Liepājas Valsts tehnikums

Tīmekļu vietne “Dzīvnieku māja”

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Profesionālā kvalifikācija Programmēšanas tehniķis

Grupas nosaukums 4PT

Projekta izstrādātājs Daniils Ternoviks

/vārds, uzvārds, paraksts/

Eksāmena datums 202\_\_. gada\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Liepāja 202\_\_

**Satura rādītājs**

[Ievads 3](#_30j0zll)

[**1. Uzdevuma formulējums 4**](#_3znysh7)

[2. Programmatūras prasību specifikācija 5](#_2et92p0)

[2.1. Produkta perspektīva 5](#_tyjcwt)

[2.2. Sistēmas funkcionālās prasības 5](#_3dy6vkm)

[2.3. Sistēmas nefunkcionālās prasības 10](#_1t3h5sf)

[2.4. Gala lietotāja raksturiezīmes 11](#_1pxezwc)

[3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums 12](#_4d34og8)

[3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts 12](#_2s8eyo1)

[3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts 12](#_17dp8vu)

[4. Sistēmas modelēšana un projektēšana 13](#_3rdcrjn)

[4.1. Sistēmas struktūras modelis 13](#_26in1rg)

[4.2. Klašu diagramma / ER diagramma (Obligāti, ja sistēmā ir klases un/vai tabulas) 13](#_lnxbz9)

[4.3. Funkcionālais un dinamiskais sistēmas modelis 13](#_35nkun2)

[4.4. Aktivitāšu diagramma (Activity) 13](#_1ksv4uv)

[4.5. Lietojumgadījumu diagramma (Use Case) 13](#_44sinio)

[4.6. Sistēmas moduļu apraksts un algoritmu shēmas 13](#_2jxsxqh)

[5. Lietotāju ceļvedis 14](#_z337ya)

[6. Testēšanas dokumentācija 15](#_3j2qqm3)

[6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums 15](#_1y810tw)

[6.2. Testpiemēru kopa 15](#_4i7ojhp)

[6.3. Testēšanas žurnāls 15](#_2xcytpi)

[7. Individuālais ieguldījums 16](#_1ci93xb)

[8. Secinājumi 17](#_3whwml4)

[9. Lietoto terminu un saīsinājumu skaidrojumi 18](#_2bn6wsx)

[10. Literatūras un informācijas avotu saraksts 19](#_qsh70q)

[PIELIKUMS 20](#_3as4poj)

# 

# Ievads

Šīs projekts, “Dzīvnieku māja”, ir mans mēģinājums padarīt dzīvnieka iegūšanu par digitālo risinājumu, ar mērķi veicināt dzīvnieka pieejamību, kā arī veidot lielu dzīvnieku pārdošanas platformu, “Dzīvnieku māja” ir liela un ērta platforma, kurā lietotāji var piedāvāt savus mājdzīvniekus, kā arī atrast sev vajadzīgo un interesantu sev dzīvnieku un iegādāties viņu.

Šajā dokumentā iekļauta informācija par:

* Programmatūras prasību specifikāciju, kas nosaka funkcionālas prasības.
* Izstrādes līdzekļu, rīku aprakstu un izvēles pamatojumu, kās skaidro tehniskos lēmumus un risinājumu izvēles loģiku.
* Sistēmas modelēšanu un projektēšanu, sniedzot pārskatu par dzīvnieku pārdošanas platformu.
* Lietotāju ceļvedi, kas palīdzēs izprast, kā lietotāji izmantos un mijiedarbosies ar platformu.
* Testēšanas dokumentāciju, kas nodrošina platformas uzticamību un drošību.
* Secinājumus, kas atspoguļo darba rezultātus un projektu kopuma.

Projekta pamatuzdevumi ietver dzīvnieku reģistrēšanas un pārvaldības procesu automatizāciju, nodrošinot lietotājiem ērtu rīku klāstu dzīvnieku augšupielādei, dzīvnieku aprakstiem, kā arī dzīvnieka cenu noteikšanai. Platformā tiek uzsvērta drošība un datu precizitāte, lai lietotāji ir pārliecināti par katru no dzīvniekiem un to vērtību.

# 1. Uzdevuma formulējums

Projekta "Dzīvnieku māja" galvenais mērķis ir izstrādāt un integrēt digitālu risinājumu, kas pilnveido un apvieno visas būtiskās funkcionalitātes, saistītas ar dzīvnieku pirkšanu, pārdošanu un apkopi. Šis projekts izceļas ar viegli saprotamu un efektīvu funkcionalitāti, nodrošinot lietotājiem ērtu izmantošanas pieredzi un daudzefektīvus rīkus.

Lietotājiem ir iespēja augšupielādēt informāciju par savu dzīvnieku uz mūsu platformas tabulas, sniedzot visu nepieciešamo informāciju. Katram lietotājam ir personīgais profils, kurā apkopota informācija par viņu, un piedāvājums apskatīt savus pirkumus vai sludinājumus.

Platforma piedāvā lietotājiem plašu dzīvnieku izvēli, izmantojot detalizētu tabulu ar iespēju izvēlēties dzīvniekus, piemēram, kaķus, suņus vai citus. Katram dzīvniekam ir iespēja pievienot attēlus, un platforma nodrošina drošu pirkšanas procesu.

Visas šīs funkcijas tiek izstrādātas tā, lai gan pieredzējušie lietotāji, gan jaunpienācēji varētu viegli izmantot platformu, izmantojot daudzefektīvus rīkus pirkumiem vai pārdošanai.

# 2. Programmatūras prasību specifikācija

Šajā nodaļā jūs redzēsiet informāciju par produkta perspektīvu, sistēmas funkcionālām prasībām, kā arī nefunkcionālām prasībām.

## 2.1. Produkta perspektīva

"Dzīvnieku māja" - tas ir unikāls tiešsaistes dzīvnieku audzētava, kas sniedz izcilas iespējas tiem, kuri meklē ne tikai ideālu mājdzīvnieku, bet arī dziļu mijiedarbību ar dzīvnieku pasauli. Mūsu projekts balstās uz rūpīgu partneru atlasi un vēlmi piedāvāt mūsu lietotājiem neaizmirstamu pieredzi, izvēloties un rūpējoties par saviem nākamajiem draugiem.

## 2.2. Sistēmas funkcionālās prasības

**P.1. Reģistrēšana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju lietotājam piereģistrēties “Dzīvnieku māja” mājaslapā

Ievaddati:

| Nosaukums | Obligats | Piezimes |
| --- | --- | --- |
| username | jā | jabūt unikālam |
| password | jā | nav |

Apstrāde:

1) Mājaslapa pārbauda vai dati ir ievadīti korekti

2) Mājaslapa pārbauda vai ievadītais vārds nav aizņemts

Izvaddti:

Lietotājs tiek pārvietots uz Autorizēšanas logu.

**P.2. Autorizēšana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju lietotājam autorizēties “Dzīvnieku māja” mājaslapā

Ievaddati:

| Nosaukums | Obligats | Piezimes |
| --- | --- | --- |
| username | jā | nav |
| password | jā | nav |

Apstrāde:

1) Mājaslapa pārbauda vai dati ir ievadīti korekti

2) Mājaslapa pārbauda vai ievadītais vārds eksistē

3) Mājaslapa pārbauda vai parole ir ievadīta korekti un sakrīt ar lietotājvārdu reģistrēto

Izvaddati:

Lietotājs tiek pārvietots uz sākuma lapu.

**P.3. Dzīvnieku pievienošana “Dzīvnieku māja” mājaslapas tabulā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram pievienot mājaslapas tabulā jauno dzīvnieku

Ievaddati:

| Nosaukums | Obligats | Piezimes |
| --- | --- | --- |
| Name | jā | nav |
| Type | jā | nav |
| Date of birth | jā | Ierakstīt ar cipariem vai izvelēties izmantojot date picker |
| Gender | jā | Izvelēties ar dropdown |

Apstrāde:

1) Mājaslapa pārbauda vai dati ir ievadīti korekti

2) Mājaslapa pārbauda dzīvnieka ID kuru vajadzēs rediģēt

Izvaddati:

Tiek pievienots jauns dzīvnieks.

**P.4. Dzīvnieku dzešana no “Dzīvnieku māja” mājaslapas tabulā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram dzēst mājaslapas tabulā esošo dzīvnieku

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Animals”

3) Lietotāja statuss ir Administrators vai Moderators

4) Tabulas laukā nospiestā poga “Delete”

Apstrāde:

1) Mājaslapa pārbauda dzīvnieka ID kuru vajadzēs dzēst

Izvaddati:

Tiek dzēsts esošais dzīvnieks.

**P.5. Lietotāja datu izlabošana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram izlabot nepieciešamos datus lietotājam

Ievaddati:

| Nosaukums | Obligats | Piezimes |
| --- | --- | --- |
| Name | jā | nav |
| Type | jā | nav |
| Date of birth | jā | Ierakstīt ar cipariem vai izvelēties izmantojot date picker |
| Gender | jā | Izvelēties ar dropdown |

Apstrāde:

1) Mājaslapa pārbauda vai dati ir ievadīti korekti

2) Mājaslapa pārbauda lietotāja ID kurām būs jāmaina datus

Izvaddati:

Lietotājam tiek izmainīti dati.

**P.6. Lietotāja statusa rediģēšana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram izmainīt tabulā lietotāja statusu

Ievaddati:

1) Atvērta mājaslapa

2) Lietotāja statuss ir Administrators vai Moderators

3) Tabulā “Users” nospiestā poga “Edit”

4) Ievadīti nepieciešamie dati

5) Nospiesta poga “Save”

Apstrāde:

1) Mājaslapa pārbauda vai jauns lietotāja statuss nesakrīt ar esošo

2) Mājaslapa pārbauda lietotāja ID kurām būs jāmaina statusu

Izvaddati:

Tiek izmainīts lietotāja statuss.

**P.7. Lietotāja izveidošana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram izveidot tabulā jauno lietotāju

Ievaddati:

1) Atvērta mājaslapa

2) Lietotāja statuss ir Administrators vai Moderators

3) Tabulā “Users” nospiestā poga “Create”

4) Ievadīti nepieciešamie dati

5) Nospiesta poga “Save”

Apstrāde:

1) Mājaslapa pārbauda vai lietotājvārds nesakrīt ar kādu no esošiem

2) Mājaslapa pārbauda vai dati tiek ierakstīti korekti

Izvaddati:

Tiek pievienots jauns lietotājs.

**P.8. Pieprasījuma nosūtīšana par dzīvnieka ņemšanu “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju visiem lietotājiem aizsūtīt pieprasījumu uz dzīvnieku paņemšanu

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Animals”

3) Nospiesta poga “Сreate a query”

4) Ievadīti nepieciešamie dati

5) Nospiesta poga “Send”

Apstrāde:

1) Mājaslapa pārbauda vai dati tiek ierakstīti korekti

2) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

3) Mājaslapa pārbauda dzīvnieka ID , ko cilvēks grib paņemt

Izvaddati:

Pieprasījums tiek nosūtīts.

**P.9. Iemaksu plānā izveidošana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju visiem lietotājiem izvelējoties dzīvnieku izveidot iemaksu plānu priekš ikmēneša maksājuma

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Animals”

3) Nospiesta poga “Сreate a query”  
 4) Nospiesta izvēlne “Installment payment”

Apstrāde:

1) Mājaslapa pārbauda vai dati tiek ierakstīti korekti

2) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

3) Mājaslapa pārbauda dzīvnieka ID , ko cilvēks grib paņemt  
 4) Mājaslapa pārbauda vai ir nospiesta izvēlne “Installment payment”  
 5) Mājaslapa parād summu kāds ir iemaksu plāns uz izvelēto dzīvnieku

Izvaddati:

Iemaksas plāns tiek paradīts.

**P.10. Pieprasījums uz dzīvnieka dresētāja “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju visiem lietotājiem izvēlējoties dzīvnieku izvēlēties dzīvnieka dresētāju uz norādītu stundu skaitu

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Animals”

3) Nospiesta poga “Сreate a query”

4) Ievadīti nepieciešamie dati  
 5) Nospiesta izvēlne “Animal trainer”

6) Izvelēts konkrētais cilvēks

7) Izvelēts stundu skaits

Apstrāde:

1) Mājaslapa pārbauda vai dati tiek ierakstīti korekti

2) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

3) Mājaslapa pārbauda dzīvnieka ID , ko cilvēks grib paņemt

4) Mājaslapa pārbauda vai ir nospiesta izvēlne “Animal trainer”

5) Mājaslapa pārbauda vai ir izvēlēts konkrēts cilvēks

6) Mājaslapa pārbauda vai ir izvēlēts stundu skaits

7) Mājaslapa pārbauda vai izvelēts dzīvnieks ir apmācāms

8) Jā dzīvnieks nav apmācāms izvada paziņojumu

Izvaddati:

Dresētājs ir izvelēts.

**P.11. Rādīt dzīvnieku tabulu “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju visiem lietotājiem redzēt tabulu ar visiem atļautiem dzīvniekiem.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta lapa “Animals”

Apstrāde:

2) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

Izvaddati:

Tabula tiek parādīta.

**P.12. Dzīvnieku filtrēšanas atlase tabulā “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju visiem lietotājiem izvēleties filtru tabulā ar dzivniekiem.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta lapa “Animals”

3) Nospiests trīsstūris blakām kadām no dzīvnieku parametram.

Apstrāde:

2) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

3) Mājaslapa pārbauda kāds parametrs ir izvelēts

Izvaddati:

Tabula tiek filtrēta.

**P.12. Lietotāja personīgais profils “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam ieiet savā personīgajā profilā un apskatīties informāciju par saviem pirkumiem un pāsutījumu vēsturi.

Ievaddati:

1) Atvērta mājaslapa

2) Nospiesta poga “My profile”

Apstrāde:

2) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

3) Mājaslapa pārbauda visas darbības veiktas no šī profila

Izvaddati:

Lietotāja profils tiek parādīts.

**P.13. Iespēja izsekot piegādes statusu lietotāja personīgaja profilā**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam ieiet savā personīgajā profilā un izsekot piegādes statusu savam pirkumam.

Ievaddati:

1) Atvērta mājaslapa

2) Nospiesta poga “My profile”

3) Nospiesta poga “Check delivery statuss”

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

2) Mājaslapa pārbauda vai lietotājam ir pirkumi un pasutījumi

Izvaddati:

Lietotāja piegādes statusa informācija tiek izvadīta.

**P.14. Iespēja nosūtiet e-pasta vēstuli vietnes administrācijai**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam aizsutīt e-pasta vēstuli administrācijai.

Ievaddati:

1) Atvērta mājaslapa

2) Tiek aiziets uz mājaslapas beigām

3) Ievadīta visa nepieciešama informācija formā

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

2) Mājaslapa pārbauda vai lietotājs ierakstījis visu informaciju.

3) Mājaslapa pārbauda vai tiek nospiesta poga “Send”

Izvaddati:

Vēstule tiek nosūtīta.

**P.15. Iespēja pievienot izvēlēto dzīvnieku favorītiem**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam pievienot izvelēto dzīvnieku favorītos lai pēc tām vieglāk ir viņus atrast.

Ievaddati:

1) Atvērta mājaslapa

2) Atverta lapa “Animals”

3) Nospiesta poga “Like” pie izvelēta dzīvnieka.

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

2) Mājaslapa pārbauda izvelēta dzīvnieka ID.

Izvaddati:

Dzīvnieks tiek pievienots favorītos.

**P.16. Iespēja apskatīties dzīvniekus pievienotus pie favorītiem**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam apskatīties dzīvniekus, kas bija iepriekš pievienoti pie favorītiem.

Ievaddati:

1) Atvērta mājaslapa

2) Nospiesta poga “Favorite”

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

2) Mājaslapa pārbauda vai lietotājam ir favorītie dzīvnieki

Izvaddati:

Tiek paradīti favorītie dzīvnieki.

**P.17. Iespēja apskatīties jaunumus un atlaides “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam apskatīties jaunumus un atlaides.

Ievaddati:

1) Atvērta mājaslapa

2) Nospiesta poga “News”

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

Izvaddati:

Tiek paradīta lapa ar jaunumiem un atlaidem.

**P.18. Iespēja tūlkot “Dzīvnieku māja” mājaslapu**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam pārtulkot mājaslapu.

Ievaddati:

1) Atvērta mājaslapa

2) Nospiesta poga “EN”, “LV” vai “RU”

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

2) Mājaslapa pārbauda vai nav dubulti izvēlēta viena valoda

Izvaddati:

Tiek pārtulkota lapa.

**P.19. Iespēja mainīt fona krāsu “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam mainīt fona krāsu mājaslapā.

Ievaddati:

1) Atvērta mājaslapa

2) Izvēlēta vajādzīga krāsa.

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

2) Mājaslapa pārbauda vai nav dubulti izvēlēta viena krāsa

Izvaddati:

Fons tiek mainīts.

**P.20. Iespēja atrāst mūs piedāvātos socialajos tīklos**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam apskatīties mūs arī sociālajos tīklos.

Ievaddati:

1) Atvērta mājaslapa

2) Aiziets uz mājaslapas apakšnodaļu

3) Nospiesta poga ar vajadzīgo sociālo tīklu

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

Izvaddati:

Lietotājs tiek aizsūtīts uz izvēlēto sociālo tīklu.

## 2.3. Sistēmas nefunkcionālās prasības

1) Programmatūras izstrādes procesa dokumentācijai ir jābūt noformētai un izstrādātai atbilstoši Latvijas Valsts standartam LVS 68:1996 par programmatūras prasību specificēšanu.

2) Lietotāju un programmatūras saskarnei ir jābūt Latvijas Republikas valsts valodā.

3) Mājaslapas dominējošas krāsām ir jābūt sekojošiem: Zils, Balts, Pelēks.

4) Mājaslapas saskarnei ir jābūt ērtai un ergonomiskai (tādai, kas minimizē IS lietotāja slodzi, piemēram, viegli uztveramai).

# 3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

Izstrādes līdzekļi un rīki, kas tika izmantoti šim projektam, ir rūpīgi atlasīti, lai nodrošinātu efektīvu un integrētu digitālo risinājumu.

1. **Backend izstrāde (Java):**

* Java tika izvēlēta kā backend valoda, jo tā piedāvā plašas iespējas un ir pazīstama ar savu uzticamību un drošību.
* Java nodrošina lielisku veiktspēju un skalējamību, kas ir būtiski digitālajiem risinājumiem ar pieaugošu apmeklētību.

1. **Frontend izstrāde (React un TSX):**

* React un TypeScript (TSX) tika izvēlēti frontend izstrādei, jo tie nodrošina modulāru un viegli uztveramu kodu.
* React piedāvā efektīvas komponentes un atjaunošanu tikai nepieciešamajām daļām, kas uzlabo lietotāja pieredzi un veiktspēju.

1. **Datu bāze (PostgreSQL un pgAdmin4):**

* PostgreSQL tika izvēlēts kā datu bāzes pārvaldības sistēma, jo tas ir atvērtā pirmkoda un piedāvā plašas iespējas attiecībā uz datu modeļiem un skaļojamību.
* pgAdmin4 ir izmantojams kā vizuāls rīks datu bāzes pārvaldībai, nodrošinot ērtu un pārredzamu interfeisu.

1. **API saskarne (Postman):**

* Postman tiek izmantots kā rīks, lai veiktu un pārbaudītu HTTP pieprasījumus uz serveri.
* Tas piedāvā intuitīvu lietošanas pieredzi un palīdz uzlabot un pārbaudīt backend API funkcionalitāti.

**Izvēles pamatojums:**

* Java un React ir populāras un plaši izmantotas tehnoloģijas ar aktīvu attīstības kopienas atbalstu.
* PostgreSQL piedāvā attiecības datu bāzes priekšrocības, kā arī atbilst digitālo risinājumu augstajiem prasību standartiem.
* Postman sniedz ērtu veidu, kā pārbaudīt un optimizēt servera API izsaukumus, nodrošinot to efektivitāti un drošību.

Šie izvēlētie līdzekļi un rīki kopā veido integrētu un efektīvu risinājumu, kas nodrošina lietotājiem ērtu un drošu pieredzi, vienlaikus nodrošinot uzticamu sistēmas darbību un datu integritāti.

## 3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts

**Backend izstrāde (Java):**

Java ir vispāratpazīta un augsti izvērtēta programmatūras valoda, kas tika izvēlēta kā backend valoda šim projektam. Tā piedāvā stabilu un drošu vidi, un tā ir ļoti piemērota lielāku projektu izstrādei. Ar bagātīgu standartbibliotēku un atbalstu plašai kopienai, Java nodrošina iespēju veiksmīgi veidot kompleksus backend risinājumus. Šī valoda ir ievērojama arī ar to, ka tā ļauj izveidot drošus un efektīvus serverus, kas ir būtiski digitālās platformas darbībai.

**Frontend izstrāde (React un TSX):**

React, kombinēts ar TypeScript (TSX), nodrošina modernu un modulāru pieeju lietotāja saskarnei. React piedāvā komponentu struktūru, kas uzlabo kodu uztveramību un atjauno tikai nepieciešamās daļas, palielinot lietotāja pieredzi. TypeScript sniedz papildu drošību, lai novērstu dažādas kļūdas jau izstrādes posmā. Šis izvēlētais kombinācijas risinājums ir īpaši piemērots projektam, kurā lietotāja saskarne ir būtiska un nepieciešama augsta veiktspēja.

**Datu bāze (PostgreSQL un pgAdmin4):**

PostgreSQL tiek izmantots kā datu bāzes pārvaldības sistēma, jo tā piedāvā attiecību datu bāzes priekšrocības un elastību datu modeļu definēšanā. Ar aktīvu attīstību un plašu kopienas atbalstu PostgreSQL nodrošina uzticamu datu uzglabāšanas sistēmu. pgAdmin4 ir vizuāls rīks, kas atvieglo datu bāzes pārvaldību, piedāvājot pārredzamu saskarni un uzlabojot administratīvās funkcijas.

**API saskarne (Postman):**

Postman ir plaši izmantots rīks, ko izmanto izstrādātāji, lai testētu un pārbaudītu API saziņu. Tas nodrošina iespēju izsaukt un pārbaudīt dažādus servera API punktus, sniedzot iespēju ērti novērtēt to darbību un efektivitāti. Šis rīks palīdz nodrošināt API drošību un pareizu funkcionēšanu.

Kopumā izvēlētie risinājuma līdzekļi ir uzmanīgi atlasīti, ņemot vērā to, kā katrs no tiem nodrošina drošību, veiktspēju un ērtību izstrādes procesā, veidojot integrētu un efektīvu digitālo platformu "Dzīvnieku māja".

## 3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts

**Alternatīvās risinājuma līdzekļi un valodas:**

**Backend izstrāde:**

Node.js un Express:

* Node.js nodrošina ātru un efektīvu servera izstrādi, balstoties uz JavaScript, un tas ir īpaši piemērots lietotāja draudzīgu un mazāk sarežģītu backend sistēmu izveidei.
* Express ir minimāls un elastīgs servera rīks, kas papildina Node.js, piedāvājot ātru izstrādi un modulāru kodu.

**Frontend izstrāde:**

Vue.js un JavaScript:

* Vue.js ir moderna un viegli uztverama JavaScript bibliotēka, kas nodrošina lietotājiem elastīgu un viegli pielāgojamu saskarni.
* Tā kā Vue.js ir viegli integrējams un saprotams, tas var būt laba alternatīva, īpaši, ja nepieciešama ātra saskarnes izstrāde.

**Datu bāze:**

MongoDB un Robo 3T:

* MongoDB piedāvā dokumentu orientētu datu bāzi, kas ir īpaši piemērota datu struktūrām, kurām bieži mainās forma.
* Robo 3T ir vizuāls rīks MongoDB pārvaldībai, piedāvājot intuitīvu saskarni un iespēju viegli rediģēt un pārvaldīt dokumentus.

**API saskarne:**

Insomnia:

* Insomnia ir alternatīvs rīks API pārbaudei un testēšanai, sniedzot intuitīvu saskarni un efektīvu veidu, kā veikt un pārbaudīt HTTP pieprasījumus.

**Iespējamais (alternatīvais) pamatojums:**

* Alternatīvie risinājumi ir izvēlēti, ņemot vērā vieglumu un ātrumu izstrādes procesā, īpaši, ja nepieciešama ātra prototipēšana vai mazākas sistēmas izstrāde.
* Vue.js un JavaScript var būt piemērota alternatīva, ja nepieciešama viegli pielāgojama un saprotama lietotāja saskarne, kas var būt īpaši noderīga nelielām projektu sākumversijām.
* MongoDB un Robo 3T nodrošina alternatīvu pieeju datu bāzes izvēlei, īpaši, ja nepieciešams elastīgs datu modelis.
* Insomnia ir viegli izmantojams rīks API saskarņu pārbaudei un testēšanai, sniedzot vienkāršu un efektīvu veidu, kā nodrošināt API funkcionalitāti.

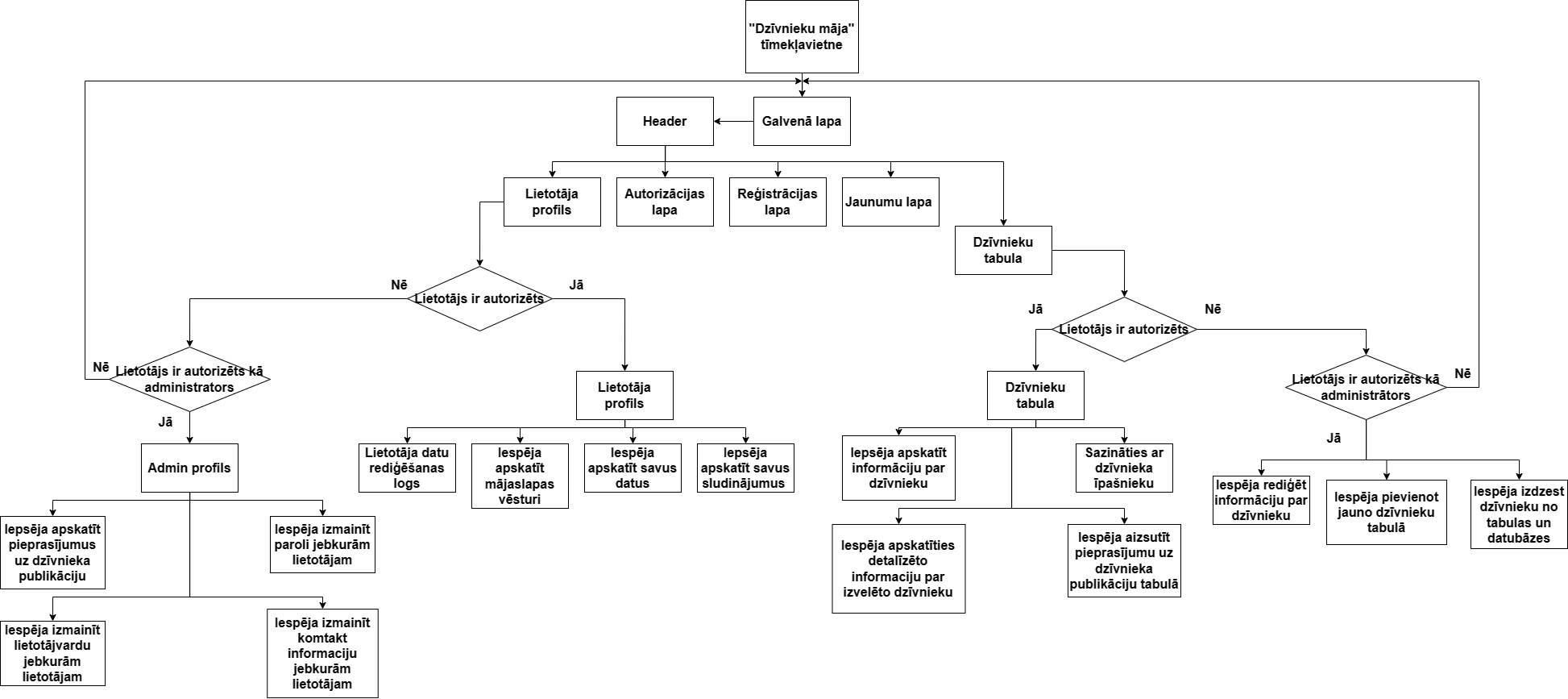
# 4. Sistēmas modelēšana un projektēšana

Šajā nodaļā tiks apskatīta mūsu vietnes sistēmas modelēšana un projektēšana, izmantojot dažādas diagrammas, lai labāk izprastu tās darbību un funkcionalitāti. Kādas diagrammas tiek izmantotas:

1. Sistēmas struktūras modelis.
2. ER diagramma.
3. Funkcionālais un dinamiskais sistēmas modelis.
4. Aktivitāšu diagramma (Activity).
5. Lietojumgadījumu diagramma (Use Case).
6. Sistēmas moduļu apraksts un algoritmu shēmas.

## 4.1. Sistēmas struktūras modelis

Šajā sadaļā tiks izstrādāts un aprakstīts sistēmas struktūras modelis, kas ietver visus galvenos komponentus un to attiecības sistēmā.

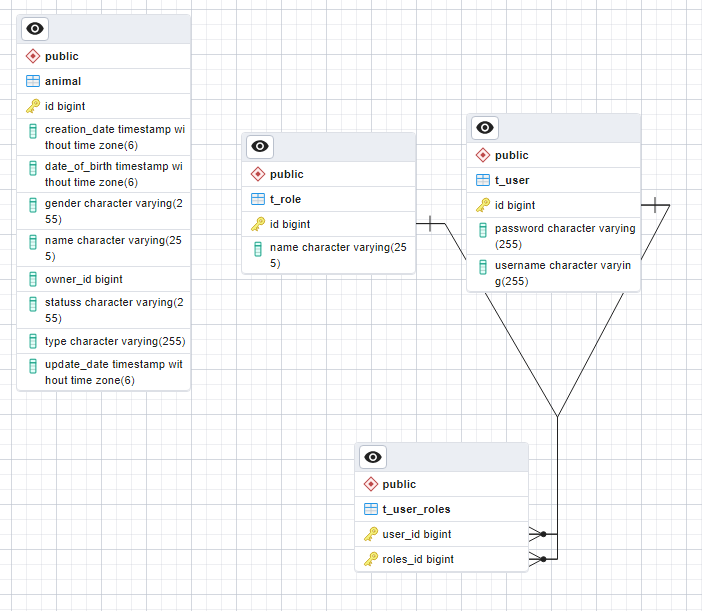


## 1.attēls. Sistēmas struktūras modelis

## 

## 4.2. ER diagramma

Šajā sadaļā tiks izstrādāta entīšu-sakaru (ER) diagramma, kas nodrošina informāciju par datu struktūru sistēmā un attiecībām starp datu elementiem.



**2.attēls. ER diagramma**

## 4.3. Funkcionālais un dinamiskais sistēmas modelis

## Šeit tiks apskatīts gan sistēmas funkciju apraksts, gan arī tās dinamiskais uzvedības modelis, iekļaujot procesu izpildes secību un mijiedarbību starp komponentiem.

## 4.3.1. Datu plusmu diagramma dzīvniekam.

## 

## 3.attēls. Datu plusmu diagramma dzīvniekam.

## 4.3.2. Datu plusmu diagramma lietotājiem.

## 4.attēls. Datu plusmu diagramma lietotājiem.

## 4.3.3. Datu plusmu diagramma administratoram.

## 

## 5.attēls. Datu plusmu diagramma administratoram.

## 4.4. Aktivitāšu diagramma (Activity)

## Šajā sadaļā tiks izveidota aktivitāšu diagramma, kas vizualizēs sistēmas darbību un procesus, norādot to secību un izpildes loģiku.

## 4.4.1. Aktivitāšu diagramma - 1.daļa

## 

## 6.attēls. Aktivitāšu diagramma - 1.daļa

## 4.4.2. Aktivitāšu diagramma - 2.daļa

## 

**7.attēls. Aktivitāšu diagramma - 2.daļā**

## 

## 

## 4.5. Lietojumgadījumu diagramma (Use Case)

## Šajā sadaļā tiks izstrādāta lietojumgadījumu diagramma, kas identificēs un aprakstīs sistēmas funkcijas un to mijiedarbību ar lietotājiem.

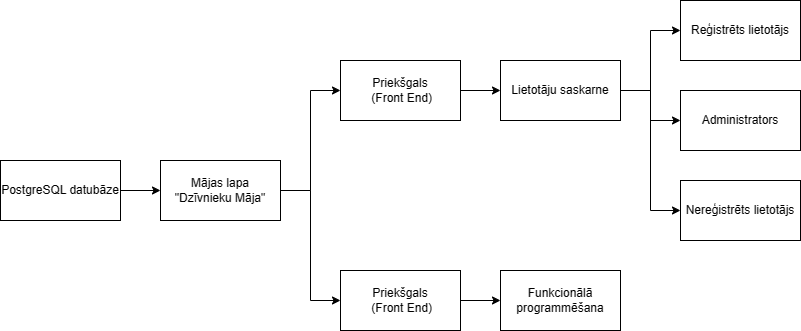
## 

## 8.attels. Lietojumgadījumu diagramma (Use Case)

## 4.6. Sistēmas moduļu apraksts un algoritmu shēmas

Mājas lapai “Dzīvnieku māja” sastāv no priekšpuses (Front End) un no aizmugures (Back End). (Skat. 9.attēlu - **Sistēmas moduļu diagramma**)

Priekšgals ir tīmekļa izstrādes daļa, kas ir atbildīga par lietotāja interfeisa izveidi, kas mijiedarbojas ar lietotāju, izmantojot parlūkprogrammu. Tā “Dzīvnieku māja” projektā tiek izmantotas tādas programmēšanas valodas kā CSS un JavaScript, lai izveidotu mājas lapa, vizuālo dizainu un funkcionalitāti.

Aizmugure ir daļa no tīmekļa izstrādes, kas ir atbildīga par datu apstrādi un glabāšanu, kā arī mijiedarbību ar tīmekļa lietojumprogrammas kliena pusi, izmantojot API. Priekš sava projekta es izmantoju sekojošas programmēšanas valodas, kā Java, PostgreSQL, lai izveidot majas lapas aizmuguri.

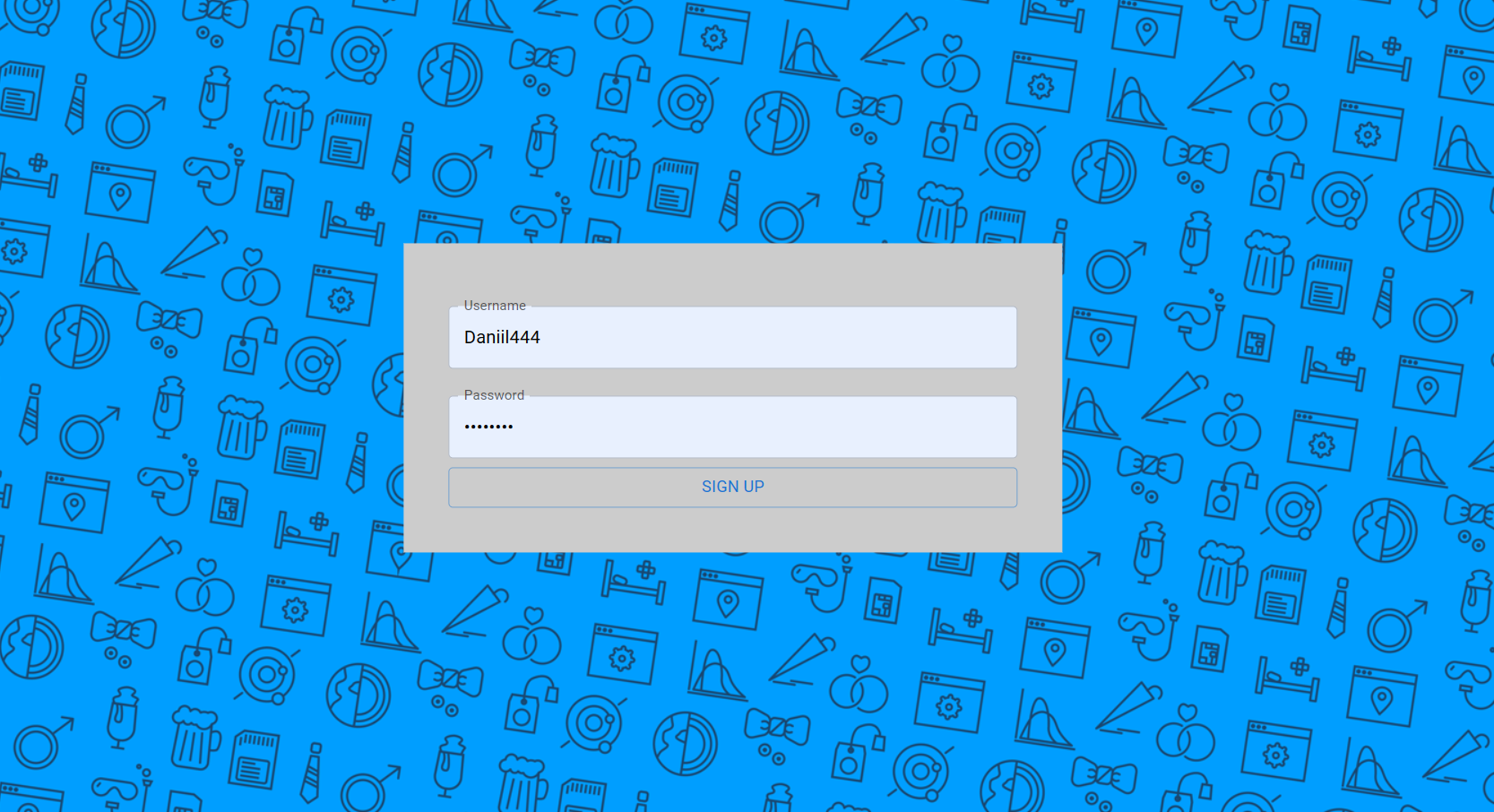
**9.attēls. Sistēmas moduļu diagramma**

# 5. Lietotāju ceļvedis

## Šaja nodaļā tiks paradīts un pateikts solis pa solim, kā lietotājs var tīkt kadā majas lapas sadaļā un izmantot tas sadaļas funkcijas.

## **5.1. Majas lapas reģistrācijas sadaļas izmantošana**

Atverot lapu, lietotājs redzēs reģistrācijas logu, kurā viņš var reģistrēties, izmantojot savu lietotājvārdu un paroli, bet, ja lietotājam vietnē jau ir konts, viņš var noklikšķināt uz pogas "Already have an account?", kas novirzīs viņu uz autorizācijas sadaļu. (Skat. 10.attēlu)

**10.attēls. Reģistrācijas sadaļas izmantošana**

## 5.2. Majas lapas autorizācijas sadaļas izmantošana

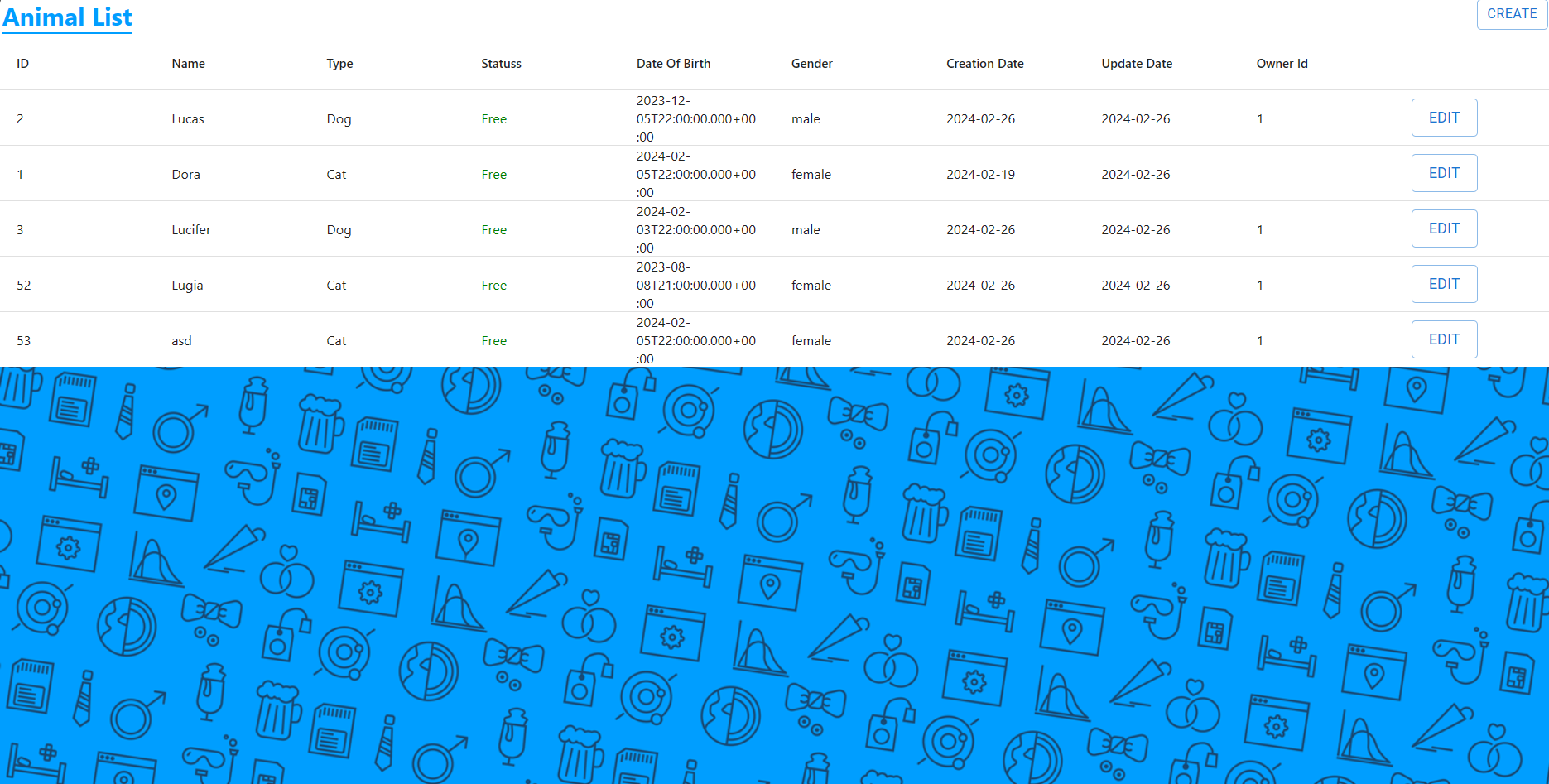
Izmantojot informāciju no iepriekšējās reģistrācijas sadaļas, mēs varam saprast, ka varam nokļūt autorizācijas sadaļā, izmantojot pogu "Already have an account?", šajā sadaļā lietotājam tiek dota iespēja autorizēt savu kontu, izmantojot reģistrācijas sadaļā ievadītos datus, piemēram, lietotājvārdu un paroli, un, ja dati atbilst iepriekš datubāzē ievadītajiem datiem, lietotājs tiek novirzīts uz sadaļu ar dzīvnieku tabulu, bet jā lietotājs ievadījis nepareizos datus, viņam tiks parādits paziņojums par to kā dati nav ievadīti pareizi, un lietotājam bus iespēja velreiz ievadīt datus.(Skat. 11.attēlu)

# 

**11.attēls. Autorizācijas sadaļas izmantošana**

## **5.**3**. M**ajas lapasdzīvnieku tābulas izmantošana un apskatīšana

Kā minēts iepriekšējā sadaļā, mēs varam nokļūt dzīvnieku tabulas sadaļā, ja veiksmīgi piesakāmies vietnē, pēc tam tiks novirzīts uz mājaslapas galveno lapu , kur ir navigācijas pogas , lai tīktu līdz dzīvnieku tābulas jauzspied navigācijas pogu “Table with animals” , tad lietotajs tiks novirzīts uz tabulu , kurā lietotājs var apskatīt visus datubāzē esošos dzīvniekus.

**12.attēls. Dzīvnieku tabulas apskatīšana**

# 

# 5.4. Dzīvnieku pievienošana

# Atrodoties sadaļā ar dzīvnieku tabulu, lietotājs var nospiest pogu "Create" (Skat. 13.attēlu), un tād viņam būs pieejama dzīvnieka pievienošanas forma, lai pievienotu dzīvnieku, ir jāaizpilda visi dati, piemēram, vārds, dzīvnieka veids (suns, kaķis utt.), dzīvnieka dzimšanas datums un dzimums, un pēc tam nospiest pogu "Save"(Skat. 14.attēlu).

# 

# 13.attēls. Dzīvnieku pievienošana

# 

## 

**14.attēls. Dzīvnieku pievienošanas forma**

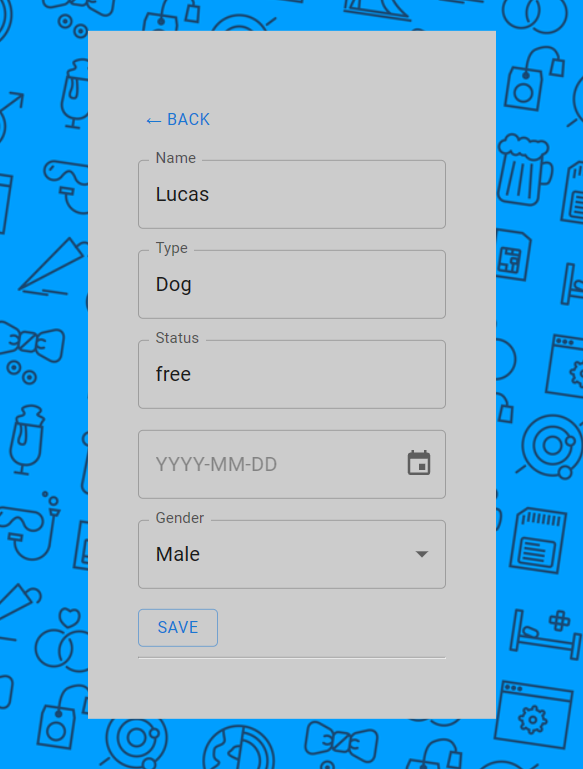
## 5.5. Esoša dzīvnieka datu rediģēšana

Atrodoties sadaļā ar dzīvnieku tabulu, lietotājs var noklikšķināt uz pogas "Edit" rindā ar dzīvnieku, kura datus viņš vēlas mainīt (Skat. 15.attēlu) , tad atvērsies dzīvnieka datu rediģēšanas forma, lai mainītu dzīvnieka datus, ir jāaizpilda visi lauki, piemēram, vārds, dzīvnieka veids, dzimšanas datums un dzimums, tad noklikšķiniet uz pogas "Save" (Skat. 16.attēlu).

# 

# 

**15.attēls. Dzīvnieka datu rediģēšana**

**16.attēls. Dzīvnieka rediģēšanas forma**

# 6. Testēšanas dokumentācija

## 6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums

## 6.2. Testpiemēru kopa

## 6.3. Testēšanas žurnāls

# 7. Individuālais ieguldījums

# 8. Secinājumi

# 9. Lietoto terminu un saīsinājumu skaidrojumi

# 10. Literatūras un informācijas avotu saraksts

# PIELIKUMS