

LATVIJAS UNIVERSITĀTES

DATORIKAS FAKULTĀTE

Programmatūras izstrādes pārvaldības sistēma

KVALIFIKĀCIJAS DARBS

Autors:

Edvards Zakovskis

Studenta apliecības numurs:

Ez18038

Darba vadītāja:

Doc. Elīna Kalniņa

RĪGA 2021

ANOTĀCIJA

Kvalifikācijas darba mērķis ir izstrādāt programmatūras izstrādes pārvaldības sistēmu, kurā varētu ērti pārvaldīt programmatūras izstrādi, redzot aktīvos uzdevumus, un uzskaitīt kļūdas programmatūrā un uzskaitīt to labošanu. Sistēmas tīmekļa API serviss tika izstrādāts izmantojot ASP.NET CORE 5 arhitektūru, kur tika izmantota C# valoda, un sistēmas tīmekļa lietotne tika izstrādāta izmantojot Angular 11 ietvaru, kā arī papildus programmatūras izstrādātājiem tika izveidots “Visual Studio Code” paplašinājums, kurā programmatūras izstrādāji var ērti mainīt uzdevumu, vai kļūdu statusu.

Atslēgvārdi: ASP.NET CORE 5, tīmekļa vietne, C#, Angular, pārvaldības sistēma

ABSTRACT

SOFTWARE DEVELOPEMENT MANAGEMENT SYSTEM

The purpose of the qualification work is to develop a software development management system that could help manage software development conveniently by seeing active tasks and listing software bugs and their fixes. The system's modern API service was developed using the ASP.NET CORE 5 architecture, which used the C # language, and the applications of this system were developed using the angular 11 framework, furthermore, “Visual studio code” extension was developed for developers to easily change tasks or bug status.

Keywords: ASP.NET CORE 5, Website, C #, Angular, Management System

Satura rādītājs

ANOTĀCIJA.....	1
ABSTRACT	2
APZĪMĒJUMU SARAKSTS.....	6
1. IEVADS	7
1.1. Nolūks.....	7
1.2. Darbības sfēra	7
1.3. Saistība ar citiem dokumentiem.....	7
1.4. Pārskats.....	7
2. VISPĀRĒJAIS APRAKSTS	9
2.1. Esošā stāvokļa apraksts.....	9
2.2. Pasūtītājs	9
2.3. Produkta perspektīva.....	9
2.4. Darījumprasības	9
2.5. Sistēmas lietotāji	10
2.6. Vispārējie ierobežojumi	10
2.7. Pieņēmumi un atkarības.....	11
3. PROGRAMMU PRASĪBU SPECIFIKĀCIJA	12
3.1. Funkcionālās prasības	12
3.1.1. Lietotāju modulis.....	16
3.1.2. Organizāciju modulis.....	20
3.1.3. Projektu modulis.....	22
3.1.4. Biļešu modulis.....	25
3.1.5. Komentāru modulis	32
3.1.6. Paplašinājuma modulis.....	34
4.2. Nefunkcionālās prasības	36

4.2.1.	Lietojamība.....	36
4.2.2.	Uzturamība	36
4.2.3.	Pielāgojamība	36
4.2.4.	Savienojamība	36
4.2.5.	Pieejamība	36
4.2.6.	Drošība	36
4.	PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....	37
4.1.	Datu bāzes projektējums.....	37
4.1.1.	Konceptuālais ER modelis	37
4.1.2.	Loģiskais modelis.....	38
4.1.3.	Datu bāzes fiziskais modelis	40
4.1.4.	Datu bāzes tabulu apraksti.....	41
4.2.	Funkciju projektējums	47
4.2.3.	Secību diagramma biļetes izveidošanas funkcijai.....	48
4.3.	Lietotāja saskarne	49
4.3.1.	Lietotāja saskarnes ekrānuzņēmumi.....	49
5.	TESTĒŠANAS DOKUMENTĀCIJA.....	58
5.1.	Testēšanas metodika	58
5.2.	Testēšanas plāns.....	58
5.3.	Moduļu testēšana	58
4.2.1	Lietotāju moduļa testēšana	58
5.3.2.	Organizāciju moduļa testēšanas rezultāti	60
5.3.3.	Projektu moduļa testēšanas rezultāti.....	61
5.3.4.	Biļešu moduļa testēšanas rezultāti	62
5.3.5.	Komentāru moduļa testēšanas rezultāti	65
5.3.6.	Paplašinājuma modulis.....	66

7. PROJEKTA ORGANIZĀCIJA.....	68
8. KONFIGURĀCIJU PĀRVALDĪBA.	69
2. DARBIETILPĪBAS NOVERTĒJUMS.....	70
3. KVALITĀTES NODROŠINĀŠANA.....	71
SECINĀJUMI	72
IZMANTOTĀS LITERATŪRAS SARAKSTS	73
PIELIKUMI.....	74

APZĪMĒJUMU SARAKSTS

Nr.	Apzīmējums	Paskaidrojums
1.	API	Lietojumprogrammas saskarne
2.	ASP.NET CORE	Atvērtā pirmkoda ietvars dažādām operētājsistēmas platformām[1]
3.	Angular	Atvērtā pirmkoda tīmekļa lietotņu ietvars, kura izstrādi uztur Google[2]
4.	Visual Studio Code	Bezmaksas koda redaktors, kurus izstrādāja “Microsoft”, kurš pieejams uz Windows, Linux un MacOS.
5.	ASP.NET CORE Identity	API, kas atbalsta lietotāja pierakstīšanās funkcionalitāti.[3]
6.	JWT žetons	“JSON tīmekļa žetons” ir veids kā droši sazināties lietotnei ar serveri.[4]
7.	IE	Pārlūkprogramma “Internet Explorer”
8.	IOS	Mobilā operētājsistēma, kas veidota un izplatīta vienīgi “Apple” ierīcēm.
9.	Android	Operētājsistēma viedtālruniem, planšetdatoriem un citām ierīcēm.

1. IEVADS

1.1. Nolūks

Šī dokumenta mērķis ir aprakstīt veidoto programmēšanas izstrādēs pārvaldības sistēmu, specificējot programmatūras prasības un plānoto projektējumu.

1.2. Darbības sfēra

Programmatūras izstrādes pārvaldības sistēma ir domāta izstrādes uzdevumu, un kļūdu reģistrēšanai un uzskatei, tādējādi palīdzot pārvaldīt programmatūras izstrādi. Sistēma ir paredzēta, gan izstrādātājiem, gan arī projekta pārvaldītājiem.

1.3. Saistība ar citiem dokumentiem

Dokumentā ir ievērotas LVS 68:1996 „Programmatūras prasību specifikācijas (PPS) ceļvedis”[6] noformēšanas prasības, savukārt PPA ir izstrādāts pēc LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai”[7] prasībām. Dokumenta noformējums ir izstrādāts pēc Latvijas Universitātes rīkojumu Nr. 1/38 “Prasības noslēguma darbu izstrādāšanai un aizstāvēšanai Latvijas Universitātē” [10] .

1.4. Pārskats

Šis dokuments sastāv no:

1. Ievada, kurš sevī ietver nolūku, darbības sfēru, saistību ar citiem dokumentiem un pārskatu
2. Vispārējā apraksta, kas ietver esošā stāvokļa aprakstu, pasūtītāju, produkta perspektīvu, darījumprasības, sistēmas lietotāju aprakstu, vispārējos ierobežojumus un pieņēmumus un atkarības.
3. Programmatūras prasību specifikācijas, kas ietver funkcionālās prasības un saistītos moduļu un to funkciju aprakstus, kā arī nefunkcionālās prasības.
4. Programmatūras projektējuma apraksta, kas sastāv no datu bāzes projektējumiem, funkciju projektējumiem un lietotāja saskarņu apraksta.
5. Testēšanas dokumentācijas, kas ietver gan testēšanas metodiku, testēšanas plānu un moduļu testēšanas aprakstus, kas ietver arī testēšanas rezultātus.
6. Projekta organizācijas izklāsta.
7. Konfigurācijas pārvaldības apraksta.
8. Darbietilpības novērtējuma.

9. Kvalitātes nodrošināšanas izklāsta
10. Secinājumiem
11. Izmantotās literatūras saraksta
12. Pielikumiem ar koda daļām

2. VISPĀRĒJAIS APRAKSTS

2.1. Esošā stāvokļa apraksts.

Pašlaik ir pārvaldības sistēmas, kuras ir domātas programmēšanas pārvaldībai, bet tām tiek veidota viena saskarne, gan izstrādātājiem, gan projekta pārvaldītājiem, kas salīdzinoši nav ērti izstrādātājiem, jo tie pamatā strādā kādā izstrādes vidē. Tādēļ izveidotās sistēmas mērķis ir pārskatāmi, ērti veikt uzdevumu un kļūdu uzskaiti, kā arī nodrošinot papildus saskarni izmantojot “Visual Studio Code” koda redaktora paplašinājumu iespējas.

2.2. Pasūtītājs

Sistēma izstrādāta pēc studenta iniciatīvas kvalifikācijas darba ietvaros. Lai izstrādātu programmatūras izstrādes pārvaldības sistēmu, kas ir reāli, nepieciešama, tiek ņemtas vērā jau esošu sistēmu kā, piemēram, “Jira”[13], “Azure DevOps Server”[14] u.t.t.

2.3. Produkta perspektīva

Produkts ir patstāvīga sistēma, kur iespējams reģistrēties organizācijām vai komandām, un kurām ir iespējams pievienot programmatūras izstrādes projektus un katram projektam varēs pievienot izstrādes uzdevumus un kļūdas, kā arī veikt to uzskaiti. Katrai organizācijai būs iespējams pievienot lietotājus, kuri varēs piekļūt attiecīgās organizācijas informācijai. Programmatūra būs izmantojama interneta pārlūkprogrammās, kā arī projekta programmatūras izstrādātājiem tiks izveidots paplašinājums, no kura tie varēs ērti izmainīt uzdevumu vai kļūdas statusu.

2.4. Darījumasprasības

Lietotnei ir šādas pamatfunkcijas:

1. Lietotāju reģistrācija, autentificēšanās;
2. Lietotāju informācijas glabāšana un rediģēšana;
3. Organizācijas izveide;
4. Organizācijas projektu pievienošana;
5. Jauna lietotājstāsta pievienošana, rediģēšana.
6. Uzdevumu un Kļūdu informācijas pievienošana rediģēšana.
7. Iespēja apskatīties lietotāja nepabeigtos uzdevumus vai kļūdas
8. Iespēja pievienot komentārus kļūdai vai uzdevumam.

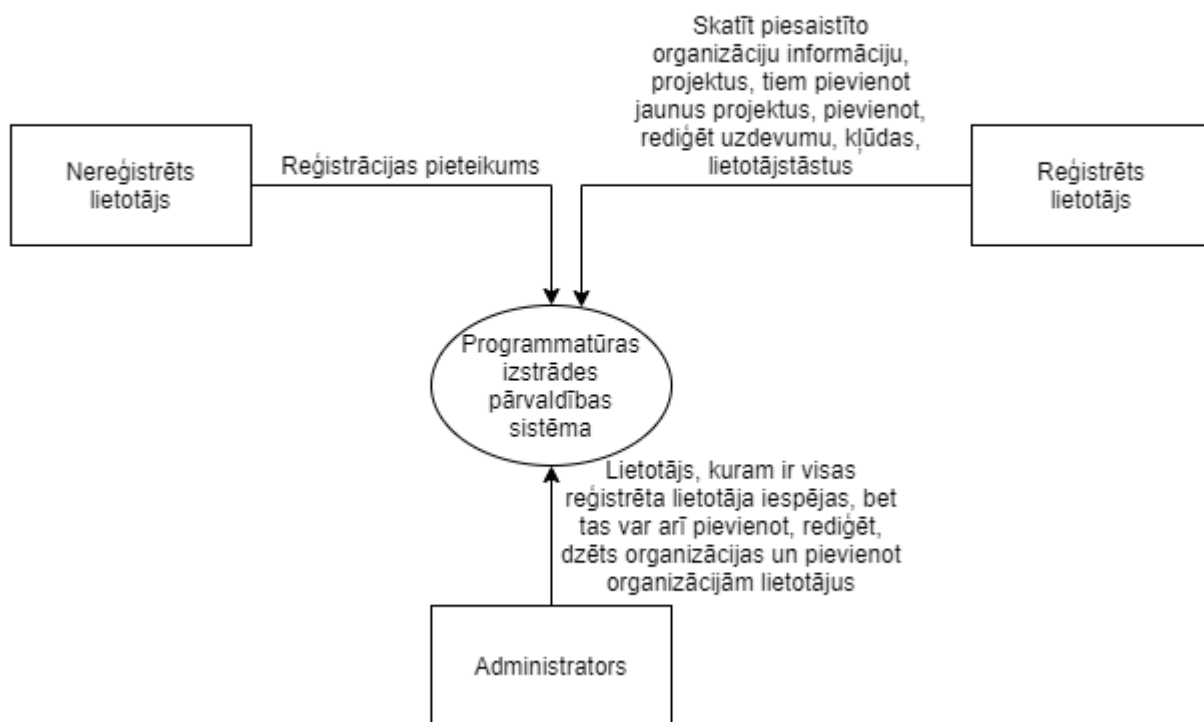
2.5. Sistēmas lietotāji

Sistēmai ir noteiktas lietotāju grupas.

-**Neregistrēti lietotāji** -reģistrēties sistēmai.

-**Reģistrētie lietotāji** – iespēja skatīt organizāciju un tai piesaistīt citus lietotājus, kā arī skatīt un pievienot projektus piesaistītajām organizācijām, kā arī šiem projektiem pievienot, rediģēt, lietotājstāstus, uzdevumus, kļūdas, kā arī pievienot komentārus gan uzdevumiem, gan kļūdām.

-**Administratori** – papildus, iespēja izveidot, rediģēt dzēst, organizācijas, kā arī tām pievienot lietotājus.



2.1 att. Datu plūsmu diagramma 0. līmenis

2.6. Vispārējie ierobežojumi

Darbam ar sistēmu nepieciešama tīmekļa piekļuve un tīmekļa pārlūkprogramma.

Tiks atbalstītas šādas pārlūkprogrammu versijas.

Pārlūkprogramma	Atbalstītās versijas
Google Chrome	Jaunākā(87.0.4280.88)

Mozilla Firefox	Jaunākā(84.01) un plašāku atbalsta versija
Microsoft Edge	Jaunākā(87.0.664.66)
IE	11
Safari	Jaunākā(Safari 14)
IOS	IOS 12, IOS 13
Android	Q (10.0), Pie (9.0), Oreo (8.0), Nougat (7.0)

2.7. Pieņēmumi un atkarības

Tiek pieņemts, ka lietotājam ir prasmes darbam ar tīmekļa pārlūkprogrammu un arī tas, ka ierīcēm ir piekļuve interneta pieslēgumam un pārlūkprogrammai. Sistēma ir paredzēta tiem izstrādātājiem, kuri lieto “Visual Studio Code” koda redaktoru, kuram tiks izveidots paplašinājums.

3. PROGRAMMU PRASĪBU SPECIFIKĀCIJA

Programmatūras prasību specififikācijas nolūks ir attēlot, specificēt sistēmas funkcijas un nefunkcionālās prasības. Programmatūras funkcionālās prasības attēlotas datu plūsmu diagrammu formā.

Programmatūras prasību specififikācija veidota, izmantojot LVS 68:1996 “Programmatūras prasību specififikācijas ceļvedis”.

Šajā nodaļā ir aprakstītas funkcionālās un nefunkcionālās prasības.

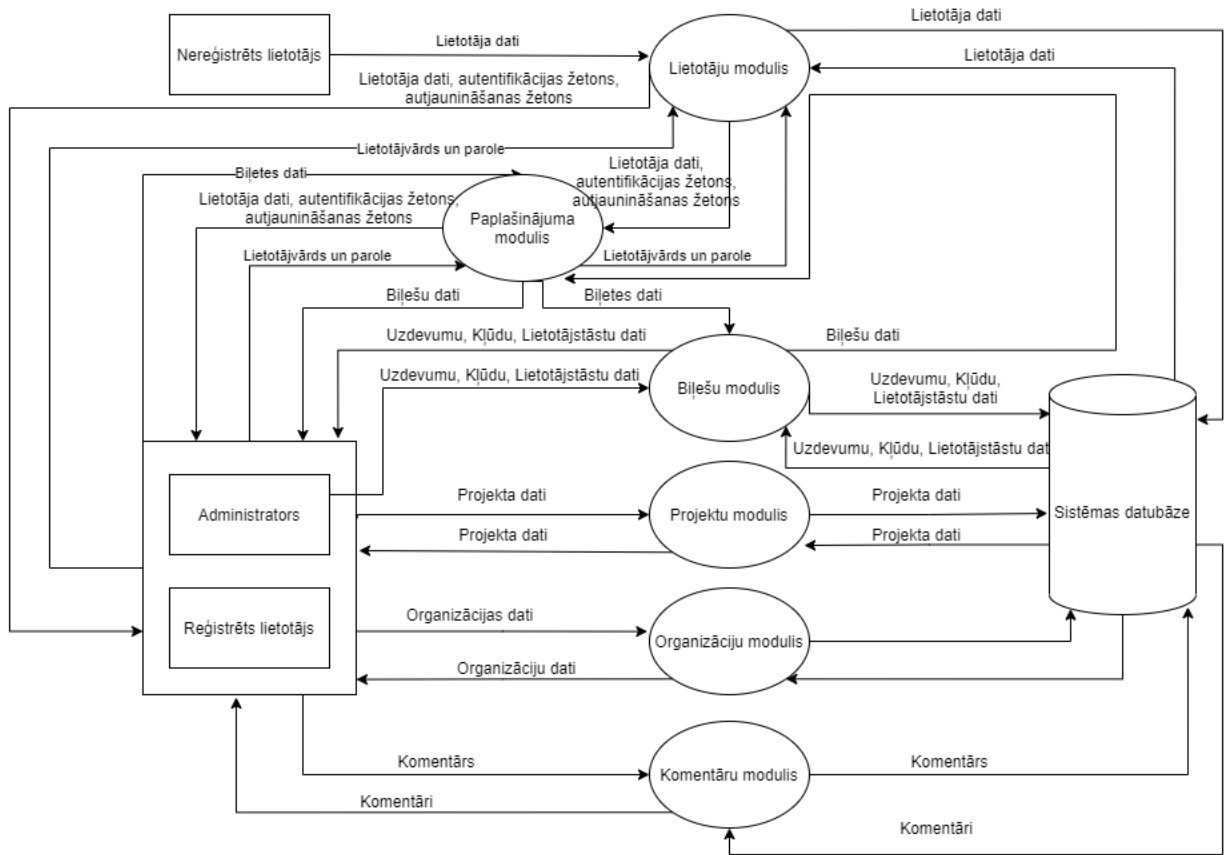
3.1. Funkcionālās prasības

Sistēma ir vienoti aprakstīta vienā diagrammā, kas redzama attēlā 3.1. Tajā var redzēt datu plūsmu pa moduļiem un datubāzēm, kā arī var redzēt kādi dati tiek sūtīti katram lietotāju tipam. Sistēma tiek iedalīta 5 moduļos, kur katram modulim ir savas funkcijas ar identifikatoriem.

Modulis	Funkcija	Identifikators	Lietotāju grupas, kuras var izmantot
Lietotāja	Reģistrācija	LM-01	Nereģistrēts lietotājs
	Autentifikācija	LM-02	Reģistrēts lietotājs
	Žetona atjaunināšana	LM-03	
	Atteikšanās	LM-04	
Organizācijas	Organizāciju skatīšana	OR-01	Administrators
	Organizācijas izveidošana	OR-02	
	Organizācijas datu atjaunināšana	OR-03	
	Organizācijas datu dzēšana	OR-04	
	Projektu skatīšanas	PR-01	Reģistrēts lietotājs

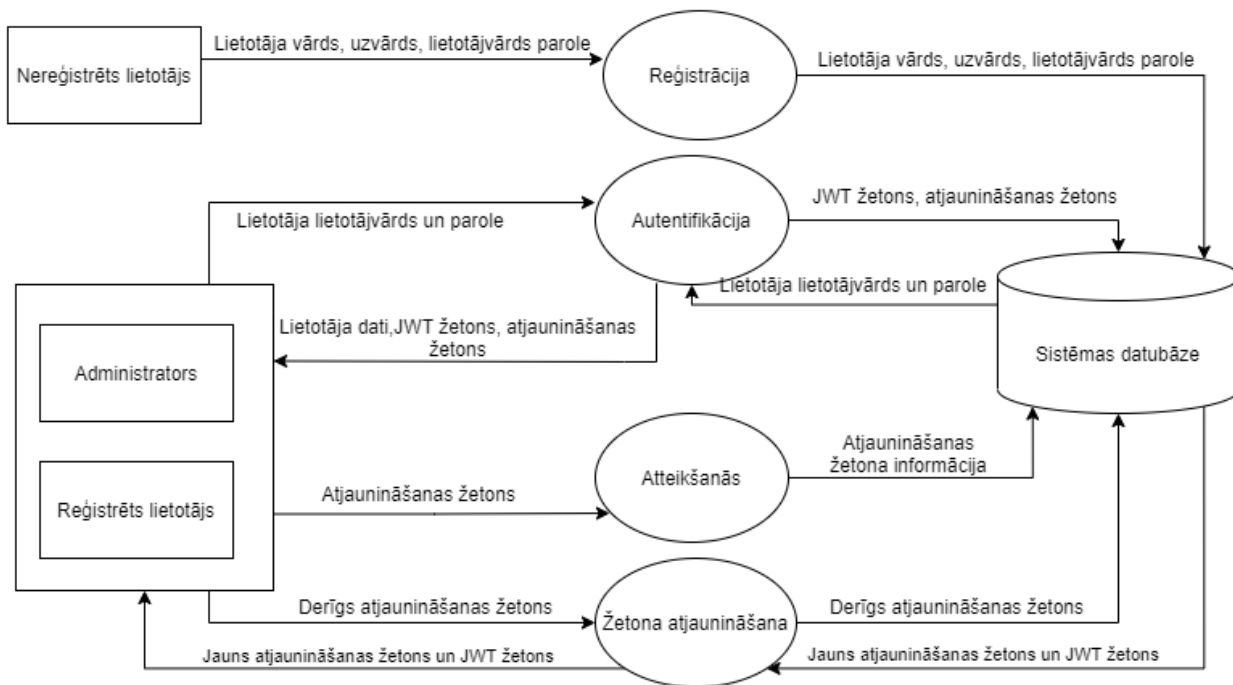
	Projektu izveidošanas funkcija	PR-02	
	Projektu rediģēšanas funkcija	PR-03	
	Projekta datu dzēšanas funkcija	PR-04	
Biļešu	Projektu lietotājstāstu skatīšana	BL-01	Reģistrēts lietotājs
	Lietotājam projektā piešķirto biļešu skats	BL-02	
	Projekta paneļa skatīšanas funkcija	BL-03	
	Lietotāja paneļa skatīšanās funkcija	BL-04	
	Lietotājstāsta izveidošanas funkcija	BL-05	
	Lietotājstāsta rediģēšana	BL-06	
	Lietotājstāsta dzēšanas	BL-07	
	Biļetes izveidošana	BL-08	
	Biļetes rediģēšana	BL-09	
	Biļetes dzēšanas funkcija	BL-10	
Komentāru	Komentāru izveide	KM-01	Reģistrēts lietotājs
	Komentāru skatīšana	KM-02	
	Komentāra dzēšanas	KM-03	Administrators

Paplašinājuma	Paplašinājuma autentifikācijas	PP-01	Reģistrēts lietotājs
	Paplašinājuma biļešu skatīšanas	PP-02	
	Paplašinājuma biļešu statusa rediģēšanas funkcija	PP-03	



3.1 attēls 1.līmeņa DPD

3.1.1. Lietotāju modulis



3.1.1. attēls 2. līmeņa DPD- Lietotāju modulim

Attēlā 3.1.1. ir attēlots lietotāju moduļa 2. līmeņa DPD. Lietotāju modulī ietilps tās funkcijas, kuras dod iespēja lietotājam sistēmā reģistrēties, autentificēties, atteikties, atjaunināt autentifikācijas žetonu. Reģistrējoties tiek nosūtīti vārds uzvārds lietotājvārds un parole. Savukārt autentificējoties lietotāja lietotājvārds un parole tiek aizsūtīts uz tīmekļa API servisu, savukārt tīmekļa API serviss atbild ar lietotāja datiem un JWT žetonu un Atjaunināšanas žetonu un tad tīmekļa lietotne novirza lietotāju uz Organizāciju skatu. Atsakoties uz tīmekļa API servisu tiek aizsūtīts atjaunināšanas žetons, kuru serviss padara par nederīgu un tad lietotājs tiek novirzīts uz autentifikācijas skatu, kas redzams 4.3.2. attēlā. Kad lietotāja JWT žetonam ir beidzies derīgums, tad tiek iegūts atjaunināts žetons, sūtot uz tīmekļa API servisu derīgu atjaunināšanas žetonu, kur atpakaļ tiek saņemts jauns, derīgs JWT žetons, kā arī jauns atjaunināšanas žetons.

3.1.1. tabula Reģistrācijas funkcija

Nosaukums	Reģistrācija	Identifikators	LM-01
Mērķis	Iespēja lietotājam reģistrēties sistēmā.		
Ievade	<ol style="list-style-type: none"> 1. Vārds 2. Uzvārds 3. E-pasts 4. Lietotājvārds 5. Parole 	<ol style="list-style-type: none"> 1. Simbolu virkne 2. Simbolu virkne 3. Simbolu virkne (obligāts) 4. Simbolu virkne (obligāts) 5. Simbolu virkne (obligāts, jābūt vismaz 6 simbolus garai, jābūt mazajiem burtiem, lielajiem burtiem, un ne Alfabēta un ne Ciparu rakstzīmei) 	
Apstrāde	<p>Pogas "Reģistrēties" nospiešanas brīdi</p> <ol style="list-style-type: none"> 1. Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1. 2. Ja parole nav vismaz 6 simbolus gara, kļūdas paziņojums 2. 3. Ja parole nesatur mazos burtus, kļūdas paziņojums 3. 4. Ja parole nesatur lielos burtus, kļūdas paziņojums 4. 5. Ja parole nesatur ne alfabēta un ne ciparu rakstzīmes, kļūdas paziņojums 5. 6. Ja lietotājvārds pastāv, tad kļūdas paziņojums 6. 7. Ja e-pasts nepareizs, tad kļūdas paziņojums 7.- 		
Izvade	Veiksmīgi	Lietotājs nonāk Organizācijas skatā	
	Kļūda	<ol style="list-style-type: none"> 1. Jāievada visi nepieciešamie dati 2. Parole jābūt vismaz 6 simbolus garai 3. Parolei jāsaturs mazie burti 4. Parolei jāsaturs lielie burti 5. Parolei jāsaturs ne alfabēta un ne ciparu rakstzīmes 	

		6. Lietotājavārds jau pastāv. 7. E-pasts ir nepareizs
--	--	--

3.1.2. tabula Autenticēšanās funkcija

Nosaukums	Autenticēšanās	Identifikators	LM-02
Mērķis	Iespēja lietotājam autenticēties sistēmā		
Ievade	1. Lietotājavārds 2. Parole	Simbolu virkne (obligāts) Paroles lauks(obligāts)	
Apstrāde	Pogas "Login" nospiešanas brīdī 1. Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1. 2. Ja Lietotājavārds netiek atrasts kļūdas paziņojums. 2. 3. Ja Parole ir nepareiza kļūdas paziņojums 2.		
Izvade	Veiksmīgi	Lietotājs nokļūst lietotnes Organizāciju skatā	
	Kļūda	1. Jāievada visi nepieciešamie dati 2. Lietotāja vārds vai parole ir nepareizs	

3.1.3. tabula Žetona atjaunināšanas funkcija

Nosaukums	Žetona atjaunināšanas funkcija	Identifikators	LM-03
Mērķis	Iespēja paildzināt autentifikācijas laiku sistēmā		
Ievade	Nav		
Apstrāde	Kad JWT žetons paliek nederīgs tiek iegūts jauns žetons nomainot arī atjaunināšanas žetonu		
Izvade	Veiksmīgi	Lietotājs paliek autenticējies	

	Kļūda	Lietotājs tiek novirzīts uz autentificēšanās skatu
--	-------	--

3.1.4. tabula Atteikšanās funkcija

Nosaukums	Atteikšanās funkcija	Identifikators	LM-04
Mērķis	Iespēja atteikties no sistēmas		
Ievade	Poga "Logout"		
Apstrāde	Atjaunināšanas žetons tiek sūtīts uz timekļa API servisu, kurā		
Izvade	Veiksmīgi	Lietotājs atsakās no sistēmas	
	Kļūda	Nav	

2.1.2. Organizāciju modulis



3.1.2. attēls 2. līmeņa DPD- Organizāciju modulim

3.1.2. attēlā attēlots organizāciju moduļa 2. līmeņa DPD. Organizāciju modulī ietilpst organizāciju skatīšanās funkcija, organizācijas izveidošanas funkcija, organizācijas datu atjaunināšanas funkcija, organizācijas dzēšanas funkcija.

3.1.4. tabula Organizāciju skatīšanās funkcija

Nosaukums	Organizāciju skatīšanās	Identifikators	OR-01
Mērķis	Iespēja skatīt lietotājam piesaistītās organizācijas.		
Ievade	Nav		
Apstrāde	No datubāzes tiek atsūtītas lietotājam piesaistītās organizācijas.		
Izvade	Veiksmīgi	Lietotājs nokļūst lietotnes Organizāciju skatā	

3.1.5. tabula Organizācijas izveidošanas funkcija

Nosaukums	Organizācijas izveidošanas funkcija	Identifikators	OR-02
Mērķis	Iespēja administratoram izveidot organizāciju		
Ievade	<ol style="list-style-type: none"> 1. Nosaukums 2. Apraksts 3. Lietotāju izvēle 	<ol style="list-style-type: none"> 1. Simbolu virkne (obligāts) 2. Simbolu virkne 3. Izvēle no dotā saraksta 	
Apstrāde	<p>Pogas "Save" nospiešanas brīdi</p> <ol style="list-style-type: none"> 1. Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1. 		
Izvade	Veiksmīgi	Tiek organizācija tiek saglabāta datubāze un izveidotājs, kā tās lietotājs	
	Kļūda	1. Jāievada visi nepieciešamie dati	

3.1.6. tabula Organizācijas datu atjaunināšanas funkcija

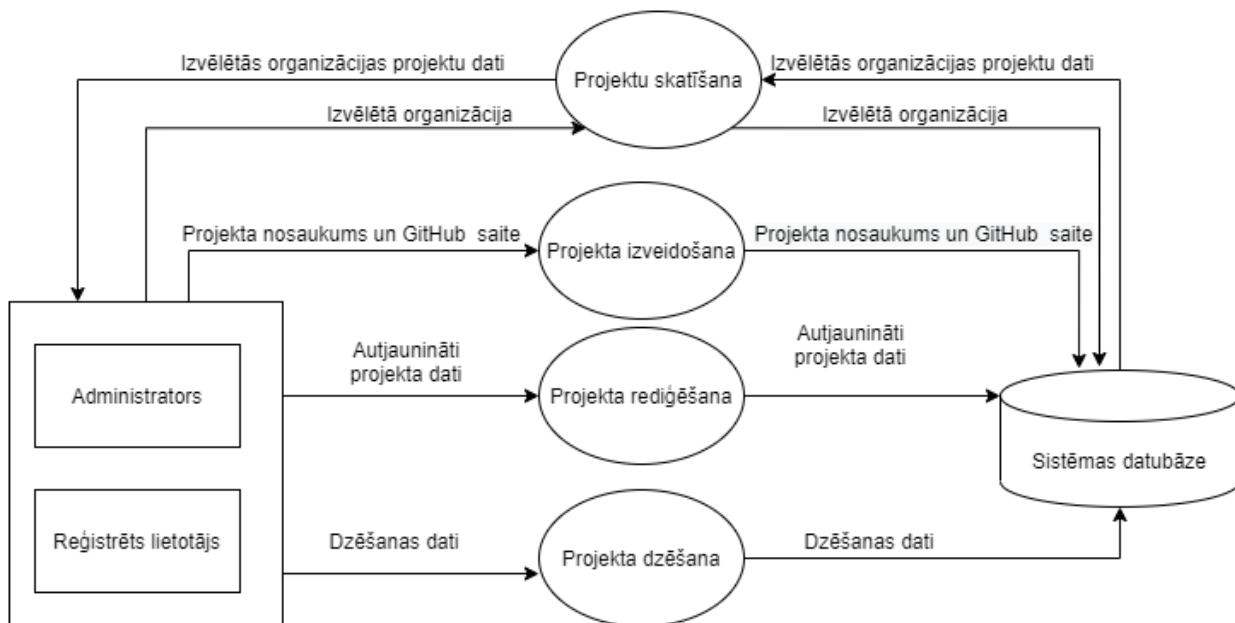
Nosaukums	Organizācijas datu atjaunināšanas funkcija	Identifikators	OR-03
Mērķis	Iespēja administratoram labot organizācijas datus		
Ievade	<ol style="list-style-type: none"> 4. Nosaukums 5. Apraksts 6. Lietotāju izvēle 	<ol style="list-style-type: none"> 4. Simbolu virkne (obligāts) 5. Simbolu virkne 6. Izvēle no dotā saraksta 	
Apstrāde	<p>Pogas "Save" nospiešanas brīdi</p> <ol style="list-style-type: none"> 1. Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1. 		
Izvade	Veiksmīgi	Tiek organizācijas dati datubāze tiek atjaunoti ar jaunajiem datiem.	

	Kļūda	1. Jāievada visi nepieciešamie dati
--	-------	-------------------------------------

3.1.7. tabula Organizācijas datu dzēšanas funkcija

Nosaukums	Organizācijas datu dzēšanas funkcija	Identifikators	OR-04
Mērķis	Iespēja administratoram dzēst organizācijas datus		
Ievade	Poga "Delete"		
Apstrāde	Pogas "Delete" nospiešanas brīdi, 1. Ja nav savienojuma kļūdas paziņojums 1.		
Izvade	Veiksmīgi	Tiek organizācijas datubāze izdzēsta organizācija.	
	Kļūda	1. Kļūda savienojumā	

3.1.3. Projektu modulis



3.1.3. attēls 2. līmeņa DPD- Projektu modulim

3.1.3. attēlā attēlots projektu moduļa 2 līmeņa DPD. Modulī ietilpst projektu skatīšanas funkcija, kura parāda izvēlētās organizācijas projektus, projektu izveidošanas funkcija, ar kuru organizācijai var pievienot jaunu projektu, projektu rediģēšanas funkcija, organizācijas dzēšanas funkcija.

3.1.8. tabula Projektu skatīšanas funkcija

Nosaukums	Projektu skatīšanas	Identifikators	PR-01
Mērķis	Iespēja skatīt organizācijas projektus		
Ievade	Organizācijas izvēle		
Apstrāde	Kad tiek izvēlēta organizācija, no datubāzes lietotājam tiek aizsūtītas izvēlētās organizācijas projekti		
Izvade	Veiksmīgi	Lietotājs nokļūst lietotnes Organizāciju skatā	
	Kļūda	1. Nav	

3.1.5. tabula Projekta izveidošanas funkcija

Nosaukums	Projekta izveidošanas funkcija	Identifikators	PR-02
Mērķis	Ir iespēja izveidot projektu		
Ievade	1. Nosaukums 2. GitHub saite	Simbolu virkne(obligāts) Simbolu virkne	
Apstrāde	Pogas "Save" nospiešanas brīdi 1. Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1.		
Izvade	Veiksmīgi	Izveidotā organizācija tiek saglabāta datubāzē un izveidotājs, kā tās lietotājs	
	Kļūda	1. Jāievada visi nepieciešamie dati	

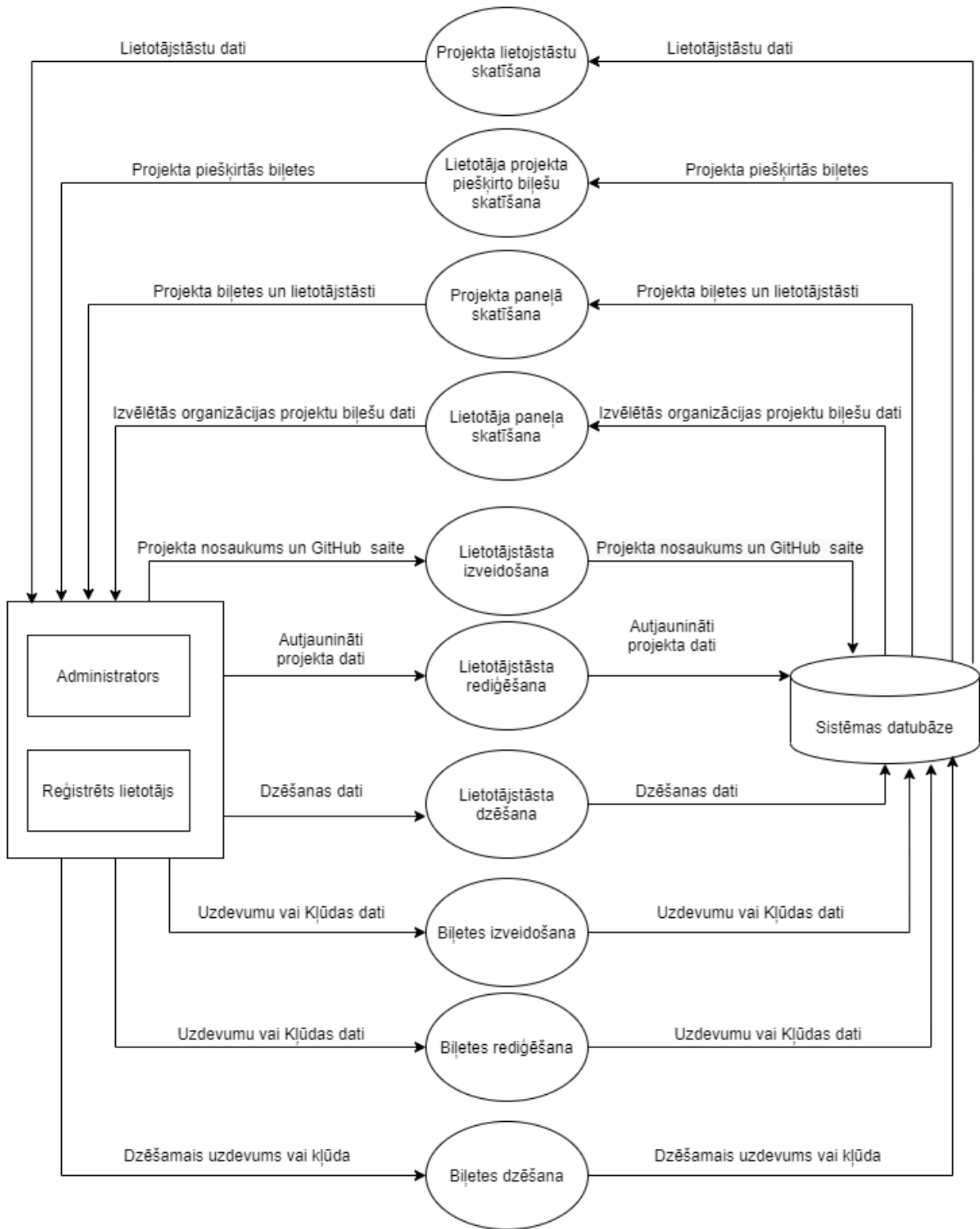
3.1.6. tabula Projekta rediģēšanas funkcija

Nosaukums	Projekta rediģēšanas funkcija	Identifikators	PR-03
Mērķis	Ir iespēja labot organizācijas datus		
Ievade	1. Nosaukums 2. GitHub saite	1. Simbolu virkne(obligāts) 2. Simbolu virkne	
Apstrāde	Pogas "Save" nospiešanas brīdi 1. Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1.		
Izvade	Veiksmīgi	Rediģētā organizācija tiek saglabāta datubāzē	
	Kļūda	1. Jāievada visi nepieciešamie dati	

3.1.7. tabula Projekta datu dzēšanas funkcija

Nosaukums	Projekta datu dzēšanas funkcija	Identifikators	PR-04
Mērķis	Ir iespēja dzēst organizācijas projektu.		
Ievade	Poga "Delete"		
Apstrāde	Pogas "Delete" nospiešanas brīdi, tiek izsaukta dzēšanas API izsaukums. 1. Ja nav savienojuma kļūdas paziņojums 1.		
Izvade	Veiksmīgi	Datubāze tiek izdzēsts projekts	
	Kļūda	1. Kļūda savienojumā	

3.1.4. Biļešu modulis.



3.1.4. attēls 2. līmeņa DPD- Biļešu moduļim

3.1.8. tabula Projekta lietotājstāstu skatīšanas funkcija

Nosaukums	Projekta lietotājstāstu skatīšanas funkcija	Identifikators	BL-01
Mērķis	Iespēja skatīt projekta lietotājstāstus		
Ievade	Navigācijas laukā izvēlēts projekts un navigēts uz Lietotājstāstu skatu	Nav	
Apstrāde	Kad tiek navigēts uz skatu no datubāzes tiek iegūti projekta lietotājstāsta dati.		
Izvade	Veiksmīgi	Lietotājs nokļūst lietotnes Organizāciju skatā	
	Kļūda	1. Nav	

3.1.9. tabula Lietotājam projektā piešķirto biļešu skatīšanas funkcija

Nosaukums	Lietotājam projektā piešķirto biļešu skatīšanas funkcija	Identifikators	BL-02
Mērķis	Iespēja skatīt projektā piešķirtās biļetes		
Ievade	Navigācijas laukā izvēlēts projekts un navigēts uz Piešķirto biļešu skatu.	Nav	
Apstrāde	Kad tiek navigēts uz skatu no datubāzes tiek iegūti projekta lietotājstāsta dati.		
Izvade	Veiksmīgi	Lietotājs nokļūst lietotnes Organizāciju skatā	
	Kļūda	1. Nav	

3.1.10. tabula Projekta paneļa skatīšanas funkcija

Nosaukums	Projekta paneļa skatīšanas funkcija	Identifikators	BL-03
Mērķis	Ir iespēja skatīt projekta uzdevumus, kļūdas un lietotārstāstus		
Ievade	Projekts	Saraksta izvēle(obligāts)	
Apstrāde	Kad lietotājs tiek aizvirzīts uz skatu, tiek iegūti projekta kļūdu, un uzdevumu, lietotārstāstu dati		
Izvade	Veiksmīgi	Lietotājs nokļūst lietotnes organizāciju skatā	
	Kļūda	Nav	

3.1.11. tabula Lietotāja paneļa skatīšanas funkcija

Nosaukums	Lietotāja paneļa skatīšanas funkcija	Identifikators	BL-04
Mērķis	Ir iespēja skatīt organizācijā piešķirtās biļetes		
Ievade	Navigācijas laukā izvēlēta organizācija	Nav	
Apstrāde	Kad tiek navigēts uz skatu no datubāzes tiek iegūti organizācijas kļūdu un uzdevumu dati, kuri ir piešķirti lietotājam		
Izvade	Veiksmīgi	Lietotājs nokļūst lietotnes Organizāciju skatā un ieraugu lietotājam piešķirtās projektu kļūdas un uzdevumus, sašķirotas pēc prioritātes	
	Kļūda	Nav	

3.1.12. tabula Lietotārstāsta izveidošanas funkcija

Nosaukums	Lietotārstāsta izveidošanas funkcija	Identifikators	BL-05
-----------	---	----------------	-------

Mērķis	Ir iespēja izveidot Lietotājstāstu	
Ievade	<ol style="list-style-type: none"> Nosaukums Apraksts Sākuma datums Beigu datums Statuss Piešķirtā lietotāja izvēle 	<ol style="list-style-type: none"> Simbolu virkne(obligāta) Simbolu virkne Datuma izvēle Datuma izvēle Saraksta izvēle Saraksta izvēle
Apstrāde	Pogas "Save" nospiešanas brīdi <ol style="list-style-type: none"> Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1. 	
Izvade	Veiksmīgi	Lietotājstāsts tiek saglabāts datubāze
	Kļūda	1. Jāievada visi nepieciešamie dati

3.1.13. tabula Lietotājstāsta rediģēšanas funkcija

Nosaukums	Lietotājstāsta rediģēšana funkcija	Identifikators	BL-06
Mērķis	Ir iespēja rediģēt Lietotājstāstu		
Ievade	<ol style="list-style-type: none"> Nosaukums Apraksts Sākuma datums Beigu datums Statuss Piešķirtā lietotāja izvēle 	<ol style="list-style-type: none"> Simbolu virkne(obligāta) Simbolu virkne Datuma izvēle Datuma izvēle Saraksta izvēle Saraksta izvēle 	
Apstrāde	Pogas "Save" nospiešanas brīdi <ol style="list-style-type: none"> Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1. 		

Izvade	Veiksmīgi	Lietotājstāsta informācija tiek datubāzē atjaunota
	Kļūda	1. Jāievada visi nepieciešamie dati

3.1.14.. tabula Lietotājstāsta dzēšanas funkcija

Nosaukums	Lietotājstāsta dzēšanas funkcija	Identifikators	BL-07
Mērķis	Ir iespēja dzēst Lietotājstāstu		
Ievade	Poga "Delete"		
Apstrāde	Pogas "Delete" nospiešanas brīdī nosūtīts pieprasījums dzēst ierakstu. 1. Ja neveiksmīgs savienojums, kļūdas paziņojums 1.		
Izvade	Veiksmīgi	Lietotājstāsts tiek izdzēsts	
	Kļūda	1. Jāievada visi nepieciešamie dati	

3.1.15. tabula Biļetes izveidošanas funkcija

Nosaukums	Biļetes izveidošanas funkcija	Identifikators	BL-08
Mērķis	Ir iespēja izveidot kļūdu vai uzdevumu		
Ievade	<ol style="list-style-type: none"> 1. Nosaukums 2. Apraksts 3. Lietotājstāsts 4. Statuss 5. Piešķirtā lietotāja izvēle 6. Prioritāte 7. Tips 	<ol style="list-style-type: none"> 1. Simbolu virkne(obligāta) 2. Simbolu virkne 3. Saraksta izvēle 4. Saraksta izvēle 5. Saraksta izvēle 6. Saraksta izvēle 7. Saraksta izvēle 	

Apstrāde	Pogas "Save" nospiešanas brīdi 1. Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums	
Izvade	Veiksmīgi	Biļete tiek saglabāta datubāzē
	Kļūda	1. Jāievada visi nepieciešamie dati

3.1.16. tabula Biļetes rediģēšanas funkcija

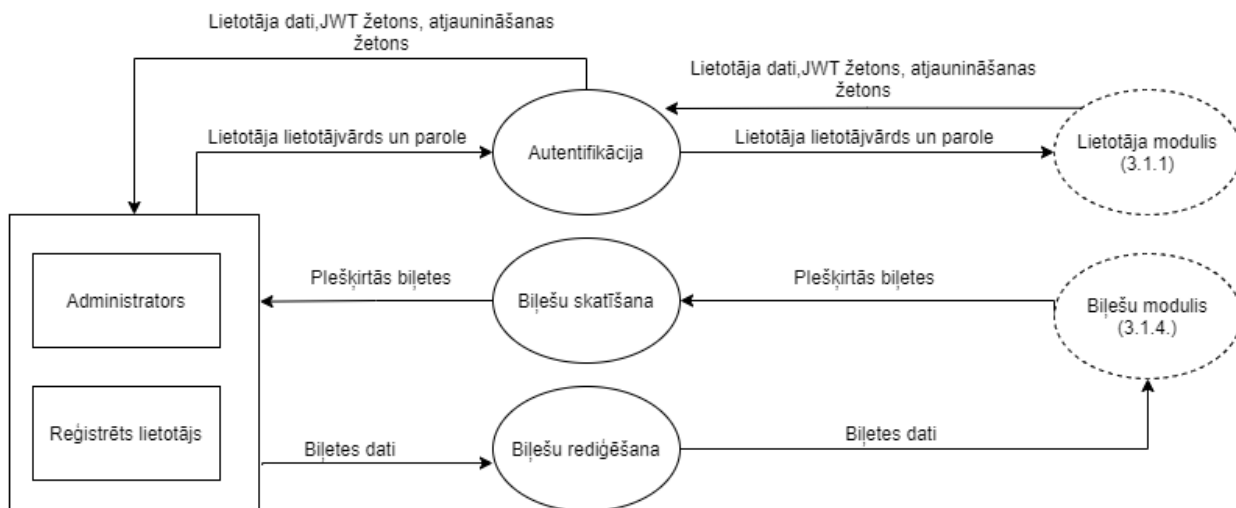
Nosaukums	Biļetes rediģēšana funkcija	Identifikators	BL-09
Mērķis	Ir iespēja rediģēt Biļeti		
Ievade	<ol style="list-style-type: none"> 1. Nosaukums 2. Apraksts 3. Lietotājstāsts 4. Statuss 5. Piešķirtā lietotāja izvēle 6. Prioritāte 7. Tips 	<ol style="list-style-type: none"> 1. Simbolu virkne(obligāta) 2. Simbolu virkne 3. Saraksta izvēle 4. Saraksta izvēle 5. Saraksta izvēle 6. Saraksta izvēle 7. Saraksta izvēle 	
Apstrāde	Pogas "Save" nospiešanas brīdi 2. Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1.		

Izvade	Veiksmīgi	Bīletes informācija tiek datubāzē atjaunota ar jauno informāciju
	Kļūda	1. Jāievada visi nepieciešamie dati

3.1.17.. tabula Bīletes dzēšanas funkcija

Nosaukums	Bīletes dzēšanas funkcija	Identifikators	BL-10
Mērķis	Ir iespēja dzēst Bīleti		
Ievade	Poga "Delete"		
Apstrāde	Pogas "Delete" nospiešanas brīdi nosūtīts pieprasījums dzēst ierakstu. 2. Ja neveiksmīgs savienojums, kļūdas paziņojums Nr. 1.		
Izvade	Veiksmīgi	Bīlete tiek izdzēsta.	
	Kļūda	1. Savienojuma kļūda	

3.1.5. Komentāru modulis



3.1.18. tabula Komentāru izveides funkcija

Nosaukums	Komentāru izveides funkcija	Identifikators	KM-01
Mērķis	Ir iespēja izveidot komentāru kļūdai vai uzdevumam		
Ievade	Izvēlētā biļete Komentārs	Saraksta izvēle(obligāts) Simbolu virkne(obligāts)	
Apstrāde	Nospiežot pogu "Add". 1.Ja nav ievadīts komentārs kļūda paziņojums 1		
Izvade	Veiksmīgi	Komentārs tiek saglabāts.	
	Kļūda	1. Ievadiet komentāru	

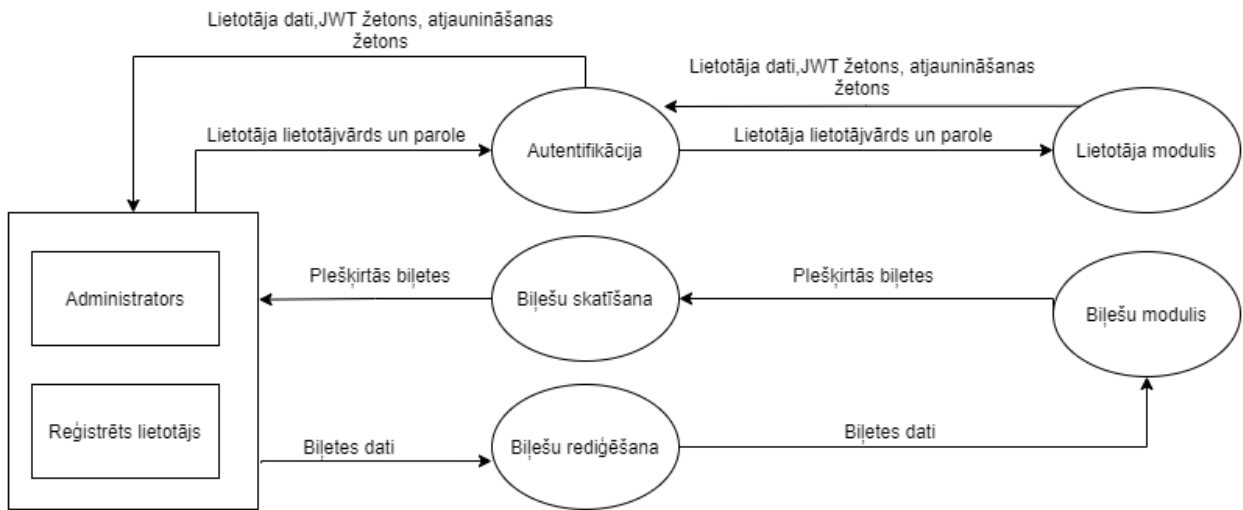
3.1.19. tabula Komentāru skatīšanas funkcija

Nosaukums	Komentāru skatīšanas funkcija	Identifikators	KM-02
Mērķis	Skatīt Biļetes komentārus		
Ievade	Atverot biļetes skatu, navigācijā jāuzspiež uz pogas “Komentāri”		
Apstrāde	Atverot Komentārus no tiek veikts API izsaukums, lai saņemtu Komentārus.		
Izvade	Veiksmīgi	Biļetes informācija tiek datubāzē atjaunota ar jauno informāciju	
	Kļūda	Nav	

3.1.20. tabula Komentāra dzēšanas funkcija

Nosaukums	Komentāra dzēšanas funkcija	Identifikators	KM-03
Mērķis	Iespēja Administratoram dzēst komentārus		
Ievade	Uz komentāra jāuzspiež uz dzēšanas pogas		
Apstrāde	Pogas “Delete” nospiešanas brīdi nosūtīts pieprasījums dzēst ierakstu. 3. Ja neveiksmīgs savienojums, kļūdas paziņojums Nr. 1.		
Izvade	Veiksmīgi	Komentārs tiek izdzēsts.	
	Kļūda	1. Savienojuma kļūda	

3.1.6. Paplašinājuma modulis



3.1.21. tabula Paplašinājuma autentifikācijas funkcija

Nosaukums	Paplašinājuma autentifikācijas funkcija	Identifikators	PP-01
Mērķis	Iespēja lietotājam autentificēties ar Visual Studio Code paplašinājumu		
Ievade	1. Lietotājavārds 2. Parole	Simbolu virkne (obligāts) Paroles lauks(obligāts)	
Apstrāde	Pogas “Enter” nospiešanas brīdī 1. Ja nav aizpildīti visi obligātie lauki, kļūdas paziņojums 1. 2. Ja Lietotājavārds netiek atrasts datubāzē kļūdas paziņojums. 2. 3. Ja Parole ir nepareiza kļūdas paziņojums 2.		
Izvade	Veiksmīgi	Lietotājs ir autentificējies	
	Kļūda	1. Jāievada visi nepieciešamie dati 2. Lietotāja vārds vai parole ir nepareizi	

3.1.22. tabula Paplašinājuma biļešu skatīšanas funkcija

Nosaukums	Paplašinājuma biļešu skatīšanas funkcija	Identifikators	PP-02
Mērķis	Iespēja lietotājam skatīt organizācijā piešķirtās biļetes		
Ievade	Pavērsts paplašinājuma skats		
Apstrāde	No tīmekļa API tiek pieprasītas piešķirtās biļetes.		
Izvade	Veiksmīgi	Biļešu informācija parādās skatā	
	Kļūda	Nav	

3.1.20. tabula Paplašinājuma biļetes rediģēšanas funkcija

Nosaukums	Paplašinājuma biļetes rediģēšana	Identifikators	PP-03
Mērķis	Iespēja lietotājam rediģēt biļetes statusu ar Visual Studio Code paplašinājumu		
Ievade	Statuss	Saraksta(izvēle)	
Apstrāde	4. Kad tiek no saraksta izvēlēts biļetes statuss, tad tiek izsaukts tīmekļa API, ar kuru tiek nomainīts biļetes statuss.		
Izvade	Veiksmīgi	Biļetes statuss ir nomainīts	
	Kļūda	2. Savienojuma kļūda	

4.2. Nefunkcionālās prasības

4.2.1. Lietojamība

Sistēmai jābūt iespējai piekļūt sistēmas datiem neizmantojot tīmekli. 90% no lietotājiem jāsaprot, kā izmantot lietotni, bez citu cilvēku palīdzības.

4.2.2. Uzturamība

Sistēmai tīmekļa lietotnei ir jābūt izstrādātai izmantojot Angular 11(vai jaunāku versiju) un ASP.NET CORE 5(vai jaunāku versiju), jo tām ir izstrādātāju atbalsts vismaz līdz 2022. gada 1. janvārim. Sistēmas API servisam ir jābūt iespējai to palaist Linux vidē.

4.2.3. Pielāgojamība

Sistēmas tīmekļa lietotnē ir vajadzīgs, lai tīmekļa saskarnē lietotājs varētu, mainīt biļetes statusu, pārvelkot to ar kursoru uz vajadzīgo lietotājstāstu un vajadzīgo statusu.

4.2.4. Savienojamība

Sistēmu jāvar izmantot ar izšķirtspēju 828 x 1792 pikseļi, 1536 x 2048 pikseļi, 1080 x 2246. Atverot citu skatu pārlūkprogrammai jāielādē jaunie dati, nevis jāpārlādē visa lapa. Pārejai no skatu uz citu skatu jānotiek 3 sekunžu laikā(Izpildes laikā netiek iekļauta tīkla veikspēja).

4.2.5. Pieejamība

Sistēmai tiek izvirzītas šādas pieejamības prasības

- Sistēmai jāstrādā ar Android 7.0 versiju.
- Sistēmai jāstrādā uz Google Chrome, Microsoft Edge, Mozilla Firefox.

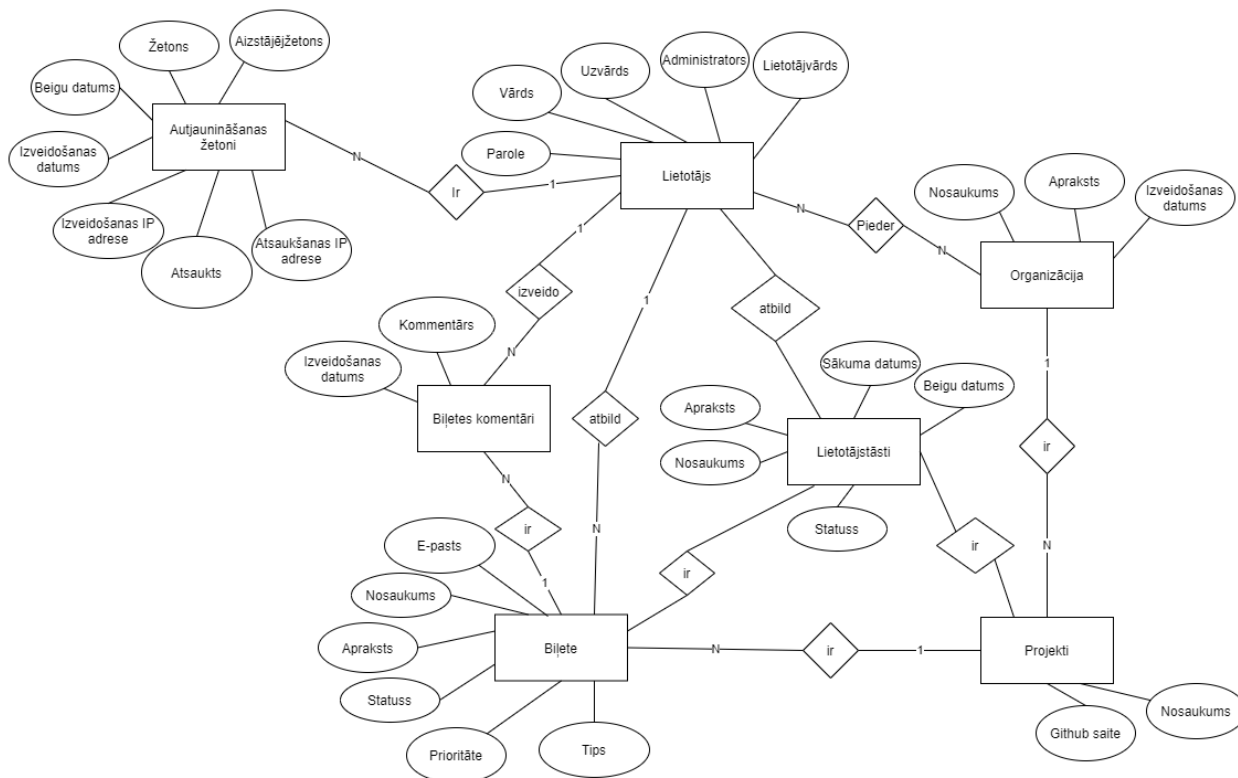
4.2.6. Drošība

Sistēmas ģenerētajiem JWT žetoniem jāatbilst RFC 7519 standartam[4].

4. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

4.1. Datu bāzes projektējums.

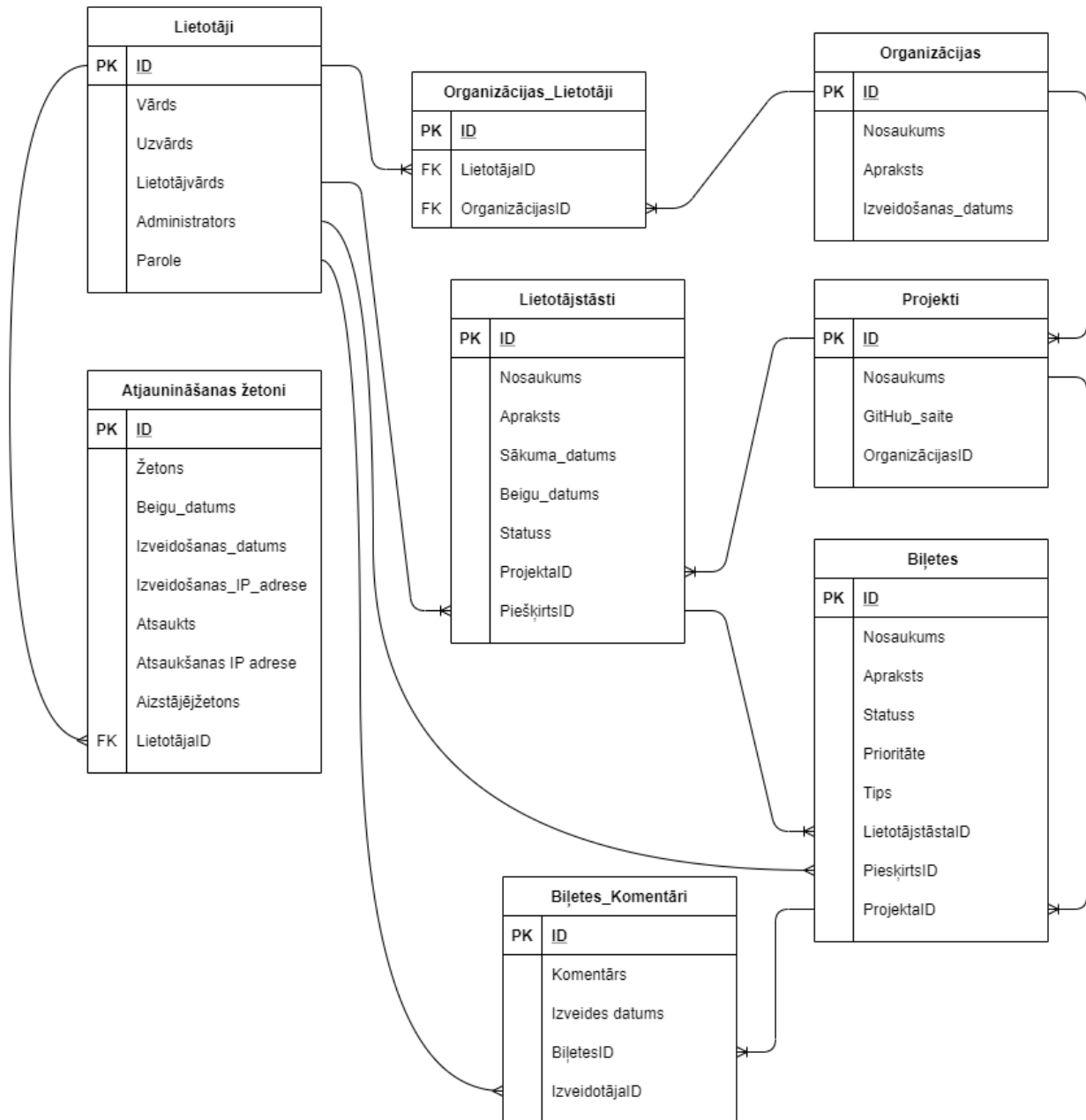
4.1.1. Konceptuālais ER modelis



2.2. Attēls Datu bāzes konceptuālais modelis

4.1. att. ir attēlots Programmēšanas izstrādes pārvaldības sistēmas datu bāzes konceptuālais modelis. Datu bāzē tiek glabāta informācija par lietotāju – vārds, uzvārds, vai ir, vai nav administrators, Lietotājvārds, kā arī tiek uzglabāta paroles jaudcēsumma ko nodrošina ASP.NET CORE Identity[4] mehānisms. Kā arī lietotājam tiek uzglabāti atjaunināšanas žetoni, ar kuriem var atjaunot JWT žetonu, kas tiek nodrošināts ar ASP.NET CORE Identity bibliotēku. Par katru no atjaunināšanas žetoniem tiek glabāts: aizstājējžetons, kā arī pats žetons, beigu datums, izveidošanas datums, izveidošanas IP adrese, tas vai žetons ir derīgs(atsaukts) , atsauksmes IP adrese. Lietotājs tiek piesaistīts organizācijai par kuru tiek glabāts nosaukums un apraksts un kurai ir projekti. Projektiem ir biļetes un lietotājstāsti, savukārt par katru biļetei ir iespējami vairāki komentāri. Biļetes tabulā iekļaujas kļūdas un uzdevumi, tos var atšķirt ar lauka tips palīdzību.

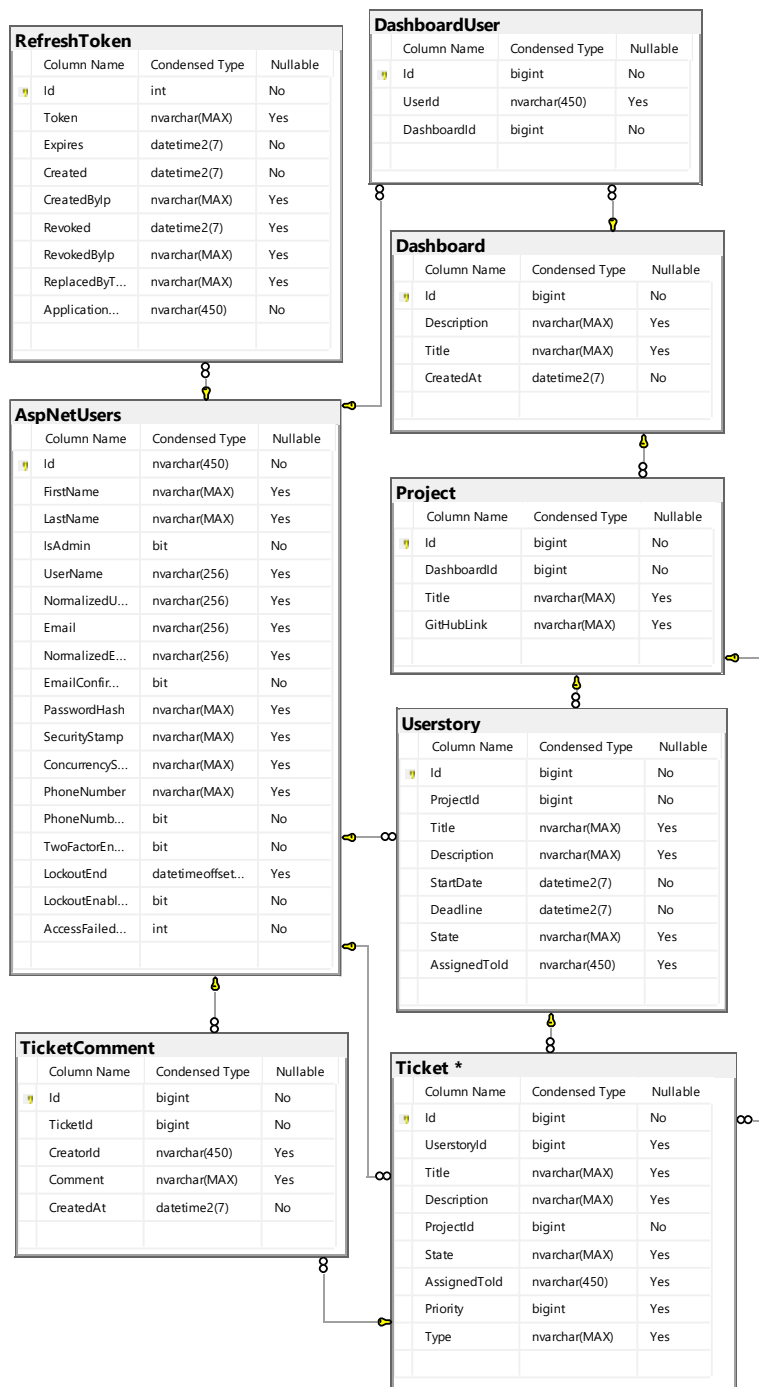
4.1.2. Loģiskais modelis



4.2.att. Datu bāzes loģiskais modelis

Datu bāzes loģiskajā modelī, kurš redzams, 4.2 attēlā datu bāzes tabulas un parādīts, kā tabulas savstarpēji saistītas tabulas. Kā redzams attēlā starp lietotāju un organizāciju ir tikusi izveidota starp tabula. Kā arī to, ka Organizācijai iespējami daudzi projekti. Projektiem iespējamās daudzas biļetes, Biļetēm ir iespējams ka tās pieder pie kāda lietotājstāsta, bet varbūt ka nepieder pie lietotājstāsta, bet pieder tikai pie projekta. Lietotājs stāsts pieder pie projekta un tam var atbilst daudzas biļetes.

4.1.3. Datu bāzes fiziskais modelis



4.3 attēls Datubāzes fiziskais modelis

Datubāzes fiziskajā modelī, kurš ir redzams 4.3 attēlā, ir redzams kā tabulas savstarpēji ir saistītas un kāds tips ir piešķirts katram atribūtam. Atjauninātājzetonu tabula attēlā ir “RefreshTokens”, Lietotāju tabula ir “AspNetUsers”, Organizācijas_Lietotāju tabula attēlā ir

“DashboardUsers”, Organizācijas tabula attēlā ir “Dashboard”, Projektu tabula attēlā ir “Project”, Lietotājstāstu tabula attēlā ir “Userstory”, biļetes tabula attēlā ir “Ticket”. Biļetes_Komentāru tabula attēlā “TicketComment”.

4.1.4. Datu bāzes tabulu apraksti

Tabulā 4.1 redzama Lietotāju tabula. 4.3 attēlā. Tabulā “AspNetUsers” jeb Lietotāja tabulā pārējos laukus izveido ASP NET CORE Identity bibliotēka.

4.1 tabula Lietotāju tabula

Nosaukums	Tips	Obligāts	Atslēga	Apraksts
Id	nvarchar(50)	jā	Primārā	Unikāls ieraksta identifikators, aizpildās automātiski, nodrošinās ASP.NET CORE Identity bibliotēka
FirstName	nvarchar(MAX)	nē		Lietotāja vārds
LastName	nvarchar(MAX)	nē		Lietotāja uzvārds
IsAdmin	bit	nē		Norāda vai ir administrators vai nav
UserName	nvarchar(256)	nē		Lietotājvārds
Email	nvarchar(256)	nē		E-pasts

Tabulā 4.2. redzama Atjaunināšanas žetonu tabula. Šī tabula tiks izmantota, lai izveidotu jaunu JWT žetonu.

4.2 tabula Atjaunināšanas žetoni

Nosaukums	Tips	Obligāts	Atslēga	Apraksts
Id	int	jā	Primārā	Unikāls ieraksta identifikators, aizpildās automātiski
Token	nvarchar(MAX)	nē		Žetons, kura ir nejauša bitu virkne
Expires	datetime	jā		Beigu datums
Created	datetime	jā		Izveidošanas datums
CreatedByIP	nvarchar(MAX)	jā		Izveidošanas IP
Revoked	datetime	nē		Atsaukts
RevokedByIP	nvarchar(MAX)	nē		Atsaukšanas IP
ReplacedByToken	nvarchar(MAX)	nē		Aizstājējžetons

4.3 tabulā redzami Organizācijas lietotāju tabula, kas ir starp tabula starp Lietotājiem un Organizāciju.

4.3.tabula Organizācijas lietotāju tabula

Nosaukums	Tips	Obligāts	Atslēga	Apraksts
Id	int	jā	Primārā	Unikāls ieraksta identifikators, aizpildās automātiski
UserId	nvarchar(50)	jā	Ārējā	Unikāls identifikators tabulas "Lietotajs" ierakstam
DashboardId	bigint	jā	Ārējā	Unikāls identifikators tabulas "Organizācija" ierakstam

4.4 tabulā var redzēt Organizācijas tabulu. Izveidošanas datuma lauku aizpilda sistēma.

4.4.tabula Organizācijas tabula

Nosaukums	Tips	Obligāts	Atslēga	Apraksts
Id	bigint	jā	Primārā	Unikāls ieraksta identifikators, aizpildās automātiski
Description	Nvarchar(max)	nē		Apraksts
Title	Nvarchar(max)	nē		Nosaukums
CreatedAt	datetime	jā		Izveidošanas datums

4.5. tabulā var redzēt projektu tabulu, kurai ir viena arējā atslēga

4.5.tabula Projektu tabula

Nosaukums	Tips	Obligāts	Atslēga	Apraksts
Id	bigint	jā	Primārā	Unikāls ieraksta identifikators, aizpildās automātiski
GitHubLink	Nvarchar(max)	nē		GitHub saite
Title	Nvarchar(max)	nē		Nosaukums
DashboardId	bigint	jā	Arējā	Unikāls identifikators tabulas "Organizācija" ierakstam

4.6. tabulā var redzēt lietotājstāstu tabulu, kurai ir divas arējās atslēgas.

4.6.tabula Lietotājstāstu tabula

Nosaukums	Tips	Obligāts	Atslēga	Apraksts
Id	bigint	jā	Primārā	Unikāls ieraksta identifikators, aizpildās automātiski
Description	Nvarchar(max)	nē		Apraksts
Title	Nvarchar(max)	nē		Nosaukums
ProjectId	bigint	jā	Arējā	Unikāls identifikators tabulas "Projekts" ierakstam

StartDate	datetime	jā		Sākuma datums
Deadline	datetime	jā		Beigu datums
State	Nvarchar(max)	ne		Statuss
AssignedToId	Nvarchar(450)	nē		PiešķirtsID (lietotājs id, kuram piešķirta biļete)

4.7 tabulā var redzēt Biļetes tabulu. Biļetes tabulā tiks glabāti uzdevumi un kļūdas. Biļetes tabulai ir neobligāts lietotājstāsta_ID, jo biļete var būt piederīga gan pie lietotājstāsta, gan arī vienkārši pie projekta.

4.7.tabula Biļetes tabula

Nosaukums	Tips	Obligāts	Atslēga	Apraksts
Id	bigint	jā	Primārā	Unikāls ieraksta identifikators, aizpildās automātiski
Description	Nvarchar(max)	nē		Apraksts
Title	Nvarchar(max)	nē		Nosaukums
ProjectId	bigint	jā	Arējā	Unikāls identifikators tabulas "Projekts" ierakstam
UserstoryId	bigint	nē	Arējā	LietotājstāstaID uz tabulu Lietotājstāsti

ProjectID	bigint	jā	Arējā	ProjektaID uz tabulu Projekti
State	Nvarchar(max)	ne		Statuss
AssignedToId	Nvarchar(450)	nē		PiešķirtsID (lietotājā ID kuram piešķirta biļete)
Priority	bigint	nē		Prioritāte
Type	Nvarchar(max)	nē		Tips

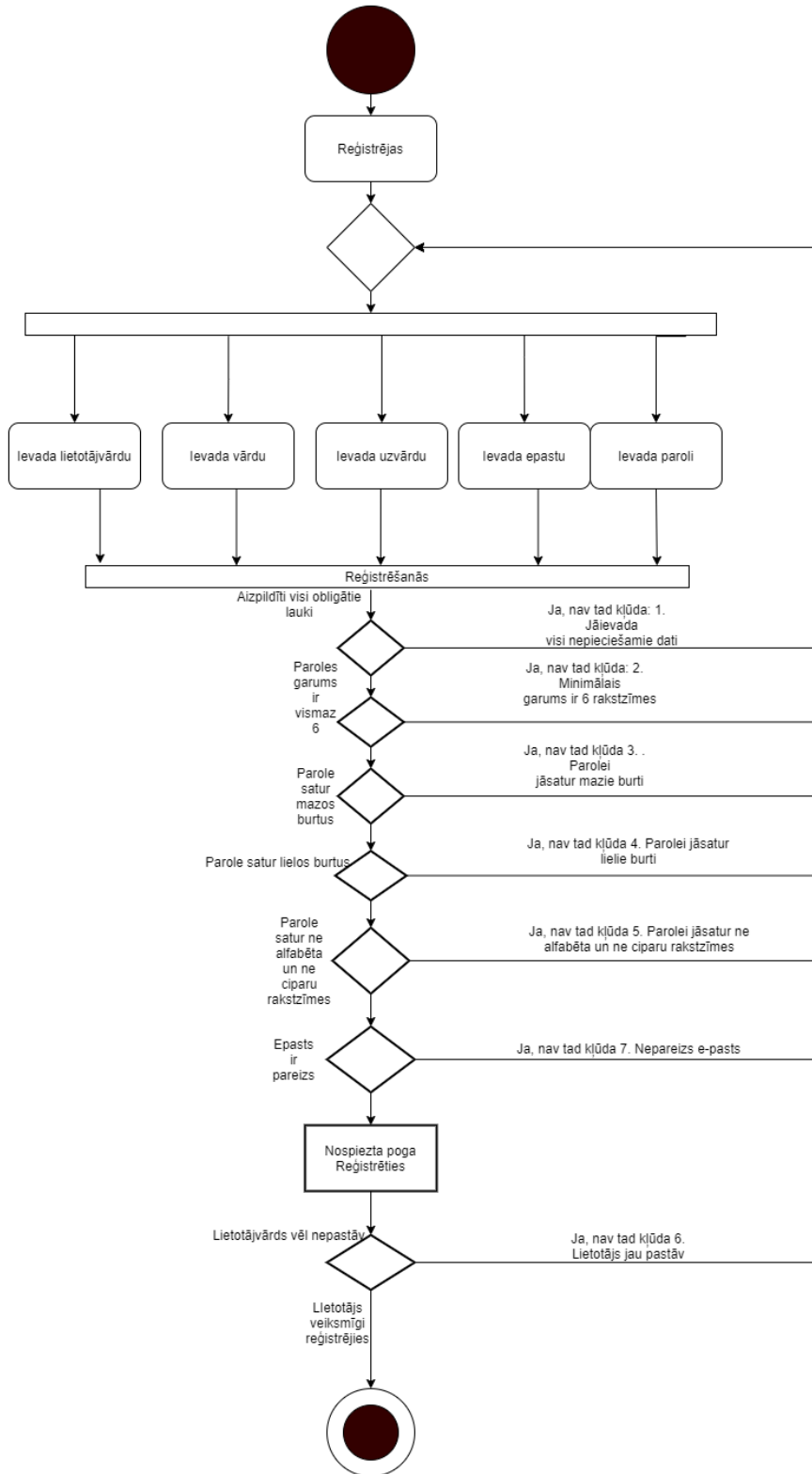
4.8 tabulā var redzēt Biļetes Komentāru tabulu. Biļetes Komentāru tabulā tiks glabāti Komentāri.

4.7.tabula Biļetes tabula

Nosaukums	Tips	Obligāts	Atslēga	Apraksts
Id	bigint	jā	Primārā	Unikāls ieraksta identifikators, aizpildās automātiski
Comment	Nvarchar(max)	nē		Komentārs
TicketId	bigint	jā		BiļetesID(Biļetes ID, par kuru ieraksts attiecas)
CreatorId	Nvarchar(450)	nē		LietotājaID(Lietotāja ID, par kuru attiecas ieraksts)

4.2. Funkciju projektējums

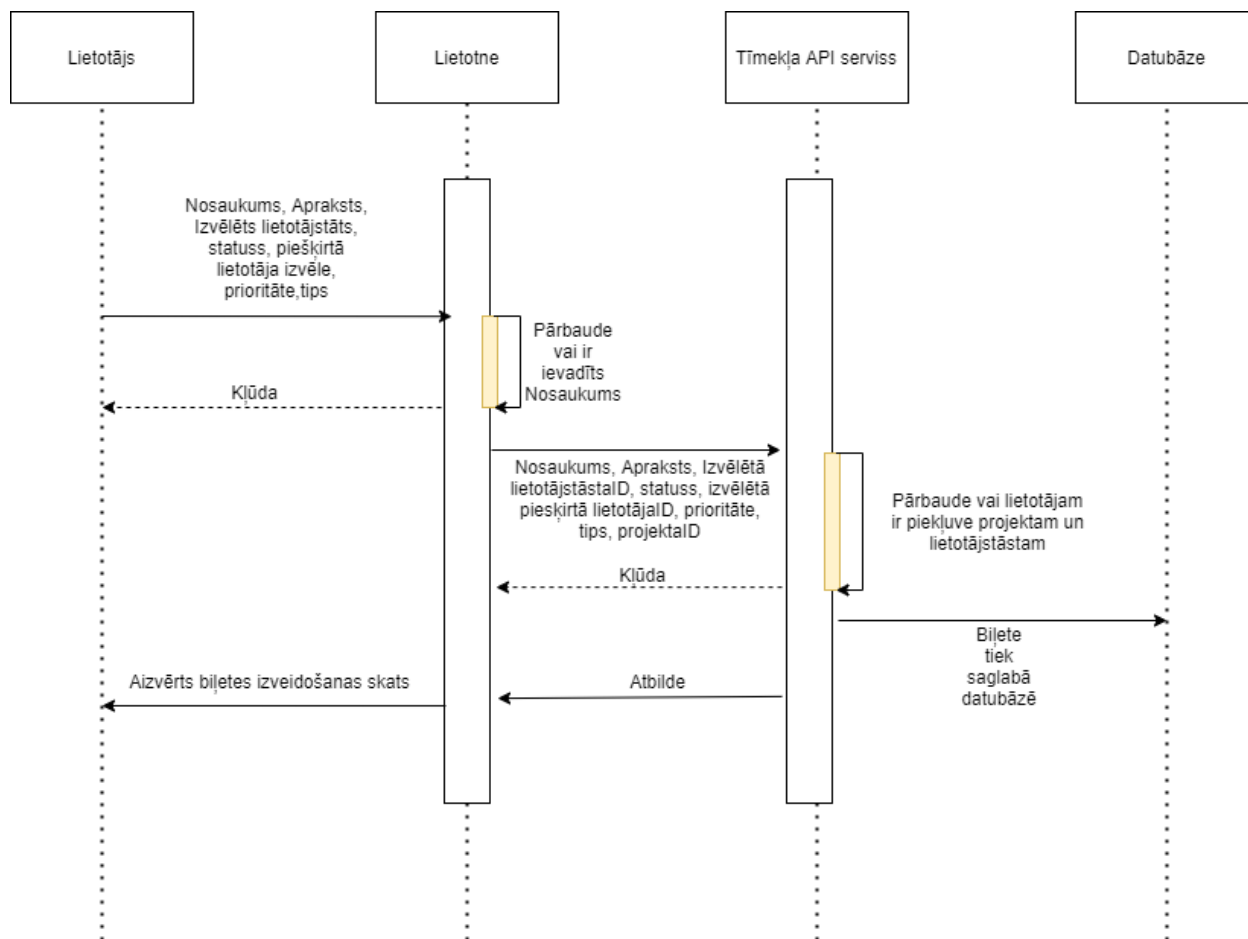
4.2.1. attēls Aktivitāšu diagramma reģistrēšanās funkcijai



4.2.1. attēls Aktivitāšu diagramma reģistrēšanās funkcijai

4.2.1. attēlā Redzama aktivitāšu diagramma reģistrēšanās funkcijai, kā redzams sākumā tiek pieprasīta ievade no lietotāja pēc tam notiek ievades lauku validēšana, Ja kāds no laukiem neatbilst nosacījumiem tiek pieprasīta atkārtota ievade. Tad, ja visi lauki atbilst priekšnosacījumiem tad tiek izsaukta reģistrēšanās tīmekļa API. Ja lietotājs jau pastāv, tad tiek izvadīta kļūda “Lietotājs jau pastāv”, ja lietotājs jau ar šādu lietotājvārdu nepastāv, tad lietotāju veiksmīgi reģistrē sistēmā.

4.2.3. Secību diagramma biļetes izveidošanas funkcijai.



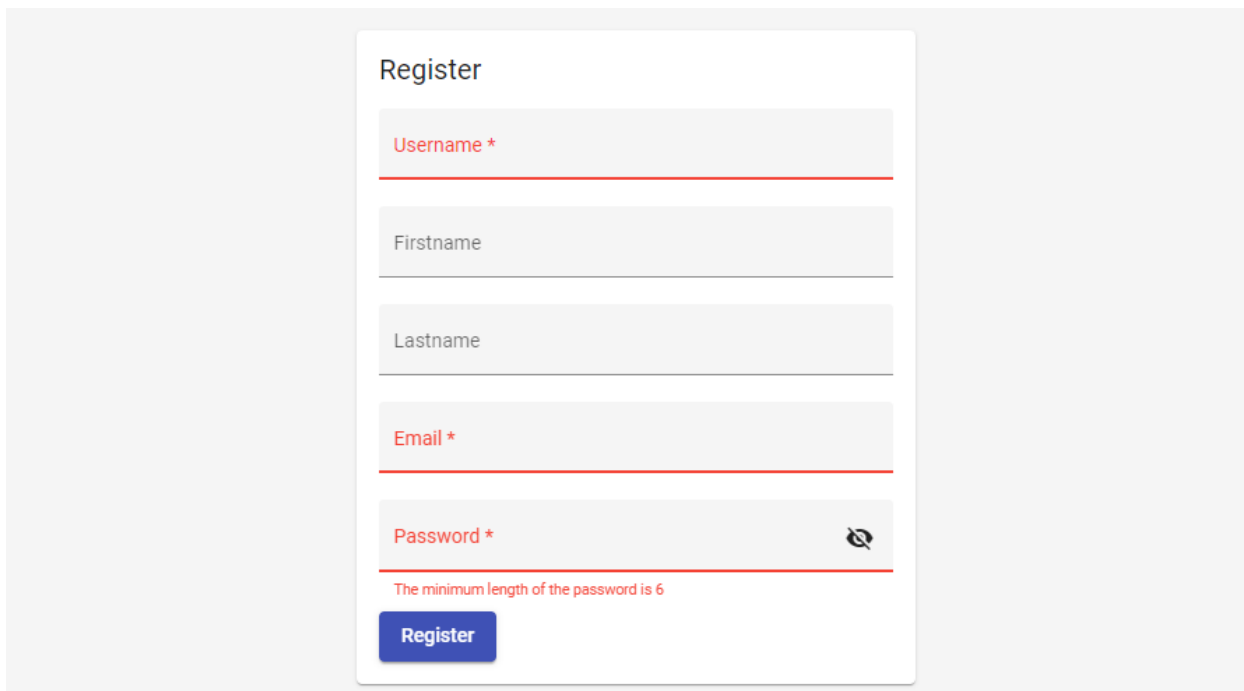
4.2.2. attēls Secību diagramma biļetes izveidošanas funkcijai.

4.2.2 attēlā ir redzama secību diagramma biļetes izveidošanas funkcijai (BL-8). Kā redzams, secību diagrammā, tad sākumā lietotājs ievada nosaukumu, aprakstu, izvēlas lietotājstāstu, statusu, lietotāju, kam piešķirt biļeti, prioritāti, tipu, tad tiek lietotnē tiek pārbaudīts vai ir ievadīts nosaukums, ja nav tad, tiek izvadīta kļūda. Tad, lietotne veic pieprasījumu uz tīmekļa API servisu, kur tiek aizsūtīti dati. Tīmekļa API serviss pārbauda, vai lietotājam ir piekļuve, ja ir tad biļete tiek

saglabāta datubāzē. Un tad no Tīmekļa API serviss tiek sūtīta atbilde uz tīmekļa lietotni, kura aizver biļetes izveidošanas skatu.

4.3. Lietotāja saskarne

4.3.1. Lietotāja saskarnes ekrānuzņēmumi.

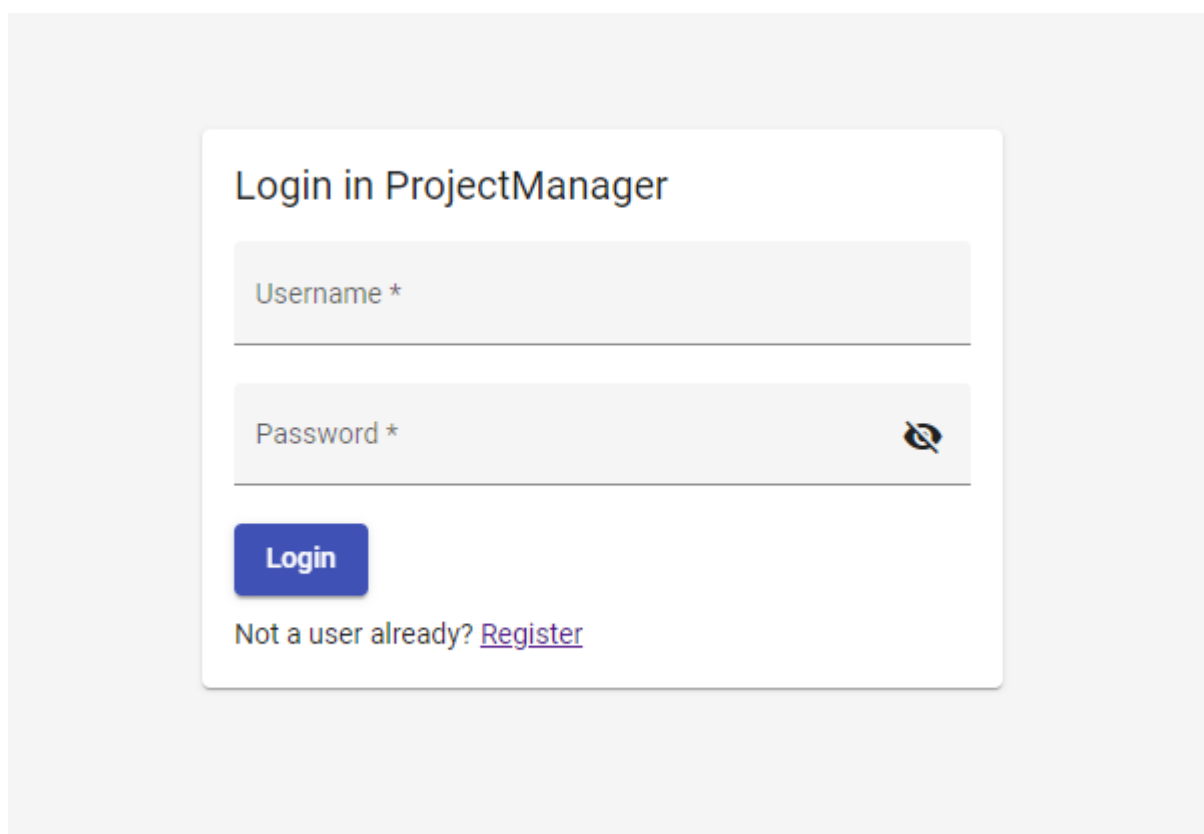


The image shows a registration form with the following fields and elements:

- Register** (Title)
- Username *** (Required field, highlighted with a red border)
- Firstname** (Text input)
- Lastname** (Text input)
- Email *** (Required field, highlighted with a red border)
- Password *** (Required field, highlighted with a red border, includes a toggle icon for visibility)
- The minimum length of the password is 6 (Note below the password field)
- Register** (Submit button)


4.3.1 attēls Lietotnes reģistrācijas skats.

4.3.1. Attēlā var redzēt lietotnes reģistrācijas skatu. Ja, kāds no obligātajiem laukiem netiek. Tad tie iekrāsojas sarkani. Savukārt, ja ir nepareizs e-pasts, tad tas iekrāsojas sarkans un Apakšā tiek parādīta ziņa, ka ir nepareizs e-pasts. Savukārt, ja parolei netiek izpildīti kādi no nosacījumiem, tad, kamēr lietotājs raksta, tad zem paroles lauka var redzēt nosacījumus kādi ir jāizpilda. Uzspiežot uz ikonas, kur attēlota acs var paslēpt vai parādīt paroli.

The image shows a login form titled "Login in ProjectManager". It features two input fields: "Username *" and "Password *". The password field includes a toggle icon for visibility. Below the fields is a blue "Login" button. At the bottom, there is a link that says "Not a user already? [Register](#)".

Login in ProjectManager

Username *

Password * 

Login

Not a user already? [Register](#)

4.3.2. attēls Lietotnes autentifikācijas skats

4.3.3.attēlā var redzēt Lietotnes autentifikācijas skatu. Ja lietotājs neievada visus obligātos laukus tie paliek sarkani. Šajā skatā zem autentifikācijas pogas ir saite uz lietotnes reģistrācijas skatu.

Project Manager Home Komanda nr.1 WebShopSystem Logout

Assigned to me
Board
Roadmap

Project board

This page shows all project work Items

	To do	Active	Resolved	Done
Unparented items	<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>		<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>	<p>Nestrādā saglabāšanas poga</p> <p>Poga nospiežas, bet nekas vairāk nenotiek</p> <p>Test1 User</p> <p>Edit</p>
Backend implementation Implement this project in backend	<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>	<p>Datubāzes izveide</p> <p>Izveidot sistēmai datubāzi</p> <p>Test1 User</p> <p>Edit</p>		
Frontend implementation Implemen this project in frontend			<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>	<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>

4.3.3. attēls Lietotnes projekta paneļa skats(Reģistrēts lietotājs)

4.3.3. attēlā redzams Lietotnes projekta paneļa skats, kur kreisajā lapas malā redzams Lietotājstāsts, kuram pieder attiecīgās biļetes(uzdevumi vai kļūdas). Augšējajā rindā var redzēt tas biļetes kurām nav lietotājstāsta.

Project Manager Home Komanda nr.1 WebShopSystem Logout

Assigned to me

Board

Roadmap

Assigned to me

This page shows all project work Items assigned to user

New bug

To do	Active	Resolved	Done
<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>	<p>Datubāzes izveide</p> <p>Izveidot sistēmai datubāzi</p> <p>Test1 User</p> <p>Edit</p>	<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>	<p>Nestrādā saglabāšanas poga</p> <p>Poga nospiežas, bet nekas vairāk nenotiek</p> <p>Test1 User</p> <p>Edit</p>
<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>		<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>	<p>Testa title</p> <p>Test description which describes the system</p> <p>Test1 User</p> <p>Edit</p>

4.3.4. attēls Lietotnes projektā piešķirto biļešu skats(Reģistrēts lietotājs)

4.3.4. attēls Lietotne Projektā piešķirto biļešu skats, kurā var redzēt Lietotājam piešķirtās biļetes.

Project Manager Home Komanda nr.1 WebShopSystem Logout

Assigned to me

Board

Roadmap

Roadmap

This page shows project userstories information

Name	Description	Unclosed items	Start date	Deadline
Visual studio code extension	Implement this project in backend	2	Jan 6, 2021	Feb 17, 2021
Documentation	Implement this project in backend	0	Jan 6, 2021	Feb 10, 2021
Backend implementation	Implement this project in backend	0	Jan 6, 2021	Feb 3, 2021
Frontend implementation	Implemen this project in frontend	0	Jan 6, 2021	Jan 27, 2021

4.3.5. attēls Lietotnes projekta lietotājstāstu skats(Reģistrēts lietotājs)

4.3.5. attēlā var redzēt lietotnes projekta lietotājstāstu skatu, kurā var redzēt projekta lietotājstāstus, to aprakstu, to cik daudz vēl ir atlikuši uzdevumi, kļūda, kā arī to, kad ir šī lietotājstāsta beigu datums.

Dashboard

This page shows all unclosed tickets assigned to user in selected dashboard sorted by priority

Nestrādā saglabāšanas poga Poga nospiežas, bet nekas vairāk nenotiek

Testa title Test description which describes the system

Testa title Test description which describes the system

Testa title Test description which describes the system

Testa title Test description which describes the system

Testa title Test description which describes the system

Datubāzes izveide Izveidot sistēmai datubāzi

Add new

4.3.6. attēls Lietotnes lietotāja paneļa skats

4.3.6. attēlā ir redzams lietotāja paneļa skats, kur lietotājs var redzēt tos piešķirtos uzdevumus un kļūdas, kuriem nav vēl statuss pabeigts, un kuras vēl ir sakārtotas pēc prioritātes. Kā arī šajā skatā lietotājs var izveidot jaunas biļetes, kā arī mainīt biļešu statusu.

ge shows all project work items

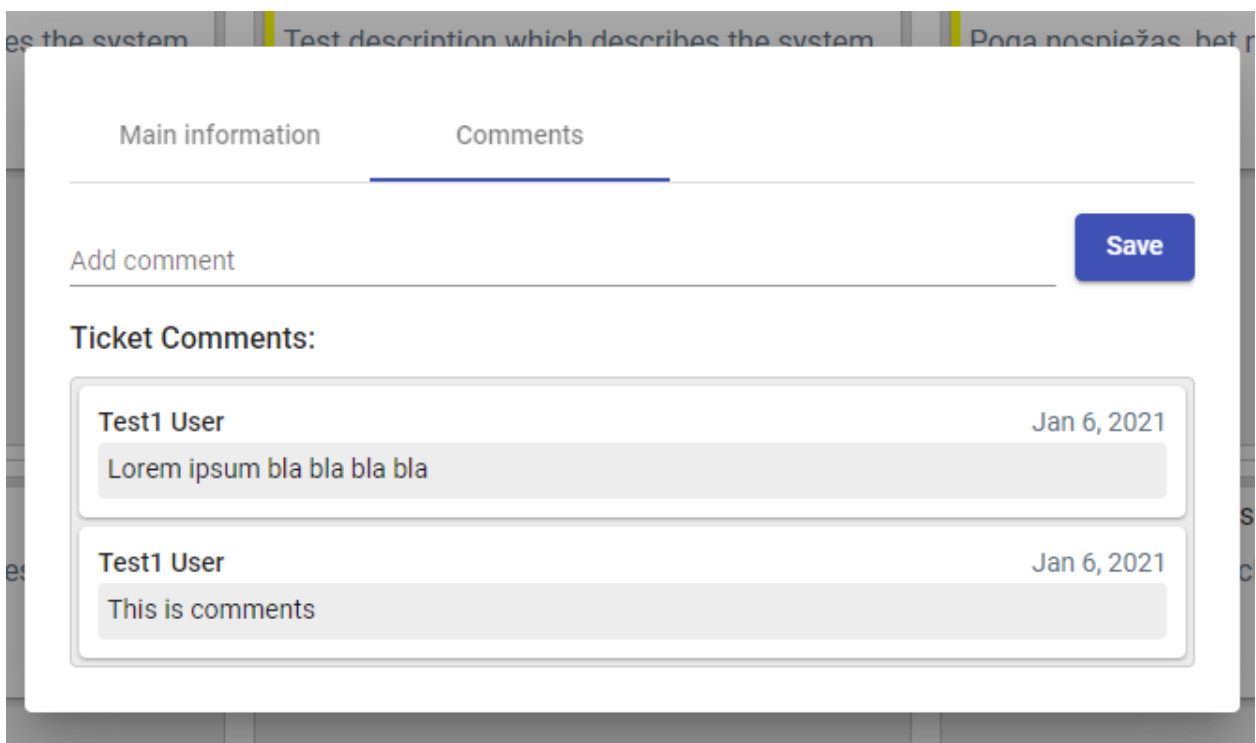
The screenshot shows a 'Main information' tab for editing a ticket. The form includes the following fields:

- Title: Testa title
- Description: Test description which describes the system
- Select Userstory: Visual studio code extension
- Assigned to: Test1 User
- Ticket state: New
- Ticket priority: (empty)
- Ticket type: Task

At the bottom of the form, there are two buttons: 'Save' and a trash icon for deletion.

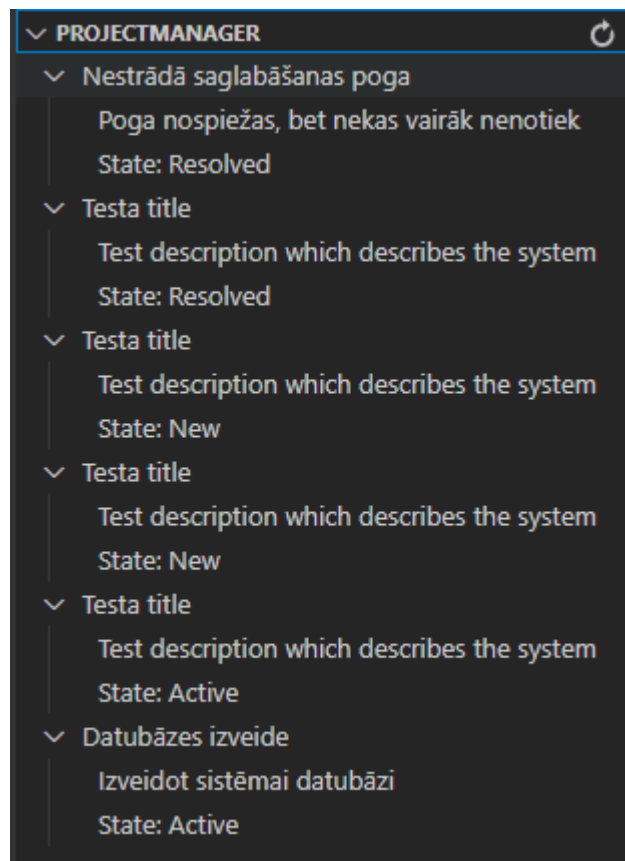
4.3.7. attēls Biļetes rediģēšanas skats

4.3.7. attēlā ir redzams Biļetes rediģēšanas skats, kurā var redzēt, to kā var rediģēt Biļeti. Kā redzamas tajā ir arī ir iespējams rediģēt nosaukumu, apraksts, izmainīt to, kuram šī biļete piešķirta, mainīt statusu, prioritāti, Biļetes tipu uz uzdevumu vai arī uz kļūdu.



4.3.8. attēls Komentāru skats(Reģistrēts Lietotājs)

4.3.8. attēlā iespējams redzēt Komentāru skatu, kurā var pievienot komentāru ievadot komentāru nospiežot pogu “Save”. Kā arī iespējams redzēt visus komentārus un to kurš šos komentārus ir uzrakstījis



4.3.8. attēls Visual Studio Code paplašinājuma skatījums.

4.3.8. attēlā var Visual Studio Code izveidoto paplašinājumu, kur var redzēt un mainīt statusu lietotājam piešķirtajām biļetēm.

5. TESTĒŠANAS DOKUMENTĀCIJA.

5.1. Testēšanas metodika

Sistēma tiek manuāli testēta, izmantojot melnās kastes metodi, tāpēc testēšana ir tieši no lietotāja skatu punkta, lai pārliecinātos, ka lietotājiem tiek atgriezts paredzētais rezultāts. Testēšana tiks veikta balstoties uz programmatūras prasību specifikācijā definētajām funkcijām.

5.2. Testēšanas plāns

Testēšana tiek veikta katrai programmatūras prasību specifikācijas definētajai funkcijai. Rezultāts tiek pārbaudīts, arī datubāzē.

5.3. Moduļu testēšana

4.2.1 Lietotāju moduļa testēšana

Tabulā 5.3.1. ir attēloti testēšanas mērķi un sagaidāmie rezultāti Lietotāju moduļa funkcijām. Savukārt 5.3.2 ir attēlot testēšanas rezultāti.

5.3.1. tabula

Nr.	Testa Mērķis	Funkcija	Sagaidāmais rezultāts
1.	Tiek aizpildīti reģistrācijas lauki un tiek spiest reģistrēties	LM-01	Lietotājs tiek reģistrēts
2.	Tiek aizpildīti lietotājvārds un paroles lauki un tiek ievadīta nepareiza parole.	LM-02	Lietotājs cilvēks netiek autentificēts un tiek izvadīts paziņojums, ka lietotājvārds vai parole ir nepareiza
3.	Lietotājs ir autentificējies uz laiku, kas ir ilgāks	LM-03	Lietotājs paliek autentificējies

	par JWT žetona dzīvesilgumu		
4.	Lietotājs nospiež pogu “Logout”, un piefiksē datubāzes tabulā vai atjaunināšanas žetons ir padarīts par nederīgu un ar lietotni “PostMan[11]” tiek izsaukts žetona atjaunināšanas metode.	LM-04	Atjaunināšanas žetons ir padarīts par nederīgu un lietotnes “PostMan” rezultāts atgriež nesekmīgu rezultātu.

5.3.2 tabula Testēšanas rezultāti

Testa nr.	Testēšanas datums	Rezultāts
1.	02.12.20	+
2.	04.12.20	LM-01 – netiek aizpildīts lietotājvārda lauks
3.	07.12.20	+
4.	10.12.20	LM-02 – netiek autentificēts, bet netiek izvadīts atbilstošs ziņojums
5.	13.12.20	LM-04 – lietotājs atsakās no sistēmas, bet skatoties datubāzē žetons nav atteikts un izmantojot PostMan ir sekmīgs rezultāts.
6.	17.12.20	+

7.	21.12.20	+
8.	24.12.20	+

5.3.2. Organizāciju moduļa testēšanas rezultāti

Tabulā 5.3.3. ir attēloti testēšanas mērķi un sagaidāmie rezultāti Lietotāju moduļa funkcijām.

Savukārt 5.3.4 ir attēlot testēšanas rezultāti.

5.3.3 tabula Organizāciju moduļa testēšanas rezultāti

Nr.	Testa Mērķis	Funkcija	Sagaidāmais rezultāts
1.	Lietotājs atver organizācijas skatīšanās skatu	OR-01	Tiek parādītas organizācijas ar visiem atribūtiem
2.	Administrators izveido organizāciju un to pašu mēģina ar Reģistrētu lietotāju	OR-02	Administrators spēj izveidot organizāciju, savukārt reģistrēts lietotājs nespēj.
3.	Administrators rediģē organizāciju un to pašu mēģina ar Reģistrētu lietotāju	OR-03	Administrators spēj rediģēt organizāciju, savukārt reģistrēts lietotājs nespēj.
4.	Administrators dzēš organizāciju un to pašu mēģina ar Reģistrētu lietotāju	OR-04	Administrators spēj dzēst organizāciju, savukārt reģistrēts lietotājs nespēj.

5.3.2 tabula Testēšanas rezultāti

Testa nr.	Testēšanas datums	Rezultāts
-----------	-------------------	-----------

1.	02.12.20	OR-01 – neparādās visas organizācijas
2.	04.12.20	OR-02 – reģistrēts lietotājs nespēj, izveidot organizāciju, bet tam izdodas, piekļūt izveidošanas skatam
3.	07.12.20	+
4.	10.12.20	+
5.	13.12.20	+
6.	17.12.20	+
7.	21.12.20	+
8.	24.12.20	+

5.3.3. Projektu moduļa testēšanas rezultāti

Tabulā 5.3.6. ir attēloti testēšanas mērķi un sagaidāmie rezultāti Projekta moduļa funkcijām.

Savukārt 5.3.7 ir attēloti testēšanas rezultāti.

5.3.6.tabula Projektu moduļa sagaidāmie rezultāti

Nr.	Testa Mērķis	Funkcija	Sagaidāmais rezultāts
1.	Tiek atvērts projektu skatījums	PR-01	Tiek parādīti organizācijas projekti
2.	Tiek izveidots projekts	PR-02	Projekts tiek izveidots un parādās sarakstā
3.	Tiek rediģēti projekta dati un tiek rediģēti otreiz	PR-03	Dati tiek rediģēti un saglabāti abas reizes

4.	Tiek dzēsti projekta dati	PR-04	Tiek izdzēsti visi projekta dati
----	---------------------------	-------	----------------------------------

5.3.7 tabula Testēšanas rezultāti

Testa nr.	Testēšanas datums	Rezultāts
1.	02.12.20	PR-01 tiek parādīti arī citu organizāciju projekti.
2.	04.12.20	PR-03 Projekta dati tiek rediģēti otrreiz, bet tas neizdodas, jo izmet kļūdu
3.	07.12.20	+
4.	10.12.20	PR-04 Netiek izdzēstas projekta biļetes
5.	13.12.20	+
6.	17.12.20	+
7.	21.12.20	+
8.	24.12.20	+

5.3.4. Biļešu moduļa testēšanas rezultāti

Tabulā 5.3.8. ir attēloti testēšanas mērķi un sagaidāmie rezultāti Biļešu moduļa funkcijām.

Savukārt 5.3.9 ir attēloti testēšanas rezultāti.

5.3.8.tabula Projektu moduļa sagaidāmie rezultāti

Nr.	Testa Mērķis	Funkcija	Sagaidāmais rezultāts
-----	--------------	----------	-----------------------

1.	Tiek atvērta projekta lietotārstāstu skats	BL-01	Tiek parādīts Projekta lietotārstāstu skats
2.	Tiek atvērta lietotājam piešķirto biļešu skats	BL-02	Tiek atvērta lietotājam piešķirto biļešu skats ar biļetēm, kuras piešķirtas lietotājam
3.	Tiek atvērta projekta panelis	BL-03	Tiek atvērta projekta panelis ar visiem projekta lietotārstāstiem un visām biļetēm
4.	Tiek atvērta lietotāja panelis	BL-04	Tiek parādītas visas biļetes, kas piešķirtas organizācijā, kuras nav aizvērtas un kuras ir sakārtotas pēc prioritātes
5.	Tiek izveidots lietotārstāsts no projekta paneļa skata	BL-05	Lietotārstāsts tiek veiksmīgi izveidots
6.	Tiek rediģēts lietotārstāsts no projekta paneļa skata	BL-06	Lietotārstāsts tiek veiksmīgi rediģēts
7.	Tiek dzēsts lietotārstāsts no projekta paneļa skata	BL-07	Lietotārstāsts tiek veiksmīgi dzēsts

8.	Tiek izveidota biļete no projekta paneļa skata	BL-08	Biļete tiek veiksmīgi dzēsta
9.	Tiek rediģēta biļete no lietotāja paneļa	BL-09	Biļete tiek veiksmīgi rediģēta
10.	Tiek dzēsta biļete no lietotāja paneļa	BL-10	Biļete tiek dzēsta no lietotāja paneļa

5.3.7 tabula Testēšanas rezultāti

Testa nr.	Testēšanas datums	Rezultāts
1.	02.12.20	BL-01 Projektu lietotājskatā neparādās neaizvērto biļešu skaits
1.	02.12.20	BL-03 Projektu panelī neparādās biļešu tips.
1.	02.12.20	BL-08 Biļetes visi lauki netiek saglabāti
2.	04.12.20	BL-05 Lietotājstāstam, nesaglabājas Beigu datum lauks
3.	07.12.20	+
4.	10.12.20	BL-10 Biļete netiek izdzēsta
5.	13.12.20	+
6.	17.12.20	+
7.	21.12.20	+
8.	24.12.20	+

5.3.5. Komentāru moduļa testēšanas rezultāti

Tabulā 5.3.10. ir attēloti testēšanas mērķi un sagaidāmie rezultāti Komentāru moduļa funkcijām. Savukārt 5.3.11 ir attēloti testēšanas rezultāti.

5.3.10.tabula Projektu moduļa sagaidāmie rezultāti

Nr.	Testa Mērķis	Funkcija	Sagaidāmais rezultāts
1.	Tiek atvērts biļetes komentāru skats, un izveidots komentārs	KM-01	Tiek parādīti biļetes komentāri
2.	Tiek atvērts biļetes komentāru skats	KM-02	Tiek parādīti biļetes komentāri
3.	Tiek dzēsts komentārs	KM-03	Tiek dzēsts biļetes komentārs

5.3.11. tabula Testēšanas rezultāti

Testa nr.	Testēšanas datums	Rezultāts
1.	02.12.20	KM-01 komentārs nevar tikt izveidots, konsolē kļūda
2.	04.12.20	KM-02 Projekta dati tiek rediģēti otrreiz, bet tas neizdodas, jo izmet kļūdu
3.	07.12.20	+
4.	10.12.20	KM-03 Komentārs neizdzēšas, API kļūda.
5.	13.12.20	+
6.	17.12.20	+

7.	21.12.20	+
8.	24.12.20	+

5.3.6. Paplašinājuma modulis

6. Nr.	Testa Mērķis	Funkcija	Sagaidāmais rezultāts
1.	Tiek aizpildīti lietotājvārds un paroles lauki un tiek ievadīta nepareiza parole.	PP-01	Lietotājs cilvēks netiek autentificēts un tiek izvadīts paziņojums, ka lietotājvārds vai parole ir nepareiza
2.	Tiek atvērts paplašinājums	PP-02	Tiek parādītas biļetes
3.	Biļetes statuss tiek rediģēts	PP-03	Biļetes statuss tiek samainīts datubāzē

5.3.11. tabula Testēšanas rezultāti

Testa nr.	Testēšanas datums	Rezultāts
1.	02.12.20	PP-03 Biļetes statuss nesamainās
2.	04.12.20	+
3.	07.12.20	+
4.	10.12.20	+
5.	13.12.20	+
6.	17.12.20	+
7.	21.12.20	+
8.	24.12.20	+

7. PROJEKTA ORGANIZĀCIJA

Sistēma tika izstrādāta kvalifikācijas darba ietvaros. Kvalifikācijas darba tēmu izdomāja pats autors. Kvalifikācijas darba autors nolēma to izstrādāt pēc ūdenskrituma modeļa. Sākumā tika apgūtas vajadzīgās tehnoloģijas. Projektu Autors apguva programmēšanas valodas C#, TypeScript, kā arī ASP.NET CORE un Angular ietvarus.

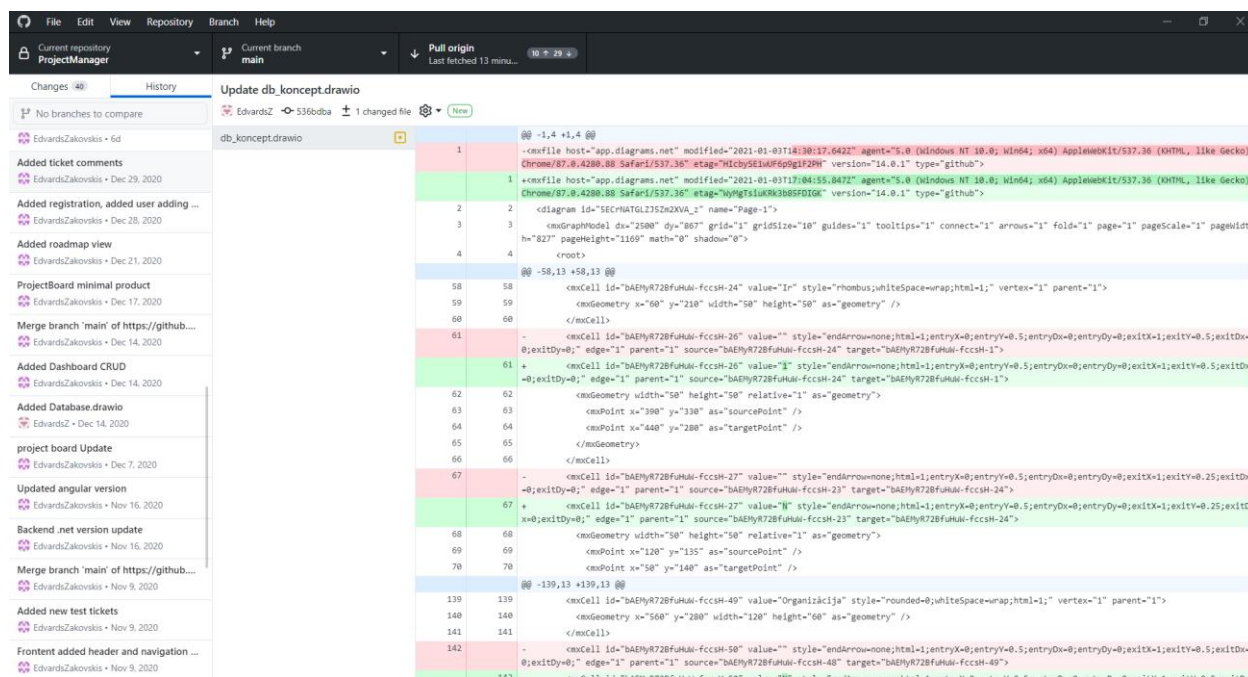
Nākamajā fāzē tika uzsākta programmas prasību specifiskācijas izstrāde un paralēli tika uzsākta programmēšana, pēc katras funkcijas izstrādes un dokumentēšanas, tika veikta šīs funkcijas testēšana un tika veikti nepieciešamie uzlabojumi kodā.

Testēšanas fāzē tika veikti manuālie testi, kas aprakstīti Moduļu testēšanas nodaļā. Ja testi bija neveiksmīgi kods, kļūdas kodā tika novērstas un tika testēts vēlreiz līdz kļūdas nebija. Sistēmas tīmekļa API serviss tika izstrādāts ar C# un ASP.NET CORE 5 ietvaru, kur tika izmantots Visual Studio IDE, savukārt Tīmekļa lietotne tika izstrādātā ar Angular un kur tika izmantots Visual Studio Code, kā izstrādes vide, kā arī Visual Studio Code paplašinājums tika izstrādāts ar TypeScript valodu.

8. KONFIGURĀCIJU PĀRVALDĪBA.

Projekta pirmkoda izstrādes versiju kontrole tiek veikta ar “GitHub”[5], Tas tika darīts, lai varētu redzēt to, kas ir darīts un arī, lai kļūdas rašanās laikā, varētu pāriet uz citām versijām. Versiju kontrole tika izmantota pēc katras svarīgās izmaiņas, vai arī tad, kad bija iekrājušās izmaiņas. Sistēmas dokumentācija tika iekļauta iekš GitHub repozitorija, tādējādi sistēmas dokumentācijā arī tika uzturēta versiju kontrole.

1.1.Attēls. Github Desktop ekrānuzņēmums



2. DARBIETILPĪBAS NOVĒRTĒJUMS

Sistēmas izstrādes darbietilpības aprēķināšanai tika izveidota tabula, kurā ir norādītas ailes optimistiskais, reālistiskais, pesimistiskais, kurā sistēmas izstrāde tika sadalīta pa darbiem.

Sistēmas darbietilpības novērtēšanu veica darba, autors.

8.1 Tabula

	Optimistiski(dienas)	Reālistiski(dienas)	Pesimistiski(dienas)	Vidēji(dienas)
Darba plānošana	2	4	8	4.66667
Lietotāju modulis	7	8	15	10
Organizāciju modulis	2	6	8	5.33333
Projektu modulis	5	8	9	7.33333
Biļešu modulis	16	17	23	18.6667
Komentāru modulis	4	8	7	6.33333
Paplašinājuma modulis	4	7	9	6.66667
Dokumentācija	7	14	19	13.3333
Kopā	47	72	98	72.3333

Pēc tabulas darba apjoma novērtēšanas tika iegūts, optimistiskais apjoms ir 51 cilvēkdienas, bet pesimistiskais 106. Reālistiskais novērtējums tika iegūts darba laikā.

Darbietilpība tika aprēķināta pēc PERT formulas, kur D ir darbietilpība, O ir optimistiskais laiks, R ir reālais laiks un P ir pesimistiskā laika daudzums.

$$D = \frac{O+4R+P}{6} = \frac{47+4*72+98}{6} = 72.2 \text{ dienas}$$

Viens cilvēks mēnesis sastāv no 20 dienām. Kopējais laiks, kas nepieciešams izstrādei ir 3.6 cilvēk mēneši.

3. KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu izstrādājamās sistēmas kvalitāti, programmatūras prasību specifikācija un programmatūras projektējuma apraksts tika veikts pēc vienotiem standartiem, vadlīnijām un prasībām. [6,7,8]

Sistēmas pirmkodam tika izmantoti palīg rīki, piemēram priekš TypeScript valodas tika izmantots ESLINT[9]. Kā arī tika skatītas citas TypeScript bibliotēkas, lai kāda ir laba prakse.

Kvalitātes nodrošināšanas nolūkos tika arī izmantota versiju kontroles sistēma GitHub.

Tika izmantotas izstrādes uzsākšanas brīdī jaunākās gan ASP.NET CORE un Angular versijas.

Testēšana tika veikta ne tikai sistēmas pamatfunkcijām, bet arī visām pērējām, tādējādi tika stabilizēta kopējā sistēmas darbība, izskaustas kļūdas.

SECINĀJUMI

Izstrādātā sistēma ir reāli lietojama sistēma, kaut arī tā tika kvalifikācijas darba ietvaros.

Izstrādes laikā darba autors iemācījās lietot ASP.NET CORE un Angular ietvarus, kurus pirms tam nemācēja. Šis darbs palīdzēja lietotājam iepazīt to kā strādāt gan lietotnes izstrādē, gan kā izstrādāt ASP.NET CORE API. Darba laikā tika labi izprasts tas, kā veidot lietotāju autentifikāciju ar žetoniem. Kā arī tas, kā izveidot JWT žetonu ASP.NET CORE ietvarā.

Darba gaitā, tika gūta atziņa, ka ar GitHub, versiju kontroli ir ļoti viegli strādāt, jo jebkurā brīdī var atgriezties pie vecās versija. Darba gaita tika saprasts arī tas, ka izmantojot lietotnes ietvarus un bibliotēkas var uzlabot darba kvalitāti, jo tiek samazināts darba apjoms.

IZMANTOTĀS LITERATŪRAS SARAKSTS

1. ASP.NET CORE dokumentācija - <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0>
2. Angular dokumentācija - <https://angular.io/docs>
3. ASP.NET CORE Identity dokumentācija - <https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-5.0&tabs=visual-studio>
4. JWT žetons - <https://tools.ietf.org/html/rfc7519>
5. GitHub - <https://github.com>
6. Rudīte Čevere, LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis”, 1996. gads, 22. lpp.
7. Rudīte Čevere, LVS 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”, 1996. gads, 13. lpp.
8. Rudīte Čevere, LVS 70:1996 „Programmatūras testēšanas dokumentācija”, 1996. gads, 17. lpp.
9. ESLINT - <https://eslint.org>
10. Latvijas Universitātes rīkojums Nr. 1/38 “Prasības noslēguma darbu izstrādāšanai un aizstāvēšanai Latvijas Universitātē”.
11. PostMan - <https://www.postman.com>
12. Atjaunināšanas žetoni - <https://jasonwatmore.com/post/2020/05/25/aspnet-core-3-api-jwt-authentication-with-refresh-tokens>
13. “Jira” - <https://www.atlassian.com/software/jira>
14. “Azure DevOps Server” - <https://azure.microsoft.com/en-us/services/devops/server/>

PIELIKUMI

Pielikumā attēlotas galvenās sistēmas nozīmīgākās funkcijas no sistēmas, gan no Lietotnes puses, gan arī no tīmekļa API puses, un Visual Studio Code paplašinājuma kods

```
drop(event: CdkDragDrop<string[]>) {
  if (event.previousContainer === event.container) {
    moveItemInArray(
      event.container.data,
      event.previousIndex,
      event.currentIndex
    );
  } else {
    transferArrayItem(
      event.previousContainer.data,
      event.container.data,
      event.previousIndex,
      event.currentIndex
    );
  }
  const id = event.container.data[event.currentIndex]['id']
  if(event.container.id === "no-userstory-new") {
    const item = this.ticketsNoUserstory.new[event.currentIndex];
    item.state = 'New'
    item.userstoryId = null;
    item.userstory = null;
    this.ticketService.updateTicket(item).subscribe();
    return;
  }
  if(event.container.id === "no-userstory-active") {
    const item = this.ticketsNoUserstory.active[event.currentIndex];
    item.state = 'Active'
    item.userstoryId = null;
    item.userstory = null;
    this.ticketService.updateTicket(item).subscribe();
    return;
  }
  if(event.container.id === "no-userstory-resolved") {
    const item = this.ticketsNoUserstory.resolved[event.currentIndex];
    item.state = 'Resolved'
    item.userstoryId = null;
    item.userstory = null;
    this.ticketService.updateTicket(item).subscribe();
    return;
  }
  if(event.container.id === "no-userstory-closed") {
    const item = this.ticketsNoUserstory.closed[event.currentIndex];
    item.state = 'Closed'
    item.userstoryId = null;
    item.userstory = null;
    this.ticketService.updateTicket(item).subscribe();
    return;
  }
  const userstoryId = event.container.id.substr(0, event.container.id.indexOf('-'));
  const item = this.userstories.find(u => u.userstory.id === userstoryId);
  const state = event.container.id.substr(event.container.id.indexOf('-')+1, event.container.id.length);
  console.log(userstoryId ,state, item[state.toLowerCase()][event.currentIndex]);
  item[state.toLowerCase()][event.currentIndex].state = state;
  item[state.toLowerCase()][event.currentIndex].userstoryId = userstoryId;
  this.ticketService.updateTicket(item[state.toLowerCase()][event.currentIndex]).subscribe();
}
```

14.1 attēls Sistēmas tīmekļa lietotnes koda daļā

12.1 attēlā var redzēt tīmekļa lietotnes kodu, kurš tiek izsaukts, kad saskarsnē biļete tiek pārvilkta uz citu stāvokli

```
    this.edit = data.edit;
    this.ticket = data.ticket;
    this.userstoryService.getAllProjectUserstories(this.ticket.projectId).subscribe( res => {
      this.allUserstories = res;
    });
    this.userService.getAll().subscribe( res => {
      this.users = res;
    })
    this.ticketFormGroup = this.createTicketGroup(this.ticket);
  }
  onNoClick(): void {
    this.dialogRef.close();
  }

  ngOnInit(): void {
  }

  createTicketGroup(ticket: Ticket): FormGroup { // creates form group for ticket
    console.log(ticket);
    return new FormGroup({
      title: new FormControl(ticket.title),
      description: new FormControl(ticket.description),
      userstoryId: new FormControl(ticket.userstoryId),
      assignedToId: new FormControl(ticket.assignedToId),
      state: new FormControl(ticket.state),
      priority: new FormControl(ticket.priority?.toString()),
      type: new FormControl(ticket.type)
    });
  }

  save(){
    const formData = this.ticketFormGroup.value; // ticket data is saved;
    this.ticket.description = formData.description;
    this.ticket.title = formData.title;
    this.ticket.userstoryId = formData.userstoryId;
    this.ticket.assignedToId = formData.assignedToId;
    this.ticket.priority = formData.priority;
    this.ticket.type = formData.type;
    this.ticket.state = formData.state;
    this.loading = true;
    if (this.edit) { // dat is being edited
      this.ticketService.updateTicket(this.ticket).subscribe( res => { this.loading = false; this.dialogRef.close(); } );
    } else { // data is being created
      this.ticketService.newTicket(this.ticket).subscribe( res => { this.loading = false; this.dialogRef.close(); } );
    }
  }

  delete() { // deletes the ticket
    const formData = this.ticketFormGroup.value;
    this.ticket.description = formData.description;
    this.ticket.title = formData.title;
    this.loading = true;
    this.ticketService.deleteTicket(this.ticket).subscribe( res => { this.loading = false; this.dialogRef.close(); } );
  }
}
```

14.2 attēls Sistēmas tīmekļa lietotnes koda daļa

```

constructor() {
  setTimeout(() => {this.ask(); }, 3000);// after extension is loaded after 3 seconds ask the user to input password
}

setItem(node: TreeItem, state: string) {
  const ticket = node.ticket;
  this.updateTicket(ticket, state);
}

async updateTicket(ticket: Ticket, state: string) { // updates the ticket in database
  ticket.state = state;
  const headers = {
    'Content-Type': 'application/json',
    'Authorization': 'Bearer ' + this.jwtToken
  };
  axios.defaults.baseURL = "http://localhost:59921";
  const result = await axios.put(`/api/Tickets/${ticket.id}`, ticket, { headers: headers});
  this.getData();
}

getTreeItem(element: TreeItem): vscode.TreeItem | Thenable<vscode.TreeItem> {
  return element;
}

getChildren(element?: TreeItem | undefined): vscode.ProviderResult<TreeItem[]> {
  if (element === undefined) {
    return this.data;
  }
  return element.children;
}

setData(tickets: Ticket[]) { // fetches the data from database
  this.data = [];
  for(const ticket of tickets) {
    const state = new TreeItem("State: " + ticket.state, undefined, ticket);
    const child = new TreeItem(ticket.description, undefined, ticket);
    const item = new TreeItem(ticket.title, [child, state], ticket);
    this.data.push(item);
  }
  this._onDidChangeTreeData.fire();
}

public async ask() {
  const options: vscode.InputBoxOptions = {
    placeholder: "Username",
    prompt: "Input username and then press enter"
  };
  vscode.window.showInputBox(options).then( username => {
    const options: vscode.InputBoxOptions = {
      placeholder: "Password",
      prompt: "Input password and then press enter",
      password: true
    };
    vscode.window.showInputBox(options).then( password => {
      if(username && password) {
        this.username = username;
        this.password = password;
        this.post();
      }
    });
  });
}

```

12.3.attēls Sistēmas Visual Studio Code paplašinājuma koda daļa

```

[HttpGet("dashboardTicketsAssignedToMe")]
0 references | EdvardsZ, 15 hours ago | 2 authors, 2 changes
public async Task<ActionResult<IEnumerable<Ticket>>> GetDashboardTicketsAssignedToMe(long dashboardId)
{
    if (!this.HaveAccessToDashboard(dashboardId))
    {
        return Unauthorized(new { message = "No access" });
    }
    return await _context.Ticket
        .Where( t => t.Project.Dashboard.Id == dashboardId && t.AssignedToId == User.GetId() && t.State != "Closed")
        .OrderBy(t => t.Priority == null).ThenBy(t=>t.Priority).ToListAsync();
}

[HttpGet("projectTickets")]
0 references | EdvardsZakovskis, 5 days ago | 1 author, 3 changes
public async Task<ActionResult<IEnumerable<Ticket>>> GetProjectTicekts(long projectId)
{
    var project = await _context.Project.FindAsync(projectId);
    var dashboardId = project.DashboardId;
    if (!this.HaveAccessToDashboard(dashboardId))
    {
        return Unauthorized(new { message = "No access" });
    }
    return await _context.Ticket
        .Where( t => t.ProjectId == projectId).Include( t => t.Userstory)
        .Include(t=> t.AssignedTo).ToListAsync();
}

[HttpGet("projectTicketsAssignedToMe")]
0 references | EdvardsZ, 15 hours ago | 2 authors, 3 changes
public async Task<ActionResult<IEnumerable<Ticket>>> GetProjectTicektsAsssignedToMe(long projectId)
{
    var project = await _context.Project.FindAsync(projectId);
    if(project == null)
    {
        return BadRequest();
    }
    var dashboardId = project.DashboardId;
    if (!this.HaveAccessToDashboard(dashboardId))
    {
        return Unauthorized(new { message = "No access" });
    }
    return await _context.Ticket
        .Where(t => t.ProjectId == projectId && t.AssignedToId == User.GetId())
        .Include(t => t.Userstory).Include(t => t.AssignedTo).ToListAsync();
}

[HttpGet("projectTicketsAndUserstories")]
0 references | EdvardsZakovskis, 4 days ago | 1 author, 3 changes
public async Task<ActionResult<ProjectTickets>> GetProjectTicektsAndUserstories(long projectId)
{
    var project = await _context.Project.FindAsync(projectId);
}

```

12.4.attēls Sistēmas tīmekļa API servisa koda daļa

Kvalifikācijas darbs „*Programmatūras izstrādes pārvaldības sistēma*” 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: *Edvards Zakovskis* _____ .01.2021.

Rekomendēju darbu aizstāvēšanai

Darba vadītāja: *Dr.sc.comp. Elīna Kalniņa* _____ .01.2021.

Recenzents: *Pēteris Krastiņš*

Darbs iesniegts 11.01.2021.

Kvalifikācijas darbu pārbaudījumu komisijas sekretāre: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.01.2021. prot. Nr. _____

Komisijas sekretārs(-e): _____