāšana, savu, draugu un visu lietotāju komentāru lasīšana, vai atteikšanas iespēju.

---

### 2.1.24. Komentāru atstāšanas sadaļa

Fails

TweetForm.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var redzēt savu atrašanas vietu un ievadīt komentāra tekstu, kuru viņš vēlas publicēt.

---

### 2.1.25. Komentāru attēlošanas sadaļa

Fails

TweetMap.js

Apraksts

Modulis izveido ekranformu, kurā lietotājs var lasīt un skatīties atstātus komentārus uz kartes, atkarība no izvēlētas kategorijas.

---

### 2.1.26. Ieteikumu kontrolieris

Fails

Advices.php

Apraksts

Veic operācijas ar lietotāja ieteikumiem izmantojot modeļus un atgriezt pieprasīto operāciju rezultātus.

---

## **2.1.27.      Objektu kontrolieris**

Fails

Objects.php

Apraksts

Sagatavo saņemtos datus darbam, pieslēdz uzdevuma veikšanai nepieciešamos modeļus un atgriež pieprasīto operāciju rezultātus vai paziņojumu kodus. Atgriež objektu kategorijas, objektus vai to grupas.

---

## **2.1.28.      Rīgas sabiedriska transporta kontrolieris**

Fails

RigasTransport.php

Apraksts

Sagatavo saņemtos datus darbam, pieslēdz uzdevuma veikšanai nepieciešamos modeļus un atgriež pieprasīto operāciju rezultātus vai paziņojumu kodus. Atgriež maršrutus uz pieturām, maršrutu pienākšanas laikus un pieturas maršrutam.

---

## **2.1.29.      Lietotāju kontrolieris**

Fails

User.php

Apraksts

Sagatavo saņemtos datus darbam, pieslēdz uzdevuma veikšanai nepieciešamos modeļus un atgriež pieprasīto operāciju rezultātus vai paziņojumu kodus. Pāradresē lietotāju uz tvitera autentifikāciju, saņem lietotāja datus, publicē lietotāja komentārus vai atgriež komentārus pieprasītajā kategorijā. Saņem pieprasījumu caur saiti uz komentāru un veic nepieciešamas darbības, lai attēlot pieprasīto komentāru.

---

## **2.1.30.      Objektu modelis**

Fails

Objects\_model.php

Apraksts

Modulis satur metodes, kas veic: objektu kategoriju atlasīšanu no datubāze, objektu atlasīšanu no datubāzes, objektu grupēšanu atkarībā no atrašanas vietas, atlasa objektus, kas ir noteiktajā apgabalā. Darbību rezultātus atgriež kontrolieriem.

---

### **2.1.31. Rīgas sabiedriskā transporta modelis**

Fails

RigasTransport\_model.php

Apraksts

Modulis satur metodes, kas veic: pieturu atlasīšanu no datubāzes, maršrutu uz noteiktas pieturas atlasīšanu no datubāzes, atlasa maršruta atiešanas laikus pieturā no datubāzes, sakārto atiešanas laikus, atgriež pieturas maršrutam. Darbību rezultātus atgriež kontrolieriem.

---

### **2.1.32. Ieteikumu modelis**

Fails

Advice\_model.php

Apraksts

Modulis satur metodi, kas veic ieteikta objekta informācijas saglabāšanu datubāzē.

---

### **2.1.33. Tvitera modelis**

Fails

Twitter\_model.php

Apraksts

Modulis satur metodes, kas veic: lietotāja nosūtīšanu uz tvitera autentifikāciju, tvitera atbildes apstrādes, lietotāju komentāru publicēšanu sistēmā un tviterī, komentāru atlasīšanas, lietotāju datu apstrādes.

---

### **2.1.34. Datu modelis**

Fails

Data\_model.php

Apraksts

Modulis satur metodes, kas veic datu pārbaudi atbilstoši to tipam, garumiem un pārbauda vai dati ir saņemti.

---

### **2.1.35. Rīgas Sabiedriska transporta informācijas ielāde**

Fails

RSTLOAD.php

Apraksts

Modulis veic datubāzes aizpildīšanu ar datiem par maršrutiem un pieturām. Modulis sūta pieprasījumus uz [www.rigassatiksm.lv](http://www.rigassatiksm.lv) pēc XML failiem, kuros ir nepieciešamā informācija, apstrādā failus, saglabā starprezultātus paša veidotos XML failos un izmanto tos, lai aizpildīt datubāzi ar nepieciešamo informāciju.

---

### **2.1.36. RST laiku pieturās lejuplādēšana**

Fails

RSTDTLOAD.php

Apraksts

Modulis ielādē no [www.rigassatiksm.lv](http://www.rigassatiksm.lv) XML failus kuri satur informāciju par maršrutu pienākšanas laikiem citās pieturās, atkarība no atiešanas laika kāda no maršruta pieturām.

---

### **2.1.37. RST laiku pieturās ielāde**

Fails

RSTDTSAVE.php

Apraksts

Modulis apstrādā moduļa RST laiku pieturās lejuplādēšana iegūtos XML failus, un saglabā pienākšanas laikus datubāzē.

---

### **2.1.38. OpenStreetMap objektu informācijas ielāde**

Fails

OSMLOAD.php

Apraksts

Modulis iegūst informāciju par Rīga esošiem objektiem no [www.openstreetmap.org](http://www.openstreetmap.org), veido apkopotas informācijas XML failu, un saglabā nepieciešamo informāciju datubāzē.

---

## 2.1.39. Notikumi

Fails

EVENT.php

Apraksts

Fiksē failā EVENT.txt padotu kļūdas ziņojumu no servera uzturēšanas moduļiem.

## 2.2. Datu dekompozīcija

Datu bāzē tiks veidotas sekojošas datu tabulas:

Datu tabula	Tabulas nolūks
Objects	Tabulas nolūks ir uzglabāt informāciju par OpenStreetMap objektiem.
ObjectCategories	Tabulas nolūks ir uzglabāt informāciju par iespējamām objektu kategorijā.
Stops	Tabulas nolūks ir uzglabāt informāciju par Rīgas sabiedriska transporta pieturām.
Trip	Tabulas nolūks ir uzglabāt informāciju par Rīgas sabiedriska transporta reisiem.
TransportType	Tabulas nolūks ir uzglabāt informāciju par Rīgas sabiedriska transporta veidiem.
TripOnStops	Tabulas nolūks ir uzglabāt informāciju par reisiem(maršrutiem), kuri ir pieturās.
DepartureTime	Tabulas nolūks ir reisu atiešanas laiku no pieturām uzglabāšana.
Advices	Tabulas nolūks ir uzglabāt informāciju par lietotāju ieteiktiem objektiem.
Timing	Tabulas nolūks ir uzglabāt informāciju par pienākšanas laikiem nākamajās pieturvietās.
Users	Tabulas nolūks ir uzglabāt informāciju par lietotājiem.
Tweets	Tabulas nolūks ir uzglabāt lietotāju komentārus.

### 3. Atkarības apraksts

#### 3.1. Starpmoduļu atkarības

Tabulā tiks izmantoti moduļu failu nosaukumi.

Modulis	Izmanto moduļus	Tiek izmantots moduļos
index.php	startup.php, startup.js, Advices.php, Objects.php, RigasTransport.php, User.php	HTTPPostRequest.js, User.php
startup.php	Labels.js, Timeout.js, Language.js, ErrorAndMessageControl.js, Markers.js, HTTPPostRequest.js, ClearScreen.js, LoadingScreen.js, DOM.js, GeoLocation.js, Startup.js, Menu.js, ObjectCategories.js, ObjectsAround.js, AdvicesLog.js, ChDayType.js, StopsSchedulBoard.js, TransportDepartureTime.js, TripOnStops.js, TripOnMap.js, LanguageChange.js, GeoTwitter.js, TweetForm.js, TweetMap.js	index.php, Users.php
HTTPPostRequest.js	index.php	AdvicesLog.js, ObjectCategories.js, ObjectsAround.js, StopsSchedulBoard.js, TransportDepartureTime.js, TripOnStops.js, TweetForm.js, TweetMap.js
LoadingScreen.js		AdvicesLog.js, ObjectCategories.js, ObjectsAround.js, StopsSchedulBoard.js, TransportDepartureTime.js, TripOnStops.js, TripOnMap.js, Menu.js, LanguageChange.js, GeoTwitter.js, TweetForm.js, TweetMap.js
ClearScreen.js		AdvicesLog.js, ObjectCategories.js, ObjectsAround.js, Menu.js, LanguageChange.js, GeoTwitter.js, TweetForm.js, TweetMap.js



Modulis	Izmanto modulis	Tiek izmantots modulis
GeoLocation.js		ObjectsAround.js (TripOnMap.js), TweetForm.js
DOM.js		AdvicesLog.js, ObjectCategories.js, ObjectsAround.js, StopsSchedulBoard.js, TransportDepartureTime.js, TripOnStops.js, TripOnMap.js, Menu.js, LanguageChange.js, GeoTwitter.js, TweetForm.js, TweetMap.js
ErrorAndMessageControl.js		AdvicesLog.js, ObjectCategories.js, ObjectsAround.js, StopsSchedulBoard.js, TransportDepartureTime.js, TripOnStops.js, TripOnMap.js, Menu.js, GeoTwitter.js, TweetForm.js, TweetMap.js
Language.js		AdvicesLog.js, ObjectCategories.js, ObjectsAround.js, StopsSchedulBoard.js, TransportDepartureTime.js, TripOnStops.js, TripOnMap.js, Menu.js, ChDayType.js, LanguageChange.js, GeoTwitter.js, TweetForm.js, TweetMap.js
Markers.js		ObjectsAround.js, TripOnMap.js, TweetMap.js
Labels.js		ObjectsAround.js, TripOnMap.js
Timeout.js		AdvicesLog.js, ObjectCategories.js, ObjectsAround.js, StopsSchedulBoard.js, TransportDepartureTime.js, TripOnStops.js, TweetForm.js, TweetMap.js
startup.js	Menu.js	index.php, Users.php
Menu.js	LoadingScreen.js, ClearScreen.js, DOM.js, Language.js	ObjectsAround.js, ObjectCategories.js
AdvicesLog.js	LoadingScreen.js, ClearScreen.js, DOM.js, HTTPPostRequest.js, Language.js,	Menu.js

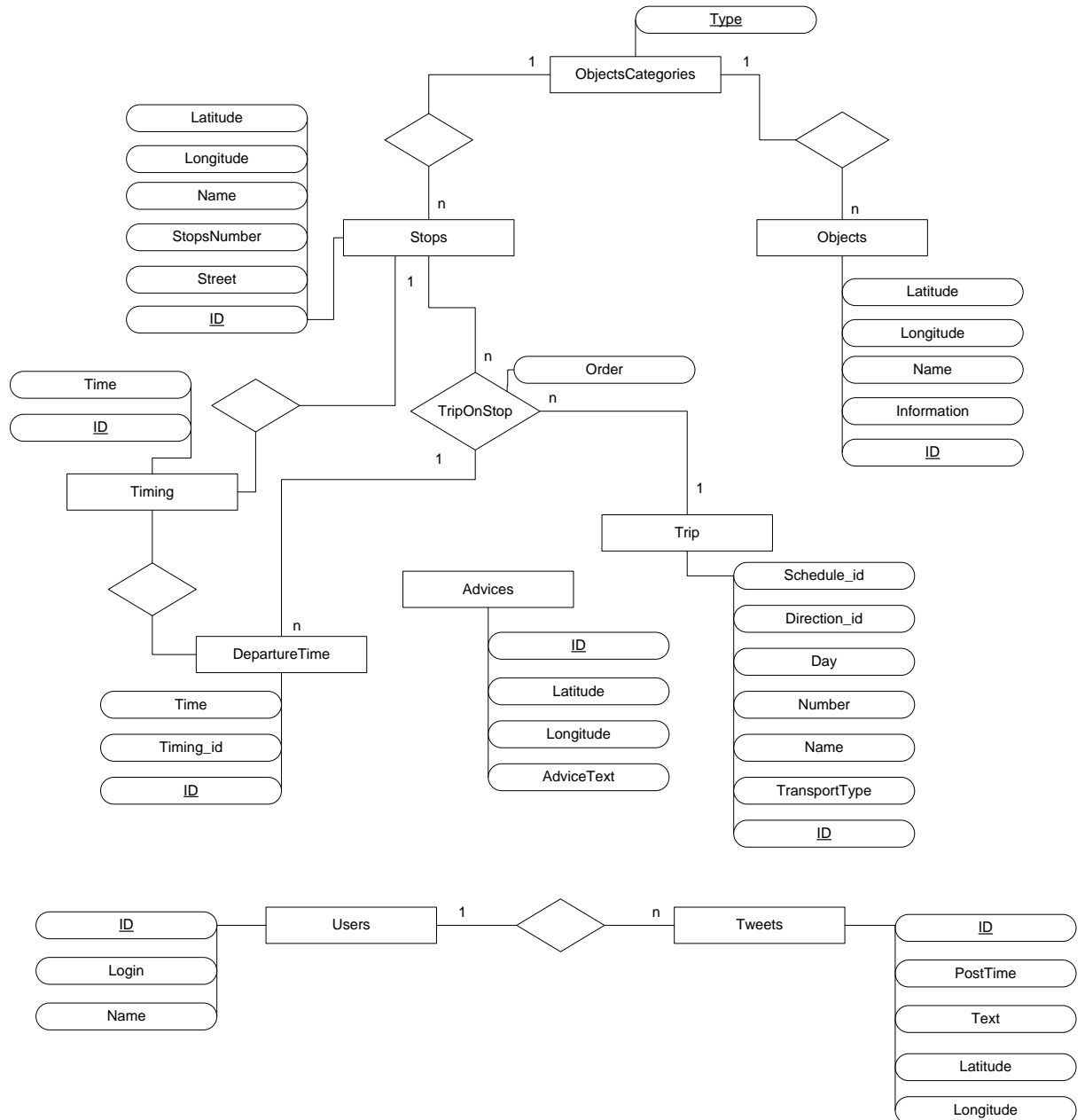
Modulis	Izmanto moduļus	Tiek izmantots moduļos
	ErrorAndMessageControl.js, Timeout.js	
ObjectCategories.js	LoadingScreen.js, ClearScreen.js, DOM.js, HTTPPostRequest.js, ObjectsAround.js, Language.js, ErrorAndMessageControl.js, Timeout.js	Menu.js
ObjectsAround.js	LoadingScreen.js, ClearScreen.js, DOM.js, HTTPPostRequest.js, Language.js, ErrorAndMessageControl.js, GeoLocation.js, Markers.js, ChDayType.js, Timeout.js	ObjectCategories.js
ChDayType.js	LoadingScreen.js, DOM.js, Language.js, StopsSchedulBoard.js	ObjectsAround.js
StopsSchedulBoard.js	LoadingScreen.js, TransportDepartureTime.js, DOM.js, HTTPPostRequest.js, Language.js, ErrorAndMessageControl.js, Timeout.js	ChDayType.js
TransportDepartureTime.js	LoadingScreen.js, TripOnStops.js, DOM.js, HTTPPostRequest.js, Language.js, ErrorAndMessageControl.js, Timeout.js	ObjectsAround.js
TripOnStops.js	LoadingScreen.js, TripOnMap.js, DOM.js, HTTPPostRequest.js, Language.js, ErrorAndMessageControl.js, Timeout.js	TransportDepartureTime.js
TripOnMap.js	LoadingScreen.js, DOM.js, Language.js, Markers.js, ErrorAndMessageControl.js, ( GeoLocation.js)	TripOnStops.js
LanguageChange.js	LoadingScreen.js, DOM.js, Language.js, ClearScreen.js	Menu.js
GeoTwitter.js	LoadingScreen.js, DOM.js, Language.js, ClearScreen.js, TweetForm.js, TweetMap.js, Timeout.js	startup.php, Menu.js
TweetForm.js	LoadingScreen.js, DOM.js, ClearScreen.js, HTTPPostRequest.js, Language.js,	GeoTwitter.js

Modulis	Izmanto moduļus	Tiek izmantots moduļos
	ErrorAndMessageControl.js, Timeout.js, GeoLocation.js	
TweetMap.js	LoadingScreen.js, DOM.js, ClearScreen.js, HTTPPostRequest.js, Language.js, ErrorAndMessageControl.js, Timeout.js	startup.php, GeoTwitter.js
Advices.php	Advices_model.php, Data_model.php	index.php
Objects.php	Objects_model.php, RigasTransport_model.php, Data_model.php	index.php
RigasTransport.php	RigasTransport_model.php, Data_model.php	index.php
User.php	Twitter_model.php, startup.php, index.php, Data_model.php	index.php
Advices_model.php		Advices.php
Objects_model.php		Objects.php
RigasTransport_model.php		Objects.php, RigasTransport.php
Twitter_model.php		User.php
Data_model.php		Advices.php, Objects.php, RigasTransport.php, User.php
RSTLOAD.php		
RSTDTLOAD.php	EVENT.php	
RSTDTSAVE.php	EVENT.php	
OSMLOAD.php		
EVENT.php		RSTDTLOAD.php, RSTDTSAVE.php

## 3.2. Datu atkarības

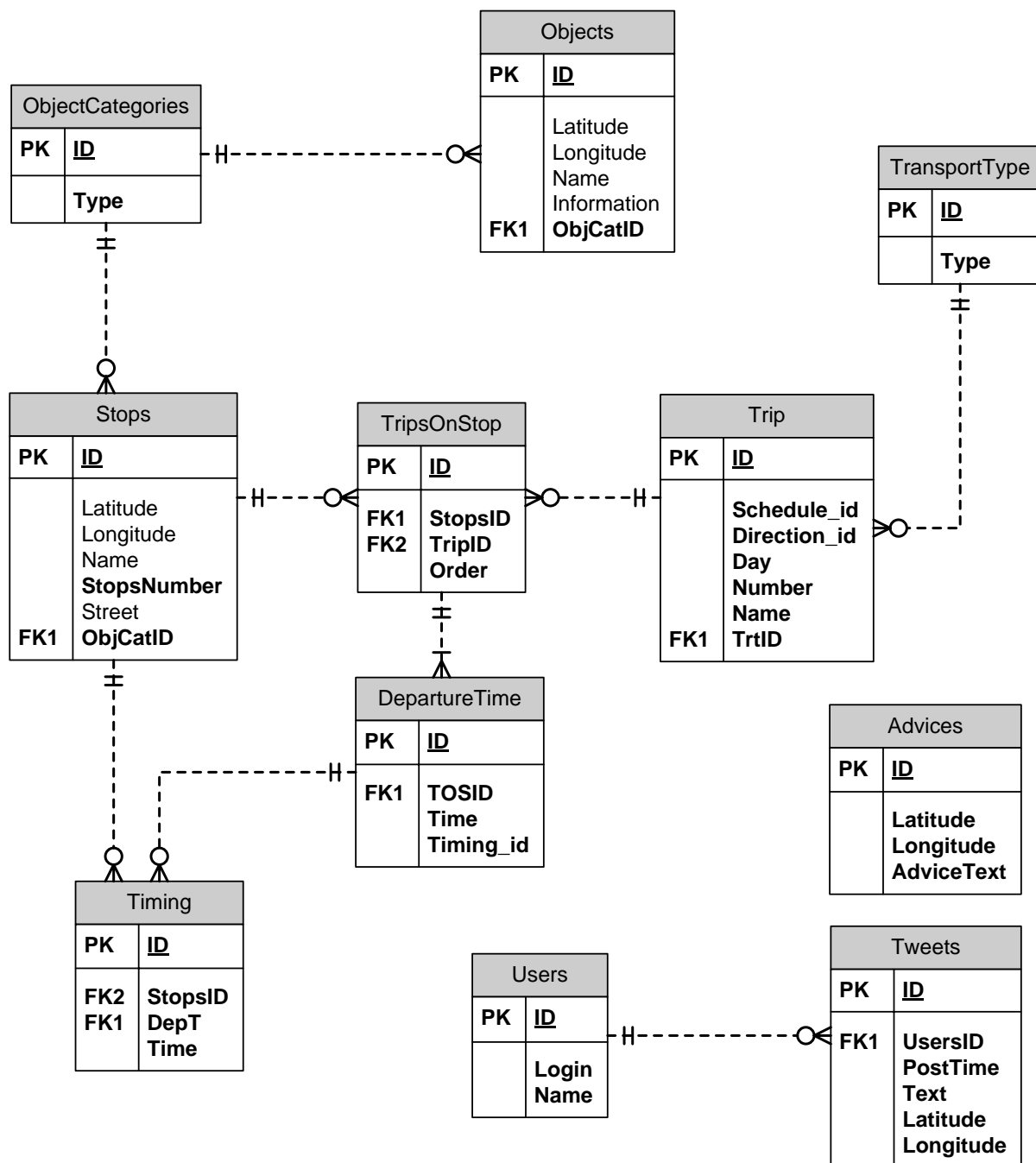
Šajā sadaļā tiek attēlotas datu bāzes tabulu savstarpējās atkarības

### 3.2.1. Konceptuālais ER modelis



3.1. att. Konceptuālais ER modelis

### 3.2.2. Realizācijas ER modelis



3.2. att. Realizācijas ER modelis

## 4. Detalizēts projektējums

### 4.1. Datu detalizēts projektējums

#### 4.1.1. Tabula „Objects”(Obj)

Tabulā „Objects” tiek glabāta informācija par objektiem, kurus sistēma iegūst no OpenStreetMap.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(10)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
Latitude	Float(9,6)	Jā	-	-	Ģeogrāfiskais platums.
Longitude	Float(9,6)	Jā	-	-	Ģeogrāfiskais garums.
Name	Varchar(100)	Jā	-	-	Objekta nosaukums.
Information	Varchar(500)	Nē	-	-	Informācija par objektu.
ObjCatID	Int(4)	Jā	-	FK	ID no tabulas „ObjectCategories”, kas norāda uz objekta kategoriju.

#### 4.1.2. Tabula „ObjectCategories”(ObjCat)

Tabulā „ObjectCategories” tiek glabāta informācija par objektu kategorijām.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(4)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
Type	Varchar(50)	Jā	-	-	Nosaukums objektu tipam.

#### 4.1.3. Tabula „Stops”(Stops)

Tabulā „Stops” tiek glabāta informācija par Rīgas sabiedriska transporta pietrām.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(10)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
Latitude	Float(9,6)	Jā	-	-	Ģeogrāfiskais platums.
Longitude	Float(9,6)	Jā	-	-	Ģeogrāfiskais garums.
Name	Varchar (100)	Jā	-	-	Pieturas nosaukums.
StopsNumber	Int(6)	Jā	-	UK	Pieturu numuri, kas tiek izmantoti Rīgas Satiksmes sniegtos XML failos.
Street	Varchar (100)	Nē	-	-	Iela uz kuras atrodas pietura.
ObjCatID	Int(4)	Jā	-	FK	ID no tabulas „ObjectCategories”, kas norāda uz objekta kategoriju. Šis objekts ir pietura.

#### 4.1.4. Tabula „Trip”(Trip)

Tabulā „Trip” tiek glabāta informācija par reisiem. Reisu definē viņa maršruts, dienā kurā viņš kursē un virziens kurā viņš brauc.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(10)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
Schedule_id	Int(6)	Jā	-	-	Vienam reisam atšķiras ar to vai tas ir brīvdienā vai darbadienā.
Direction_id	Int(6)	Jā	-	-	Parametrs, kas izšķir kurā virzienā brauc transports.
Day	Int(1)	Jā	-	-	0- darbadienā, 1-brīvdiena.
Number	Int(4)	Jā	-	-	Maršruta numurs. Piemēram 8, 53, 3.
Name	Varchar(100)	Jā	-	-	Maršruta nosaukums.
TrtID	Int(1)	Ja	-	FK	ID no tabulas „TransportType”, kas sabiedriska transporta veidu.

#### 4.1.5. Tabula „TransportType”(TrtID)

Tabulā „TransportType” tiek glabāta informācija par Rīgas sabiedriska transporta veidiem.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(1)	Jā	-	PK	Unikālais identifikators. 1 - autobuss; 2 - trolejbuss; 3 - tramvajs.
Type	Varchar(50)	Jā	-	-	Transportlīdzekļa nosaukums.

#### 4.1.6. Tabula „TripOnStops”(TOS)

Tabulā „TripOnStops” tiek glabāta informācija par reisiem pieturās.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(10)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
StopsID	Int(10)	Jā	-	FK	ID no tabulas „Stops”.
TripID	Int(10)	Jā	-	FK	ID no tabulas „Trip”.
Order	Int(5)	Jā	-	-	Kārtas numurs pieturai, noteiktajam maršrutam.

#### 4.1.7. Tabula „DepartureTime”(DepT)

Tabulā „DepartureTime” tiek glabāta informācija reisu atiešanas laikiem no pieturām.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(10)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
TOSID	Int(10)	Jā	-	FK	ID no tabulas „TripOnStops”.
Time	TIME	Jā	-	-	Pienākšanas laiks
Timing_id	Int(10)	Jā	-	-	Id ko izmanto RS lai noteikt laiku.



#### 4.1.8. Tabula „Advices”(Advic)

Tabulā „Advices” tiek glabāti lietotāju ieteiktie objekti.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(10)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
Latitude	Float(9,6)	Jā	-	-	Ģeogrāfiskais platums.
Longitude	Float(9,6)	Jā	-	-	Ģeogrāfiskais garums.
AdviceText	Varchar(500)	Jā	-	-	Ieteikuma apraksts.

#### 4.1.9. Tabula „Timing”(Tim)

Tabulā „Timing” tiek glabāta informācija par noteiktajā laika atvēršanas transportlīdzekļa pienākšanas laikiem nākamajās pieturās.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(10)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
StopsID	Int(10)	Jā	-	FK	ID no tabulas „Stops”. Pietura uz kuru pienaks.
DepT	Int(10)	Jā	-	FK	ID no tabulas „DepartureTimes”.
Time	TIME	Jā	-	-	Laiks cikos pienāks sekojošā pieturā.

#### 4.1.10. Tabula „Users”(Users)

Tabulā „Users” tiek glabāta informācija par lietotājiem, kuri autentificējās sistēmā.

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(10)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
Login	Varchar(255)	Jā	-	UK	Unikālais lietotājvārds ar, kuru pieslēdzās tviterim.
Name	Varchar(255)	Jā	-	-	Lietotāja vārds.

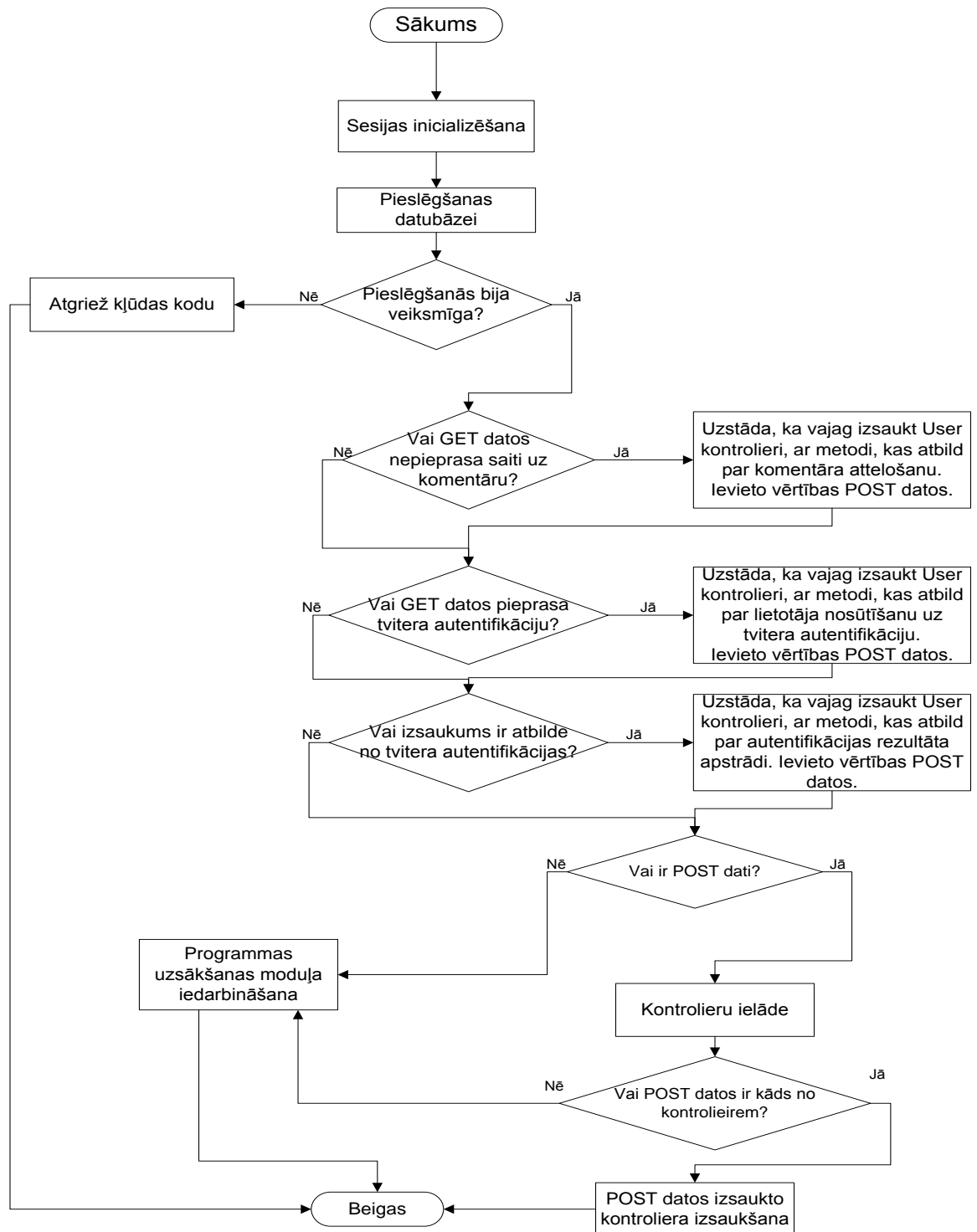
#### 4.1.11. Tabula „Tweets”(Tweets)

Lauka nosaukums	Datu tips	Obligāts	Noklusētā vērtība	PK/FK /UK	Apraksts
ID	Int(10)	Jā	-	PK	Unikālais identifikators. Tiek ģenerēts automātiski.
UsersID	Int(10)	Jā	-	FK	ID no tabulas „Users”.
PostTime	Timestamp	Jā	-	-	Laiks cikos izveidots šis ieraksts datubāzē.
Latitude	Float(9,6)	Jā	-	-	Ģeogrāfiskais platums.
Longitude	Float(9,6)	Jā	-	-	Ģeogrāfiskais garums.
Text	Varchar(140)	Jā	-	-	Komentāra teksts.

## 4.2. Moduļu detalizēts projektējums

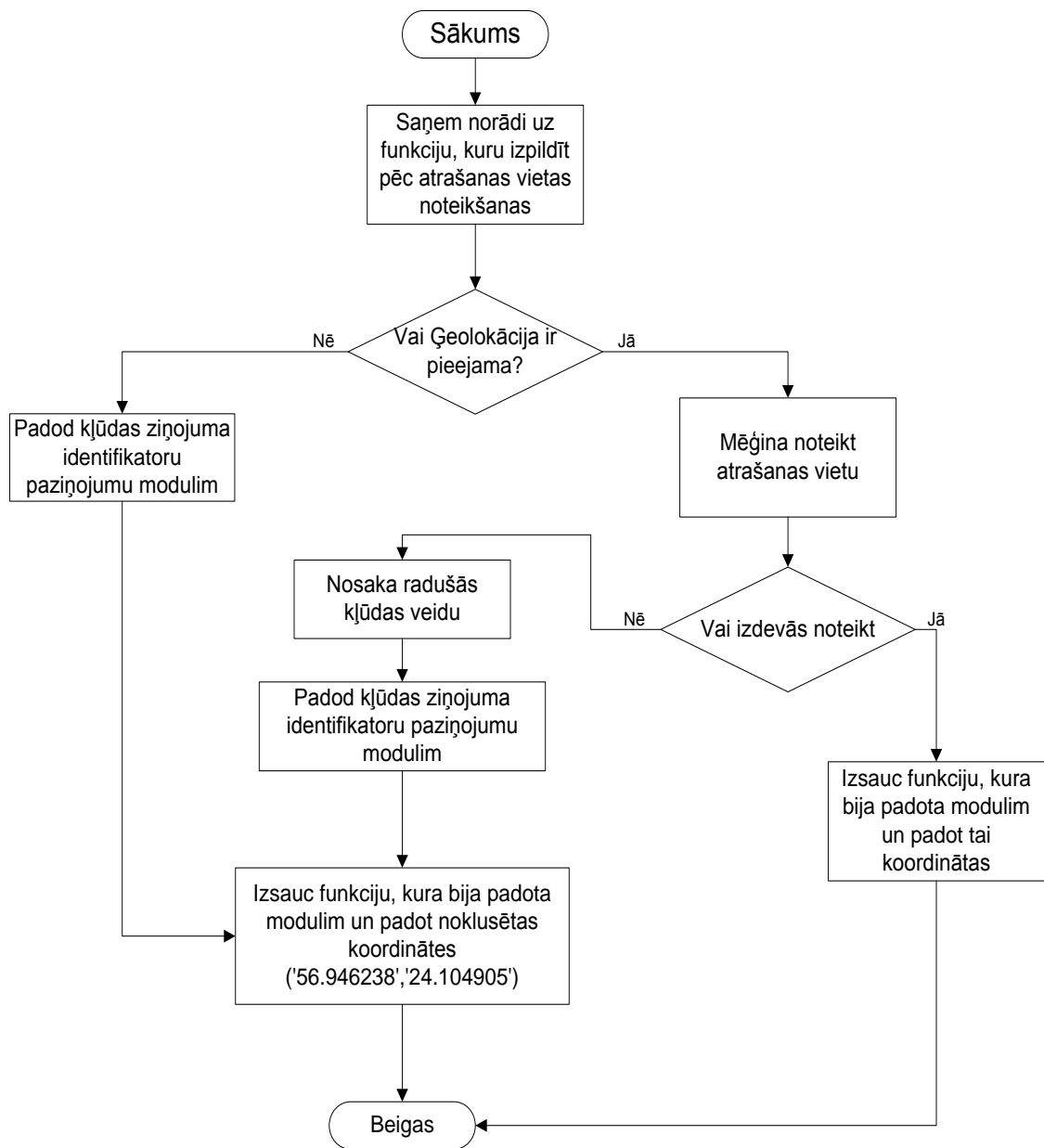
Apgabala robežas, kurā sistēmai ir informācija par objektiem – platums (latitude) 56.700000 līdz 57.100000, garums (longitude) 23.860000 līdz 24.400000. Turpmāk teksta šo apgabalu sauks par „Rīgas apgabals”.

### 4.2.1. Galvenā kontroliera moduļa algoritma projektējums



4.1. att. Galvenā kontroliera moduļa algoritma blokshēma

#### 4.2.2. Atrašanas vieta noteikšanas moduļa algoritma projektējums

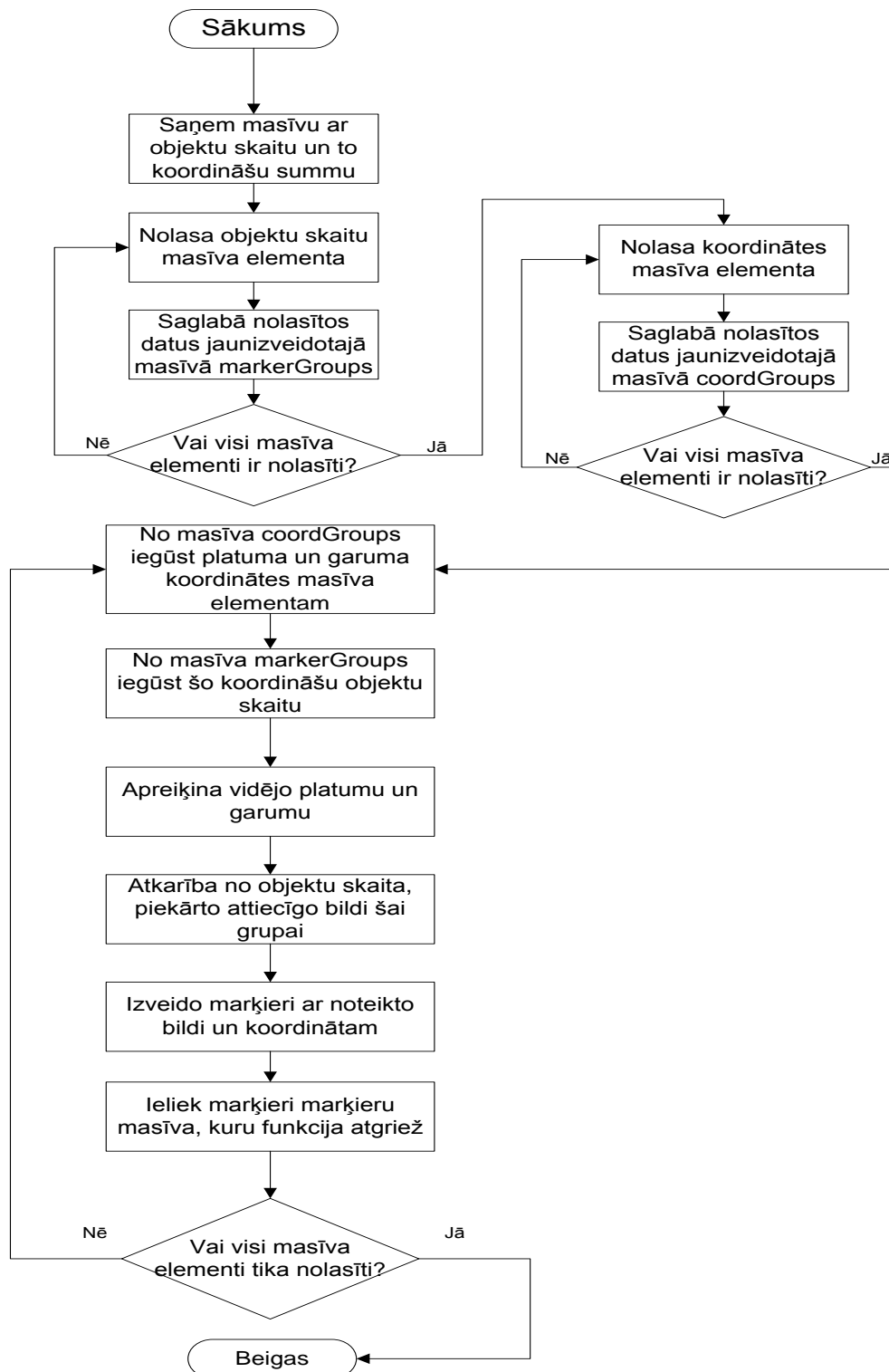


4.2. att. Atrašanas vieta noteikšanas moduļa algoritma blokshēma

### 4.2.3. Marķieru pārvaldības moduļa algoritma projektējums

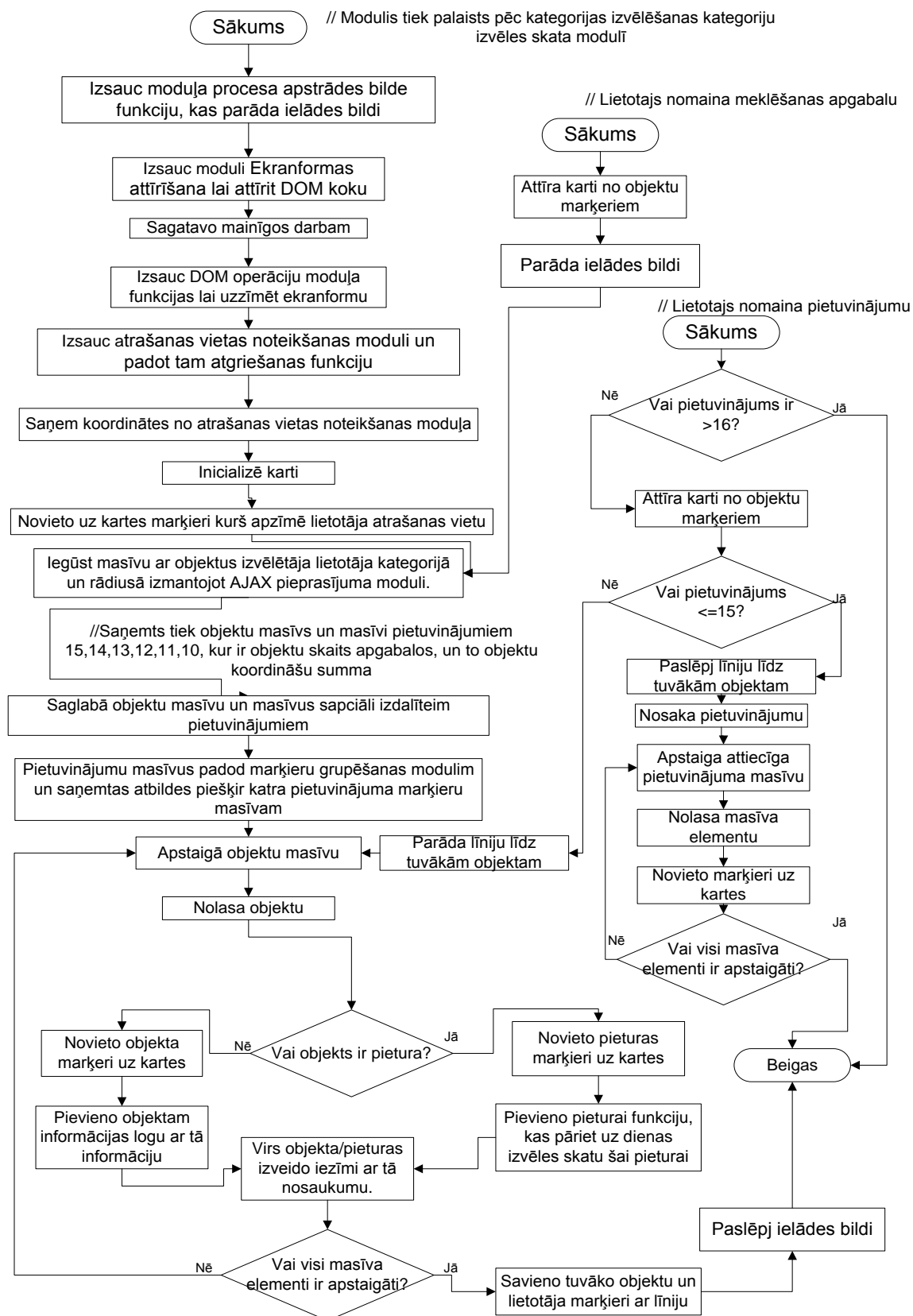
#### 4.2.3.1. Marķieru grupēšanas funkcijas algoritma projektējums

Masīvs, kuru saņem modulis ir divdimensiju masīvs, kur pirmajā dimensija ir objektu skaits, otrajā dimensija ir viņu kopēja koordināšu summa tiem objektiem.



4.3. att. Marķieru grupēšanas moduļa algoritma blokshēma

#### 4.2.4. „Objekti apkārt man” moduļa algoritma projektējums



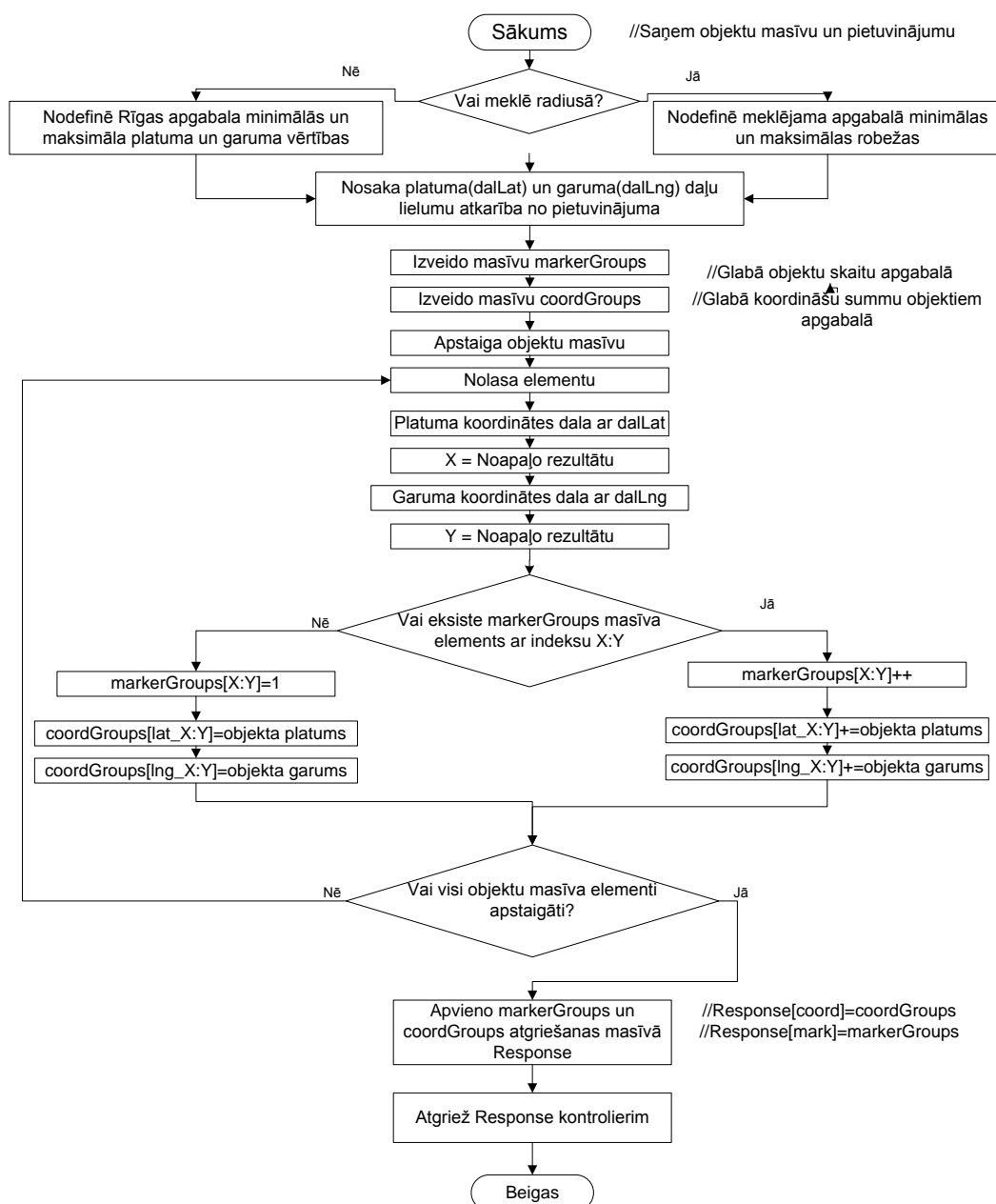
4.4. att. Objektu atrašanas vietas skata moduļa algoritma blokskhēma

Ja lietotais atrodas ārpus Rīgas apgabala, viņu ir jāinformē ar paziņojumu. Ja lietotais atrodas Rīgas apgabalā, jākontrolē to, lai viņš, pārvietojoties kartē, netiktu ārpus tās.

## 4.2.5. Objektu modeļa moduļa algoritma projektējums

### 4.2.5.1. Metodes „Cluster” algoritma projektējums

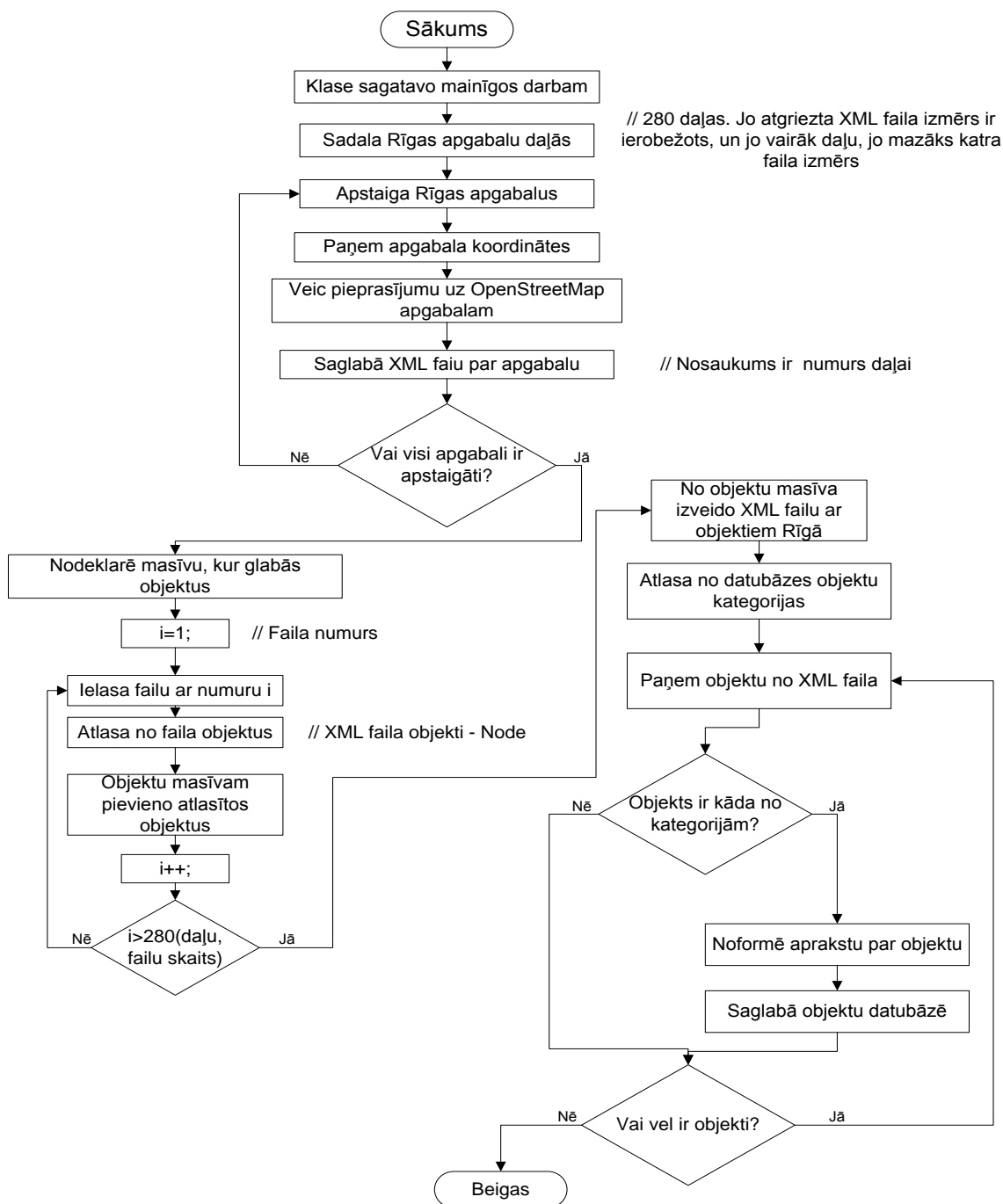
Šī metode izveido no objektu masīva, divdimensiju masīvu, kur pirmajā dimensijā ir objektu skaits grupā, otrā, to koordināšu summa. Objekti tiek apvienoti grupā ja tie ir kādā apgabala iekšienē. Apgabalu izmēri ir definēti katram pietuvinājumam savi: 15 – platums 0.005, garums 0.01; 14 – platums 0.01, garums 0.02 ; 13 – platums 0.02, garums 0.04; 12 – platums 0.04, garums 0.08; 11 – platums 0.08, garums 0.16; 10 – platums 0.16, garums 0.32. Rīgas apgabals tiek sadalīta mazākos apgabalos atkarība no pietuvinājuma un marķieriem tiek noteikts, kurā apgabalā tie atrodas.



4.5. att. Metodes „Cluster” algoritma blokskhēma

#### 4.2.6. OpenStreetMap objektu informācijas ielādes moduļa algoritma projektējums

Modulī informācijas iegūšanai no OpenStreetMap tiek izmantota hipersaite ko piedāvā OpenStreetMap veidotāji- <http://api.openstreetmap.org/api/0.6/map?bbox=<minimālais garums>,<minimālais platums>,<maksimālais garums>,<maksimālais platums>>. Pieprasījums uz doto saiti atgriež XML datus par to reģionu.



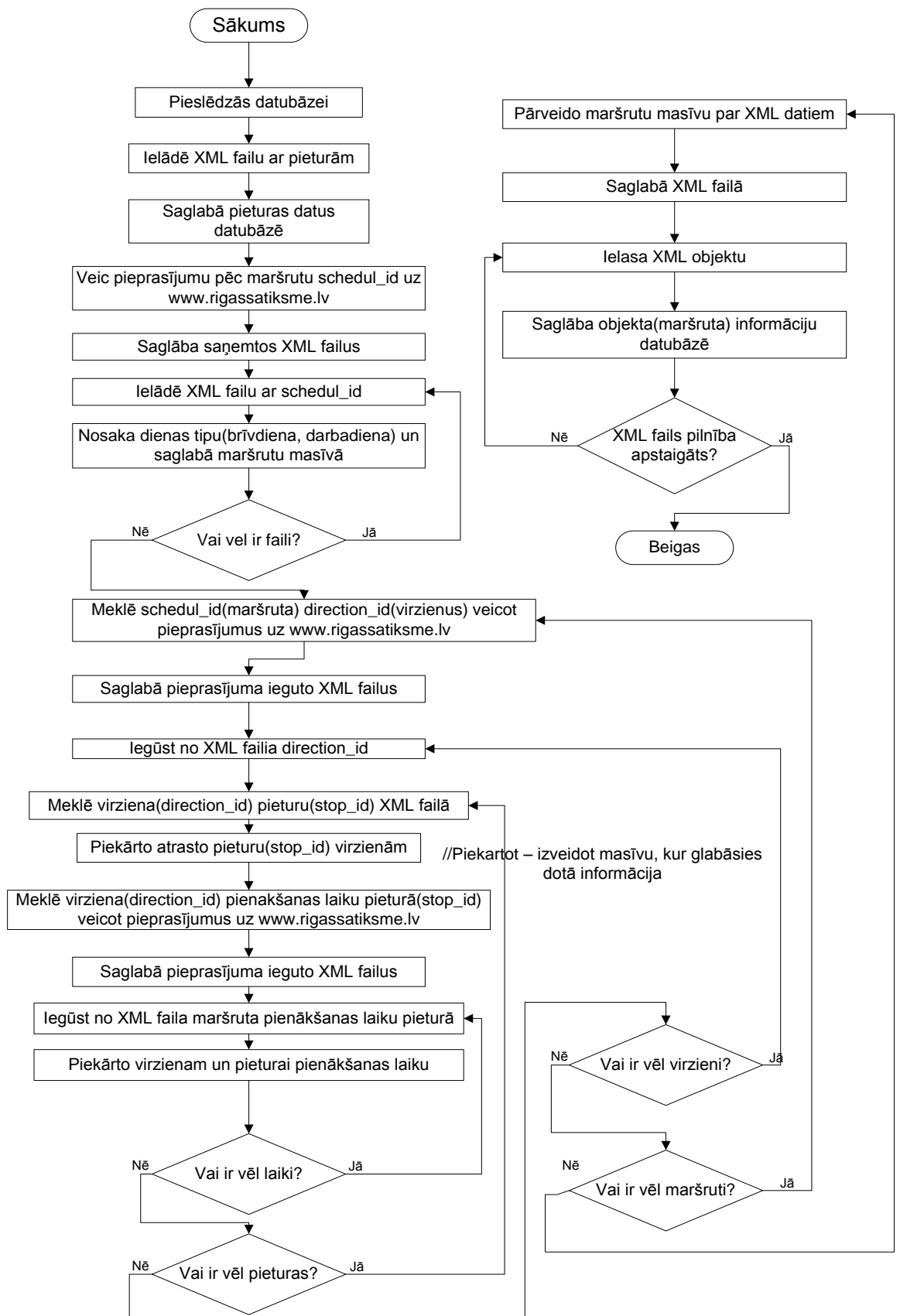
4.6. att. OpenStreetMap objektu informācijas ielādes moduļa algoritma blokshēma



#### 4.2.7. Rīgas Sabiedriskā transporta informācijas ielādes moduļa algoritma projektējums

Modulis iegūst informāciju par maršrutiem no [www.rigassatiksme.lv](http://www.rigassatiksme.lv) XML formātā.

- Nosūtot pieprasījumu uz [http://saraksti.rigassatiksme.lv/?a=p.routes&transport\\_id=<transport\\_id>&t=xml&l=lv](http://saraksti.rigassatiksme.lv/?a=p.routes&transport_id=<transport_id>&t=xml&l=lv) var iegūt XML failu ar informāciju par noteikta transporta veida maršrutiem. Vienam maršrutam ir divi identifikatori – viens darba dienā, otrs brīvdienā. Identifikatoru atslēga ir `schedule_id`.
- Nosūtot pieprasījumu uz [http://saraksti.rigassatiksme.lv/?a=p.directions&schedule\\_id=<schedul\\_id>&t=xml&l=lv](http://saraksti.rigassatiksme.lv/?a=p.directions&schedule_id=<schedul_id>&t=xml&l=lv) var iegūt XML failu ar pieturām maršrutā katrā virziena un to identifikatorus – `stop_id`, kā arī virziena identifikatoru – `direction_id`.
- Nosūtot pieprasījumu uz [http://saraksti.rigassatiksme.lv/?a=p.schedule&direction\\_id=<direction\\_id>&stop\\_id=<stop\\_id>&t=xhtml&l=lv](http://saraksti.rigassatiksme.lv/?a=p.schedule&direction_id=<direction_id>&stop_id=<stop_id>&t=xhtml&l=lv) var iegūt XML failu ar maršruta pienākšanas laikiem pieturā.

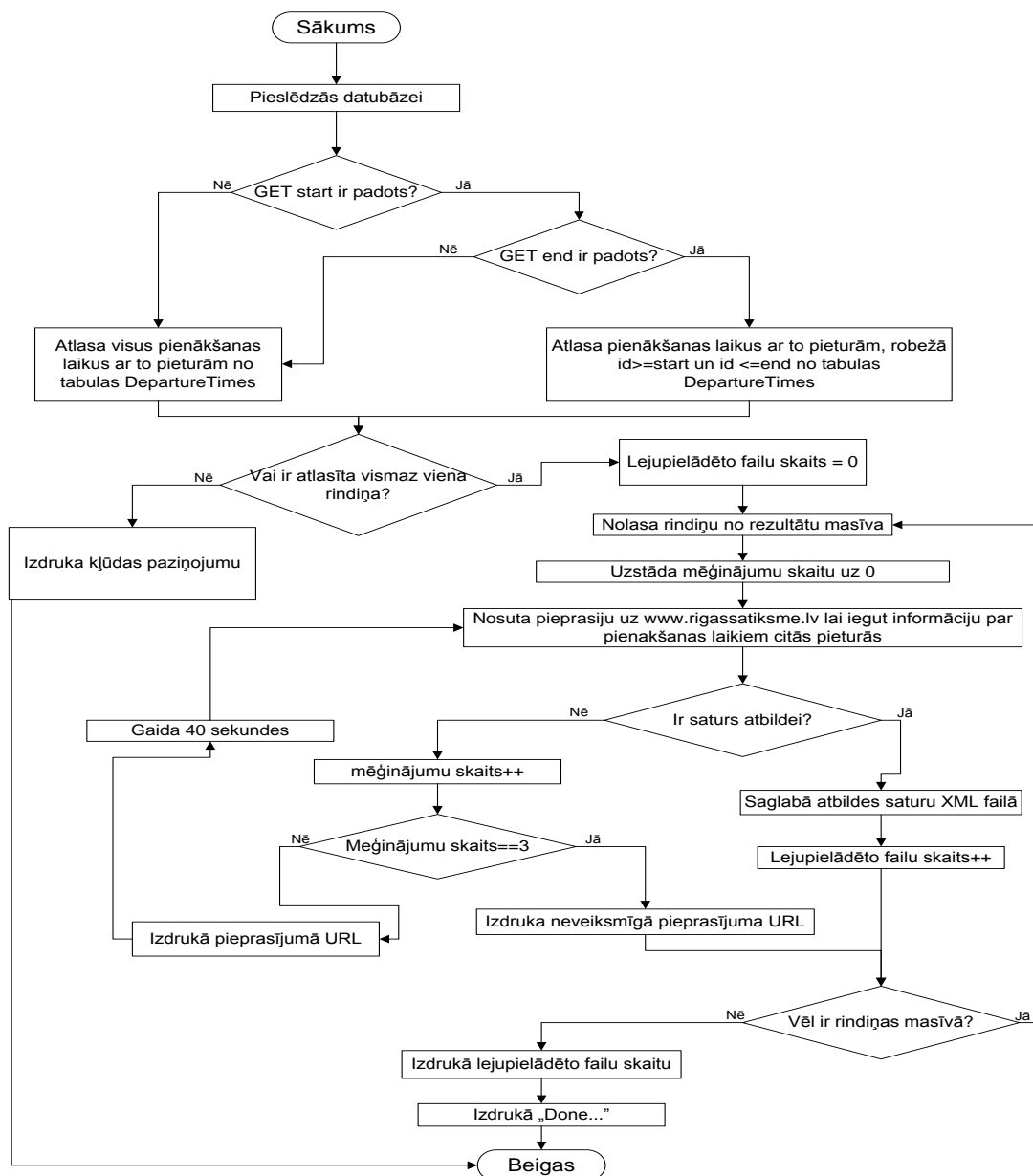


4.7. att. Rīgas Sabiedriskā transporta informācijas ielādes moduļa algoritma blokshēma

#### 4.2.8. RST laiku pieturās ielādes moduļa algoritma projektējums

Modulis lejupielādē informāciju par maršruta pienākšanas laikiem dažādas pieturās atkarībā no atiešanas laikā kāda no pieturām. Šo informāciju var iegūt nosūtot pieprasījumu uz saiti [http://saraksti.rigassatiksme.lv/?a=p.timing&timing\\_id=<timig\\_id>&t=xhtml&l=lv](http://saraksti.rigassatiksme.lv/?a=p.timing&timing_id=<timig_id>&t=xhtml&l=lv). Timing\_id jāņem no datubāzes tabulas DepartureTime – viņš ir piekārtots katram transporta pienākšanas laikam kāda no pieturām.

Modulim var padot GET parametrus formā RSTDT.php?start=<cipars>&end=<cipars>, kur *start* un *end* ir indeksu robežas, tad modulis veiks operāciju tikai tiem pienākšanas laikiem, kas ir šajā robežā. Neko nenoradot, modulis apstrādās visus pienākšanas laikus. Tas ir domāts, ja datu lejuplādēšana ir nepieciešamība veikt uz vairākiem datoriem vai, lai sadalīt vienu procesu uz vairākiem paralēliem procesiem.



4.8. att. RST laiku pieturās ielādes moduļa algoritma blokskhēma

# TESTĒŠANAS DOKUMENTĀCIJA

## 1. Ievads

Šajā dokumentā ir aprakstīti sistēmas „Mobilā lietojumprogrammatūra navigācijai Rīgā” vienibtestēšanas rezultāti. Testēšana notika gan pēc katras funkcijas atsevišķas norealizēšanas, gan kad tika izstrādāta visa sistēma, lai pārbaudīt vai viss strādā atbilstoši sagaidītajam rezultātam.

## 2. Vienibtestēšanas rezultāti

Vienibtestēšanas rezultāti tika apkopoti tabulā ar sekojošiem laukiem:

- Nr. – numurs, kas identificē konkrēto testu;
- Testa apraksts – apraksts testa darbības konkrētajam testējamajam vienumam;
- Sagaidāmais rezultāts – apraksts par to, kas tiek sagaidīt. Ja sagaidāmais rezultāts ietver sevī kādu paziņojumu, tad tiek norādīts tā identifikators(apskatīt paziņojumus var pielikumā sadaļā „Paziņojumi”);
- Statuss – norāda uz to vai testa rezultāts atbilst sagaidītajam, ja jā tad lauka ieraksta +, ja nē, apraksta problēmu un vai problēma bija atrisināta;

Testa apraksta norādes (X) norāda uz to, kādu testu jāizpilda pirms šī testa, un kā šis tests ir turpinājums testa X veiktai darbībai(testa X testa apraksts un dotā testa apraksts). Pēc katra vienuma notestēšanas tiek dots problēmu kopsavilkums, kurā apraksta radušās problēmas testpiemēros, un vai tas tika novērsts.

Testēšanā vienumi tiks pārbaudīti uz parastiem datiem, robežvērtībām, tukšiem laukiem, nekorektiem datiem. Tiks pārbaudīta arī lietotāja saskarne – izskats un funkcionalitāte.

Piezīme: testējot uz viedai tālruņa ar operētai sistēmu Android 2.1. jāņem vērā, ka tas neatbalsta pinch-to-zoom kustību kartei, un šis netiek pārbaudīts(tikai manuāla pietuvinājuma maiņa).

## 2.1. Lietotāja pieslēgšanās sistēmai

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
1.1.	Veic pieprasījumu uz adresi rmap.wtf.lv .	Atvērās sistēmas mājas lapa ,kas uzsākas ar slīdrādi, kurā tiek attēlots OpensStreetMap, Google, tvitera un Rīgas satiksmes logo un sistēmas logo divu sekunžu laika katrs. Atveras Sākulapa.	+
1.2.	Veic pieprasījumu uz adresi rmap.wtf.lv/index.php.		+
1.3.	Veic pieprasījumu uz kādu failu direktija rmap.wtf.lv/dbfunctions/	Parādās logs, kurā prasa ievadīt administratora lietotājvārdu un paroli. Piekļuve pie failiem notiek tikai pēc pareizu datu ievadīšanas.	+

## 2.2. Sākulapa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
2.2.	Atver sadaļu	Sadaļas forma atbilst pielikumā aprakstītaj Sākulapas formai. Parādīts saraksts ar elementiem	+
2.1.	Pieskārās sadaļu saraksta elementiem.	Parādās procesa ielādes logs, un atvērās attiecīgā sadaļa.	+

## 2.3. Ieteikumu sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
3.1.	Atver sadaļu.	Sadaļas forma atbilst pielikumā aprakstītaj Ieteikumu	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
		sadaļas formai. Kartes pietuvinājums ir 15, un centrs uzstādīts uz punktu (56.946285,24.104518).	
3.2.	(3.1.) Samazina pietuvinājumu tik cik ir iespējams.	Pietuvinājums beidz samazināties, pie pietuvinājuma 10.	+
3.3.	(3.2.) Pieskārās brīvi izvēlētajā vietā uz kartes.	Par kartes centru kļūst pieskāriena punkts. Pieskāriena punktā „nokrīt” marķieris.	+
3.4.	(3.3.) Pieskārās ekrānam un velk karti(darg). Atlaiž pirkstu.	Kartē pārvietošanas virzienā un paliek tajā stāvokli, kāda tā bija līdz pirksta atlaišanai no ekrāna.	+
3.5.	(3.4.) Pieskārās ekrānam ar diviem pirkstiem un velk tos kopā vai izplēš	Ja pirkstus savelk kopā, kartes pietuvinājums samazinās, ja izplēš – palielinās.	+
3.6.	Norāda objekta atrašanas vietu, un nospiež pogu [Tālāk].	Parādās teksta ievades lauks. Poga [Tālāk] paliek par neaktīvu.	+
3.7.	Nenorāda objekta atrašanas vietu, un nospiež pogu [Tālāk].		+
3.8.	(3.6.) Ievada ieteikuma tekstu (robežās no 1 līdz 500 simboli) teksta laukā. Nospiež pogu [Tālāk].	Poga [Tālāk] kļūst par aktīvu. Pēc tās nospiešanas, parādās procesa apstrādes bilde. Parādās paziņojums MSG_THX, un atvērās sākumlapa.	+
3.9.	(3.7.) Ievada tekstu (robežās no 1 līdz 500 simboli) teksta laukā. Nospiež pogu [Atpakaļ]. Norāda	Pēc pirmās pogas [Tālāk] nospiešanas parādās ieteikuma ievades teksta lauks. Poga [Tālāk] paliek	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
	objekta atrašanas vietu. Nospiež pogu [Tālāk] divas reizes.	neaktīva. Pēc teksta ievadīšanas pogā [Tālāk] paliek neaktīva. Pēc objekta norādīšanas un divas reizes nospiežot pogu [Tālāk], parādās procesa apstrādes bilde. Parādās paziņojums MSG_THX, un atvērās Sākulapa.	
3.9.	(3.6.) Ieteikuma teksta laukā ievada 10 tukšumus	Poga [Tālāk] paliek neaktīva.	+
3.10.	Ieteikuma teksta laukā ievada 501 simbolu.	501 simbols netiek ievadīts.	+
3.11.	Ieteikuma teksta laukā mēģina iekopēt 1000 simbolu garu virkni.	Teksta laukā paliek tikai 500 simboli.	+
3.12.	Norāda objekta atrašanas vietu, bet nenorāda ieteikuma tekstu. FireBug JavaScript konsolē palaiž kodu: document. getElementById("Send_Button") .className= 'next_button enabled_button'. Nospiež pogu [Tālāk].	Parāda paziņojumu ERR_TXT500 un turpina rādīt ieteikuma sadaļu.	+
3.13.	Nenorāda objekta atrašanas vietu, un nenorāda ieteikuma tekstu. FireBug JavaScript konsolē palaiž kodu: document.	Parāda paziņojumu ERR_UNDEF un turpina rādīt ieteikuma sadaļu.	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
	getElementById("Send_Button") .className= 'next_button enabled_button'. Nospiež pogu [Tālāk].		
3.14.	FireBug JavaScript konsolē palaiž kodu: HTTPPostRequest ('index.php', 'controler=advices&method=SaveAdvice&lat=we& lng=qwe&adviceText=aass' );	Atgriež paziņojuma kodu ERR_CORD.	+
3.15.	FireBug JavaScript konsolē palaiž kodu: HTTPPostRequest ('index.php', 'controler=advices&method=SaveAdvice&adviceText=as' );	Atgriež paziņojuma kodu ERR_NCORD.	+
3.16.	FireBug JavaScript konsolē palaiž kodu: HTTPPostRequest ('index.php', 'controler=advices&method=SaveAdvice&lat=2&lng=23' );	Atgriež paziņojuma kodu ERR_NADTXT.	+
3.17.	FireBug JavaScript konsolē palaiž kodu: HTTPPostRequest ('index.php', 'controler=advices&method=SaveAdvice&lat=1218 .66&lng=6789.56&adviceText=as' );	Atgriež paziņojuma kodu ERR_CORD.	+
3.18.	FireBug JavaScript konsolē palaiž kodu: HTTPPostRequest ('index.php',	Atgriež paziņojuma kodu MSG_THX.	+



Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
	'controler=advices&method=SaveAdvice&lat=-180.00&lng=-90.00&adviceText=asasas' );		
3.18.	Nospiež pogu [Atpakaļ] kad tiek rādīta karte	Atgriežas uz sākumlapu.	+
3.19.	Norāda objekta atrašanās vietu, ieteikuma tekstu. Izslēdz interneta pieslēgumu. Nospiež pogu [Tālāk], lai nosūtīt datus.	Parādās procesa apstrādes bilde, un 14 sekundē pēc pieprasījuma nosūtīšanas parādās paziņojums ERR_TIME14. Procesā apstrādes bilde pazūd, un turpina rādīt sadaļu tādā pašā stāvoklī kā pirms datu nosūtīšanas	+

## 2.4. Kategoriju izvēles sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
4.1.	Atver sadaļu.	Rāda <b>procesa apstrādes bildi</b> . Kad procesa apstrādes bilde pazūd parāda sadaļu. Sadaļas forma atbilst pielikumā aprakstītajā Kategoriju izvēles sadaļas formai. Zem galvenes saraksts ar objektu kategorijām (pavisam 13).	+
4.2.	Izslēdz interneta pieslēgumu un mēģina atvērt sadaļu.	Parādās procesa apstrādes bilde, un 14 sekundē pēc pieprasījuma nosūtīšanas parādās paziņojums ERR_TIME14. Procesā apstrādes bilde pazūd un lietotājam rāda sākumlapu.	+
4.3.	(4.1.) Pieskārās objektu kategorijai sarakstā.	Parāda procesa apstrādes bildi, kad apstrāde ir	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
		pabeigta, parāda objektu atrašanas vietu sadaļu.( kas tiek attēlots netiek apskatīts šajā testā)	
4.4.	(4.1.)Nospiež pogu [Atpakaļ].	Atvērās sadaļa Sākumlapa.	+

## 2.5. „Objekti apkārt man” sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
5.1.	Izvēlās kādu no objektu kategorijām Kategoriju izvēles sadaļa.	Rāda procesa apstrādes bildi.	+
5.2.	(5.1.)Neapstiprina atrašanas vietas koordināšu nodošanu sistēmai.	Lietotāj marķieris tiek novietots pozīcija ('56.946238','24.104905'). Sadaļas forma atbilst pielikumā aprakstītāj „Objekti apkārt man” sadaļas formai. Kartes pietuvinājums ir 16. Kartes centrā atrodas lietotāja marķieris. Uz kartes tiek attēloti objektu marķieri, ja objekts ir 500x500m apgabalā(noklusētais meklēšanas apgabals).	+
5.3.	Izvēlās kādu no objektu kategorijām Kategoriju izvēles sadaļa. Apstiprina atrašanas vietas koordināšu nodošanu sistēmai. Atrodas ārpus Rīgas.	Parāda paziņojumu MSG_NOINR. Lietotāj marķieris tiek novietots pozīcija mobila tālruņa atrašanas pozīcija. Sadaļas forma atbilst pielikumā aprakstītāj „Objekti apkārt man” sadaļas formai. Kartes pietuvinājums ir 16. Kartes centrā atrodas	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
		lietotāja marķieris. Lietotājs var brīvi pārvietoties visā pasaules kartē.	
5.4.	Atrašanas vieta – Rīga, Zeļļu iela 8. Meklē objektus „Ātrā ēdināšana”. Apstiprina atrašanas vietas koordināšu nodošanu sistēmai.	Lietotāj marķieris tiek novietots pozīcija mobila tālruņa atrašanas pozīcija. Sadaļas forma atbilst pielikumā aprakstītāj „Objekti apkārt man” sadaļas formai. Kartes pietuvinājums ir 16. Parāda paziņojumu MSG_NOINR(neviens objekts netiek atrasts pat 3x3km apgabalā).	+
5.5.	(5.4.) Samazina pietuvinājumu tik cik ir iespējams.	Pietuvinājums beidz samazināties, pie pietuvinājuma 10.	+
5.6.	(5.5.) Pieskārās brīvi izvēlētāja vietā Vecrīgā uz kartes.	Par kartes centru kļūst pieskāriena punkts.	+
5.7.	(5.5.) Pieskārās kartei, kur parādīta pilsēta Jelgava.	Parādās paziņojums MSG_OUTRIGA un nekas nemainās.	+
5.8.	(5.7.) Pieskārās ekrānam un velk karti(darg). Atlaiž pirkstu.	Kartē pārvietojas vilkšanas virzienā un paliek tajā stāvokli, kāda tā bija līdz pirksta atlaišanai no ekrāna. Nesanāk pārvietoties ārpus Rīgas apgabala.	+
5.9.	(5.8.) Pieskārās ekrānam ar diviem pirkstiem un velk tos kopā vai izplēš	Ja pirkstus savelk kopā, kartes pietuvinājums samazinās, ja izplēš – palielinās.	+
5.10.	(5.9.) Novieto divus pirkstus uz punktiem, kas uz	Karte pietuvinās, bet tas centrs neaizbīdās tālāk par	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
	kartes apzīmē pilsētas Jelgava un Valgunde, un savēl pirkstus kopā.	Rīgas apgabala robežu.	
5.11.	(5.4.) Nomaina meklēšanas apgabalu uz „Rādīt visus”	Parāda procesa apstrādes bildi. Pēc procesa apstrādes bildes pazušanas, skats uz karti paliek tāds pats, ka pirms meklēšanas apgabala nomainīšanas, un parādās visi objekti ar tipu „Ātrā ēdināšana”.	+
5.12.	(5.11.) Palielina pietuvinājumu divās reizes(līdz 18).	Virs objektu marķieriem parādās iezīmes ar objektu nosaukumu. Virs objektiem, kuriem sistēmā nav informācijas par tā nosaukumu, nekas neparādās.	+
5.13.	(5.12.)Pieskārās objektam, virs kura ir iezīme.	Atvērās informācijas logs, kurā ir parādīta informācija par objektu. Par kartes centru kļūst objekts kurām pieskārās.	+
5.14.	(5.13.)Pieskārās brīvi izvēlētajā vietā, kartē.	Atvērtais informācijas logs aizvērās.	+
5.15.	(5.14.) Nomaina meklēšanas apgabalu uz 200x200m	Parāda procesa apstrādes bilde. Pēc procesa apstrādes bildes pazušanas, skats uz karti paliek tāds pats, ka pirms meklēšanas apgabala nomainīšanas. Uz kartes nav objektu marķieru un iezīmju.	+
5.16.	(5.15.) Maina kartes pietuvinājumu no 18 līdz 20	Virs redzamiem objektiem ir redzamas iezīmes visos pietuvinājumos.	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
5.17.	(5.16.) Maina karte pietuvinājumu no 18 līdz 16	Virs objektiem pazūd iezīmes.	+
5.18.	(5.17.) Maina kartes pietuvinājumu no 16 uz 15	Redzami objekti tiek apvienoti grupās. Pazūd objektu marķieri. Parādā marķieri kas atspoguļo objektu grupas. Tur kur objekti ir 1-2 grupā, marķieris ir zils, 3-4 – zaļš, 5-10 – dzeltens, vairāk par 10 – sarkans. Grupu marķieri novietojās vidējās koordinātes no visu marķieru koordināšu summas.	Marķieri nepareizi apvienojas grupās, un objektu skaits grupās atšķiras. Salabots.
5.19.	(5.18.) Maina kartes pietuvinājumu no 15 līdz 10	Objektu grupas paliek redzamas visos līmeņos. Objektu kopējais skaits visās grupās nemainās.	+
5.20.	Pieskārās marķierim, kas apzīmē objektu grupu.	Par kartes centru kļūst objektu grupa kurai pieskārās.	+
5.21.	(5.20.) Maina kartes pietuvinājumu no 10 uz 16	Objektu grupas pazūd. Uz kartes ir redzami atsevišķi marķieri kas apzīmē konkrētos objektus.	+
5.22.	Atrašanas vieta – Rīga, Gaujas iela 8. Meklē objektus „Sabiedriska transporta pietura”. Apstiprina atrašanas vietas koordināšu nodošanu sistēmai.	Parādās procesa apstrādes bildi. Pēc procesa apstrādes bildes pazušanas lietotāj marķieris tiek novietots pozīcija mobila tālruņa atrašanas pozīcija. Sadaļas forma atbilst pielikumā aprakstītaj „Objekti apkārt man” sadaļas formai. Kartes pietuvinājums ir 16. Uz kartes ir parādīta pietura „Aizsaules iela”	+
5.23.	(5.22.) Pieskārās pieturai	Parāda procesa apstrādes bildi, kad apstrāde ir pabeigta, parāda Dienas izvēles sadaļu.( kas tiek	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
		attēlots netiek apskatīts šajā testā)	
5.24.	Nospiež pogu [Atpakaļ]	Atvērās Kategoriju izvēles sadaļa.	+

## 2.6. Dienas izvēles sadaļa

Pirms veikt šos testus, vajag atjaunot datubāzi, lai tā atbilstu Rīgas satiksmes sniegtiem datiem, kas atvieglos rezultātu salīdzināšanu un noskaidrošanu vai tie ir pareizi. Šajos testos nav aprakstīts kādas tieši pieturas, laiki, maršruti tiek izvēlēti, bet testētais pats brīvi veic izvēli, un salīdzina sistēmas sniegtos datus, ar to, ko piedāvā Rīgas satiksme.

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
6.1.	Atver sadaļu.	Rāda procesa apstrādes bildi. Kad procesa apstrādes bilde pazūd parāda sadaļu. Sadaļas forma atbilst pielikumā aprakstītaj Dienas izvēles sadaļas formai.	+
6.2.	(6.1) Izvēlas kādu no dienas tipiem no saraksta.	Parāda procesa apstrādes bildi, kad apstrāde ir pabeigta, parāda Pieturas saraksta sadaļu.( kas tiek attēlots netiek apskatīts šajā testā)	+
6.3.	(6.1.) Nospiež pogu [Atpakaļ].	Atvērās sadaļa „Objekti apkārt man”, kas ir tādā pašā stāvoklī, ka pirms pieturas izvēles.	+

## 2.7. Pieturas saraksta sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
7.1.	Atver sadaļu.	Rāda procesa apstrādes bildi. Kad procesa apstrādes	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
		bilde pazūd parāda sadaļu. Sadaļas forma atbilst pielikumā aprakstītāj Pieturas saraksta sadaļas formai. Tiek parādīti maršruti pieturās. To transportu veidu sarakstus, kuru šajā pietura nav, nerada.	
7.2.	(7.1) Izvēlas maršrutu no saraksta.	Parāda procesa apstrādes bildi, kad apstrāde ir pabeigta, parāda Maršruta pienākšanas laiku sadaļu.( kas tiek attēlots netiek apskatīts šajā testā)	+
7.3.	(7.1.) Nospiež pogu [Atpakaļ].	Atvērās Dienas izvēles sadaļa.	+
7.4.	Sadaļa „Objekti apkārt man” izvēlas pieturu Rātslaukums. Dienas izvēles sadaļā izvēlās darbadienu.	Parāda procesa apstrādes bildi. Parāda paziņojumu ERR_NTF un atgriež uz dienas izvēles sadaļu.	+

## 2.8. Maršruta pienākšanas laiku sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
8.1.	Atver sadaļu.	Rāda procesa apstrādes bildei. Kad procesa apstrādes bilde pazūd parāda sadaļu. Sadaļas forma atbilst pielikumā aprakstītāj Pieturas maršruta pienākšanas laiku sadaļas formai. Sarakstā ir attēloti transporta pienākšanas laiki pieturā.	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
8.2.	(8.1) Izvēlas laiku no saraksta.	Parāda procesa apstrādes bildi. Kad apstrāde ir pabeigta, parāda Maršruta laiku pieturās sadaļu. (kas tiek attēlots netiek apskatīts šajā testā)	+
8.3.	(8.1.) Nospiež pogu [Atpakaļ].	Atvērās Pieturas maršrutu sadaļa.	+

## 2.9. Maršruta laiku pieturās sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
9.1.	Atver sadaļu.	Rāda procesa apstrādes bildi. Kad procesa apstrādes bilde pazūd parāda sadaļu. Sadaļas forma atbilst pielikumā aprakstītaj Maršruta laiku pieturās sadaļas formai.. Sarakstā tiek parādīts pienākšanas laiks izvēlētajā pieturā, un visi tie pienākšanas laiki pieturās, kas maršrutā ir pēc dotās.	+
9.2.	(9.1) Nospiež pogu [Rādīt kartē].	Parāda procesa apstrādes bildi, kad apstrāde ir pabeigta, parāda Maršruta uz kartes sadaļu.(kas tiek attēlots netiek apskatīts šajā testā)	+
9.3.	(9.1.) Nospiež pogu [Atpakaļ].	Atvērās Maršruta pienākšanas laiku sadaļa.	+



## 2.10. Maršruta uz kartes sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
9.1.	Atver sadaļu.	<p>Rāda procesa apstrādes bildi. Kad procesa apstrādes bilde pazūd parāda sadaļu. Sadaļas forma atbilst pielikumā aprakstītaj Maršruta uz kartes sadaļas formai.</p> <p>Kartes attālinājums un izvietojums ir tāds, ka ekrāna var redzēt visas maršruta pieturas un lietotāja atrašanas vietas marķieri.</p>	+
9.2.	(9.1.) Pieskārās brīvi izvēlētajā vietā uz kartes.	Par kartes centru kļūst pieskāriena punkts.	+
9.3.	(9.2.) Pieskārās ekrānam un velk karti(darg). Atlaiž pirkstu.	Kartē pārvietojas vilkšanas virzienā un paliek tajā stāvokli, kāda tā bija līdz pirksta atlaišanai no ekrāna.	+
9.4.	(9.3.) Pieskārās ekrānam ar diviem pirkstiem un velk tos kopā vai izplēš	Ja pirkstus savelk kopā, kartes pietuvinājums samazinās, ja izplēš – palielinās.	+
9.5.	(9.4.) Uztāda pietuvinājumu uz 13.	Virs pieturu marķieriem parādās iezīmes ar pienākšanas laiku pieturā un pieturas nosaukumu.	+
9.6.	(9.5.) Maina pietuvinājumu no 13 līdz 20.	Visos pietuvinājumos iezīmes virs pieturām paliek redzamas.	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
9.7.	(9.5.) Maina pietuvinājumu no 13 uz 12 un mazāk.	Nomainot pietuvinājumu no 13 uz 12, iezīmes virs pieturām pazūd. Visos mazākos pietuvinājumos iezīmes neredz.	+
9.8.	(9.7.) Pieskārās kādam no pieturas marķierim.	Atvērās informācijas logs, kurā ir parādīta informācija par pieturu- pienākšanas laiks, nosaukums, iela(ja ir). Par kartes centru kļūst pieturas marķieris kurām pieskārās.	+
9.9.	(9.8.)Pieskārās brīvi izvēlētajā vietā, kartē.	Atvērtais informācijas logs aizvērās.	
9.8.	(9.1.) Nospiež pogu [Atpakaļ].	Atvērās Maršruta pienākšanas laiku sadaļa.	+

## 2.11. ĢeoTvitera sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
11.1.	Atver sadaļu.	Rāda procesa apstrādes bildi. Kad procesa apstrādes bilde pazūd parāda sadaļu. Sadaļas forma atbilst pielikumā aprakstītaj ĢeoTvitera sadaļas formai. Ja lietotājs ir autorizējies sistēmā, viņam rāda iespēju atstāt komentāru „skatīties savus, draugu un visu lietotāju atstātos komentārus, un atteikšanas iespēju. Ja lietotājs nav autorizējies sistēmā, viņam piedāvā autorizēties ar tviteri vai skatīties visu lietotāju	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
		atstātos komentārus.	
11.2.	(11.1.) Izvēlās kādu no piedāvātajām darbībām.	Parāda procesa apstrādes bildi un pāriet pie izvēlētajām darbībām.	+
11.3.	(11.1.) Nospiež pogu [Atpakaļ].	Atvērās Sākulapa.	+
11.4.	(11.1.) Lietotājs ir autorizējies sistēmā. Nospiež atteikšanas pogu.	Parāda procesa apstrādes bilde. Kad bilde pazūd, rāda sadaļu, kā neautorizētām lietotājām.	+
11.5.	(11.1.) Neautorizēts lietotājs izvēlās autorizēties ar tviteri.	Lietotājs tiek pārdresēts uz tvitera autorizāciju.	+
11.6.	(11.5.) Lietotājs veiksmīgi autorizējas tviterī.	Lietotājs tiek atgriezts uz sistēmas ĢeoTvitera sadaļu, un viņam rāda sadaļu kā autorizētam lietotājam.	+
11.7.	(11.5.) Lietotājs atsākās autorizēties un sniegt savu informāciju.	Lietotājs tiek atgriezts uz sistēmas pirmo sākulapu.	+

## 2.12. Komentāru atstāšanas sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
12.1.	Atver sadaļu.	Sadaļas forma atbilst pielikumā aprakstītajam Komentāru atstāšanas sadaļas formai. Kartes pietuvinājums ir 15 un centrs uzstādīts mobilas	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
		ierīces atrašanas vietu.	
12.2.	(12.1.) Samazina pietuvinājumu tik cik ir iespējams. Pietuvina tik cik ir iespējams. Mēģina vilkt karti vai nospiež kādā brīvi izvēlētajā vietā uz kartes.	Pietuvinājums beidz samazināties, pie pietuvinājuma 10, bet beidz pietuvināties pie pietuvinājuma 20. Velkot vai pieskaroties kartei, karte nekustās.	+
12.3.	(12.2.) Nospiež pogu [Tālāk].	Atveras komentāra teksta ievadīšanas lauks. Poga [Tālāk] kļūst par neaktīvu.	+
12.4.	(12.3.) Ievada komentāra tekstu (robežās no 1 līdz 140 simboli) teksta laukā. Nospiež pogu [Tālāk].	Poga [Tālāk] kļūst par aktīvu. Pēc tās nospiešanas, parādās procesa apstrādes bilde. Parādās paziņojums MSG_TWP, un atvērās ĢeoTvitera sadaļa.	+
12.5.	(12.3.) Nospiež pogu [Atpakaļ]. Nospiež pogu tālāk. Ievada teksta laukā „hi”, Nospiež pogu tālāk.	Pēc pogas [Atpakaļ] nospiešanas, virs kartes pazūd teksta lauks. Nospiežot pogu [Tālāk] – teksta lauks parādās, bet poga [Tālāk] kļūst par neaktīvu. Pēc teksta ievadīšanas, pogā [Tālāk] kļūst aktīva, un pēc pogas [Tālāk] nospiešanas, parādās procesa apstrādes bilde. Parādās paziņojums MSG_TWP, un atvērās ĢeoTvitera sadaļa.	+
12.6.	(12.3.) Komentāra teksta laukā ievada 10 tukšumus	Poga [Tālāk] paliek neaktīva.	+
12.7.	(12.4.) Komentāra teksta laukā ievada 141 simbolu vai iekopē 1000 simbolus.	Teksta laukā paliek tikai 140 simboli	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
12.8.	FireBug JavaScript konsolē palaiž kodu: HTTPPostRequest ('index.php', 'controler=user&method=PostTweet')	Atgriež paziņojuma kodu ERR_UNDEF.	+
12.9.	FireBug JavaScript konsolē palaiž kodu: HTTPPostRequest ('index.php', 'controler=user&method=PostTweet&lat=we&lng=qwe&tweet=aass' );	Atgriež paziņojuma kodu ERR_CORD ja ir autentificējies, bet ERR_UNDEF ja nav .	+
12.10.	Nospiež pogu [Atpakaļ] kad tiek rādīta karte	Atgriežas uz ĢeoTviteri.	+
12.11.	Izslēdz internetu, ĢeoTvitera sadaļa izvēlās komentāra atstāšanas darbību.	Parādās procesa apstrādes bilde, un 14 sekundē pēc pieprasījuma nosūtīšanas parādās paziņojums ERR_TIME14. Procesā apstrādes bilde pazūd, un turpina rādīt Komentāru atstāšanas sadaļu.	+

### 2.13. Komentāru attēlošanas sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
13.1.	Atver sadaļu izvēloties, kad no kategorijām ĢeoTvitera sadaļām.	Rāda procesa apstrādes bildi. Kad procesa apstrādes bilde pazūd parāda sadaļu. Sadaļas forma atbilst pielikumā aprakstītaj Komentāru attēlošanas sadaļas formai. Uz kartes tiek parādīti pēdējie 10 komentāri izvēlētajā kategorijā. Kartes centrā ir jaunākais	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
		komentārs.	
13.2.	(13.1.) Pieskārās brīvi izvēlētāja vietā uz kartes.	Par kartes centru kļūst pieskāriena punkts.	+
13.3.	(13.2.) Pieskārās ekrānam un velk karti(darg). Atlaiž pirkstu.	Kartē pārvietojas vilkšanas virzienā un paliek tajā stāvokli, kāda tā bija līdz pirksta atlaišanai no ekrāna.	+
13.4.	(13.3.) Pieskārās ekrānam ar diviem pirkstiem un velk tos kopā vai izplēš	Ja pirkstus savelk kopā, kartes pietuvinājums samazinās, ja izplēš – palielinās. Maksimālais pietuvinājums 20, minimālais 10.	+
13.5.	(13.4.) Pieskārās kādam no komentāru marķierim.	Atvērās informācijas logs, kurā ir parādīta informācija par komentāru. Par kartes centru kļūst komentāra marķieris kurām pieskārās.	+
13.6.	(13.5.)Pieskārās brīvi izvēlētāja vietā, kartē.	Atvērtais informācijas logs aizvērās.	+
13.7.	(13.6.) Nomaina attēlojamo komentāru skaitu.	Uz kartes tiek radīts vismaz viens marķieris un nekairāk par izvēlēto vērtību. Par kartes centru kļūst jaunākais komentārs un tā informācijas logs tiek parādīts	Informācijas logā lietotāja vārds attēlojas, ka „undefined”. Salabots.
13.7.	Ar pogu [Nākamais] un [Iepriekšējais] palīdzību, pārslēdzas starp komentāriem.	Karte pārvietojas pie nākama/iepriekšēja komentāra atkarība no publicēšanas laika. Kad pariet pie kāda komentāra tad tā informācijas logs atveras, bet iepriekšēja komentāra informācijas logs aizvērās.	+

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
13.8.	Lietotājs, kurš nepublicēja komentārus izvēlas skatīties savus komentārus. Lietotājs kurām nav draugu izvēlās skatīties savus draugus. Jebkurš lietotājs grib apskatīt visus komentārus, bet neviens komentārs nav publicēts.	Parāda paziņojumu ERR_NFTF, un atgriež uz GeoTvitera sadaļu.	+
13.9.	(13.1.) Nospiež pogu [Atpakaļ].	Atvērās GeoTvitera sadaļa.	+

## 2.14. Valodas maiņa un valodas maiņas sadaļa

Nr.	Testa apraksts	Sagaidāmais rezultāts	Statuss
14.1.	Atver sadaļu.	Rāda procesa apstrādes bildi. Kad procesa apstrādes bilde pazūd parāda sadaļu. Sadaļas forma atbilst pielikumā aprakstītaj Valodas maiņas sadaļas formai.	+
14.2.	(14.1) Izvēlas kādu valodu no saraksta.	Parāda procesa apstrādes bildi, kad apstrāde ir pabeigta, parāda sākumlapu. Saskarnes valoda ir nomainīta uz izvēlēto.	+

## PROJEKTA ORGANIZĀCIJA

Projekta organizācija tika izpildīta pēc inkrementāla dzīves cikla modeļa. Tika izvirzītas pirmās idejas par to, ko produkts varētu darīt. Pēc īdēju apkopošanas, tika izvēlētas svarīgākas, un tādas, kuras var nerealizēt atvēlētajā laikā. No šīm īdējām bija izvirzītas prasības, kuras tika apkopotas „Programmatūras prasību specifikācija”. Pēc prasību apkopošanas, tika uzrakstīts sistēmas projektējums, pēc kura tika būvēta sistēma. Izstrāde bija uzsākta ar moduļu izstrādi, kas atbild par informācijas ievākšanu sistēmā. Kad sistēma bija aizpildīta ar nepieciešamiem datiem tika izstrādātas pirmās pamat funkcijas. Pēc funkciju nerealizēšanas tika pārbaudīta sistēmas darbība, kuras rezultātā varēja redzēt sistēmas uzvedību uz mobilā tālruņa. Tika izstrādāts jauns(uzlabots) „Programmatūras projektējuma apraksts”, veiktas izmaiņas sistēmas uzbūve, lai optimizēt sistēmas ātrdarbību, kas iekļāva sevī kodā optimizēšanu, iebūvēto funkciju veidošanu, dažu moduļu pārņemšanu uz serveri.

Izstrādātajai sistēmai pasūtītāja nav, tas ir paša izdomātā lietojumprogrammatūra, tāpēc pats izvirzīju prasības sistēmai, kuras apkopāju programmatūras prasību specifikācijas un pēc šī dokumenta izstrādāju arī programmatūras projektējuma aprakstu. Pēc šiem dokumentiem tika uzprogrammēta sistēma un notestēta.

Šo projektu izstrādāju patstāvīgi, bet idejas un ieteikumus guvu no sava kvalifikācijas darba vadītāja, ka arī viņa kolēģiem.



## KVALITĀTE NODROŠINĀŠANA

Lai nodrošināt izstrādāta projekta kvalitāti, tika veiktas sekojošas darbības:

1. Programmatūras dokumentācija tika veikta saskaņā ar valsts standartiem:
  - LVS 68:1996 *Programmatūras prasību specifikācijas ceļvedis*,
  - LVS 72:1996 *Ieteicamā prakse programmatūras projektējuma aprakstīšanai*.
2. Visām programmatūras daļām tika veikta vienībtestēšana, lai pārliecinātos par sistēmas pareizu darbību, un testu rezultāti tika pierakstīti testēšanas žurnālā. Daļas, kur bijā konstatētas kļūdas, tika veiktas korekcijas kodā, un atkārtotā testēšana.
3. Programmatūras daļām, kas apstrādāja lielus datu apjomus iesaistot tajā lietotāja izmantoto ierīci, tika mērīta ātrdarbība un veikta koda optimizēšana, lai sasniegt labāko rezultātu. Šajos mērījumos tika izmantoti vairāki rīki tādi ka – Fire Bug, Google Chrome izstrādātāju iebūvētais rīks.
4. Izstrādātājs kods tika būvēts noteiktā struktūrā, lai līdzīgo moduļu starpā, tas būtu strukturēts un noformēts vienādi, ka arī līdzīgu aprēķinu, uzdevumu veikšana moduļos, būtu noformēta līdzīgi, kas uzlaboja koda saprotamību.
5. Kodu daļas, kas ir līdzīgas un atkārtojas vairākos moduļos, tika iznestās atsevišķas funkcijās, kas atviegloja sistēmas būvēšanu un modificēšanu.
6. Sistēma tika būvēta tā, lai to varētu viegli modificēt, pielikt jaunus moduļus klāt, vai uzlabot jau esošos.
7. Sistēmas uzturēšanas funkciju piekļūšanai tika uzlikta parole, lai tas izmantot varētu tikai par sistēmu atbildīga persona.
8. Lietotāja saskarne veidotā vienotā, nepārprotamā stilā, intuitīvi viegli uztverama. Saskarne ir būvēta tā, lai būtu viegli strādāt ar sistēmu tieši uz mobilā tālruņa.

## KONFIGURĀCIJU PARVALDĪBA

Tā kā programmprodukta izstrādē piedalījās tikai viens programmētājs, speciāli konfigurācijas pārvaldības rīki netika izmantoti, bet tika izmantots tikai versiju vadības rīks Mercurial. Izmantojot šo rīku, varēja iegūt sistēmas izmaiņu vēsturi, un sistēmas koda sabojāšanas rezultāta, atgriezties pie kādas no iepriekšējam versijām. Sistēmas izstrāde notika uz izstrādātāja datora. Kad bija izveidots kāds no moduļiem, lai pārbaudīt tā darbību uz mobilā tālruņa, sistēmas izmaiņas tika saglabātas uz servera repozitorijā izmantojot Mercurial. Papildus pašas sistēmas koda saglabāšanai uz servera, bija divas datubāzes, viena uz izstrādātāja datora, otra uz servera, ka arī uz papildus datu nesējiem tika glabātas noeksportētais datubāzes saturs.

Sistēmas versijas numurs veidots no trim cipariem x.y.z , kur x norāda uz lietojumprogrammatūras paaudzi, y – lielām izmaiņām sistēmas uzbūvē un izmaiņām tās prasībās pret to, z - nelielām izmaiņām programmas kodā.

Pašreizējā sistēmas versija ir 1.1.0.

## DARBIETILPĪBAS NOVĒRTĒJUMS

Viens no veidiem ka tika veikta darbietilpības novērtēšana bija izmantojot COCOMO 81 Intermediate Model Implementation.[4.]

Metodes formula:  $E = a_i (LOC/1000)^{b_i} * EAF$ , kur

**E** – Programmas izstrādei nepieciešamie personmēneši

**LOC** = nepieskaņoto funkcijpunktu skaits \* vidējs programmrindīņu skaits viena funkcijpunkta realizācijai.

**EAF** (effort adjustment factor) – aprēķināts pēc formulas  $EAF = E_1 * E_2 * \dots * E_{15}$ .

**a<sub>i</sub>**, **b<sub>i</sub>** – koeficienti, kas ir atkarīgi no projekta tipa.

	Sarežģītī	Vidēji	Vienkārši	Kopā
Ievadi	6*0	4*2	3*0	8
Izvadi	7*1	5*4	4*8	52
Iekšējie datu faili	15*0	10*0	7*9	63
Ārējas saskarnes	10*1	7*2	5*2	24
Vaicājumi	6*0	4*1	3*8	28
Nepieskaņotu funkcijpunktu skaits:				175

Vidējais rindīņu skaits funkcijai JavaSriptam 54. Par php nav datu ,bet dažos avotos bija minēts, ka 53. Tad prognozējamais rindīņu skaits kodam(LOC) būs  $53*175=9275$ .

Parametrs	Vērtība	
Produkta atribūti		
Drošums	Vidēja	1
Datubāzes izmēs	Zema	0.94
Produkta sarežģītība	Zema	0.85
Datoru atribūti		
Izpildes laika ierobežojumi	Vidēja	1
Atmiņas ierobežojumi	Vidēja	1
Vides mainīgums	Zema	0.87
Datora atbildes laiks	Vidēja	1
Personāla atribūti		
Analītiskās spējas	Vidēja	1
Pieredze aplikāciju izstrādē	Vidēja	1
Programmētāju spējas	Augsta	0.86
Izstrādes vides pieredze	Vidēja	1
Programmēšanas valodas pieredze	Vidēja	1
Projekta atribūti		
Programmatūras izstrādes rīku izmantošana	Vidēja	1
Izstrādes rīku izmantošana	Augsta	0.91
Izstrādes grafika uzliktie ierobežojumi	Vidēja	1
Reizinājums ( EAF ):		0.544

Projekta tips ir „Organic”, jo izstrādāju projektu viens pats. No tā izriet, ka  $a_i = 3.2$ , bet  $b_i = 1.05$ .

Risinājums:  $E = 3.2 * (9275/1000)^{(1.05)} * 0.544 = 18.04$  personmēneši.[5.]

Papildu COCOMO aprēķinam veicu arī savu aptuveno aprēķinu balstoties uz savām spējām un pieredzi.

Pirmkārt, iepriekš aprēķinātais rindiņu skaits kodā, bijā stipri lielāks par to, ko spētu izstrādāt trijos personmēnešos. Rindiņu skaitu aprēķināšanai ņemšu moduļu skaitu sistēmā – 36 moduļi. Plānojot PPA biju sadalījis moduļus tā, ka katra no viņiem rindiņu skaits vidēji varētu būt ap 100, aptuveni tik daudz parasti koda rindiņu man vajag, lai uzrakstīt kāda problēmas risinājumu. Sanāk, ka aptuveni paredzētais koda rindiņu skaits ir 3600. Ja ieliek šo skaitli iepriekš iegūtajā COCOMO formulā iegūst –  $E = 6.68$  personmēneši. Izmantojot šos pašus skaitļus formulā COCOMO 81 Intermediate Model Implementation iegūst 5.52 personmēnešus.[4.]

## IZMANTOTĀS TEHNOLOĢIJAS

Sistēmas izstrādē tika izmantotas sekojošas tehnoloģijas:

- Programmēšanas valoda PHP;
- Programmēšanas valoda JavaScript;
- AJAX;
- CSS 3;
- Datubāze MySql 5.1.36;
- Wampserver 2.0;
- Izstrādes vide NetBeans IDE 6.9.1;
- Teksta redaktors Notepad++;
- Datubāzes pārvaldības rīks phpMyAdmin;
- Datubāzes eksportēšanas un rezerves kopiju veidošanas rīks MySQL Administrator 1.1.9;
- Rīks - Fire Bug;
- Google Chrome izstrādātāju iebūvētais rīks;
- Versiju vadības rīks Mercurial;

# PROGRAMMATŪRAS PIRMKODS

Sistēma tika izstrādāta izmantojot valodas PHP, JavaScript un HTML. Šajā sadaļā tiek iekļauti tikai daži faili no visas sistēmas-pa vienam no katra moduļu tipa. Dažas kodu rindas var tikt sadalītas pa vairākām rindām, un par īstu rindas beigu uzskatīt par vietu, kur rindas beigas parādās simbols „ ; ”. Katra faila(moduļa) nolūks ir aprakstīts dokumenta „Programmatūras projektējuma apraksts”.

## 1. Galvenā kontroliera saturs

```
<?php

Class Main{

    public $mysql_user;
    public $mysql_host;
    public $mysql_password;
    public $mysql_db;
    public $connection;

    public function __construct($data) {

        if($this->DataBaseConnect()) {
            $data=$this->ShowLinkToMap($data);
            $data=$this->TwitterLogin($data);
            $data=$this->TwitterLoginCallback($data);
            $this->UserCheck();
            $this->LoadControlers($data);
        }

    }

    public function DataBaseConnect() { //Metode izveido pieslegumu uz datubāzi

        mb_language('uni');
        mb_internal_encoding('UTF-8');

        $this->mysql_host = "localhost";
        $this->mysql_user = "sb_dbu";
        $this->mysql_password = "stasta3U";
        $this->mysql_db = "apmap";

        $this->connection = @mysql_connect($this->mysql_host, $this->mysql_user ,
        $this->mysql_password);

        if ($this->connection) {
            mysql_set_charset("utf8",$this->connection);
            mysql_select_db($this->mysql_db, $this->connection);
            return true;
        }
        else{
            $this->ShowSturtUpScreen();
            return false;
        }
    }
}
```

```

} //-----metode DataBaseConnect beigas

public function LoadControllers($data){ //Metode kurā ielāde nepieciešamo
kontrolieri
$ComplitData=0; //saskaita cik parametri tik saņemta (kontroliris, metode)
if (isset($data["controller"])) {
    $Controller=$data["controller"];
    $ComplitData++;
}

if (isset($data["method"])) {
    $Method=$data["method"];
    $ComplitData++;
}

// ja ir noteikts ka ir dati gan par kontrolieri gan par metodi tad veic
kontroliera pieslegšanu
if ($ComplitData==2) {
    include("controllers/Objects.php");
    include("controllers/Advices.php");
    include("controllers/RigasTransport.php");
    include("controllers/User.php");

    switch ($Controller){
        case "objects":
            // objektam padotie parametri ir izsaucāmā kontroliera metode, norāde
            uz klasi Main un post dati
            $Objects= new Objects($Method,$this,$data);
            break;

        case "advices":
            $Advices= new Advices($Method,$this,$data);
            break;

        case "rigastransport":
            $RigasTransport= new RigasTransport($Method,$this,$data);
            break;

        case "user":
            $User= new User($Method,$this,$data);
            break;

        default:
            $this->ShowSturtUpScreen(); //izsauc programmas uzskāšanu ja nav padota
            kontroliera
            break;
    }

}

else{
    //ja post datos nav kontroliera un tā metode izsauc programmas uzsakšanas
    logu
    $this->ShowSturtUpScreen();
}

} // -----metodes
LoadControllers beigas

public function ShowSturtUpScreen(){ //Metode kas ielāde HTML failu ar kuru
uzsaksies programma
if (!$this->connection) {
    $Data['dberror']=true;

```

```

}
    $Data['onload']='constStartUp()'; // onload izsaucāma funkcija
    include("view/startup.php");
} // -----metodes
ShowSturtUpScreen beigas

public function TwitterLogin($data){ // Šī metode palīdz lietotāju nostādīt
parametrus, lai lietotāj nonāk uz uz tvitera autentifikāciju
if((isset($_GET['controller'])) &&
(isset($_GET['method'])) && ($_GET['controller']=='user') && ($_GET['method']=='
Login')){
    $data=array();
    $data['controller']='user';
    $data['method']='Login';
}

return $data;
} // -----metodes TwitterLogin beigas

public function TwitterLoginCallback($data){ // Metode izveido pareizo
kontroliera izsaukumu, kas atbild par atbildes no tvitera apstrādi

if((isset($_GET['controller'])) &&
(isset($_GET['method'])) && ($_GET['controller']=='user') && ($_GET['method']=='
LoginRespons')){
    if ((isset($_SESSION['wait_twitter_respons'])) ) &&
($_SESSION['wait_twitter_respons']==1)){
        $data=array();
        $data['controller']='user';
        $data['method']='LoginRespons';}
    else{
        header('Location: /index.php');
    }
}

return $data;
} // -----metodes TwitterLoginCallback beigas

public function ShowLinkToMap($data){ // ----- Metode izveido pareizu
pieprasījumu uz kontrolieri kas atbild par komentāru atelošanas pec
hipersaites
if(isset($_GET['tm'])){
    $data=array();
    $data['controller']='user';
    $data['method']='LinkToMap';
    $data['tm']=$_GET['tm'];
}

return $data;
} //-----metodes ShowLinkMap beigas

```



```

public function UserCheck(){ // Metode pārbauda vai lietotāja
autentifikācijas dati ir uzstādīti pareizi
if(isset($_SESSION['login'])){

    if(isset($_SESSION['uid']) && isset($_SESSION['name']) &&
isset($_SESSION['login']) && $_SESSION['login']==1 ){
        $_SESSION['login']=1;
    }
    else{
        $_SESSION['uid']=0;
        $_SESSION['login']=0;
        $_SESSION['name']=0;
        $_SESSION['tid']=0;
    }
}
else{
    $_SESSION['uid']=0;
    $_SESSION['login']=0;
    $_SESSION['name']=0;
    $_SESSION['tid']=0;
}

} //----- metode UserCheck beigas

} // Klases Main beigas

session_start();
ini_set("display_errors","2"); //Ļauj attēlot kļūdas
ERROR_REPORTING(E_ALL);

if(isset($_POST)==false){ // mainīgais $data kurš aizpildīts ar POST
datiem vai izveidots tukšs
    $data=array();
}
else{
    $data=$_POST;
}

$main = new Main($data);
?>

```

## 2. Rīgas sabiedriska transporta kontroliera saturs

```
<?php

Class RigasTransport{
public $CallingObject;//glabā index.php klasi, lai izmantot tas iekšējos
parametrus
public $Model=array(); //masīvs kurā glabās modeļu klasēs
public $PostData;
public function __construct($Method='0',$Calling='0',$data='0'){

$this->CallingObject=$Calling; //Calling objekts kurš izsauca šo klasi,
izdara šo piešķiršanu,lai augstāk definētie parametri tādi ka konekcijas
varētu tikt izmantotas arī šajā klasē

if((isset($Method)==false)||($Method=='0')){
    header('Location: /index.php');
    return;
}

$this->PostData=$data;
$this->InitializeModels();
$this->$Method();
}

public function InitializeModels(){ //metode kura pieslēdz kontrolierim
nepieciešamos modeļus
    include("models/RigasTransport_model.php");
    include("models/Data_model.php");
    $this->Model["Data_model"]= new Data_model();
    $this->Model["RigasTransport_model"]= new RigasTransport_model();
} //-----metodes InitializeModels beigas

public function GetStopsSchedulBoard(){ //metode sameklē pieturā pienakošos
maršrūtus

$RequiredFields= array ( 'StopID' => 'ERR_UNDEF' , 'DayID' => 'ERR_UNDEF')
;
$Response = $this->Model["Data_model"]->CheckFieldSet($RequiredFields,
$this->PostData);

if($Response!=true){
    echo json_encode($Response);
    return;
}

$Scheduls=array(); //šajā masīva glabās maršrūtus
//saņemto datu pārbaude
// Sameklēs maršrūtus pieturā atkarībā no dienas tipa
$Scheduls = $this->Model["RigasTransport_model"]->GetTransportOnStop($this->
PostData["StopID"],$this->PostData["DayID"]);

if($Scheduls==false){
    echo json_encode("ERR_NTF");
    return;
}

echo json_encode($Scheduls);
} //-----metodes GetStopsSchedulBoard beigas
```

```

public function GetDepartureTimes(){ // metode kura sameklē maršruta
pienākšanas laikus pieturā saņemto datu pārbaude
$RequiredFields= array ( 'TOSID' => 'ERR_UNDEF' , 'Hours' => 'ERR_UNDEF',
'Minutes' => 'ERR_UNDEF') ;
$Response = $this->Model["Data_model"]->CheckFieldSet($RequiredFields,
$this->PostData);

if($Response!=true){
    echo json_encode($Response);
    return;
}

if((is_int((int)$this->PostData["Hours"])==false)|| (is_int((int)$this-
>PostData["Minutes"])==false)){
    echo json_encode('ERR_UNDEF');
    return;
}

// iegūst pienākšanas laikus pieturā
$DepartureTimes=array();// šajā masīva glabās maršruta pienākšanas laikus
pieturā
$DepartureTimes = $this->Model["RigasTransport_model"]->
GetDepartureTimes($this->PostData["TOSID"],$this->PostData["Hours"],
$this->PostData["Minutes"]);

if($DepartureTimes==false){
    echo json_encode("ERR_NDTF");
    return;
}
echo json_encode($DepartureTimes);
} //-----metodes GetDepartureTimes beigas

public function GetRoutsData(){ // metode kas iegust maršruta pieturas
// saņemto datu pārbaude

$RequiredFields= array ( 'TOSID' => 'ERR_UNDEF' , 'TimeID' => 'ERR_UNDEF')
;
$Response = $this->Model["Data_model"]->CheckFieldSet($RequiredFields,
$this->PostData);

if($Response!=1){
    echo json_encode($Response);
    return;
}

$TripID = $this->Model["RigasTransport_model"]->GetTripID($this-
>PostData["TOSID"]); //iegust maršruta identifikātoru
$RoutsData=$this->Model["RigasTransport_model"]->GetRoutsData($this-
>PostData["TimeID"], $TripID); // iegust pieturas maršrutā

if($RoutsData==false){
    echo json_encode('ERR_NOSTOPS');
    return;
}

echo json_encode($RoutsData);
} //-----metodes GetRoutsData beigas

}
?>

```

### 3. Rīgas sabiedriska transporta modeļa saturs

```
<?php
Class RigasTransport_model{

public function __construct(){
}

public function GetStops(){ // metode kas atlasa pieturas no datubazes

$returnArray=array();

$select = 'SELECT * FROM stops';

$result = @mysql_query($select);

if($result){

    while($row=mysql_fetch_array($result,MYSQL_ASSOC))
    {
        $returnArray[]=$row;
    }
}
return $returnArray;

} //-----metodes GetStops beigas

public function GetTransportOnStop($StopID,$DayID){ // metode kura iegust
maršrutu sarakstu pietura
if($DayID==2){ // 2 nozīme ka lietotāis meklē transportu "tagad"

$WeeksDay=date("D"); // iegust dienas saīsināto nosaukumu

switch($WeeksDay) { // nosaka kādu dienas tipu jāmeklē
case 'Mon': $DayID = 0; break;
case 'Tue': $DayID = 0; break;
case 'Wed': $DayID = 0; break;
case 'Thu': $DayID = 0; break;
case 'Fri': $DayID = 0; break;
case 'Sat': $DayID = 1; break;
case 'Sun': $DayID = 1; break;
}
}
$returnArray=array();
// atlasa tos maršrūtus, kuri ir izvēlētajā dienā un pietura
$select = "Select triponstops.id as TOSID, StopsID, Schedule_id, trip.id as
Trip_id, Direction_id, Day, Number, Name, Type
From trip, transporttype, triponstops
where StopsID = '".mysql_real_escape_string($StopID)."' and TripID =
trip.ID and TrtID = transporttype.ID and
Day='".mysql_real_escape_string($DayID)."'
order by Number ASC";

$result = @mysql_query($select);

if($result==false){
return false;}
}
```

```

if($result){

    while($row=mysql_fetch_array($result,MYSQL_ASSOC))
    {
        $returnArray[]=$row;
    }
}
return $returnArray;

} //-----metodes
GetTransportOnStop beigas

public function GetDepartureTimes($TOSID, $Hours, $Minutes){ // metode kura
iegust maršruta pienākšanas laikus pietura
$returnArray=array();
$Time=$Hours.":".$Minutes; // savieno kopa stundas un minutes
// atlasa no datubazes tos laikus, kas ir lielaki par ieguto
$select = "SELECT *
FROM `departuretime`
WHERE TOSID='".mysql_real_escape_string($TOSID)."' and Time>='". $Time."'";

$result = @mysql_query($select);

if($result==false){
return false;}

if($result){

    while($row=mysql_fetch_array($result,MYSQL_ASSOC))
    {
        $returnArray[]=$row;
    }
}
return $returnArray;

} //----- metodes GetDepartureTimes beigas

public function GetTripID($TOSID){ //metode kura zinot atiecibu
pietura/maršruts iegust maršruta identifikatoru

$select = "SELECT TripID
FROM `triponstops`
WHERE ID='".mysql_real_escape_string($TOSID)."'";

$result = @mysql_query($select);

if($result==false){
return false;}

    if (@mysql_num_rows($result)==0) {
        return false;
    }
    //tagad esam droši, ka ir atlasijies vismaz viens ieraksts
    $returnItem = mysql_fetch_assoc($result);

return $returnItem['TripID'] ;

}

```

```

public function GetRoutsData($TimeID, $TripID){ // metode kura iegust
maršruta datus

$returnArray=array();

$select="SELECT time, stops.id as id, stops.name as name, stops.street as
street, triponstops.order as stopsorder, stops.Latitude as lat,
stops.Longitude as lng
FROM timing, stops, triponstops
WHERE
DepT=".mysql_real_escape_string($TimeID). " and stops.id=timing.StopID and
stops.id = triponstops.stopsId and triponstops.tripId =
".mysql_real_escape_string($TripID). "
ORDER BY triponstops.order ASC";

$result = @mysql_query($select);

if($result==false){
return false;}

    if (@mysql_num_rows($result)==0) {
        return false;
    }

    while($row=mysql_fetch_array($result,MYSQL_ASSOC))
    {
        $returnArray[]=$row;
    }

return $returnArray;

} //-----metodes GetRoutsData beigas

}

?>

```

## 4. Rīgas sabiedriska transporta informācijas ielādes moduļa saturs

```
<?php
Class RSTLOAD{

public $TransportArray;
public $doc;
public $Transports;

public function __construct(){
// izveido pieslegumu uz datubāzi
    mb_language('uni');
    mb_internal_encoding('UTF-8');

    $mysql_host = "localhost";
    $mysql_user = "sb_dbu";
    $mysql_password = "stasta3U";
    $mysql_db = "apmap";

    $connection = mysql_connect($mysql_host, $mysql_user , $mysql_password);
    mysql_set_charset("utf8",$connection);
    mysql_select_db($mysql_db, $connection);

    $this->TransportArray= array(); //šaja masīva glabāsies visa informācija
    par maršurtiem

    $this->doc = new DOMDocument('1.0', 'UTF-8'); //izveido jauno XML failu.
    Enkodingam vajag būt UTF-8 jo ir latviešu burti
    $this->doc->formatOutput = true;
    $this->Transports = $this->doc->createElement("RST"); //visa informācija
    saturēsies šaja taga
    $this->doc->appendChild($this->Transports );
} //-----konstruktorā beigas

public function LoadStops(){ // metode ielāde no XML faila pieturas
    datubāzē
    $empty="TRUNCATE TABLE `stops`";
    mysql_query($empty);
    $empty="TRUNCATE TABLE `departuretime`";
    mysql_query($empty);
    $empty="TRUNCATE TABLE `triponstops`";
    mysql_query($empty);
    $File="rstxml/stops.xml";
    $Content=@simplexml_load_file($File); //ielade pieturu XML faila saturu
    $Stops=$Content->stop;

    foreach($Stops as $Stop){

        $StopsNumber=$Stop['id'];
        $Latitude=$Stop['lat'];
        $Longitude=$Stop['lng'];
        $Name=$Stop['name'];

        if(isset($Stop['street'])==false){ //dažam pieturām XML faila nav
        norādītas ielas
            $Street="-";
        }
        else{
            $Street=$Stop['street'];
        }
    }
}
```





```

        if($FreeDay!=""){ // ja maršruts kurse brīvdienās tad ievac info par to
            $this->TransportArray[$i]["rout 6-7"]=array();
            $this->TransportArray[$i]["rout 6-7"]["schedul_id"]=$FreeDay;
            $this->TransportArray[$i]["rout 6-7"]["directions"]=array();
            $DirectionArray=$this->GetDirection($FreeDay,$Transport,"96");//96-
brīvdiena
            $DirNumber=1;

            foreach($DirectionArray as $Direction){
                $this->TransportArray[$i]["rout 6-
7"] ["directions"] ["direction_". $DirNumber]=$Direction;
                $DirNumber++;
            }
        }
    }

    $this->SaveToXML($this->TransportArray,$Transport); // izveidoto masīvu
pārveido par XML objektu
    $this->TransportArray= array(); // iztīra masīvu no datiem
} // -----metodes MakeTransportXML beigas

public function GetScheduls($Content){ // iegūst no XML faila scheduls_id
transportiem
    $Routes=$Content->routes;
    $RoutesArray=array();
    $i=0;

    foreach($Routes->route as $Route){ // apstaiga visus transportus un atrod to
schedule_id darbadienai un brīvdienai
        $WorkingDaySchedulId =(string)$Route->days->day[0]->schedule_id;
        $FreeDaySchedulId =(string)$Route->days->day[6]->schedule_id;
        $RoutesArray[$i]=array();
        $RoutesArray[$i]["rout 1-5"]=$WorkingDaySchedulId;
        $RoutesArray[$i]["rout 6-7"]=$FreeDaySchedulId;
        $i++;
    }

    return $RoutesArray;
} //-----metodes GetScheduls beigas

public function GetDirection($SchedulID,$Type,$DayType){ // metode izveido
masīvu katram virzienam maršrutā ar tā informāciju

    $Content =@simplexml_load_file('rstxml/'.$Type.'.xml/'.$SchedulID.'.xml');
    // ielāde maršruta xml failu

    $DirectionArray= array();
    $i=0;
    $dir=$Content->directions->direction;

    foreach($dir as $direction){ // apstaiga maršruta virzienus un katram veido
masīvu ar tā informāciju

        $DirectionArray[$i]=array();
        $DirectionArray[$i]["direction_id"]=(string)$direction['direction_id'];
        $DirectionArray[$i]["number"]=(string)$direction->number;
        $DirectionArray[$i]["name"]=(string)$direction->name;
        $DirectionArray[$i]["day_type"]=$DayType;
        $DirectionArray[$i]["transport_type"]=$Type;
    }
}

```

```

        $DirectionArray[$i]["schedul_id"]=(string)$SchedulID;
        $Stops=$this->GetWhereStops($direction,$Type,$DayType); // iegūsim masīvu
ar pieturām kur pienāk, un laikiem tajās
        $DirectionArray[$i]["stops"]=$Stops;
        $i++;
    }

    return $DirectionArray;

} //-----metodes GetDirection beigas

public function GetWhereStops($direction,$Type,$DayType){ // metode iegūst
informāciju par pieturām maršrutā un pienākšanas laikiem tajās
$Stops=array(); // masīvs kuru atgriezīs
$StopIndex=0;

foreach($direction->stops->stop as $stop){ //apstaiga visas pieturas
maršrutā
    $Stops["stop_". $StopIndex]=array();
    $Stops["stop_". $StopIndex]["stop_id"]=(string)$stop['stop_id'];
    $Stops["stop_". $StopIndex]["departure_times"]=array();
    // atrod pienākšanas laikus pieturās
    $Stops["stop_". $StopIndex]["departure_times"]=$this-
>GetDepartureTime((string)$direction['direction_id'],(string)$stop['stop_id
'], $Type,$DayType);
    (string)$stop['stop_id'];
    $StopIndex++;
}

return $Stops;

} //-----metodes GetWhereStops beigas

//31darbadienas 96 brīvdienas 127 darbadienas un brīvdienas
public function GetDepartureTime($DirectionID,$StopID,$Type,$DayType){ //
metode sameklē transporta pienākšanas laikus pieturās

$this->DownloadDepartureTimeXML($DirectionID,$StopID,$Type); //Ielāde
failus ar pienākšanas laikiem. ATKOMENTĒT ja vajag ielādēt datus

$content
=@simplexml_load_file("rstxml/".$Type."<u>leavsxm</u>l/direction_". $DirectionID."<u>_
stop_". $StopID."<u>.xml</u>"); // nolasa ielādēto failu
$DepartureTimeArray=array();

foreach($content->days->day as $Day){ // veido masīvu ar pienākšanas
laikiem atkarība no dienas
    if(($Day['day']==$DayType)||($Day['day']==127)){ // ja atbilst
pieprasītajam dienas tipam
        foreach($Day->hour as $hour){
            foreach($hour->minutes as $minut){
                $timing=$minut['id'];
                $DepartureTimeArray[]=array(
                    "time" => $hour['hr'].":".$minut,
                    "timing" => (string)$timing );
            }
        }
        //timig_id ir RST izmantotais id lai atrast laikus citās pieturās
        }
    }
}

```

```

        } //foreach hours beigas

    } // if beigas
} // foreach day beigas

return $DepartureTimeArray;
} //-----metodes GetDepartureTime beigas

public function DownloadDepartureTimeXML($DirectionID,$StopID,$Type){ //
metode lejuplāde XML failu par maršrutu noteiktajā virzienā
$url
='http://saraksti.rigassatiksme.lv/?a=p.schedule&direction_id='.$DirectionID.
&stop_id='.$StopID.'&t=xml&l=lv';
$useragent=$_SERVER['HTTP_USER_AGENT']; // bez šā parametra pieprasījums
netiek apstrādāts
// pieprasījums uz rīgas satiksmi
    $ch = curl_init();
    curl_setopt( $ch, CURLOPT_USERAGENT, $useragent );
    curl_setopt( $ch, CURLOPT_URL, $url );
    curl_setopt( $ch, CURLOPT_HEADER, false);
    curl_setopt( $ch, CURLOPT_POST, 0);
    curl_setopt( $ch, CURLOPT_FOLLOWLOCATION, true );
    curl_setopt( $ch, CURLOPT_ENCODING, '' );
    curl_setopt( $ch, CURLOPT_RETURNTRANSFER, true );
    curl_setopt($ch,CURLOPT_REFERER,'');
    curl_setopt( $ch, CURLOPT_AUTOREFERER, true );
    curl_setopt( $ch, CURLOPT_CONNECTTIMEOUT, 10 );
    curl_setopt( $ch, CURLOPT_TIMEOUT, 10 );
    curl_setopt( $ch, CURLOPT_MAXREDIRS, 10 );
    $content = curl_exec( $ch );
    $response = curl_getinfo( $ch );
    curl_close ( $ch );
// iegūto informāciju saglabā failā
$myFile =
"rstxml/".$Type."leavsxml/direction_".$DirectionID."_stop_".$StopID.".xml";
$fh = fopen($myFile, 'w') or die("can't open file");
fwrite($fh, $content);
fclose($fh);
return;
} //-----metode DownloadDepartureTimeXML beigas

public function DownloadSchedulXML($SchedulID,$Type){ // metode lejuplāde
XML failu par maršrutu

$url =
'http://saraksti.rigassatiksme.lv/?a=p.directions&schedule_id='.$SchedulID.
&t=xml&l=lv';
$useragent=$_SERVER['HTTP_USER_AGENT']; // bez šā parametra pieprasījums
netiek apstrādāts
// pieprasījums uz rīgas satiksmi
    $ch = curl_init();
    curl_setopt( $ch, CURLOPT_USERAGENT, $useragent );
    curl_setopt( $ch, CURLOPT_URL, $url );
    curl_setopt( $ch, CURLOPT_HEADER, false);;
    curl_setopt( $ch, CURLOPT_POST, 0);
    curl_setopt( $ch, CURLOPT_FOLLOWLOCATION, true );
    curl_setopt( $ch, CURLOPT_ENCODING, '' );

```

```

    curl_setopt( $ch, CURLOPT_RETURNTRANSFER, true );
    curl_setopt( $ch, CURLOPT_REFERER, '' );
    curl_setopt( $ch, CURLOPT_AUTOREFERER, true );
    curl_setopt( $ch, CURLOPT_CONNECTTIMEOUT, 10 );
    curl_setopt( $ch, CURLOPT_TIMEOUT, 10 );
    curl_setopt( $ch, CURLOPT_MAXREDIRS, 10 );
    $content = curl_exec( $ch );
    $response = curl_getinfo( $ch );
    curl_close ( $ch );
    // iegūto informāciju saglabā failā
    $myFile = "rstxml/".$Type."/".$SchedulID.".xml";
    $fh = fopen($myFile, 'w') or die("can't open file");
    fwrite($fh, $content);
    fclose($fh);
    return;
}

public function SaveToXML($Data,$Type){ // metode saņem masīvu ar
informāciju par maršrutiem noteiktam transportlīdzeklā tipam
// un izveido no tā XML objektu
$Transport = $this->doc->createElement($Type); //visa informācija
saturēsies šajā tagā
$this->Transports->appendChild($Transport);

foreach( $Data as $Routes ){ //Apstrāda katra masīva elementu atsevišķi
veidojot no tā XML elementus

    foreach($Routes as $OneRoute){ //apstrāda katru maršrutu

        if(isset($OneRoute["directions"] )){
            foreach($OneRoute["directions"] as $Direction){ // apstrāda
katra maršruta virzienu

                $Route = $this->doc->createElement( "route" ); //Katram
elementam galvenais tags objects kas saturēs informāciju par objektu

                $Element = $this->doc->createElement( "number" );
                $Element->appendChild($this->doc-
>createTextNode($Direction["number"]));
                $Route->appendChild($Element);

                $Element = $this->doc->createElement( "name" );
                $Element->appendChild($this->doc-
>createTextNode($Direction["name"]));
                $Route->appendChild($Element);

                $Element = $this->doc->createElement( "transport_type" );
                $Element->appendChild($this->doc-
>createTextNode($Direction["transport_type"]));
                $Route->appendChild($Element);

                $Element = $this->doc->createElement( "direction_id" );
                $Element->appendChild($this->doc-
>createTextNode($Direction["direction_id"]));
                $Route->appendChild($Element);

                $Element = $this->doc->createElement( "day_type" );
                $Element->appendChild($this->doc-
>createTextNode($Direction["day_type"]));
                $Route->appendChild($Element);
            }
        }
    }
}

```

```

        $Element = $this->doc->createElement( "schedul_id" );
        $Element->appendChild($this->doc-
>createTextNode($Direction["schedul_id"]));
        $Rout->appendChild($Element);

        $Element = $this->doc->createElement( "stops" ); //
        $order = 1; //order norada uz pieturas secību(kārtu) maršrutā

        foreach($Direction["stops"] as $Stop){ // izveido pieturu objektus

            $StopElement = $this->doc->createElement( "stop" );

            $StopID = $this->doc->createElement( "stop_id" );
            $StopID->appendChild($this->doc-
>createTextNode($Stop["stop_id"]));
            $StopElement->appendChild($StopID);

            $OrderNum = $this->doc->createElement( "order" );
            $OrderNum->appendChild($this->doc-
>createTextNode($order));
            $StopElement->appendChild($OrderNum);

            $DepartureTimes= $this->doc->createElement(
"departure_times" );

            foreach($Stop['departure_times'] as $DepartureTime){ //
katrai pieturai izveido tas pienakšanas laiku objektus
                $DepT= $this->doc->createElement( "departure_time"
);

                $Time = $this->doc->createElement( "time" );
                $Time->appendChild($this->doc-
>createTextNode($DepartureTime['time']));
                $DepT->appendChild($Time);

                $Timing = $this->doc->createElement( "timing_id" );
                $Timing->appendChild($this->doc-
>createTextNode($DepartureTime['timing']));
                $DepT->appendChild($Timing);
                $DepartureTimes->appendChild($DepT);
            }
            $StopElement->appendChild($DepartureTimes); // pieturu
elementiem pakarto pienakšanas laikus tajās
            $Element->appendChild($StopElement); // pieturu tagam
pakarto ieguto pieturu
            $order++;
        }

        $Rout->appendChild($Element);

        $Transport->appendChild( $Rout );
    } // foreach $OneRout["directions"] beigas
} // if $OneRout["directions"] beigas
} // foreach $Routs beigas
} // foreach $Data beigas

} // -----metode SaveToXML
beigas

public function SaveXML(){ // metode saglabā XML objektu failā
$this->doc->saveXML();
$this->doc->save("rstxml/RST.xml") ;

```

```

} //-----metode SaveXML beigas

public function LoadToDatabase(){// metode izsauc saglabāšanas funkcijas
apstrādājamajiem datiem
//attīra datubāzi no datiem
$empty="TRUNCATE TABLE `departuretime`;
mysql_query($empty);
$empty="TRUNCATE TABLE `trip`;
mysql_query($empty);
$empty="TRUNCATE TABLE `triponstops`;
mysql_query($empty);

$Content=@simplexml_load_file("rstxml/RST.xml"); //ielāde xml failu ar kūru
strādās
foreach($Content as $TransportGroup){ // apstaigā katra transporta veida
grupu

    foreach($TransportGroup as $Rout){ // saglabā informāciju par katru
maršrutu
        $TripID=$this->LoadTransport($Rout->schedul_id,$Rout-
>direction_id,$Rout->day_type,$Rout->transport_type,$Rout->number,$Rout-
>name);

        foreach($Rout->stops->stop as $Stop){ // saglaba informāciju
par pieturām maršrutā
            $TOSID=$this->LoadWherStops($TripID, $Stop->stop_id,$Stop-
>order);

            foreach($Stop->departure_times->departure_time as $Time){
// saglabā informāciju par maršruta pienākšanas laikiem pieturās
                $this->LoadDepartureTime($TOSID, (string)$Time-
>time,(string)$Time->timing_id);
            }

        }

    }

}

} //-----metodes LoadToDatabase beigas

public function
LoadTransport($SchedulID,$DirectionID,$DayType,$TransportType,$Number,$Name
){ // metode saglabā informāciju par maršrutu datbāzē
$Day;
$TrtID; //transporta identifikators
//nosaka kāda diena ir šis maršruts, un piešķit tam identifikatoru
switch($DayType){ //0- darbadienā, 1-brīvdiena.
case 31:
$Day=0;
break;

case 96:
$Day=1;
break;
}
//nosaka transporta tipa identifikatoru
switch($TransportType){
case "bus":

```

```

$TrtID=1;
break;

case "trolleybus":
$TrtID=2;
break;

case "tram":
$TrtID=3;
break;
}
// saglabā datubāzē
$insert="INSERT INTO `apmap`.`Trip` ( `Schedule_id`,`Direction_id`, `Day`,
`Number`, `Name`, `TrtID` ) VALUES ('".$ScheduleID."', '".$DirectionID."',
'".$Day."', '".$Number."', '".$Name."', '".$TrtID."');

mysql_query($insert);
$ID = mysql_insert_id(); //iegust automatiski ģenerēto ID vērtību
return $ID; //tā bus nepieciešama lai veidot saistības
} //-----metodes LoadTransport beigas


public function LoadWhereStops($TripID, $StopsNumber,$Order){// metode
saglabā informāciju par maršruta pieturām datbāzē
// vajag uzzināt kāds ir pieturas id mūsu datubāzē atkarība no tā id kas ir
Rīgas satiksmei, jo XML failos ir RS pieturu id
$select = "SELECT ID
FROM `stops`
WHERE StopsNumber = '".mysql_real_escape_string($StopsNumber)."'";

$result = @mysql_query($select);

if($result==false){
return false;}

    if (@mysql_num_rows($result)==0) {
        return false;
    }

    $answer = mysql_fetch_assoc($result);
    $StopID=$answer['ID'];
    // tagad kad zin id mūsu datubāzē var saglabāt pieturas un maršruta
    saistību
    $insert="INSERT INTO `apmap`.`triponstops` ( `StopsID`,`TripID`,`Order`)
VALUES ('".$StopID."', '".$TripID."', '".$Order."')";
    mysql_query($insert);
    $ID = mysql_insert_id();//iegust automatiski ģenerēto ID vērtību
    return $ID;//tā bus nepieciešama lai veidot saistības
} //-----metodes LoadWhereStops beigas


public function LoadDepartureTime($TOSID,$Time,$Timing){// metode saglabā
maršruta pienākšanas laikus pieturā
$insert="INSERT INTO `apmap`.`departuretime` ( `TOSID`,`Time`,`Timing_id`)
VALUES ('".$TOSID."', '".$Time."', '".$Timing."');
mysql_query($insert);
} //-----metodes LoadDepartureTime beigas

} // -----Klases beigas

```

```
$RST=new RSTLOAD();  
$RST->LoadStops(); // ieladē pieturas  
$RST->MakeTransportXML("bus"); //izveido XML objektu ar info par  
autobusiem, tamvājiem un trollejbusiem  
$RST->MakeTransportXML("tram");  
$RST->MakeTransportXML("trolleybus");  
$RST->SaveXML(); // saglaba izveidoto XML objektu failā  
$RST->LoadToDatabase(); // saglabā informāciju par maršrutiem datubāze  
izmantojot XML failu  
?>
```



## 5. Komentāru attēlošanas sadaļas moduļa saturs

```
var mapTM;
var markersArrayTM; // objektu markieri glabājas šeit
var mapTMinitialized;
var currentPositionTM;
var infowindowArrayTM;

function constTweetMapForm() { // Komentāru attēlošanas sadaļas konstruktors
  LoadingScreenShow();
  ClearScreen();

  // sagatavo mainīgos darbam
  currentPositionTM=0;
  mapTMinitialized=false;
  mapTM=Number.NaN;
  markersArrayTM =new Array();
  infowindowArrayTM = new Array();

  DrawTweetMapForm();

  var latitude = '56.946238'; //noklūsetais kartes centrs
  var longitude = '24.104905';

  InitializeMapTM(latitude,longitude);

  if(mapTMinitialized==true){

    var tweets=GetTweets(10); // pēc noklūsejuma atlasa jaunākos 10 komentārus
    if(tweets==false){return;}

    for(var i in tweets){ //apstaiga komentārus un novieto tos uz kartes
      PlaceTweetOnMap(tweets[i]['Name'], tweets[i]['Latitude'],
        tweets[i]['Longitude'], tweets[i]['Text'], tweets[i]['PostTime'],
        tweets[i]['TweetId']);
    }

    currentPositionTM=1; // pēc noklūsejuma vajadzēs aizbīdīt karti pie
    jaunā komentāra
    LoadingScreenHide();
    google.maps.event.trigger(mapTM, 'resize'); //taka visa sadaļa bija
    noslēpta kamer tika ielādēta, vajag izsaukt šo notikumu
    //lai tiktu pārbaudīta pareiza kartes attēlošana

    GoToTweet(currentPositionTM, 'No'); // 'No' jo nav iepriekšēja komentāra
    ko skatījās lietotājs
  }
  else
  { //ja karte nebija veiksmīgi ielādēta
    msg('ERR_IMAP');
    constGeoTwitterForm();
  }
} //-----funkcijas constTweetMapForm beigas

function DrawTweetMapForm() { // funkcijas, kura uzzīmē Komentāru
  attēlošanas sadaļu

  DomAdd('content','footer','div');
  DomAdd('footer','Heading_box','div','heading_midle');
  DomAdd('Heading_box','Heading_top','h1');
```

```

DomTextAdd('Heading_top',lang('map'));
DomAdd('content','map_canvas','div','map_full');
DomAdd('footer','Back_Button','div','back_button');
DomAddEvent('Back_Button','click',BackToChTM);
// izkritis izvelnes izveidošana

DomAdd('footer','Drop_List_Button','div','right_droplist_button');
DomAdd('content','Drop_List','ul','right_droplist hidden');
DomAddEvent('Drop_List_Button','click',ShowDropListTM);

DomAdd('Drop_List','10_droplist_li','li','list_item_next list_selected');
DomTextAdd('10_droplist_li','10 '+lang('last'));
DomAddEvent('10_droplist_li','click', TweetCountChenage);

DomAdd('Drop_List','20_droplist_li','li','list_item_next');
DomTextAdd('20_droplist_li','20 '+lang('last'));
DomAddEvent('20_droplist_li','click', TweetCountChenage);

DomAdd('Drop_List','40_droplist_li','li','list_item_next');
DomTextAdd('40_droplist_li','40 '+lang('last'));
DomAddEvent('40_droplist_li','click', TweetCountChenage);

DomAdd('Drop_List','50_droplist_li','li','list_item_next');
DomTextAdd('50_droplist_li','50 '+lang('last'));
DomAddEvent('50_droplist_li','click', TweetCountChenage);

DomAdd('content','ControlPanel','div')
DomAdd('ControlPanel','Prev','div','panel');

DomAddEvent('Prev','click', PrevTweet);
DomAdd('ControlPanel','Next','div','panel');

DomAddEvent('Next','click', NextTweet);
} //-----funkcijas DrawTweetMapForm beigas

function InitializeMapTM(latitude,longitude){ // funkcija, kura izveido
karti

var latlng = new google.maps.LatLng(latitude,longitude); // Sakuma uzstada
karti noklusetajas koordinates
var myOptions = { // noklusetie kartes uzstadijumi
  zoom: 16,
  minZoom:10,
  mapTypeControl:false,
  center: latlng,
  streetViewControl: false,
  mapTypeId: google.maps.MapTypeId.ROADMAP,
  zoomControl: true,
  zoomControlOptions: {
    style: google.maps.ZoomControlStyle.ANDROID,
    position: google.maps.ControlPosition.LEFT_BOTTOM
  },
  panControl:false
};

mapTM = new
google.maps.Map(document.getElementById("map_canvas"),myOptions);

// kad mainas apskatamais apgabals, ja veic resize lai but parliecinatam ka
karte ieladeta pareizi

```

```

google.maps.event.addListener(mapTM, 'bounds_changed', function() {
google.maps.event.trigger(mapTM, 'resize');});

google.maps.event.addListener(mapTM, 'click', function(event) {
mapTM.panTo(event.latLng);}); //click beigas

mapTM.initialized=true; //karte izveidota vieksmigi
} //-----funkcijas InitializeMapTM beigas

function ShowDropListTM(){ //funkcija kura parada atelojamo komentaru skaita
mainas sarakstu, kad pieskaras pogai
document.getElementById('Drop_List').className='right_droplist';
document.getElementById('Drop_List_Button').className='right_droplist_button
up_droplist_button';
DomRemoveEvent('Drop_List_Button', 'click',ShowDropListTM);
DomAddEvent('Drop_List_Button','click',HideDropListTM); //lai pieskaroties
pogai tagad saraksts pazustu
}//-----funkcijas ShowDropListTM beigas

function HideDropListTM(){//funkcija kura paslepj atelojamo komentaru skaita
mainas sarakstu, kad pieskaras pogai
document.getElementById('Drop_List').className='right_droplist hidden';
document.getElementById('Drop_List_Button').className='right_droplist_button';
DomRemoveEvent('Drop_List_Button', 'click',HideDropListTM);
DomAddEvent('Drop_List_Button','click',ShowDropListTM); //lai pieskaroties
pogai tagad saraksts paraditos
}//-----funkcijas HideDropListTM beigas

function GetTweets(limit){ // funkcija izveido pieprasījumu pēc
komentāriem un atgriež saņemtos

StartTimeout(BackToChTM);//uzstada taimeri lai pieprasījums neilgtu vairāk
par 14 sekundem
var tweetsJson= HTTPPostRequest ('index.php',
'controller=user&method=GetTweets&type='+selectedTweetSearch+'&limit='+limit
);
StopTimeout();

var tweets=JSON.parse(tweetsJson);
if(typeof tweets== "string" && (tweets.substr(0,3)=='ERR' ||
tweets.substr(0,3)=='MSG')){//Parbauda vai atbilde satur kadu no pazinojumu
kodiem
var bool= msg(tweets); // izsauc funkciju kura parada pazinojumu un
atgriež true ja tas ir parasts paziņojums un false ja kļūdas paziņojums

if(bool==false){
BackToChTM();
LoadingScreenHide();
return false;
}
}
return tweets;

}//-----funkcijas GetTweets beigas

```

```

function PlaceTweetOnMap(user, lat, lng, text, time, id){// funkcija
novieto komentaru markieru uz kartes un pievieno tiem informacijas logus

var location = new google.maps.LatLng(lat,lng); //atrašanas punkt
izveidošana
var title=user;
var image = new google.maps.MarkerImage("images/markers/twitter.png",
    new google.maps.Size(24, 24),
    new google.maps.Point(0, 0),
    new google.maps.Point(12, 12));

var marker = new google.maps.Marker({
position: location,
icon:image,
});

chunks = text.match(/.{1,30}/g); // sadala virkni ja taja ir vairak par 30
simboliem
text= chunks.join("<br>");

var info="<p class='tuser'>" + user + "</p><p class='ttext'>" + text + "</p><p
class='ttime'>" + time + "</p>";
var infowindow = new google.maps.InfoWindow({
    content: info,
});

//uz pieskariena jaatver informacijas logs
marker.onclick= function(){
    mapTM.panTo(location);
    infowindow.open(mapTM, marker);
}

google.maps.event.addListener(marker, 'click', function() {
    mapTM.panTo(location);
    infowindow.open(mapTM, marker);
});

google.maps.event.addListener(mapTM, 'click', function() {
    infowindow.close();
});

//novieto markierus un saglaba vinus un informacijas logus masivos
marker.setMap(mapTM);
currentPositionTM++;
infowindowArrayTM[currentPositionTM]=infowindow;
markersArrayTM[currentPositionTM]=marker;
}//-----funkcijas PlaceTweetOnMap beigas

function TweetCountChenage(){ //funkcija kura veic jauna pieprasita
komentaru skaitu atlasianu un uzlikšanu uz kartes
LoadingScreenShow();
ClearOverlays(markersArrayTM); //visus komentaru markierus notira no kartes

markersArrayTM =new Array();
infowindowArrayTM = new Array();
currentPositionTM=0;
var limit=this.id.slice(0,-12);
var tweets=GetTweets(limit); //atlasa pieprasito komentaru skaitu
if(tweets==false){return;}

```

```

for(var i in tweets){
    PlaceTweetOnMap(tweets[i]['Login'], tweets[i]['Latitude'],
tweets[i]['Longitude'], tweets[i]['Text'], tweets[i]['PostTime'],
tweets[i]['TweetId']);
}
currentPositionTM=1;

var dropUl=document.getElementById('Drop_List');
var lis= dropUl.getElementsByTagName('li'); // padara visus saraksta
elementus par neizveletiem
for(var i in lis){
    lis[i].className='list_item_next';
}
this.className='list_item_next list_selected'; //this šobrīd tas saraksta
elements kuram pieskaras
GoToTweet(currentPositionTM, 'No');
LoadingScreenHide();
} //-----funkcijas TweetCountChenage beigas

function PrevTweet() { // funkcijas kura pariet pie nakama komentara

var prevPos=currentPositionTM;
currentPositionTM=currentPositionTM-1;
if(currentPositionTM<1){
    currentPositionTM=markersArrayTM.length-1;
}

GoToTweet(currentPositionTM,prevPos) // pāriešanas funkcija
} //-----funkcijas PrevTweet beigas

function NextTweet() { // funkcijas kura pariet pie iepriekšējā komentara
var prevPos=currentPositionTM;
currentPositionTM=currentPositionTM+1;
if(currentPositionTM>markersArrayTM.length-1){
    currentPositionTM=1;
}

GoToTweet(currentPositionTM,prevPos); // pāriešanas funkcija
} //-----funkcijas NextTweet beigas

function GoToTweet(i, y) { // funkcija palīdz parslekties starp komentāriem
if(y!='No'){ // Ja bija kāds "iepriekšējais" komentārs, aizver tā
informācijas logu
infowindowArrayTM[y].close();
}
var latlng = markersArrayTM[i].getPosition(); // pariet pie pieprasīta
komentāra
mapTM.panTo(latlng);
infowindowArrayTM[i].open(mapTM, markersArrayTM[i]); //atver pie prasīta
komentāra informāciju
} //-----funkcijas GoToTweet beigas

function BackToChTM() { //funkcija atgriež uz iepriekšējo sadalu
//atbrīvo visus mainīgos no vērtībām
currentPositionTM=Number.NaN;
mapTM=Number.NaN;
infowindowArrayTM= Number.NaN;
markersArrayTM =Number.NaN;
mapTMInitialized=Number.NaN;

```

```

selectedTweetSearch=Number.NaN;
DomRemoveEvent('Back_Button', 'click', BackToChTM); //noniem no pogas
atpakaļ no funkciju
constGeoTwitterForm();
}//-----funkcijas BackToChTM beigas

```

## 6. Atrāšanas vietas noteikšanas moduļa saturs

```

function GeoLocation(CallbackFunction){ //funkcija nosāka lietotāja
atrašanas vietu

    if (navigator.geolocation) //vai ģeolokācija ir pieejama
    {
        navigator.geolocation.getCurrentPosition(

            function (position) { // veiksmīgais iznākums
                CallbackFunction(position.coords.latitude,position.coords.longitude);
            // padot callback funkcijai koordinātes
            },

            function (error){switch(error.code) // neveiksmīgais iznākums
                {
                    //Ja rodas kļūdas jārada kļūdas paziņojums,°
                    //bet vienlaigā ļauj skatīties karti, pieņemot
                    noklusēto lietotāja atrašanas vietu
                    case error.TIMEOUT:
                        msg('ERR_GUNDEF');
                        CallbackFunction();
                        break;
                    case error.POSITION_UNAVAILABLE:
                        msg('ERR_GUNDEF');
                        CallbackFunction();
                        break;
                    case error.PERMISSION_DENIED:
                        CallbackFunction();
                        break;
                    case error.UNKNOWN_ERROR:
                        msg('ERR_GUNDEF');
                        CallbackFunction();
                        break;});
                }
            else{
                msg('ERR_GUNDEF');
                CallbackFunction();
            }
        }
    }

```

## SECINĀJUMI

Kvalifikācijas darbā izstrādāju mobilo tālrunu lietojumprogrammatūru, kuras galvenās funkcijas ir sniegt lietotājiem informāciju par objektiem viņu tuvumā, un sniegt informāciju par maršrutu sarakstiem pieturās, ka arī, atstāt komentārus sava atrašanās vietā.

Projekta izstrādes laikā bija iespēja pielietot jau iegūtas zināšanas universitātē, bet ļoti daudzas zināšanas vajadzēja apgūt pašmācības ceļā. Šī bija pirmā pieredzē mājas lapu izstrādē, kas paredzēti viedai tālruniem. Paplašināju un ieguvu zināšanas PHP, Javascript, CSS 3 un Ajax tehnoloģiju izmantošanā. Ieguvu zināšanas par to, kā sistēmas apmainās sava starpā ar datiem izmantojot XML datus, un kā var ievākt nepieciešamo informāciju no Rīgas satiksmes un OpenStreetMap. Iemācījos ieintegrēt Twitter.com un Google Maps savā sistēmā. Guvu pieredzi darbā ar dažādiem rīkiem, kas var palīdzēt programmas vienībtestēšana, operāciju izpildes laika mērīšanā.

Izstrādājot šo projektu, vajadzēja atcerēties, ka sistēma ir domāta viedai tālruniem, kuru ātrdarbība un atmiņas resursi ir stipri zemāki, nekā stacionāram vai portatīvam datoram, tāpēc daudz laika tika pavadīts pie kodā optimizēšanas un sistēmas ātrdarbības uzlabošanas.

Šī projekta izstrādes rezultāta guvu plašākas zināšanas par projekta izstrādes procesu un programmēšanas iemaņas cēlās augstāk, par tām, kas bija pirms šī projekta. Nobeidzot šo projektu, pamanīju lietas, kuras man patika un mēģināšu tas pielietot arī citos projektus, un tās kurās es citā projektā, taisītu savādāk.

## IZMANTOTĀ LITERATŪRA

1. LVS 68:1996, „Programmatūras prasību specifikācijas ceļvedis”;
2. LVS 72:1996, „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”;
3. LVS 73:1996, „Programmatūras vienībtestēšana” ;
4. COCOMO 81 darbietilpības aprēķināšanas metode [tiešsaiste] – [atsauce 15.02.2010].  
Pieejams: [http://sunset.usc.edu/research/COCOMOII/cocomo81\\_pgm/cocomo81.html](http://sunset.usc.edu/research/COCOMOII/cocomo81_pgm/cocomo81.html) ;
5. COCOMO darbietilpības aprēķināšanas metode [tiešsaiste] – [atsauce 15.02.2010].  
Pieejams: <http://arthurdejong.org/cocomo/cocomo.html> .
6. Google Maps Javascript API V3 [tiešsaiste] – [atsauce 21.03.2011].Pieejams:  
<http://code.google.com/apis/maps/documentation/javascript/reference.html> ;
7. Termini un definīcijas [ tiešsaiste] – [atsauce 11.04.2011]. Pieejams: <http://termini.lv/> ;
8. Termins – AGPS [tiešsaiste] - [atsauce 11.04.2011]. Pieejams:  
[http://en.wikipedia.org/wiki/Assisted\\_GPS](http://en.wikipedia.org/wiki/Assisted_GPS) ;
9. Vikipēdijas raksts „Ģeogrāfiskai platums” [tiešsaiste] – [atsauce 11.04.2011]. Pieejams:  
[http://lv.wikipedia.org/wiki/Ģeogrāfiskais\\_platums](http://lv.wikipedia.org/wiki/Ģeogrāfiskais_platums);
10. Vikipēdijas raksts „Ģeogrāfiskai garums” [tiešsaiste] – [atsauce 11.04.2011]. Pieejams:  
[http://lv.wikipedia.org/wiki/Ģeogrāfiskais\\_garums](http://lv.wikipedia.org/wiki/Ģeogrāfiskais_garums) ;
11. OpenStreetMap [tiešsaiste] – [atsauce 11.04.2011]. Pieejams:  
<http://wiki.openstreetmap.org/wiki/NewUserGuideLV> ;
12. Vikipēdijas raksts „UTF-8” [tiešsaiste] – [atsauce 11.04.2011]. Pieejams:  
<http://lv.wikipedia.org/wiki/UTF-8> ;
13. Vikipēdijas raksts „Cascading Style Sheets" [tiešsaiste] – [atsauce 11.04.2011]. Pieejams:  
[http://lv.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://lv.wikipedia.org/wiki/Cascading_Style_Sheets) .
14. Vikipēdijas raksts „Cascading Style Sheets" [tiešsaiste] – [atsauce 11.04.2011]. Pieejams:  
<http://lv.wikipedia.org/wiki/Twitter>



# PIELIKUMI

## 1. Ekranformu piemēri

### 1.1. Sākumlapas forma

Kad sistēma ir ielādēta, lietotājam tiek parādīta sākumlapa, kurā viņš sarakstā var izvēlēties kādu darbību viņš vēlas veikt.



*1.1. att. Sākumlapas forma*

## 1.2. Ieteikumu sadaļas forma

Ieteikumu sadaļā sastāv no divām formām, kur vienā lietotājs var izvēlēties ieteikta objekta atrašanas vietu uz kartes un otrā norādīt ieteikta objekta aprakstu. Ir divas pogas [Tālāk] – zilā krāsā, [Atpakaļ] – sarkanā.



1.2. att. Ieteikumu sadaļas forma

### 1.3. Kategoriju izvēles sadaļas forma

Šajā sadaļā lietotājs var izvēlēties, kādas kategorijas objektus viņš vēlās redzēt uz kartes. Formā ir poga [Atpakaļ] un saraksts ar objektu kategorijām.



1.3. att. Kategoriju izvēles sadaļas forma

#### 1.4. „Objekti apkārt man” sadaļas forma

Formā ir attēlotā kartē ar objektiem. Katra tipa objektiem ir savs unikāls marķieris. Uz kartes tiek attēlota lietotāja atrašanās vieta ar marķieri cilvēciņa formā. Uz galvenes poga [Atpakaļ] un izkrītoša saraksta atvēršanas aizvēršanas poga.



1.4. att. „Objekti apkārt man” sadaļas forma

## 1.5. Dienas izvēles sadaļas forma

Šajā sadaļā lietotājs var izvēlēties, kurā dienā viņš meklē transportu. Formā ir saraksts ar dienu tipiem. Uz galvenes ir poga [Atpakaļ].



1.5. att. Dienas izvēles sadaļas forma

## 1.6. Pieturas saraksta sadaļas forma

Šajā sadaļā lietotājs var redzēt maršrutus, kuri pienāk izvēlētajā pieturā. Forma ir saraksti ar maršrutiem. Katra veida transports ir savā atsevišķā sarakstā. Ja kāda transporta veida nav pieturā, tas saraksts netiek rādīts. Uz galvenes ir poga [Atpakaļ].



1.6. att. Pieturas saraksta sadaļas forma

## 1.7. Maršruta pienākšanas laiku sadaļas forma

Šajā sadaļā lietotājs var redzēt noteikta maršruta pienākšanas laikus pieturā. Formā tiek attēlots saraksts ar pienākšanas laikiem. Uz galvenes ir poga [Atpakaļ].



1.7. att. Maršruta pienākšanas laiku sadaļas forma

## 1.8. Maršruta laiku pieturās sadaļās forma

Šajā sadaļā lietotājs var apskatīties maršruta pienākšanas laikus pieturās pēc izvēlētas. Formā tiek attēlots saraksts ar pienākšanas laikiem un pieturu nosaukumiem. Uz galvenes ir poga [Atpakaļ] kreisajā malā, un labajā malā ir poga, kas ļauj apskatīt maršrutu uz kartes.



1.8. att. Maršruta laiku pieturās sadaļās forma



## 1.9. Maršruta uz kartes sadaļas forma

Šajā sadaļā, lietotājs var redzēt karti, uz kuras tiek attēlota viņa atrašanās vieta, un maršruta pieturas. Sakuma pieturas marķieris ir gaiši zilā krāsā, pieturas, kurās transports ir bijis – pelēkā, pietura, kuru skatās lietotājs – zaļa, nākamās pieturas – zila, un gala pietura – sarkana. Uz galvenes ir poga [Atpakaļ].



1.9. att. Maršruta uz kartes sadaļas forma

### 1.10. ĢeoTvitera sadaļas forma

Šajā sadaļā lietotājs var izvēlēties kādu darbību viņš vēlās veikt ĢeoTviterī. Sadaļas piedāvātas darbības ir atkarīgas no tā, vai lietotājs ir autorizējies sistēmā, vai nav.



1.10. att. ĢeoTvitera sadaļas forma

### 1.11. Komentāru atstāšanas sadaļas forma

Šajā sadaļā lietotājs var atstāt komentārus savā atrašanās vietā. Sadaļa sastāv no kartes, uz kuras tiek attēlota atrašanās vieta un komentāra ievades tekstā lauka. Ir divas pogas [Tālāk] – zilā krāsā, [Atpakaļ] – sarkanā. Ar šo pogu palīdzību var pārslēgties starp skatiem, iziet no sadaļas, vai nosūtīt komentāru uz sistēmu.



1.11. att. Komentāru atstāšanas sadaļa forma

## 1.12. Komentāru attēlošanas sadaļa forma

Šajā sadaļā lietotājs var lasīt citu lietotāju komentārus. Komentāri uz kartes tiek attēloti ar marķieru palīdzību, kuriem pieskaroties, atvērās informācijas logs, kurā ir komentāra teksts. Papildus manuālai komentāru izvēlei, starp komentāriem var pārslēgties ar pogu [Nākamais] un [Iepriekšējais] palīdzību, kas atrodas uz formas apakšējās joslas.



1.12.att. Komentāru attēlošanas sadaļas forma

### 1.13.Valodas maiņas sadaļas forma

Šajā sadaļā lietotājs var nomainīt saskarnes valodu. Forma sastāv no saraksta ar pieejamām valodām.



1.13 att. Valodas maiņas sadaļas forma

## 2. Paziņojumi

Identifikatoru pirmā puse ERR vai MSG norāda uz paziņojumu tipu – ERR – kļūdas paziņojums, MSG – informējošais paziņojums.

Identifikators	Ziņojuma teksts
ERR_DB	Nesanāk pieslēgties datubāzei!
ERR_UNDEF	Notika kļūda! Pamēģiniet vēlreiz vēlāk!
ERR_NOF	Neviens objekts netika atrasts!
ERR_NG	Radās problēma nosakot jūsu atrašanas vietu!
ERR_TXT500	Simbolu skaitam jābūt robeža no 0 līdz 500!
ERR_CORD	Notika kļūda koordinātu apstrādē!
ERR_NCORD	Noradiet koordinātes!
ERR_NADTXT	Noradiet ieteikuma tekstu!
ERR_NCF	Neviena kategorija netika atrasta!
ERR_GUNDEF	Noteiktu Jūsu atrašanas vietu neizdevās!
ERR_IMAP	Karti neizdevās ielādēt!
ERR_NTF	Transporti šai pieturai netika atrasti
ERR_NDTF	Transporta pienākšanas laikus neizdevās atrast!
ERR_NOSTOPS	Neviena cita pietura netika atrasta!
MSG_THX	Paldies!
MSG_NOINR	Neviens objekts 3km rādiusā netika atrasts. Pamēģiniet meklēt neierobežotā attālumā!
MSG_NIRIGA	Programma sniedz informāciju par objektiem tikai Rīgā!
MSG_OUTRIGA	Jūs vērtāt apskatīt tikai Rīgu!
ERR_TIME14	Pieprasījums izpildījās pārāk ilgi!
MSG_TWP	Ziņojums veiksmīgi pievienots!
ERR_NFTF	Komentārus neizdevās atrast!
ERR_FAUTH	Neizdevās autorizēties sistēmā!
ERR_TXT140	Simbolu skaitam jābūt robeža no 0 līdz 140!

Kvalifikācijas darbs „*Mobilā lietojumprogrammatūra navigācijai Rīgā*” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Jevgēnijs Cibankovs* \_\_\_\_\_ .05.2011.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *B.dat. Ēriks Klotiņš* \_\_\_\_\_ .05.2011.

Recenzents: *M.dat. Kārlis Gobleja*

Darbs iesniegts 26.05.2011.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Dainis Dosbergs* \_\_\_\_\_

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

\_\_\_\_.06.2011. prot. Nr. \_\_\_\_\_, vērtējums \_\_\_\_\_ (\_\_\_\_\_)

Komisijas sekretārs(-e): \_\_\_\_\_