

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

BIĻEŠU TIRDZNIECĪBAS UN REISU UZSKAITES
SISTĒMA

KVALIFIKĀCIJAS DARBS

Autors: Artūrs Aleksāns

Stud. apl. aa08291

Darba vadītājs: profesors, dekāns, Dr. h. dat.

Juris Borzovs

RĪGA 2010

ANOTĀCIJA

Biļešu tirdzniecības un reisu uzskaites sistēma ir paredzēta lietošanai autoostās un uzņēmumos, kas nodarbojas ar pasažieru pārvadājumiem. Ar tās palīdzību var efektīvi organizēt biļešu tirdzniecības procesus, reisu uzskaiti un aktuālās informācijas par reisu kustību iegūšanu.

Kvalifikācijas darba ietvaros tika izstrādāta šāda sistēma.

Atslēgvārdi: biļešu tirdzniecība, uzskaites sistēma.

ABSTRACT

Ticket sales and bus routes accounting system is meant for bus stations and companies, which are dealing with passenger traffic. With its help, they can efficiently manage ticket sales processes, route accounting and obtain actual information on bus movement.

Within my qualification work this system was developed.

Keywords: ticket sales, accounting system.

SATURA RĀDĪTĀJS

APZĪMĒJUMI.....	1
IEVADS.....	2
PĒTĀMĀ PROBLĒMA.....	2
DARBA MĒRĶI UN UZDEVUMI	2
IZMANTOTĀS METODES	2
FAKTOLOĢISKĀ MATERIĀLA AVOTI.....	3
DARBA STRUKTŪRA	3
1. NODAĻA.....	4
PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	4
1.1. IEVADS	4
1.1.1. <i>Nolūks</i>	4
1.1.2. <i>Darbības sfēra</i>	4
1.1.3. <i>Definīcijas, akronīmi un saīsinājumi</i>	4
1.1.4. <i>Saisība ar citiem dokumentiem</i>	6
1.1.5. <i>Dokumenta pārskats</i>	6
1.2. VISPĀRĒJS APRAKSTS	6
1.2.1. <i>Produkta perspektīva</i>	6
1.2.1 <i>Produkta funkcijas</i>	6
1.2.2 <i>Lietotāju raksturiezīmes</i>	7
1.2.3 <i>Vispārējie ierobežojumi</i>	8
1.2.3.1 Tīkls.....	8
1.2.3.2 OS atbalsts un programmatūra	8
1.2.3.3 Aparatūra	8
1.2.4 <i>Pieņemumi un atkarības</i>	9
1.2.4.1 Izziņas par reisu kustības sarakstu	9
1.2.4.2 Reisa laika norāde.....	9
1.3 FUNKCIONĀLĀS PRASĪBAS	9
1.3.1 <i>Lietotāju pārvaldība</i>	9
1.3.1.1 Jauna lietotāja izveide.....	9
1.3.1.2 Lietotāja datu maiņa	10
1.3.1.3 Lietotāja dzēšana.....	11
1.3.2 <i>Pieturu pārvaldība</i>	11
1.3.2.1 Pieturas izveide.....	11
1.3.2.2 Pieturas datu maiņa	12
1.3.2.3 Pieturas dzēšana	13
1.3.3 <i>Maršrutu pārvaldība</i>	13
1.3.3.1 Jauna maršruta izveide.....	13
1.3.3.2 Maršruta datu rediģēšana	14
1.3.3.3 Maršruta dzēšana.....	15
1.3.4 <i>Lietotāju pilnvarošānās</i>	16
1.3.4.1 Lietotāja pieteikšanās	16
1.3.4.2 Lietotāja atteikšanās	16
1.3.5 <i>Bīlešu pārdošana un atpakaļpieņemšana</i>	17
1.3.5.1 Bīletes pārdošana	17
1.3.5.2 Bīletes atpakaļpieņemšana.....	19
1.3.6 <i>Bīletes rezervācija un rezervācijas atcelšana</i>	20
1.3.6.1 Bīletes rezervācija	20
1.3.6.2 Bīletes rezervācijas atcelšana	22
1.3.7 <i>Reisu kustības saraksta apskate</i>	22
1.4 NEFUNKCIONĀLĀS PRASĪBAS	24
1.4.1 <i>Datu drošība</i>	24
1.4.2 <i>Pieejamība</i>	24
2. NODAĻA.....	25
PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....	25
2.1. IEVADS	25
2.1.1. <i>Nolūks</i>	25

2.1.2.	<i>Darbības sfēra</i>	25
2.1.3.	<i>Definīcijas, akronīmi, saīsinājumi</i>	25
2.2.	SAISTĪBA AR CITIEM DOKUMENTIEM.....	27
2.3.	DEKOMPZĪCIJAS APRAKSTS.....	27
2.4.	ATKARĪBAS APRAKSTS.....	28
2.4.1.	<i>Konteksta līmeņa datu plūsmas diagramma</i>	28
2.4.2.	<i>Pirmā līmeņa datu plūsmas diagramma</i>	29
2.4.3.	<i>Otrā līmeņa datu plūsmas diagrammas</i>	31
2.5.	DETALIZĒTS DATU PROJEKTĒJUMS.....	38
2.5.1.	<i>Tabula „user”</i>	38
2.5.2.	<i>Tabula „stop”</i>	39
2.5.3.	<i>Tabula „route”</i>	39
2.5.4.	<i>Tabula „route_stop”</i>	39
2.5.5.	<i>Tabula „trip”</i>	40
2.5.6.	<i>Tabula „ticket”</i>	40
2.6.	FIZISKAIS ER MODELIS.....	41
3.	NODAĻA.....	42
	TESTĒŠANAS DOKUMENTĀCIJA.....	42
3.1.	IEVADS.....	42
3.1.1.	<i>Nolūks</i>	42
3.1.2.	<i>Darbības sfēra</i>	42
3.1.3.	<i>Saistība ar citiem dokumentiem</i>	42
3.2.	TESTĒŠANAS REZULTĀTI.....	43
3.2.1.	<i>Lietotāja pieteikšanās</i>	43
3.2.2.	<i>Lietotāja izveide</i>	43
3.2.3.	<i>Lietotāja datu maiņa</i>	44
3.2.4.	<i>Lietotāja dzēšana</i>	44
3.2.5.	<i>Pieturas izveide</i>	44
3.2.6.	<i>Pieturas datu maiņa</i>	45
3.2.7.	<i>Pieturas dzēšana</i>	45
3.2.8.	<i>Reisu kustības saraksta atlase</i>	46
3.2.9.	<i>Biļetes pārdošana</i>	46
3.2.10.	<i>Biļetes rezervācija</i>	47
4.	PROGRAMMATŪRAS PROJEKTA ORGANIZĀCIJA.....	48
5.	KVALITĀTES NODROŠINĀŠANA.....	49
6.	KONFIGURĀCIJU PĀRVALDĪBA.....	50
7.	DARBIETILPĪBAS APRĒĶINS.....	51
8.	PROGRAMMATŪRAS PIRMKODS.....	52
9.	DOKUMENTĀRĀ LAPA.....	61

APZĪMĒJUMI

Apzīmējums	Skaidrojums
DB	Datu bāze
ER	Entītiņu relācijas
PPS	Programmatūras prasību specifikācija
PPA	Programmatūras projektējuma apraksts
HTML	Hiperteksta marķēšanas valoda
PHP	PHP Hypertext Preprocessor – servera puses skriptu valoda
JavaScript	Lietotāja puses skriptu valoda
AJAX	Aktīvais JavaScript un HTML
Apache	Tīmekļa serveris
MySQL	Datu bāzu pārvaldības sistēma
SQL	Strukturēta vaicājumu valoda

IEVADS

Pētāmā problēma

Uzņēmumiem, kas nodarbojas ar pasažieru pārvadājumiem, pamata uzdevums ir nodrošināt klientam – pasažierim nokļūšanu no viņa atrašanās vietas viņam vēlamajā vietā, ņemot vērā arī datuma un laika apstākļus, gūstot no tā peļņu. Lai tiktu galā ar šo uzdevumu ir nepieciešams veids kā efektīvi organizēt tajā iesaistītos vienumus.

Darba mērķi un uzdevumi

Kā kvalifikācijas darba mērķis tika izvirzīta augstākminētās problēmas atrisināšana, izveidojot biļešu tirdzniecības un reisu uzskaites sistēmu.

Lai sasniegtu darba mērķi, tika izvirzīti sekojoši uzdevumi:

1. izpētīt, kā notiek pasažieru pārvadājumi, kādi ir procesi starp pasažiera vēlmi kaut kur doties un viņa nonākšanu izvēlētajā vietā un kāda šajos procesos ir nozīme pārvadātājam. Kas ir jānodrošina pasažiera ērtībām un kādas loģistikas funkcijas ir nepieciešamas pašam pārvadātājam;
2. sastādīt programmatūras prasību specifikāciju un veikt programmatūras projektējumu;
3. izstrādāt programmatūras produktu, iespējams, mainīt, uzlabot, projektējumu;
4. veikt izstrādātā programmatūras produkta vienībtestēšanu
5. sagatavot kvalifikācijas darbam nepieciešamo dokumentāciju

Izmantotās metodes

Lai izpētītu pasažieru pārvadājumu specifiku, tika veikti novērojumi Rīgas Starptautiskajā Autoostā. Izpēte tika veikta gan no pasažiera, gan no pārvadātāja puses.

Vadoties no apkopotās informācijas tika sastādīta programmatūras prasību specifikācija un projektēta programmatūra.

Tika izvēlēts veidot tīmekļa bāzētu sistēmu un izstrādāt to izmantojot PHP un JavaScript tehnoloģijas, Apache tīmekļa serveri un MySQL datu bāzi.

Faktoloģiskā materiāla avoti

Programmatūrs prasību specifikācija tika sastādīta atbilstoši valsts standartam LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis” un programmatūras projektējuma apraksts atbilstoši LVS 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”. Kvalifikācijas darba pareizai noformēšanai tika izmantots Latvijas Universitātes noteikumi „Noslēguma darbu izstrādāšanas un aizstāvēšanas kārtība”. Jautājumos par programmatūras produkta izstrādi atbildes tika meklētas tīmeklī.

Darba struktūra

Darbs satur izstrādāto programmatūras prasību specifikāciju, programmatūras projekta aprakstu, testēšanas dokumentācijas fragmentus un dokumentāciju par izstrādes projekta organizāciju, izmantotajiem kvalitātes nodrošināšanas un konfigurāciju pārvaldības paņēmieniem un darbietilpības novērtējumu. Darbā iekļauts arī reprezentatīvs izstrādātā programmatūras produkta pirmā modeļa fragments.

1. NODAĻA

PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Programmatūras prasību specifikācijas (PPS) nolūks ir precīzi aprakstīt pasūtītāja un piegādātāja idejas par topošā programmatūras produkta prasībām. PPS mērķis ir aprakstīt to, kas programmatūrai ir jādara, neaprakstot, kā tas ir jādara.

Šis dokuments ir paredzēts pasūtītājam, lai dotu iespēju kontrolēt izstrādi un pārliecināties, ka gala sistēma atbildīs sākotnēji norunātajām prasībām un ietvers sākotnēji norunāto funkcionalitāti. Izstrādājamajā sistēmā tiks realizētas tikai un vienīgi tās prasības, kas iekļautas šajā dokumentā.

1.1.2. Darbības sfēra

Dotais dokuments specificē prasības attiecībā uz Biļešu tirdzniecības un reisu uzskaites sistēmas (tālāk šajā dokumentā: sistēma) programmatūras produktu.

Sistēmai jānodrošina pamata funkcijas, kas saistītas ar autobusu biļešu tirdzniecību un pasažieru pārvadājumu loģistiku.

Dotu programmatūras produktu var pielietot autoostas un uzņēmumi, kas nodarbojas ar pasažieru pārvadājumiem.

1.1.3. Definīcijas, akronīmi un saīsinājumi

Akronīms, saīsinājums	Paskaidrojums
DB	Datu bāze
ID	Datu bāzes objekta eksemplāra identifikators
OS	Operētājsistēma
Definīcija	Paskaidrojums
lietotājs	Autoostas vai uzņēmuma darbinieks, kas strādā ar sistēmu
pietura	Apdzīvota vieta vai nosaukums

maršruts	Vairāku pieturu apvienojums
reiss	Ar reisu saprotama autobusa kursēšana pa noteiktu maršrutu ar konkrētu atiešanas datumu un laiku
sākuma stacija	Maršruta sākumpietura, no kuras sākas maršruts
beigu pietura	Pietura maršrutā, līdz ar kuru beidzas pasažiera izvēlētais brauciens.
lietotāja lomas identifikators	Mainīgais, kas norāda lietotāja piederību kādai no PPS 1.2.3 punktā aprakstītajām lomām.
kustības grafika kods	Simbolu virkne, kas reprezentē nedēļas dienas, kurās kursē reiss, piemēram, 12345—nozīmē, ka reiss kursē tikai nedēļas dienās no pirmdienas līdz piektdienai un brīvdienās nekursē.
reisa ietilpība	Cilvēku skaits, ko konkrētais reiss var pārvest no sākuma stacijas līdz pēdējai pieturai maršrutā.
reisa spēkā stāšanās datums	Datums no kura reiss kursē konkrētajā maršrutā.
reisa darbības beigu datums	Datums līdz kuram reiss kursē konkrētajā maršrutā.
platforma	Vieta, kur maršruta sākumpunktā pasažieri gaida pienākam autobusu.
autobusa pienākšanas laiks	Laika vērtība, kas norāda cikoss autobuss piestāj pie maršruta pieturas.
braukšanas maksa	Naudas vērtība latos, kas norāda cik pasažierim jāmaksā par braucienu no sākuma pieturas līdz vēlamajai pieturai maršrutā.
vietu tips	Vietas tipējas brīvajās un rezervētajās.
biļešu drukas veids	Biļešu druka pa vienai vai grupas biļetes druka.
maršruta identifikators	Skaitlis, 1-4 simboli, kas apzīmē maršruta ID iekš DB.
reisa identifikators	Skaitlis, 9-12 simboli, kas veidojas sakonkatenējot maršruta identifikatoru ar datumu, apzīmē reisa ID iekš DB.
biļetes identifikators	Skaitlis, 10-14 simboli, kas veidojas sakonkatenējot maršruta identifikatoru ar datumu un ar vietas numuru, apzīmē biļetes ID iekš DB.
rezervācijas parole	Pasažiera nosaukta parole vietas rezervācijai.

1.1.4. Saistība ar citiem dokumentiem

Dokumenta sastādīšanā izmantots standarts LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis”, kas garantē dokumenta standartu atbilstību.

1.1.5. Dokumenta pārskats

Dokuments sastāv no 4 nodaļām:

1. Ievads. Ievadinformācija, kas satur dokumenta nolūku, identificē programmatūras projektu, nepieciešamos skaidrojumus, saistību ar citiem dokumentiem;
2. Vispārējs apraksts. Izklāstīta informācija par produkta funkcijām, lietotāja raksturierzīmēm, vispārējiem ierobežojumiem, pieņēmumiem un atkarībām;
3. Funkcionālās prasības. Detalizēti aprakstīti funkciju mērķi, ievaddati, apstrāde, izvaddati un kļūdu paziņojumi;
4. Nefunkcionālās prasības. Informācija ar datu drošību un sistēmas pieejamību.

1.2. Vispārējs apraksts

1.2.1. Produkta perspektīva

Dotais programmatūras produkts ir pilnīgi pašpietiekams, neatkarīgs no citiem produktiem, lietojams jebkurā autoostā vai uzņēmumā, kas nodarbojas ar pasažieru pārvadājumiem, kur sistēmas nodrošinātais funkciju kopums ir pietiekams.

1.2.1 Produkta funkcijas

Sistēma paredzēta gan lietošanai tīmeklī, gan lietošanai uzņēmumā. Tīmeklī publiskai apskatei pieejams reisu kustības saraksts. Sistēmas lietošana uzņēmumā paredz darbības saistītas ar biļešu tirdzniecību un sistēmas loģistiku un, lai tām piekļūtu, tās lietotājam jāpilnvarojas.

Tīmeklī pieejamās funkcijas:

- reisu kustības saraksta apskate.

Tikai uzņēmuma darbiniekiem pieejamās funkcijas:

- lietotāju pārvaldība

- pieturu pārvaldība
- maršrutu pārvaldība
- pilnvarošanās
- biļešu pārdošana un atpakaļpieņemšana
- biļešu rezervācija un rezervācijas atcelšana

1.2.2 Lietotāju raksturiezīmes

Sistēmas lietotāji iedalās divās grupās: tīmekļa vietnes apmeklētājos un pilnvarotajos sistēmas lietotājos.

Tīmekļa vietnes apmeklētāji ir jebkuri interesenti, kas vēlas uzzināt aktuālo informāciju par reisu kustības sarakstu. Pilnvarotie sistēmas lietotāji ir autoostu vai uzņēmumu, kas nodarbojas ar pasažieru pārvadājumiem, darbinieki. Atkarībā no pieejamajām funkcijām, Sistēmas lietotāji dalās sekojošās lomās:

Loma	Pieejamās funkcijas
0. Tīmekļa vietnes apmeklētājs	Reisu kustības saraksta apskate uzņēmuma tīmekļa vietnē.
1. Kasieris	Pilnvarošanās; Biļetes pārdošana un atpakaļpieņemšana.
2. Izziņu sniedzējs	Pilnvarošanās; Biļetes rezervācija un biļetes rezervācijas atcelšana.
3. Administrators	Pilnvarošanās; Lietotāja izveide; Lietotāja datu maiņa; Lietotāja dzēšana; Pieturas izveide; Pieturas datu maiņa; Pieturas dzēšana; Maršruta izveide; Maršruta datu maiņa; Maršruta dzēšana.

(1. tabula. Lietotāju iedalījums lomās.)

Lietotājiem tiek izvirzītas šādas prasības: tīmekļa vietnes apmeklētājiem - pamatzināšanas darbā ar datoru un tīmekļa pārlūkprogrammām; pilnvarotajiem lietotājiem - pamatzināšanas darbā ar datoru un tīmekļa pārlūkprogrammām, prasme lietot drukas iekārtas – atkarībā no konkrētās lomas.

1.2.3 Vispārējie ierobežojumi

1.2.3.1 Tīkls

Sistēmas tīmekļa vietnes lietošanai ir nepieciešams interneta pieslēgums ar minimāli 256 kilobitu sekundē datu pārraides ātrumu, lai lietotāji varētu pieslēgties pie sistēmas un ērti iegūt ielādējamās lapas saturu un saņemt atbildi no servera. Uzņēmumā ir nepieciešams kvalitatīvs, gan ātruma, gan pārraides kvalitātes, gan drošības ziņā lokālais tīkls. Vienīgā vajadzība izejai ārpus vietējā tīkla būs nepieciešama sistēmas uzstādīšanai vairākās autoostās vai uzņēmumos, kas vedīs pie DB servera centralizēšanas.

1.2.3.2 OS atbalsts un programmatūra

Sistēma tiek atbalstīta jebkuras operētājsistēmas vidē, uzstādīšanai nepieciešams datubāzes serveris (piemēram, MySQL), tīmekļa serveris (piemēram, Apache), PHP skriptēšanas valodas atbalsts. Tīmekļa vietnei jābūt labi attēlojamai populārākajās tīmekļa pārlūkprogrammās. Uzņēmumā Sistēmas lietošanai tiks izmantota Mozilla Firefox pārlūkprogramma.

1.2.3.3 Aparatūra

Sistēmas efektīvai darbībai nepieciešams serveris, kas spētu vienlaicīgi apstrādāt vismaz 1000 datu bāzes ierakstus laikā, kas mazāks par 1 sekundi un cietais disks, kurā varētu glabāties ne tikai operētājsistēma ar atbalstošo programmatūru, bet arī datu bāzes ieraksti par vismaz pusgadu ilgu laika posmu.

1.2.4 Pieņēmumi un atkarības

1.2.4.1 Izziņas par reisu kustības sarakstu

Lai sniegtu ziņas par reisu kustību sarakstu, nepieciešams, lai būtu automātiski pieejama aktuālā informācija par visiem reisiem.

1.2.4.2 Reisa laika norāde

Sistēma laika skaitlisko vērtību atspoguļo „vietējā” vērtībā, tas ir uzstādītajā servera laika joslā un vērtībā. Laiks, kas attēlots reisu grafikos, nav atkarīgs no laika pārejas uz vasaras vai ziemas vērtību, bet paliek nemainīgs.

1.3 Funkcionālās prasības

1.3.1 Lietotāju pārvaldība

1.3.1.1 Jauna lietotāja izveide

1.3.1.1.1 Mērķis:

Lietotāja loma: administrators.

Funkcija ir nepieciešama lietotāja izveidei un attiecīgās lietotāja lomas nodalīšanai.

1.3.1.1.2 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- lietotājvārds, 1-20 simboli;
- parole, 5-20 simboli;
- lietotāja vārds, 1-20 simboli;
- lietotāja uzvārds, 1-20 simboli;
- lietotāja kategorijas identifikators.

1.3.1.1.3 Apstrāde:

Sistēma pārbauda:

- vai visi lauki aizpildīti;
- vai lietotājs ar šādu lietotājvārdu jau DB.

Dati tiek saglabāti DB.

1.3.1.1.4 Izvaddati:

Sistēma pazino par veiksmīgu lietotāja izveidi vai kļūdu.

1.3.1.1.5 Kļūdu paziņojumi:

Sistēma paziņo:

- kuri lauki ir aizpildīti nepareizi;
- ja lietotājs ar šādu lietotājevārdu jau ir DB.
- ja neizdevās izveidot lietotāju.

1.3.1.2 Lietotāja datu maiņa

1.3.1.2.1 Mērķis:

Lietotāja loma: administrators.

Funkcija ir nepieciešama, lai mainītu sistēmā esoša lietotāja datus.

1.3.1.2.2 Ievaddati:

Lietotājs izvēlas, kura lietotāja datus vēlas mainīt un ievada laukos jauno informāciju, atbilstoši PPS 1.3.2.1.2 prasībā uzstādītajam formātam. Vienīgais lauks, kura saturu nedrīkst mainīt ir lietotājevārds.

1.3.1.2.3 Apstrāde:

Sistēma atlasa no DB visus ievadītos lietotājus.

Veicot lietotāja datu maiņu, sistēma pārbauda, vai laukos ievadītā informācija atbilst

1.3.1.1.2 prasībā uzstādītajam formātam.

Dati tiek saglabāti DB.

1.3.1.2.4 Izvaddati:

Sistēma izvada visus DB ievadītos lietotājus.

Pēc konkrēta lietotāja izvēles, sistēma izvada laukus jaunās informācijas ievadei.

Sistēma paziņo par veiksmīgu lietotāja datu maiņu vai kļūdu.

1.3.1.2.5 Kļūdu paziņojumi:

Sistēma paziņo:

- kuri lauki aizpildīti nepareizi.
- ja neizdevās mainīt lietotāja datus.

1.3.1.3 Lietotāja dzēšana

1.3.1.3.1 Mērķis:

Lietotāja loma: administrators.

Funkcija nepieciešama lietotāja dzēšanai no sistēmas.

1.3.1.3.2 Ievaddati:

Lietotājs izvēlas, kuru lietotāju vēlas dzēst.

Apstiprinājums par ieraksta dzēšanu, apstiprinu/neapstiprinu.

1.3.1.3.3 Apstrāde:

Sistēma atlasa no DB visus ievadītos lietotājus.

Veicot lietotāja ieraksta dzēšanu tiek izsaukts logs ar pamudinājumu apstiprināt vēlmi dzēst lietotāju.

DB tiek dzēsts lietotāja ieraksts.

1.3.1.3.4 Izvaddati:

Sistēma izvada visus DB ievadītos lietotājus.

Sistēma paziņo par veiksmīgu lietotāja dzēšanu vai kļūdu.

1.3.1.3.5 Kļūdu paziņojumi:

Sistēma paziņo, ja neizdevās dzēst lietotāju.

1.3.2 Pieturu pārvaldība

1.3.2.1 Pieturas izveide

1.3.2.1.1 Mērķis:

Lietotāja loma: administrators.

Funkcija ir nepieciešama jaunas pieturas izveidei.

1.3.2.1.2 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- pieturas nosaukums, 1-30 simboli;
- atzīme, vai tā ir sākuma pietura.

1.3.2.1.3 Apstrāde:

Sistēma pārbauda:

- vai visi lauki aizpildīti;
- vai pietura ar šādu nosaukumu jau nav DB.

Dati tiek saglabāti DB.

1.3.2.1.4 Izvaddati:

Sistēma paziņo par veiksmīgu pieturas izveidi vai kļūdu.

1.3.2.1.5 Kļūdu paziņojumi:

Sistēma paziņo:

- kuri lauki ir aizpildīti nepareizi;
- ja pietura ar šādu nosaukumu jau ir DB.
- ja neizdevās izveidot pieturu.

1.3.2.2 Pieturas datu maiņa

1.3.2.2.1 Mērķis:

Lietotāja loma: administrators.

Funkcija ir nepieciešama pieturas datu maiņai.

1.3.2.2.2 Ievaddati:

Lietotājs izvēlas, kuras pieturas datus vēlas mainīt un ievada laukos jauno informāciju, atbilstoši PPS 1.3.2.1.2 prasībā uzstādītajam formātam.

1.3.2.2.3 Apstrāde:

Sistēma atlasa no DB visas ievadītās pieturas.

Veicot pieturas datu maiņu, sistēma pārbauda, vai laukos ievadītā informācija atbilst

1.3.2.1.2 prasībā uzstādītajam formātam.

Sistēma pārbauda vai pietura ar šādu nosaukumu jau nav DB.

Dati tiek saglabāti DB.

1.3.2.2.4 Izvaddati:

Sistēma paziņo par veiksmīgu datu maiņu vai kļūdu.

1.3.2.2.5 Kļūdu paziņojumi:

Sistēma paziņo:

- kuri lauki ir aizpildīti nepareizi;
- ja pietura ar šādu nosaukumu jau ir DB.
- ja neizdevās mainīt pieturas datus.

1.3.2.3 Pieturas dzēšana

1.3.2.3.1 Mērķis:

Lietotāja loma: administrators.

Funkcija nepieciešama pieturas dzēšanai no sistēmas.

1.3.2.3.2 Ievaddati:

Lietotājs izvēlas, kuru pieturu vēlas dzēst.

Apstiprinājums par ieraksta dzēšanu, apstiprinu/neapstiprinu.

1.3.2.3.3 Apstrāde:

Sistēma atlasa no DB visas ievadītās pieturas.

Sistēma pārbauda, vai pietura nav saistīta ar kādu no maršrutiem.

Veicot pieturas ieraksta dzēšanu tiek izsaukts logs ar pamudinājumu apstiprināt vēlmi dzēst pieturu.

DB tiek dzēsts pieturas ieraksts.

1.3.2.3.4 Izvaddati:

Sistēma izvada visas DB ievadītās pieturas.

Sistēma paziņo par veiksmīgu pieturas dzēšanu vai kļūdu.

1.3.2.3.5 Kļūdu paziņojumi:

Sistēma paziņo:

- ja pietura ir saistīta ar kādu no maršrutiem;
- ja neizdevās dzēst pieturu.

1.3.3 Maršrutu pārvaldība

1.3.3.1 Jauna maršruta izveide

1.3.3.1.1 Mērķis:

Lietotāja loma: administrators.

Funkcija nepieciešama jauna maršruta izveidei.

1.3.3.1.2 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- maršruta nosaukums, 1-100 simboli;
- kustības grafika kods;
- reisa ietilpību;
- reisa spēkā stāšanās datumu;

- reisa darbības beigu datumu;
- platformas numuru;
- pievienot maršrutam sākuma staciju;
- pievienot maršrutam pieturas no DB;
- katrai pievienotajai pieturai norādīt:
 - autobusa pienākšanas laiku;
 - pieturas numurs maršrutā pēc kārtas (numurācija sākas ar skaitli 1 maršruta sākuma stacijai);
 - braukšanas maksu;

1.3.3.1.3 Apstrāde:

Sistēma pārbauda:

- vai visi lauki aizpildīti;
- vai maršruts ar šādu nosaukumu jau nav DB.

Dati tiek saglabāti DB.

1.3.3.1.4 Izvaddati:

Sistēma paziņo par veiksmīgu maršruta izveidi vai kļūdu.

1.3.3.1.6 Kļūdu paziņojumi:

Sistēma paziņo:

- kuri lauki ir aizpildīti nepareizi;
- ja maršruts ar šādu nosaukumu jau ir DB;
- ja maršrutam nav pievienota vismaz viena pietura;
- ja neizdevās izveidot maršrutu.

1.3.3.2 Maršruta datu rediģēšana

1.3.3.2.1 Mērķis:

Lietotāja loma: administrators.

Funkcija ir nepieciešama maršruta datu maiņai.

1.3.3.2.2 Ievaddati:

Lietotājs izvēlas, kura maršruta datus vēlas mainīt un:

- pēc izvēles pievieno vai atņem no maršruta pieturas;
- ievada laukos jauno informāciju, atbilstoši PPS 1.3.3.1.2 prasībā uzstādītajam formātam.

1.3.3.2.3 Apstrāde:

Sistēma pārbauda:

- vai visi lauki aizpildīti;
- vai maršruts ar šādu nosaukumu jau nav DB.

Dati tiek saglabāti DB.

1.3.3.2.4 Izvaddati:

Sistēma paziņo par veiksmīgu maršruta datu maiņu vai kļūdu.

1.3.3.2.5 Kļūdu paziņojumi:

Sistēma paziņo:

- kuri lauki ir aizpildīti nepareizi;
- ja maršruts ar šādu nosaukumu jau ir DB;
- ja maršrutam nav pievienota vismaz viena pietura;
- ja neizdevās mainīt maršruta datus.

1.3.3.3 Maršruta dzēšana

1.3.3.3.1 Mērķis:

Lietotāja loma: administrators.

Funkcija nepieciešama maršruta dzēšanai no sistēmas.

1.3.3.3.2 Ievaddati:

Lietotājs izvēlas, kuru maršrutu vēlas dzēst.

Apstiprinājums par ieraksta dzēšanu, apstiprinu/neapstiprinu.

1.3.3.3.3 Apstrāde:

Sistēma atlasa no DB visus ievadītos maršrutus.

Veicot maršruta ieraksta dzēšanu tiek izsaukts logs ar pamudinājumu apstiprināt vēlmi dzēst maršrutu.

DB tiek dzēsts maršruta ieraksts.

1.3.3.3.4 Izvaddati:

Sistēma izvada visus DB ievadītos maršrutus.

Sistēma paziņo par veiksmīgu maršruta dzēšanu vai kļūdu.

1.3.3.3.5 Kļūdu paziņojumi:

Sistēma paziņo, ja neizdevās dzēst maršrutu.

1.3.4 Lietotāju pilnvarošanās

1.3.4.1 Lietotāja pieteikšanās

1.3.4.1.1 Mērķis:

Lietotāja loma: kasieris, izziņu sniedzējs, administrators.

Funkcija nepieciešama, lai ļautu lietotājam piekļuvi sistēmai.

1.3.4.1.2 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- lietotājvārds atbilstoši PPS 1.3.1.1.2 prasībā uzstādītajam formātam;
- parole atbilstoši PPS 1.3.1.1.2 prasībā uzstādītajam formātam.

1.3.4.1.3 Apstrāde:

Sistēma pārbauda:

- vai lietotājs ar attiecīgo lietotājvārdu ir DB;
- vai lietotāja ievadītā parole sakrīt ar attiecīgā lietotāja paroli DB.

1.3.4.1.4 Izvaddati:

Sistēma paziņo par veiksmīgu pieteikšanos un ļauj piekļuvi lietotāja lomai atbilstošajām funkcijām vai paziņo par kļūdu.

1.3.4.1.5 Kļūdu paziņojumi:

Sistēma paziņo:

- ja lietotājs ar ievadīto lietotājvārdu nav DB;
- ja lietotājs ievadījis neatbilstošu paroli;
- ja neizdevās pieteikties.

1.3.4.2 Lietotāja atteikšanās

1.3.4.2.1 Mērķis:

Lietotāja loma: kasieris, izziņu sniedzējs, administrators.

Funkcija nepieciešama, lai lietotājs varētu beigt darbu ar sistēmu.

1.3.4.2.1 Ievaddati:

Lietotājs izvēlas beigt darbu.

Apstiprinājums par darba beigšanu, apstiprinu/neapstiprinu.

1.3.4.2.1 Apstrāde:

Sistēma pabeidz lietotāja sesiju.

1.3.4.2.1 Izvaddati:

Pēc veiksmīgas atteikšanās lietotājs atgriežas pieteikšanās lapā.

1.3.4.2.1 Kļūdu paziņojumi:

-

1.3.5 Biļešu pārdošana un atpakaļpieņemšana

1.3.5.1 Biļetes pārdošana

1.3.5.1.1 Mērķis:

Lietotāja loma: kasieris.

Funkcija nepieciešama biļetes pārdošanai pasažierim.

1.3.5.1.2 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- sākuma stacija;
- beigu pietura;
- reisa datums;
- reisa atiešanas laiks.

1.3.5.1.3 Apstrāde:

Sistēma pārbauda vai ir aizpildīti lauki:

- sākuma stacija;
- beigu pietura;
- reisa datums;
- reisa atiešanas laiks.

Sistēma atlasa no DB maršrutus, kas no sākuma stacijas caur beigu pieturu pastāv norādītajā datumā ar atiešanas laiku lielāku par ievadīto.

1.3.5.1.4 Izvaddati:

Sistēma izvada visus DB maršrutus, kas atbilst atlases kritērijiem vai paziņo par kļūdu.

1.3.5.1.5 Kļūdu paziņojumi:

Sistēma paziņo:

- kuri lauki ir aizpildīti nepareizi;
- ja nekas netika atlasīts.

1.3.5.1.6 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- konkrēta maršruta izvēle;
- iespēja izvēlēties citu beigu pieturu konkrētajā maršrutā.

1.3.5.1.7 Apstrāde:

Sistēma fiksē izvēlēto maršrutu vai fiksē izvēlēto maršrutu un atlasa no DB visas konkrētā maršruta pieturas un fiksē izvēlēto beigu pieturu.

Sistēma pārbauda, vai DB reisa ieraksts jau ir izveidots un, ja nav, izveido to.

Sistēma atlasa brīvo un rezervēto vietu skaitu konkrētajā reisā.

1.3.5.1.8 Izvaddati:

Sistēma izvada brīvo un rezervēto vietu izvēlni.

1.3.5.1.9 Kļūdu paziņojumi:

Sistēma paziņo:

- ja uz konkrēto reisu nav brīvu vietu;
- ja uz konkrēto reisu nav rezervētu vietu;
- ja neizdevās izvadīt brīvo un rezervēto vietu izvēlni.

1.3.5.1.10 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- vai biļete tiks pārdota no brīvajām, vai no rezervētajām vietām.

1.3.5.1.11 Apstrāde:

Sistēma atlasa no DB brīvo vietu numurus vai rezervēto vietu numurus un rezervācijas paroles konkrētajam reisam.

1.3.5.1.12 Izvaddati:

Sistēma izvada vietu numuru izvēlni un rezervācijas paroles, ja izvēlas vietas no rezervētajām.

1.3.5.1.13 Kļūdu paziņojumi:

Sistēma paziņo, ja neizdevās izvadīt vietu numuru izvēlni.

1.3.5.1.14 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt vietas vai vietu numurus.

1.3.5.1.15 Apstrāde:

Sistēma fiksē izvēlēto vietu numurus un neļauj tās izvēlēties citam lietotājam.

1.3.5.1.16 Izvaddati:

Sistēma izveido DB biļetes ierakstu vai ierakstus.

Sistēma izvada biļešu drukas veida izvēlni.

1.3.5.1.17 Kļūdu paziņojumi:

Sistēma paziņo:

- ja neizdevās izveidot DB biļetes ierakstu;
- ja neizdevās izvadīt biļešu drukas veida izvēlni.

1.3.5.1.18 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- vai drukāt biļetes pa vienai, vai drukāt grupas biļeti.

1.3.5.1.19 Apstrāde:

Sistēma sagatavo teksta failu drukas printerim ar sekojošu informāciju:

- biļetes identifikators;
- sākuma stacija;
- beigu pietura;
- reisa datums;
- reisa atiešanas laiks;
- maršruta nosaukums;
- vietas numurs vai numuri;
- pienākšanas laiks;
- pārdošanas datums;
- kasiera vārds un uzvārds.

Sistēma nosūta teksta failu uz drukas printeri.

1.3.5.1.20 Izvaddati:

Tiek izdrukāta biļete.

Sistēma sagatavojas jaunas biļetes pārdošanai.

1.3.5.1.21 Kļūdu paziņojumi:

Sistēma paziņo, ja:

- neizdevās sagatavot teksta failu drukas printerim;
- neizdevās drukāt biļeti;

1.3.5.2 Biļetes atpakaļpieņemšana

1.3.5.2.1 Mērķis

Lietotāja kategorija: kasieris.

Funkcija nepieciešama biļetes atpakaļpieņemšanai.

1.3.5.2.1 Ievaddati

Nepieciešams nodrošināt iespēju ievadīt biļetes identifikatoru.

Apstiprinājums par biļetes atpakaļpieņemšanu, apstiprinu/neapstiprinu.

1.3.5.2.1 Apstrāde

Sistēma pārbauda:

- vai ir aizpildīts lauks biļetes identifikators;
- vai DB eksistē biļetes ieraksts ar ievadīto ID;

1.3.5.2.1 Izvaddati

Sistēma atlasa no DB biļetes ierakstu.

Sistēma paziņo par veiksmīgu biļetes atpakaļpieņemšanu vai kļūdu.

1.3.5.2.1 Kļūdu paziņojumi

Sistēma paziņo:

- ja nav ievadīts biļetes identifikators;
- ja biļete ar šādu ID nav atrasta DB;
- ja neizdevās atpakaļpieņemt biļeti.

1.3.6 Biļetes rezervācija un rezervācijas atcelšana

1.3.6.1 Biļetes rezervācija

1.3.6.1.1 Mērķis:

Lietotāja loma: izziņu sniedzējs.

Funkcija nepieciešama biļetes rezervācijai.

1.3.6.1.2 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- sākuma stacija;
- beigu pietura;
- reisa datums;
- reisa atiešanas laiks.

1.3.6.1.3 Apstrāde:

Sistēma pārbauda vai ir aizpildīti lauki:

- sākuma stacija;
- beigu pietura;
- reisa datums;
- reisa atiešanas laiks.

Sistēma atlasa no DB maršrutus, kas no sākuma stacijas caur beigu pieturu pastāv norādītajā datumā ar atiešanas laiku lielāku par ievadīto.

1.3.6.1.4 Izvaddati:

Sistēma izvada visus DB maršrutus, kas atbilst atlasas kritērijiem vai paziņo par kļūdu.

1.3.6.1.5 Kļūdu paziņojumi:

Sistēma paziņo:

- kuri lauki ir aizpildīti nepareizi;
- ja nekas netika atlasīts.

1.3.6.1.6 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- konkrēta maršruta izvēle.

1.3.6.1.7 Apstrāde:

Sistēma fiksē izvēlēto maršrutu.

Sistēma pārbauda, vai DB reisa ieraksts jau ir izveidots un, ja nav, izveido to.

Sistēma atlasa brīvo vietu skaitu konkrētajā reisā.

1.3.6.1.8 Izvaddati:

Sistēma izvada vietu numuru izvēlni.

1.3.6.1.9 Kļūdu paziņojumi:

Sistēma paziņo:

- ja uz konkrēto reisu nav brīvu vietu;
- ja neizdevās izvadīt vietu numuru izvēlni.

1.3.6.1.10 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt

- vietas vai vietu numurus;
- rezervācijas paroli, 1-20 simboli.

1.3.6.1.11 Apstrāde:

Sistēma fiksē izvēlēto vietu numurus un neļauj tās izvēlēties citam lietotājam.

Sistēma pārbauda, vai ar šādu paroli jau nav rezervēta biļete.

1.3.6.1.12 Izvaddati:

Sistēma izveido DB biļetes ierakstu vai ierakstus.

Sistēma paziņo par veiksmīgu biļetes rezervāciju vai kļūdu.

1.3.6.1.13 Kļūdu paziņojumi:

Sistēma paziņo:

- ja ar šādu paroli jau ir rezervēta biļete;
- ja neizdevās izveidot DB biļetes ierakstu;
- ja neizdevās rezervēt biļeti.

1.3.6.2 Biļetes rezervācijas atcelšana

1.3.6.2.1 Mērķis

Lietotāja kategorija: izziņu sniedzējs.

Funkcija nepieciešama biļetes rezervācijas atcelšanai.

1.3.6.2.2 Ievaddati

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- rezervācijas parole;

Apstiprinājums par biļetes rezervācijas atcelšanu, apstiprinu/neapstiprinu.

1.3.6.2.3 Apstrāde

Sistēma pārbauda:

- vai paroles lauks ir aizpildīts;
- vai DB ir biļetes ieraksts ar attiecīgo paroli.

Sistēma atlasa informāciju par biļeti ar attiecīgo paroli.

Sistēma izdzēš biļetes DB ierakstu un uzstāda tās vietu vai vietas kā brīvas.

1.3.6.2.4 Izvaddati

Sistēma izvada biļetes DB ierakstu.

Sistēma paziņo par veiksmīgu biļetes rezervācijas atcelšanu vai kļūdu.

1.3.6.2.5 Kļūdu paziņojumi

Sistēma paziņo:

- ja nav aizpildīts paroles lauks;
- ja DB nav ieraksta ar attiecīgo rezervācijas paroli;
- ja neizdevās dzēst biļetes DB ierakstu;
- ja neizdevās atcelt biļetes rezervāciju.

1.3.7 Reisu kustības saraksta apskate

1.3.7.1 Mērķis:

Lietotāja loma: tīmekļa vietnes apmeklētājs.

Funkcija nepieciešama aktuālā reisu kustības saraksta apskatei.

1.3.7.2 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- sākuma stacija;
- beigu pietura;

- reisa datums;
- reisa atiešanas laiks no;
- reisa atiešanas laiks līdz.

1.3.7.3 Apstrāde:

Sistēma pārbauda vai ir aizpildīti lauki:

- sākuma stacija;
- beigu pietura;
- reisa datums;
- reisa atiešanas laiks no;
- reisa atiešanas laiks līdz.

Sistēma atlasa no DB maršrutus, kas no sākuma stacijas caur beigu pieturu pastāv norādītajā datumā ar atiešanas laiku lielāku par ievadīto.

1.3.7.4 Izvaddati:

Sistēma izvada visus DB maršrutus, kas atbilst atlasē kritērijiem vai paziņo par kļūdu.

1.3.7.5 Kļūdu paziņojumi:

Sistēma paziņo:

- kuri lauki ir aizpildīti nepareizi;
- ja nekas netika atlasīts.

1.3.7.6 Ievaddati:

Nepieciešams nodrošināt iespēju ievadīt sekojošu informāciju:

- konkrēta maršruta izvēle.

1.3.7.7 Apstrāde:

Sistēma fiksē izvēlēto maršrutu.

Sistēma atlasa no DB pieturas izvēlētajā maršrutā.

1.3.7.8 Izvaddati:

Sistēma izvada pieturu sarakstu konkrētajam maršrutam.

1.3.7.9 Kļūdu paziņojumi:

Sistēma paziņo:

- ja neizdevās izvadīt pieturu sarakstu.

1.4 Nefunkcionālās prasības

1.4.1 Datu drošība

Sistēma lielāko daļu lietotāja ievadīto datu pārbauda ar Javascript palīdzību, kas nodrošina uz serveri atnākošo datu pareizību. Servera pusē saņemtos datus sistēma apstrādā pret SQL injekcijām.

1.4.2 Pieejamība

Sistēmas tīmekļa vietnei jābūt pieejamai ikvienam interesentam reisu kustības saraksta apskatei. Pārējā sistēmas daļa pieejama tikai autoostas vai uzņēmuma, kas nodarbojas ar pasažieru pārvadājumiem, darbiniekiem.

2. NODAĻA

PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Nolūks

Programmatūras projektējuma apraksta nolūks ir aprakstīt, kā dotās sistēmas programmatūras prasību specifikācijā iekļautās prasības tiks realizētas programmatūras izstrādes vidē, lai atvieglotu programmatūras projektēšanu un izstrādi.

Dotais dokuments apraksta Biļešu tirdzniecības un reisu uzskaites sistēmas programmatūras projektējumu.

2.1.2. Darbības sfēra

Dotais dokuments apraksta programmatūras projektējumu attiecībā uz Biļešu tirdzniecības un reisu uzskaites sistēmas (tālāk šajā dokumentā: sistēma) programmatūras produktu.

Sistēmai jānodrošina pamata funkcijas, kas saistītas ar autobusu biļešu tirdzniecību un pasažieru pārvadājumu loģistiku.

Doto programmatūras produktu var pielietot autoostas un uzņēmumi, kas nodarbojas ar pasažieru pārvadājumiem.

2.1.3. Definīcijas, akronīmi, saīsinājumi

Akronīms, saīsinājums	Paskaidrojums
DB	Datu bāze
ID	Datu bāzes objekta eksemplāra identifikators
OS	Operētājsistēma
Definīcija	Paskaidrojums
lietotājs	Autoostas vai uzņēmuma darbinieks, kas strādā ar Sistēmu
pietura	Apdzīvota vieta vai nosaukums
maršruts	Vairāku pieturu apvienojums
reiss	Ar reisu saprotama autobusa kursēšana pa noteiktu maršrutu ar konkrētu

	atiešanas datumu un laiku
sākuma stacija	Maršruta sākumpietura, no kuras sākas maršruts
beigu pietura	Pietura maršrutā, līdz ar kuru beidzas pasažiera izvēlētais brauciens.
lietotāja lomas identifikators	Mainīgais, kas norāda lietotāja piederību kādai no PPS 1.2.3 punktā aprakstītajām lomām.
kustības grafika kods	Simbolu virkne, kas reprezentē nedēļas dienas, kurās kursē reiss, piemēram, 12345—nozīmē, ka reiss kursē tikai nedēļas dienās no pirmdienas līdz piektdienai un brīvdienās nekursē.
reisa ietilpība	Cilvēku skaits, ko konkrētais reiss var pārvest no sākuma stacijas līdz pēdējai pieturai maršrutā.
reisa spēkā stāšanās datums	Datums no kura reiss kursē konkrētajā maršrutā.
reisa darbības beigu datums	Datums līdz kuram reiss kursē konkrētajā maršrutā.
platforma	Vieta, kur maršruta sākumpunktā pasažieri gaida pienākam autobus.
autobusa pienākšanas laiks	Laika vērtība, kas norāda cik autobuss pietāj pie maršruta pieturas.
braukšanas maksa	Naudas vērtība latos, kas norāda cik pasažierim jāmaksā par braucienu no sākuma pieturas līdz vēlamojai pieturai maršrutā.
vietu tips	Vietas tipējas brīvajās un rezervētajās.
biļešu drukas veids	Biļešu druka pa vienai vai grupas biļetes druka.
maršruta identifikators	Skaitlis, 1-4 simboli, kas apzīmē maršruta ID iekš DB.
reisa identifikators	Skaitlis, 9-12 simboli, kas veidojas sakonkatenējot maršruta identifikatoru ar datumu, apzīmē reisa ID iekš DB.
biļetes identifikators	Skaitlis, 10-14 simboli, kas veidojas sakonkatenējot maršruta identifikatoru ar datumu un ar vietas numuru, apzīmē biļetes ID iekš DB.
rezervācijas parole	Pasažiera nosaukta parole vietas rezervācijai.

2.2. Saistība ar citiem dokumentiem

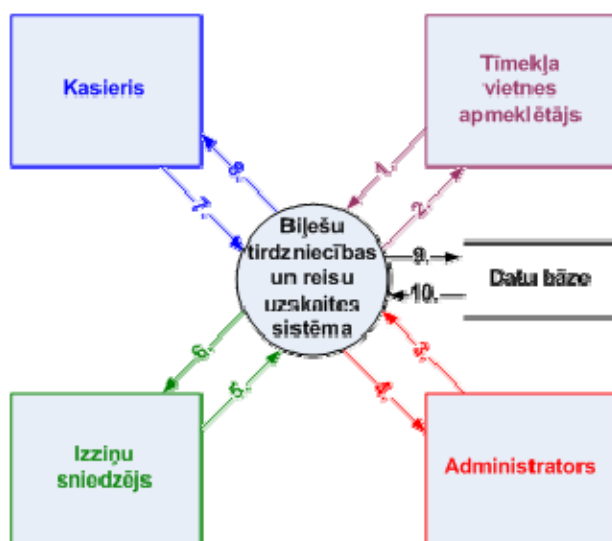
Dokumenta sastādīšanā izmantots standarts LVS 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”, kas garantē dokumenta standartu atbilstību.

2.3. Dekompozīcijas apraksts

Tabula	Apraksts
user	Tabulas mērķis ir glabāt sistēmas lietotāja pamatinformāciju.
route	Tabulas mērķis ir glabāt informāciju par maršrutiem.
stop	Tabulas mērķis ir glabāt informāciju par maršrutu pieturvietām.
route_stop	Tabulas mērķis ir apvienot pieturvietas maršrutos.
trip	Tabulas mērķis ir fiksēt informāciju par katru pa maršrutu kursējošo reisu.
ticket	Tabulas mērķis ir fiksēt informāciju par pārdotajām vai rezervētajām biļetēm.

2.4. Atkarības apraksts

2.4.1. Konteksta līmeņa datu plūsmas diagramma



(1. attēls. Konteksta līmeņa datu plūsmas diagramma)

1. Reisu kustības saraksta meklēšanas dati.
2. Reisu kustības saraksts.
3. Lietotāju, pieturu, maršrutu izveides, maiņas, dzēšanas dati.
4. Lietotāju, pieturu, maršrutu dati, apstiprinājumi par izmaiņām.
5. Biļetes rezervācijas, rezervācijas atcelšanas dati.
6. Biļetes rezervācija, rezervācijas atcelšanas apstiprinājums.
7. Biļetes pārdošanas, atpakaļpieņemšanas dati.
8. Biļete, biļetes atpakaļpieņemšanas apstiprinājums.
9. SQL vaicājumi.
10. SQL vaicājumu rezultāti.

2.4.2. Pirmā līmeņa datu plūsmas diagramma



(2. attēls. Pirmā līmeņa datu plūsmas diagramma)

Piezīme: pārskatāmības labad, lietotāju pilnvarošanās process diagrammā ir nodalīts pie katra lietotāja atsevišķi, taču pēc būtības ir kopējs visiem pilnvarotajiem lietotājiem.

- 1.1. Maršruta atlasē kritēriji, konkrēta maršruta izvēle.
- 1.2. Maršruta atlasē SQL vaicājums; pieturu atlasē SQL vaicājums.
- 2.1. Maršruta atlasē rezultāts; pieturu atlasē rezultāts.
- 2.2. Reisu kustības saraksts; konkrēta maršruta pieturu saraksts.
- 3.1. Pieteikšanās dati; atteikšanās pamudinājums.
- 3.2. Pieteikšanās SQL vaicājums.
- 4.1. Lietotāja dati.
- 4.2. Tiesības darbam ar sistēmu; atteikšanās apstiprinājums.
- 3.3. Lietotāja izveides dati; lietotāja datu maiņas dati; lietotāja dzēšanas pamudinājums.
- 3.4. Lietotāja izveides SQL vaicājums; lietotāja datu maiņas SQL vaicājums; lietotāja

dzēšanas SQL vaicājums.

4.3. Lietotāju atlases rezultāts; paziņojums par izmaiņām DB.

4.4. Lietotāju saraksts; paziņojums par lietotāja izveidi; paziņojums par lietotāja datu maiņu, paziņojums par lietotāja dzēšanu.

3.5. Pieturas izveides dati; pieturas datu maiņas dati; pieturas dzēšanas pamudinājums.

3.6. Pieturas izveides SQL vaicājums; pieturas datu maiņas SQL vaicājums; pieturas dzēšanas SQL vaicājums.

4.5. Pieturu atlases rezultāts; paziņojums par izmaiņām DB.

4.6. Pieturu saraksts; paziņojums par pieturas izveidi; paziņojums par pieturas datu maiņu, paziņojums par pieturas dzēšanu.

3.7. Maršruta izveides dati; maršruta datu maiņas dati; maršrutam pievienojamo, dzēšamo pieturu dati; maršruta dzēšanas pamudinājums.

3.8. Maršruta izveides SQL vaicājums; maršruta datu maiņas SQL vaicājums; maršruta dzēšanas SQL vaicājums.

4.7. Maršrutu atlases rezultāts; pieturu atlases rezultāts; paziņojums par izmaiņām DB.

4.8. Maršrutu saraksts; pieturu saraksts; paziņojums par maršruta izveidi; paziņojums par maršruta datu maiņu, paziņojums par maršruta dzēšanu.

5.1. Pieteikšanās dati; atteikšanās pamudinājums.

5.2. Pieteikšanās SQL vaicājums.

6.1. Lietotāja dati.

6.2. Tiesības darbam ar sistēmu; atteikšanās apstiprinājums.

5.3. Maršruta atlases kritēriji, konkrēta maršruta izvēle, vietu numuru izvēle, rezervācijas parole.

5.4. Maršruta atlases SQL vaicājums; vietu atlases SQL vaicājums, rezervācijas paroles ievades SQL vaicājums.

6.3. Maršruta atlases rezultāts; vietu atlases rezultāts; paroles ievades apstiprinājums.

6.4. Reisu kustības saraksts, rezervācijai pieejamo pieturu saraksts, rezervācijas apstiprinājums.

5.5. Rezervācijas parole.

5.6. Biļetes atlases SQL vaicājums; rezervācijas dzēšanas SQL vaicājums.

6.5. Biļetes atlases rezultāts; rezervācijas dzēšanas apstiprinājums.

6.6. Rezervācijas atcelšanas apstiprinājums.

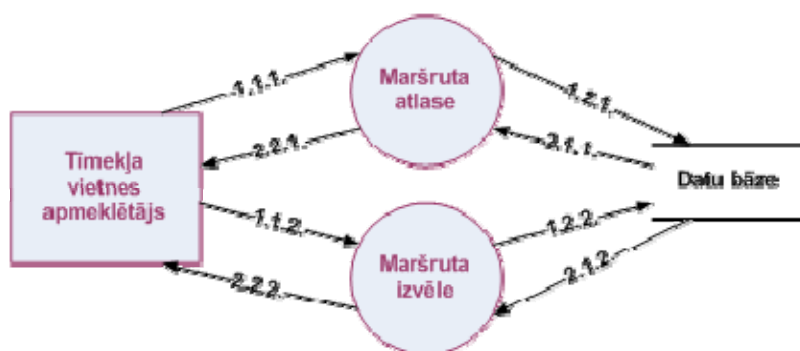
7.1. Pieteikšanās dati; atteikšanās pamudinājums.

7.2. Pieteikšanās SQL vaicājums.

8.1. Lietotāja dati.

- 8.2. Tiesības darbam ar sistēmu; atteikšanās apstiprinājums.
- 7.3. Maršruta atlasē kritēriji; konkrēta maršruta izvēle; citas beigu pieturas izvēle; vietu tipa izvēle; vietu numuru izvēle; drukas veida izvēle.
- 7.4. Maršruta atlasē SQL vaicājums; pieturu atlasē SQL vaicājums; pieejamo pārdošanai brīvo un rezervēto vietu saraksta atlasē SQL vaicājums; reisa izveides SQL vaicājums; biļetes izveides SQL vaicājums.
- 8.3. Maršruta atlasē rezultāts; pieturu atlasē rezultāts; pieejamo vietu atlasē rezultāts; reisa izveides apstiprinājums; biļetes izveides apstiprinājums.
- 8.4. Reisu kustības saraksts; konkrēta maršruta pieturu saraksts; pieejamo pārdošanai brīvo vai rezervēto vietu saraksts; biļetes pārdošanas apstiprinājums; biļetes drukas apstiprinājums.
- 7.5. Biļetes identifikators.
- 7.6. Biļetes atlasē SQL vaicājums; biļetes dzēšanas SQL vaicājums.
- 8.5. Biļetes atlasē rezultāts; biļetes dzēšanas apstiprinājums.
- 8.6. Biļetes atpakaļpieņemšanas apstiprinājums.

2.4.3. Otrā līmeņa datu plūsmas diagrammas



(3. attēls. Reisu kustības saraksta apskates datu plūsmas diagramma)

1.1.1. Sākuma stacija; beigu pietura; reisa datums; reisa atiešanas laiks no; reisa atiešanas laiks līdz.

1.2.1. No ievadītajiem parametriem sastādītais maršruta atlasē SQL vaicājums.

2.1.1. Maršruta atlasē SQL rezultāts.

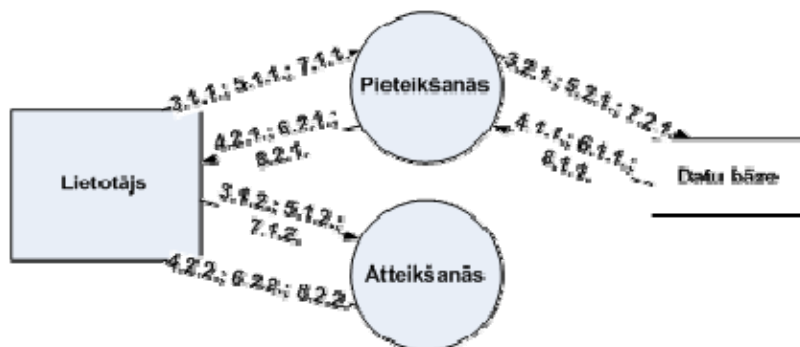
2.2.1. Par katru atlasīto maršrutu izvada: maršruta nosaukumu, reisa spēkā stāšanās datumu, reisa darbības beigu datumu, atiešanas laiku no sākuma stacijas, pienākšanas laiku ievadītajā beigu pieturā, kustības grafika kodu, reisa ietilpību, atiešanas platformas numuru, brauciena maksu līdz ievadītajai beigu pieturai.

1.1.2. Konkrēta maršruta izvēle.

1.2.2. Konkrētā maršruta pieturu atlasē SQL vaicājums.

2.1.2. Pieturu atlasē SQL rezultāts.

2.2.2. Par katru atlasīto pieturu izvada: pieturas numuru pēc kārtas, pieturas nosaukumu, pienākšanas laiku pieturā, braukšanas maksu līdz pieturai.



(4. attēls. Lietotāja pilnvarošanās datu plūsmas diagramma)

3.1.1.; 5.1.1.; 7.1.1. Lietotājvārds, parole.

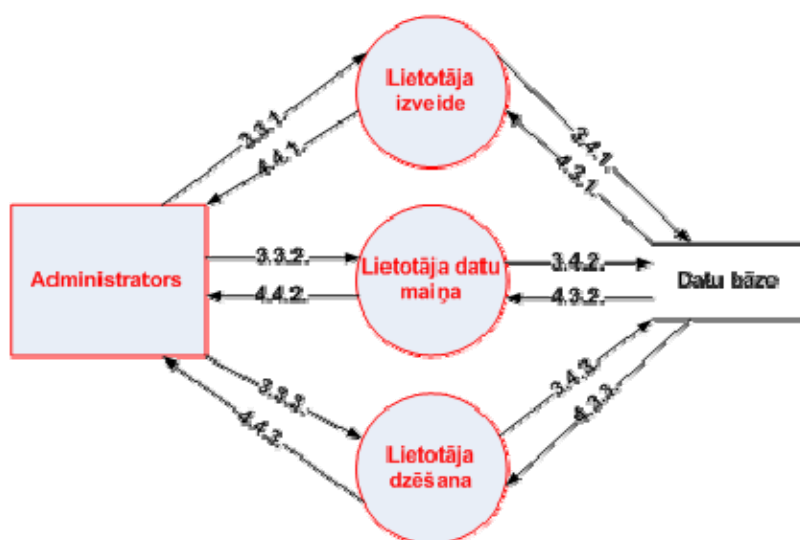
3.2.1.; 5.2.1.; 7.2.1.; No ievadītajiem datiem sastādītais lietotāja atlasē SQL vaicājums.

4.1.1.; 6.1.1.; 8.1.1. Lietotāja identifikators, vārds, uzvārds, loma.

4.2.1.; 6.2.1.; 8.2.1. Lomai atbilstošo tiesību pieskīšana.

3.1.2.; 5.1.2.; 7.1.2. Pamudinājums beigt darbu ar sistēmu.

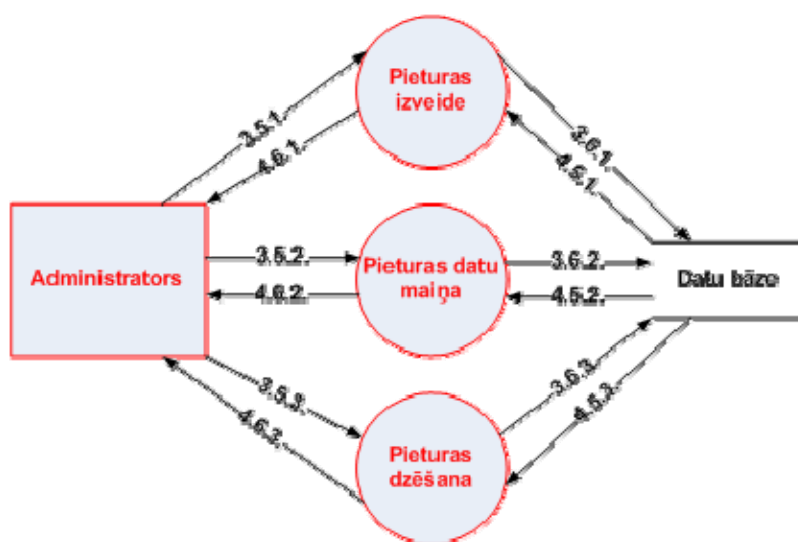
4.2.2.; 6.2.2.; 8.2.2. Lietotāja darba sesijas pabeigšana.



(5. attēls. Lietotāju pārvaldības datu plūsmas diagramma)

3.3.1. Lietotājvārds; parole; lietotāja vārds; lietotāja uzvārds; lietotāja kategorijas identifikators.

- 3.4.1. No ievadītajiem datiem sastādītais lietotāja izveides SQL vaicājums.
- 4.3.1. Paziņojums par izmaiņām DB.
- 4.4.1. Paziņojums par veiksmīgu lietotāja izveidi.
- 3.3.2. Konkrēta lietotāja izvēle; pēc izvēles tiek ievadīti jaunie dati: parole; lietotāja vārds; lietotāja uzvārds vai lietotāja kategorijas identifikators.
- 3.4.2. Lietotāju atlases SQL vaicājums; no ievadītajiem datiem sastādītais lietotāja datu maiņas SQL vaicājums.
- 4.3.2. Lietotāju atlases SQL rezultāts; paziņojums par izmaiņām DB.
- 4.4.2. Atlasītais lietotāju saraksts, par katru lietotāju: lietotāja ID, lietotājvārds, parole, lietotāja vārds, lietotāja uzvārds, lietotāja kategorijas identifikators; paziņojums par veiksmīgu lietotāja datu maiņu.
- 3.3.3. Konkrēta lietotāja izvēle; pamudinājuma apstiprināšana.
- 3.4.3. Lietotāju atlases SQL vaicājums.
- 4.3.3. Lietotāju atlases SQL rezultāts; paziņojums par izmaiņām DB.
- 4.4.3. Atlasītais lietotāju saraksts, par katru lietotāju: lietotāja ID, lietotājvārds, lietotāja vārds, lietotāja uzvārds, lietotāja kategorijas identifikators; vaicājums apstiprināt vai noraidīt pamudinājumu dzēst konkrēto lietotāju; paziņojums par veiksmīgu lietotāja dzēšanu.



(6. attēls. Pieturu pārvaldības datu plūsmas diagramma)

- 3.5.1. Pieturas nosaukums, atzīme, vai tā ir sākuma stacija.
- 3.6.1. No ievadītajiem datiem sastādītais pieturas izveides SQL vaicājums.
- 4.5.1. Paziņojums par izmaiņām DB.
- 4.6.1. Paziņojums par veiksmīgu pieturas izveidi.
- 3.5.2. Konkrēta pieturas izvēle; pēc izvēles tiek ievadīti jaunie dati: pieturas nosaukums;

atzīme, vai konkrētā pietura ir sākuma stacija.

3.6.2. Pieturu atlasē SQL vaicājums; no ievadītajiem datiem sastādītais pieturas datu maiņas SQL vaicājums.

4.5.2. Pieturu atlasē SQL rezultāts; paziņojums par izmaiņām DB.

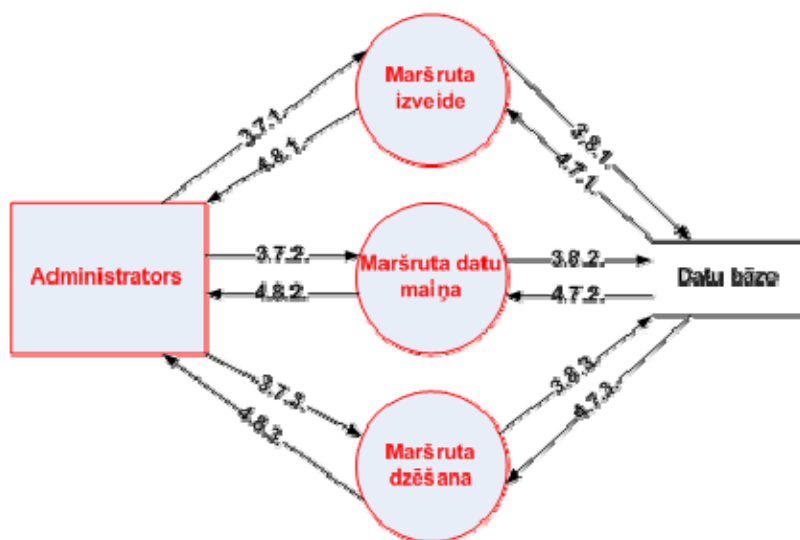
4.6.2. Atlasītais pieturu saraksts, par katru pieturu: pieturas ID, pieturas nosaukums, atzīme, vai konkrētā pietura ir sākuma stacija; paziņojums par veiksmīgu pieturas datu maiņu.

3.5.3. Konkrēta pieturas izvēle; pamudinājuma apstiprināšana.

3.6.3. Pieturu atlasē SQL vaicājums.

4.5.3. Pieturu atlasē SQL rezultāts; paziņojums par izmaiņām DB.

4.6.3. Atlasītais pieturu saraksts, par katru pieturu: pieturas ID, pieturas nosaukums, atzīme, vai konkrētā pietura ir sākuma stacija; vaicājums apstiprināt vai noraidīt pamudinājumu dzēst konkrēto pieturu; paziņojums par veiksmīgu pieturas dzēšanu.



(7. attēls. Maršrutu pārvaldības datu plūsmas diagramma)

3.7.1. Maršruta nosaukums; kustības grafika kods; reisa ietilpība; reisa spēkā stāšanās datums; reisa darbības beigu datums; atiešanas platformas numurs; pievienotās pieturas, par katru no tām: pieturas numurs maršrutā pēc kārtas, pienākšanas laiks pieturā, braukšanas maksa līdz pieturai.

3.8.1. Pieturu atlasē SQL vaicājums; no ievadītajiem datiem sastādītais maršruta izveides SQL vaicājums; no ievadītajiem datiem sastādītie pieturu pievienošanas SQL vaicājumi.

4.7.1. Pieturu atlasē SQL rezultāts. Paziņojums par izmaiņām DB.

4.8.1. Atlasītais pieturu saraksts, par katru pieturu: pieturas ID, pieturas nosaukums, atzīme, vai konkrētā pietura ir sākuma stacija; paziņojums par veiksmīgu maršruta izveidi.

3.7.2. Konkrēta maršruta izvēle; pēc izvēles tiek ievadīti jaunie dati: maršruta nosaukums;

kustības grafika kods; reisa ietilpība; reisa spēkā stāšanās datums; reisa darbības beigu datums; atiešanas platformas numurs; pieturas, par katru pēc ivēles tiek ievadīti jaunie dati: pieturas numurs maršrutā pēc kārtas, pienākšanas laiks pieturā, braukšanas maksa līdz pieturai.

3.8.2. . Maršrutu atlases SQL vaicājums; pieturu atlases SQL vaicājums; no ievadītajiem datiem sastādītais maršruta datu maiņas SQL vaicājums; no ievadītajiem datiem sastādītie maršrutā iekļauto pieturu datu maiņas SQL vaicājumi.

4.7.2. Maršrutu atlases SQL rezultāts; pieturu atlases SQL rezultāts. Paziņojums par izmaiņām DB.

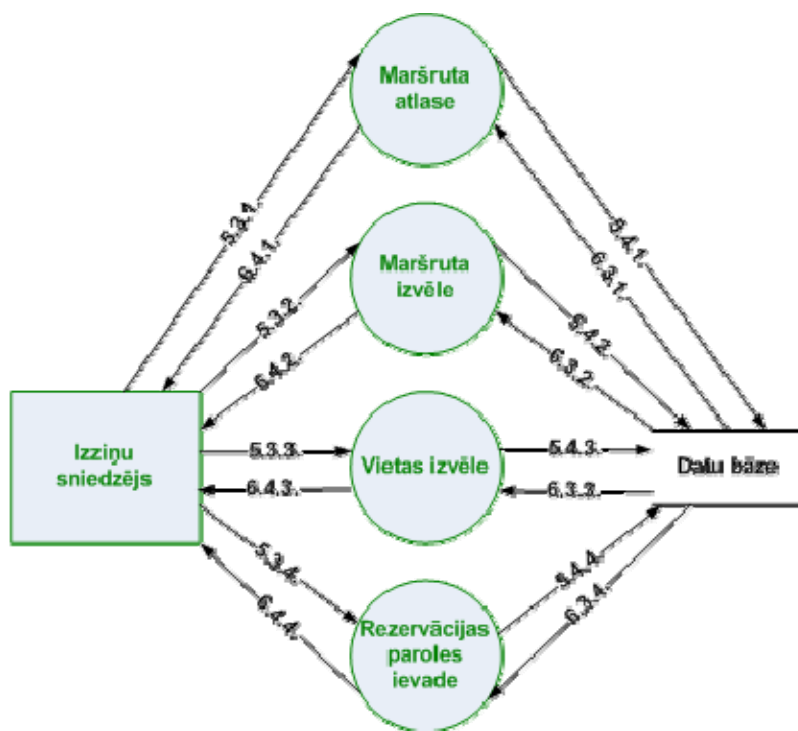
4.8.2. Atlasītais maršrutu saraksts, par katru maršrutu: maršruta nosaukums; kustības grafika kods; reisa ietilpība; reisa spēkā stāšanās datums; reisa darbības beigu datums; atiešanas platformas numurs; atlasītais pieturu saraksts, par katru pieturu: pieturas ID, pieturas nosaukums, atzīme, vai konkrētā pietura ir sākuma stacija; paziņojums par veiksmīgu maršruta datu maiņu.

3.7.3. Konkrēta maršruta izvēle; pamudinājuma apstiprināšana.

3.8.3. Maršrutu atlases SQL vaicājums; maršruta pieturu atlases SQL vaicājums.

4.7.3. Maršrutu atlases SQL rezultāts; maršruta pieturu atlases SQL rezultāts; paziņojums par izmaiņām DB.

4.8.3. Atlasītais maršrutu saraksts, par katru maršrutu: maršruta ID; maršruta nosaukums; kustības grafika kods; reisa ietilpība; reisa spēkā stāšanās datums; reisa darbības beigu datums; atiešanas platformas numurs; atlasītais pieturu saraksts, par katru pieturu: pieturas ID, pieturas nosaukums, atzīme, vai konkrētā pietura ir sākuma stacija; paziņojums par veiksmīgu maršruta dzēšanu.



(8. attēls. Biļetes rezervācijas datu plūsmas diagramma)

5.3.1. Sākuma stacija; beigu pietura; reisa datums; reisa atiešanas laiks no; reisa atiešanas laiks līdz.

5.4.1. No ievadītajiem parametriem sastādītais maršruta atlases SQL vaicājums.

6.3.1. Maršruta atlases SQL rezultāts.

6.4.1. Par katru atlasīto maršrutu izvada: maršruta nosaukumu, reisa spēkā stāšanās datumu, reisa darbības beigu datumu, atiešanas laiku no sākuma stacijas, pienākšanas laiku ievadītajā beigu pieturā, kustības grafika kodu, reisa ietilpību, atiešanas platformas numuru, brauciena maksu līdz ievadītajai beigu pieturai.

5.3.2. Konkrēta maršruta izvēle.

5.4.2. Konkrētā maršruta pieturu atlases SQL vaicājums; konkrētā maršruta brīvo vietu atlases SQL vaicājums.

6.3.2. Pieturu atlases SQL rezultāts; brīvo vietu atlases SQL vaicājums.

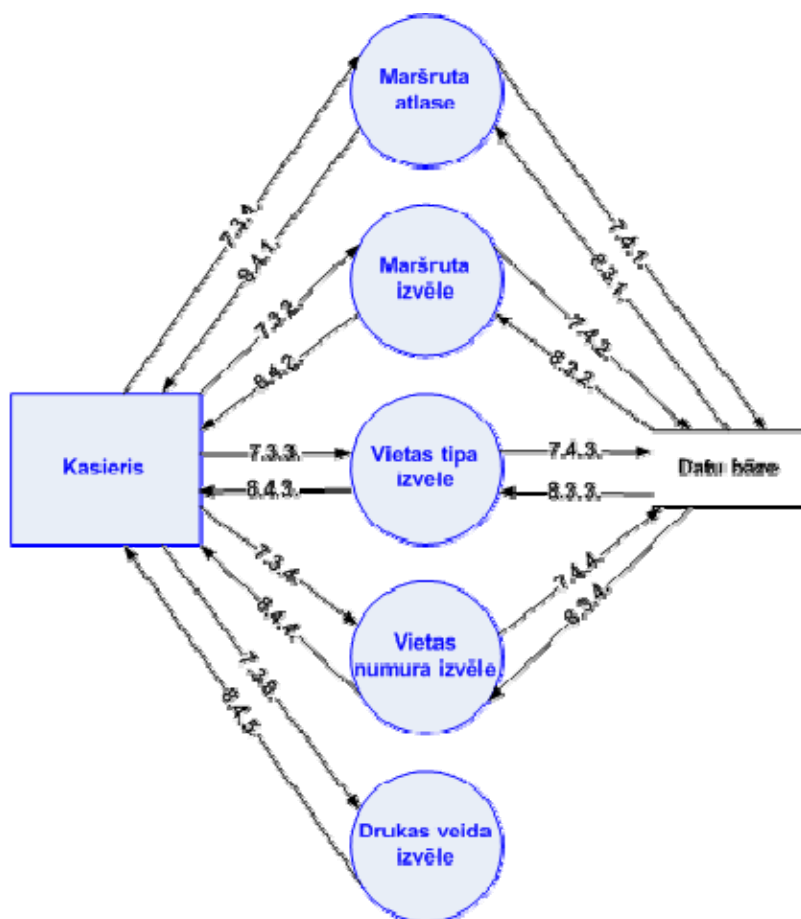
6.4.2. Par katru atlasīto pieturu izvada: pieturas numuru pēc kārtas, pieturas nosaukumu, pienākšanas laiku pieturā, braukšanas maksu līdz pieturai. Brīvo vietu saraksts.

5.3.3. Konkrētu vietu izvēle.

5.4.3. Sistēma neļauj citam lietotājam izvēlēties atzīmētās vietas – īslaicīgas vietu rezervācijas SQL vaicājums.

6.3.3. Paziņojums par veiksmīgu īslaicīgo rezervāciju. Lietotājs var turpināt rezervācijas procesu.

- 6.4.3. Paroles ievades lauks.
- 5.3.4. Rezervācijas parole.
- 5.4.4. Vietu rezervācijas SQL vaicājums.
- 6.3.4. Paziņojums ar izmaiņām DB.
- 6.4.4. Paziņojums par veiksmīgu biļetes rezervāciju.



(9. attēls. Biļetes pārdošanas datu plūsmas diagramma)

7.3.1. Sākuma stacija; beigu pietura; reisa datums; reisa atiešanas laiks no; reisa atiešanas laiks līdz.

7.4.1. No ievadītajiem parametriem sastādītais maršruta atlases SQL vaicājums.

8.3.1. Maršruta atlases SQL rezultāts.

8.4.1. Par katru atlasīto maršrutu izvada: maršruta nosaukumu, atiešanas laiku no sākuma stacijas, pienākšanas laiku ievadītajā beigu pieturā, brauciena maksu līdz ievadītajai beigu pieturai.

7.3.2. Konkrēta maršruta izvēle vai citas beigu pieturas maršrutā izvēle.

7.4.2. Konkrētā maršruta pieturu atlases SQL vaicājums; konkrētā maršruta brīvo un rezervēto vietu atlases SQL vaicājums.

- 8.3.2. Pieturu atlasē SQL rezultāts. Brīvo un rezervēto vietu atlasē rezultāts.
- 8.4.2. Vietas tipa izvēlne.
- 7.3.3. Vietas tipa izvēle.
- 7.4.3. Brīvo vai rezervēto vietu atlasē SQL vaicājums, atbilstoši izvēlētajam tipam.
- 8.3.3. Izvēlēta vietu tipa vietu atlasē SQL rezultāts.
- 8.4.3. Izvēlēta tipa pārdošanai pieejamo vietu saraksts.
- 7.3.4. Vietu numuru izvēle.
- 7.4.4. Sistēma neļauj citam lietotājam izvēlēties atzīmētās vietas – īslaicīgas vietu rezervācijas SQL vaicājums. Izvēlēto vietu apstiprināšanas gadījumā – pārdotās biļetes ieraksta izveides SQL vaicājums.
- 8.3.4. Paziņojums par veiksmīgu īslaicīgo rezervāciju; paziņojums par veiksmīgu biļetes noformēšanu.
- 8.4.4. Drukāšanas veida izvēlne.
- 7.3.5. Drukāšanas veida izvēle.
- 8.4.5. Sistēma sagatavo teksta failu atbilstoši izvēlētajam drukāšanas veidam un nosūta teksta failu uz drukāšanas printeri.

2.5. Detalizēts datu projektējums

2.5.1. Tabula „user”

Tabulas mērķis ir glabāt pamatinformāciju par sistēmas lietotāju.

Lauks	Lauka datu tips	Lauka tips	Var būt tukšs?	Ir unikāls?
userId	int(4)	PK	Nē	Jā
username	varchar(20)		Nē	Nē
password	varchar(20)		Nē	Nē
userRealName	varchar(20)		Nē	Nē
userSurname	varchar(20)		Nē	Nē
userRole	tinyint(1)		Nē	Nē

2.5.2. Tabula „stop”

Tabulas mērķis ir glabāt pamatinformāciju par maršrutu pieturvietām.

Lauks	Lauka datu tips	Lauka tips	Var būt tukšs?	Ir unikāls?
stopId	varchar(4)	PK	Nē	Jā
stopName	varchar(30)		Nē	Jā
stopIsStation	tinyint(1)		Nē	Nē

2.5.3 Tabula „route”

Tabulas mērķis ir glabāt pamatinformāciju par maršrutiem.

Lauks	Lauka datu tips	Lauka tips	Var būt tukšs?	Ir unikāls?
routeId	varchar(4)	PK	Nē	Jā
routeName	varchar(100)		Nē	Nē
routeSchedule	varchar(7)		Nē	Nē
routeCapacity	int(2)		Nē	Nē
routeValidFrom	int(8)		Nē	Nē
routeValidTill	int(8)		Nē	Nē
routePlatform	int(2)		Nē	Nē

2.5.4 Tabula „route_stop”

Tabulas mērķis ir glabāt informāciju par pieturu attiecībām ar maršrutiem.

Lauks	Lauka datu tips	Lauka tips	Var būt tukšs?	Ir unikāls?
routeFk	varchar(4)	FK	Nē	Nē
stopFk	varchar(4)	FK	Nē	Nē
stopNum	int(2)		Nē	Nē
cost	decimal(5,2)		Nē	Nē
arriveTime	int(4)		Nē	Nē

2.5.5 Tabula „trip”

Tabulas mērķis ir glabāt informāciju par reisiem.

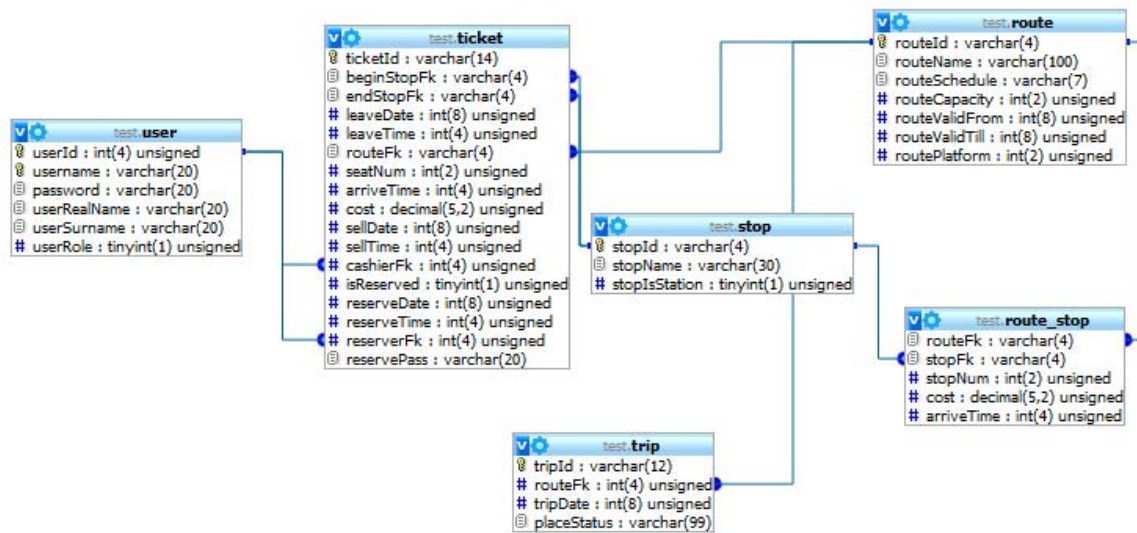
Lauks	Lauka datu tips	Lauka tips	Var būt tukšs?	Ir unikāls?
tripId	varchar(12)	PK	Nē	Jā
routeFk	varchar(4)	FK	Nē	Nē
tripDate	int(8)		Nē	Nē
placeStatus	varchar(99)		Nē	Nē

2.5.6. Tabula „ticket”

Tabulas mērķis ir glabāt informāciju par pārdotajām vai rezervētajām biļetēm.

Lauks	Lauka datu tips	Lauka tips	Var būt tukšs?	Ir unikāls?
ticketId	varchar(14)	PK	Nē	Jā
beginStopFk	varchar(4)	FK	Nē	Nē
endStopFk	varchar(4)	FK	Nē	Nē
leaveDate	int(8)		Nē	Nē
leaveTime	int(4)		Nē	Nē
routeFk	varchar(4)	FK	Nē	Nē
seatNum	int(2)		Nē	Nē
arriveTime	int(4)		Nē	Nē
cost	decimal(5,2)		Nē	Nē
sellDate	int(8)		Nē	Nē
sellTime	int(4)		Nē	Nē
cashierFk	int(4)	FK	Nē	Nē
isReserved	tinyint(1)		Nē	Nē
reserveDate	int(8)		Jā	Nē
reserveTime	int(4)		Jā	Nē
reserverFk	int(4)	FK	Jā	Nē
reservePass	varchar(20)		Jā	Jā

2.6. Fiziskais ER modelis



(10. attēls. Fiziskā ER modeļa diagramma)

3. NODAĻA

TESTĒŠANAS DOKUMENTĀCIJA

3.1. Ievads

3.1.1. Nolūks

Šis dokuments satur biļešu tirdzniecības un reisu uzskaites sistēmas testēšanas rezultātus. Testēšanas dokumentācijā apkopotas testēšanas gaitā izmantotās situācijas, ievaddati un programmatūras izvadītie dati.

3.1.2. Darbības sfēra

Dotais dokuments specificē prasības attiecībā uz Biļešu tirdzniecības un reisu uzskaites sistēmas (tālāk šajā dokumentā: sistēma) testēšanas dokumentāciju.

3.1.3. Saistība ar citiem dokumentiem

Dokumenta izstrādē izmantoti sekojoši dokumenti: programmatūras prasību specifikācija un programmatūras prasību apraksts.

3.2. Testēšanas rezultāti

3.2.1. Lietotāja pieteikšanās

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Nav aizpildīts kāds no laukiem.	Paziņojums par katru neaizpildīto lauku.	Iziets
Ievadīts neeksistējoša DB lietotāja lietotājvārds.	Paziņojums par sistēmā neeksistējošu lietotāju ar ievadīto lietotājvārdu.	Iziets
Lietotājvārdam neatbilstošas paroles ievade.	Paziņojums par neatbilstošu ievadīto paroli.	Iziets
Ievadīts DB eksistējoša lietotāja lietotājvārds un atbilstoša parole.	Sistēma pāriet uz lietotāja lomai atbilstošu lapu.	Iziets

3.2.2. Lietotāja izveide

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Nav aizpildīts kāds no laukiem.	Paziņojums par katru neaizpildīto lauku.	Iziets
Ievadīts pārsniegts vai nepietiekošs simbolu skaits kādā no laukiem.	Paziņojums par katru nekorekti aizpildīto lauku.	Iziets
DB eksistējoša lietotāja lietotājvārda ievade.	Paziņojums par jau eksistējošu DB lietotāju ar ievadīto lietotājvārdu.	Iziets
Nospiesta poga izveidot lietotāju.	Sistēma izveido lietotāju un paziņo par veiksmīgu lietotāja izveidi.	Iziets

3.2.3. Lietotāja datu maiņa

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Lietotājs izvēlas mainīt lietotāju datus.	Izvada visu DB ievadīto lietotāju sarakstu.	Iziets
Izvēlas konkrētu lietotāju.	Izvada informāciju par attiecīgo lietotāju.	Iziets
Nav aizpildīts kāds no laukiem.	Paziņojums par katru neaizpildīto lauku.	Iziets
Ievadīts pārsniegts vai nepietiekošs simbolu skaits kādā no laukiem.	Paziņojums par katru nekorekti aizpildīto lauku.	Iziets
Nospiesta poga mainīt.	Sistēma maina lietotāja datus un paziņo par veiksmīgu lietotāja datu maiņu.	Iziets

3.2.4. Lietotāja dzešana

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Lietotājs izvēlas dzēst lietotāju.	Visu DB ievadīto lietotāju saraksta izvide.	Iziets
Lietotājs izvēlas konkrētu lietotāju.	Sistēma dzēš lietotāju un paziņo par veiksmīgu lietotāja dzešanu.	Iziets

3.2.5. Pieturas izveide

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Nav aizpildīts kāds no laukiem.	Paziņojums par katru neaizpildīto lauku.	Iziets
Ievadīts pārsniegts vai nepietiekošs simbolu skaits kādā no laukiem.	Paziņojums par katru nekorekti aizpildīto lauku.	Iziets

DB eksistējošas pieturas nosaukuma ievade.	Paziņojums par jau eksistējošu DB pieturu ar ievadīto nosaukumu.	Iziets
Nospiesta poga izveidot pieturu.	Sistēma paziņo par veiksmīgu pieturas izveidi.	Iziets

3.2.6. Pieturas datu maiņa

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Lietotājs izvēlas mainīt pieturas datus.	Izvada visu DB ievadīto pieturu sarakstu.	Iziets
Izvēlas konkrētu pieturu.	Izvada informāciju par attiecīgo pieturu.	Iziets
Nav aizpildīts kāds no laukiem.	Paziņojums par katru neaizpildīto lauku.	Iziets
Ievadīts pārsniegts vai nepietiekošs simbolu skaits kādā no laukiem.	Paziņojums par katru nekorekti aizpildīto lauku.	Iziets
Nospiesta poga mainīt.	Sistēma maina pieturas datus un paziņo par veiksmīgu pieturas datu maiņu.	Iziets

3.2.7. Pieturas dzēšana

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Lietotājs izvēlas dzēst pieturu.	Visu DB ievadīto pieturu saraksta izvade.	Iziets
Lietotājs izvēlas konkrētu pieturu.	Sistēma dzēš pieturu un paziņo par veiksmīgu pieturas dzešanu.	Iziets
Pietura atrodas kāda maršruta sastāvā.	Paziņojums par neveiksmīgu dzēšanu un paskaidrojums.	Iziets

3.2.8. Reisu kustības saraksta atlase

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Nav aizpildīts kāds no laukiem.	Paziņojums par katru neaizpildīto lauku.	Iziets
Ievadīts pārsniegts vai nepietiekošs simbolu skaits kādā no laukiem.	Paziņojums par katru nekorekti aizpildīto lauku.	Iziets
Datuma vai laika laukos ievadīti negatīvi skaitļi vai burti.	Paziņojums par katru no nekorektajiem ievaddatiem.	Iziets
Meklēšana pēc ievadītajiem kritērijiem	Reisu kustības saraksta izvade.	Iziets
DB neeksistē kritērijiem atbilstošs maršruts.	Paziņojums par to, ka nekas netika atrasts.	Iziets
Nospiešana uz konkrēta maršruta.	Pieturu saraksta konkrētajā maršrutā izvade.	Iziets

3.2.9. Biļetes pārdošana

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Nav aizpildīts kāds no laukiem.	Paziņojums par katru neaizpildīto lauku.	Iziets
Ievadīts pārsniegts vai nepietiekošs simbolu skaits kādā no laukiem.	Paziņojums par katru nekorekti aizpildīto lauku.	Iziets
Datuma vai laika laukos ievadīti negatīvi skaitļi vai burti.	Paziņojums par katru no nekorektajiem ievaddatiem.	Iziets
Meklēšana pēc ievadītajiem kritērijiem	Reisu kustības saraksta izvade.	Iziets
DB neeksistē kritērijiem atbilstošs maršruts.	Paziņojums par to, ka nekas netika atrasts.	Iziets

Konkrēta maršruta izvēle	Vietu tipa izvēlnes izvade.	Iziets
Vietu tipa izvēle.	Pārdošanai pieejamo vietu izvade.	Iziets
Vietu norāde.	Drukas veida izvēlnes izvade.	Iziets
Drukas veida izvēle.	Attiecīga teksta faila sagatavošana.	Iziets

3.2.10. Biļetes rezervācija

Testpiemērs	Sagaidāmais rezultāts	Rezultāts
Nav aizpildīts kāds no laukiem.	Paziņojums par katru neaizpildīto lauku.	Iziets
Ievadīts pārsniegts vai nepietiekošs simbolu skaits kādā no laukiem.	Paziņojums par katru nekorekti aizpildīto lauku.	Iziets
Datuma vai laika laukos ievadīti negatīvi skaitļi vai burti.	Paziņojums par katru no nekorektajiem ievaddatiem.	Iziets
Meklēšana pēc ievadītajiem kritērijiem	Reisu kustības saraksta izvade.	Iziets
DB neeksistē kritērijiem atbilstošs maršruts.	Paziņojums par to, ka nekas netika atrasts.	Iziets
Konkrēta maršruta izvēle	Brīvo vietu saraksta izvade.	Iziets
Rezervācijas vietu izvēle.	Lūgums ievadīt rezervācijas paroli.	Iziets
Rezervācijas paroles ievade.	Rezervācijas veikšana, paziņojums par veiksmīgu rezervāciju.	Iziets
Rezervācijas parole nav ievadīta.	Lūgums ievadīt rezervācijas paroli.	Iziets

4. PROGRAMMATŪRAS PROJEKTA ORGANIZĀCIJA

Projekta organizācija notika pēc prototipēšanas modeļa. Sākotnēji tika veikta izpēte Rīgas Starptautiskajā Autoostā, lai iegūtu informāciju par biļešu tirdzniecības un pasažieru pārvadājumu specifiku. Tika apkopotas visas tās prasības, kas ir nepieciešamas sistēmas pamatfunktionalitātei un tika izstrādāts sistēmas pirmais prototips – lietotāja saskarne ar dažām pamatfunkcijām. Uzreiz pēc prototipa izstrādes, tika pakāpeniski pilnveidota funkcionalitāte un veidoti jauni prototipi, līdz prasības tika izstrādātas pilnīgi. Tādā pašā veidā notika visu lietotāju lomu izpēte, un prototipu izveide un pilnveidošana.

5. KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu kvalitatīva programmatūras produkta izstrādi tika veiktas sekojošas darbības:

- Projektu pavadošā dokumentācija tika izstrādāta saistībā ar Latvijas Valsts standartiem.
- Lai nodrošinātu koda saprotamību un lasāmību, tika veikta pieturēšanās pie labās programmēšanas prakses.
- Izstrādātajām sistēmas daļām tika veikta vienībtestēšana.
- Izstrādes laikā atrastās kļūdas vai nepilnības tika uzreiz izlabotas.

6. KONFIGURĀCIJU PĀRVALDĪBA

Sistēmas izstrādē speciāli konfigurācijas pārvaldības rīki netika izmantoti. Tiklīdz sistēmā tika veiktas vērā ņemamas izmaiņas, ieviestas jaunas funkcijas vai mainīta funkcionalitāte, tika veikti pirmkoda un datubāzes dublējumi. Tie tika saglabāti ārējos datu nesējos un augšupielādēti uz Google servera, lai garantētu dubultu drošību tehnisku kļūmju gadījumā un samazinātu kaitējumu izstrādes procesam līdz minimumam. Dublējumu atšķiršanai, tie tika glabāti atsevišķās mapītēs, kas vēlāk arī tika arhivētas. To nosaukumos tika iekļauts arī dublēšanas datums un laiks. Arī izstrādājot PPS, PPA un pašu kvalifikācijas darbu tika veikti dokumentācijas dublējumi. Tā kā sistēma joprojām ir izstrādes procesā, tai nav piešķirts nekāds versijas numurs.

7. DARBIETILPĪBAS APRĒĶINS

Darbietilpības aprēķināšanai tika izvēlēts COCOMO modelis.

Pārskatot projektu ieguvu aptuveni šādu funkcijpunktu skaitu:

	Sarežģīti	Vidēji	Vienkārši	Kopā
Ievadi	6*3+	4*3+	3*10=	60
Izvadi	7*3+	5*3+	3*9=	63
Iekšējie datu faili	15*0+	10*0+	7*6=	42
Ārējie interfeisa faili	10*0+	7*0+	5*0=	0
Vaicājumi	6*4+	4*14+	3*27=	161
Nepieskaņotu funkcijpunktu skaits:				=326

Izmantojot PHP, JavaScript un SQL valodas konstantes, ieguvu paredzamo rindiņu skaitu ap 12000 koda rindu un paredzamo darba izpildes laiku: 9.39 personmēneši.

Dotajā izstrādes stadijā var paredzēt, ka koda rindiņu skaits būs mazāks, jo tiek izmantots DRY (Don't Repeat Yourself) princips un līdzīgie programmas koda moduļi tiek apvienoti funkcijās.

Ja salīdzina ar eksistējošām sistēmām, tad Rīgas Starptautiskās Autoostas biļešu tirdzniecības sistēmu rakstīja grupa no 3 programmētājiem un projekta realizācija viņiem prasīja 4 personmēnešus, līdz ar to, varu teikt, ka mans aprēķins varētu būt diezgan precīzs.

8. PROGRAMMATŪRAS PIRMKODS

```
<?php
//Mainīgie savienojuma saites parametru maiņai
$servername = 'localhost';
$username = 'root';
$password = "";
$dbase = 'test';
$connectionErr = 'Neizdevās savienoties ar serveri!';
$dbaseErr = 'Neizdevās savienoties ar datu bāzi!';

$connection = mysql_connect($servername, $username, $password);
//Ja savienojas ar DB serveri, izvēlas DB un noklusējuma simbolu kodējumu
if ($connection) {
    mysql_select_db($dbase) or die ($dbaseErr);
    mysql_set_charset('utf8', $connection);
}
else {
    die ($connectionErr);
}

//funkcija paredzēta datu apstrādei pirms nosūtīšanas uz serveri, tā pasargā sistemu no SQL
injekcijām
function checkInput($value) {
    if (!get_magic_quotes_gpc()) {
        $value = stripslashes($value);
        $value = mysql_real_escape_string($value);
        return $value;
    }
    else {
        $value = mysql_real_escape_string($value);
        return $value;
    }
}

?>
```

/* funkcija dateInput ģenerē dienas, mēneša, vai gada izvēles izskrītošā saraksta tipa, kur pēc noklusējuma izvēlēta tekošā diena, mēnesis, vai gads */

```
function dateInput($id,$name,$size,$type) {
    if ($type == "day") {
        $today = date("d");
        echo "<select id=\"\".$id.\"\" name=\"\".$name.\"\" size=\".$size.>";
        for ($i = 1; $i <= 31; $i++) {
            if($i == $today){
                echo "<option value=\"\" . $i . \"\" selected>" . $i;
            }
            else {
                echo "<option value=\"\" . $i . \"\">" . $i;
            }
        }
        echo "</select>";
    }
    else if ($type == "month") {
        $thisMonth = date("m");
        echo "<select id=\"\".$id.\"\" name=\"\".$name.\"\" size=\".$size.>";
        for ($i = 1; $i <= 12; $i++) {
            if($i == $thisMonth){
                echo "<option value=\"\" . $i . \"\" selected>" . $i;
            }
            else {
                echo "<option value=\"\" . $i . \"\">" . $i;
            }
        }
        echo "</select>";
    }
    else {
        $thisYear = date("Y");
        echo "<select id=\"\".$id.\"\" name=\"\".$name.\"\"
size=\".$size.>";
    }
}
```

```

        for ($i = $thisYear; $i <= $thisYear + 1; $i++) {
            if($i == $thisYear){
                echo "<option value=\"\" . $i . \"\" selected>" . $i;
            }
            else {
                echo "<option value=\"\" . $i . \"\">" . $i;
            }
        }
        echo "</select>";
    }
}

/* funkcija stopInput izveido teksta lauku pieturas ievadei kurā iebūvēts JavaScript pieturas
nosaukuma priekšā teicēja izsaukums */
function stopInput($id,$name,$isStation) {
    echo "<input type=\"text\" id=\"\" . $id . \"\" name=\"\" . $name . \"\"
onkeyup=\"suggestStop(\" . $id . \",\" . $id . \"Suggest\",\" . $isStation . \");\"
autocomplete=\"off\"/><div id=\"\" . $id . \"Suggest\"></div>";
}

/* funkcija textInput izveido teksta lauku ar padotajiem lauka ID un nosaukumu */
function textInput($id,$name) {
    echo "<input type=\"text\" id=\"\" . $id . \"\" name=\"\" . $name . \"\"
autocomplete=\"off\" />";
}

/* funkcija passInput izveido teksta lauku par padotajiem lauka ID un nosaukumu */
function passInput($id,$name) {
    echo "<input type=\"password\" id=\"\" . $id . \"\" name=\"\" . $name . \"\"
autocomplete=\"off\" />";
}

/* funkcija check pārbauda, vai nav vai a $_GET palīdzību ir padots ievads, vai ievadā nav
tukša simbolu virkne un atgriež $_GET datus apstrādātus pret SQL injekcijām*/
function check($varName) {
    if (isset($_GET[$varName]) && $_GET[$varName] != "")
    {
        return checkInput($_GET[$varName]);
    }
}

```

```

        else {
            exit;
        }
    }
?>

// Reisa atlasē skripta fails
<?php
//Uzsāk sessiju
session_start();

if (isset($_SESSION["role"])) {
    //Ja sessija ir autentificēta, dodas uz sistēmas lietotāja kategorijai atbilstošo lapu un
    pārtrauc skripta darbu.
    if ($_SESSION["role"] != "1") {
        header("location:index.php");
        exit();
    }

    if (isset($_GET["logout"])) {
        session_destroy();
        header("location:index.php");
    }
}

else {
    header("location:index.php");
    exit();
}

require('script.php');
require('database.php');
// Ievadā saņemtos mainīgos glabā sessijas masīvā, lai vienkāršotu AJAX darbu un novērstu
nepieciešamību katru nākamo soli izsaukt ar funkcijām ar daudz parametriem, arī ļauj pabeigt
biļetes iegādi nejauša pārlūka aizvēršanas vai datora izslēgšanās gadījumā
$_SESSION["beginStop"] = check("beginStop");
$_SESSION["endStop"] = check("endStop");

```

```

$_SESSION["day"] = check("day");
$_SESSION["month"] = check("month");
$_SESSION["year"] = check("year");
$_SESSION["date"] = $_SESSION["year"] . $_SESSION["month"] . $_SESSION["day"];
$_SESSION["hours"] = check("hours");
$_SESSION["minutes"] = check("minutes");
$_SESSION["weekday"] = date("N", mktime(0, 0, 0, $_SESSION["month"],
$_SESSION["day"], $_SESSION["year"]));
//atiestata reisaId, lai atgriešanās biļetes pirkšanas pirmajā solī, neglabātos palikusī
informācija no kāda no nākamajiem soļiem
unset($_SESSION["routeId"]);
//SQL vaicājums, kas atlasa derīgos reisus pēc ievadītajiem parametriem
$query = "
SELECT *
FROM (
SELECT route.routeId, route_stop.cost AS tarifs, route_stop.arriveTime AS pienak
FROM route
JOIN route_stop ON route.routeId = route_stop.routeFk
JOIN STOP ON route_stop.stopFk = stop.stopId
WHERE stop.name=\'' . $_SESSION["endStop"] . '\''
) AS A
JOIN (
SELECT route.routeId, route.routeName AS reisa_nosaukums, arriveTime AS atiet
FROM route
JOIN route_stop ON route.routeId = route_stop.routeFk
JOIN STOP ON route_stop.stopFk = stop.stopId
WHERE stop.name=\'' . $_SESSION["beginStop"] . '\''
AND route.schedule LIKE \''%\" . $_SESSION["weekday"] . '%\'
AND time >= \'' . $_SESSION["hours"] . $_SESSION["minutes"] . '\''
AND valid_from <= \'' . $_SESSION["year"] . $_SESSION["day"] .
$_SESSION["month"] . '\''
AND route_stop.stopnum=\''1\'
) AS B ON A.id = B.id
ORDER BY atiet";

```

```

$result = mysql_query($query);

if(mysql_num_rows($result) == 0) {
    echo 'Nekas netika atrasts.';
}
else {
    //atgriež rezultātus jau tabulas veidā
    echo "<table id=\"list\">";
    echo "<tr><th>Reisa
nosaukums</th><th>Atiet</th><th>Pienāk</th><th>Tarifs</th></tr>";
    while($row = mysql_fetch_array($result)) {
        echo "<tr><td>";
        echo $row["reisa_nosaukums"];
        echo "</td><td>";
        echo $row["atiet"];
        echo "</td><td>";
        echo $row["pienak"];
        echo "</td><td>";
        echo $row["tarifs"];
        echo "</td></tr>";
        echo "<input type=\"button\" value=\"Izvēlēties\"
onClick=\"step2(' . $row[\"id\"] . ');\" />";
        echo "<input type=\"button\" value=\"Cita pietura\"
onClick=\"step15(' . $row[\"id\"] . ');\" />";
        echo "<input type=\"button\" value=\"Aizvērt\"
onClick=\"clearStep15(' . $row[\"id\"] . ');\" />";
        echo "<div id=\"s\" . $row[\"id\"] . \"\" ></div>";
    }
    echo "</table>";
}
}
?>

```

```

//Pilnvarošanās skripta fails
<?php
//Pieprasa savienošānās ar datu bāzi skriptu - savienojas ar datu bāzi.
require("database.php");
//Pieprasa nodefinēto funkciju skriptu.
require("script.php");
//Uzsāk sesiju
session_start();
//Pārbauda, vai sesija ir autentificēta.
if (isset($_SESSION["role"])) {
    //Ja sesija ir autentificēta, dodas uz sistēmas lietotāja kategorijai atbilstošo lapu un
    pārtrauc skripta darbu.
    if ($_SESSION["role"] == "1") {
        header("location:cashier.php");
        exit();
    }
    if ($_SESSION["role"] == "2") {
        header("location:informationService.php");
        exit();
    }
    if ($_SESSION["role"] == "3") {
        header("location:administrator.php");
        exit();
    }
}

//Pretējā gadījumā lietotājs nav autentificējies.
else {
    //Pārbauda, vai caur POST ir padoti lietotājvārds un parole. Pārbauda, vai to
    vietā nav padotas tukšumzīmes.
    if ( isset($_POST["username"]) && (trim($_POST["username"])!="") &&
    isset($_POST["password"]) && (trim($_POST["password"])!="")) {
        //Apstrādā POST, lai izsargātos no datu bāzes injekcijām un piešķir
        attiecīgiem mainīgajiem.

```

```

$username = checkInput($_POST["username"]);
$password = checkInput($_POST["password"]);
//Sastāda vaicājumu datu bāzei, izmantojot lietotāja padotos datus.
$query = "
SELECT *
FROM user
WHERE username='".$username.'"
AND password='".$password.'";
//Izpilda vaicājumu un rezultātu saglabā mainīgajā.
$result=mysql_query($query);
//Ja vaicājums atgriež 1 rindu, lietotājs ir ievadīts datu bāzē.
    if (mysql_num_rows($result) == 1) {
        //Izveido objektu ar atgrieztajiem laukiem.
        $row = mysql_fetch_object($result);
        //Piešķir sesijai lietotāja datus.
        $_SESSION["role"] = $row->role;
        $_SESSION["username"] = $row->username;
        $_SESSION["name"] = $row->name;
        $_SESSION["surname"] = $row->surname;
        //Dodas uz sistēmas lietotāja kategorijai atbilstošo lapu.
        if ($_SESSION["role"] == 1) {
            header("location:cashier.php");
            exit();
        }
        if ($_SESSION["role"] == 2) {

header("location:informationService.php");

            exit();

        }
        if ($_SESSION["role"] == 3) {

header("location:administrator.php");

            exit();

        }
    }
}

```

```

        //Pretējā gadījumā lietotājs datu bāzē netika atrasts.
        else{
            //Paziņo par neveiksmīgu autentifikāciju un
pārtrauc skripta darbu.

            $errmsg = "Lietotājs ar ievadīto lietotājvārdu
un paroli nav atrasts datu bāzē. Lūdzu pārlicinieties par datu pareizību un mēģiniet
autenticēties vēlreiz!";
        }
    }
}
?>
<?php echo '<?xml version="1.0" encoding="UTF-8" ?>'?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="lv" xml:lang="lv">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Biļešu tirdzniecības un reisu uzskaites sistēma - Pilnvarošanās</title>
    </head>
    <body>
        <form action="<?php echo $_SERVER["PHP_SELF"]; ?>" method="POST"
>
        <?php
            //Ja autentifikācija notikusi neveiksmīgi, parāda kļūdas paziņojumu.
            if (isset($errmsg)) {
                echo $errmsg;
            }
        ?>
        <input type="text" name="username" />
        <input type="password" name="password" />
        <input type="submit" value="Autenticēties" />
    </form>
</body>
</html>

```

9. DOKUMENTĀRĀ LAPA

Kvalifikācijas darbs „*Bīlešu tirdzniecības un reisu uzskaites sistēma*” 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: *Artūrs Aleksāns* _____ **31.05.2010.**

Rekomendēju darbu aizstāvēšanai

Darba vadītāja: *Dr. h. dat. Juris Borzovs* _____ **.05.2010.**

Recenzents: *M. dat. Maija Ļaksa*

Darbs iesniegts 31.05.2010.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Dainis Dosbergs* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2010. prot. Nr. _____, vērtējums _____ (_____)

Komisijas sekretārs(-e): _____