**RĪGAS VALSTS TEHNIKUMS**

DATORIKAS NODAĻA

Izglītības programma: Programmēšana

**KVALIFIKĀCIJAS DARBS**

**Automašīnu izmēģinājuma braucienu   
rezervācijas platforma**

Paskaidrojošais raksts 21 lpp.

Audzēknis: Ivars Levāns

Prakses vadītājs: Igors Litvjakovs

Nodaļas vadītājs: Normunds Barbāns

**Rīga 2024**

**Anotācija**

Kvalifikācijas darbs raksturo procesu “Drive Wise” platformas izstrādē, kas nodrošinās plašu automašīnu katalogu nomas rezervācijām, kā arī padziļinātu platformas datu rediģēšanas funkcionalitāti administratoriem. Platformas izstrādes mērķis ir nodrošināt intuitīvu nomas procesu lietotājiem, marķējumus par sistēmā pieejamajiem automobiļiem, iespēju rezervēt noteiktus laika periodu, saņemt aprēķinu par pakalpojuma izmaksām un pārvaldīt veiktās rezervācijas datus. Frontend izstrādē tika izvēlēts Next.js, kas ļauj efektīvi izstrādāt servera pusē renderētas lietotnes, nodrošinot ātru lapas ielādi un uzlabotu lietotāja pieredzi. Tāpat tika izmantots Tailwind, kas nodrošina ātru un efektīvu stilizāciju. Backend izstrādē tika izmantots TypeScript, kas nodrošina statisko tipizāciju, REST API realizāciju, kopumā uzlabojot koda lasāmību un samazinot kļūdu skaitu izstrādes procesā. Kā datubāze tika izvēlēta PostgreSQL, kas ir pazīstama ar savu uzticamību un spēju apstrādāt lielus datu apjomus. Papildus tika izmantots Prisma ORM, kas atvieglo datu modeļu pārvaldību, kā arī nodrošina efektīvu datu mijiedarbību ar Next.js un PostgreSQL. Visual Studio Code tika izvēlēta kā izstrādes vide dēļ plašā rīku un paplašinājumu klāsta un valodu atbalsta.

Kvalifikācijas darba raksts sastāv no šādām daļām: ievads, uzdevuma nostādne, prasību specifikācija, uzdevuma risināšanas līdzekļu izvēles pamatojums, programmatūras produkta modelēšana un projektēšana, datu struktūru apskats, lietotāja ceļvedis, nobeigums un pielikumi. Ievadā tiek skaidrots, kāpēc ir nepieciešams izveidot minēto programmsistēmu, tās nozīmīgumu un praktisko pielietojumu. Uzdevuma nostādnē ir izklāstīti galvenie kvalifikācijas darba mērķi un specifiskie uzdevumi. Prasību specifikācijā ir definētas sistēmas funkcijas un drošības prasības, kā arī detalizēti aprakstīta informācija, ko sistēma saņem un izsniedz lietotājam. Uzdevuma risināšanas līdzekļu izvēles pamatojumā ir izskaidrots izvēlēto programmēšanas valodu, teksta redaktoru, relāciju datubāžu pārvaldības sistēmu un datu apmaiņas programmu izmantojums sistēmas izveidē. Programmatūras produkta modelēšanas un projektēšanas apskatā tiek pētītas datu plūsmas diagrammas, kas ilustrē sistēmas funkcionalitāti, ER diagrammas datubāzes struktūras vizuaizācijai un sistēmas arhitektūras detalizēts apraksts. Datu struktūru apskats ietver visu datu tabulu struktūras aprakstu un attiecīgo tabulu savstarpējo saistību shēmu. Lietotāja ceļvedī ir izklāstīts sistēmas vizuālais noformējums un sniegti norādījumi par tās efektīvu izmantošanu.

Kopumā kvalifikācijas darba apjoms ir \_\_ lpp., kurā ietilpst \_\_attēli, \_\_tabulas un \_\_pielikumi.

**ANNOTATION**

The qualification paper describes the process of developing the Drive Wise platform, which will provide an extensive car rental catalogue for bookings as well as advanced editing functionality for administrators. The aim of the platform development is to provide an intuitive rental process for users, labelling of the cars available in the system, the possibility to book specific time periods, to get an estimate of the cost of the service and to manage the data of the booked reservation. For the frontend development, Next.js was chosen, which allows efficient development of server-side rendered applications, ensuring fast page loading and an improved user experience. Tailwind was also used, which provides fast and efficient styling. TypeScript was used for backend development, providing static typing, REST API implementation, improving overall code readability and reducing errors during development. PostgreSQL was chosen as the database, which is known for its reliability and ability to handle large amounts of data. Additionally, Prisma ORM was used to facilitate the management of data models, as well as to ensure efficient data interaction with Next.js and PostgreSQL. Visual Studio Code was chosen as the development environment due to its wide range of tools and extensions and language support.

The qualification paper consists of the following parts: introduction, problem statement, requirements specification, justification for the choice of the means of solving the problem, modelling and design of the software product, overview of data structures, user guide, conclusion and annexes. The introduction explains why it is necessary to develop the software system, its importance and practical applications. The assignment outlines the main objectives and specific tasks of the qualification. The requirements specification defines the functions and security requirements of the system and describes in detail the information that the system receives and provides to the user. The rationale for the choice of the means of solving the problem explains the use of the chosen programming languages, text editors, relational database management systems and data exchange programs in the development of the system. The software product modelling and design review examines data flow diagrams illustrating the functionality of the system, ER diagrams to visualise the database structure and a detailed description of the system architecture. The overview of data structures includes a description of the structure of all data tables and a diagram of the interrelationships between the relevant tables. A user guide outlines the visual design of the system and provides guidance on its effective use.

The total length of the qualification work is \_\_ pages, including \_\_images, \_\_ tables and \_\_ attachments.

**Saturs**

[IEVADS](#_Toc148995681) 3

[1. UZDEVUMA NOSTĀDNE 4](#_Toc148995682)

2. PRASĪBU SPECIFIKĀCIJA 6

2.1. Ieejas un izejas informācijas apraksts 6

*2.1.1. Ieejas informācijas apraksts* 6

*2.1.2. Izejas informācijas apraksts* 7

*2.1.3. Ārējās informācijas apraksts* 7

2.2. Funkcionālās prasības 8

2.3. Nefunkcionālās prasības 9

3. UZDEVUMU RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS 11

4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA 12

4.1. Sistēmas struktūras modelis 12

*4.1.1. Sistēmas arhitektūra* 12

*4.1.2. Sistēmas ER modelis* 13

4.2. Funkcionālās sistēmas modelis 14

*4.2.1. Datu plūsmu modelis* 14

5. DATU STRUKTŪRAS APRAKSTS 20

5.1. Tabulu relāciju shēma 21

6. LIETOTĀJA CEĻVEDIS 20

6.1. Sistēmas prasības aparatūrai un programmatūrai 21

6.2. Sistēmas instalācija un palaišana 21

6.3. Programmas apraksts 21

*6.3.1. Autentifikācija* 12

*6.3.2. Rezervācija* 12

*6.3.4. Profila labošana* 12

*6.3.4. Sistēmas pārvaldība* 12

6.4. Testa piemērs 21

NOBEIGUMS 22

INFORMĀCIJAS AVOTI 23

**IEVADS**

Mūsdienās ir ļoti svarīgi izmantot tehnoloģijas, lai uzlabotu mūsu dzīves kvalitāti un padarītu ikdienu efektīvāku. Šajā kontekstā vēlos izstrādāt risinājumu - Auto izīrēšanas platforma izmēģinājumu braucienu rezervēšanai.

Pašlaik, bez šīs sistēmas izmantošanas, cilvēkiem, kuriem ir interese iegādāties automašīnu, ir jāsaskaras ar vairākām sarežģītām problēmām. Tirgū pieejamā informācija par automašīnām var būt izkaisīta, nepilnīga un grūti salīdzināma. Lietotājiem trūkst skaidras priekšstata par automobiļu marku un modeļu plašo klāstu, un viņi ir spiesti pavadīt vairāk laika meklējot nepieciešamo informāciju.

Salīdzinot šīs platformas risinājumu ar esošajiem analogiem, ir skaidri novērojamas vairākas priekšrocības:

* luxrent.lv - piedāvā plašu automašīnu piedāvājumu, taču negarantē ērtu modeļu pārskatu un detalizētus aprakstus par modeļiem un to atšķirībām. Mērķis ir radīt platformu, kas nodrošinātu rūpīgi izstrādātu informāciju un skaidrus modeļu aprakstus, lai palīdzētu lietotājiem veikt labi informētu izvēli.
* rentclub.lv - piedāvā mašīnu nomas iespējas, taču mājaslapa ir grūti pielietojama. Mērķis ir radīt platformu, kura nodrošinātu vienkāršu un intuitīvu pieredzi, kas padara automašīnu izvēles procesu daudz saprotamāku.
* rigacars.lv - mājaslapa nepiedāvā pietiekami intuitīvu satura izkārtojumu. Mērķis ir radīt platformu ar skaidru automašīnu katalogu, lai lietotāji varētu viegli veikt salīdzinājumus un atrast vēlamo transportlīdzekli.

Šī auto tirdzniecības platforma ir paredzēta cilvēkiem, kuri vēlas iegādāties automašīnu, taču vēlas to darīt ar pēc iespējas mazākām šķēršļiem darījuma procesā. Vēlos piedāvāt vienkāršu un intuitīvu veidu, kā pārskatīt automobiļu klāstu, konfigurēt nomas datus pēc personīgiem kritērijiem un rezervēt automašīnu izmēģinājuma braucienam.

1. **UZDEVUMA NOSTĀDNE**

Kvalifikācijas darba uzdevums ir izveidot *Automašīnu tirdzniecības platformu* izmēģinājuma braucienu rezervēšanai. Sistēmā nepieciešams realizēt iespēju izvēlēties automašīnu markas un modeļu sasaistot klientu un pārdevēju intereses.

Pakalpojums ir tendēts cilvēkiem ar interesi automobiļa iegādē. Šī platforma ir aktuāla, jo pašreizējais automašīnu pārdošanas platformu piedāvājums tirgū nav pietiekami organizēts un intuitīvs. Lielākā daļa esošo platformu nesniedz lietotajiem pietiekami skaidri prezentētu informāciju par automašīnām. Sniegtās informācijas kvalitāte klientiem rada liekas šaubas par izvēli un produktu atšķirībām. Šīs problēmas sekas izraisa nepieciešamību pavadīt vairāk laika meklējot un pārbaudot dažādās iespējas nenonākot pie galējā pirkuma.

**Sistēmas pamatdarbības ietver:**

* automobiļu modeļu katalogs;
* datuma un laikam rezervācija automašīnas izmēģinājumam;
* administratīva satura rediģēšana;
* administratīva datu pārskatīšana un dzēšana;
* lietotāju autorizācija platformā;
* lietotāja personīgo datu rediģēšana.

Paredzēts, ka sistēmu lietos trīs lietotāju klases: Administrators, Viesis un Lietotājs (skat. 1. att.).

## 

1.att. Lietojumgadījuma diagramma

# 2. PRASĪBU SPECIFIKĀCIJA

# 2.1 Ieejas un izejas informācijas apraksts

***2.1.1. Ieejas informācijas apraksts***

Sistēmā tiks nodrošināta šādas ieejas informācijas apstrāde.

1. Informācija par **lietotājiem** sastāvēs no šādiem datiem.

* Lietotājvārds – burtu teksts ar izmēru līdz 32 rakstzīmēm.
* E-pasts – burtu un ciparu teksts ar izmēru līdz 64 rakstzīmēm.
* Telefona numurs – cipari ar izmēru līdz 20 rakstzīmēm.
* Admin – boolean, izvēle start lietotāju un administratoru

2. Informācija par **automašīnām** sastāvēs no šādiem datiem.

* km/l – float ciparu vērtība kilometriem uz litru.
* Šosejas km/l – float ciparu vērtība kilometriem uz litru.
* Kategorija – burtu teksts ar izmēru līdz 255 rakstzīmēm.
* Pārnesumkārba – burtu teksts ar izmēru līdz 255 rakstzīmēm.
* Degviela – burtu teksts ar izmēru līdz 255 rakstzīmēm.
* Ražotājs – burtu teksts ar izmēru līdz 255 rakstzīmēm.
* Modelis – burtu teksts ar izmēru līdz 255 rakstzīmēm.
* Gads – cipari ar izmēru līdz 4 rakstzīmēm.
* Ražotājs – burtu teksts ar izmēru līdz 255 rakstzīmēm.
* Nomas faktors – float ciparu vērtība nomas koeficentam.
* Cena – float ciparu vērtība nomas cenai.
* Pieejamība – boolean vērtība modeļa pieejamībai sistēmā.

3. Informācija par **pārdošanas vietu** sastāvēs no šādiem datiem.

* Valsts – burtu teksts ar izmēru līdz 255 rakstzīmēm.
* Pilsēta – burtu un ciparu teksts ar izmēru līdz 255 rakstzīmēm.
* Adrese – burtu un ciparu teksts ar izmēru līdz 255 rakstzīmēm.
* Epasts – burtu un ciparu teksts ar izmēru līdz 255 rakstzīmēm.
* Telefona numurs – cipari ar izmēru līdz 20 rakstzīmēm.

4. Informācija par **īri** sastāvēs no šādiem datiem.

* Sākuma datums – datuma un laika vērtība.
* Beigu datums – datuma un laika vērtība.
* Statuss – boolean vērtība, kas norāda vai noma pašlaik ir procesā.
* Kopējā cena – float vērtība nomas cenas izvadei.

***2.1.2. Izejas informācijas apraksts***

1. **LAPAS:** ~~lietotāja konfigurācijas informācija. Platforma izvada konfigurēto automašīnu ar detalizētu informācijas pārskatu.~~

## 2.2 Funkcionālās prasības

1. Jānodrošina lietotāju reģistrācija.
   1. Veikt reģistrācijas datu ievadi un pārbaudīt to formatējumu. Izvadīt kļūdu paziņojumus formāta kļūdu gadījumos.
   2. Attiecīgi informēt par kļūdām gadījumos, kad tiek konstatētas problēmas ar formātu.

1.3. Salīdzināt ievadītos datus ar esošo datu bāzi, lai noskaidrotu, vai lietotājs jau ir reģistrējies.

1.4. Pārbaudīt paroles drošības līmeni. Ja tas neatbilst noteikumiem, sniegt atbilstošu paziņojumu.

1. Jānodrošina lietotāja autorizācija.
   1. Pārbaudīt ievadītos autorizācijas datus un veikt to formatējuma verifikāciju.
   2. Izvadīt atbilstošus paziņojumus par kļūdām vai veiksmīgu autorizāciju.
   3. Nodrošināt drošības pasākumus, piemēram, paroles aizmiršanas gadījumā, lai paroli nomainītu.
   4. Pārbaudīt lietotāja statusu un atļaut piekļuvi tikai autorizētiem lietotājiem.
2. Jānodrošina administrācijas paneļa funkcionalitāte.
   1. Nodrošināt iespēju pievienot jaunas automašīnas vai atjaunināt esošo informāciju.
   2. Piedāvāt iespēju skatīt un rediģēt esošās rezervācijas datus, piemēram, mainīt braucienu datumu vai laiku.
   3. Nodrošināt iespēju rediģēt lietotāju piekļuves tiesības un informāciju. Funkcija administratoriem, lai izveidotu un pārvaldītu citus administratorus sistēmā.
   4. Nodrošināt funkciju automašīnu dzēšanai no sistēmas
   5. Nodrošināt iespēju administratoriem dzēst rezervācijas, kuras varētu būt atceltas vai nevajadzīgas.
3. Jānodrošina lietotāja profila pārskata funkcionalitāte paneļa.
   1. Sniedz iespēju lietotājam rediģēt un atjaunināt personīgo informāciju, piemēram, vārdu, uzvārdu, e-pasta adresi vai citus profilam saistītus datus.
   2. Piedāvā funkciju saglabāt veiktās profila izmaiņas.
4. Jānodrošina lietotāja profila pārskata funkcionalitāte paneļa.
   1. Sniedz iespēju lietotājam rediģēt un atjaunināt personīgo informāciju, piemēram, vārdu, uzvārdu, e-pasta adresi vai citus profilam saistītus datus.
   2. Piedāvā funkciju saglabāt veiktās profila izmaiņas.
5. Jānodrošina produktu pārskata funkcionalitāte lietotājiem.
   1. Nodrošina iespēju lietotājam apskatīt visus pieejamos automobiļus sistēmā.

6.2. Piedāvā informatīvu un viegli pārskatāmu sarakstu ar esošajiem automobiļiem.

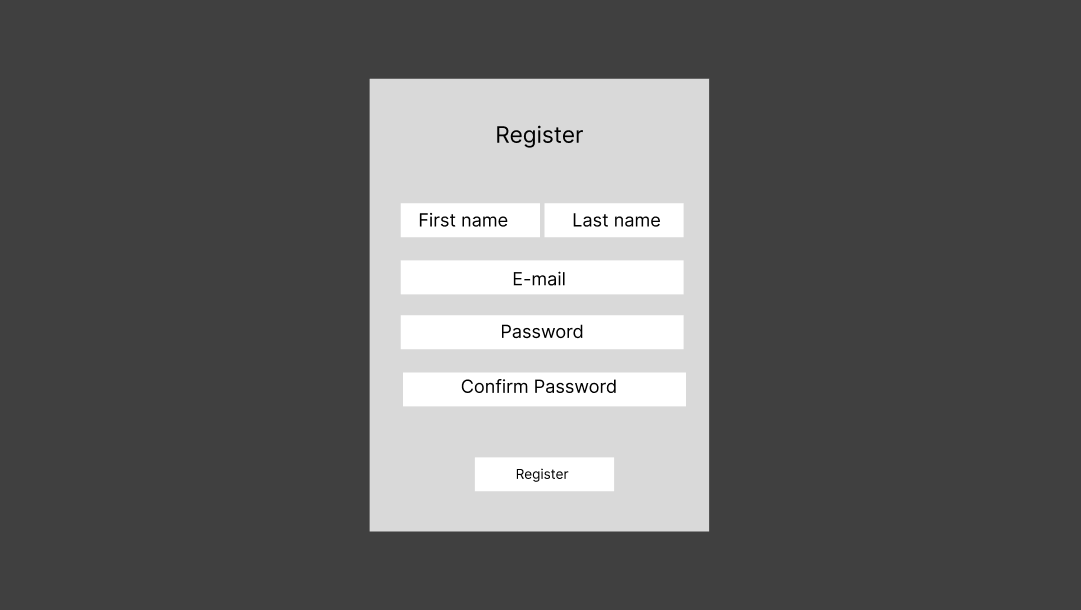
6.3. Sniedz detalizētu informāciju par automobiļiem, ieskaitot modeļus, cenas, pieejamību un citas būtiskas īpašības.

6.4. Ļauj lietotājam filtrēt rezultātus pēc dažādiem kritērijiem, piemēram, modeļa, cenas vai pieejamības.

1. Jānodrošina produktu konfigurācijas funkcionalitāte lietotājiem.
   1. Piedāvā lietotājam iespēju pielāgot un konfigurēt automobiļus saskaņā ar saviem vēlējumiem un prasībām.
   2. Attēlo reāllaika izmaiņas, ļaujot lietotājam vizuāli novērtēt veiktās konfigurācijas rezultātu.
   3. Informē par iespējamiem papildus izdevumiem, lai lietotājs būtu pilnībā informēts par savu izvēli. Tiek izvadīts kopējais izmaksu apkopojums.

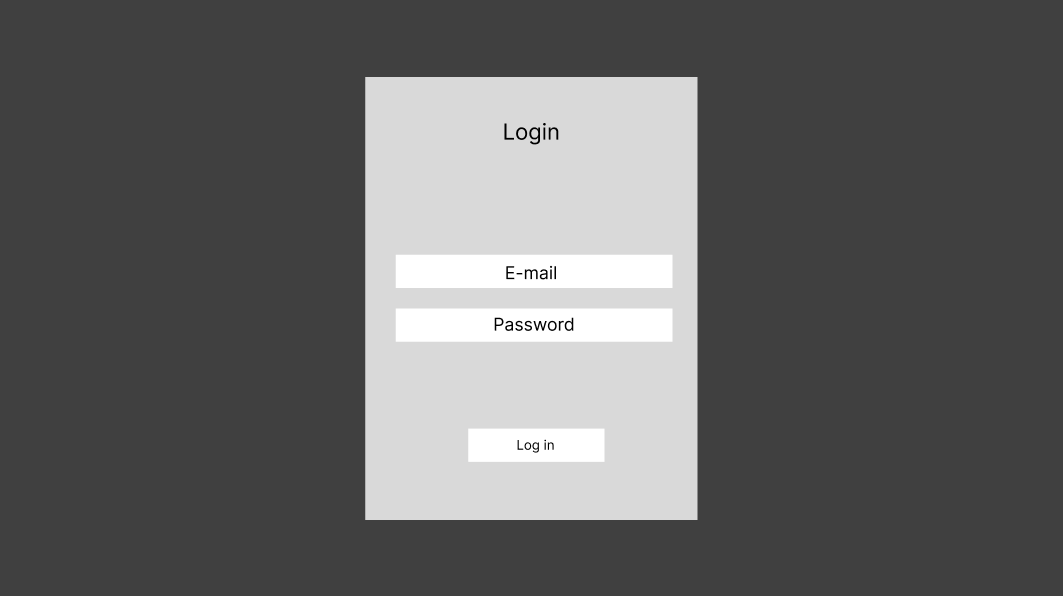
## 2.3 Nefunkcionālās prasības

1. Sistēmas saskarnes valodai ir jābūt angļu valodā.
2. Jānodrošina tīmekļa lietojumprogrammas pielāgošanas ekrāna izmēriem, kas mūsdienās tiek lietoti, lai to varētu izmantot uz dažādiem monitora izmēriem.
3. Tekstam ir jābūt Helvetica fontā.
4. Jānodrošina vienkāršas sistēmas atjaunināšanas un uzturēšanas iespējas, lai ieviestu jaunas funkcijas vai labotu iespējamās kļūdas.
5. Jānodrošina elastīgas un drošas piekļuves kontroles iespējas, lai ierobežotu piekļuvi noteiktai informācijai un funkcionalitātei atbilstoši lietotāja lomai.



2.att. Sistēmas reģistrācijas skice

Šī skice demonstrē lapu ērtam un saprotamam interfeisam reģistrācijas procesa nodrošināšanai. Tā piedāvā ievadīt nepieciešamos datus, lai pieteikties sistēmai jaunajiem lietotājiem.



3.att. Sistēmas pievienošanās skice

Šī skice demonstrē lapu ērtam un saprotamam interfeisam pavienošanās procesa nodrošināšanai. Tā piedāvā ievadīt nepieciešamos datus, lai pieteikties sistēmai esošajiem lietotājiem.

# UZDEVUMU RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS

**Frontend:**

* Next.js 14 ir React framework, kas paredzēts pilnvērtīgu tīmekļa lietojumprogrammu veidošanai. Izmanto React Components, lai izveidotu lietotāja saskarnes, un Next.js funkcijas un optimizācijas. Next.js automātiski konfigurē React nepieciešamos rīkus, piemēram, komplektēšanu un kompilēšanu.
* Tailwind CSS 3.3.6 izceļas ar savu modularitāti, ātrumu un skaidrajām klasēm. Nodrošina iepriekš definētus stila elementus, atvieglojot izstrādi un nodrošinot atsaucīgu dizainu.

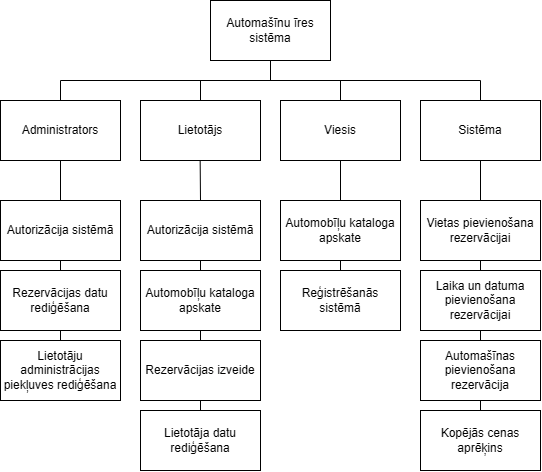
**Backend:**

* Typescript 5.3 piedāvā statisku tipu pārbaudi, kas var palīdzēt novērst kļūdas jau izstrādes laikā. Izmantojot moduļus un importus, veicina kodu atkārtotu pielietošanu un uzturēšanu.
* Prisma 5 ir nākamās paaudzes Node.js un TypeScript ORM. Uzlabo darbu ar datu bāzēm, pateicoties intuitīvajam datu modelim, automātiskajai migrācijai, tipu drošībai un automātiskajai pabeigšanai. Padara datu bāzu darbu vienkāršāku un drošāku, piedāvājot modernus risinājumus un efektīvu izstrādi.
* PostgreSQL 16.1 piedāvā plašu funkciju un paplašinājumu klāstu, kas atbalsta sarežģītus datu pārvaldības uzdevumus. Izmanto uzlabotu SQL valodas implementāciju, kas nodrošina procedurālo valodu, trigeru, pārskatu un citu datu pārvaldības koncepciju atbalstu.

# PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

# 4.1 Sistēmas struktūras modelis

***4.1.1. Sistēmas arhitektūra***

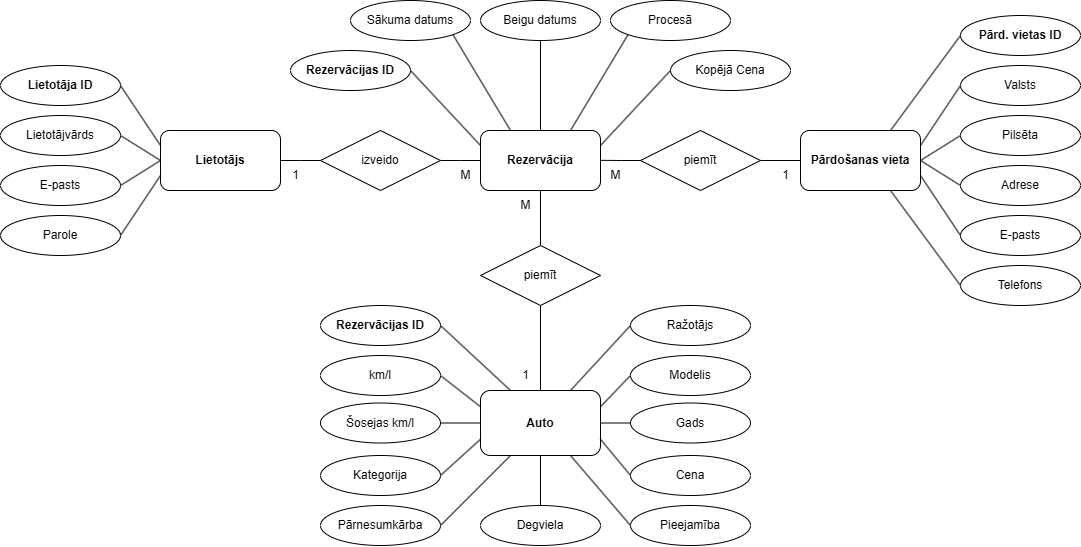


2.att. Sistēmas arhitektūra

* **Administratoru modulis.** Administratoriem sistēmā ir pieejama datu kontrole sistēmā. Personīgā autorizācija, produkta datu rediģēšana, lietotāju piekļuves pārvaldība un rezervāciju datu rediģēšana. Administrators var autorizēties, lai iegūtu piekļuvi administratīvajam panelim, rediģēt informāciju par lietotājiem un pārdošanas vietām, kā arī pārvaldīt esošās rezervācijas.
* **Lietotāju modulis.** Lietotāji ir sistēmas galvenie lietotāji, kuriem ir pieeja automobiļu katalogam, rezervāciju veidošanai un savu lietotāja datu rediģēšanai. Viņi var pārlūkot pieejamo automobiļu katalogu, pielāgot automobiļu nomas datumu un laiku, izveidot rezervācijas un rediģēt savu personīgo informāciju.
* **Viesu modulis.** Viesis ir nepieautorizēts lietotājs, kuram ir ierobežota pieeja sistēmai. Viņam ir iespēja veikt reģistrāciju, pārlūkot ierobežotu automobiļu katalogu, bet viņš nevar veikt rezervācijas vai rediģēt lietotāja datus.
* **Sistēmas modulis.** Sistēma izpilda tehniskās funkcijas, kas saistītas ar datu apstrādi un iekšējo sistēmas darbību. Pēc rezervācijas izveides no lietotāja puses, sistēma pievieno datus par pārdošanas vietas datiem.

***4.2.1. Sistēmas ER modelis***

Sistēmas ER-modelis sastāv no 4 entitijām (skat. 2. att.), kas nodrošina pamat informācijas uzglabāšanu un apstrādi. “Rezervācija” raksturo rezervācijas, kas uzglabā lietotāja izvēlēto laiku un datumu, kopējo cenu, kā arī statusu pabeigšanas gadījumos. “Lietotājs” raksturo lietotājvārdu, e-pastu un paroli, kas reģistrējoties tiek saglabāta sistēmā. “Pārdošanas vieta” raksturo tirdzniecības vietas valsti, pilsētu, adresi un telefona numuru. “Auto” raksturo automobiļa km/l, šosejas km/l, kategoriju, pārnesumkārbu, degvielu, ražotāju, modeli, gadu, cenu un pieejamību sistēmā.



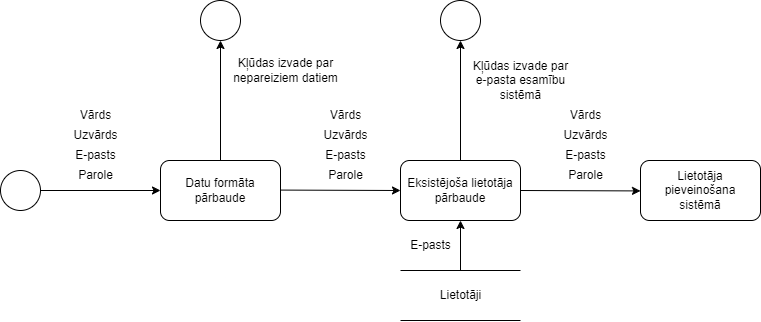
3.att. Sistēmas ER-diagramma

# 4.2 Funkcionālās sistēmas modelis

***4.2.1. Datu plūsmu modelis***

1. Lietotāja izveidošana.

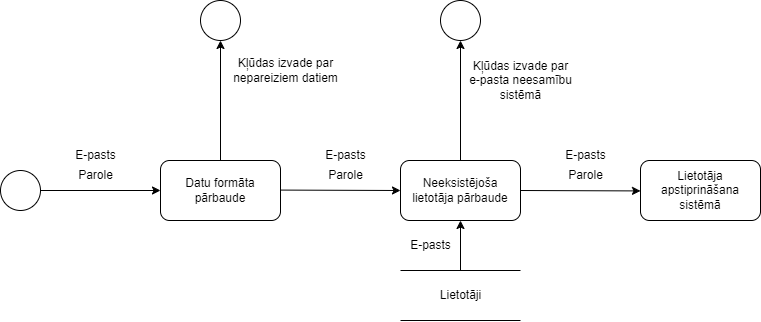
Lietotājiem būs iespēja reģistrēt savu kontu sistēmā. 4.1. attēlā ir redzama lietotāja datu ievade, sekojošā formatējuma un drošības nosacījumu pārbaude. Tai seko pārbaude par konta pārbaude par iepriekšējo eksistenci caur e-pasta reģistrāciju datu bāzē. Procesa beigās lietotāja saturs tiek saglabāts, reģistrējot jauno lietotāju sistēmā.



4.att. Lietotāju izveidošanas datu plūsmas diagramma

1. Lietotāja pievienošanās sistēmā.

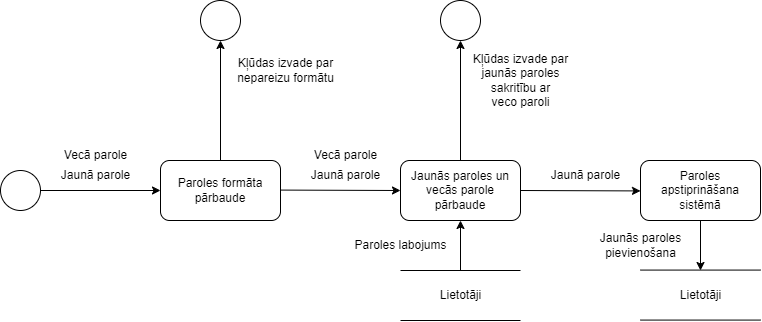
Lietotājiem būs iespēja ar izveidoto kontu pievienoties sistēmā. 4.2. attēlā ir redzama lietotāja datu ievade, seko formatējuma pārbaude. Tai seko pārbaude par konta neeksistēšanas gadījumu caur e-pastu datu bāzē. Procesa beigās lietotāja saturs tiek saglabāts, pievienojot lietotāju sistēmā.



4.1.att. Lietotāju pievienošanas datu plūsmas diagramma

1. Lietotāja paroles atjaunošana.

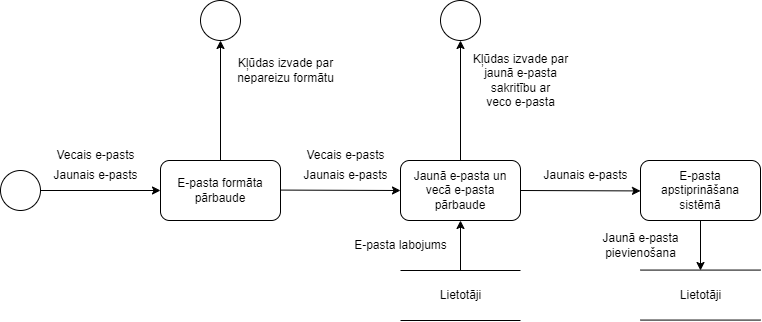
Lietotājiem būs iespēja ar atjaunot savu paroli sistēmā. 4.3. attēlā ir redzama lietotāja paroles ievade, seko formatējuma pārbaude. Tai seko pārbaude par paroles sakritību ar iepriekšējo. Procesa beigās lietotāja parole tiek saglabāta sistēmā.



4.2.att. Lietotāju paroles atjaunošanas datu plūsmas diagramma

1. Lietotāja e-pasta atjaunošana.

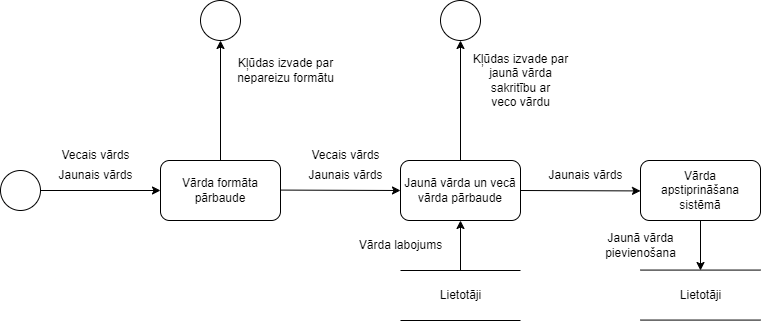
Lietotājiem būs iespēja ar atjaunot savu e-pastu sistēmā. 4.4. attēlā ir redzama lietotāja e-pasta ievade, seko formatējuma pārbaude. Tai seko pārbaude par e-pastu sakritību ar iepriekšējo. Procesa beigās lietotāja e-pasts tiek saglabāts sistēmā.



4.3.att. Lietotāju e-pasta atjaunošanas datu plūsmas diagramma

1. Lietotāja vārda atjaunošana.

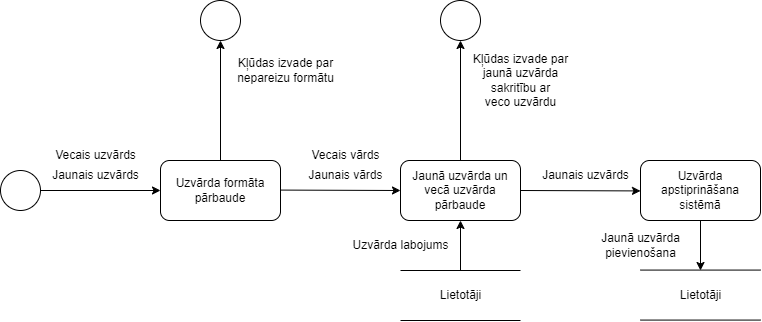
Lietotājiem būs iespēja ar atjaunot savu vārdu sistēmā. 4.5. attēlā ir redzama lietotāja vārda ievade, seko formatējuma pārbaude. Tai seko pārbaude par vārda sakritība ar iepriekšējo. Procesa beigās lietotāja vārds tiek saglabāts sistēmā.



4.4.att. Lietotāju vārda atjaunošanas datu plūsmas diagramma

1. Lietotāja uzvārda atjaunošana.

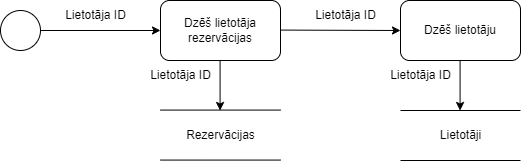
Lietotājiem būs iespēja ar atjaunot savu uzvārdu sistēmā. 4.6. attēlā ir redzama lietotāja uzvārda ievade, seko formatējuma pārbaude. Tai seko pārbaude par uzvārda sakritība ar iepriekšējo. Procesa beigās lietotāja uzvārds tiek saglabāts sistēmā.



4.5.att. Lietotāju uzvārda atjaunošanas datu plūsmas diagramma

1. Lietotāja profila dzēšana, administratīva lietotāja dzēšana.

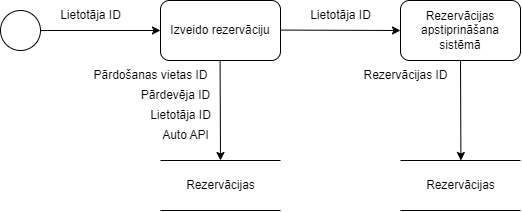
Lietotājiem būs iespēja ar dzēst savu profilu, administratoriem būs iespēja dzēst savu un citu lietotāju profilus sistēmā. 4.7. attēlā ir redzama ar lietotājiem saistīto rezervāciju dzēšana, kurai seko lietotāja datu izdzēšana no sistēmas. Procesa beigās lietotāja tiek izņemts no sistēmas.



4.6.att. Lietotāju profila dzēšanas datu plūsmas diagramma

1. Lietotāja rezervācijas izveidošana.

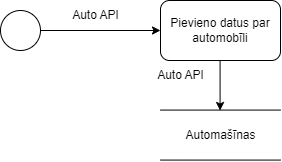
Lietotājiem būs iespēja izvedot rezervācijas sistēmā. 4.8. attēlā ir redzama ar lietotāja rezervācijas izveide un lietotāja, pārdošanas vietas, pārdevēja un automašīnu datu pievienošana. Seko rezervācijas izveide sistēmā ar unikālo identifikatoru. Procesa beigās rezervācija tiek pievienota sistēmai.



4.7.att. Lietotāju rezervācijas izveidošanas datu plūsmas diagramma

1. Administratīvā produktu datus rediģēšana.

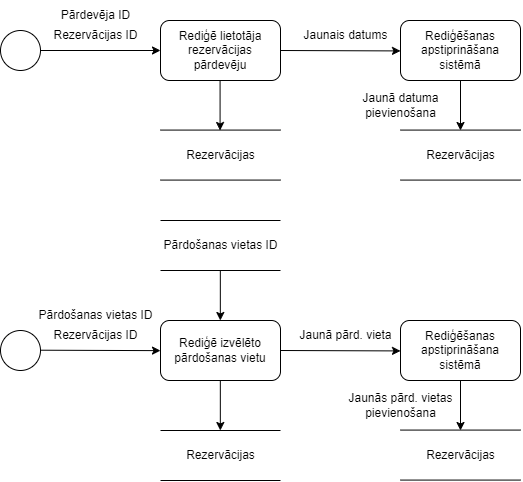
Lietotājiem būs iespēja izvedot rezervācijas sistēmā. 4.9. attēlā ir redzama lietotāja rezervācijas izveide un lietotāja, pārdošanas vietas, pārdevēja un automašīnu datu pievienošana. Seko rezervācijas izveide sistēmā ar unikālo identifikatoru. Procesa beigās rezervācija tiek pievienota sistēmai.



4.8.att. Administratīvās produktu rediģēšanas datu plūsmas diagramma

1. Administratīvā rezervācijas datu rediģēšana.

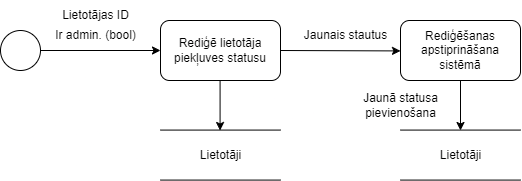
Administratoriem būs iespēja rediģēt rezervācijas sistēmā. 4.10. attēlā ir redzama izvēlētā pārdevēja rediģēšana, kurai seko jaunā datuma izvēle. Pārdošanas vietas rediģēšanas gadījumā tiek izvēlēta jaunā vieta un saglabāta sistēmā. Procesa beigās jaunie rezervācijas dati tiek pievienota sistēmai.



4.9.att. Administratīvās rezervācijas datu rediģēšanas datu plūsmas diagramma

1. Administratīvā lietotāju piekļuves datu rediģēšana.

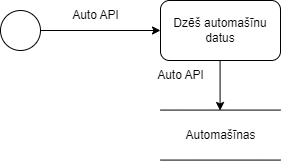
Administratoriem būs iespēja rediģēt lietotāju statusu sistēmā. 4.11. attēlā ir redzama “Ir admin” statusa rediģēšana, kurai seko jaunā statusa pievienošana, kritēriju padarot patiesu. Procesa beigās jaunie piekļuves statusa dati tiek pievienota sistēmai.



4.10.att. Administratīvās lietotāju piekļuves datu rediģēšanas datu plūsmas diagramma

1. Administratīvā produktu dzēšana.

Administratoriem būs iespēja dzēst pievienotos produktus sistēmā. 4.12. attēlā ir redzama ārējā avota datu dzēšana. Procesa beigās tiek dzēsti izvēlētie dati sistēmā.



4.11.att. Administratīvās produktu dzēšanas plūsmas diagramma

# 5. DATU STRUKTŪRAS APRAKSTS

Datu bāze sastāv no 4 tabulām, kas sevī satur informāciju par lietotāju, pārdevēju, rezervācijām un tirdzniecības vietām.

1. Tabula “**user**” glabā datus par lietotājiem un administratoriem.

2. Tabula “**car**” glabā datus par automašīnām.

3. Tabula “**rental**” glabā datus par nomas automobīļu rezervācijām.

4. Tabula “**dealership**” glabā datus par tirdzniecības vietām.

Tabula “user” ir saistīta ar tabulu “rental”

1. tabula

Tabulas “**user**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | id | varchar | 25 | Primārā atslēga, unikāls identifikators, pēc noklusējuma cuid() |
| 2. | name | varchar | 255 | Lietotāja vārds |
| 3. | email | varchar | 255 | Lietotāja e-pasts, unikāls |
| 4. | password | varchar | 255 | Lietotāja parole |
| 5. | emailVerified | timestamp | - | Datums un laiks, kad e-pasts tika verificēts, var būt null |
| 6. | image | varchar | 255 | Lietotāja profila attēls, var būt null |
| 7. | createdAt | timestamp | - | Lietotāja izveides laiks, pēc noklusējuma tagad() |
| 8. | updatedAt | timestamp | - | Datums un laiks, kad lietotājs pēdējo reizi tika atjaunināts, tiek atjaunināts automātiski |
| 9. | admin | boolean | - | Booleans karogs, lai norādītu, vai lietotājs ir administrators, pēc noklusējuma false |

Tabula “car” ir saistīta ar tabulu “rental”

1. tabula

Tabulas “**car**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | id | varchar | 25 | Primārā atslēga, unikāls identifikators, pēc noklusējuma cuid() |
| 2. | kmpl | float | - | Kilometri uz litru (kmpl) automašīnai |
| 3. | highway\_kmpl | float | - | Kilometri uz litru (kmpl) uz šosejas automašīnai |
| 4. | category | varchar | 255 | Automobiļa kategorija |
| 5. | transmission | varchar | 255 | Automobiļa pārnesumu veids |
| 6. | fuel\_type | varchar | 255 | Automobiļa degvielas veids |
| 7. | manufacturer | varchar | 255 | Automobiļa ražotājs |
| 8. | model | varchar | 255 | Automobiļa modelis |
| 9. | year | int | - | Automobiļa ražošanas gads |
| 10. | slug | varchar | 255 | URL draudzīgs identifikators automobiļiem |
| 11. | rental\_factor | float | - | Nomāšanas koeficients cenas aprēķināšanai |
| 12. | rental\_price | float | - | Pamata nomas cena automašīnai |
| 13. | available | boolean | - | Automobiļa pieejamības statuss, pēc noklusējuma true |
| 14. | createdAt | timestamp | - | Automobiļa izveides laiks, pēc noklusējuma tagad() |
| 15. | updatedAt | timestamp | - | Datums un laiks, kad automašīna pēdējo reizi tika atjaunināta, tiek atjaunināts automātiski |

Tabula “dealership” ir saistīta ar tabulu “rental”.

1. tabula

Tabulas “**dealership**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | id | varchar | 25 | Primārā atslēga, unikāls identifikators, pēc noklusējuma cuid() |
| 2. | country | varchar | 255 | Valsts, kurā atrodas dīleris |
| 3. | city | varchar | 255 | Pilsēta, kurā atrodas dīleris |
| 4. | address | varchar | 255 | Dīlera adrese |
| 5. | email | varchar | 255 | Dīlera e-pasts |
| 6. | phone | varchar | 20 | Dīlera tālruņa numurs |

Tabula “rental” ir saistīta ar tabulām “user”, “car”, “dealership”.

1. tabula

Tabulas “**rental**” struktūra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1. | id | varchar | 25 | Primārā atslēga, unikāls identifikators, pēc noklusējuma cuid() |
| 2. | userId | float | 25 | Ārējā atslēga, norāda uz User(id) |
| 3. | carId | float | 25 | Ārējā atslēga, norāda uz Car(id) |
| 4. | dealershipId | varchar | - | Ārējā atslēga, norāda uz Dealership(id) |
| 5. | startDate | varchar | - | Nomāšanas sākuma datums |
| 6. | endDate | varchar | - | Nomāšanas beigu datums |
| 7. | ongoing | boolean | - | Nomāšanas statuss, vai tas notiek |
| 8. | totalPrice | float | - | Kopējā nomas cena |
| 9. | createdAt | timestamp | - | Nomāšanas izveides laiks, pēc noklusējuma tagad() |
| 10. | updatedAt | timestamp | - | Datums un laiks, kad nomāšana pēdējo reizi tika atjaunināta, tiek atjaunināts automātiski |

# 5.1 Tabulu relāciju shēma

# 

5.att. Tabulu relāciju shēma

**SECINĀJUMI**

1. **Projekta būtība:**

Projekta galvenā būtība ir izveidot automašīnu tirdzniecības platformu izmēģinājuma braucienu rezervēšanai, piedāvājot lietotājiem iespēju viegli un ērti izvēlēties automašīnas, konfigurēt tās atbilstoši savām vēlmēm un rezervēt izmēģinājuma braucienus.

1. **Sistēmas priekšrocības:**

Projekta mērķis ir piedāvāt uzlabotu pieredzi salīdzinājumā ar esošajām automobiļu tirdzniecības platformām. Jaunā platforma nodrošinās detalizētu informāciju par automobiļiem, iespēju konfigurēt automobiļus un vienkāršotu izmēģinājuma braucienu rezervēšanu.

1. **Pamatfunkcionalitāte:**

Sistēma ietver trīs galvenās lietotāju klases: Administratori, lietotāji un viesi. Katrai klasei ir savas specifiskas funkcijas, sākot no automobiļu datu pārvaldības līdz lietotāju autorizācijai un rezervācijai.

1. **Tehnoloģiju pielietojums:**

Projekta tehnoloģiskā bāze: Next.js un React front-end izstrādei, Typescript statiskai tipa pārbaudei, Tailwind CSS dizainam. Datu bāzei un datu pārvaldībai izmanto Prisma un PostgreSQL.

1. **Funkcionālās un nefunkcionālās prasības:**

Projekta projektā ir skaidri definētas funkcionālās prasības, piemēram, lietotāju reģistrācija, automašīnas konfigurācija, administrācijas paneļa funkcionalitāte utt. Nefunkcionālās prasības ietver sistēmas mērogojamību, drošību un labu lietotāja pieredzi.

1. **Sistēmas arhitektūra un modelēšana:**

Projekta arhitektūra ietver administratora, lietotāja un viesa moduļus, katram no tiem ir savi atbildības un piekļuves līmeņi. ER modelis tiek izmantots, lai skaidri definētu sistēmas datu struktūru un savstarpējās attiecības starp vienībām.

**INFORMĀCIJAS AVOTI**

1. ER diagramma - <https://www.lucidchart.com/pages/er-diagrams>
2. UML diagramma - <https://www.lucidchart.com/pages/uml-use-case-diagram>
3. Datu plūsmas diagramma - <https://www.lucidchart.com/pages/data-flow-diagram>
4. Next.js dokumentācija - <https://nextjs.org/docs>
5. NextAuth.js dokumentācija - <https://next-auth.js.org/getting-started/introduction>
6. Typescript dokumentācija - <https://www.typescriptlang.org/>
7. React-toastify dokumentācija - <https://fkhadra.github.io/react-toastify/introduction>
8. Lucide React dokumentācija - <https://lucide.dev/guide/packages/lucide-react>
9. React DayPicker dokumentācija - <https://react-day-picker.js.org/>
10. Radix dokumentācija - <https://www.radix-ui.com/themes/docs/overview/getting-started>
11. Material UI dokumentācija - https://mui.com/material-ui/getting-started/
12. Tailwind CSS dokumentācija - <https://tailwindcss.com/>
13. Prisma dokumentācija - <https://www.prisma.io/>
14. PostgreSQL dokumentācija - <https://www.postgresql.org/>