



Liepājas Valsts tehnikums

CarRent(AutoRezervācijasSistēma)

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Profesionālā kvalifikācija

Programmēšanas tehniķis

Grupas nosaukums

4PT

Projekta izstrādātājs

Dāvis Barons

Eksāmena datums 2024. gada 20. jūnijs

Liepāja 2023

Saturs

Saturs	1
Ievads	4
1.Uzdevuma formulējums	5
2. Programmatūras prasību specifikācija	6
2.1. Produkta perspektīva	6
2.2. Sistēmas funkcionālās prasības	6
2.3. Sistēmas nefunkcionālās prasības.....	14
2.4. Gala lietotāja raksturiezīmes	15
3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums.....	16
3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts	16
3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts	17
4. Sistēmas modelēšana un projektēšana.....	19
4.1. Sistēmas struktūras modelis	20
4.2. Klašu diagramma / ER diagramma.....	21
4.3. Funkcionālais un dinamiskais sistēmas modelis	21
4.4. Aktivitāšu diagramma (Activity).....	22
4.5. Lietojumgadījumu diagramma (Use Case).....	24
4.6. Sistēmas moduļu apraksts un algoritmu shēmas	25
5. Lietotāju ceļvedis	26
1.Ielogošanās sistēmā.	26
2.Sākuma sadaļa.	27
3.Nomas sadaļa.....	28
4.Admin sadaļa	29
6. Testēšanas dokumentācija.....	31
6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums.....	31
6.2. Testpiemēru kopa	32

Identifikātoru atšifrējums	32
6.3. Prasības	32
6.4. Testpiemēri	34
6.5. Testēšanas žurnāls	35
7. Lietoto terminu un saīsinājumu skaidrojumi.....	38
8. Secinājumi.....	39
9. Literatūras un informācijas avotu saraksts	41
Pielikums	42

Ievads

Dokumentā tiks aprakstīta mājaslapas "CarRent" dokumentācija, kā piemēram programmas darbība, kā arī tās funkcionālās un nefunkcionālās prasības. Šeit prasības tiks ļoti skaidri un saprotami attēlotas, lai ar dokumentu varētu ērti un viegli darboties, un saprast programmas darbības principu caur dokumentāciju, kas šeit tiks sarakstīta.

Dokumentā tiks aprakstīts uzdevuma formulējums, tas ir tā darbības princips un funkcionālās darbības, kā arī programmas idejas apraksts. Tiks aprakstītas sistēmas izstrādes līdzekļu un rīku pamatojums, tas ir ar ko tiks veidota šī programmatūra, kurā programmēšanas valodā tiks taisīta programmas funkcionālā daļa, kā arī tiks aprakstīts, kur tika veidota programmatūras nefunkcionālā daļa, kā piemēram, kur izveidots dizains, tas ir piemēram "Header", "Footer", "Button" u.c.

Programmai tiks izveidotas diagrammas, lai dokumenta lasītājam būtu vieglāk shematiski saprast funkcijas nozīmi, un kā arī saprast, kā autors ir domājis šo izveidot, kā arī dokumentā tiks izveidots testēšanas žurnāls, lai varētu aplūkot veiksmīgos testpiemērus, kā arī neveiksmīgos testpiemērus.

Programmas mērķis ir izveidot auto rezervācijas mājaslapu, kas ir "CarRent" web lapa, kur lietotājiem būs iespēja ērti un viegli rezervēt auto internetā uz noteiktu laiku, bez nekādiem sarežģījumiem, par vislabākajām cenām.

Šī programma būs konkurents tādiem auto rezervēšanas biznesmeņiem kā piemēram "CityBee" un "BoltDrive", tādējādi šai programmai būs jābūvē vieglāka un ērtāka pieeja auto rezervācijai, lai tieši šī programma būtu paša popularitātes augšgalā.

Mērķa auditorija šai programmai būtu cilvēki, kam nav pieejams auto, proti cilvēki, kam ir autovadītāja apliecība, tie uzrādot savus datus kā arī vadītāja apliecību, spēs norezervēt auto un doties sev izvēlēta braucienā, uz izvēlēta tarifa ilguma.

1.Uzdevuma formulējums

Mājaslapa "CarRent" ir neatkarīga un no pārlūkprogrammas startējama mājaslapa. Tā lietotājam ļauj atrast sev vispiemērotāko auto rezervācijai vadoties pēc jūsu pašu patikas. Lietotājam ir iespēja arī reģistrēties mājaslapā, un tāda veida ir iespēja saglabāt savus datus datu bāzē, ir arī iespēja aplūkot mājaslapu kā viesim, bet to varēs tikai aplūkot, neko nevarēs izdarīt mājaslapā.

Atverot mājaslapu lietotājam tiek piedāvāts ievadīt savu informāciju, tas ir reģistrēties, lai būtu vieglāk pēc tam rezervēt auto bez nekādiem sarežģījumiem ar vienas pogas klikšķi, bet protams var arī mājaslapu apskatīt bez saviem ievadītajiem datiem.

Tad, kad lietotājs būs veicis ielogošanos mājaslapā, tas tiks novirzīts sākuma sadaļā, kur būs neliels apraksts par mājaslapu un neliels auto klāsts. Lietotājs varēs vadīties pa navigācijas joslu, jo caur tās varēs nokļūt pārējās sadaļās. Nākamā sadaļa, uz kuru varēs pārvietoties, būs Rezervācijas sadaļa, kur notiks mājaslapas galvenās funkcijas, kur būs iespējams aplūkot auto, kuri būs iespējams rezervēt, kā arī būs iespēja izpētīt auto specifikāciju, izlaiduma gadu, motora tilpumu utt. Galu galā nospiežot pogu rezervēt, lietotājam būs iespējams izvēlēties tarifu, uz cik ilgu laiku vēlēties rezervēt auto, tad tiks attēlots kartē, kur atrodas šī rezervētais auto tavā pilsētā. Tad beigu beigās pēdējā sadaļā būs attēloti lietotāju dati, un tā saucās "Profila" sadaļa, kur lietotājiem būs iespējams aplūkot savus datus, ar ko tie ir reģistrējušies mājaslapā.

Kā arī būs admin/moderātoru sadaļa, kur varēs pārlūkot lietotāju rezervācijas.

Pārsvarā šī programma tiks izstrādāta tā, lai lietotājam būtu ērti un viegli iespējams sev rezervēt nepieciešamo auto, programmā tiks izstrādātas vieglas un lietotājam saprotamas funkcijas.

2. Programmatūras prasību specifikācija

Šī dokumenta sadaļa satur: “CarRent” Auto rezervācijas sistēmas produkta perspektīvas aprakstu, detalizēti aprakstītas sistēmas administratora un lietotāja funkcionālās un nefunkcionālās prasības, kā arī gala lietotāja raksturiezīmes.

2.1. Produkta perspektīva

Šīs programmas “CarRent” perspektīva ir, izstrādāta lai vienkāršotu un racionalizētu auto rezervēšanas procesu. Lietotne satur lietotājam draudzīgu saskarni, viegli lietojamas rezervācijas izvades funkcijas un vienkāršotu identitātes autorizācijas sistēmu. Lietotne nodrošina reāllaika auto rezervāciju kā arī tā nodrošina, to lai lietotājs varētu veikt rezervāciju ērti, un apdomāti. Lietotnes panākumi ir atkarīgi no tās spējas nodrošināt lietotājiem ērtu un ātru lietošanas pieredzi.

2.2. Sistēmas funkcionālās prasības

P.1. Mājaslapas “Auto rezervācijas sistēma” uzsākšana.

Mērķis:

Funkcija “Uzsākšana” nodrošina iespēju lietotājam atvērt mājaslapu “Auto rezervācijas sistēma”.

Ievaddati:

Ievadot mājaslapas adresi un nospiežot “ENTER”.

Apstrāde:

Notiek mājaslapas palaišana, tiek apstrādāta ievadītā hipersaite.

Izvaddati:

Lietotājam tiek piedāvāta iespēja reģistrēties vai ielogoties.

P.2. Mājaslapas “Auto rezervācijas sistēma” aizvēršana.

Mērķis:

Funkcija “x” nodrošina iespēju lietotājam beigt darbību ar mājaslapu “Auto rezervācijas sistēma”

Ievaddati:

Pogas “x” nospiešana mājaslapas augšējā labajā stūrī.

Apstrāde:

Funkcija pārbauda vai lietotājs ir nospiedis pogu “x” kas domāta mājaslapas aizvēršanai.

Izvaddati:

Tiek aizvērta mājaslapa “Auto rezervācijas sistēma”.

P.3. Mājaslapas “Auto rezervācijas sistēma” Nomas sadaļas atvēršana.

Mērķis:

Poga “Noma” nodrošina iespēju lietotājam atvērt sadaļu noma”.

Ievaddati:

Pogas “Noma” nospiešana mājaslapas augšā.

Apstrāde:

Funkcija pārbauda vai lietotājs ir nospiedis pogu “Noma” un vai nav jau atvērta šī sadaļa.

Izvaddati:

Aizved uz sadaļu “Noma”

P.4. Mājaslapas “Auto rezervācijas sistēma” pilsetas kartes atvēršana.

Mērķis:

Poga rezervēšanas sadaļa “Noma” nodrošina iespēju lietotājam atvērt konkrēta auto rezervācijas logu, kur arī atrodas karte ar auto atrašanās vietu ”.

Ievaddati:

Pogas “Rezervēt” nospiešana mājaslapas nomas sadaļā.

Apstrāde:

- 1) Funkcija pārbauda vai lietotājs ir nospiedis pogu “Rezervēt” un vai nav jau atvērta šī sadaļa.

Izvaddati:

Aizved uz auto rezervācijas logu kur attēlota karte.

P.5. Mājaslapas “Auto rezervācijas sistēma” Sākuma sadaļas atvēršana.

Mērķis:

Sadaļa “Sākums” nodrošina iespēju lietotājam atgriezties sākumā no jebkuras sadaļas, sākumlapa piedāvā informāciju par mājaslapu.

Ievaddati:

Poga “Sākums” nospiešana mājaslapas augšā

Apstrāde:

Funkcija pārbauda vai lietotājs atrodas sadaļā “Sākums”.

Izvaddati:

Aizved uz sadaļu “Sākums”.

P.6. Mājaslapas “Auto rezervācijas sistēma” lietotāju datu ievade reģistrācijā.

Mērķis:

Funkcija “Ievadīt datus” nodrošina iespēju lietotājam ievadīt savus datus, lai piekļūtu mājaslapai.

Ievaddati:

3.tabula**Ar reģistrāciju(Jauns lietotājs)**

Nosaukums	Obligāts	Piezīmes
Lietotājvārds	Jā	
Parole	Jā	
Auto vadītāja apliecība	Jā	

4.tabula**Bez reģistrācijas(Esošs lietotājs)**

Nosaukums	Obligāts	Piezīmes
Lietotājvārds	Jā	
Parole	Jā	

Apstrāde:

Nospiežot reģistrēšanās pogu, sistēma pārbauda vai visi obligātie ievaddati ir aizpildīti.

Veiksmīgas pārbaudes rezultātā, visi dati saglabājas datu bāzē.

Izvaddati:

Paziņojums par nepieciešamību aizpildīt visus obligātos ievaddatus

Paziņojums par veiksmīgu pieteikšanu.

P.10. Mājaslapas “Auto rezervācijas sistēma” rezervāciju dzēšana.

Mērķis:

Funkcija “Rezervāciju dzēšana” ir domāta, lai var izdzēst rezervācijas

Ievaddati:

Nospiež pogu dzēst

Apstrāde:

Pārbauda, vai lietotājs ir administrators.

Izvaddati:

Paziņojums par veiksmīgi izdzēstu rezervāciju.

P.11. Mājaslapas “Auto rezervācijas sistēma” Rezervāciju rediģēšana.

Mērķis:

Funkcija “Rezervāciju rediģēšana” ir domāta, lai var izlabot rezervāciju informāciju.

Ievaddati:

Nospiež pogu rediģēt.

Rediģēt, ievadlauku to, kas nepieciešams.

Apstrāde:

Pārbauda, vai lietotājs ir administrators.

Pārbauda vai visi ievaddati ir ievadīti

Izvaddati:

Paziņojums par nepieciešamība aizpildīt visus obligātos ievaddatus.

Paziņojums par veiksmīgi rediģētu rezervāciju

P.12. Mājaslapas “Auto rezervācijas sistēma” lietotāju autentifikācija.

Mērķis:

Funkcija “Lietotāju autentifikācija” ir paredzēta administratoriem, kur viņi varēs aptiprināt reģistrēto lietotāju ar pievienotu vadītāja apliecību

Ievaddati:

Atver sadaļu “Lietotāji”

Nospiež pogu pārbaudīt

Izvēlas atteikt vai apstiprināt

Apstrāde:

Pārbauda vai visi ievaddatu lauki ir ievadīti.

Pārbauda vai lietotāja autovadītāja apliecība ir derīga.

Ja viss ir, tad lietotājs tiek apstiprināts un tiek dota iespēja ielogoties.

Izvaddati:

Paziņojums "Lietotājs verificēts!".

Paziņojums "Lietotājs ir veiksmīgi pievienots!".

P.13. Mājaslapas "Auto rezervācijas sistēma" Lietotāju dzēšana.

Mērķis:

Funkcija "Lietotāju dzēšana" ir paredzēta administratoriem, kur viņi varēs dzēst lietotājus.

Ievaddati:

Nospiež pogu "Lietotāji".

Nospiež uz dzēšanas ikonas.

Apstrāde:

Pārbauda, vai tāds jau nav izdzēsts.

Izvaddati:

Paziņojums "Lietotājs jau sen izdzēsts!".

Paziņojums "Lietotājs veiksmīgi izdzēsts!".

P.14. Mājaslapas "Auto rezervācijas sistēma" paroles maiņa.

Mērķis:

Funkcija "Paroles maiņa" paredzēta, kad vēlas nomainīt paroli.

Ievaddati:

Nospiež pogu "Nomainīt paroli"

7.tabula

Paroles maiņa

Nosaukums	Obligāts	Piezīmes
Vecā parole	Jā	Sākuma jāievada vecā parole
Jaunā parole	Jā	Parole būs šifrēta
Jaunā parole (atkārtoti)	Jā	

Apstrāde:

Pārbauda, vai vecā parole sakrīt ar pašreizējo paroli.

Samaina veco paroli ar jauno.

Izvaddati:

Paziņojums "Parole nesakrīt!".

Paziņojums “Parole veiksmīgi nomainīta!”.

P.15. Mājaslapas “Auto rezervācijas sistēma” tarifu pievienošana.

Mērķis:

Funkcijas “Tarifu pievienošana” ir paredzēta moderatoriem un administratoriem, kad pievieno jaunus auto rezervācijas tarifus.

Ievaddati:

5.tabula

Nosaukums	Obligāts	Piezīmes
Auto markas nosaukums	Jā	Nepieciešams lai zinātu kuram auto pievienot tarifu.
1h tarifs - 4 Eur	Jā	Nosaki savu noteikto tarifa summu uz attiecīgo laika periodu.(? vietā jāraksta tevis izdomātā tarifa summa.)
2h tarifs - 6 Eur	Jā	Nosaki savu noteikto tarifa summu uz attiecīgo laika periodu.
5h tarifs - 12 Eur	Jā	Nosaki savu noteikto tarifa summu uz attiecīgo laika periodu.
1d tarifs - 40 Eur	Jā	Nosaki savu noteikto tarifa summu uz attiecīgo laika periodu.
Auto pieejamība	Jā	Atzīmē sistēmā cik šādi auto pieejami.

Apstrāde:

Pārbauda vai visi ievaddati ir aizpildīti.

Ja ir ievadīti visi dati, tad saglabājas datu bāzē

Izvaddati:

Paziņojums par nepieciešamību aizpildīt visus obligātos ievaddatus

Paziņojums par veiksmīgu pievienošanu.

P.16. Mājaslapas “Auto rezervācijas sistēma” Tarifu dzēšana.

Mērķis:

Funkcija “Tarifu dzēšana” ir domāta, lai var izdzēst aktīvos tarifus.

Ievaddati:

Nospiež pogu “Dzēst”

Apstrāde:

Pārbauda, vai lietotājs ir moderators vai administrators.

Izvad dati:

Paziņojums par veiksmīgi izdzēstu tarifu.

P.17. Mājaslapas “Auto rezervācijas sistēma” Tarifu rediģēšana.

Mērķis:

Funkcija “Tarifu rediģēšana” ir domāta, lai var izlabot tarifu informāciju.

Ievaddati:

Nospiež pogu rediģēt

Rediģēt, ievadlauku to, kas nepieciešams.

Apstrāde:

Pārbauda, vai lietotājs ir moderators vai administrators.

Pārbauda vai visi ievaddati ir ievadīti

Izvad dati:

Paziņojums par nepieciešamību aizpildīt visus obligātos ievaddatus

Paziņojums par veiksmīgi rediģētu tarifu.

P.18. Mājaslapas “Auto rezervācijas sistēma” Datuma perioda izvēle.

Mērķis:

Funkcija “Datuma perioda izvēle” ir domāta, lai var izvēlēties uz cik ilgu laiku nepieciešams rezervēt auto.

Ievaddati:

Nospiež uz kalendāra

Izvēles nepieciešamos datumus no-līdz.

Apstrāde:

Pārbauda vai visi ievaddati ir ievadīti

Izvad dati:

Paziņojums par veiksmīgi izvēlētu datumu.

P.19. Mājaslapas “Auto rezervācijas sistēma” auto rediģēšana.

Mērķis:

Funkcija “Auto rediģēšana” ir domāta, lai var rediģēt auto datus.

Ievaddati:

Atver sadaļu "Auto"

Spiež uz rediģēšanas ikonas un rediģē.

Apstrāde:

Pārbauda vai visi ievaddati ir ievadīti

Izvaddati:

Paziņojums par veiksmīgi rediģētu auto.

P.20. Mājaslapas “Auto rezervācijas sistēma” auto dzēšana.

Mērķis:

Funkcija “Auto dzēšana” ir domāta, lai var dzēst auto datus.

Ievaddati:

Atver sadaļu "Auto"

Spiež uz dzēšanas ikonas un dzēš ierakstu.

Apstrāde:

Pārbauda vai atlasīts auto dzēšanai.

Izvaddati:

Paziņojums par veiksmīgi dzēstu auto.

P.21. Mājaslapas “Auto rezervācijas sistēma” auto pievienošana.

Mērķis:

Funkcija “Auto pievienošana” ir domāta, lai var pievienot jaunus auto nomai.

Ievaddati:

Atver sadaļu "Auto"

Spiež uz "Pievienot auto" un pievieno ierakstu.

Apstrāde:

Pārbauda vai visi ievaddati ir aizpildīti..

Izvaddati:

Paziņojums par veiksmīgi pievienotu auto.

P.20. Mājaslapas “Auto rezervācijas sistēma” Stripe apmaksa.

Mērķis:

Funkcija “Stripe apmaksa” ir domāta, lai veikt apmaksu par rezervāciju.

Ievaddati:

Atver sadaļu "Noma"

Spiež uz pogas "Rezervēt"

Apstrāde:

Pārbauda vai nospiesta poga "Rezervēt".

Izvaddati:

Atveras Stripe apmaksa.

2.3. Sistēmas nefunkcionālās prasības

1. **Valodas prasības:** Mājaslapai jābūt pilnībā pieejamai latviešu valodā, tādējādi nodrošinot lietotājiem vieglu un saprotamu piekļuvi satura un funkcijām.
2. **Lietotājam draudzīga saskarne:** Mājaslapas saskarnei jābūt ērtai un ergonomiskai, kas nozīmē, ka tā ir intuitīva un viegli lietojama visiem lietotājiem.
3. **Veiktspēja:** Sistēmai jābūt veiktspējīgai, lai vienlaicīgi varētu apkalpot līdz 100 lietotājiem, neuztraucoties par darbības ātrumu vai sistēmas pārslodzi.
4. **Datu apstrādes ātrums:** Lai uzlabotu lietotāju pieredzi, datu apstrādei jānotiek ātri, neilgāk par 1 sekundi, nodrošinot lietotājiem ātru atbildi uz pieprasījumiem vai darbībām mājaslapā.
5. **Adaptācija uz dažādām ierīcēm:** Mājaslapai jābūt responsīvai, tai jāpielāgojas dažādām ierīcēm, piemēram, datoriem, planšetdatoriem un mobīlajām telefonām, lai nodrošinātu optimālu skatīšanos un lietošanu jebkurā ierīcē.
6. **Krāsu shēma:** Dominējošās mājaslapas krāsas ir pelēkas, balta vai melna, kas var nodrošināt vienkāršu, modernu un profesionālu izskatu.

7. **Autorizācija un datu bāze:** Mājaslapai jānodrošina droša autorizācija lietotājiem, lai piekļūtu noteiktai informācijai vai veiktu darbības, un jāizmanto datu bāze, lai efektīvi glabātu un pārvaldītu lietotāju datus un saturu.

2.4. Gala lietotāja raksturiezīmes

Lietotāji ir galvenie sistēmas lietotāji, kuri izmanto "CarRent" auto rezervācijas platformu, lai atrastu un rezervētu automašīnas savām vajadzībām. Viņi var pieteikties, izmantojot savus kontus vai reģistrējoties jauniem kontiem, lai piekļūtu rezervāciju funkcijām. Lietotāji izvēlas automašīnas, pēc tam rezervē tās un veic maksājumus, izmantojot sistēmas norēķinu funkcijas. Viņi arī var pārvaldīt savu profilu, redzēt savas rezervācijas un saņemt atbalstu no sistēmas administratoriem, ja rodas kādas problēmas vai jautājumi.

Administratori ir sistēmas pārvaldnieki un uzraudzītāji. Viņi ir atbildīgi par vispārēju sistēmas darbību un efektivitāti, nodrošinot, ka platforma darbojas kā paredzēts. Administratori pārvalda lietotāju kontus, uzstāda sistēmas konfigurācijas un nodrošina sistēmas drošību. Viņi var arī risināt lietotāju problēmas, kā arī nodrošināt tehnisko atbalstu un palīdzību lietotājiem, ja nepieciešams. Galvenais administrators ir atbildīgs par sistēmas kopējo darbību un koordinē citus darbiniekus, kas strādā pie platformas attīstības un uzturēšanas.

3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

Šajā nodaļā tiks aprakstīti izstrādes rīki un programmēšanas valodas, kas tiks izmantoti šajā projekta tapšanā, un vēl tiks paskaidrots, kādus alternatīvus rīkus es izmantošu šim projektam, un kādus citus es varēju izmantot realizējot šo projektu "CarRent"

3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts

Projekta "CarRent" izstrāde ir solis uz priekšu modernās programmēšanas pasaules virzienā, apvienojot savā darbā vairākus tehnoloģiskus un organizatoriskus elementus. Centrālais rīks šajā ceļā ir Visual Studio Code, kas ir kļuvis par uzticamu un efektīvu sabiedroto manā ikdienas darbā. Tā kā tas piedāvā plašu papildinājumu klāstu un intuitīvu interfeisu, tas sniedz ne tikai ērtus instrumentus koda rakstīšanai, bet arī rūpīgu koda pārvaldību un testēšanu.

Izvēloties JavaScript bibliotēku React projektam, ir ņemta vērā gan iepriekšējā pieredze, gan pārliecība par tās spēju radīt modernas un responsīvas mājaslapas. Tādējādi, izvēloties zināmu tehnoloģiju, tas atvieglo attīstības procesu un ļauj koncentrēties uz jaunu funkcionalitāti.

Darbs uz Windows darbstacijas ir loģisks solis, ņemot vērā darba vidi un komfortu, ko sniedz šī operētājsistēma. Tā nodrošina stabilu un uzticamu darba platformu, kas ļauj koncentrēties uz paša projekta būtību, nevis tehniskiem sarežģījumiem.

Microsoft Word izmantošana dokumentācijai sniedz plašas iespējas informācijas strukturēšanā un formatēšanā, nodrošinot skaidru un viegli saprotamu dokumentāciju. Tas palīdz ne tikai dokumentēt projektu, bet arī viegli dalīties informācijā ar citiem iesaistītajiem.

Testēšanas process, kas tiek veikts izmantojot Microsoft Excel, ir vēl viens svarīgs solis, lai nodrošinātu mājaslapas kvalitāti un pareizu darbību. Šis plānošanas un analīzes rīks ļauj rūpīgi izstrādāt testēšanas plānus, ierakstīt rezultātus un veikt nepieciešamās izmaiņas, lai uzlabotu produktu.

GitHub kā versiju kontroles rīks ne tikai sniedz drošu un ērtu veidu, kā pārvaldīt kodu un sadarboties ar citiem izstrādātājiem, bet arī nodrošina projektam skaidru struktūru un iespēju sekot līdzi izmaiņām. Tas veicina efektīvu projektu pārvaldību un palīdz saglabāt kodu organizētu un viegli pieejamu visiem iesaistītajiem.

Kopumā, šis projekts ir vērtīga un rūpīgi plānota iniciatīva, kas izmanto modernus rīkus un tehnoloģijas, lai nodrošinātu veiksmīgu rezultātu un uzlabotu produktivitāti un efektivitāti darba procesos.

- **Visual Studio Code (VS Code)** ir Microsoft izstrādāts avota koda redaktors, kas ir pieejams operētājsistēmām Windows, Linux un macOS. Tas ir plaši izmantots programmētāju vidū, piedāvājot daudzas funkcijas, kas padara kodu rakstīšanu un rediģēšanu efektīvāku un ērtāku. Šīs funkcijas ietver atbalstu atklūdošanai, kas ir ļoti noderīga kļūdu atrašanai un labošanai, sintakses izcelšanu, kas palīdz izcelt dažādas kodu daļas atkarībā no to tipa vai funkcijas, inteligēnto koda pabeigšanu, kas piedāvā ieteikumus un aizpilda kodu pēc daļējas rakstīšanas, fragmentus, kas ir iepriekš definēti kodu gabali, kas var būt viegli ievietoti, un koda pārveidošanu, kas ļauj veikt dažādas operācijas ar kodu, piemēram, pārkārtot, iekavas vai mainīt mainīgo nosaukumus.
- **React** ir JavaScript bibliotēka, kas tiek izmantota lietotāja saskarņu veidošanai. Tā piedāvā efektīvas un atkārtojamas veidnes, kas ļauj ātri izveidot modernas un responsīvas lietotāja saskarnes.
- **Microsoft Word** ir plaši pazīstams teksta procesors, kas piedāvā daudzas funkcijas dokumentu veidošanai, rediģēšanai un formatēšanai. Tas ir ērts rīks dokumentācijas izveidei, sniedzot iespējas strukturēt un vizuāli pielāgot tekstu.
- **Microsoft Excel** ir izklājlapu lietojumprogrammatūra, kas ļauj veidot tabulārus un veikt datu analīzi. Tas ir noderīgs rīks datu organizēšanai, aprēķiniem un vizuālizācijai.
- **GitHub** ir tīmeklī balstīts Git repozitoriju mitināšanas pakalpojums, kas sniedz iespēju glabāt un pārvaldīt kodu, kā arī veikt sadarbību un versiju kontroli projektos. Tas ir plaši izmantots izstrādes vidē, piedāvājot daudzas funkcijas, piemēram, izmaiņu izsekošanu, sadarbību, problēmu izsekošanu un kodu pārskatus.

3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts

"Mājaslapas "CarRent" veidošanai sākotnēji varētu izmantot "Eclipse" kā izstrādes rīku. Esmu iepriekš strādājis ar šo un esmu pieradis pie tās darbības principiem. Tomēr, neskatoties uz šo iespēju, izvēlējos strādāt ar "Visual Studio Code" (VSCode). Galvenais iemesls tam ir tas, ka VSCode man ir labāk pazīstams un piedāvā plašāku papildinājumu klāstu un lietojamības ērtības, kas palīdzēs efektīvi veidot "CarRent" mājaslapu.

Lai gan iespējams izvēlētos programmēšanas valodu PHP, esmu nolēmis dot priekšroku "React" JavaScript bibliotēkai. React piedāvā modernas un elastīgas iespējas mājaslapas izstrādei, un man ir iepriekšēja pieredze ar šo tehnoloģiju, kas padara darbu ar to īpaši ērtu un produktīvu.

Lai aprakstītu programmas darbību un funkcionalitāti, varētu izmantot arī "Google Docs", taču izvēlējos "Microsoft Word". Jaunās versijās tam ir vairāk formatēšanas opciju un funkciju, kas nodrošina plašākas iespējas dokumentācijas izveidē un organizēšanā.

Testēšanai, kā minēts, varētu izmantot "Google Sheets", taču izvēlējos strādāt ar "Microsoft Excel". Šī programma piedāvā plašākas funkcijas testēšanas plānu veidošanai un testu rezultātu analīzei, tādējādi nodrošinot precīzāku un efektīvāku testēšanas procesu.

Visbeidzot, lai veiktu versiju kontroli un sadarbību ar citiem izstrādātājiem, joprojām izmantos "GitHub". Tas ir stabils un drošs rīks, kas nodrošina efektīvu kodu pārvaldību un sadarbību, kā arī ļauj viegli sekot līdzi projektam un veikt izmaiņas kodā."

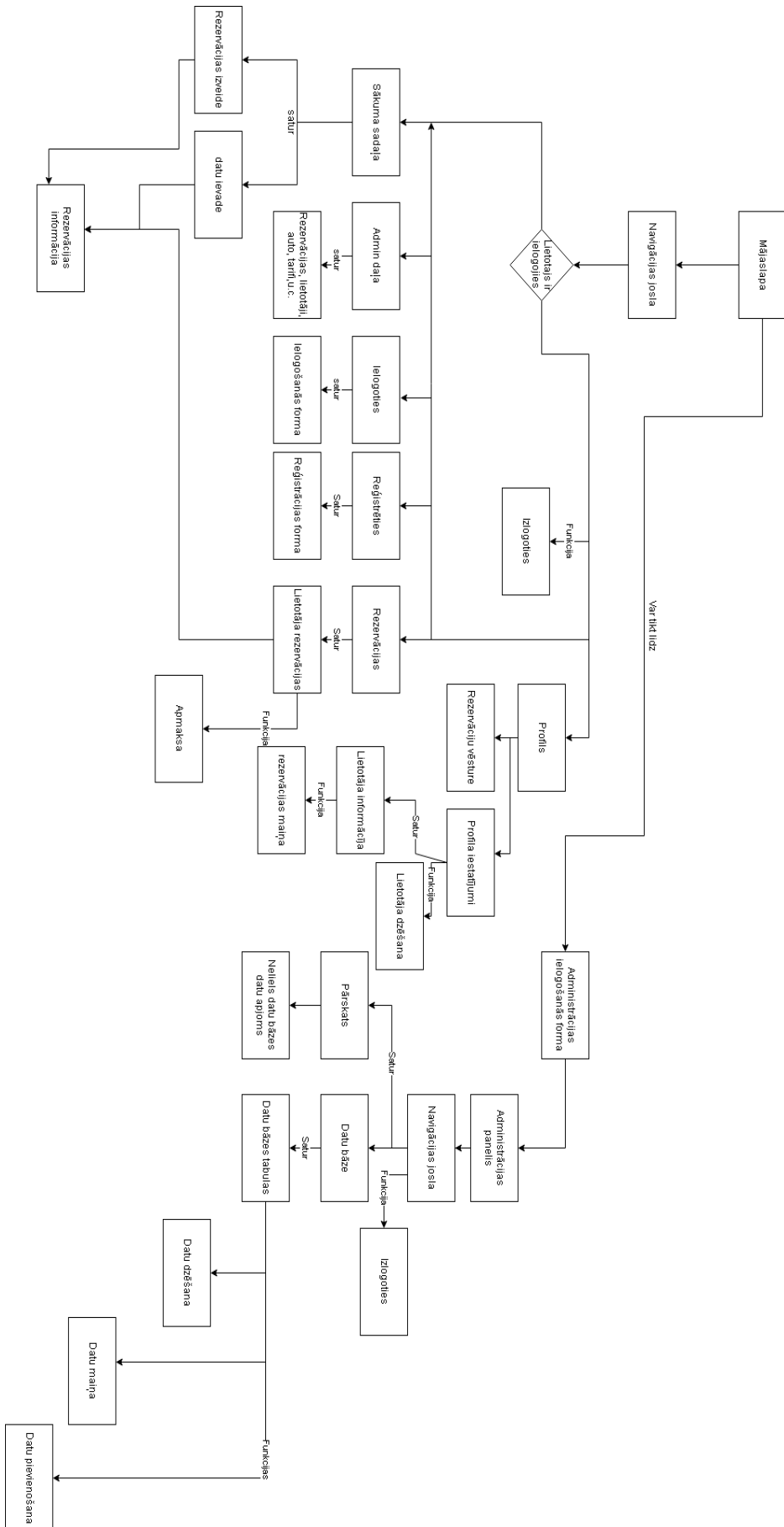
Šis paplašinātais teksts sniedz detalizētu informāciju par izvēlētajiem rīkiem un tehnoloģijām "CarRent" mājaslapas izveidei un atspoguļo racionālus iemeslus, kādēļ tie ir izvēlēti.

- **Eclipse** ir vairāku valodu programmatūras izstrādes vide, kas sniedz attīstītājiem integrētu izstrādes vidi un paplašināmu spraudņu sistēmu. Šī izstrādes vide ir plaši izmantota programmētāju kopienā, jo tā nodrošina daudzus rīkus un funkcijas, kas atvieglo programmu izstrādi.
- **PHP: Hypertext Preprocessor** ir atklātā pirmkoda skriptu valoda, kas tika izstrādāta sākotnēji ar mērķi nodrošināt dinamisku tīmekļa lapu ģenerēšanu servera pusē. PHP ir ļoti populāra un plaši izmantota valoda web izstrādē, jo tā piedāvā plašu funkciju klāstu, kas ļauj veikt datu apstrādi, sazināties ar datu bāzēm un veidot dinamiskus un interaktīvus tīmekļa resursus.
- **GoogleDocs** ir tiešsaistes tekstapstrādes programma, kas ļauj lietotājiem veidot, rediģēt un koplietot dokumentus tiešsaistē. Tas nodrošina iespēju veikt darbu pie dokumentiem neatkarīgi no ierīces un vietas, kurā tiek izmantots internets, un ļauj vairākiem lietotājiem vienlaikus rediģēt dokumentus.
- **GoogleSheets** ir izklājlapu lietojumprogramma, kas piedāvā iespēju veidot, rediģēt un dalīties izklājlapās tiešsaistē. Tas nodrošina daudzas funkcijas, kas ļauj organizēt un analizēt datus tabulārā formātā, veikt aprēķinus un vizualizācijas, kā arī sadarboties ar citiem lietotājiem pie kopīgiem datiem.

4. Sistēmas modelēšana un projektēšana

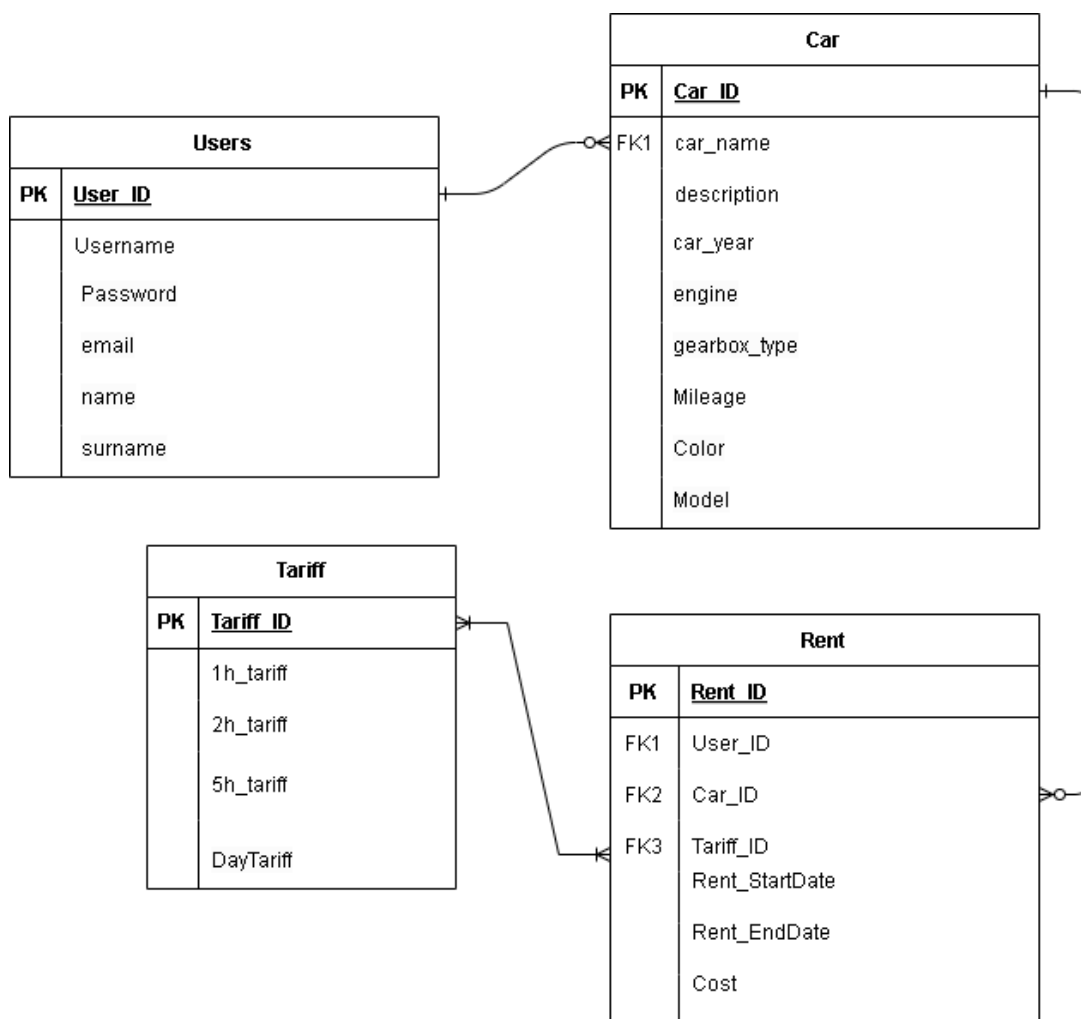
Sistēmas modelēšana un projektēšana attēlo kādi moduļi ir saistīti viens ar otru un kādi moduļi ir neatkarīgi.
mājaslapa “CarRent” visi galvenie moduļi ir ievietoti sākumlapā.(Skatīt 4.1.Modeli)

4.1. Sistēmas struktūras modelis



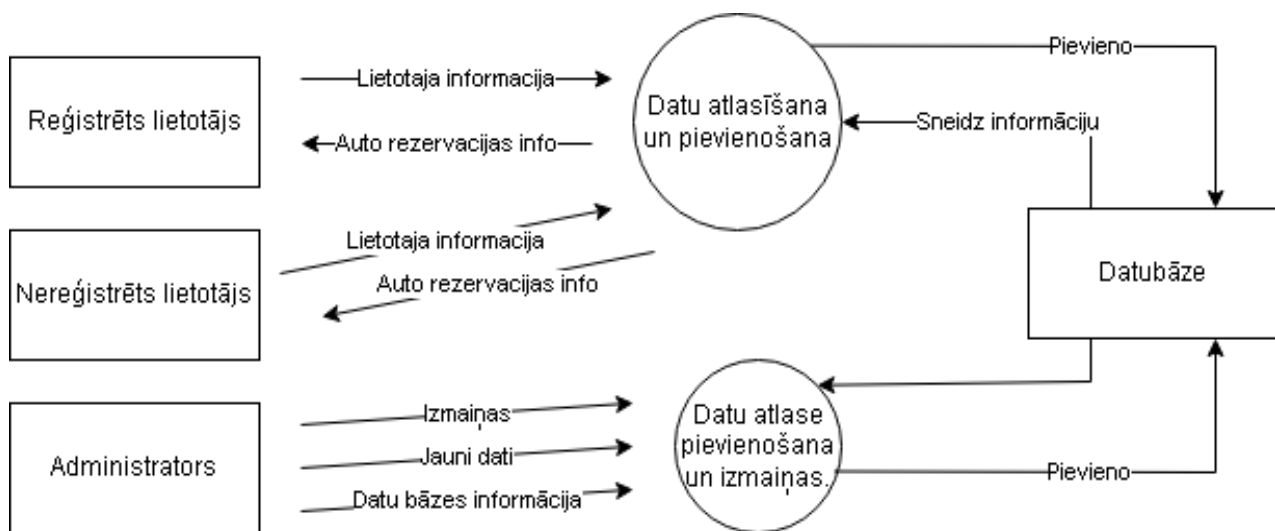
Grafisks attēlojums kas attēlo "CarRent" sistēmas moduli, un mājaslapas darbības principu.

4.2. Klašu diagramma / ER diagramma



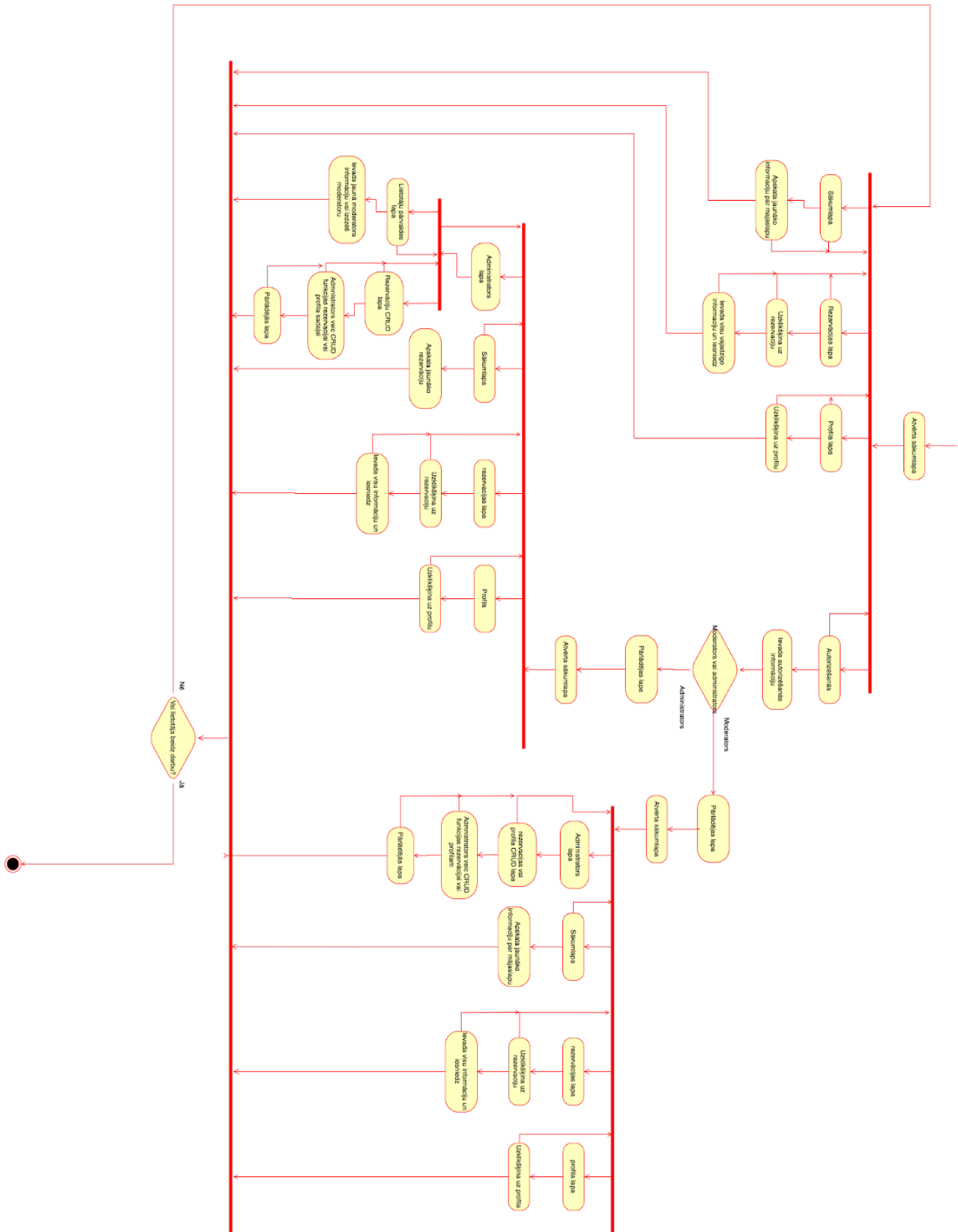
4.3. Funkcionālais un dinamiskais sistēmas modelis

Šeit ir apskatāma sistēmas datu plūsmas diagramma, kur tiek attēlots kādus datus katrs lietotāju tips apstrādā un redz (Skatīt .attēlu). Šeit ir apskatāma un lietotāja stāvokļa diagramma kur tiek attēlota datu apstrāde kura tiek veikta mājaslapas palaišanas procesā.

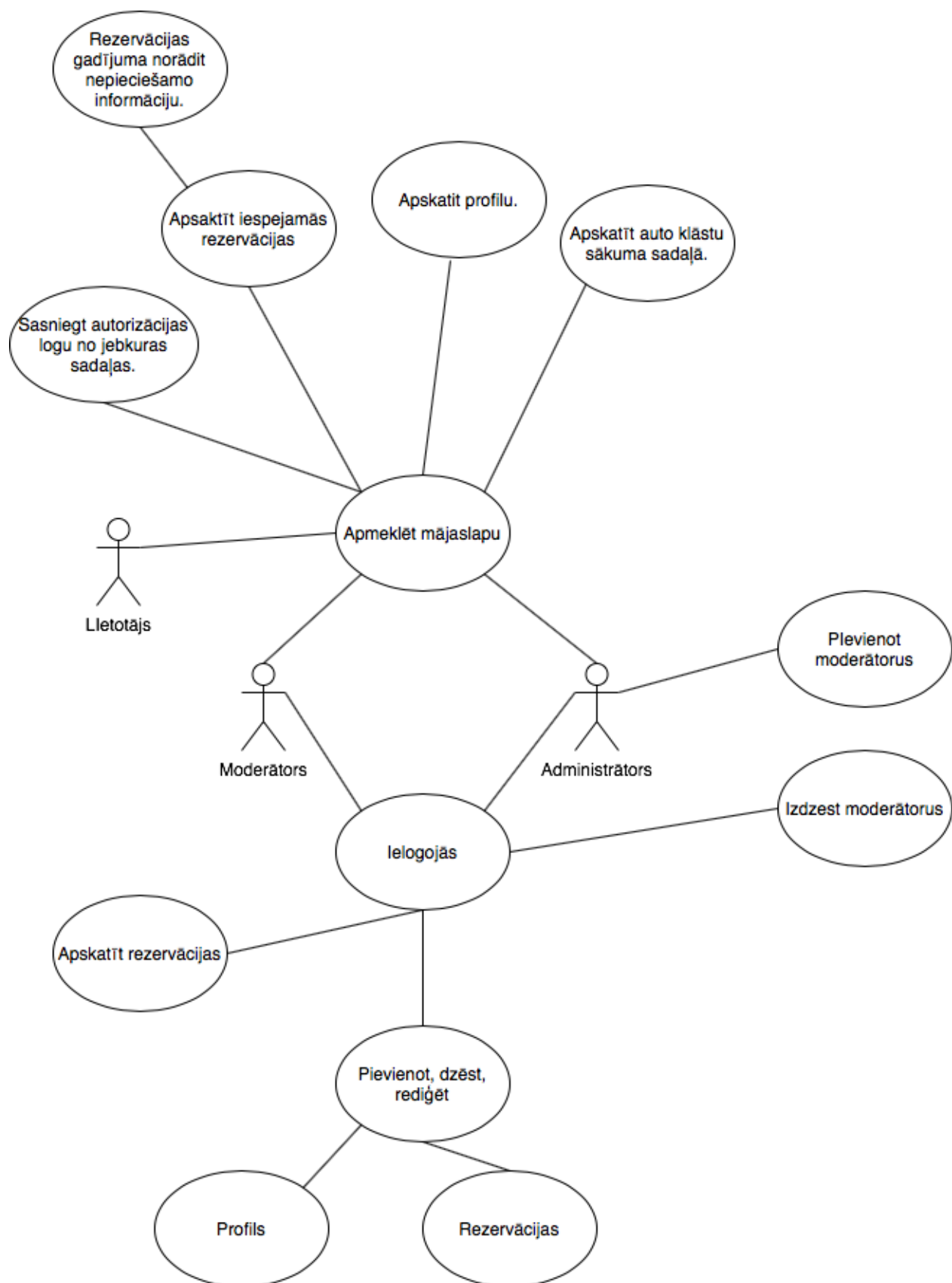


4.4. Aktivitāšu diagramma (Activity)

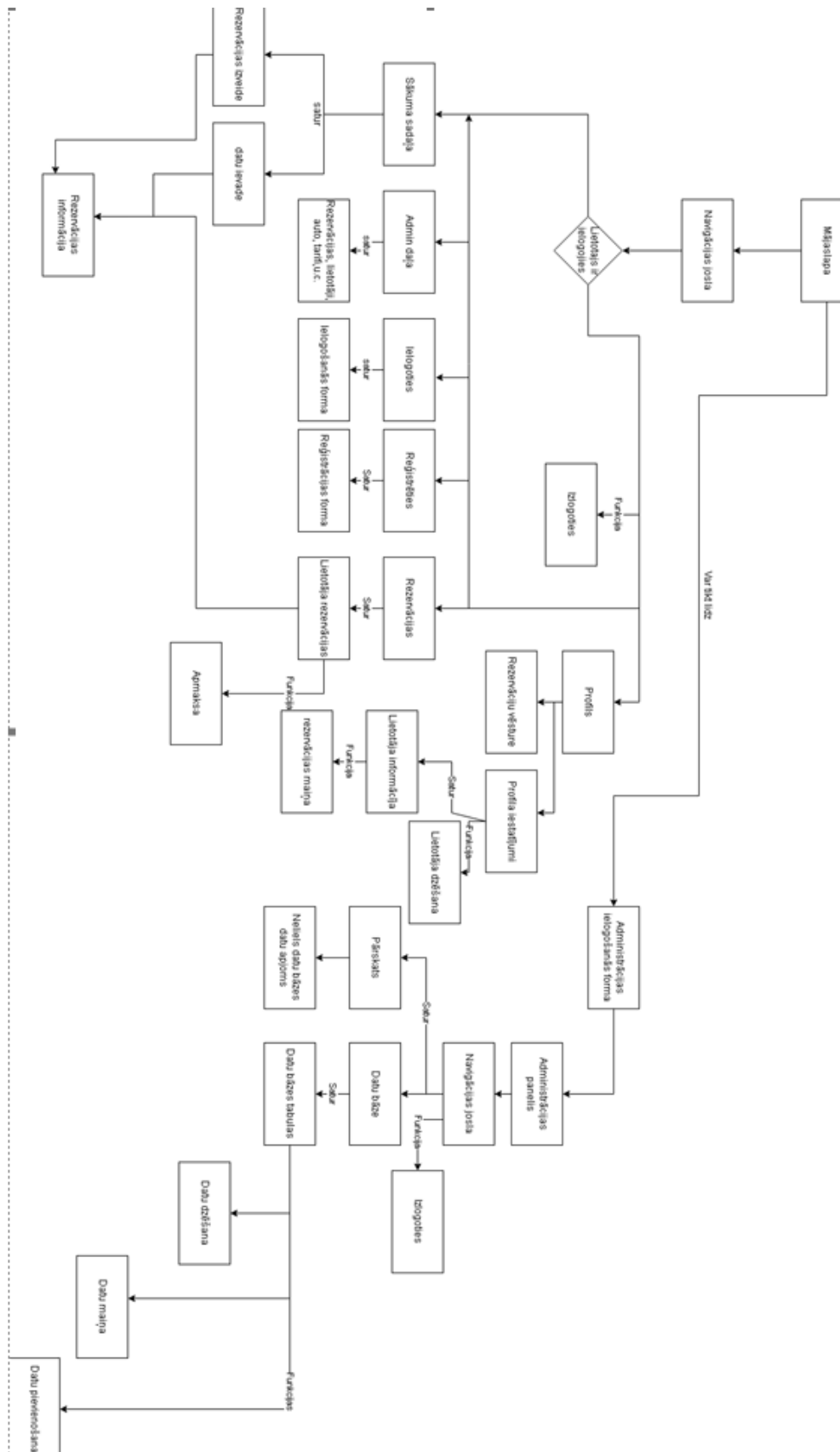
Šis ir grafiskais attēlojums kas attēlo mājaslapas "CarRent" admina un lietotāja atļautās un pieejamās darbības mājaslapā.



4.5. Lietojumgadījumu diagramma (Use Case)



Šeit ir apskatāma sistēmas moduļu diagramma (Skatīt .attēlu). Šī diagramma attēlo veidojamās sistēmas moduļus un moduļu daļas, tas ir, saskarnes daļas.



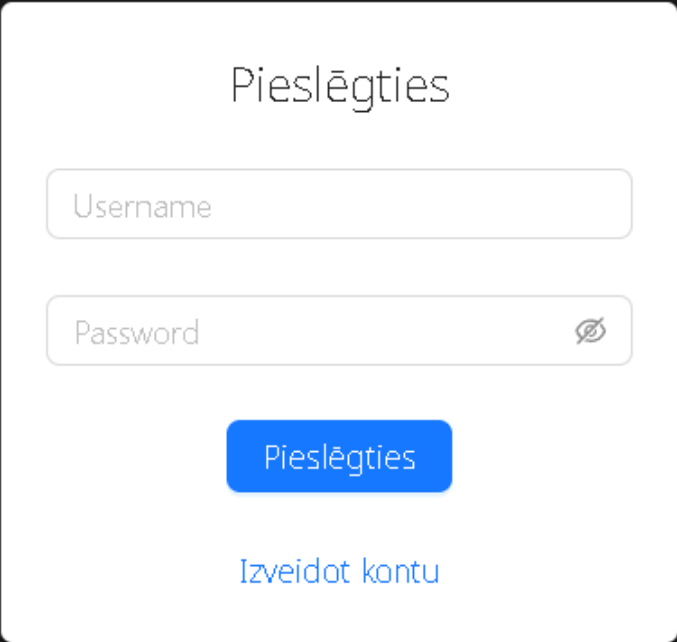
5. Lietotāju ceļvedis

Šajā nodaļā būs iespējams aplūkot kā izmantot mājaslapu "CarRent"

5.1.Reģistrēšanās sistēmā.

Kad lietotājs atver mājaslapu tam tiek piedāvāts ielogoties sistēmā, bet ja nav izveidots konts spiežam uz "Izveidot kontu" un tad lietotājs tiks novirzīts un reģistrācijas lauku, kur lietotājs ievada "lietotājvārdu" un "Paroli" , kā arī pievieno vadītāja apliecību, kur pēc tam lietotājam jāgaida kamēr administrators autentificēs lietotāju.(Skatīt 1.attēlu un 2.attēlu).

1.Attēls

The image shows a login interface on a dark background. A white rectangular box contains the text 'Pieslēgties' at the top. Below it are two input fields: 'Username' and 'Password'. The 'Password' field has a small eye icon to its right. Below the input fields is a blue button with the text 'Pieslēgties'. At the bottom of the box is a blue link that says 'Izveidot kontu'.

2.Attēls

The image shows a registration form titled "Reģistrācija" (Registration) centered on a dark background. The form is a white rounded rectangle containing three input fields and two buttons. The first field is labeled "Lietotājsvārds" (Username). The second field is labeled "Parole" (Password) and includes a toggle icon for visibility. The third field is labeled "Braukšanas apliecība" (Driving license). Below the fields is a blue button labeled "Reģistrēties" (Register), and at the bottom is a blue link labeled "Pieslēgties" (Log in).

Reģistrācija

Lietotājsvārds

Parole

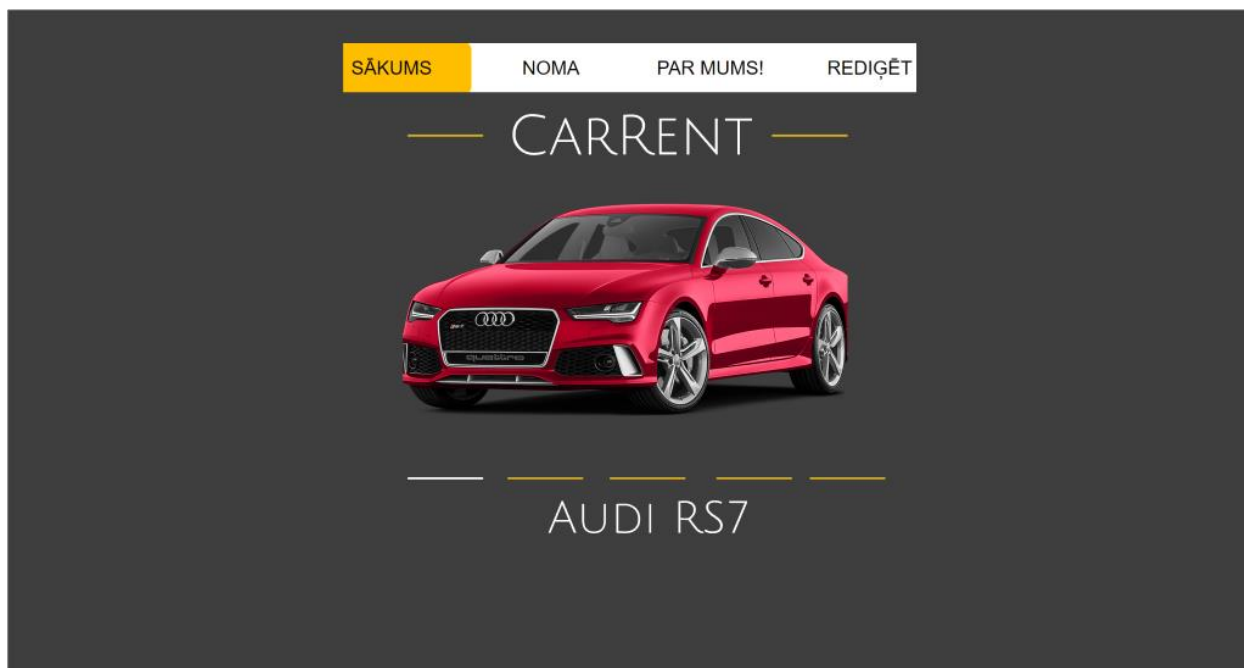
Braukšanas apliecība

Reģistrēties

Pieslēgties

5.2.Sākuma sadaļa.

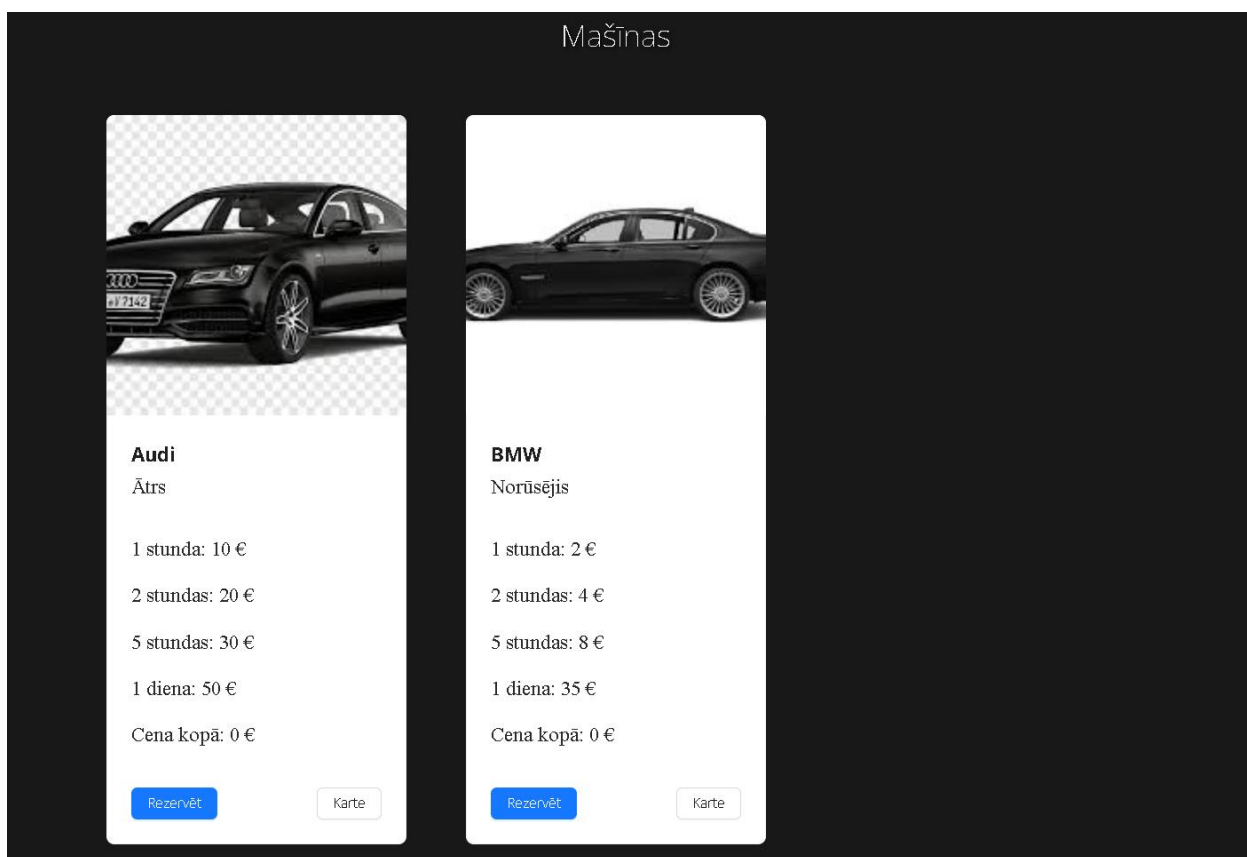
Nonākot Sākuma sadaļā, lietotājam iespējams aplūkot slideshowa auto ko var rezervēt, un kā arī nelielu info par mājaslapu, kā arī iespējams aplūkot sadaļas uz kurām lietotājs varēs doties.(Skatīt 2. un 3. attēlu).



5.3.Nomas sadaļa.

Atverot nomas sadaļu paveras auto klāsts kur lietotājam iespējams aplūkot auto ko rezervēt. Iespējams izvēlēties datumu no kura līdz kuram lietotājs vēlēšies rezervēt auto. Kad lietotājs izvēlēšies auto ko rezervēs, tad spiež uz pogas "Rezervēt".(Skatīt 4.Attēlu)

4.Attēls



Kad lietotājs noklikšķinājis uz pogas rezervēt, tad atveras logs ar izvēlēto auto, auto izvēlēta tarifa cena, un auto aprakstu, kā arī karte ar auto atrašanās vietu, pēc tam klikšķinot uz pogas apstiprināt, tad spiest pogu rezervēt un auto tiks rezervēts.

5.4.Admin sadaļa.

Šajai rediģēšanas sadaļai varēs piekļūt tikai administrators, šajā sadaļā administrators var dzēst, rediģēt pievienot auto, kā arī šeit ir lietotāju tabula kur administrators, var autentificēt lietotāju pārbaudot, lietotāja vadītāja apliecību, kā arī administratoram iespējams apskatīt visas veiktās rezervācijas. (Skatīt 6.Attēlu, 7.Attēlu un 8.Attēlu)

6.Attēls

Sākums	Noma	Par Mums	Mašīnas	Lietotāji	Rezervācijas	Izlogoties
<div> <div>Rezervēt mašīnu</div> <div>Mašīnu noma</div> </div>						
Marka	Apraksts	1h	2h	5h	1 diena	
Audi	Ātrs	10 €	20 €	30 €	50 €	
BMW	ATRAKA MASINA	2 €	5 €	8 €	20 €	
bmw	Ienis	2 €	4 €	6 €	7 €	
OPEL OMEGA	2.5TDI BMW MOTORS	69 €	123 €	69.99 €	6.99 €	

7.Attēls

Sākums	Noma	Par Mums	Mašīnas	Lietotāji	Rezervācijas	Izlogoties
Username	Statuss	Izveidots				
Uldis	Verificēts	2024-06-02 22:33:56		Parbaudīt	Deļst	
arturs	Verificēts	2024-06-02 22:42:56		Parbaudīt	Deļst	
markus	Verificēts	2024-06-03 12:57:21		Parbaudīt	Deļst	

8.Attēls

Sākums	Noma	Par Mums	Mašīnas	Lietotāji	Rezervācijas	Izlogoties
Lietotāja ID	Mašīnas ID	Intervāli		Cena	Izveidots	Darbība
665ccb4059e8b6f1213817da	665cc8c2f1bb9369cad21910	2024-06-11 04:04:00 - 2024-06-13 06:07:00		74.1 €	2024-06-02 22:44:27	
665c988be62947cb7c45722a	665d7c4f28dd2c26b252bba8	2024-06-03 04:04:00 - 2024-06-05 06:06:00		45.07 €	2024-06-03 11:18:54	
665d938102d2e3a1c38c5d0e	665cc82ff1bb9369cad218e0	2024-06-03 04:04:00 - 2024-06-05 04:04:00		100 €	2024-06-03 12:59:00	
665c988be62947cb7c45722a	665d7cc632fafcdade29ffdf3	2024-06-04 02:02:00 - 2024-06-06 05:05:00		20.1 €	2024-06-03 23:18:26	

6. Testēšanas dokumentācija

Šajā sadaļā tiks aprakstīts detalizēts testēšanas process, kas tiek veikts, lai nodrošinātu programmatūras augstu kvalitāti un darbības pareizību. Testēšanas dokumentācija ir būtiska, lai varētu veikt gan manuālo, gan automatizēto testēšanu, kā arī izmantotos testēšanas veidus, tai skaitā white box un black box testēšanu.

Manuālā testēšana ietver manuālo pārbaudi, kā arī interaktīvu mājaslapas izmēģināšanu, lai pārliecinātos par tās lietotājam draudzīgo interfeisu un funkcionalitāti. Turklāt, tiek veikta automatizētā testēšana, kur tiek izveidoti testa scenāriji, kuri tiek izpildīti automātiski, lai nodrošinātu sistēmas stabilitāti un pareizu darbību atkārtotajā veidā.

Svarīga ir arī testēšanas dokumentācija, kurā tiek detalizēti aprakstīti visi veiktie testi, to rezultāti un atbilstība sistēmas specifikācijām. Šie dati tiek ierakstīti Excel tabulā, kas nodrošina skaidru pārskatu par testēšanas procesu un tā rezultātiem.

Turklāt, šajā dokumentācijā tiks sniegts detalizēts apraksts par izmantotajiem testēšanas veidiem, piemēram, white box un black box testēšanu. White box testēšana tiek veikta, pārbaudot programmas iekšējo struktūru un kodu, lai pārliecinātos par tās pareizu darbību un efektivitāti. Savukārt black box testēšana koncentrējas uz programmas ārējo interfeisu un funkcionalitāti, lai pārliecinātos par tās pareizu darbību lietotājam.

Šāda testēšanas dokumentācija ir būtiska, lai nodrošinātu sistēmas atbilstību specifikācijām un tās uzticamību un darbības stabilitāti. Tā sniedz ieskatu par visiem veiktajiem testēšanas darbiem, to rezultātiem un nodrošina pamatu turpmākai sistēmas uzlabošanai un pilnveidošanai.

6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums

Testēšanas procesā tiks plaši izmantotas gan white box, gan black box testēšanas metodes. Šīs divas pieejas nodrošina kompaktu un pilnīgu pārbaudi, sniedzot ieskatu sistēmas darbībā un funkcionalitātē no dažādiem skatupunktiem.

White box testēšana sniegs ieskatu sistēmas iekšējā struktūrā un kodā. Tās galvenais mērķis ir izpētīt programmas loģiku un darbību, lai identificētu iespējamus defektus un optimizētu kodu. Šī metode palīdzēs nodrošināt efektivitāti un drošību, garantējot, ka programmatūra darbojas pareizi un ir pasargāta no potenciālajiem drošības riskiem.

No otras puses, black box testēšana koncentrēsies uz sistēmas uzvedību no gala lietotāja perspektīvas. Šajā metodē testētāji pārbaudīs sistēmas funkcionalitāti un saskarni, neņemot vērā tās iekšējo struktūru. Galvenais mērķis ir nodrošināt, ka programmatūra atbilst specifikācijām un nodrošina lietotājiem kvalitatīvu un intuitīvu pieredzi.

Testu rezultātu ierakstīšanai un organizēšanai tiks izmantota Excel tabula. Šī tabula nodrošinās skaidru struktūru un vieglu sapratni gan testētājiem, gan izstrādātājiem. Tajā tiks ierakstīti visi veiktie testi, to rezultāti un atbilstība specifikācijām, kas nodrošinās pārskatāmu un efektīvu testēšanas procesu.

Šāda pieejas kombinācija nodrošinās pilnīgu pārbaudi un augstu programmatūras kvalitāti. Ar white box un black box testēšanas metodēm tiks veikta visaptveroša sistēmas pārbaude, savukārt Excel tabula nodrošinās efektīvu rezultātu dokumentēšanu un pārskatāmību. Tādējādi būs optimizēts gan testēšanas process, gan rezultātu dokumentācija, nodrošinot augstu programmatūras kvalitāti un uzticamību.

6.2. Testpiemēru kopa

6.Tabula

Identifikatoru atšifrējums	
Piemērs: TP.SES.LOG.01	
PR	Prasība
TP	Testpiemērs
Nosaka piederību modulim:	
SAK	Sākums
LOG	Ielogošanās
SAKSAD	Sākuma sadaļa
NOM	Rezervācija
RED	Rediģēt ierakstu
STA	Startēšana
AIZ	Aizvērsšana
LOG	Ielogošanās
SES	Sesija
IZLOG	Izlogošanās
LP	Laika Perioda maiņa
FIL	Filtrācija
PIEV	Pievienot ierakstu
DZ	Dzēst ierakstu

Identifikātoru atšifrējums

6.3. Prasības

Prasības ID	Prasība
PR.01.	Mājaslapas „CarRent” startēšana
Sākums	
PR.SAK.01.	Mājaslapas „CarRent” aizveršana
PR.SAKSAD.02.	Sākuma sadaļas ielāde
Mājaslapa	
PR.SAKSAD.01.	Atgriešanās uz sākuma sadaļu
PR.NOM.02.	Auto rezervācija mājaslapā "CarRent"
PR.RED.03.	Ierakstu rediģēšana admin sadaļā
PR.LOG.04.	Ielogošanās mājaslapā "CarRent"
PR.SES.05.	Sesijas uzsākšana
PR.IZLOG.06.	Izlogošanās no mājaslapas "CarRent"
PR.LP.07.	Laika perioda maiņa
PR.FIL.08.	Auto filtrācija
PR.PIEV.09.	Ierakstu pievienošana
PR.DZ.10.	Ierakstu dzēšana

6.4. Testpiemēri

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosauējumi	Testpiemēra apraksts Black Box	Testpiemēra izpildes soļi	Testpiemēra izpildes dati	Testpiemēra sagatavošana	Pasūtīt ID
TP.STA.01	Programmas startēšana	Savā pārlūkprogrammā atver mājaslapu	Mājaslapas palašana kādā no pārlūkpi	1) Peles kreisā taustiņa dubultklikšķis	Peles kreisā taustiņa klikšķis	Startēta mājaslapa "Car	PR.01.
TP.SAK.01	Programmas apturēšana	Jābūt startētai mājaslapai "CarRent"	Uz pārlūkprogrammas startētas mājas	1) Peles kreisā taustiņa klikšķis uz p	Peles kreisā taustiņa klikšķis	Aizvērta mājaslapa "Car	PR.SAK.01.
TP.SAK.02	Programmas apturēšana	Jābūt startētai mājaslapai "CarRent"	Uz pārlūkprogrammas startētas mājas	1) Peles labā taustiņa klikšķis uz pog	Peles labā taustiņa klikšķis	Mājaslapas darbība ne	PR.SAK.01.
TP.SAKSAD.01	Sākuma sadaļas atvēršana	Jābūt startētai mājaslapai "CarRent"	Sākuma sadaļas atvēršana	1) Peles kreisā taustiņa klikšķis uz p	Peles kreisā taustiņa klikšķis	Atveras sākuma sadaļa	PR.SAKSAD.02
TP.SAKSAD.02	Sākuma sadaļas atvēršana	Jābūt startētai mājaslapai "CarRent"	Sākuma sadaļas atvēršana	1) Peles kreisā taustiņa klikšķis uz p	Peles labā taustiņa klikšķis	Programmas darbība n	PR.SAKSAD.02
TP.NOM.02	Rezervācijas sadaļas ielāde	Jābūt startētai mājaslapai "CarRent"	Pārsūtģšanās no jebkuras izvēlnes uz	1) Peles kreisā taustiņa klikšķis uz	Peles kreisā taustiņa klikšķis	Aizvērta jebkura cita s	PR.NOM.02
TP.NOM.02	Rezervācijas sadaļas ielāde	Jābūt startētai mājaslapai "CarRent"	Pārsūtģšanās no jebkuras izvēlnes uz	1) Peles labā taustiņa klikšķis uz pog	Peles labā taustiņa klikšķis	Programmas darbība	PR.NOM.03
TP.SAKSAD.SAK.01	Aizvērtāšanās uz sākuma sadaļu	Jābūt ielādētai sākuma sadaļai	Pārsūtģšanās no jebkuras sadaļas uz s	1) Peles kreisā taustiņa klikšķis uz p	Peles kreisā taustiņa klikšķis	No jebkuras sadaļas pā	PR.SAKSAD.01.
TP.RED01	Ierakstu rediģēšana admi	1) Jābūt admi	Ievadīto datu rediģēšana	1) Rediģēšanas ikonas nospiešana	Rediģēšanas ikonas nospiešana	Rediģēšanas pogas n	PR.RED.01.
TP.DZ.02	Ierakstu dzēšana admi	1) Jābūt admi	Ievadīto datu dzēšana	1) dzēšanas ikonas nospiešana	Dzēšanas ikonas nospiešana	Dzēšanas pogas nospie	PR.DZ.01.
PR.IZLOG.06	Lietotāja izlogošanās	1) Jābūt reģistrētam	Lietotāja izlogošanās no mājaslapas	LogOut pogas nospiešana	LogOut pogas nospiešana	Nospiežot pogu Log	PR.IZLOG.08.
PR.LOG.06	Lietotāja ielogošanās	1) Jābūt reģistrētam	Lietotāja ielogošanās sistēmā	Nepieciešamo lauku aizpildīšana un	Nepieciešamo lauku aizpildīšana	Nospiežot pogu "Pieslē	PR.LOG.08.
PR.LP.01.	Auto rezervācijas tarifa laika	1) Jābūt reģistrētam	Tarifa laika maiņa	Kreisā klikšā nospiešana uz kalendār	Kreisā klikšā nospiešana uz ka	Tarifa laika maiņa līdz	PR.LP.09
PR.LP.02.	Auto rezervācijas tarifa laika	1) Jābūt reģistrētam	Tarifa laika maiņa	Labā klikšā nospiešana uz kalendārā	Labā klikšā nospiešana uz kal	tarifis netiek mainīts	PR.LP.09
TP.PIEV.02	Ierakstu pievienošana admi	1) Jābūt admi	Datu pievienošana	1) Pievienošanas ikonas nospiešana	Pievienošanas ikonas nospiešana	Pievienošanas pogas	PR.PIEV.01.

6.5. Testēšanas žurnāls

Testēšanas ID	Datums	Testpiemēra ID	Testpiemēra nosaukums	Testēja	Statuss	Kļūdas ziņojums	Kļūdas ziņojuma Nr.
Black Box							
TZ.B.01	15.5.2024	TP.SAK.01	Mājaslapas startēšanas	Dāvis Barons	Veiksmīgs		
TZ.B.02	15.5.2024	TP.SAK.AI Z.01	Mājaslapas aizveršana	Dāvis Barons	Veiksmīgs		
TZ.B.03	15.5.2024	TP.SAK.AI Z.02	Mājaslapas aizveršana	Dāvis Barons	Veiksmīgs		
TZ.B.04	15.5.2024	TP.SAKSA D.01	Sākuma sadaļas ielāde	Dāvis Barons	Veiksmīgs		
TZ.B.05	15.5.2024	TP.SAKSA D.02	Sākuma sadaļas ielāde	Dāvis Barons	Veiksmīgs		
TZ.B.06	15.5.2024	TP.NOM.01	Rezervācijas sadaļas ielāde	Dāvis Barons	Veiksmīgs		
TZ.B.07	15.5.2024	TP.NOM.02	Rezervācijas sadaļas ielāde	Dāvis Barons	Veiksmīgs		
TZ.B.08	15.5.2024	TP.SAKSA D.SAK.01	Atgriešanās uz sākuma sadaļu no jebkuras sadaļas	Dāvis Barons	Veiksmīgs		
TZ.B.09	15.5.2024	TP.RED.01	Ierakstu rediģēšana admin sadaļā	Dāvis Barons	Veiksmīgs		
TZ.B.07	15.5.2024	TP.DZ.02	Ierakstu dzēšana admin sadaļā	Dāvis Barons	Veiksmīgs		
TZ.B.10	15.5.2024	PR.IZLOG. 06.	lietotāju izlogošanās	Dāvis Barons	Veiksmīgs		

TZ.B.11	15.5.2024	PR.LOG.06.	lietotāju ielogošanās	Dāvis Barons	Veiksmīgs		
TZ.B.12	15.5.2024	PR.LP.01.	tarifa laika maiņa	Dāvis Barons	Veiksmīgs		
TZ.B.13	15.5.2024	PR.LP.02.	tarifa laika maiņa	Dāvis Barons	Veiksmīgs		
TZ.B.14	15.5.2024	TP.PIEV.02	Uzvara loga parādīšanā noņemot visus transportlīdzekļus paredzētajās vietās	Dāvis Barons	Veiksmīgs		
TZ.B.15	15.5.2024	PR.FIL.08.	Spēles atsākšana pēc uzvaras	Dāvis Barons	Veiksmīgs		
White Box							
TZ.W.01	16.5.2024	TP.STA.03	Mājaslapas startēšana	Dāvis Barons	Veiksmīgs		
TZ.W.02	16.5.2024	TP.SAK.01	Mājaslapas darbības apturēšana	Dāvis Barons	Veiksmīgs		
TZ.W.03	16.5.2024	TP.SAKSA D.01	Sadaļas "Sākums " atveršana	Dāvis Barons	Veiksmīgs		
TZ.W.04	16.5.2024	TP.NOM.02	Sadaļas "Rezervācija " atveršana	Dāvis Barons	Veiksmīgs		

TZ.W.05	16.5.2024	TP.SAKSA D.SAK.01	Atgriešanās uz sākuma sadaļu no jebkuras sadaļas	Dāvis Barons	Veiksmīg s		
TZ.W.06	16.5.2024	TP.RED.01	Tabulu ierakstu redigēšana	Dāvis Barons	Veiksmīg s		
TZ.W.07	16.5.2024	TP.DZ.02	Ierakstu dzēšana	Dāvis Barons	Veiksmīg s		
TZ.W.08	16.5.2024	PR.IZLOG. 06.	Lietotāju izlogošanās	Dāvis Barons	Veiksmīg s		
TZ.W.09	16.5.2024	PR.LOG.06.	Lietotāju ielogošanās	Dāvis Barons	Veiksmīg s		
TZ.W.10	16.5.2024	PR.LP.01.	Rezervācija tarifa maiņa	Dāvis Barons	Veiksmīg s		
TZ.W.11	16.5.2024	TP.PIEV.02	Auto pievienošana rezervācijai	Dāvis Barons	Veiksmīg s		
TZ.W.12	16.5.2024	PR.FIL.08.	Auto filtrēšana	Dāvis Barons	Veiksmīg s		

//

7. Lietoto terminu un saīsinājumu skaidrojumi

Termins	Skaidrojums
Header	Augšējā daļa vai sākuma daļa kādā dokumentā, lapā vai datu struktūrā
Footer	Apakšējā daļa vai beigu daļa kādā dokumentā, lapā vai datu struktūrā
Button	Poga
WEB	Mājaslapa internetā
Admin	Persona ar papildu piekļuvi un kontroli pār sistēmu.
Moderators	Persona, kas pārvalda un uzrauga saturu.
Saskārne	Lietotāja interfeiss, kurā notiek mijiedarbība ar sistēmu.
WhiteBox	Testēšanas metode, kur testētājam ir pilna informācija par sistēmas kodu un darbību.
BlackBox	Testēšanas metode, kur testētājam nav informācijas par sistēmas kodu un struktūru.

8. Secinājumi

Sasniegums, rezultāta novērtējums

Mājaslapa "CARRENT" veiksmīgi nodrošina lietotājiem iespēju rezervēt automašīnas tiešsaistē, piedāvājot vienkāršu un intuitīvu lietotāja interfeisu.

Izvirzīto uzdevumu sasniegšanas analīze

Kas izdevies:

- **Lietotāja interfeiss:** Dizains ir moderns un ērti lietojams, nodrošinot vieglu navigāciju.
- **Rezervācijas funkcionalitāte:** Sistēma nodrošina visas nepieciešamās funkcijas rezervācijas veikšanai, ieskaitot automašīnu izvēli, rezervācijas datumu un laiku izvēli.

Kas nē:

- **Papildus funkcionalitātes trūkums:** Daži plānotie uzlabojumi, piemēram, automātiskās atgādinājumu sistēmas un verifikācijas programmas integrācija, vēl nav realizēti.

Darba apjoms

Sistēmas struktūra:

- **Failu skaits:** Aptuveni 60 faili.
- **Moduļu skaits:** 6 galvenie moduļi
- **Funkciju skaits:** Ap 20 funkcijas, kas nodrošina dažādu mājaslapas daļu darbību.

Koda apjoms:

- **Pašrakstīto komandrindu skaits:**
 - Dizainam: 1000 rindas apmēram.
 - Funkcionalitātei: 1500 rindas.
- **Kopīgais komandrindu skaits (ar ģenerētām un kopīgi veidotām):**
- Aptuveni 3000 rindas.

Testēšana

Programma tika testēta uz dažādiem pārlūkiem (Chrome, Firefox, Microsoft Edge). Testēšana bija sekmīga, un tika novērstas visas atrastās kļūdas.

Problēmas un izaicinājumi

Izstrādes laikā bija vairākas problēmas:

- **Datu bāzes veidošana:** Bija tehniskie izaicinājumi saistībā ar datu bāzes veidošanu, jo to sāku veidot kad puslīdz bija gatavs projekts.
- **Laika ierobežojums:** Liels izaicinājums bija tas, ka bija viss saplānots izveidot mājaslapu "CarRent" līdz eksāmena dienai, bet tika nosacīti nodošanas termiņš samazināts.

Nākotnes ieceres

- **Papildu funkcionalitāte:** Integrēt lojalitātes programmu un automātiskās atgādinājumu sistēmas.
- **Uzlabot integrāciju:** Pievienot maksājumu un apdrošināšanas pakalpojumu integrācijas iespējas.
- **Paplašināt analītiku:** Iekļaut uzlabotas datu analīzes un pārskatu veidošanas iespējas.

9. Literatūras un informācijas avotu saraksts

1. <https://ant.design/components/overview>
2. <https://www.w3schools.com/REACT/DEFAULT.ASP>
3. <https://stackoverflow.blog/2021/10/20/why-hooks-are-the-best-thing-to-happen-to-react/>
4. <https://www.figma.com/files/recent-and-sharing/recently-viewed?fuid=1362383268793938484>
5. <https://www.patterns.dev/react/>
6. <https://reacttraining.com/>
7. <https://www.freecodecamp.org/>
8. <https://www.php.net/manual/en/index.php>

Pielikums

Tarifu aprēķināšanas funkcija.(Skatīt 9.Attēlu)

```
useEffect(() => {
  if (
    reservationInterval &&
    reservationInterval[0] &&
    reservationInterval[1]
  ) {
    const intervalStart = reservationInterval[0].toDate(); // Convert Moment.js object to JavaScript Date object
    const intervalEnd = reservationInterval[1].toDate(); // Convert Moment.js object to JavaScript Date object

    // Calculate the duration in milliseconds
    const durationMilliseconds = intervalEnd - intervalStart;
    // Convert duration to hours
    const durationHours = durationMilliseconds / (1000 * 60 * 60);

    let total = 0;

    // Calculate total price based on the duration
    let remainingHours = durationHours;
    const oneDayPrice = car.OneD;
    const fiveHourPrice = car.FiveH;
    const twoHourPrice = car.TwoH;
    const oneHourPrice = car.OneH;

    // Calculate full days
    const fullDays = Math.floor(durationHours / 24);
    total += fullDays * oneDayPrice;

    // Calculate remaining hours after full days
    remainingHours -= fullDays * 24;

    // Calculate 5-hour blocks
    const fiveHourBlocks = Math.floor(remainingHours / 5);
    total += fiveHourBlocks * fiveHourPrice;
    remainingHours -= fiveHourBlocks * 5;

    // Calculate 2-hour blocks
    const twoHourBlocks = Math.floor(remainingHours / 2);
    total += twoHourBlocks * twoHourPrice;
    remainingHours -= twoHourBlocks * 2;

    // Calculate remaining hours
    total += remainingHours * oneHourPrice;

    total = parseFloat(total.toFixed(2));

    setTotalPrice(total);
  }, [reservationInterval, car]);
}
```

9.Attēls

Lietotāju saglabāšana sistēmā.(Skatīt 10.attēlu)

10.Attēls

```
// Models komponente
▼ const Models = () => {
  const [users, setUsers] = useState([]); // Definē users stāvokli un setUsers funkciju, lai atjauninātu šo stāvokli

  // Funkcija, lai iegūtu lietotāju datus no servera
  ▼ const fetchUsers = async () => {
    try {
      const token = localStorage.getItem("token"); // Iegūst token'u no localStorage
      if (!token) {
        console.error("Token not found"); // Ja token nav atrasts, izvada kļūdas ziņu
        return;
      }

      // Veic fetch pieprasījumu, lai iegūtu lietotāju datus
      const response = await fetch("http://localhost:8000/api/lietotaji", {
        headers: {
          Authorization: `Bearer ${token}`, // Pievieno token'u pieprasījuma galvenē
        },
      });
      if (response.ok) {
        const data = await response.json(); // Ja pieprasījums veiksmīgs, parsē atbildi kā JSON
        setUsers(data); // Atjaunina users stāvokli ar iegūtajiem datiem
      } else {
        console.error("Failed to fetch users"); // Ja pieprasījums nav veiksmīgs, izvada kļūdas ziņu
      }
    } catch (error) {
      console.error("Error fetching users:", error); // Apstrādā kļūdas gadījumā
    }
  };

  // useEffect hook, lai izpildītu fetchUsers funkciju pēc komponentes montēšanas
  useEffect(() => {
    fetchUsers(); // Izsauc fetchUsers funkciju
  }, []); // Tukšs masīvs kā atkarība, lai šī funkcija izsauktos tikai vienu reizi pēc komponentes montēšanas
```

Kartes funkcionalitāte(Skatīt 11. 12.Attēlu)

11.Attēls

```
// Definē Map komponentu ar nepieciešamajiem rekvizītiem
const Map = ({ center, zoom, displayMarker, allowMarkerAddition, onMapClick }) => {
  const mapRef = useRef(null); // Izveido atsauci uz karti
  const [marker, setMarker] = useState(null); // Izveido stāvokli marķierim

  // Efekts, kas izpildās komponenta montēšanas un atjaunošanas laikā
  useEffect(() => {
    if (!mapRef.current) {
      // Ja karte vēl nav izveidota, izveido to un iestata sākotnējo skatu
      mapRef.current = L.map("map").setView(center, zoom);

      // Pievieno OpenStreetMap slāni
      L.tileLayer("https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {
        attribution:
          '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
      }).addTo(mapRef.current);

      // Pievieno marķieri, ja displayMarker ir paties
      if (displayMarker) {
        addMarker(center);
      }
    } else {
      // Ja karte jau ir izveidota, atjauno tās skatu
      mapRef.current.setView(center, zoom);
    }
  });

  // Funkcija, kas apstrādā klikšķi uz kartes
  const handleClick = (event) => {
    const { lat, lng } = event.latlng; // Iegūst klikšķa koordinātas

    if (!allowMarkerAddition) {
      return; // Ja marķieru pievienošana nav atļauta, pārtrauc funkciju
    }

    if (marker) {
      mapRef.current.removeLayer(marker); // Ja marķieris jau eksistē, noņem to no kartes
    }

    addMarker([lat, lng]); // Pievieno jaunu marķieri klikšķa vietā

    if (onMapClick) {
      onMapClick({ lat, lng }); // Izsauc onMapClick funkciju, ja tā ir norādīta
    }
  };

  mapRef.current.on("click", handleClick); // Pievieno klikšķa notikumu kartē

  return () => {
    mapRef.current.off("click", handleClick); // Noņem klikšķa notikumu komponenta atmontēšanas laikā
  };
}, [center, zoom, displayMarker, allowMarkerAddition, onMapClick]); // Atkārtoti izpilda efektu, ja mainās kāds no šiem rekvizītiem
```

```

mapRef.current.on("click", handleClick); // Pievieno klikšķa notikumu kartē

return () => {
  mapRef.current.off("click", handleClick); // Noņem klikšķa notikumu komponenta atmontēšanas laikā
};
}, [center, zoom, displayMarker, allowMarkerAddition, onMapClick]); // Atkārtoti izpilda efektu, ja mainās kāds no šiem rekvizītiem

// Funkcija marķiera pievienošanai kartē
const addMarker = (position) => {
  const customIcon = new Icon({
    iconUrl: markerIconPng, // Marķiera ikonas URL
    iconSize: [25, 41], // Ikonas izmērs
    iconAnchor: [12, 41], // Ikonas enkura punkts
  });

  const newMarker = L.marker(position, { icon: customIcon }).addTo(
    mapRef.current
  ); // Izveido un pievieno jaunu marķieri kartē
  setMarker(newMarker); // Atjauno marķiera stāvokli
};

return (
  // Atgriež HTML struktūru kartes attēlošanai
  <div style={{ width: "100%", height: "300px" }}>
    <div id="map" style={{ width: "100%", height: "100%" }}></div>
  </div>
);
};

export default Map; // Eksportē Map komponentu

```

Saite uz projekta GitHub.

<https://github.com/DavisA4/Eksamens>