



LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

LIETOTŅU MONITORINGA RISINĀJUMS
KVALIFIKĀCIJAS DARBS

Autors: **Dāvis Mednis**

Studenta apliecības Nr. DM14018

Darba vadītāja: Dr. sc. comp. Vineta Arnicāne

RĪGA 2016

ANOTĀCIJA

Lietotņu monitoringa risinājums ir tiešsaistes rīks, kas atvieglo dažādu programmatūras produktu izstrādi un uzturēšanu. Tā galvenā funkcija ir saņemt un apstrādāt programmatūras sūtītos kļūdas paziņojumus un auditācijas pierakstus. Lietotņu monitoringa risinājums apstrādā un apkopo saņemtos datus vienuviet, kā arī izsūta brīdinājuma e-pastus avārijas gadījumā. Potenciālie risinājuma lietotāji apkopotajiem datiem var piekļūt izmantojot tīmekļa lietotni.

Projekts īstenots, izmantojot modernas izstrādes tehnoloģijas bāzētas NodeJS vidē, kā datu uzturēšanas slāni lietojot PostgreSQL datu bāzes vadības sistēmu.

Atslēgvārdi: programmatūras monitorings, kļūdu paziņojumi, programmatūras izstrāde, NodeJS, PostgreSQL.

ABSTRACT

Application monitoring tool is an online helper tool for software developers. Its main function is to receive and process error reports and system logs sent by other software applications. The application monitoring tool processes and gathers the received data in one place and sends notification e-mails in case of a major bug. Potential users of this tool can access all gathered data using a web application.

The project is developed using modern development technologies based on the NodeJS environment. The data layer of the application is based on PostgreSQL data base.

Keywords: software monitoring, error reports, software development, NodeJS, PostgreSQL.

SATURS

| | |
|--|----|
| Ievads | 8 |
| Nolūks | 8 |
| Darbības sfēra..... | 8 |
| Saistība ar citiem dokumentiem | 8 |
| Pārskats | 8 |
| Apzīmējumu saraksts | 9 |
| 1. Vispārējais apraksts..... | 10 |
| 1.1 Esošā stāvokļa apraksts..... | 10 |
| 1.2 Pasūtītājs | 10 |
| 1.3 Produkta perspektīva..... | 10 |
| 1.4 Produkta funkcionalitāte | 10 |
| 1.5 Sistēmas lietotāji | 11 |
| 1.6 Sistēmas ierobežojumi | 12 |
| 1.7 Pieņēmumi un atkarības | 12 |
| 2. Programmatūras prasību specifikācija | 13 |
| 2.1 Konceptuālais ER modelis..... | 13 |
| 2.2 Nefunkcionālās prasības | 14 |
| 2.2.1 Veiktspējas prasības | 14 |
| 2.2.2 Drošības prasības..... | 14 |
| 2.2.3 Pieejamības prasības | 14 |
| 2.3 Funkcionālās prasības | 15 |
| 2.3.1 Autentifikācijas modulis | 16 |
| 2.4.1.1 AUTH-1 Sistēmas lietotāju autentifikācija | 17 |
| 2.4.1.2 AUTH-2 Monitorēto lietotņu autentifikācija | 18 |

| | |
|--|----|
| 2.4.1.3 AUTH-3 Lietotāju tiesību pārbaude..... | 18 |
| 2.4.2 Lietotāju un tiesību pārvaldības modulis | 19 |
| 2.4.2.1 USR-1 Lietotāju saraksts | 21 |
| 2.4.2.2 USR-2 Jauna lietotāja izveidošana..... | 21 |
| 2.4.2.3 USR-3 Lietotāja datu iegūšana..... | 22 |
| 2.4.2.4 USR-4 Lietotāja labošana | 22 |
| 2.4.2.5 USR-5 Lietotāja dzēšana..... | 23 |
| 2.4.3 Lietotņu pārvaldības modulis | 23 |
| 2.4.3.1 APP-1 Lietotņu saraksts..... | 25 |
| 2.4.3.2 APP-2 Jaunas lietotnes pievienošana | 25 |
| 2.4.3.3 APP-3 Lietotnes datu iegūšana | 26 |
| 2.4.3.4 APP-4 Lietotnes labošana | 26 |
| 2.4.3.5 APP-5 Lietotnes dzēšana..... | 27 |
| 2.4.4 Reģistrēto kļūdu modulis | 27 |
| 2.4.4.1 ERR-1 Kļūdas reģistrēšana | 28 |
| 2.4.4.2 ERR-2 Kļūdu saraksts | 29 |
| 2.4.4.3 ERR-3 Kļūdas paziņojuma datu iegūšana..... | 30 |
| 2.4.4.4 ERR-4 Kļūdas atrisināšana | 30 |
| 2.4.5 Reģistrēto notikumu modulis | 30 |
| 2.4.5.1 EVT-1 Notikuma reģistrēšana | 31 |
| 2.4.5.2 EVT-2 Notikumu saraksts..... | 32 |
| 2.4.5.3 EVT-3 Notikuma datu iegūšana..... | 32 |
| 2.4.6 Reģistrēto auditācijas pierakstu modulis..... | 33 |
| 2.4.6.1 LOG-1 Auditācijas pieraksta reģistrēšana | 34 |
| 2.4.6.2 LOG-2 Auditācijas pierakstu saraksts..... | 34 |
| 2.4.6.3 LOG-3 Auditācijas pierakstu straume..... | 35 |
| 2.4.7 Sistēmas pārvaldības modulis | 35 |

| | | |
|---------|---|----|
| 2.4.7.1 | SYS-1 E-pasta izsūtīšana..... | 36 |
| 2.4.7.2 | SYS-2 E-pastu saraksts | 37 |
| 2.4.7.3 | SYS-3 Sistēmas auditācijas pierakstu saraksts..... | 37 |
| 2.4.7.4 | SYS-4 Sistēmas auditācijas pieraksta reģistrēšana | 38 |
| 3. | Programmatūras projektējuma apraksts | 39 |
| 3.1 | Datu bāzes projektējums | 39 |
| 3.1.1 | Datu bāzes fiziskais modelis | 39 |
| 3.1.2 | Tabulu projektējums..... | 41 |
| 3.1.2.1 | Tabula “users” | 41 |
| 3.1.2.2 | Tabula “projects” | 42 |
| 3.1.2.3 | Tabula “events” | 42 |
| 3.1.2.4 | Tabula “errors” | 43 |
| 3.1.2.5 | Tabula “error_events” | 43 |
| 3.1.2.6 | Tabula “emails” | 43 |
| 3.1.2.7 | Tabula “system_logs”..... | 44 |
| 3.1.2.8 | Starptabula “project_users” | 44 |
| 3.1.2.9 | Lietotņu auditācijas pierakstu krātuve..... | 45 |
| 3.2 | Funkciju dekompozīcija..... | 45 |
| 3.2.1 | Autentifikācijas moduļa dekompozīcija..... | 45 |
| 3.2.2 | Lietotāju pārvaldības moduļa dekompozīcija | 45 |
| 3.2.3 | Lietotņu pārvaldības moduļa dekompozīcija | 46 |
| 3.2.4 | Reģistrēto kļūdu moduļa dekompozīcija..... | 46 |
| 3.2.5 | Reģistrēto notikumu moduļa dekompozīcija..... | 46 |
| 3.2.6 | Reģistrēto auditācijas pierakstu moduļa dekompozīcija | 46 |
| 3.2.7 | Sistēmas pārvaldības moduļa dekompozīcija..... | 47 |
| 3.3 | Saskarnes projektējums..... | 47 |
| 3.3.1 | Pieteikšanās forma..... | 47 |

| | | |
|-------|--|----|
| 3.3.2 | Datu tabula | 48 |
| 3.3.3 | Datu labošanas/pievienošanas forma | 49 |
| 3.3.4 | Sistēmas paziņojumi..... | 50 |
| 4. | Testēšanas dokumentācija | 51 |
| 4.1 | Vienībtestēšanas testpiemēri..... | 51 |
| 4.1.1 | Autentifikācijas moduļa testpiemēri..... | 51 |
| 4.1.2 | Lietotāju pārvaldības moduļa testpiemēri | 51 |
| 4.1.3 | Lietotņu pārvaldības moduļa testpiemēri | 52 |
| 4.1.4 | Reģistrēto kļūdu moduļa testpiemēri..... | 52 |
| 4.1.5 | Reģistrēto notikumu moduļa testpiemēri | 52 |
| 4.1.6 | Reģistrēto auditācijas pierakstu moduļa testpiemēri..... | 53 |
| 4.1.7 | Sistēmas pārvaldības moduļa testpiemēri | 53 |
| 4.2 | Automatizētās testēšanas scenāriji..... | 53 |
| 4.3 | Testēšanas žurnāls..... | 53 |
| 5. | Projekta pārvaldība..... | 55 |
| 5.1 | Projekta organizācija..... | 55 |
| 5.2 | Konfigurācijas pārvaldība..... | 55 |
| 5.3 | Kvalitātes nodrošināšana | 56 |
| 6. | Darbietilpības novērtēšana | 57 |
| | Nobeigums | 58 |
| | Izmantotie avoti..... | 59 |
| | Pielikumi | 60 |

IEVADS

Nolūks

Kvalifikācijas darba mērķis ir izstrādāt programmatūras lietotņu monitoringa risinājumu, kas atvieglotu programmatūras izstrādātāju un uzturētāju darbu, paātrinot problēmu identificēšanas un atrisināšanas procesa gaitu. Risinājumu sākotnēji plānots ieviest un izmantot informācijas sistēmu izstrādē un uzturēšanā autora prakses uzņēmumā. Risinājums tiks padarīts pieejams, pielāgojams un uzlabojams ikvienam interesentam atvērtā koda veidā.

Darbības sfēra

Izstrādātā programmatūras risinājuma mērķis ir nodrošināt vidi, kas vienuviet ievāc un uzkrāj izstrādājamo un uzturamo programmatūras produktu kļūdu paziņojumus, auditācijas pierakstus un procesu parametrus, tādejādi atvieglot problēmu identificēšanu, atrisināšanu un apziņošanu avārijas gadījumā.

Saistība ar citiem dokumentiem

Šī dokumenta saturā esošā programmatūras prasību specifikācija ir izstrādāta saskaņā ar standarta LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis” ieteikumiem. Saturā esošais programmatūras projektējuma apraksts veidots ņemot vērā Latvijas valsts standarta LVS 72:1996, “Ieteicamā programmatūras projektējuma apraksts” vadlīnijas. Programmatūras testēšanas dokumentācija izstrādāta vadoties pēc Latvijas valsts standarta LVS 70:1996, „Programmatūras testēšanas dokumentācija” vadlīnijām.

Pārskats

Šis dokuments satur šādas sadaļas:

- Ievads, kurā definēts kvalifikācijas darba ietvaros izstrādātās programmatūras un tās dokumentācijas nolūks, darbības sfēra, kā arī aprakstīta saistība ar citiem dokumentiem. Ievads satur arī apzīmējumu un saīsinājumu sarakstu.
- Pirmā nodaļa – Vispārējais apraksts, kurā īsumā aprakstītas izstrādātās programmas pamatiezīmes, raksturlielumi un ierobežojumi.
- Otrā nodaļa – Programmatūras prasību specifikācija, kas ietver funkcionālo un nefunkcionālo prasību aprakstus.
- Trešā nodaļa – Programmatūras projektējuma apraksts, kurā definēts detalizēts datu bāzes projektējums, funkciju dekompozīcija un lietotāju saskarnes projektējums.

- Ceturtā nodaļa – Testēšanas dokumentācija, kas iezīmē izstrādes laikā lietoto testu gaitu.
- 5. Projekta pārvaldība, 6. darbietilpības novērtēšana un nobeiguma daļa.

Apzīmējumu saraksts

| Apzīmējums | Skaidrojums |
|------------|---|
| kebab-case | Pieraksta formāts vārdus rakstot ar mazajiem burtiem un atdalot ar domu zīmi. |
| JSON | JavaScript Objektu Notācija (<i>JavaScript Object Notation</i>). Datu apmaiņas un uzglabāšanas formāts. |
| API | Lietojumprogrammas saskarne (<i>Application Programming Interface</i>). |
| REST API | Lietojumprogrammas saskarne, kas būvēta pēc RESTful API principiem |
| OWASP | <i>Open Web Application Security Project</i> ir tīmekļa komūna, kas specializējas tīmekļa drošības jomā |
| Jenkins | Nepārtrauktās integrācijas, piegādes un automatizācijas rīks. |
| SQL | Strukturēto vaicājumu valoda |
| NoSQL | Datu bāžu uzbūves un datu uzglabāšanas metodoloģija, kas paredz brīvu datu uzglabāšanas shēmu pretēji <i>SQL</i> tabulu un kolonnu uzbūvei. |

1. VISPĀRĒJAIS APRAKSTS

1.1 Esošā stāvokļa apraksts

Globālajā tīmeklī ir pieejami dažādi lietotņu monitoringa risinājumi, taču to izpildījumā sastopamas tādas problēmas kā risinājums ir novecojis vai slikti uzturēts, risinājums tiek piedāvāts kā *SaaS (Software as a Service)*¹, līdz ar to lielākajā daļā gadījumu ir par maksu, kā arī nav uzstādāms kā iekšējais risinājums, vai arī risinājums balstīts tikai uz kādu konkrētu platformu, līdz ar to ir ierobežota tā lietošana dažādās platformās un programmēšanas valodās.

1.2 Pasūtītājs

Aprakstītā risinājuma pasūtītājs ir programmatūras risinājumu izstrādes uzņēmums SIA “E-Synergy”. Potenciālie risinājuma lietotāji būs pasūtītāja nodarbinātie programmatūras izstrādātāji un uzturētāji, kā arī programmētāji ārpus uzņēmuma, jo risinājumu plānots piedāvāt ikvienam, atvērtā koda veidā.

1.3 Produkta perspektīva

Aprakstītais monitoringa risinājums savā būtībā paredz sasaisti ar jebkādiem programmatūras izstrādājumiem, kas spēj, izmantojot tīmekli, sūtīt savas darbības auditācijas pierakstus, kļūdas paziņojumus un datus par citiem notikumiem. Monitoringa risinājums sūtītos datus saņem, apstrādā un saglabā atrādīšanai.

Monitoringa risinājums lieto tīmekļa pakalpojumu *Gravatar*², lietotāju kontu attēlu ģenerēšanai.

1.4 Produkta funkcionalitāte

Lietotņu monitoringa risinājuma galvenais uzdevums ir nodrošināt auditācijas pierakstu un kļūdas paziņojumu ievākšanu, apstrādi, atrādīšanu un glabāšanu no dažādiem programmatūras produktiem. Monitoringa risinājumam ir tīmekļa lietotnes saskarne, kas tā lietotājam ļauj izveidot un organizēt piekļuves tiesības citiem lietotājiem, uzstādīt auditācijas pierakstu un kļūdu paziņojumu pieejas citiem programmatūras produktiem un apskatīt ievāktos

¹Software as a Service// <http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service>

² Gravatar// <https://en.gravatar.com/>

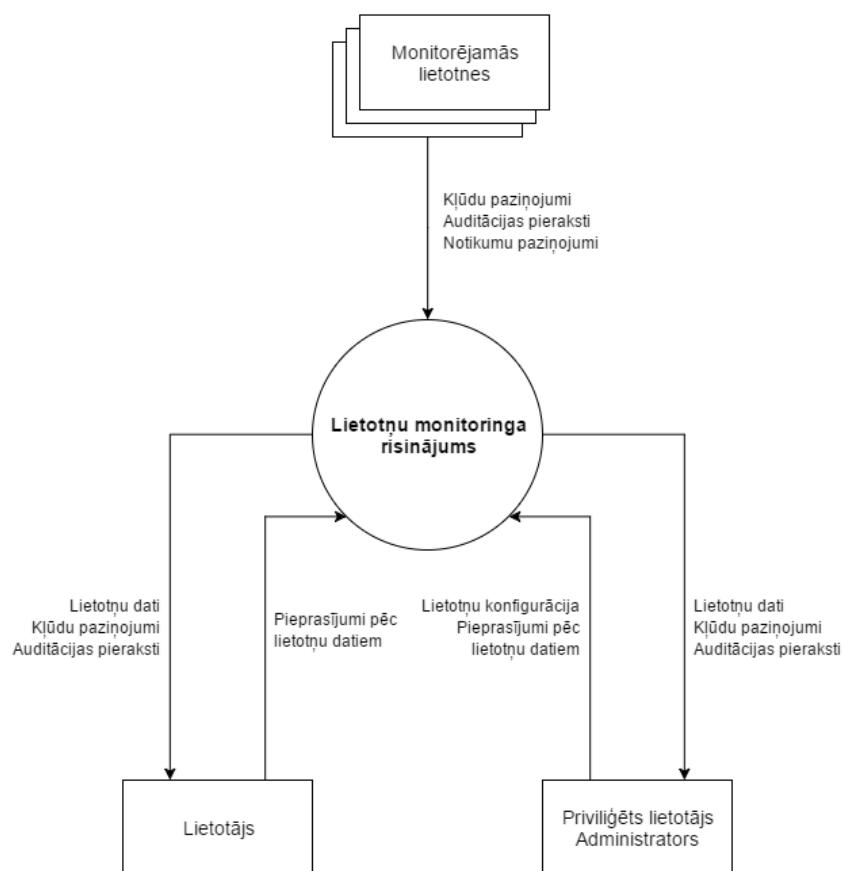
datus no monitorētajiem produktiem. Monitoringa risinājums nodrošina lietotāju apziņošanu pa e-pastu, kļūdas paziņojuma saņemšanas gadījumā.

1.5 Sistēmas lietotāji

Risinājuma lietotāji iedalās 3 lietotāju grupās:

- Parasts lietotājs, kurš var pieteikties sistēmā un apskatīt tam piešķirto lietotņu iekrātos auditācijas pierakstus, kļūdas paziņojumus un citus datus.
- Privilēģēts lietotājs, kurš var izveidot un konfigurēt sevis izveidotas lietotnes.
- Administrators, kurš var izveidot un labot risinājuma lietotāju kontus, apskatīt un rediģēt jebkuru risinājumā reģistrēto lietotni un apskatīt sistēmas auditācijas pierakstus.

Katra nākamā lietotāju grupa var darīt arī to, ko iepriekšējā. Neregistrētiem lietotājiem nav pieejas sistēmai.



1.1. att. Nultā līmeņa datu plūsmu diagramma

Monitoringa risinājuma lietotājiem jābūt pamatzināšanām par tīmekļa pārlūkprogrammas lietošanu un izpratnei par modernas lietotāja saskarnes darbību.

Lai pilnvērtīgi izmantotu risinājuma piedāvātās lietotājam jābūt zināšanām un izpratnei par programmatūras kļūdu paziņojumiem un auditācijas pierakstiem, vēlams arī par to avotu – pašiem programmatūras izstrādājumiem.

Lai uzstādītu monitoringa risinājuma sasaisti ar saistītajām programmatūras lietotnēm, lietotājam jābūt zināšanām par saistīto lietotņu uzbūvi un jāpārvalda to izstrādē lietotā programmēšanas valoda un pamatprincipi.

1.6 Sistēmas ierobežojumi

Risinājumā jaunus lietotājus iespējams izveidot tikai esošam lietotājam – administratoram. Jaunam lietotājam pašam reģistrēties nav iespējams. Risinājumu uzstādot, pieejams viens iepriekš uzstādīts lietotājs ar administratora tiesībām.

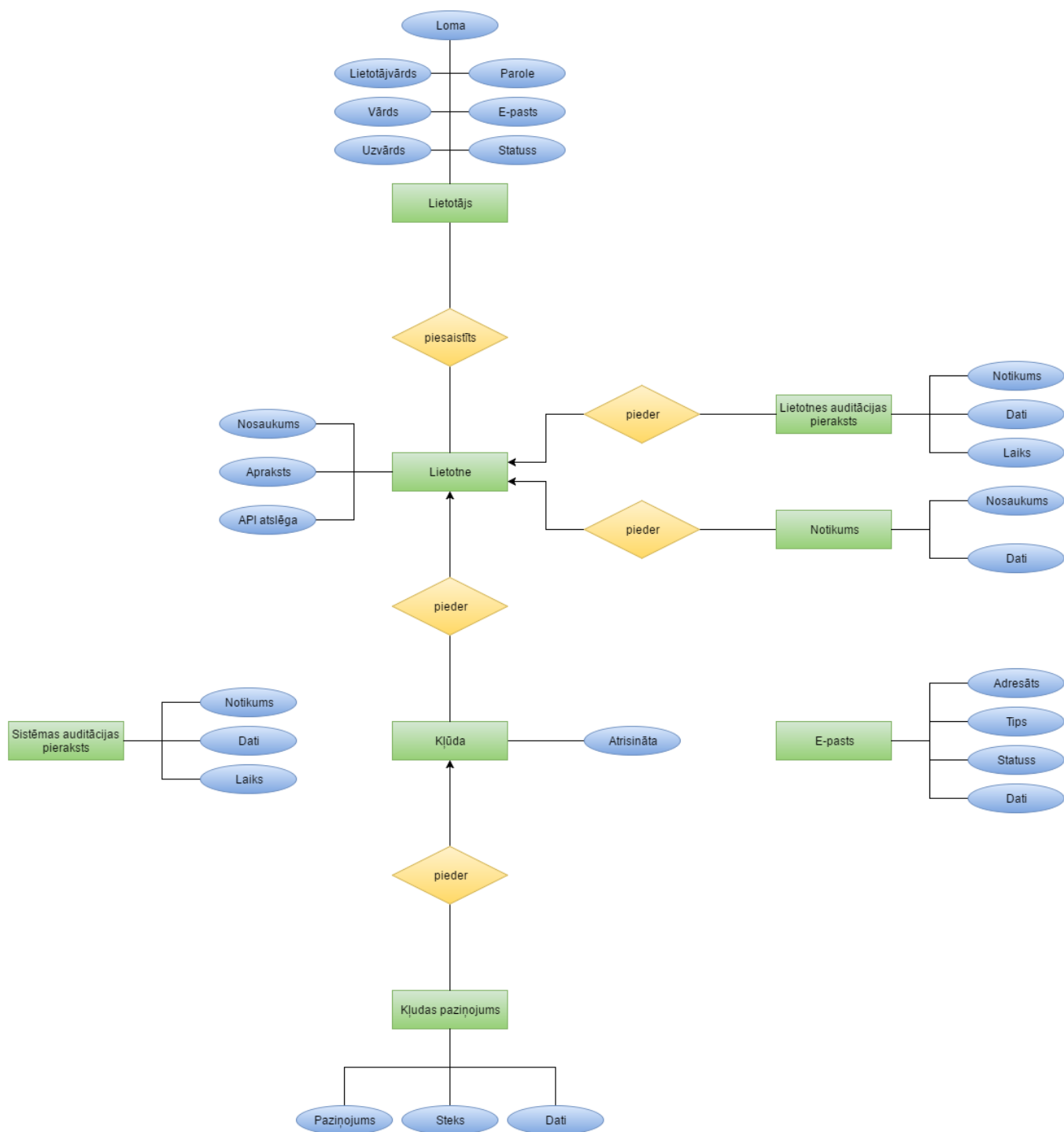
1.7 Pieņēmumi un atkarības

Sistēmas lietotāja saskarne tiek izstrādāta angļu valodā, dokumentācija tiek izstrādāta latviešu valodā. Tiek pieņemts, ka tulkošana tiek veikta izstrādes procesā.

Tiek pieņemts, ka sistēmai ir piekļuve globālajam tīmeklim, lai saņemtu kļūdas paziņojumus no saistītajām lietotnēm. Pieļaujama arī darbība lokālā tīkla ietvaros.

2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

2.1 Konceptuālais ER modelis



2.1. att. Datu bāzes konceptuālais ER modelis

Lietotņu monitoringa risinājuma datu bāze sastāv no 8 entītijām. Lietotne var būt saistīta ar vairākiem lietotājiem un otrādi, veidojot saiti – daudz pret daudz. Lietotnei var būt vairāki auditācijas pieraksti, notikumi un kļūdas, veidojot saiti – viens pret vairākiem, kur viena ir

lietotne. Kļūdai var būt vairāki paziņojumi, veidojot saiti – viens pret vairākiem, kur viena ir kļūda. Sistēmas auditācijas pieraksti un e-pasti ir neatkarīgas entītijas un tām saišu nav.

2.2 Nefunkcionālās prasības

2.2.1 Veiktspējas prasības

Risinājums spēj nodrošināt 20 lietotāju vienlaicīgu sistēmas lietošanu, kuri vidēji ikkatras 10 sekundes izdara pieprasījumu pēc datiem vai jaunas lapas. Risinājuma atbildes laiks (neskaitot laiku, ko paņem datu sūtīšana pa tīklu), ir vidēji zem 1000 ms uz pieprasījumu un 95% atbilžu ir zem 2000 ms.

Risinājums spēj nodrošināt 10 monitorējamo lietotņu apkalpošanu, kuras vidēji ik pēc 10 sekundēm iesūta datus par savu darbību. Risinājums sūta atbildi monitorētajām lietotnēm nekavējoties, negaidot iesūtīto datu apstrādi. Iesūtīto datu apstrāde notiek neatkarīgi.

2.2.2 Drošības prasības

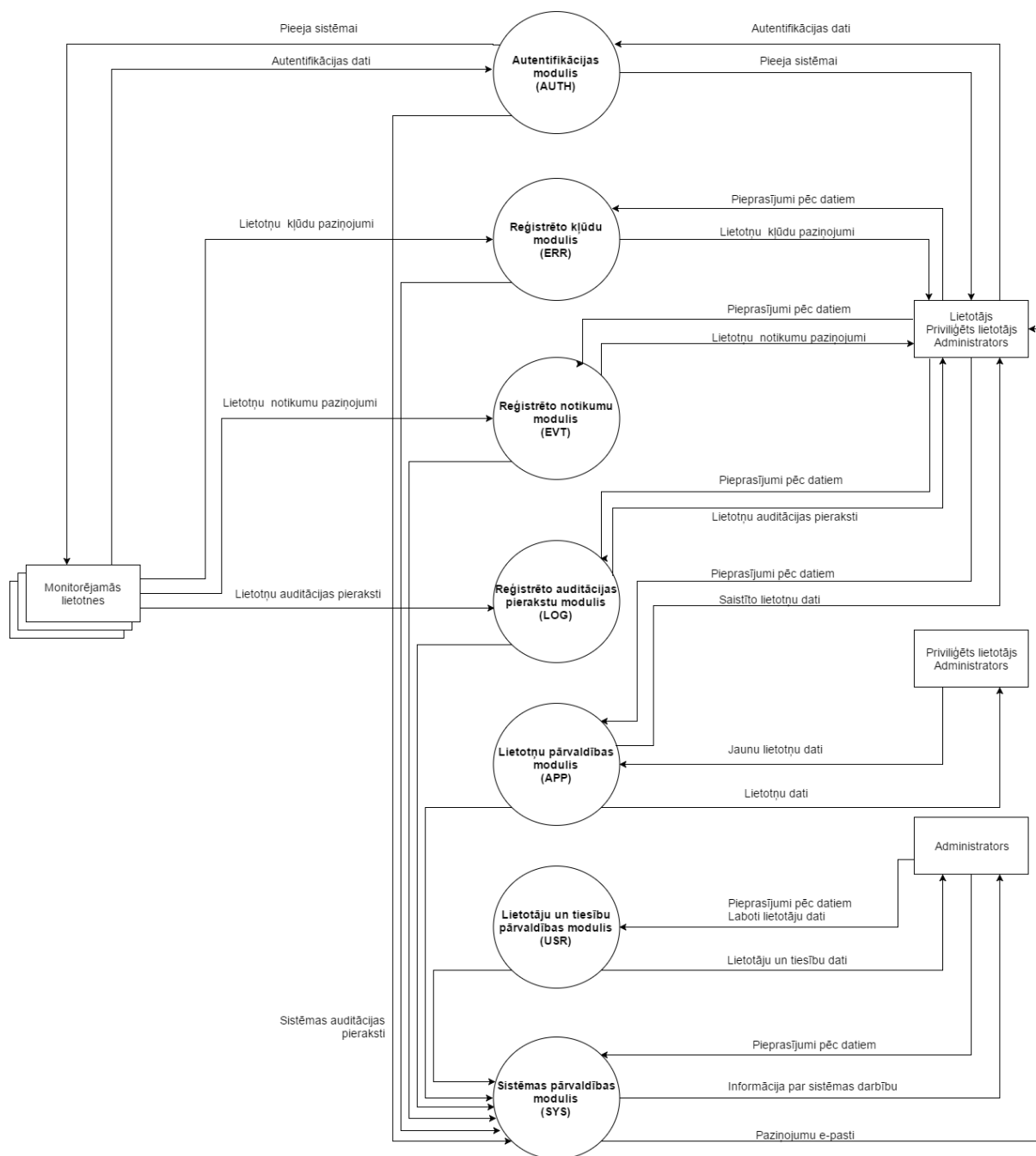
Risinājums nedrīkst pieļaut OWASP³ aprakstītās visvairāk izplatītās drošības ievainojamības. Jānodrošina lietotāja ievadīto datu un monitorēto lietotņu iesūtīto datu validācija, lai samazinātu uzbrukuma iespējas. Kļūdu un sistēmas paziņojumi neatklāj lietotājiem nekādu informāciju par risinājumu. Detalizēti paziņojumi redzami tikai administratoriem.

2.2.3 Pieejamības prasības

Izstrādātajam risinājumam jābūt pieejamam jebkurā dienā un diennakts laikā bez darbības pārtraukumiem. Pieļaujami darbības pārtraukumi uzturēšanas vajadzībām, kas nepārsniedz 6 stundas mēnesī.

³ OWASP// <https://www.owasp.org/>

2.3 Funkcionālās prasības



2.2. att. *Pirmā līmeņa datu plūsmu diagramma*

Biežāk sastopamie funkciju izvades paziņojumi uzskaitīti sekojošajās paziņojumu tabulās un to identifikatori izmantoti funkciju aprakstos.

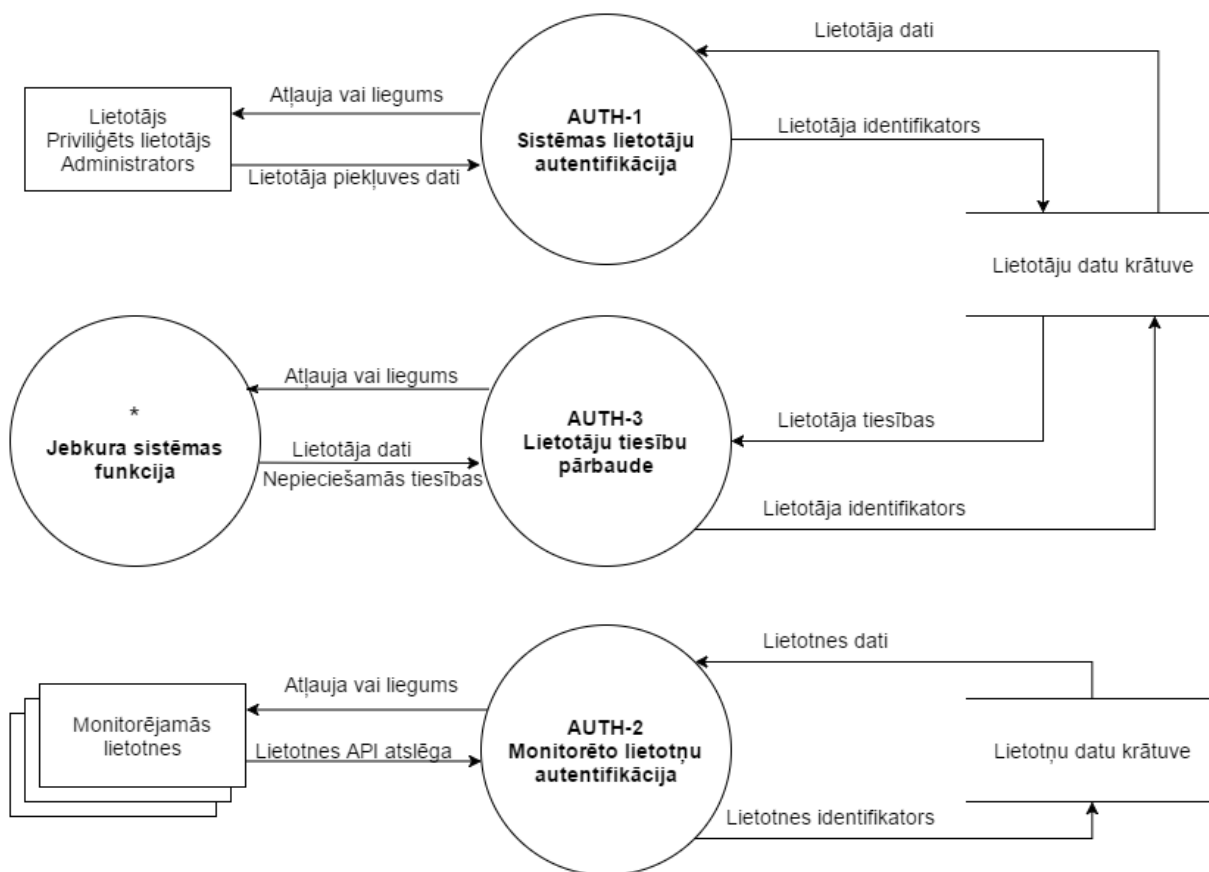
| Identifikators | Paziņojums un iemesls | Atbildes HTTP kods |
|----------------|---|--------------------|
| S1 | Apstiprinājums par veiksmīgu darbības iznākumu. | 200 |
| S2 | Sarakstā netika atlasīts neviens ieraksts. | 200 |

| Identifikators | Skaidrojums | Atbildes HTTP kods |
|----------------|--|--------------------|
| E1 | Paziņojums informē, ka nav aizpildīti visi obligātie lauki vai vērtības, kā arī par nekorekti aizpildītajiem laukiem. Paziņojuma piemēri: <ul style="list-style-type: none"> • <lauka nos.> ir obligāts! • <lauka nos.> nav norādīts korekti. | 400 |
| E2 | Paziņojums informē, ka lietotājs nav autentificējies, izpaužas kā pārvirzīšana un autentifikācijas skatu. | 401 |
| E3 | Paziņojums informē, ka lietotājam nav tiesību veikt darbību, izpaužas kā pārvirzīšana uz sākumlapu vai paziņojums formā: <ul style="list-style-type: none"> • Jums nav tiesību veikt šo darbību. | 403 |
| E4 | Paziņojums informē, ka pieprasītais resurss netika atrasts formā: <ul style="list-style-type: none"> • <resurss> netika atrasts! | 404 |
| E5 | Paziņojums informē, ka sistēmā notikusi kļūda. Piemērs: <ul style="list-style-type: none"> • Sistēmas kļūda! | 500 |

Tiek pieņemts, ka jebkura funkcija var kā kļūdas paziņojumu atgriezt E5, gadījumā, ja notikusi sistēmas kļūda, E2 – gadījumā, ja lietotājs nav autentificējies vai E3, ja lietotājam nav tiesību veikt konkrēto darbību. Īpaši gadījumi paskaidroti pie katras funkcijas.

2.3.1 Autentifikācijas modulis

| |
|--|
| Mērķis |
| Modulis atbild par lietotāju un monitorēto lietotņu autentificēšanos un tiesību pārbaudi. |
| Pieejamība |
| Moduļa funkcijas AUTH-1 un AUTH-2 pieejamas bez autentificēšanās, jo pašas pilda autentificēšanas funkciju. Funkcija AUTH-3 pieejama tikai autentificētiem lietotājiem, jo pēc būtības nevar funkcionēt, ja lietotājs nav autentificējies. |



2.3. att. Autentifikācijas moduļa otrā līmeņa datu plūsmu diagramma

2.4.1.1 AUTH-1 Sistēmas lietotāju autentifikācija

| Mērķis | | | | |
|--|---------------|----------------|-----------------|-----------|
| Šai funkcijai jānodrošina sistēmā reģistrēto lietotāju autentifikācija. | | | | |
| Ievaddati | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? |
| 1. | Lietotājvārds | Simbolu virkne | 1-15 simboli | Jā |
| 2. | Parole | Simbolu virkne | 1-255 simboli | Jā |
| Apstrāde | | | | |
| Atlasa lietotāju pēc ievadītā lietotājvārda un pārbauda vai ievadītā parole sakrīt, vai lietotājs ir aktīvs un nav dzēsts. Ja pieteikšanās veiksmīga, lietotāju pārvirza uz sākuma lapu. | | | | |
| Izvaddati | | | | |
| Apstiprinājums par veiksmīgu pieteikšanos sistēmā. | | | | |
| Kļūdu paziņojumi | | | | |

E1, ja nav aizpildīti visi lauki.

Paziņojums par nepareiziem datiem(E2), ja netika atrasts attiecīgs lietotājs, nesakrita parole, tas bija neaktīvs vai dzēsts.

2.4.1.2 AUTH-2 Monitorēto lietotņu autentifikācija

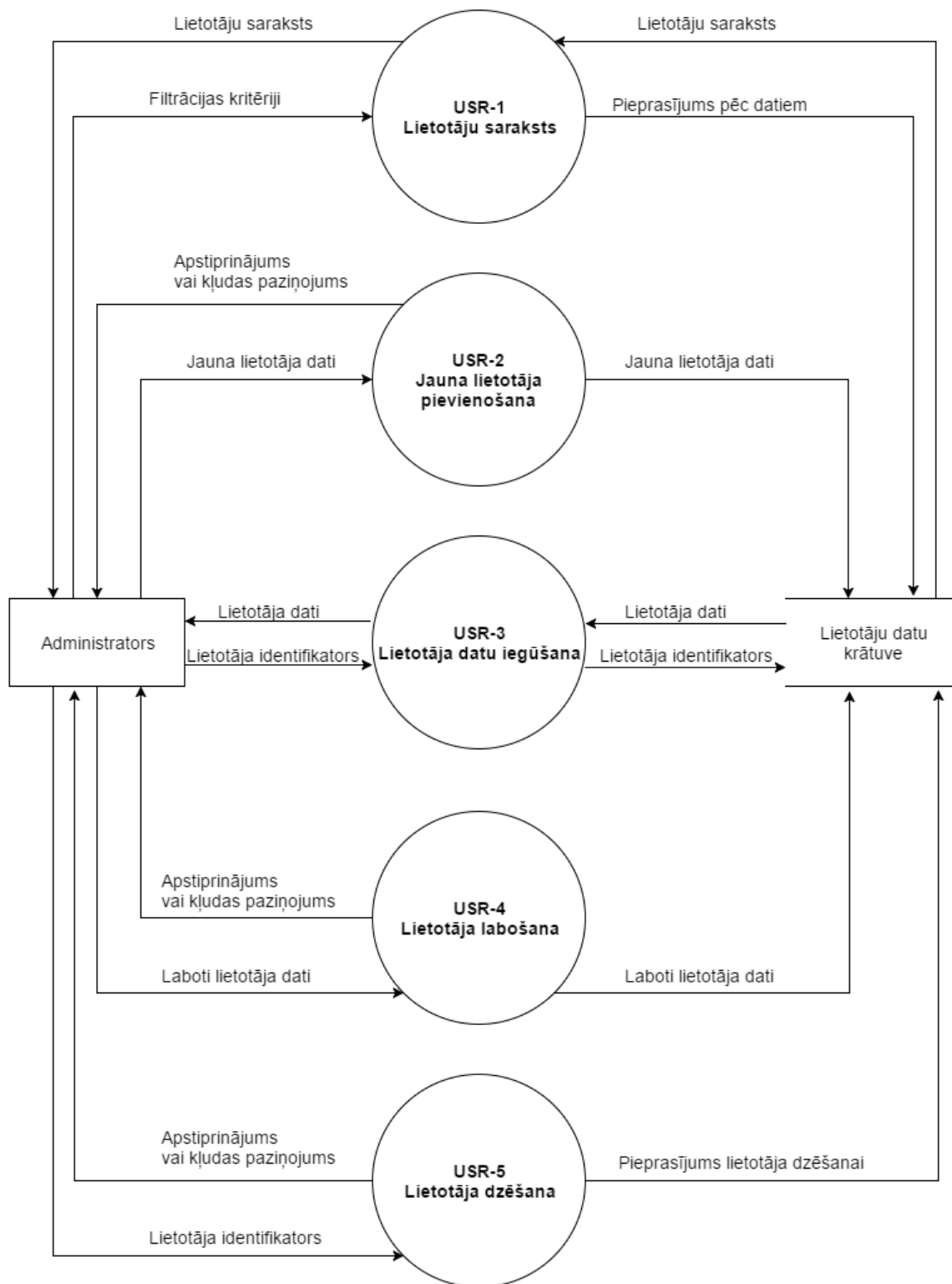
| | | | | |
|--|----------------|----------------|-----------------|-----------|
| Mērķis | | | | |
| Šai funkcijai jānodrošina sistēmā reģistrēto lietotņu autentifikācija. | | | | |
| Ievaddati | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? |
| 1. | Identifikators | Simbolu virkne | 1-60 simboli | Jā |
| 2. | API atslēga | Simbolu virkne | 32 simboli | Jā |
| Apstrāde | | | | |
| Atlasa lietotni pēc norādītā identifikatora un pārbauda vai norādītā API atslēga sakrīt, vai lietotne ir aktīva un nav dzēsta. | | | | |
| Izvaddati | | | | |
| Atļauja piekļūt sistēmai. | | | | |
| Kļūdu paziņojumi | | | | |
| E1, ja nav aizpildīti visi lauki. | | | | |
| E2, ja netika atrasta attiecīga lietotne, nesakrita API atslēga, tā bija neaktīva vai dzēsta. | | | | |

2.4.1.3 AUTH-3 Lietotāju tiesību pārbaude

| | | | | |
|---|--|--|--|--|
| Mērķis | | | | |
| Šai funkcijai jānodrošina lietotāju tiesību pārbaude attiecībā pret pieprasīto darbību. | | | | |
| Ievaddati | | | | |
| Ievaddati ir veicamā darbība un darbības pieprasītājam piešķirtās tiesības. | | | | |
| Apstrāde | | | | |
| Pārbauda vai lietotājam ir tiesības veikt norādīto darbību | | | | |
| Izvaddati | | | | |
| Atļauja veikt darbību. | | | | |
| Kļūdu paziņojumi | | | | |
| E3, ja lietotājam nav tiesību veikt darbību. | | | | |

2.4.2 Lietotāju un tiesību pārvaldības modulis

| |
|---|
| Mērķis |
| Modulis atbild par risinājuma lietotāju, ar tiem saistīto datu un tiesību pārvaldību. |
| Pieejamība |
| Moduļa funkcijas pieejamas tikai sistēmas administratoriem. |



2.4. att. Lietotāju pārvaldības moduļa otrā līmeņa datu plūsmu diagramma

2.4.2.1 USR-1 Lietotāju saraksts

| Mērķis | | | | | |
|---|---------------------------------|---------------------------|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina sistēmā reģistrēto lietotāju saraksta atrādīšana, filtrēšana un kārtošana. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Atslēgvārdi | Simbolu virkne | 1-255 simboli | Nē | |
| 2. | Kārtošanas virziens un atribūts | Atribūts, augoši/dilstoši | 1 atribūts | Nē | ID/ augoši |
| 3. | Lapa | Skaitlis | 1 - n | Nē | 1 |
| Apstrāde | | | | | |
| Atlasa pieprasītos lietotājus pēc meklēšanas kritērijiem un sakārto pieprasītajā secībā pēc pieprasītās kolonnas. | | | | | |
| Izvaddati | | | | | |
| Saraksts ar sistēmas lietotājiem vai S2, ja neviens netika atrasts | | | | | |
| Kļūdu paziņojumi | | | | | |
| Atrunāti nodaļas sākumā. | | | | | |

2.4.2.2 USR-2 Jauna lietotāja izveidošana

| Mērķis | | | | | |
|---|------------------|--|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina jauna lietotāja reģistrācija sistēmā. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Vārds | Simbolu virkne | 1-30 simboli | Nē | |
| 2. | Uzvārds | Simbolu virkne | 1-30 simboli | Nē | |
| 3. | Lietotājvārds | Simbolu virkne | 1 – 15 simboli | Jā | |
| 4. | Parole | Simbolu virkne | 1 – 255 simboli | Jā | |
| 5. | Parole atkārtoti | Simbolu virkne | 1 – 255 simboli | Jā | |
| 6. | E-pasts | Simbolu virkne | 1 – 255 simboli | Jā | |
| 7. | Statuss | Aktīvs/ neaktīvs | 1 statuss | Jā | Aktīvs |
| 8. | Loma | Lietotājs/privilēģēts lietotājs/administrators | 1 loma | Jā | Lietotājs |
| Apstrāde | | | | | |

| |
|---|
| Pārbauda vai aizpildītas visas obligātās vērtības, vai visas vērtības aizpildītas korekti, vai sakrīt abi paroļu lauki. Vērtību formāts: <ul style="list-style-type: none"> E-pasts: xxxx@yyyy.vv |
| Izvaddati |
| Izveidotā lietotāja identifikators, un paziņojums S1. |
| Kļūdu paziņojumi |
| Nekorektu datu gadījumā paziņojums E1. Citi gadījumi atrunāti nodaļas sākumā. |

2.4.2.3 USR-3 Lietotāja datu iegūšana

| |
|--|
| Mērķis |
| Šai funkcijai jānodrošina lietotāja datu atrādīšana. |
| Ievaddati |
| Funkcijas ievaddati ir pieprasītā lietotāja identifikators (ID). |
| Apstrāde |
| Atlasa pieprasīto lietotāju |
| Izvaddati |
| Pieprasītā lietotāja dati, un paziņojums S1. |
| Kļūdu paziņojumi |
| E4, ja lietotājs netika atrasts. Citi gadījumi atrunāti nodaļas sākumā. |

2.4.2.4 USR-4 Lietotāja labošana

| |
|--|
| Mērķis |
| Šai funkcijai jānodrošina eksistējoša lietotāja datu maiņa. |
| Ievaddati |
| Ievaddati ir labojamā lietotāja identifikators un labotie lietotāja dati. (Ievaddatu vērtības skatīt funkcijā USR-2) |
| Apstrāde |
| Apstrādes nosacījumus skatīt funkcijā USR-2. |
| Izvaddati |

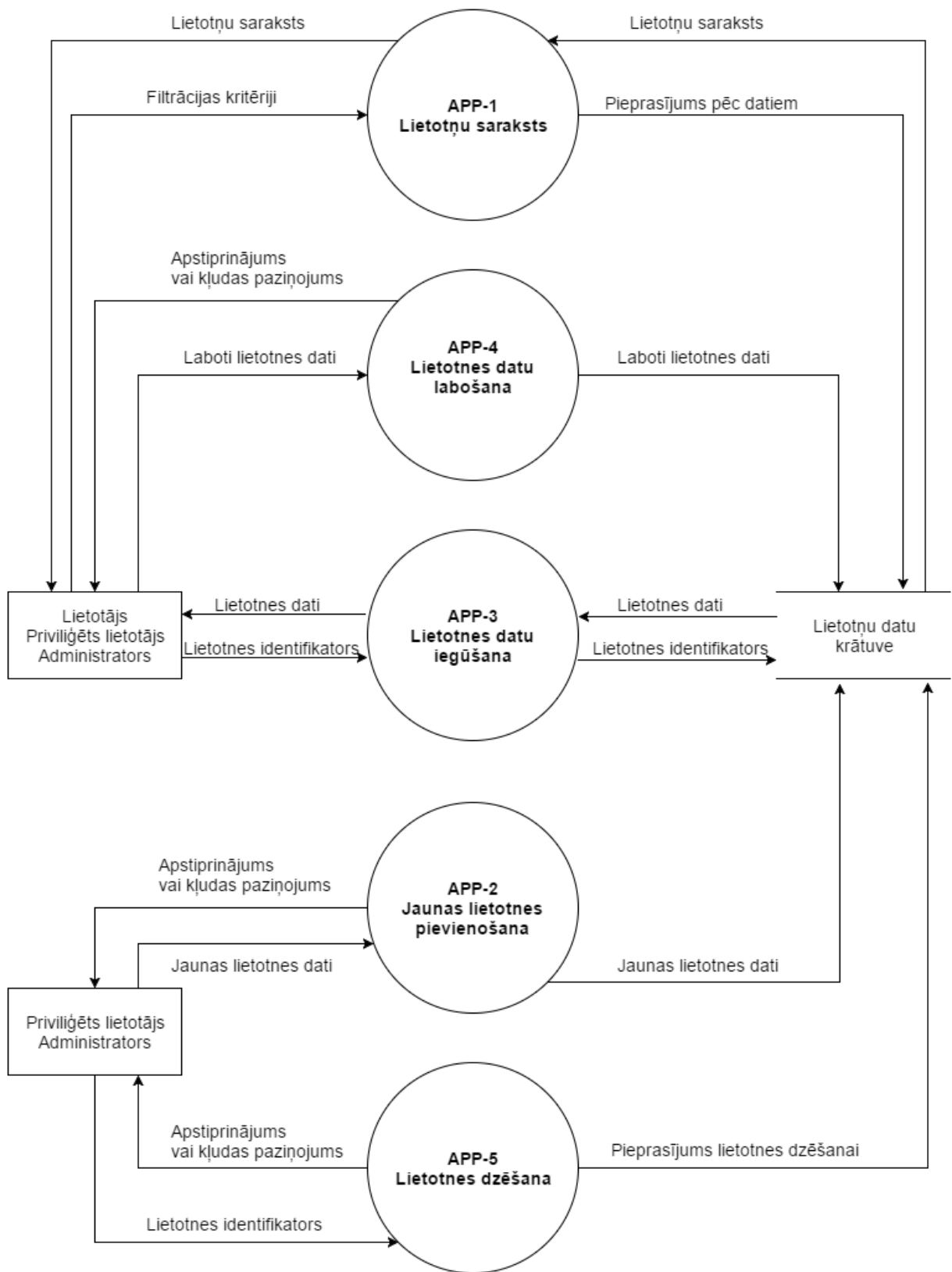
| |
|--|
| Korektu datu ievades gadījumā – paziņojums S1. |
| Kļūdu paziņojumi |
| Nekorektu datu gadījumā paziņojums E1. E4, ja lietotājs netika atrasts. Citi gadījumi atrunāti nodaļas sākumā. |

2.4.2.5 USR-5 Lietotāja dzēšana

| |
|---|
| Mērķis |
| Šai funkcijai jānodrošina lietotāja dzēšana. |
| Ievaddati |
| Dzēšamā lietotāja identifikators ID. |
| Apstrāde |
| Pārbauda vai lietotājs nedzēš pats savu kontu. Uzstāda norādītā lietotāja statusu – dzēsts. |
| Izvaddati |
| Dzēšanas izpildes gadījumā – paziņojums S1. |
| Kļūdu paziņojumi |
| E4, ja lietotājs netika atrasts. Paziņojums “Dzēst savu kontu nav iespējams!” gadījumā, ja lietotājs vēlas dzēst savu kontu. Citi gadījumi atrunāti nodaļas sākumā. |

2.4.3 Lietotņu pārvaldības modulis

| |
|--|
| Mērķis |
| Modulis atbild par monitorēto lietotņu reģistrēšanu sistēmā, to piekļuves pārvaldību. |
| Pieejamība |
| Parastiem lietotājiem pieejamas tikai lietotnes, kurās tie reģistrēti kā dalībnieki. Labot un dzēst lietotni var lietotāji, kas tajā reģistrēti ar lomu administrators. Priviliģēti lietotāji var izveidot lietotnes un veikt visas iepriekšminētās darbības ar tādiem pašiem nosacījumiem. Administratori var veikt visas iepriekš minētās darbības bez nosacījumiem. |



2.5. att. Lietotņu pģrvaldģbas modģļa otrģ lģmeņa datu plģsmu diagramma

2.4.3.1 APP-1 Lietotņu saraksts

| Mērķis | | | | | |
|--|---------------------------------|---------------------------|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina sistēmā pievienoto lietotņu saraksta atrādīšana, filtrēšana un kārtošana. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Atslēgvārdi | Simbolu virkne | 1-255 simboli | Nē | |
| 2. | Kārtošanas virziens un atribūts | Atribūts, augoši/dilstoši | 1 atribūts | Nē | ID/ augoši |
| 3. | Lapa | Skaitlis | 1 - n | Nē | 1 |
| Apstrāde | | | | | |
| Atlasa pieprasītās lietotnes pēc meklēšanas kritērijiem un sakārto pieprasītajā secībā pēc pieprasītās kolonnas. | | | | | |
| Izvaddati | | | | | |
| Saraksts ar lietotnēm vai S2, ja neviena netika atrasta. | | | | | |
| Kļūdu paziņojumi | | | | | |
| Atrunāti nodaļas sākumā. | | | | | |

2.4.3.2 APP-2 Jaunas lietotnes pievienošana

| Mērķis | | | | | |
|--|---------------------------|---------------------------|-------------------|-----------|-------------|
| Šai funkcijai jānodrošina jaunas lietotnes pievienošana sistēmā. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Nosaukums | Simbolu virkne | 1-30 simboli | Jā | |
| 2. | Apraksts | Simbolu virkne | 1-255 simboli | Nē | |
| 3. | Piederīgie lietotāji | Lietotājs | vairāki lietotāji | Nē | |
| 4. | Piederīgo lietotāju lomas | Dalībnieks/administrators | 1 loma | Nē | Dalībnieks |
| Apstrāde | | | | | |
| Pārbauda vai aizpildītas visas obligātās vērtības. | | | | | |
| Izvaddati | | | | | |
| Lietotnes API identifikators un API atslēga, paziņojums S1. | | | | | |

Kļūdu paziņojumi

Paziņojums E1, ja nav aizpildīti obligātie lauki.

Citi paziņojumi atrunāti nodaļas sākumā.

2.4.3.3 APP-3 Lietotnes datu iegūšana**Mērķis**

Šai funkcijai jānodrošina lietotnes datu atrādīšana.

Ievaddati

Lietotnes identifikators (ID).

Apstrāde

Atlasa pieprasīto lietotni.

Izvaddati

Pieprasītās lietotnes dati, paziņojums S1.

Kļūdu paziņojumi

E4, ja lietotne netika atrasta.

Citi paziņojumi atrunāti nodaļas sākumā.

2.4.3.4 APP-4 Lietotnes labošana**Mērķis**

Šai funkcijai jānodrošina eksistējošas lietotnes datu maiņa.

Ievaddati

Ievaddati ir labojamās lietotnes identifikators (ID) un labotie lietotnes dati. (Ievaddatu vērtības skatīt funkcijā APP-2)

Apstrāde

Pārbauda vai ievadītie dati ir korekti.

Izvaddati

Korektu datu ievades gadījumā, paziņojums S1

Kļūdu paziņojumi

Paziņojums E1, ja nav aizpildīti obligātie lauki.

E4, ja lietotne netika atrasta.

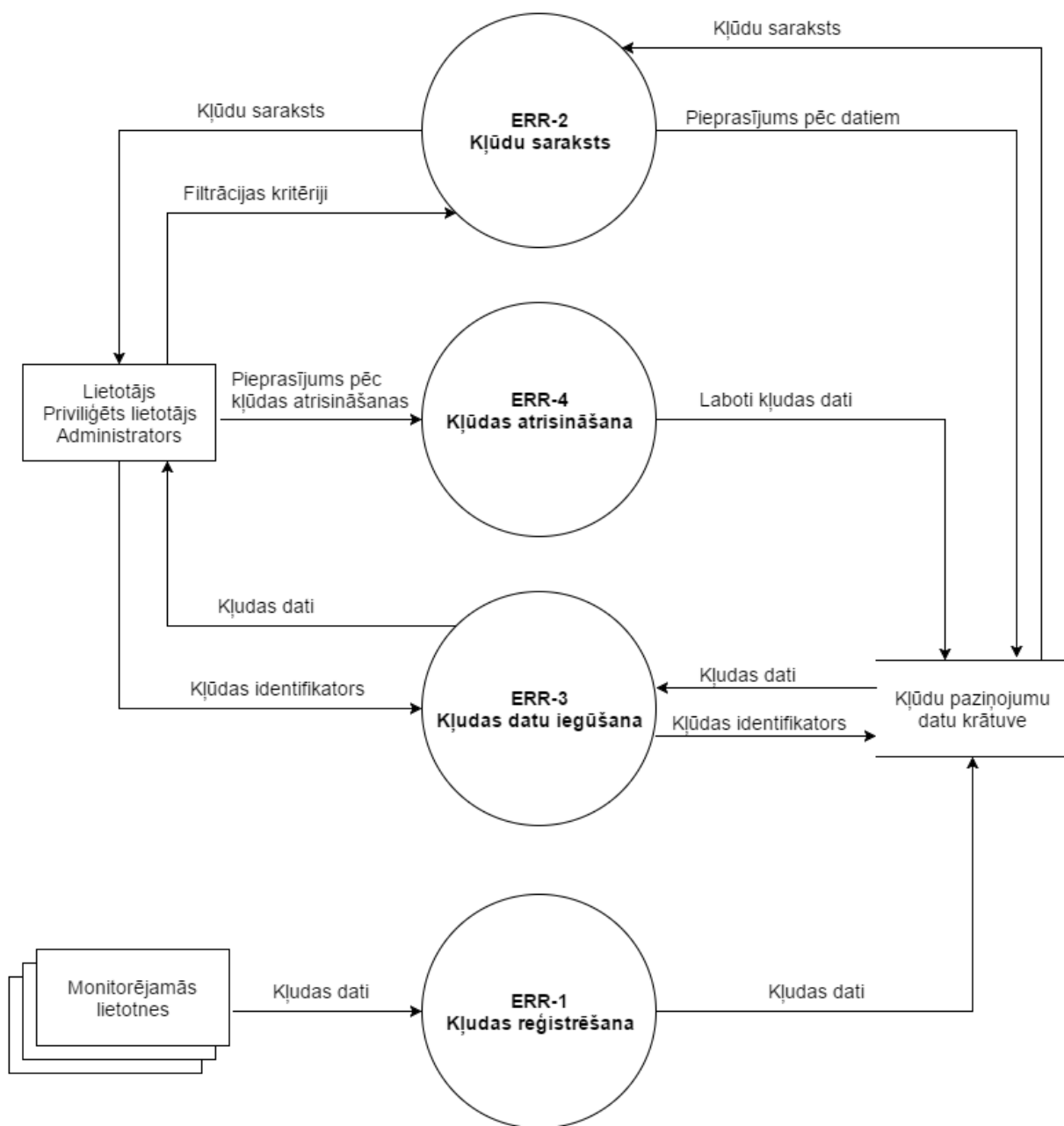
Citi paziņojumi atrunāti nodaļas sākumā.

2.4.3.5 APP-5 Lietotnes dzēšana

| |
|---|
| Mērķis |
| Šai funkcijai jānodrošina lietotnes dzēšana. |
| Ievaddati |
| Dzēšamās lietotnes identifikators (ID). |
| Apstrāde |
| Uzstāda norādītā lietotāja statusu – dzēsts. |
| Izvaddati |
| Dzēšanas izpildes gadījumā – paziņojums S1. |
| Kļūdu paziņojumi |
| E4, ja lietotne netika atrasta. Citi paziņojumi atrunāti nodaļas sākumā. |

2.4.4 Reģistrēto kļūdu modulis

| |
|--|
| Mērķis |
| Modulis atbild par monitorēto lietotņu iesūtīto kļūdas paziņojumu apstrādi un atrādīšanu. |
| Pieejamība |
| Jebkuram sistēmas lietotājam ir pieeja tikai to lietotņu kļūdu paziņojumiem, kurām tiem ir vismaz skatīšanas pieeja. (Skatīt pieejamību lietotņu modulī) |



2.6. att. Reģistrēto kļūdu moduļa otrā līmeņa datu plūsmu diagramma

2.4.4.1 ERR-1 Kļūdas reģistrēšana

| Mērķis | | | | | |
|---|------------------------------|----------------|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina lietotnes sūtīta ienākošā kļūdas paziņojuma apstrāde un reģistrācija. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Lietotnes API identifikators | Simbolu virkne | 32 simboli | Jā | |

| | | | | | |
|--|-------------------|----------------|----------------|----|--|
| 2. | Kļūdas paziņojums | Simbolu virkne | 1-255 simboli | Jā | |
| 3. | Kļūdas steks | Simbolu virkne | 1-4096 simboli | Nē | |
| 4. | Citi dati | JSON objekts | | Nē | |
| Apstrāde | | | | | |
| Identificē, vai kļūda iesūtīta jau iepriekš un, attiecīgi, sistēmā reģistrē vainu jaunu kļūdu, vai esošai kļūdai reģistrē jaunu kļūdas paziņojumu. | | | | | |
| Izvaddati | | | | | |
| Veiksmīgas reģistrēšanas gadījumā – paziņojums S1. | | | | | |
| Kļūdu paziņojumi | | | | | |
| E4, ja lietotne netika atrasta. Citi paziņojumi atrunāti nodaļas sākumā. | | | | | |

Laukā ‘citi dati’ iespējams saglabāt JSON objektu ar kļūdai zīmīgiem datiem. Šo datu saturs atkarīgs no to avota lietotnes konfigurācijas.

2.4.4.2 ERR-2 Kļūdu saraksts

| Mērķis | | | | | |
|--|-------------------|---|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina sistēmā reģistrēto lietotņu kļūdu paziņojumu atrādīšana. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Saistītā lietotne | Kāda no sistēmā reģistrētajām lietotnēm | 1 - n | Nē | |
| 2. | Lapa | Skaitlis | 1 - n | Nē | 1 |
| Apstrāde | | | | | |
| Atlasa pieprasītos lietotnes kļūdas paziņojumus pēc meklēšanas kritērijiem. | | | | | |
| Izvaddati | | | | | |
| Saraksts ar kļūdu paziņojumiem vai S2, ja neviens netika atrasts. | | | | | |
| Kļūdu paziņojumi | | | | | |
| Atrunāti nodaļas sākumā. | | | | | |

2.4.4.3 ERR-3 Kļūdas paziņojuma datu iegūšana

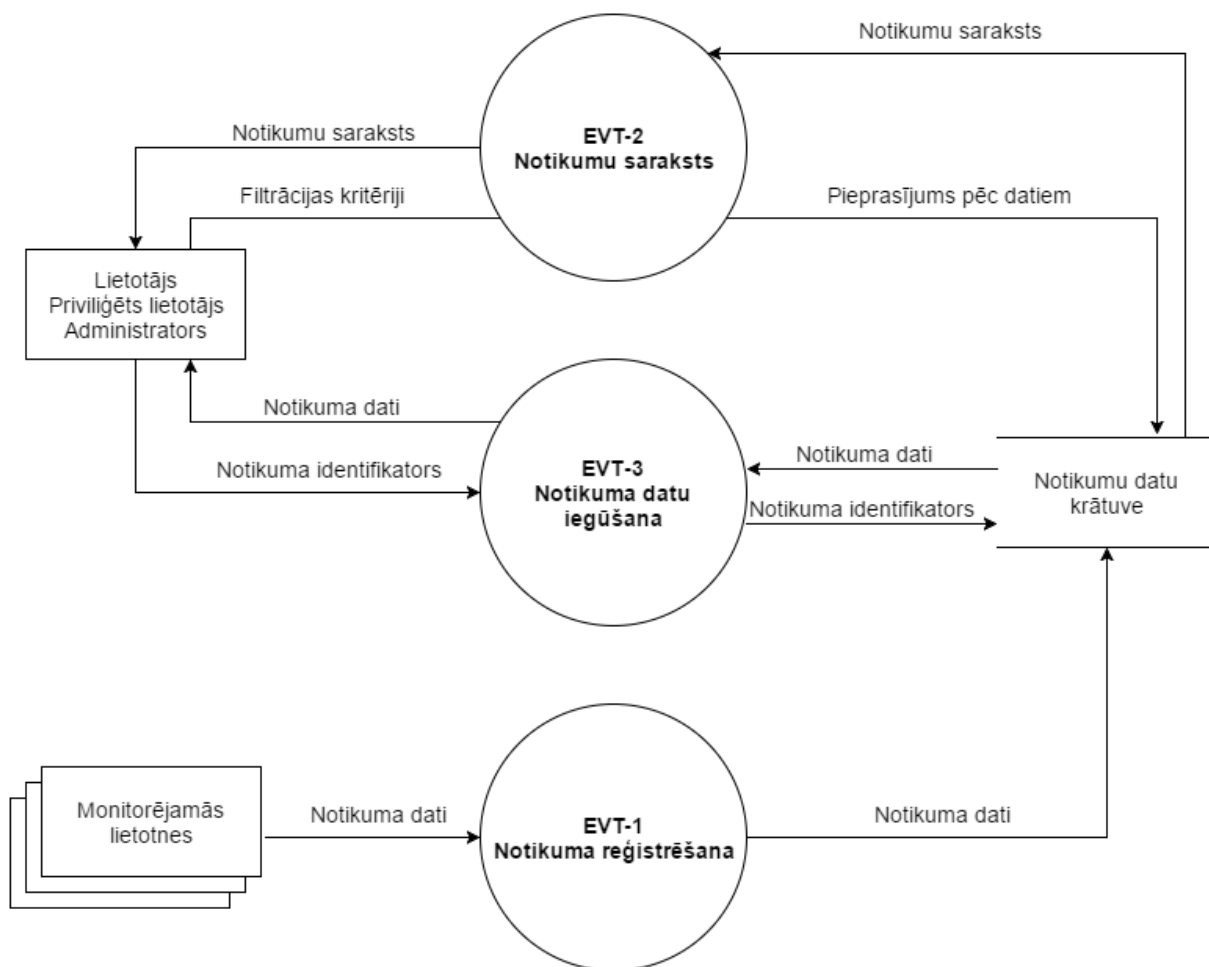
| |
|---|
| Mērķis |
| Šai funkcijai jānodrošina sistēmā reģistrēto lietotnes kļūdu paziņojumu atrādīšana. |
| Ievaddati |
| Kļūdas paziņojuma identifikators. |
| Apstrāde |
| Atlasa pieprasīto lietotnes kļūdas paziņojumu pēc identifikatora. |
| Izvaddati |
| Kļūdas paziņojuma informācija, kļūdas steks. |
| Kļūdu paziņojumi |
| E4, ja paziņojums netika atrasts. Citi paziņojumi atrunāti nodaļas sākumā. |

2.4.4.4 ERR-4 Kļūdas atrisināšana

| |
|---|
| Mērķis |
| Šai funkcijai jānodrošina sistēmā reģistrēto lietotnes kļūdu atrisināšanas statusa maiņa. |
| Ievaddati |
| Kļūdas paziņojuma identifikators. |
| Apstrāde |
| Atlasa pieprasīto lietotnes kļūdas paziņojumu pēc identifikatora un atzīmē kā atrisinātu. |
| Izvaddati |
| Veiksmīgas darbības rezultātā – paziņojums S1. |
| Kļūdu paziņojumi |
| E4, ja paziņojums netika atrasts. Citi paziņojumi atrunāti nodaļas sākumā. |

2.4.5 Reģistrēto notikumu modulis

| |
|--|
| Mērķis |
| Modulis atbild par monitorēto lietotņu iesūtīto notikumu apstrādi un atrādīšanu. |
| Pieejamība |
| Jebkuram sistēmas lietotājam ir pieeja tikai to lietotņu notikumiem, kurām tiem ir vismaz skatīšanas pieeja. (Skatīt pieejamību lietotņu modulī) |



2.7. att. Reģistrēto notikumu otrā līmeņa datu plūsmu diagramma

2.4.5.1 EVT-1 Notikuma reģistrēšana

| Mērķis | | | | | |
|--|------------------------------|----------------|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina lietotnes sūtīta ienākošā notikuma apstrāde un reģistrācija. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Lietotnes API identifikators | Simbolu virkne | 32 simboli | Jā | |
| 2. | Notikuma nosaukums | Simbolu virkne | 1-30 simboli | Jā | |
| 3. | Citi dati | JSON objekts | | Nē | |
| Apstrāde | | | | | |
| Reģistrē notikumu sistēmā. | | | | | |

| |
|---|
| Izvaddati |
| Veiksmīgas reģistrēšanas gadījumā – paziņojums S1. |
| Kļūdu paziņojumi |
| E4, ja lietotne netika atrasta. Citi paziņojumi atrunāti nodaļas sākumā. |

Laukā ‘citi dati’ iespējams saglabāt JSON objektu ar notikumam zīmīgiem datiem. Šo datu saturs atkarīgs no to avota lietotnes konfigurācijas.

2.4.5.2 EVT-2 Notikumu saraksts

| Mērķis | | | | | |
|--|-------------------|---|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina sistēmā reģistrēto lietotnes notikumu saraksta atrādīšana. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Saistītā lietotne | Kāda no sistēmā reģistrētajām lietotnēm | 1 - n | Nē | |
| 2. | Lapa | Skaitlis | 1 - n | Nē | 1 |
| Apstrāde | | | | | |
| Atlasa pieprasītos lietotnes notikumus pēc meklēšanas kritērijiem un sakārto pieprasītajā secībā pēc pieprasītās kolonnas. | | | | | |
| Izvaddati | | | | | |
| Saraksts ar notikumiem vai S2, ja neviens netika atrasts. | | | | | |
| Kļūdu paziņojumi | | | | | |
| Atrunāti nodaļas sākumā. | | | | | |

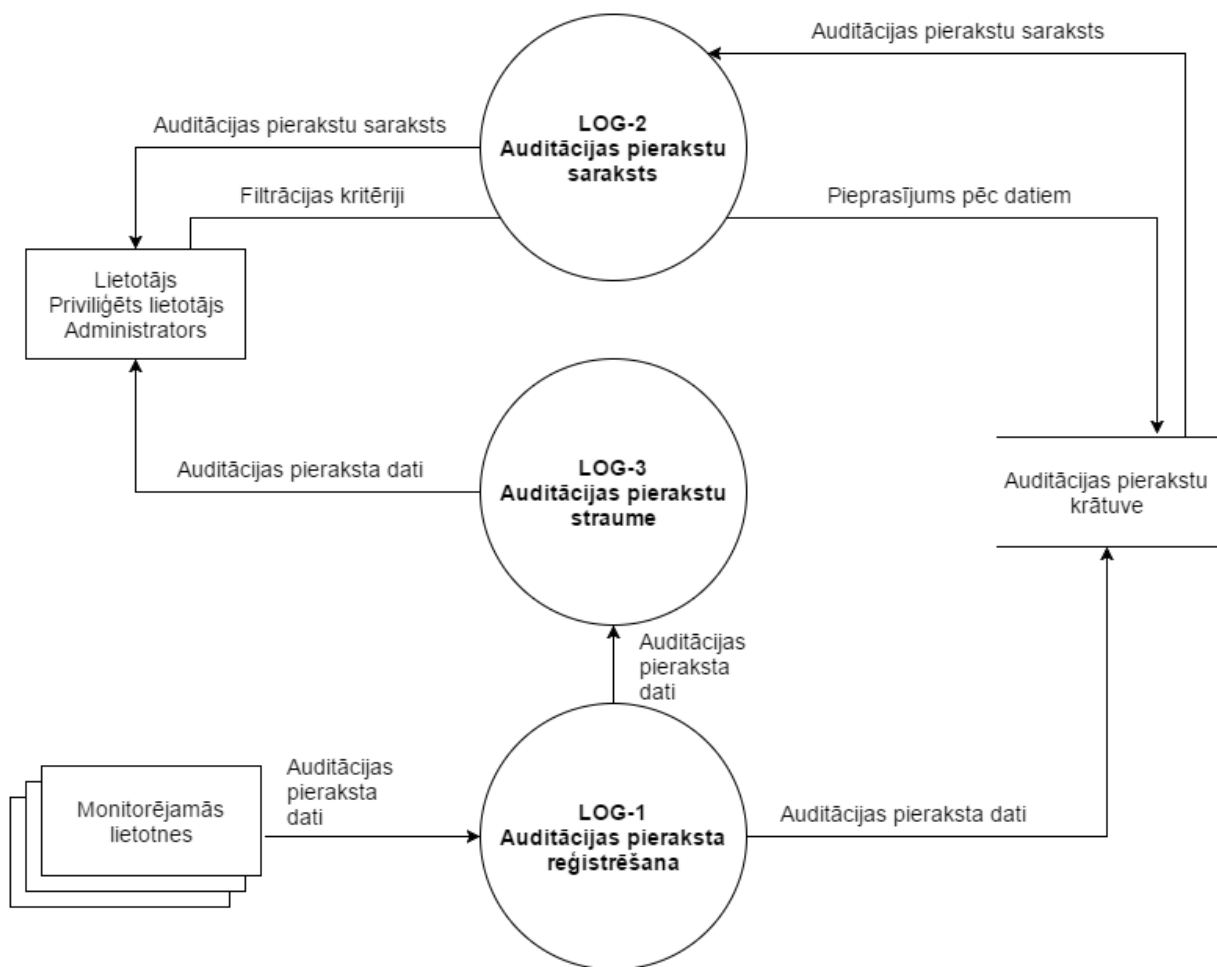
2.4.5.3 EVT-3 Notikuma datu iegūšana

| | |
|--|--|
| Mērķis | |
| Šai funkcijai jānodrošina sistēmā reģistrētā lietotnes notikuma datu atrādīšana. | |
| Ievaddati | |
| Notikuma identifikators. | |
| Apstrāde | |
| Atlasa pieprasīto lietotnes notikumu pēc identifikatora. | |

| |
|---|
| Izvaddati |
| Notikuma informācija. |
| Kļūdu paziņojumi |
| E4, ja notikums netika atrasts. Citi paziņojumi atrunāti nodaļas sākumā. |

2.4.6 Reģistrēto auditācijas pierakstu modulis

| |
|---|
| Mērķis |
| Modulis atbild par monitorēto lietotņu iesūtīto auditācijas pierakstu apstrādi un atrādīšanu. |
| Pieejamība |
| Jebkuram sistēmas lietotājam ir pieeja tikai to lietotņu auditācijas pierakstiem, kurām tiem ir vismaz skatīšanas pieeja. (Skatīt pieejamību lietotņu modulī) |



2.8. att. Reģistrēto auditācijas pierakstu moduļa otrā līmeņa datu plūsmu diagramma

2.4.6.1 LOG-1 Auditācijas pieraksta reģistrēšana

| Mērķis | | | | | |
|---|------------------------------|----------------|-----------------|-----------|------------------|
| Šai funkcijai jānodrošina lietotnes sūtīta ienākošā auditācijas pieraksta apstrāde un reģistrācija. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Lietotnes API identifikators | Simbolu virkne | 32 simboli | Jā | |
| 2. | Auditācijas pieraksta dati | JSON objekts | | Jā | |
| 3. | Auditācijas pieraksta laiks | Laika zīmogs | | Nē | Saņemšanas laiks |
| Apstrāde | | | | | |
| Reģistrē auditācijas pierakstu sistēmā. | | | | | |
| Izvaddati | | | | | |
| Veiksmīgas reģistrēšanas gadījumā – paziņojums S1. | | | | | |
| Kļūdu paziņojumi | | | | | |
| E4, ja lietotne netika atrasta. Citi paziņojumi atrunāti nodaļas sākumā. | | | | | |

2.4.6.2 LOG-2 Auditācijas pierakstu saraksts

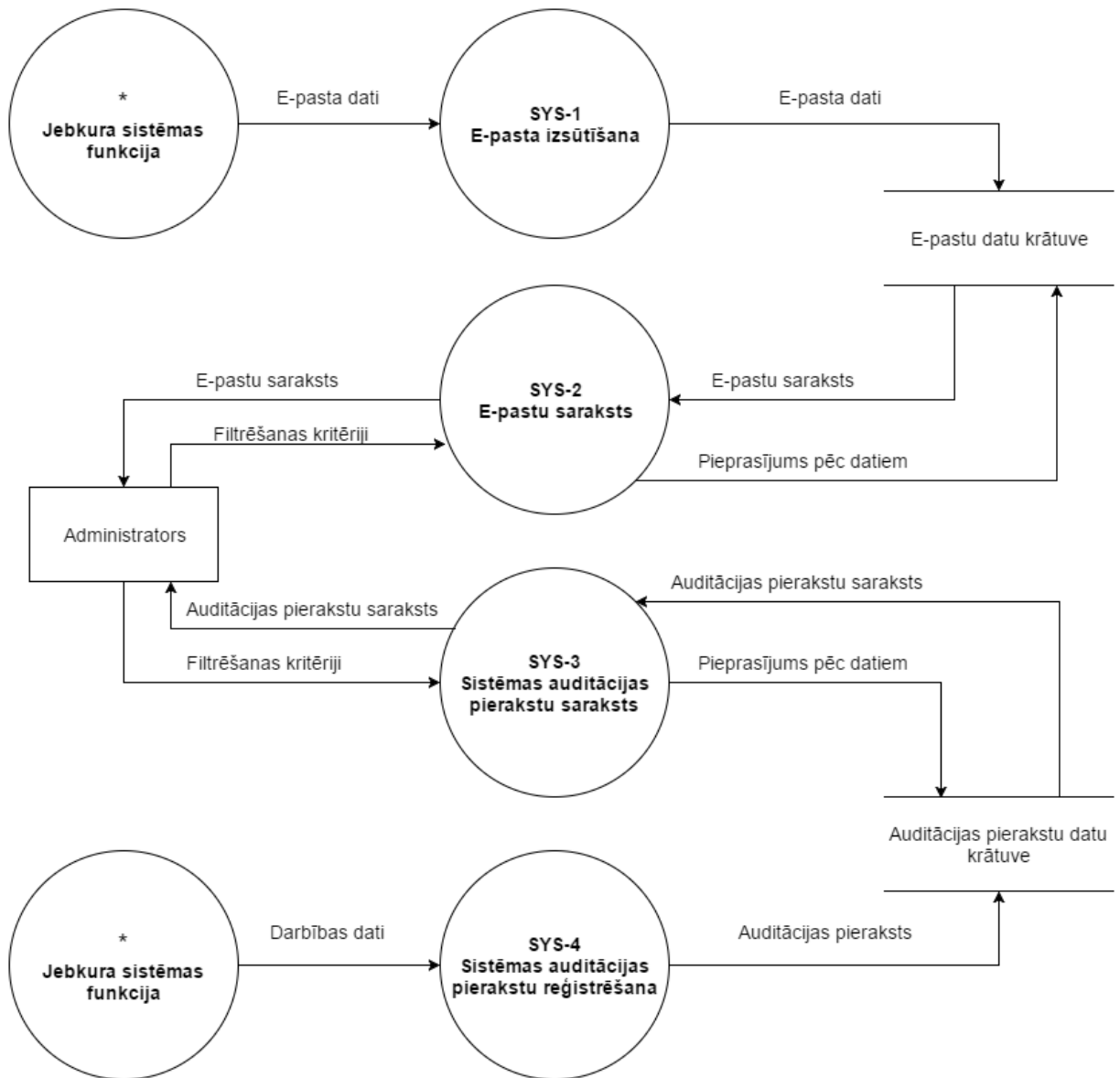
| Mērķis | | | | | |
|--|-------------------|---|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina sistēmā reģistrēto lietotnes notikumu saraksta atrādīšana. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Saistītā lietotne | Kāda no sistēmā reģistrētajām lietotnēm | 1 lietotne | Jā | |
| 2. | Lapa | Skaitlis | 1 - n | Nē | 1 |
| Apstrāde | | | | | |
| Atlasa pieprasītos lietotnes auditācijas pierakstus pēc meklēšanas kritērijiem. | | | | | |
| Izvaddati | | | | | |
| Saraksts ar auditācijas pierakstiem vai S2, ja neviens netika atrasts. | | | | | |
| Kļūdu paziņojumi | | | | | |
| E4, ja lietotne netika atrasta. Atrunāti nodaļas sākumā. | | | | | |

2.4.6.3 LOG-3 Auditācijas pierakstu straume

| |
|--|
| Mērķis |
| Šai funkcijai jānodrošina sistēmā ienākošo lietotnes auditācijas pierakstu atrādīšana reālā laikā. |
| Ievaddati |
| Lietotnes identifikators. |
| Apstrāde |
| Gaida ienākošas auditācijas pierakstus un sūta lietotājam reālā laikā. |
| Izvaddati |
| Auditācijas pierakstu dati. |
| Kļūdu paziņojumi |
| E4, ja lietotne netika atrasta. Citi paziņojumi atrunāti nodaļas sākumā. |

2.4.7 Sistēmas pārvaldības modulis

| |
|---|
| Mērķis |
| Modulis atbild par monitoringa risinājuma sistēmas pārvaldības funkcionalitāti priekš administratoriem. |
| Pieejamība |
| Sistēmas pārvaldības modulis pieejams tikai administratoriem. Funkcija SYS-1 ir sistēmas funkcija, kas ievadu saņem no sistēmas, bet izvadi var veikt jebkurai lietotājam. Funkcija SYS-4 arī ir sistēmas funkcija, kas ievadu saņem no citiem sistēmas moduļiem. |



2.9. att. Sistēmas pārvaldības moduļa otrā līmeņa datu plūsmu diagramma

2.4.7.1 SYS-1 E-pasta izsūtīšana

| |
|---|
| Mērķis |
| Šai funkcijai jānodrošina lietotāju apziņošana pa e-pastu, kļūdas paziņojuma ienākšanas gadījumā. |
| Ievaddati |
| Informācija par lietotnes kļūdas paziņojumu. |
| Apstrāde |
| Atlasa visus lietotnei saistītos lietotājus un izsūta paziņojuma e-pastus. |

| |
|-------------------------|
| Izvaddati |
| Paziņojuma e-pasti. |
| Kļūdu paziņojumi |
| E5 |

2.4.7.2 SYS-2 E-pastu saraksts

| Mērķis | | | | | |
|--|---------------------------------|---------------------------|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina sistēmas izsūtīto e-pastu saraksta atrādīšana. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Atslēgvārdi | Simbolu virkne | 1-255 simboli | Nē | |
| 2. | Kārtošanas virziens un atribūts | Atribūts, augoši/dilstoši | 1 atribūts | Nē | ID/ augoši |
| 3. | Lapa | Skaitlis | 1 - n | Nē | 1 |
| Apstrāde | | | | | |
| Atlasa sistēmas izsūtīto e-pastu datus pēc meklēšanas kritērijiem. | | | | | |
| Izvaddati | | | | | |
| Saraksts ar sistēmas izsūtītajiem e-pastiem vai S2, ja neviens netika atrasts. | | | | | |
| Kļūdu paziņojumi | | | | | |
| Atrunāti nodaļas sākumā. | | | | | |

2.4.7.3 SYS-3 Sistēmas auditācijas pierakstu saraksts

| Mērķis | | | | | |
|--|---------------------------------|---------------------------|-----------------|-----------|-------------|
| Šai funkcijai jānodrošina sistēmas auditācijas pierakstu atrādīšana. | | | | | |
| Ievaddati | | | | | |
| Nr. | Nosaukums | Vērtība | Garums/daudzums | Obligāts? | Noklusējums |
| 1. | Atslēgvārdi | Simbolu virkne | 1-255 simboli | Nē | |
| 2. | Kārtošanas virziens un atribūts | Atribūts, augoši/dilstoši | 1 atribūts | Nē | ID/ augoši |
| 3. | Lapa | Skaitlis | 1 - n | Nē | 1 |
| Apstrāde | | | | | |
| Atlasa sistēmas auditācijas pierakstus pēc meklēšanas kritērijiem. | | | | | |

| |
|---|
| Izvaddati |
| Saraksts ar sistēmas auditācijas pierakstiem vai S2, ja neviens netika atrasts. |
| Kļūdu paziņojumi |
| Atrunāti nodaļas sākumā. |

2.4.7.4 SYS-4 Sistēmas auditācijas pieraksta reģistrēšana

| |
|--|
| Mērķis |
| Šai funkcijai jānodrošina sistēmas auditācijas pierakstu reģistrēšana. |
| Ievaddati |
| Informācija par notikušo darbību sistēmā. |
| Apstrāde |
| Reģistrē auditācijas pierakstu. |
| Izvaddati |
| Nav. |
| Kļūdu paziņojumi |
| E5 |

3. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

Lietotņu monitoringa risinājums sastāv no divām daļām – servera puses risinājuma un tīmekļa lietotnes. Risinājumu izstrādē izmantota programmēšanas valoda *JavaScript*. Servera puses risinājums būvēts *NodeJS*⁴ vidē, datu glabāšanai izmantojot *PostgreSQL*⁵ datu bāzes vadības sistēma un *MongoDB*⁶ *NoSQL* datu bāze. Datu bāzes entītiju abstrakcijai lietota bibliotēka *SequelizeJS*.⁷ Tīmekļa lietotne būvēta izmantojot *AngularJS*⁸ ietvaru, par pamatu ņemot *Bootstrap Admin LTE*⁹ administrācijas paneļa sagatavi. Tīmekļa lietotnes būvējumu sagatavošanu nodrošina *Gulp*¹⁰ procesu pārvaldības rīks. Abos risinājumos pakotņu un bibliotēku uzturēšana organizēta izmantojot *NPM(Node Package Manager)*¹¹ rīku. Tīmekļa lietotne un monitorētās lietotnes ar serveri datu apmaiņu veic izmantojot *JSON REST API* saskarni.

3.1 Datu bāzes projektējums

3.1.1 Datu bāzes fiziskais modelis

Tabula “users” (skat. 3.1.att.) atbilst datu bāzes konceptuālā modeļa (skat. 2.1.att.) lietotāja entītijai, tabula “projects” atbilst lietotnes entītijai, tabula “events” atbilst notikuma entītijai, tabula “errors” atbilst kļūdas entītijai, tabula “error_events” atbilst kļūdas paziņojuma entītijai, tabula “emails” atbilst epasta entītijai, tabula “system_logs” atbilst sistēmas auditācijas pieraksta entītijai. Tabula “project_users” uztur saiti starp lietotājiem un lietotnēm.

⁴ NodeJS// <https://nodejs.org/en/>

⁵ PostgreSQL// <https://www.postgresql.org/>

⁶ MongoDB// <https://www.mongodb.com/>

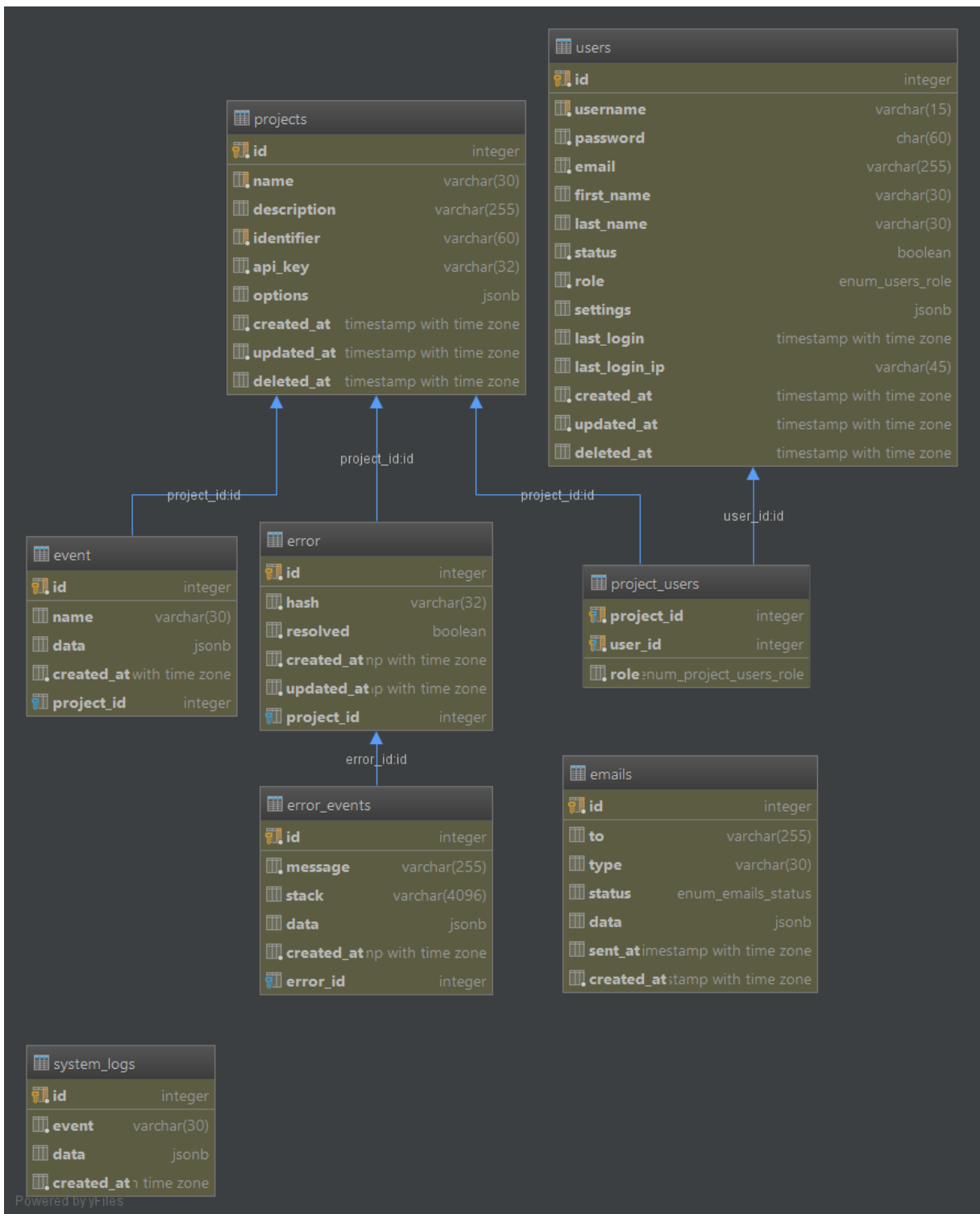
⁷ SequelizeJS// <http://docs.sequelizejs.com/en/latest/>

⁸ AngularJS// <https://angularjs.org/>

⁹ Admin LTE administration panel// <https://almsaeedstudio.com/AdminLTE>

¹⁰ Gulp// <http://gulpjs.com/>

¹¹ NPM// <https://www.npmjs.com/>



3.1. att. Datu bāzes fiziskais modelis

3.1.2 Tabulu projektējums

| Apzīmējums | Skaidrojums |
|------------------------|-------------------------------|
| PK | Primārā atslēga |
| FK(<tabula> <kolonna>) | Ārējā atslēga |
| UQ | Unikalitāte |
| ++ | Automātiski pieaugoša vērtība |

3.1.2.1 Tabula “users”

Tabula paredzēta sistēmas lietotāju datu glabāšanai.

| Nosaukums | Datu tips | Pazīmes | NULL? | Noklusējums | Apraksts |
|---------------|------------------------------------|---------|-------------|-------------|---|
| id | INTEGER | PK | NOT NULL | ++ | Unikāls identifikators. |
| first_name | VARCHAR(30) | | NULL | | Lietotāja vārds. |
| last_name | VARCHAR(30) | | NULL | | Lietotāja uzvārds. |
| username | VARCHAR(15) | UQ | NOT NULL | | Lietotāja lietotājvārds, kas tiek izmantots, lai pieteiktos sistēmā. |
| password | CHAR(15) | | NOT NULL | | Lietotāja parole šifrēta ar bcrypt algoritmu. Maksimālais garums atbilst bcrypt jaucejvērtības garumam. |
| email | VARCHAR(255) | | NOT NULL | | Lietotāja E-pasta adrese. Maksimālais garums atbilst RFC2821 ierobežojumiem. |
| status | BOOLEAN | | NOT NULL | TRUE | Lietotāja statuss sistēmā (aktīvs/neaktīvs). |
| role | ENUM('user', 'poweruser', 'admin') | | NOT NULL | user | Lietotāja loma sistēmā. |
| last_login | TIMESTAMP | | NULL | | Pēdējās pieteikšanās laiks. |
| last_login_ip | VARCHAR(45) | | NULL | | Pēdējās pieteikšanās IP adrese. |
| created_at | TIMESTAMP | | NOT NULL | | Izveidošanas datums un laiks. |

| | | | | | |
|------------|-----------|--|-------------|--|---------------------------------------|
| updated_at | TIMESTAMP | | NOT NULL | | Pēdējās atjaunošanas datums un laiks. |
| deleted_at | TIMESTAMP | | NULL | | Dzēšanas datums un laiks. |

3.1.2.2 Tabula “projects”

Tabula paredzēta sistēmā reģistrēto lietotņu datu glabāšanai.

| Nosaukums | Datu tips | Pazīmes | NULL? | Noklusējums | Apraksts |
|-------------|--------------|---------|-------------|-------------|--|
| id | INTEGER | PK | NOT NULL | ++ | Unikāls identifikators. |
| name | VARCHAR(30) | UQ | NOT NULL | | Lietotāja piešķirts projekta nosaukums. |
| description | VARCHAR(255) | | NULL | | Projekta apraksts. |
| identifier | VARCHAR(60) | UQ | NOT NULL | | Projekta API identifikators, kuru veido normalizēts (kebab-case) projekta nosaukums. |
| api_key | VARCHAR(32) | | NOT NULL | | Projekta API atslēga, md5 jaucējvērtības veidolā. |
| created_at | TIMESTAMP | | NOT NULL | | Izveidošanas datums un laiks. |
| updated_at | TIMESTAMP | | NOT NULL | | Pēdējās atjaunošanas datums un laiks. |
| deleted_at | TIMESTAMP | | NULL | | Dzēšanas datums un laiks. |

3.1.2.3 Tabula “events”

Tabula paredzēta sistēmā ienākošo un reģistrēto notikumu datu glabāšanai.

| Nosaukums | Datu tips | Pazīmes | NULL? | Noklusējums | Apraksts |
|------------|-------------|-----------------|-------------|-------------|-----------------------------------|
| id | INTEGER | PK | NOT NULL | ++ | Unikāls identifikators. |
| name | VARCHAR(30) | | NOT NULL | | Notikuma nosaukums. |
| data | JSONB | | NULL | | Notikuma dati. |
| project_id | INTEGER | FK(projects id) | NOT NULL | | Saistītā projekta identifikators. |

| | | | | | |
|------------|-----------|--|-------------|--|-------------------------------|
| created_at | TIMESTAMP | | NOT NULL | | Izveidošanas datums un laiks. |
|------------|-----------|--|-------------|--|-------------------------------|

3.1.2.4 Tabula “errors”

Tabula paredzēta sistēmā reģistrēto kļūdu glabāšanai.

| Nosaukums | Datu tips | Pazīmes | NULL? | Noklusējums | Apraksts |
|------------|-------------|-----------------|-------------|-------------|--|
| id | INTEGER | PK | NOT NULL | ++ | Unikāls identifikators. |
| hash | VARCHAR(32) | | NOT NULL | | Kļūdas paziņojuma šifrēta md5 jaucējvērtība. |
| resolved | BOOLEAN | | NOT NULL | FALSE | Notikuma dati. |
| project_id | INTEGER | FK(projects id) | NOT NULL | | Saistītā projekta identifikators. |
| created_at | TIMESTAMP | | NOT NULL | | Izveidošanas datums un laiks. |
| updated_at | TIMESTAMP | | NOT NULL | | Pēdējās atjaunošanas datums un laiks. |

3.1.2.5 Tabula “error_events”

Tabula paredzēta sistēmā ienākošo un reģistrēto kļūdas paziņojumu glabāšanai.

| Nosaukums | Datu tips | Pazīmes | NULL? | Noklusējums | Apraksts |
|------------|---------------|---------------|-------------|-------------|----------------------------------|
| id | INTEGER | PK | NOT NULL | ++ | Unikāls identifikators. |
| message | VARCHAR(255) | | NOT NULL | | Kļūdas paziņojums. |
| stack | VARCHAR(4096) | | NULL | | Kļūdas steks. |
| data | JSONB | | NULL | | Kļūdas dati. |
| error_id | INTEGER | FK(errors id) | NOT NULL | | Saistītās kļūdas identifikators. |
| created_at | TIMESTAMP | | NOT NULL | | Izveidošanas datums un laiks. |

3.1.2.6 Tabula “emails”

Tabula paredzēta sistēmas izsūtīto e-pastu reģistrēšanai.

| Nosaukums | Datu tips | Pazīmes | NULL? | Noklusējums | Apraksts |
|------------|-----------------------------------|---------|----------|-------------|------------------------------|
| id | INTEGER | PK | NOT NULL | ++ | Unikāls identifikators. |
| to | VARCHAR(255) | | NOT NULL | | Saņēmēja e-pasta adrese. |
| type | VARCHAR(30) | | NOT NULL | | E-pasts tips. |
| status | ENUM('pending', 'sent', 'failed') | | NOT NULL | pending | E-pasta nosūtīšanas statuss. |
| data | JSONB | | NULL | | E-pasta dati. |
| created_at | TIMESTAMP | | NOT NULL | | E-pasta izveidošanas laiks. |
| sent_at | TIMESTAMP | | NULL | | E-pasta nosūtīšanas laiks. |

3.1.2.7 Tabula "system_logs"

Tabula paredzēta sistēmas auditācijas pierakstu glabāšanai.

| Nosaukums | Datu tips | Pazīmes | NULL? | Noklusējums | Apraksts |
|------------|-------------|---------|----------|-------------|-------------------------------|
| id | INTEGER | PK | NOT NULL | ++ | Unikāls identifikators. |
| event | VARCHAR(30) | | NOT NULL | | Auditācijas pieraksta tips. |
| data | JSONB | | NULL | | Auditācijas pieraksta dati. |
| created_at | TIMESTAMP | | NOT NULL | | Izveidošanas datums un laiks. |

3.1.2.8 Starptabula "project_users"

Starptabula paredzēta sistēmā reģistrēto lietotņu un to lietotāju saišu glabāšanai.

| Nosaukums | Datu tips | Pazīmes | NULL? | Noklusējums | Apraksts |
|------------|-----------|-----------------------|----------|-------------|------------------------------------|
| project_id | INTEGER | PK FK(projects id) | NOT NULL | | Saistītā projekta identifikators. |
| user_id | INTEGER | PK FK(users id) | NOT NULL | | Saistītā lietotāja identifikators. |

| | | | | | |
|------|-------------------------|--|----------|--------|--------------------------|
| role | ENUM('member', 'admin') | | NOT NULL | member | Lietotāja loma projektā. |
|------|-------------------------|--|----------|--------|--------------------------|

3.1.2.9 Lietotņu auditācijas pierakstu krātuve

Lietotņu iesūtītos auditācijas ierakstus paredzēts glabāt bez shēmas datu bāzē, jeb NoSQL. Katras lietotnes auditācijas pieraksti glabāsies savā ierakstu kolekcijā JSON formātā. Aptuvenā JSON objekta shēma vienam ierakstam:

| Auditācijas pieraksta JSON formāta piemērs |
|--|
| <pre>{ 'timestamp': 'xxxxx', 'type': 'user:login', 'data': {...} }</pre> |

3.2 Funkciju dekompozīcija

3.2.1 Autentifikācijas moduļa dekompozīcija

| Kontrolieris "AuthController": |
|--|
| login() – Veic lietotāja autentifikāciju sistēmā. (<i>AUTH-1</i>) |
| user() – Atgriez pašlaik autentificētā lietotāja datus |
| Klase "AuthProvider": |
| middleware() – Nodrošina katra lietotāja pieprasījuma autentificēšanu izmantojot JSON Web Token principu (<i>AUTH-1</i>) |
| appMiddleware – Nodrošina katra lietotnes pieprasījuma autentificēšanu izmantojot lietotnes API atslēgu (<i>AUTH-2</i>) |

3.2.2 Lietotāju pārvaldības moduļa dekompozīcija

| Kontrolieris "UsersController": |
|--|
| list() – Nodrošina lietotāju saraksta izvadi (<i>USR-1</i>) |
| get() – Nodrošina lietotāja datu iegūšanu (<i>USR-3</i>) |
| new() – Nodrošina jauna lietotāja izveidošanu (<i>USR-2</i>) |

| |
|--|
| update() – Nodrošina lietotāja labošanu (<i>USR-4</i>) |
| delete() – Nodrošina lietotāja dzēšanu (<i>USR-5</i>) |

3.2.3 Lietotņu pārvaldības moduļa dekompozīcija

| |
|---|
| Kontrolieris “ProjectsController”: |
| list() – Nodrošina lietotņu saraksta izvadi (<i>APP-1</i>) |
| get() – Nodrošina lietotnes datu iegūšanu (<i>APP-3</i>) |
| new() – Nodrošina jaunas lietotnes izveidošanu (<i>APP-2</i>) |
| update() – Nodrošina lietotnes datu labošanu (<i>APP-4</i>) |
| delete() – Nodrošina lietotnes dzēšanu (<i>APP-5</i>) |

3.2.4 Reģistrēto kļūdu moduļa dekompozīcija

| |
|--|
| Kontrolieris “ErrorsController”: |
| error() – Nodrošina kļūdas paziņojuma reģistrāciju (<i>ERR-1</i>) |
| list() – Nodrošina kļūdu paziņojumu saraksta izvadi (<i>ERR-2</i>) |
| get() – Nodrošina kļūdu paziņojuma datu izvadi (<i>ERR-3</i>) |
| solve() – Nodrošina kļūdas atzīmēšanu kā atrisināte (<i>ERR-4</i>) |

3.2.5 Reģistrēto notikumu moduļa dekompozīcija

| |
|--|
| Kontrolieris “EventsController”: |
| register() – Nodrošina notikuma reģistrāciju (<i>EVT -1</i>) |
| list() – Nodrošina notikumu saraksta izvadi (<i>EVT-2</i>) |
| get() – Nodrošina notikuma datu izvadi (<i>EVT -3</i>) |

3.2.6 Reģistrēto auditācijas pierakstu moduļa dekompozīcija

| |
|---|
| Kontrolieris “LogsController”: |
| register() – Nodrošina auditācijas pieraksta reģistrāciju (<i>LOG -1</i>) |
| list() – Nodrošina auditācijas pierakstu saraksta izvadi (<i>LOG-2</i>) |
| get() – Nodrošina auditācijas pieraksta datu izvadi (<i>LOG -3</i>) |

3.2.7 Sistēmas pārvaldības moduļa dekompozīcija

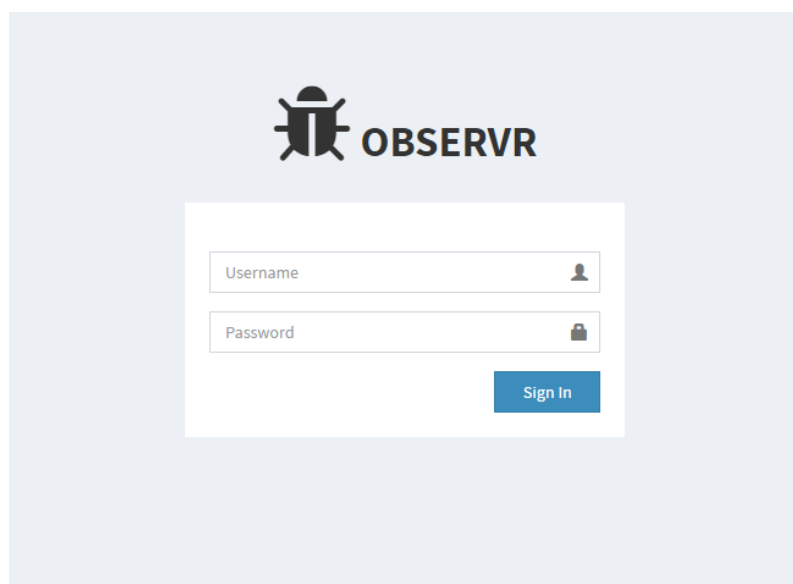
| |
|---|
| Kontrolieris “ServiceController”: |
| logs() – Nodrošina sistēmas auditācijas pierakstu saraksta izvadi (SYS - 3) |
| emails() – Nodrošina izsūtīto e-pastu saraksta izvadi (SYS-2) |
| Klase “Logger” |
| log() – Reģistrē sistēmas auditācijas pierakstu (SYS-4) |
| Klase “Mailer” |
| sendMail() – Izsūta e-pastu (SYS-1) |

3.3 Saskarnes projektējums

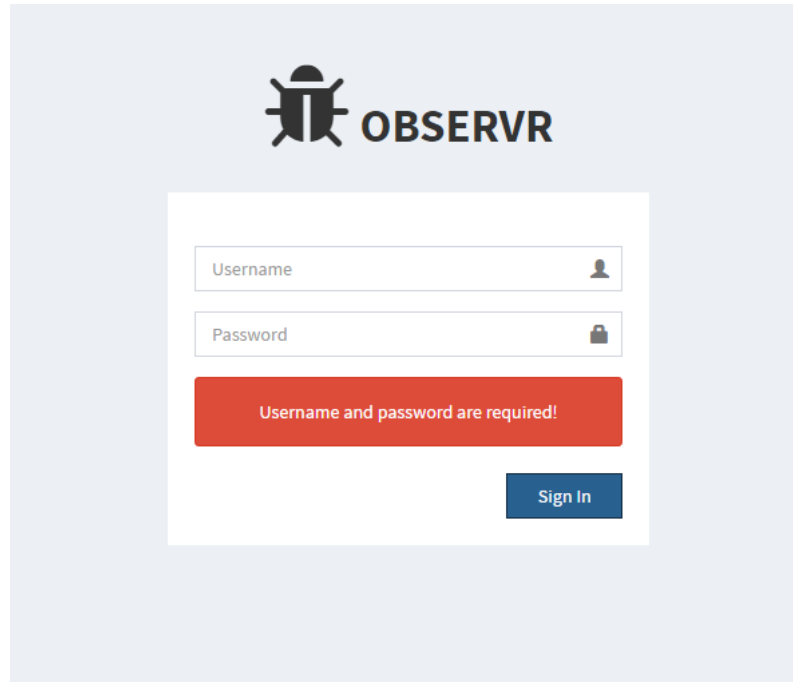
Šajā nodaļā attēloti sistēmas ekrānformu un saskarnes elementu piemēri:

- Pieteikšanās forma
- Datu saraksts
- Datu ievades/ labošanas forma
- Sistēmas paziņojums

3.3.1 Pieteikšanās forma



3.2. att. Lietotāja pieteikšanās ekrānforma



3.3. att. Lietotāja pieteikšanās ekrānforma ar paziņojumu

3.3.2 Datu tabula

Search + New user

| Name | E-mail | Role | |
|-----------------|--------------------------|-------|----------------------------------|
| Dāvis Mednis | davis.mednis@esynergy.lv | admin | <input type="button" value="✎"/> |
| Testa Lietotājs | test@test.lv | user | <input type="button" value="✎"/> |

Previous 1 Next

3.4. att. Datu izvades saraksts

3.3.3 Datu labošanas/pievienošanas forma

New user

First name
Testa

Last name
Lietotājs

Username
test

Email
test@test.lv

Role
User

Status

Password

Password 2x

✓ Passwords match

3.5. att. Datu pievienošanas/ labošanas forma

New user

First name
First name
! First name is required!

Last name
Last name
! Last name is required!

Username
Last name
! Username is required!

Email
E-mail
! E-mail is required!

Role
Role
! Role is required!

Status

Password
Password

Password 2x
Repeat password

3.6. att. Nekorekti aizpildīta datu pievienošanas/ labošanas forma ar paziņojumiem

3.3.4 Sistēmas paziņojumi



3.7. att. Sistēmas paziņojums

4. TESTĒŠANAS DOKUMENTĀCIJA

Izstrādes gaitā tika veikti risinājuma funkciju vienībtesti. Testēšana notika manuāli, vadoties pēc uzstādītajām funkcionālajām prasībām.

Tika izstrādāti arī automatizēti integrācijas testi, izmantojot *mocha* Javascript testēšanas ietvaru.

4.1 Vienībtestēšanas testpiemēri

4.1.1 Autentifikācijas moduļa testpiemēri

| Nr. | Darbība | Sagaidāmais rezultāts |
|-----|---|--|
| 1. | Pieslēgšanās sistēmai ar pareiziem pieslēgšanās datiem. | Atveras sistēmas sākuma ekrāns. |
| 2. | Pieslēgšanās sistēmai ar nepareiziem pieslēgšanās datiem. | Izlec paziņojums par nepareiziem datiem. |
| 3. | Mēģinājums piekļūt sadaļai, kurai lietotājam nav tiesību piekļūt. | Lietotājs tiek novirzīts uz sākumlapu. |

4.1.2 Lietotāju pārvaldības moduļa testpiemēri

| Nr. | Darbība | Sagaidāmais rezultāts |
|-----|--|---|
| 4. | Tiek veidots vai saglabāts lietotājs ar korekti aizpildītiem visiem formas laukiem. | Izlec paziņojums par veiksmīgu saglabāšanu un dati nonāk datu bāzē. |
| 5. | Tiek veidots vai saglabāts lietotājs ar nekorekti aizpildītiem formas laukiem. Norādīts nepareizs e-pasts – nekorektā formātā bez @ zīmes. | Lauks iekrāsojas sarkans un parādās kļūdas paziņojums. Dati netiek saglabāti. |
| 6. | Tiek veidots vai saglabāts lietotājs ar nekorekti aizpildītiem formas laukiem. Nav aizpildīts kāds no obligātajiem laukiem. | Lauks iekrāsojas sarkans un parādās kļūdas paziņojums. Dati netiek saglabāti. |

| | | |
|----|--------------------------------|---|
| 7. | Tiek veikta lietotāja dzēšana. | Sistēma pārjautā, vai tiešām vēlaties dzēst. Ja atbild ja, lietotājs tiek dzēsts, ja nē, lietotājs paliek formas skatā. |
|----|--------------------------------|---|

4.1.3 Lietotņu pārvaldības moduļa testpiemēri

| Nr. | Darbība | Sagaidāmais rezultāts |
|-----|---|--|
| 8. | Tiek veidota vai saglabāta lietotne ar korekti aizpildītiem visiem formas laukiem. | Izlec paziņojums par veiksmīgu saglabāšanu un dati nonāk datu bāzē. |
| 9. | Tiek veidots vai saglabāts lietotājs ar nekorekti aizpildītiem formas laukiem. Nav aizpildīts kāds no obligātajiem laukiem. | Lauks iekrāsojas sarkans un parādās kļūdas paziņojums. Dati netiek saglabāti. |
| 10. | Tiek veikta lietotnes dzēšana. | Sistēma pārjautā, vai tiešām vēlaties dzēst. Ja atbild ja, lietotne tiek dzēsta, ja nē, lietotājs paliek formas skatā. |

4.1.4 Reģistrēto kļūdu moduļa testpiemēri

| Nr. | Darbība | Sagaidāmais rezultāts |
|-----|---|--|
| 11. | Tiek simulēts no lietotnes atsūtītas kļūdas pieteikums. | Pieteiktā kļūda nonāk datu bāzē un parādās kļūdu sarakstā. |
| 12. | Tiek veikta kļūdas atzīmēšana kā atrisināta. | Mainās kļūdas statuss uz apstiprinātu. |

4.1.5 Reģistrēto notikumu moduļa testpiemēri

| Nr. | Darbība | Sagaidāmais rezultāts |
|-----|--|--|
| 13. | Tiek simulēts no lietotnes atsūtīta notikuma pieteikums. | Pieteiktais notikums nonāk datu bāzē un parādās notikumu sarakstā. |

4.1.6 Reģistrēto auditācijas pierakstu moduļa testpiemēri

| Nr. | Darbība | Sagaidāmais rezultāts |
|-----|---|---|
| 14. | Tiek simulēts no lietotnes atsūtīta auditācijas pieraksta pieteikums. | Pieteiktais auditācijas pieraksts nonāk datu bāzē, tas parādās auditācijas pierakstu sarakstā un auditācijas pierakstu straumē. |

4.1.7 Sistēmas pārvaldības moduļa testpiemēri

| Nr. | Darbība | Sagaidāmais rezultāts |
|-----|---|--|
| 15. | Tiek simulēts no lietotnes atsūtītas kļūdas pieteikums. | Visiem lietotnei saistītajiem lietotājiem tiek nosūtīts e-pasts par notikušo kļūdu. E-pasts parādās izsūtīto e-pastu sarakstā. |

4.2 Automatizētās testēšanas scenāriji

Automatizētā integrācijas testēšana tika veikta ar nolūku pārlicināties, vai izstrādes gaitā nav tikusi salauzta kāda no risinājuma funkcionalitātes vienībām. Automātiskie testi tika izpildīti izstrādes gaitā.

Tika izveidotas divu veidu automātisko testu kopas. Pirmais tests izpildīja saraksta atlasīšanas funkciju visiem datu bāzes modeļiem, tādā veidā pārlicinoties, ka modeļa definīcija sakrīt ar datu bāzes tabulu un kolonnu definīcijām. Ja ir nesakrītības, tests neizpildās.

Otrais tests veica izsaukumus uz visiem REST API ieejas punktiem, pārbaudot, vai tie neatbild ar kļūdainu statusa paziņojumu, piemēram, 500, kas nozīmē, ka programmā ieviesusies kļūda.

4.3 Testēšanas žurnāls

| TP Nr. | Datums/ Rezultāts | Piezīmes | Datums/ Rezultāts | Piezīmes |
|--------|-------------------|---|-------------------|----------|
| 1. | 21.05.2016 / NOK | Funkcionalitāte nestrādā, lietotājs var piekļūt jebkurai lapai. | 27.05.2016 / OK | |
| 2. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 3. | 21.05.2016 / OK | | 27.05.2016 / OK | |

| | | | | |
|-----|------------------|--|-----------------|--|
| 4. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 5. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 6. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 7. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 8. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 9. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 10. | 21.05.2016 / NOK | Neparādījās vaicājamo paziņojums, lai turpinātu darbu, nācās pārladēt lapu. | 27.05.2016 / OK | |
| 11. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 12. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 13. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 14. | 21.05.2016 / OK | | 27.05.2016 / OK | |
| 15. | 21.05.2016 / NOK | E-pasts netika nosūtīts | 27.05.2016 / OK | |

5. PROJEKTA PĀRVALDĪBA

5.1 Projekta organizācija

Kvalifikācijas darba izstrādes procesā tika pielietots spirāles izstrādes modelis¹². Darba izstrāde tika veikta iterācijās, katrā iterācijā precizējot prasības un vēlamā gala rezultāta vīziju. Šāds izstrādes dzīves cikla modelis tika pielietots, jo izstrādes sākumā nebija skaidras visas pasūtītāja prasības, bet bija zināma aptuvenā gala produkta vīzija.

Izstrādes gaitā tika pielietota prototipēšanas metode, kad izstrādātājs sagatavo daļēji funkcionālu produkta vai tā moduļa prototipu un pirms tālākas funkcionalitātes piešķiršanas prezentē to pasūtītājam un precizē prasības.

Autors izstrādes procesā pielietoja “gumijpīlošanas” (*rubber ducking*)¹³ problēmu risināšanas metodi, lai atvieglotu programmatūras koda izstrādi brīžos, kad procesā bija radušās problēmas. Autors secināja, ka šī metode tiešām strādā un atvieglo programmatūras izstrādes procesu.

Lai labāk organizētu izstrādi, tika lietoti arī *scrum* metodes elementi, konkrētāk, ikdienas apspriedes starp autoru un izstrādes vadītāju.

5.2 Konfigurācijas pārvaldība

Lietotņu monitoringa risinājuma izstrādes gaitā tika lietota versiju kontroles sistēma *Git*. Izstrādājot risinājumu, pēc katra funkcionalitātes vienuma izstrādes un pat biežāk, tika veikta programmatūras koda reģistrēšana versiju kontrolē, un progress tika augšupielādēts tiešsaistes privātā koda krātuvē *GitHub*. Lai atvieglotu *Git* lietošanu, papildus tā lietošanai no komandrindas, tika lietots arī rīks *SourceTree* priekš *Windows*. *SourceTree* nodrošina pārskatāmu saskarni operācijām ar versiju kontroles sistēmu. Autors centās darba izstrādes gaitā izmantot nozares profesionāļu ieteikto *Git* zarošanas modeli¹⁴, taču ņemot vērā, ka

¹² Pressman R.S. Software Engineering. A Practioner's Approach. The McGraw-Hill Comp., Inc., 2000.

¹³ Hayes D. The Psychology Underlying the Power of Rubber Duck Debugging// <https://pressupinc.com/blog/2014/06/psychology-underlying-power-rubber-duck-debugging/>

¹⁴ Driessen V. A successful Git branching model// <http://nvie.com/posts/a-successful-git-branching-model/>

projekts tika izstrādāts vienatnē un tā apmēri ir salīdzinoši mazi, šī modeļa lietošana šajā gadījumā ievērojamas priekšrocības nedeva.

Datu bāzes tabulu un kolonnu definīcijas tika uzturētas izmantojot migrāciju principu, ko piedāvā lietotā *SequelizeJS* datu bāzes abstrakcijas bibliotēka.

Lai organizētu izstrādes progresu un kļūdu novēršanu, darba uzdevumi un kļūdas tika reģistrētas *GitHub Issues* sadaļā.

Lai atvieglotu būvējumu piegādes testa vidē, tika izmantots nepārtrauktās integrācijas rīks *Jenkins*¹⁵. *Jenkins* veica automatizētas piegādes testa vidē pēc katras koda augšupielādes tiešsaistes koda krātuvē.

5.3 Kvalitātes nodrošināšana

Izstrādājot programmas kodu tika ievērotas nozares profesionāļu un entuziastu vidū pieņemtās labās prakses *AngularJS* un *NodeJS* lietotņu izstrādē. Tas sevī ietver pārskatāmas un loģiskas failu struktūras lietošanu, viendabīga un lasāma koda rakstīšanu un pārskatāmu koda konstrukciju lietošanu. Funkcijas un klases tika komentētas izmantojot *JSDoc*¹⁶ standarta komentārus. Lai uzlabotu koda kvalitāti tika lietots koda statiskās analīzes rīks *JSLint*¹⁷ un izstrādes vidē *PHPStorm IDE* iebūvētie palīgrīki.

Programmatūras prasību specifikācijas (2. sadaļa), projektējuma apraksta(3. sadaļa) un testēšanas dokumentācijas(4. sadaļa) izstrādē tika ņemti vērā sekojošo LVS standartu ieteikumi:

- LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis;
- LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai;
- LVS 70:1996 Programmatūras testēšanas dokumentācija.

¹⁵ Jenkins// <https://jenkins.io/>

¹⁶ JSDoc// <http://usejsdoc.org/>

¹⁷ JSLint// <https://www.npmjs.com/package/gulp-jshint>

6. DARBIETILPĪBAS NOVĒRTĒŠANA

Darbietilpības novērtēšana tika veikta konsultējoties ar pieredzējušiem nozares pārstāvjiem un lietojot autora līdzšinējo pieredzi strādājot ar līdzīgām tehnoloģijām un prasībām. Novērtēšana tika veikta, sadalot veicamo darbu sīkākās loģiskās daļas – moduļos. Vērtēšanā tika pielietota metode, prognozējot optimistisko, reālistisko, un pesimistisko izstrādes darbietilpības laiku. Galējā vērtība iegūta pēc formulas $([\text{optimistiskais}] + 4 * [\text{reālistiskais}] + [\text{pesimistiskais}]) / 6$.

Visi lielumi norādīti stundās.

| Pozīcija | Optimistiskais | Reālistiskais | Pesimistiskais | Prognozētais |
|-----------------------------|----------------|---------------|----------------|--------------|
| Modulis AUTH | 32 | 40 | 45 | 39,5 |
| Modulis USR | 30 | 32 | 40 | 33 |
| Modulis APP | 44 | 48 | 54 | 48.3 |
| Modulis ERR | 44 | 48 | 54 | 48.3 |
| Modulis EVT | 35 | 40 | 45 | 40 |
| Modulis LOG | 30 | 32 | 40 | 33 |
| Modulis SYS | 35 | 40 | 45 | 40 |
| Izstrādes vides uzstādīšana | 20 | 24 | 32 | 24 |
| Komunikācija ar pasūtītāju | 10 | 16 | 20 | 15,6 |
| Dokumentācijas izstrāde | 148 | 160 | 170 | 158,3 |
| KOPĀ: | | | | 480 |

Kopējā darbietilpība tik prognozēta 480h apmērā, jeb 60 cilvēkdienas, jeb 3 cilvēkmēneši.

NOBEIGUMS

Kvalifikācijas darba ietvaros izstrādātais programmatūras produkts atbilst sākotnēji nospraustajiem mērķiem un prasībām. Lietotņu monitoringa risinājums tiks ieviests lietošanā autora prakses uzņēmumā, lai atvieglotu izstrādātājiem programmatūras izstrādi un uzturēšanu.

Izstrādātajam produktam ir liels attīstības potenciāls un to nākotnē plānots uzlabot un papildināt tā funkcionalitāti, kā arī publicēt publiskā koda krātuvē pakalpojumā *GitHub*.

Autors kvalifikācijas darba izstrādes laikā apguva un padziļināja zināšanas darba izstrādē izmantoto tehnoloģiju lietošanā, kā arī apguva pielietotās projekta organizācijas metodes.

IZMANTOTIE AVOTI

1. LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis”
2. LVS 72:1996, “Ieteicamā programmatūras projektējuma apraksts”
3. LVS 70:1996, „Programmatūras testēšanas dokumentācija”
4. Driessen V. A successful Git branching model// <http://nvie.com/posts/a-successful-git-branching-model/> (skatīts 15.03.2016)
5. Hayes D. The Psychology Underlying the Power of Rubber Duck Debugging// <https://pressupinc.com/blog/2014/06/psychology-underlying-power-rubber-duck-debugging/> (skatīts 24.05.2016)
6. Pressman R.S. Software Engineering. A Practioner's Approach. The McGraw-Hill Comp., Inc., 2000. pp. 915.
7. Admin LTE administration panel// <https://almsaeedstudio.com/AdminLTE> (skatīts 21.04.2016)
8. AngularJS// <https://angularjs.org/> (skatīts 21.04.2016)
9. Gulp// <http://gulpjs.com/> (skatīts 21.04.2016)
10. Gravatar// <https://en.gravatar.com/> (skatīts 21.04.2016)
11. Jenkins// <https://jenkins.io/> (skatīts 21.04.2016)
12. JSDoc// <http://usejsdoc.org/> (skatīts 21.04.2016)
13. JSLint// <https://www.npmjs.com/package/gulp-jshint> (skatīts 21.04.2016)
14. MongoDB// <https://www.mongodb.com/> (skatīts 21.04.2016)
15. NodeJS// <https://nodejs.org/en/> (skatīts 21.04.2016)
16. NPM <https://www.npmjs.com/> (skatīts 21.04.2016)
17. OWASP// <https://www.owasp.org/> (skatīts 19.03.2016)
18. PostgreSQL// <https://www.postgresql.org/> (skatīts 21.04.2016)
19. SequelizeJS// <http://docs.sequelizejs.com/en/latest/> (skatīts 21.04.2016)
20. Software as a Service// <http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service> (skatīts 15.03.2016)

Programmas ieejas punkta *app.js* kods

```
var express      = require('express');
var http         = require('http');
var bodyParser   = require("body-parser");
var cookieParser = require("cookie-parser");
var clc          = require('cli-color');
var winston      = require('winston');
var morgan       = require('morgan');

winston.info(clc.bgGreen(clc.black("STARTING APP")));

var app = express();
var server = http.Server(app);

// Read application configuration
app.config = require('./config/config.js');
app.environment = process.env.NODE_ENV || app.config.env || 'production';
app.debug = app.config.debug || app.environment == 'development';
var port = process.env.PORT || app.config.port || 4100;

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
app.use(cookieParser());

// Bootstrap DB models
app.db = require('./models/index.js');

// Bootstrap application services
app.services = {};

var Queue = require('./services/queue.js');
app.services.queue = new Queue();
var Mailer = require('./services/mailer.js');
app.services.mailer = new Mailer(app);

// Bootstrap controllers
var RoutesLoader = require('./controllers/index.js');
var loader = RoutesLoader(app);
app.use(loader.router);

// Expose application port
server.listen(port, function () {
  winston.info(clc.bgGreen(clc.black("SERVER LISTENING ON " + port)));
});

module.exports = app;
```

Klases "ProjectsController" kods

```

var _ = require('lodash');
var md5 = require('md5');

/**
 *
 * ProjectsController. Responsible for project REST API endpoints.
 *
 * @param _app
 * @returns {ProjectsController}
 * @constructor
 */
function ProjectsController(_app) {
  this.app = _app;
  this.db = this.app.db;
  this.name = 'projects';
  this.exposed = '*';
  this.public = [];

  return this;
}

/**
 *
 * Returns list of projects.
 *
 * @param params
 * @param done
 * @param req
 * @returns Promise
 */
ProjectsController.prototype.list = function (params, done, req) {
  var that = this;
  var query = this.db.project.makeGenericQuery(params, {attributes:
{exclude: ['apiKey']}});

  if (!(req.user.role == 'admin' && params.admin)) {
    query.include = [
      {
        model: that.db.user,
        as: 'members',
        where: {id: req.user.id},
        attributes: ['id']
      }
    ];
  }

  return this.db.project.findAndCountAll(query)
    .then(function (result) {
      done(result);
    });
};

/**
 *
 * Returns a single project instance.

```

```

*
* @param params
* @param done
* @returns Promise
*/
ProjectsController.prototype.get = function (params, done) {
  var that = this;
  return this.db.project.findOne({
    where: {id: params.id},
    include: [
      {
        model: that.db.user,
        as: 'members',
        attributes: ['id', 'firstName', 'lastName']
      }
    ]
  }).then(function (project) {
    project = project.get();
    var members = {};
    project.members.forEach(function (member) {
      members[member.id] = {
        id: member.id,
        role: member.projectUsers.role,
        name: member.firstName + ' ' + member.lastName
      };
    });
    project.members = members;
    done(project);
  });
};

/**
 *
 * Creates a new project instance
 *
 * @param params
 * @param done
 * @returns Promise
 */
ProjectsController.prototype.new = function (params, done) {
  var project = params;
  project.apiKey = md5(Date.UTC + project.name);

  project.identifier = _.kebabCase(project.name);
  return this.db.project.create(params)
    .then(function (project) {
      project = project.get();
      done(project);
    });
};

/**
 *
 * Updates an existing project instance.
 *
 * @param params
 * @param done
 * @returns Promise
 */
ProjectsController.prototype.update = function (params, done) {

```

```

var that = this;
delete params.identifier;
delete params.apiKey;
var project;
return that.db.sequelize.transaction({
  deferrable: that.db.Sequelize.Deferrable.SET_DEFERRED,
  logging: console.log
}, function (t) {
  return that.db.project.findOne({where: {id: params.id},
transaction: t})
    .then(function (_project) {
      project = _project;
      project.set(params);
      return project.save({transaction: t})
    }).then(function (_project) {
      project = _project;
      var userIds = [];
      _.forOwn(params.members, function (val, key) {
        userIds.push(val.id)
      });
      return that.db.user.findAll({where: {id: {$in: userIds}}});
    }).then(function (users) {
      users.forEach(function (user) {
        user.projectUsers = {role:
params.members[user.id].role};
      });
      return project.setMembers(users, {transaction: t});
    }).then(function () {
      project = project.get();
      project.members = params.members;
      done(project);
    })
  });
});

/**
 *
 * Marks a project instance as deleted.
 *
 * @param params
 * @param done
 * @returns Promise
 */
ProjectsController.prototype.delete = function (params, done) {
  return this.db.project.destroy({where: {id: params.id}})
    .then(function (project) {
      done(project);
    })
};

module.exports = ProjectsController;

```

Lietotāju saraksta skata kods

```

<div class="col-md-12">
  <div class="box box-primary spinner" ng-controller="GridController as
ul" alias="ul" module="users">
    <div class="box-header">
      <div class="col-md-3">
        <div class="input-group">
          <input type="text" class="form-control"
placeholder="Search">
          <span class="input-group-btn">
            <button type="button" class="btn btn-default btn-
flat">
              <i class="fa fa-search"></i>
            </button>
          </span>
        </div>
      </div>
      <div class="col-md-3">
        </div>
      <div class="col-md-3">
        </div>
      <div class="col-md-3">
        <button ui-sref="app.users.new" type="button" class="btn
btn-success btn-flat pull-right">
          <i class="fa fa-plus"></i> New user
        </button>
      </div>
    </div>
    <!-- /.box-header -->
    <div class="box-body table-responsive">
      <table class="table table-striped table-bordered officer-grid">
        <tr>
          <th col-header sortable="true" text="Name"
key="firstName" click="ul.sortBy(key, sortable)" sort="ul.sort">
          <th col-header sortable="true" text="E-mail"
key="email" click="ul.sortBy(key, sortable)" sort="ul.sort">
          <th col-header sortable="true" text="Role" key="role"
click="ul.sortBy(key, sortable)" sort="ul.sort">
          <th></th>
        </tr>
        <tr ng-repeat="row in ul.rows">
          <td ng-bind="::(row.firstName + ' ' + row.lastName)"
class="top-pad"></td>
          <td ng-bind="::row.email" class="top-pad"></td>
          <td ng-bind="::row.role" class="top-pad"></td>
          <td>
            <button ui-sref="app.users.edit({id: row.id})"
type="button" class="btn btn-primary btn-sm btn-flat pull-right">
              <i class="fa fa-pencil"></i>
            </button>
          </td>
        </tr>
      </table>

```

```
        <div class="text-center">
            <uib-pagination total-items="ul.total" ng-model="ul.page"
next-text="Next" previous-text="Previous" items-per-page="ul.per_page" max-
size="7" ng-change="ul.loadData()"></uib-pagination>
        </div>
    </div>
    <!-- /.box-body -->
    <div class="box-footer">

        </div>
    </div>
    <!-- /.box -->
</div>
<!-- /.col -->
```

Automatizētā testa kods

```

process.env.NODE_ENV = 'test';

var assert = require("assert");
var request = require('supertest');
var app = require("../app.js");

describe('Data models', function () {
  Object.keys(app.db).forEach(function (model) {
    {
      if (model !== 'sequelize' && model !== 'Sequelize') {
        it(model + ' works', function (done) {
          app.db[model].findAll({limit: 1}).then(function () {
            done();
          }, function (err) {
            done(err);
          });
        });
      }
    }
  });
});

describe('API', function () {
  for (var r = 0; r < app.routes.length; r++) {
    var route = app.routes[r];
    it('API endpoint ' + route.url + ' works', function (done) {
      request(app)[route.method](route.url)
        .expect(function (res) {
          var statuses = [200, 400, 401];
          if (statuses.indexOf(res.statusCode) > -1) {
            return true;
          } else {
            throw new Error('Expected ' + statuses + ' got ' +
res.statusCode);
          }
        })
        .end(done)
    })
  }

  for (var r = 0; r < app.routes.length; r++) {
    var route = app.routes[r];
    it('API endpoint auth ' + route.url + ' works', function (done) {
      if (!route.isPublic) {
        request(app)[route.method](route.url)
          .expect(function (res) {
            var statuses = [401];
            return true;
            if (statuses.indexOf(res.statusCode) > -1) {
              return true;
            } else {
              throw new Error('Expected ' + statuses + ' got
' + res.statusCode);
            }
          })
      }
    })
  }
}

```

```
        .end(done)
    } else {
        request(app)[route.method](route.url)
            .expect(function(res) {
                var statuses = [200, 400];
                return true;
                if (statuses.indexOf(res.statusCode) > -1) {
                    return true;
                } else {
                    throw new Error('Expected ' + statuses + ' got
' + res.statusCode);
                }
            })
            .end(done)
    }
})
}
});
```

3.

Kvalifikācijas darbs „*Lietotņu monitoringa risinājums*” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Dāvis Mednis* _____ .05.2016.

Rekomendēju darbu aizstāvēšanai

Darba vadītāja: **Dr. sc. comp. Vineta Arnicāne** _____ .05.2016.

Recenzents: *M.dat. Jurijs Šļamovs*

Darbs iesniegts 30.05.2016.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2016. prot. Nr. _____

Komisijas sekretārs(-e): _____