**Profesionālās izglītības kompetences centrs**

**“RĪGAS VALSTS TEHNIKUMS”**

DATORIKAS NODAĻA

Izglītības programma: Programmēšana

**Moduļa “Datu bāzu programmēšana”  
Patversmes sistēma**

2022./2023.m.g.

**SATURS**

[IEVADS 3](#_Toc130499982)

[1. UZDEVUMA NOSTĀDNE 4](#_Toc130499983)

[1.1. Sistēmas lietotāji un to mijiedarbība ar sistēmu 4](#_Toc130499984)

[1.2. Uzdevuma risināšanas līdzekļu izvēles pamatojums 5](#_Toc130499985)

[2. DATU BĀZES PROJEKTĒŠANA 6](#_Toc130499986)

[2.1. Konceptuālais datu modelis 6](#_Toc130499987)

[2.2. ER datu modelis 7](#_Toc130499988)

[3. DATU BĀZES IZSTRĀDE 8](#_Toc130499989)

[3.1. Datu bāzes fiziskā struktūra 8](#_Toc130499990)

[Pielikumi 10](#_Toc130499991)

[1. pielikums. ER modelis 10](#_Toc130499992)

[2. pielikums. Tabulas shēma 11](#_Toc130499993)

# **IEVADS**

Lai uzlabotu lietotāju(patversmes apmeklētāju) pieredzi ir nepieciešamība izveidot patversmes uzskaites datu sistēmu, kas būtu speciāli pielāgota patversmes vajadzībām un piedāvātu vajadzīgās funkcijas, piemēram, iespēju reģistrēties, pieteikt vizīti un aplūkot, vai dzīvnieki atrodas patversmē reālajā laikā.

Pirms sākt izstrādāt patversmes uzskaites datu sistēmu, ir svarīgi veikt tirgus izpēti un saprast, kādi ir līdzīgi produkti, kādi ir to trūkumi un priekšrocības. Šāda informācija palīdzēs izstrādāt labāku un konkurētspējīgāku produktu. Ir apkopoti vairāki līdzīgi produkti, kas piedāvā patversmes datu sistēmas: https://ulubele.org, https://patversme.lv, http://www.dzd.lv. Lai gan katrs no tiem ir nedaudz atšķirīgs, tie visi piedāvā līdzīgas funkcijas un mērķus. Šiem visiem trīs produktiem pietrūkst dažas svarīgas lietas – lietotājs spēs reģistrēties, lietotājam būs iespēja pieteikt vizīti, kā arī aplūkot vai dzīvnieki atrodas patversmē reālajā laikā. Šī sistēma būs izveidota speciāli patversmes vajadzībām, tāpēc tā būs ērtāka un vieglāk lietojama nekā līdzīgi produkti, kas var nebūt pielāgoti šādai uzdevumam.

Patversmes uzskaites datu sistēma ir nepieciešama, jo tā būtu speciāli pielāgota patversmes vajadzībām un piedāvās vajadzīgās funkcijas. Šāda sistēma būs viegli lietojama un ērtāka nekā līdzīgi produkti, kas nav pielāgoti šādam uzdevumam. Tādējādi, izstrādājot šādu sistēmu, varētu radīt labāku un konkurētspējīgāku produktu.

# **1. UZDEVUMA NOSTĀDNE**

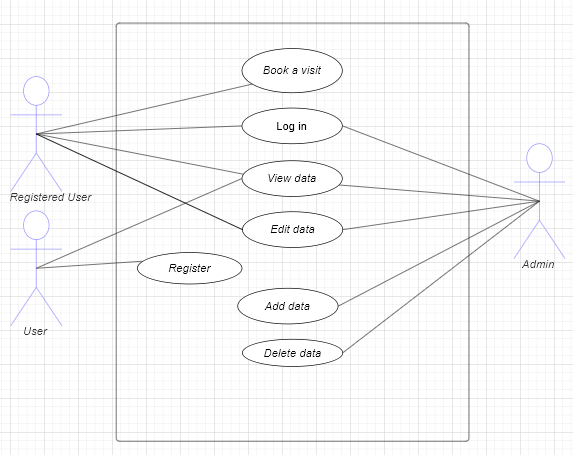
Noslēguma darba uzdevums ir izveidot un automatizēt patversmes uzskaites datu sistēmu. Uz doto brīdi izvērtējot mājaslapas ir atrasti trūkumi, kā arī lietas, ko varētu automatizēt. Nav iespējams veikt reģistrāciju un ienākt mājaslapā, nav iespējams pieteikt vizīti konkrētam dzīvniekam, nav iespējams uzzināt vai dzīvnieks atrodas patversmē.

Ar šī projekta palīdzību potenciālie apmeklētāji vai jaunie saimnieki varēs vieglāk pieteikt vizītes un apkopot informāciju par patversmē pieejamajiem dzīvniekiem un aplūkot informāciju par tiem.

Patversmes sistēmai ir jāizpilda sekojošas funkcionalitātes:

* Lietotāja reģistrēšana;
* Lietotāja autorizēšanās;
* Lietotāja datu labošana;
* Lietotāja profila apskate;
* Dzīvnieka datu pievienošana;
* Dzīvnieka datu rediģēšana;
* Dzīvnieka datu dzēšana;
* Lietotāja vizītes pieteikšana pie dzīvnieka.

## **1.1. Sistēmas lietotāji un to mijiedarbība ar sistēmu**

 Lietošanas gadījumu diagrammā 1.1.1.att. ir attēlotas trīs lietotāju grupas - reģistrēts lietotājs, nereģistrēts lietotājs un administrators. Nereģistrēts lietotājs var tikai apskatīt datus un ir iespēja reģistrēties, lai piekļūtu pārējām darbībām. Reģistrēts lietotājs var ienākt mājaslapā, apskatīt datus un rediģēt datus par sevi, kā arī pieteikt vizīti. Administratoram ir visvairāk opcijas – viņš spēj ienākt, apskatīt datus, pievienot datus, rediģēt datus un dzēst datus.

1.1.1.att. Lietošanas gadījuma diagramma

## **1.2. Uzdevuma risināšanas līdzekļu izvēles pamatojums**

Sistēma ir paredzēta pārlūkprogrammai, tādēļ to realizācijai tiek izmantoti jaunākie programmatūras risinājumi.

Datubāzes izstrādei tiks izmantots phpMyAdmin, jo ir paredzēts MySQL datubāžu pārvaldīšanai izmantojot tīmekļa saskarni. phpMyAdmin ļauj izmantojot pārlūkprogrammu pārvaldīt MySQL serveri, izpildot SQL komandas un apskatīt datubāzu un tabulu saturu.

Datubāzei tiks izmantos MySQL, versija - 10.4.27-MariaDB. Tika izmantota MySQL datubāze, jo atbalsta phpMyAdmin.

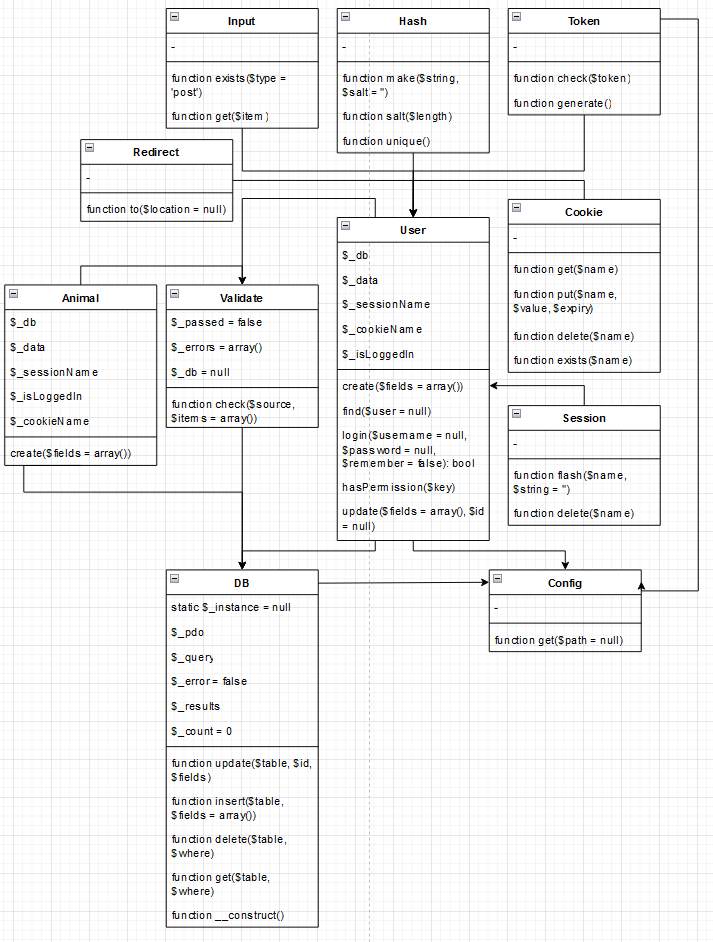
Serverim tiks izmantots XAMPP Control Panel v3.3.0, lai varētu palaist Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.0.25 Database client version: libmysql - mysqlnd 8.0.25 PHP version: 8.0.25, jo to atbalsta phpMyAdmin.

Sistēmas izstrādei tika izmantots Visual Studio Code Version: 1.76.2 (user setup), Date: 2023-03-14T17:55:54.936Z, Electron: 19.1.11, Chromium: 102.0.5005.196, Node.js: 16.14.2, V8: 10.2.154.26-electron.0, OS: Windows\_NT x64 10.0.19044. Izmantoju Visual Studio Code, jo šī man ir labi zināma vide, kurā es spēju produktīvi darboties.

Tīmekļa lapas izveidei tiks izmantots PHP, JavaScript, HTML5 un CSS. Tīmekļa lapas back-end izveidei tiks izmantots PHP, lai sazinātos ar datubāzi arī tiks izmantots PHP, jo tas dod elastību, lai izveidotu savienojumu un strādātu ar dažādām datu bāzēm. Lai sazinātos ar front-end daļu (HTML5 un CSS) tiks izmantots JavaScript un PHP.

# **2. DATU BĀZES PROJEKTĒŠANA**

## **2.1. Konceptuālais datu modelis**

 Lai atspoguļotu datu struktūru, dati tika sadalīti loģiskos elementos (entītijās). Izmatojot UML klašu diagrammu 2.1.1.att. tika definēti galvenākie atribūti katrai entītijai, kā arī dažas metodes, kuras nākotnē tika izmantotas.

2.1.1.att. Lietošanas gadījuma diagramma

## **2.2. ER datu modelis**

Datu bāzes projektēšanā datu kopu un saišu starp tām attēlošanai tika lietota relāciju saišu diagramma, kas sastāv no divu veidu objektiem – entītijām (loģiskais objekts) un relācijām (attiecības). ER modelis (sk.1.pielikumā) sastāv no 4 entītijām, kas atspoguļo datu aprīti sistēmā.

* **”Groups”** – apraksta lietotāja lomu (lietotājs vai administrators). Atribūtu kopums sevī ietver lomas nosaukumu un pieejas numuru lomas nosaukumam.
* **”Users”** – apraksta lietotāju. Atribūtu kopums sevī ietver lietotājvārdu, paroli, saltu, vārdu, uzvārdu, epastu, telefona numuru un grupu pie kuras pieder lietotājs.
* **”Animals”** – apraksta dzīvnieku. Atribūtu kopums sevī ietver dzīvnieka vārdu, vecumu, bildi, vai ir čipots, ja ir čipots, tad čipa numuru.
* **”User\_session”** – apraksta lietotāja sesiju, ja ir aķesēts “remember me”. Atribūtu kopums sevī ietver hash.

Datu bāzes relācijas uzrāda kā savstarpēji ir savienotas divas vai vairāk entītijas.

* Starp Groups un Users attiecība ir viens pret daudziem, jo vienai grupai var būt vairāki lietotāji, bet viens lietotājs var piederēt tikai pie vienas grupas.
* Starp Users un User\_session attiecība ir viens pret vienu, jo vienam lietotājam var piederēti tikai viena sesija.
* Starp Users un Animals attiecība ir daudzi pret daudziem, jo vairāki lietotāji var pieteikt vizīti pie vairāki dzīvniekiem un vairāki dzīvnieki var saņemt vizīti no vairākiem lietotajiem.

# **3. DATU BĀZES IZSTRĀDE**

## **3.1. Datu bāzes fiziskā struktūra**

Datu bāzes projektēšanas rezultātā tika veidotas vairākas tabulas un starp tām tika definēta tabulu relācija, jeb saistība, kas norāda, ka tabulas saistītajos laukos ir vienādi dati. Saites var būt trejādas: viens-pret-vienu, viens pret daudziem un daudzi pret daudziem. Saiti daudzi pret daudziem datu bāzes sistēmā nevar tik tieši realizēt. Lai realizētu šo saiti, tā tiek sadalīta divās viens pret daudziem saitēs. Sadalīšanas procesā radās vēl viena tabula, kas nodrošina saistību starp divām jaunajām saitēm, kā rezultātā visas tabulas ir saistītas savā starpā ar viens pret daudziem. Tabulas shēma ir dota 2. pielikumā

Datubāze sastāv no 5 tabulām, kuras satur informāciju par lietotājiem, dzīvniekiem, apmeklējumiem, lietotāju sesijām un lietotāju grupām.

Tabula **“users”** satur reģistrētu lietotāju datus. Telefona numurā nav iekļauts valsts kods, jo sistēma nav paraudzēta starptautiskai lietošanai.

3.1.1.tabula

**Tabulas “users” struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr** | **Lauka nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1 | id | Auto inc | 11 | Reģistrēta lietotāja kārtas numurs. |
| 2 | username | Varchar | 20 | Reģistrēta lietotajā lietotājvārds |
| 3 | password | Varchar | 64 | Reģistrēta lietotajā šifrēta parole |
| 4 | salt | Varchar | 64 | Nejauši skaitļi, ko pievieno parolei, lai šifrētu to. |
| 5 | name | Varchar | 50 | Reģistrēta lietotajā vārds |
| 6 | surname | Varchar | 50 | Reģistrēta lietotajā uzvārds |
| 7 | email | Varchar | 50 | Reģistrēta lietotajā e-pasts |
| 8 | phone | Varchar | 20 | Reģistrēta lietotajā telefona numurs |
| 9 | group | Int | 11 | Reģistrēta lietotajā grupa (loma) |

Tabula **“users\_session”** satur lietotāja hash, ja ir aķeksēts “acerēties mani”. Dati glabājās septiņas dienas, un šajā tabulā tiek salīdzināts session hash (session cookie) ar lietotāja datiem.

3.1.2.tabula

**Tabulas “users\_session” struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr** | **Lauka nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1 | id | Auto inc | 11 | Reģistrēta lietotāja sesijas numurs pēc kārtas |
| 2 | user\_id | int | 11 | Reģistrēta lietotāja id |
| 3 | hash | varchar | 64 | Reģistrēta lietotāja hash |

Tabula **“groups”** satur lomas nosaukumu, tās pieejas numuru.

3.1.3.tabula

**Tabulas “groups” struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr** | **Lauka nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1 | id | Auto inc | 11 | Grupas numurs pēc kārtas |
| 2 | name | varchar | 20 | Satur lomas nosaukumu (admin, standard) |
| 3 | premissions | text | - | Lomas pieejas numurs (standard user: 0; admin: 1) |

Tabula **“animals”** satur datus par dzīvnieka vārdu, vecumu, tipu, bildi, vai ir čipots, ja ir čipots, tad čipa numuru.

3.1.4.tabula

**Tabulas “animals” struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr** | **Lauka nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1 | id | Auto inc | 11 | Dzīvnieka numurs pēc kārtas |
| 2 | name | varchar | 50 | Dzīvnieka vārds |
| 3 | age | Int | 11 | Dzīvnieka vecums |
| 4 | type | Enum | ('kaķis', 'suns', 'cits') | Dzīvnieka tips |
| 5 | picture | Blob | - | Dzīvnieka bilde |
| 6 | chip | Tinyint | 1 | Vai dzīvniekam ir čips (1- jā, 2-nē) |
| 7 | chipNumber | Int | 11 | Ja ir čips, tad čipa numurs |

Tabula **“visit”** satur lietotāja un dzīvnieka identifikācijas numuru, kā arī datumu, pēc kura varēs uzzināt kurā dienā konkrētais cilvēks dosies pie konkrētā dzīvnieka.

3.1.5.tabula

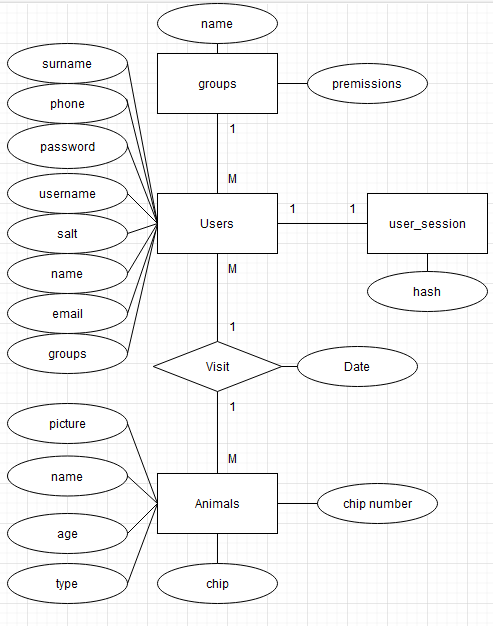
**Tabulas “visit” struktūra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr** | **Lauka nosaukums** | **Tips** | **Izmērs** | **Apraksts** |
| 1 | id | Auto inc | 11 | Dzīvnieka numurs pēc kārtas |
| 2 | Date | date | - | Dzīvnieka vārds |
| 3 | User\_id | Int | 11 | Dzīvnieka vecums |
| 4 | Animals\_id | Int | 11 | Dzīvnieka tips |

# **Pielikumi**

## 1. pielikums.

**ER modelis**



## 2. pielikums.

**Tabulas shēma**