

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**E-pasta pakalpojumu nodrošināšanas platformas
“cartero.io” prototipa izstrāde
KVALIFIKĀCIJAS DARBS**

Autors:

Georgijs Pliska – gp17023

Darba vadītāja:

Dr.dat. Reinholds Zviedris

Rīga, 2019

ANOTĀCIJA

Kvalifikācijas darba mērķis ir aprakstīt e-pasta pakalpojumu nodrošināšanas platformas “cartero.io” prototipa izstrādi.

Prototipa mērķis ir parādīt kā var izskatīties un funkcionēt sistēma, kura ļautu jebkurai personai, ar vai bez tehniskām zināšanām, ar dažiem klikšķiem izveidot sev privātu serveri, uz kura tiek automātiski uzstādīta e-pasta servera programmatūra ar administratīvo vidi, kas ļauj rediģēt un konfigurēt e-pasta piegādes sistēmu.

Sistēma sastāv no 3 moduļiem – lietotāju, finanšu un serveru.

Atslēgvārdi: Go, prototips, e-pasts, Vue.js.

ABSCTRACT

Development of E-Mail service platform “cartero.io” prototype

The aim of the qualification work is to describe the development of the prototype of the e-mail service platform "cartero.io".

The purpose of the prototype is to show how to make a system that would allow any person with and without technical knowledge to create their own private server on which the e-mail server software with administrative interface is installed and allows one to edit and configure e-mail delivery system.

The system consists of 3 modules - user, financial and server.

Keywords: Go, prototype, e-mail, Vue.js.

SATURS

SATURS	4
DEFINĪCIJAS, AKRONĪMI UN SAĪSINĀJUMI	6
IEVADS	7
Nolūks	7
Darbības sfēra	7
Saisība ar citiem dokumentiem	7
Pārskats	7
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	8
1.1. Produkta perspektīva	8
1.2. Produkta funkcijas	8
1.3. Lietotāja raksturiezīmes	9
1.4. Vispārējie ierobežojumi	9
1.5. Pieņēmumi un atkarības	9
1.6. Nefunkcionālās prasības	10
1.6.1. Veiktspējas prasības	10
1.6.2. Drošība	10
1.6.3. Uzturamība	10
1.6.4. Izmantojamība	10
1.7. Funkcionālās prasības	10
1.7.1. Lietotāja modulis	12
1.7.2. Serveru modulis	19
1.7.3. Finanšu modulis	22
2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	25
2.1. Datubāzes projektējums	25
2.2. Konceptuālais modelis	25
2.3. Loģiskais modelis	25
2.4. Fiziskais modelis	26
2.5. Datubāzes tabulu skati	27
2.5.1. Tabula “Lietotajs”	27
2.5.2. Tabula “Pasūtījums”	27
2.5.3. Tabula “Rēķins”	28
2.5.4. Tabula “Pakalpojums”	28
2.5.5. Tabula “Pasūtījums/Pakalpojums”	28
2.5.6. Tabula “Pasūtījums/Rēķins”	29

3. LIETOTĀJU SASKARNES PROJEKTĒJUMS	30
3.1. Sākumlapā	30
3.2. Lietotāju autentificēšanās un pievienošanās lapas	31
3.3. Lietotāja galvenais ekrāns	33
3.4. Lietotāja profils	34
4. TESTĒŠANA	36
4.1. Lietotāju moduļa funkciju testēšana	36
4.2. Finanšu moduļa funkciju testēšana	37
4.3. Serveru moduļa funkciju testēšana	37
5. PROJEKTA PĀRVALDĪBA	39
5.1. Projekta organizācija	39
5.2. Kvalitātes nodrošināšana	39
5.3. Konfigurāciju pārvaldība	39
5.4. Darbietilpības novērtēšana	40
NOBEIGUMS	41
IZMANTOTIE AVOTI	42
PIELIKUMI	43

DEFINĪCIJAS, AKRONĪMI UN SAĪSINĀJUMI

PPS - Programmatūras prasību specifikācija

PPA - Programmatūras projektējuma apraksts

Git - Versiju kontroles pārvaldības sistēma

Gitlab - Repozitoriju glabātuve

Vue.js - JavaScript valodā izveidots ietvars tīmekļa lietotnēm (<https://vuejs.org/>)

Vuetify - Uz "Vue.js" ietvara bāzes izveidots papildinājums tīmekļa lietotņu izstrādei (<https://vuetifyjs.com>)

Echo - Go valodā izveidots ietvars servera lietotnēm (<https://echo.labstack.com/>)

Docker - tehnoloģija kurā izmanto operētājsistēmas līmeņa virtualizāciju, un nodrošina programmatūras izstrādi un piegādi standartizētu programmatūras pakotņu ietvaros, ko sauc par konteineriem (<https://www.docker.com/>)

IEVADS

Nolūks

Kvalifikācijas darbs ir izveidots ar mērķi aprakstīt E-pastu platformas prototipa izstrādi, kas tika izveidota šī darba ietvaros.

Kvalifikācijas darbs sastāv no programmatūras prasību specifikācijas, programmatūras prasību apraksta un testēšanas dokumentācijas.

Darbības sfēra

Izstrādātais E-pastu platformas prototips paredzēts lietošanai jebkurai personai bez tehniskām zināšanām, pieļaujot e-pastu servera izveidi. Prototipā ir iespējams redzēt un mainīt datus par nomātajiem serveriem, veidot maksājumus par serveriem un rediģēt savu profilu.

Saistība ar citiem dokumentiem

Dokumenta izveidē un noformēšanā tiek ievērotas standarta LV 68:1996 “Programmatūras prasību specifikācijas ceļvedis” [1] un LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai” [2] rekomendācijas un prasības.

Pārskats

Kvalifikācijas darbā ietilpst programmatūras prasību specifikācija (PPS), programmatūras projektējuma apraksts (PPA), testēšanas dokumentācija, darbietilpības aprēķins un izstrādātās sistēmas koda fragmenti. Darbs iekļauj 28 tabulu, 18 attēlus

1. daļa – Definīcijas.
2. daļa – Ievads.
3. daļa – Programmatūras prasību specifikācija.
4. daļa – Programmatūras projektējuma apraksts.
5. daļa – Testēšanas dokumentācija.
6. daļa – Projekta pārvaldība.
7. daļa – Nobeigums.
8. daļa – Izmantotie avoti.
9. daļa – Pielikumi.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Produkta perspektīva

E-pastu platformas prototips tika veidots ar mērķi izveidot minimāli lietojamu produktu. Tiek plānots, ka pilna sistēma dos iespēju jebkuram interesantam vienas sistēmas ietvaros izveidot, nokonfigurēt un administrēt e-pastu serveri bez tehniskām zināšanām. Sistēma ļauj ātri un ērti redzēt datus par visiem serveriem un ērti veikt visus maksājumus par serveriem.

Sistēma tika veidota tā, lai nākotnē būtu iespējams pievienot papildus funkcionalitāti pēc nepieciešamības.

1.2. Produkta funkcijas

Platformas galvenās funkcijas ir serveru izveidošana un uzturēšanā, cenas aprēķināšana un maksājumu pieprasīšana un nodrošināšana, lietotāju glābšana.

Tabulā 1.1 ir aprakstīti visi moduļi, kādi būs sistēmai:

1.1. tabula Sistēmas funkcijas sadalītas moduļos

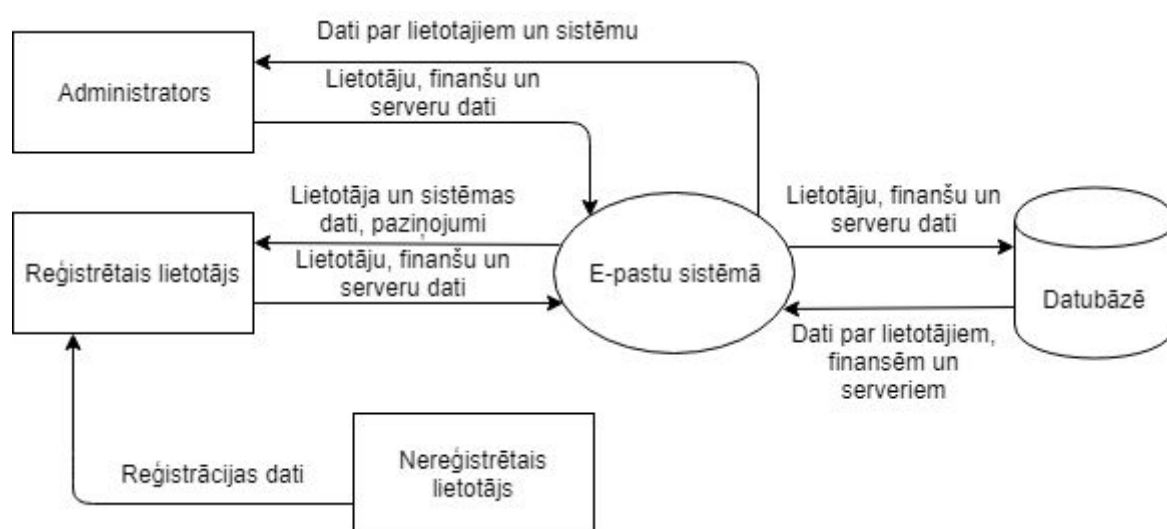
Modulis	Funkcija	Identifikators
Lietotāja	Profila skatīšana	LM-01
	Paroles atjaunošana	LM-02
	Autentificēšanās	LM-03
	Izrakstīšanās	LM-04
	Reģistrācija	LM-05
	Profila dzēšana	LM-06
	E-pastu atjaunošana	LM-07
Serveru	Servera pievienošana	SM-01
	Servera rediģēšana	SM-02
	Servera dzēšana	SM-03
	Servera skatīšana	SM-04
Finanšu	Rēķinu apskatīšana	FM-01
	Rēķinu apmaksa	FM-02

1.3. Lietotāja raksturiezīmes

Lai lietotājs varētu izmantot sistēmu, tam nepieciešams - pamatzināšanas darbā ar interneta pārlūkprogrammām, dators un pieeja internetam. Lietotāji tiešā veidā izmanto tīmekļa lietotni caur datorā uzstādītu pārlūkprogrammu. Izstrādājamās sistēmas kontekstā lietotāji tiek iedalīti autorizētos un neautorizētos lietotājos.

Lai lietotājs varētu izmantot sistēmu, tam jābūt reģistrētam lietotājam. Neregistrēts lietotājs var kļūt par sistēmas lietotāju, aizpildot reģistrācijas anketu. Sistēmai ir administrators, kurš redz visus datus un tiek pievienots sistēmai manuāli.

Attēlā 1.1 attēloti sistēmas moduļi, to mijiedarbība ar datubāzi.



Attēls 1.1. Prototipa 0. līmeņa DPD

1.4. Vispārējie ierobežojumi

Lai lietotājs varētu izmantot aprakstīto sistēmu, lietotājam ir nepieciešams dators, telefons vai planšetdators, kuram ir piekļuve internetam un ir uzstādīta kāda no pārlūkprogrammām, kas spēj sadarboties ar sistēmu.

1.5. Pieņēmumi un atkarības

Pašreizējā sistēma ir tikai prototips, kurai nav pieejams pilnais sistēmas funkcionālais spektrs. Sistēma jau ir pieejama internetā, bet uz doto brīdi tiek izmantota testēšanas nolūkiem. Nākotnē ir plānots sistēmu paplašināt ar pilno funkcionālo daļu un uzsākt testēšanu ar reāliem lietotājiem.

1.6. Nefunkcionālās prasības

1.6.1. Veiktspējas prasības

Tiek paredzēts, ka ikdienā prototipu var izmantot vismaz 20 cilvēki. Sistēmas darbības izpildes laikam jābūt ne vairāk kā 5 sekundes.

1.6.2. Drošība

Sistēmai var un drīkst piekļūt tikai reģistrēti lietotāji. Visas paroles tiek šifrētas. Izvietojot sistēmu internetā, tai ir jāizmanto HTTPS protokols un SSL šifrēšana, lai nodrošinātu lielāku datu apmaiņas drošību.

1.6.3. Uzturamība

Sistēma tika izstrādāta, lai nākotnē to varētu papildināt ar jaunu funkcionalitāti un nepieciešamības gadījumā mainīt esošo.

Sistēma izstrādātā ar konteinerizācijas pieeju, lai katru sistēmas servisu varētu darbināt atsevišķā konteinerī, kas atvieglotu sistēmas konfigurēšanu, atjaunošanu, uzlabošanu un paplašināšanu.

Visiem sistēmas konteineriem ir jāstrādā aiz reversā starpniekservera.

1.6.4. Izmantojamība

Sistēma ir jāstrādā un jābūt pieejamai visu laiku. Vienīgais izņēmums, kad sistēma var būt nepieejama ir sistēmas atjaunošanas gadījumā. Sistēmu jāvar ērti izmantot no visām ierīcēm - datora, planšetdatora vai mobilā telefona.

1.7. Funkcionālās prasības

Visas aprakstītās funkcijas ir pieejamas tikai reģistrētiem lietotājiem. Šajā nodaļā apkopota informācija par sistēmas moduļiem, to funkcijām un ieviesto datu validāciju, kļūdu un brīdinājuma paziņojumiem.

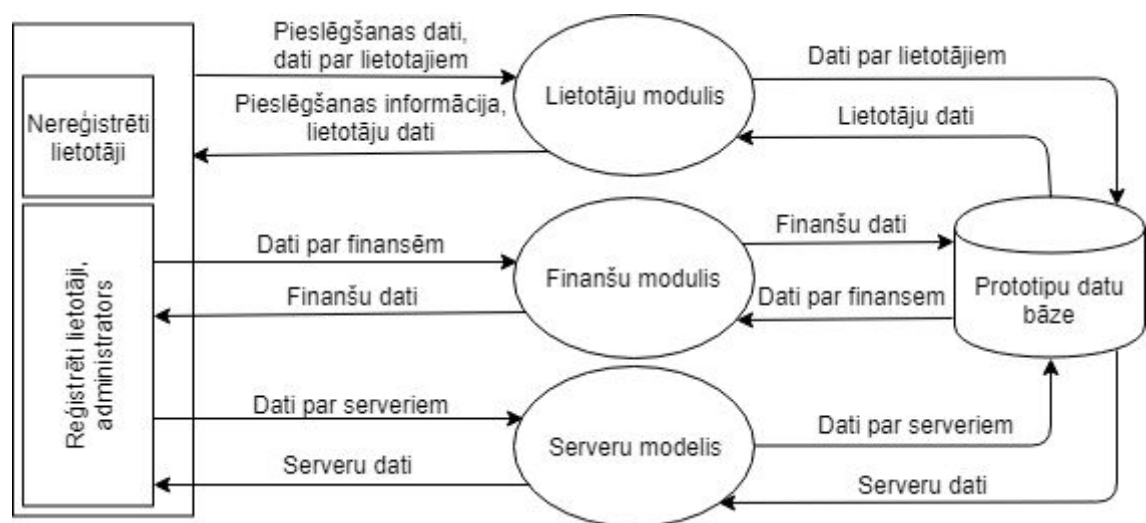
Tabulā 1.2 tiek aprakstīti visi iespējamie sistēmas paziņojumi.

1.2. tabula Sistēmas paziņojumu saraksts

Paziņojuma numurs	Paziņojuma tekst
1	”{Lauka Nosaukums} is required”
2	“E-mail must be valid”

3	“Password must be greater than 6 characters and shorter than 50”
4	“Passwords don’t match!”
5	“Email or Password is wrong”
6	“Name must be greater than 1 character and shorter than 250”
7	“Surname must be greater than 1 character and shorter than 250”
8	“User with this email already exists”
9	“Domain must be greater than 1 character and shorter than 64”

Attēlā 1.2 attēloti sistēmas moduļi, to mijiedarbība ar datubāzi.



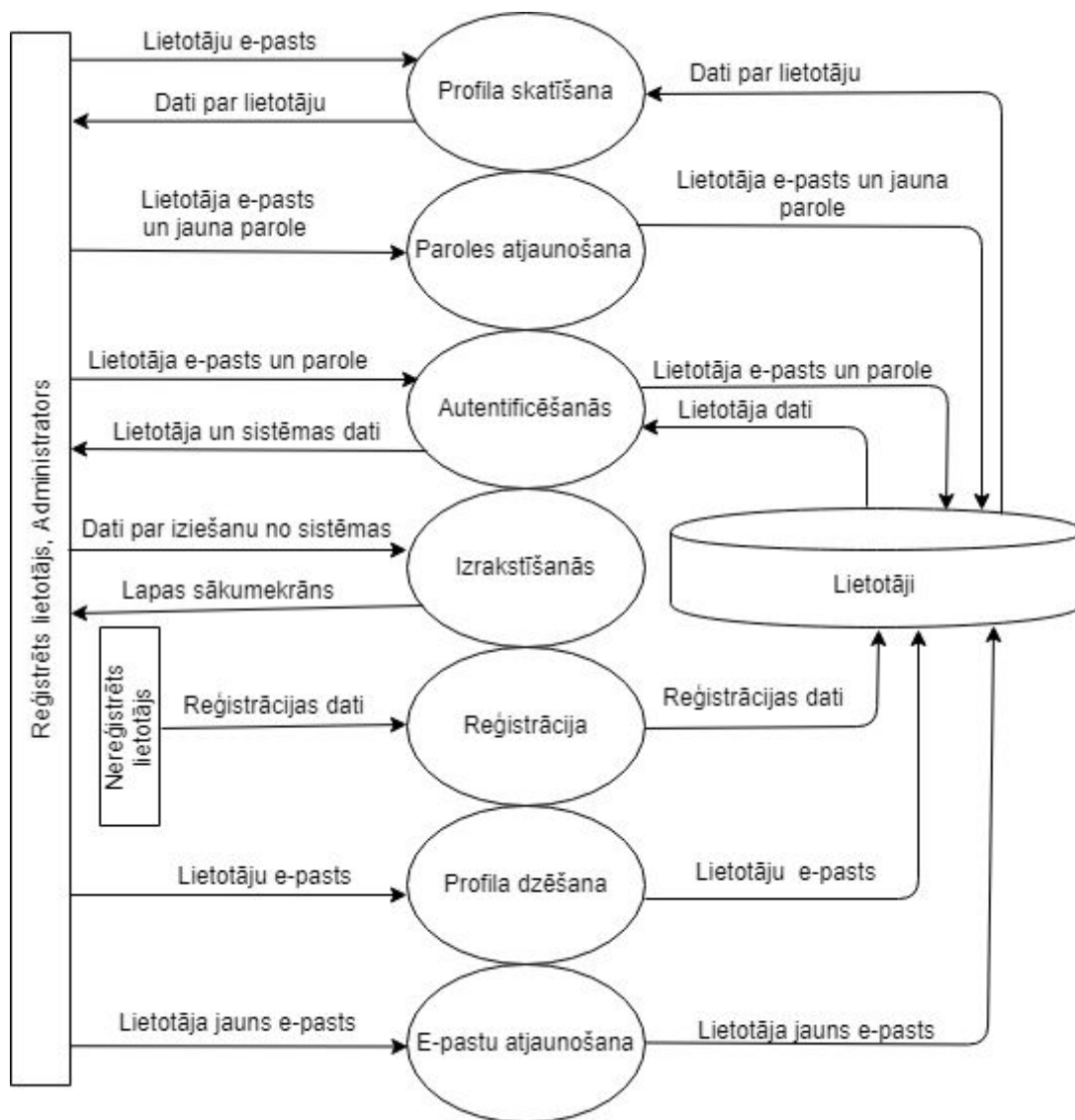
Attēls 1.2. Prototipā 1. līmeņa DPD

Prototips sastāv no 3 moduļiem, kuri nodrošinās tā darbību:

- lietotāju modulī glabāsies visa informācija par lietotājiem;
- serveru modulis nodrošinās visas darbības ar serveriem, t.i. ātri un precīzi iegūt nepieciešamo informāciju par serveru statusu un pieejamību, kā arī jaunu serveru izveidi vai esošo likvidēšanu;
- finanšu modulis piedāvā iespēju veikt visas darbības, kas saistītas ar norēķiniem.

1.7.1 Lietotāja modulis

Lietotāju moduļa mērķis ir apstrādāt un koordinēt sistēmas lietotājus. Attēlā 1.3. attēlots lietotāja modulis un tā mijiedarbība ar datubāzi.



Attēls 1.3. 2. līmeņa datu plūsmu diagramma lietotāju moduļim

1.7.1.1. Profila skatīšana

Tabulā 1.3 tiek aprakstīta profila apskatīšanās funkcija.

1.3. tabula Profila skatīšanās funkcijas apraksts

Identifikators	LM-01
-----------------------	-------

Nosaukums	Profila skatīšana
Apraksts	
Ir iespējams apskatīt savu profilu.	
Lietotāju grupa	
Reģistrēts lietotājs, Administrators	
Ievade	
Poga "PROFILE"	
Apstrāde	
-	
Izvade	
Parāda lietotāja profilu un saistīto informāciju	

1.7.1.2. Paroles atjaunošana

Tabulā 1.4 tiek aprakstīta paroles atjaunošanas funkcija.

1.4. tabula Paroles atjaunošana funkcijas apraksts

Identifikators	LM-02
Nosaukums	Paroles mainīšana
Apraksts	
Ir iespējams mainīt paroli.	
Lietotāju grupa	
Reģistrēts lietotājs, Administrators	
Ievade	

1. Vecā parole. Simbolu virkne garumā no 6 līdz 50 simboliem.
2. Jauna parole. Simbolu virkne garumā no 6 līdz 50 simboliem.
3. Jaunās paroles atkārtojums. Simbolu virkne garumā no 6 līdz 50 simboliem. Jāsakrīt ar vērtību, kura ievadīta laukā “jauna parole”.

Apstrāde

Ja nav aizpildīts kāds no laukiem, parādās paziņojums nr. 1

Ja laukā “Jaunā parole”, “Vecā parole” vai “Jaunās paroles atkārtojums” ir nenasniegts vai pārsniegts atļautais simbolu skaits, parādās paziņojums nr. 4

Ja lauks “Jaunās paroles atkārtojums” nesakrīt ar laukā “Jaunā parole” ievadītājiem datiem, parādās paziņojums nr. 3

Izvade

Ja darbības laikā rodas kļūda, lietotājam tā jānovērš un atkārtoti jāievada dati un jānospiež “UPDATE PASSWORD”

Ja netiek konstatētas kļūdas, lietotājam parādās sistēmas galvenā lapa.

Jaunā parole tiek ierakstīta datubāzē.

1.7.1.3. Autentificēšanās

Tabulā 1.5 tiek aprakstīta autentificēšanās funkcija.

1.5 tabula Autentificēšanās funkcijas apraksts

Identifikators	LM-03
Nosaukums	Autentificēšanās
Apraksts	
Ir iespēja ieiet sistēmā	
Lietotāju grupa	
Reģistrēts lietotājs, Administrators	
Ievade	

1. E-pasts. Simbolu virkne no 6 līdz 50 simboliem. Jāatbilst e-pasta formātam. Unikāls.
2. Parole. Simbolu virkne garumā no 6 līdz 50 simboliem.

Apstrāde

Ja nav aizpildīts kāds no laukiem, parādās paziņojums nr. 1

Ja laukā “E-pasts” ir pārsniegts atļautais simbolu skaits vai ievadītie dati neatbilst e-pasta formātam, parādās paziņojums nr. 2

Ja laukā “Parole” ir nesasniegts vai pārsniegts atļautais simbolu skaits, parādās paziņojums nr. 3

Ja e-pasts sistēmā neeksistē, vai nepareiza parole, parādās paziņojums nr.5

Izvade

Ja darbības laikā rodas kļūda, lietotājam tā jānovērš un atkārtoti jāievada dati un jānospiež “SIGN IN”.

Ja netiek konstatētas kļūdas, lietotājs nokļūst sistēmas galvenajā lapā.

1.7.1.4. Izrakstīšanās

Tabulā 2.6 tiek aprakstīta izrakstīšanās funkcija.

1.6. tabula Izrakstīšanās funkcijas apraksts

Identifikators	LM-04
Nosaukums	Izrakstīšanās
Apraksts	
Ir iespējams iziet no sistēmas	
Lietotāju grupa	
Reģistrēts lietotājs, Administrators	
Ievade	
Poga “LOGOUT”	

Apstrāde
-
Izvade
Lietotājam tiek attēlota sistēmas sākuklapa

1.7.1.5. Reģistrācija

Tabulā 1.7 tiek aprakstīta reģistrācijas funkcija.

1.7. tabula **Reģistrācijas funkcijas apraksts**

Identifikators	LM-05
Nosaukums	Reģistrācija
Apraksts	
Funkcija paredzēta, lai lietotājs tiek reģistrēts sistēmā un spētu to izmantot	
Lietotāju grupa	
Neregistrēts lietotājs	
Ievade	
<ol style="list-style-type: none"> 1. Vārds. Simbolu virkne garumā no 1 līdz 250 simboliem 2. Uzvārds. Simbolu virkne garumā no 1 līdz 250 simboliem 3. E-pasts. Simbolu virkne no 6 līdz 50 simboliem. Jāatbilst e-pasta formātam. Unikāls. 4. Parole. Simbolu virkne garumā no 6 līdz 50 simboliem. 	
Apstrāde	
<p>Ja nav aizpildīts kāds no laukiem, parādās paziņojums nr. 1</p> <p>Ja laukā “Vārds” ir pārsniegts atļautais simbolu skaits, parādās paziņojums nr. 6</p> <p>Ja laukā “Uzvārds” ir pārsniegts atļautais simbolu skaits, parādās paziņojums nr. 7</p> <p>Ja laukā “E-pasts” ir pārsniegts atļautais simbolu skaits vai ievadītie dati neatbilst e-pasta formātam, parādās paziņojums nr. 2</p>	

Ja laukā “Parole” ir nenasniegts vai pārsniegts atļautais simbolu skaits, parādās paziņojums nr. 3

Ja lietotājs ar tādu e-pastu jau eksistē, parādās paziņojums nr. 8

Izvade

Ja darbības laikā rodas kļūda, lietotājam tā jānovērš un atkārtoti jāievada dati un jānospiež “JOIN”

Ja netiek konstatētas kļūdas, lietotājam parādās sistēmas galvenā lapa.

Dati tiek saglabāti datubāzē.

1.7.1.6. Profila dzēšana

Tabulā 1.8 tiek aprakstīta profila dzēšanas funkcija.

1.8. tabula Profila dzēšanas funkcijas apraksts

Identifikators	LM-06
Nosaukums	Profila dzēšana
Apraksts	Lietotājiem ir iespēja izdzēst lietotāja profilus.
Lietotāju grupa	Reģistrēts lietotājs, Administrators
Ievade	1. Parole. Simbolu virkne garumā no 6 līdz 60 simboliem.
Apstrāde	Ja laukā “Parole” ir nenasniegts vai pārsniegts atļautais simbolu skaits, parādās paziņojums nr. 3 Kad lietotājs apstiprina savu vēlni dzēst profilu, sadaļa tiek izdzēsta no datubāzēs.
Izvade	

Ja netiek konstatētas kļūdas, lietotājam parādās sistēmas sākumlapa.
Dati tiek dzēsti no datubāzē.

1.7.1.7. E-pastu atjaunošana

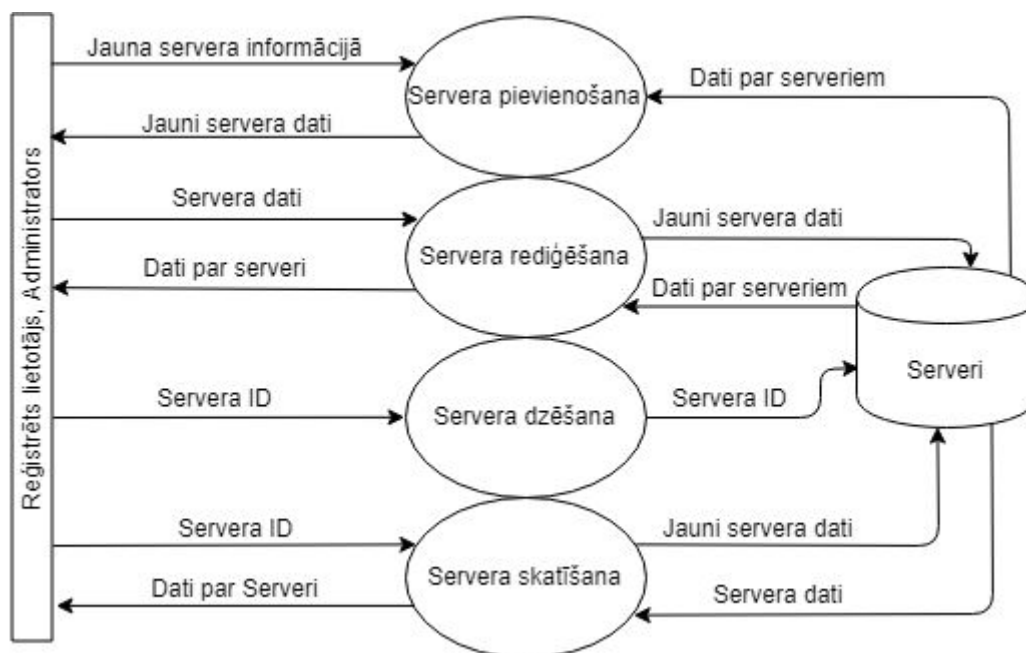
Tabulā 1.9 tiek aprakstīta e-pastu atjaunošanas funkcija.

1.9. tabula E-pastu atjaunošanas funkcijas apraksts

Identifikators	LM-07
Nosaukums	E-pastu atjaunošana
Apraksts	
	Lietotājiem ir iespēja atjaunot e-pastu
Lietotāju grupa	
	Reģistrēts lietotājs, Administrators
Ievade	
	1. Jaunais e-pasts. Simbolu virkne garumā no 6 līdz 60 simboliem.
Apstrāde	
	Ja nav aizpildīts kāds no laukiem, parādās paziņojums nr. 1 Ja laukā "E-pasts" ir pārsniegts atļautais simbolu skaits vai ievadītie dati neatbilst e-pasta formātam, parādās paziņojums nr. 2
Izvade	
	Ja darbības laikā rodas kļūdā, lietotājam tā jānovērš un atkārtoti jāievada dati un jānospiež "UPDATE EMAIL". Ja netiek konstatētas kļūdas, lietotājs nokļūst sistēmas galvenajā lapā.

1.7.2. Serveru modulis

Serveru moduļa mērķis ir veidot, rediģēt, konfigurēt un dzēst serverus. Attēlā 1.4. attēlots servera modulis un tā mijiedarbība ar datubāzi.



Attēls 1.4. 2. līmeņa datu plūsmu diagramma serveru modulim

1.7.2.1. Servera pievienošana

Tabulā 1.10 tiek aprakstīta servera izveidošanas funkcija.

1.10. tabula Servera pievienošanas funkcijas apraksts

Identifikators	SM-01
Nosaukums	Servera pievienošana
Apraksts	Lietotājiem ir iespēja izveidot serveri.
Lietotāju grupa	
	Reģistrēts lietotājs, Administrators
Ievade	

<p>1. Nosaukums. Simbolu virkne garumā no 1 līdz 250 simboliem</p> <p>2. Domēns. Simbolu virkne garumā no 1 līdz 64 simboliem.</p>
Apstrāde
<p>Ja nav aizpildīts kāds no laukiem, parādās paziņojums nr. 1</p> <p>Ja laukā “Nosaukums” ir pārsniegts atļautais simbolu skaits, parādās paziņojums nr. 6</p> <p>Ja laukā “Domēns” ir pārsniegts atļautais simbolu skaits, parādās paziņojums nr. 10</p>
Izvade
<p>Ja darbības laikā rodas kļūda, lietotājam tā jānovērš un atkārtoti jāievada dati un jānospiež “CREATE”</p> <p>Ja netiek konstatētas kļūdas, lietotājam parādās sistēmas galvenā lapa.</p> <p>Dati tiek saglabāti datubāzē.</p>

1.7.2.2. Servera rediģēšana

Tabulā 1.11 tiek aprakstīta servera rediģēšanas funkcija.

1.11. tabula Servera rediģēšanas funkcijas apraksts

Identifikators	SM-02
Nosaukums	Servera rediģēšana
Apraksts	
Lietotājiem ir iespēja rediģēt serveri	
Lietotāju grupa	
Reģistrēts lietotājs, Administrators	
Ievade	
1. Nosaukums. Simbolu virkne garumā no 1 līdz 250 simboliem.	
Apstrāde	

Ja nav aizpildīts kāds no laukiem, parādās paziņojums nr. 1

Ja laukā "Nosaukums" ir pārsniegts atļautais simbolu skaits, parādās paziņojums nr. 6

Izvade

Ja darbības laikā rodas kļūda, lietotājam tā jānovērš un atkārtoti jāievada dati un jānospiež "SAVE"

Ja netiek konstatētas kļūdas, lietotājam parādās sistēmas galvenā lapa.

Dati tiek saglabāti datubāzē.

1.7.2.3. Servera dzēšana

Tabulā 1.12 tiek aprakstīta servera dzēšanas funkcija.

1.12. tabula Servera dzēšanas funkcijas apraksts

Identifikators	SM-03
Nosaukums	Servera dzēšana
Apraksts	
Lietotāju grupām ir iespēja dzēst serveri.	
Lietotāju grupa	
Reģistrēts lietotājs, Administrators	
Ievade	
Poga "DELETE"	
Apstrāde	
Kad lietotājs apstiprina savu vēlni dzēst vienību, sadaļa tiek izdzēsta no datubāzēs.	
Izvade	
Ja netiek konstatētas kļūdas, lietotājam parādās sistēmas galvenā lapa. Dati tiek dzēsti no datubāzes.	

1.7.2.4. Servera skatīšana

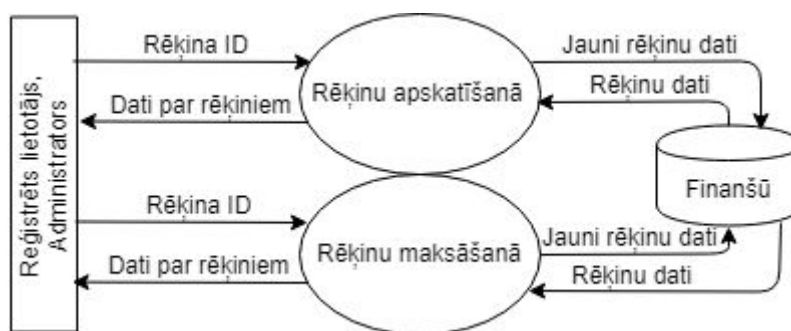
Tabulā 1.13 tiek aprakstīta servera skatīšanas funkcija.

1.13. tabula Servera skatīšanas funkcijas apraksts

Identifikators	SM-04
Nosaukums	Servera skatīšana
Apraksts	
	Ir iespējams apskatīt serveri.
Lietotāju grupa	
	Reģistrēts lietotājs, Administrators
Ievade	
	Poga "EDIT"
Apstrāde	
	-
Izvade	
	Parāda servera informāciju.

1.7.3. Finanšu modulis

Finanšu moduļa mērķis ir apstrādāt finanšu datus t.i transakcijas, vai pakalpojums ir/nav apmaksāts. Attēlā 1.5. attēlots finanšu modulis un tā mijiedarbība ar datubāzi.



Attēls 1.5. 2. līmeņa datu plūsmu diagramma finanšu modulim

1.7.3.1. Rēķinu apskatīšana

Tabulā 1.14 tiek aprakstīta Rēķinu apskatīšanas funkcija.

1.14. tabula Rēķinu apskatīšanas funkcijas apraksts

Identifikators	FM-01
Nosaukums	Rēķinu apskatīšana
Apraksts	
	Lietotājiem ir iespēja apskatīt savus rēķinus
Lietotāju grupa	
	Reģistrēts lietotājs, Administrators
Ievade	
	Poga "FINANCE"
Apstrāde	
	-
Izvade	
	Parāda visus rēķinus.

1.7.3.2. Rēķinu apmaksāšana

Tabulā 1.15 tiek aprakstīta rēķinu apmaksāšanas funkcija.

1.15. tabula Rēķinu apmaksāšanas funkcijas apraksts

Identifikators	FM-02
Nosaukums	Rēķinu apmaksāšana
Apraksts	
Lietotājiem ir iespēja apmaksāt rēķinu	
Lietotāju grupa	
Reģistrēts lietotājs, Administrators	
Ievade	
Poga "PAY"	
Apstrāde	
-	
Izvade	
Lietotājs tiek aizsūtīts uz citu sistēmu, kurā tiek veikts maksājums, un sistēma atsūta atpakaļ maksājuma statusu. Ja statuss ir pozitīvs, tad datubāzē tas tiek ierakstīts.	

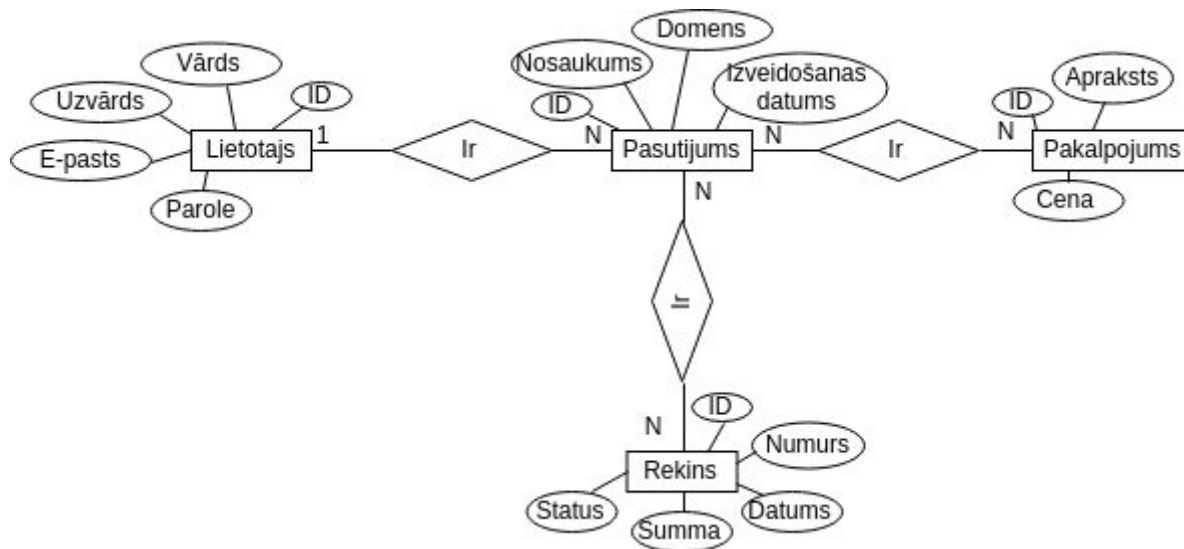
2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Datubāzes projektējums

Datubāzes projektējums sastāv no datubāzes konceptuālā modeļa, loģiskā modeļa un fiziskā modeļa. Sistēmas dati tiek glabāti vienā datubāzē. Tiek paredzēts, ka sistēmā būs 6 tabulas – Lietotājs, Rēķins, Pakalpojums, Pasūtījums, Pasūtījumi/Rēķins, Pasūtījumi/Pakalpojumi.

2.2. Konceptuālais modelis

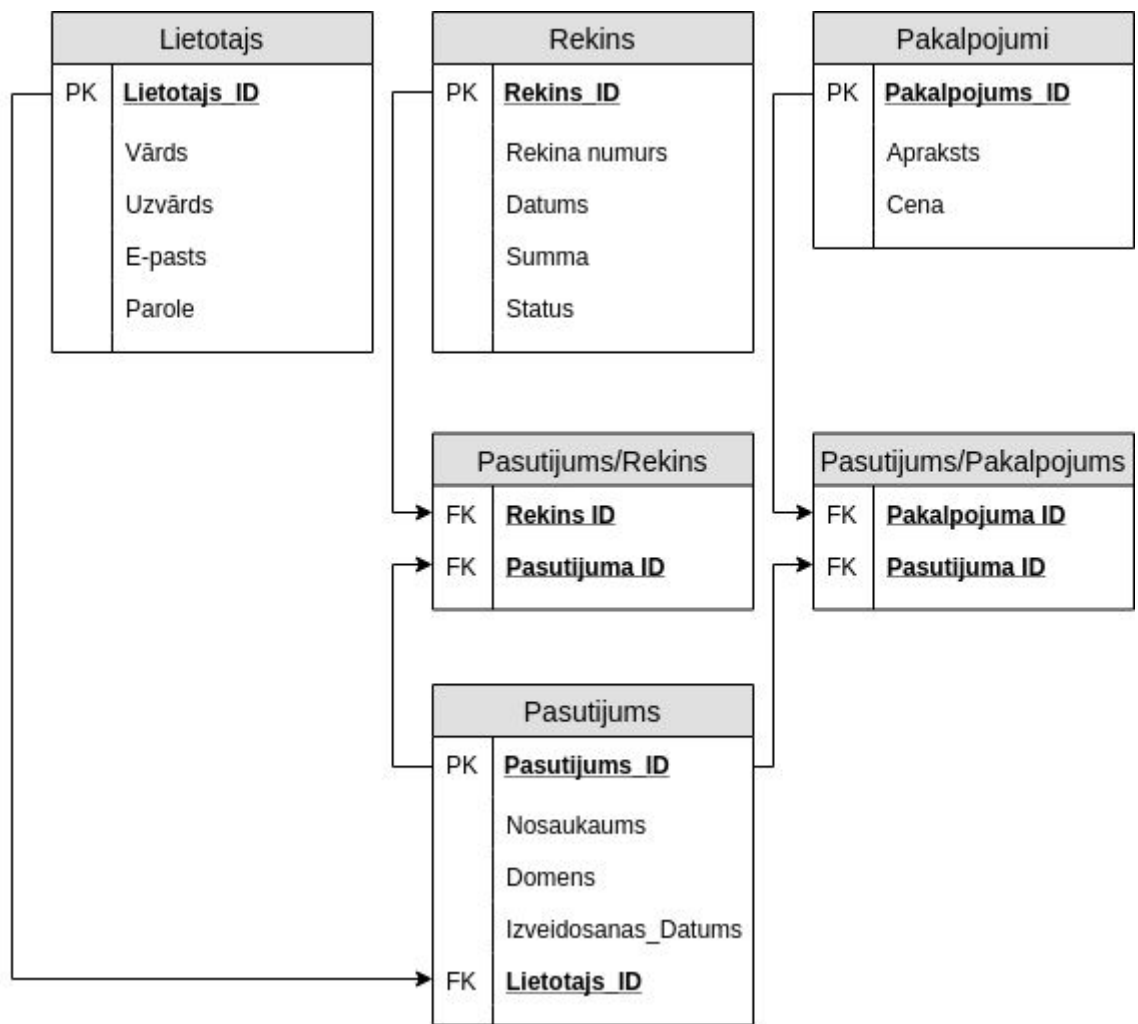
Attēlā 2.1 attēlots datubāzes konceptuālais modelis, kas raksturo galvenās relācijas starp prototipa moduļiem. Tiek paredzēts, ka sistēmā būs 4 entītijas – Lietotājs, Rēķins, Pakalpojums, Pasūtījums.



Attēls 2.1. Konceptuālais ER modelis

2.3. Loģiskais modelis

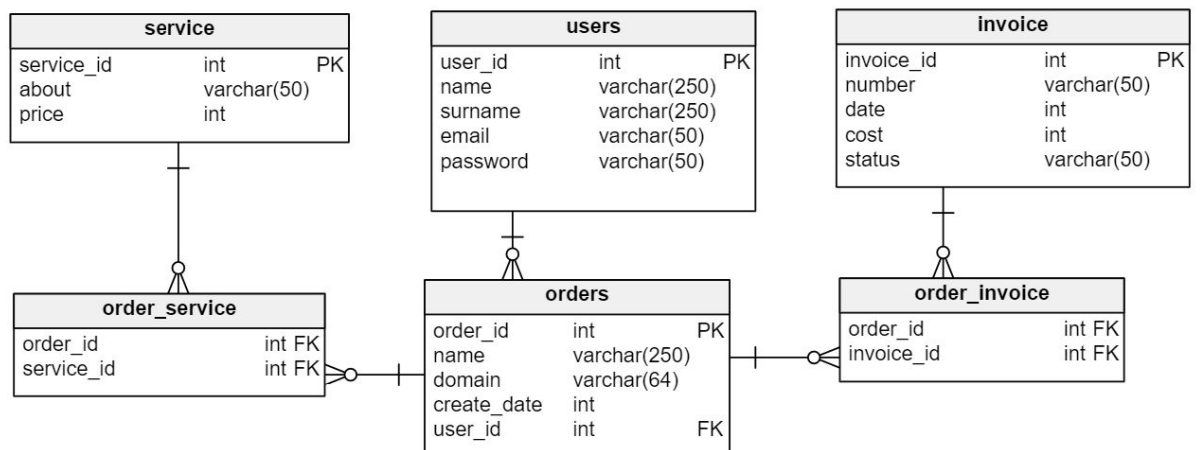
Attēla 2.2 ir redzams loģiskais datubāzes modelis. Datubāzes loģiskajā modelī attēlotas sistēmas tabulas ar primāro un ārējo atslēgu lauku identifikatoriem.



Attēls 2.2. Prototipa datubāzes loģiskais modelis

2.4. Fiziskais modelis

Attēla 2.3 ir redzams fiziskais datubāzes modelis



Attēls 2.3. Prototipa datubāzes fiziskais modelis

2.5. Datubāzes tabulu skati

2.5.1. Tabula “Lietotajs”

Tabulā 2.1. tiek glabāta informācija par visiem lietotājiem.

2.1. tabula Tabulas “Lietotāji” apraksts

Atslēgas	Nosaukums	Datu tips	Obligāts	Apraksts
PK	ID	Integer autoincrement	Jā	
	Vārds	Varchar(250)	Jā	Lietotāja vārds
	Uzvārds	Varchar(250)	Jā	Lietotāja uzvārds
	E-pasts	Varchar(50) unique	Jā	Lietotāja e-pasts
	Parole	Varchar(50)	Jā	Lietotāja parole

2.5.2. Tabula “Pasūtījums”

Tabulā 2.2. tiek glabāta informācija par visiem serveriem.

2.2. tabula Tabulas “Pasūtījums” apraksts

Atslēgas	Nosaukums	Datu tips	Obligāts	Apraksts
PK	ID	Integer autoincrement	Jā	
	Nosaukums	Varchar(250)	Jā	Pasūtījuma nosaukums
	Domēns	Varchar(64)	Jā	Pasūtījuma domēns
	Izveidošanas datums	integer	Jā	Pasūtījuma izveidošanas datums
FK	Lietotajs_ID	Integer autoincrement	Jā	Lietotāja ID, kurš izveidoja pasūtījumu

2.5.3. Tabula “Rēķins”

Tabulā 2.3. tiek glabāta informācija par visiem rēķiniem.

2.3. tabula Tabulas “Rēķins” apraksts

Atslēgas	Nosaukums	Datu tips	Obligāts	Apraksts
PK	ID	Integer autoincrement	Jā	
	Rēķina numurs	Varchar(50)	Jā	Rēķina numurs
	Datums	integer	Jā	Rēķina izveidošanas datums
	Summa	integer	Jā	Rēķina summa
	Status	Varchar(50)	Jā	Vai rēķins ir apmaksāts

2.5.4. Tabula “Pakalpojums”

Tabulā 2.4. tiek glabāta informācija par visiem pakalpojumiem.

2.4. tabula Tabulas “Pakalpojums” apraksts

Atslēgas	Nosaukums	Datu tips	Obligāts	Apraksts
PK	ID	Integer autoincrement	Jā	
	Apraksts	Varchar(50)	Jā	Pakalpojums apraksts
	Cena	integer	Jā	Pakalpojuma cena par mēnesi

2.5.5. Tabula “Pasūtījums/Pakalpojums”

Tabulā 2.5. tiek glabāta informācija par pasūtījumiem un pakalpojumiem.

2.5. tabula Tabulas “Pasūtījums/Pakalpojums” apraksts

Atslēgas	Nosaukums	Datu tips	Obligāts	Apraksts
FK	Pakalpojumu_ID	Integer autoincrement	Jā	Pakalpojuma ID
FK	Pasūtījums_ID	Integer autoincrement	Jā	Pasūtījuma ID

2.5.6. Tabula “Pasūtījums/Rēķins”

Tabulā 2.6. tiek glabāta informācija par visiem serveriem.

2.6. tabula Tabulas “Pasūtījums/Rēķins” apraksts

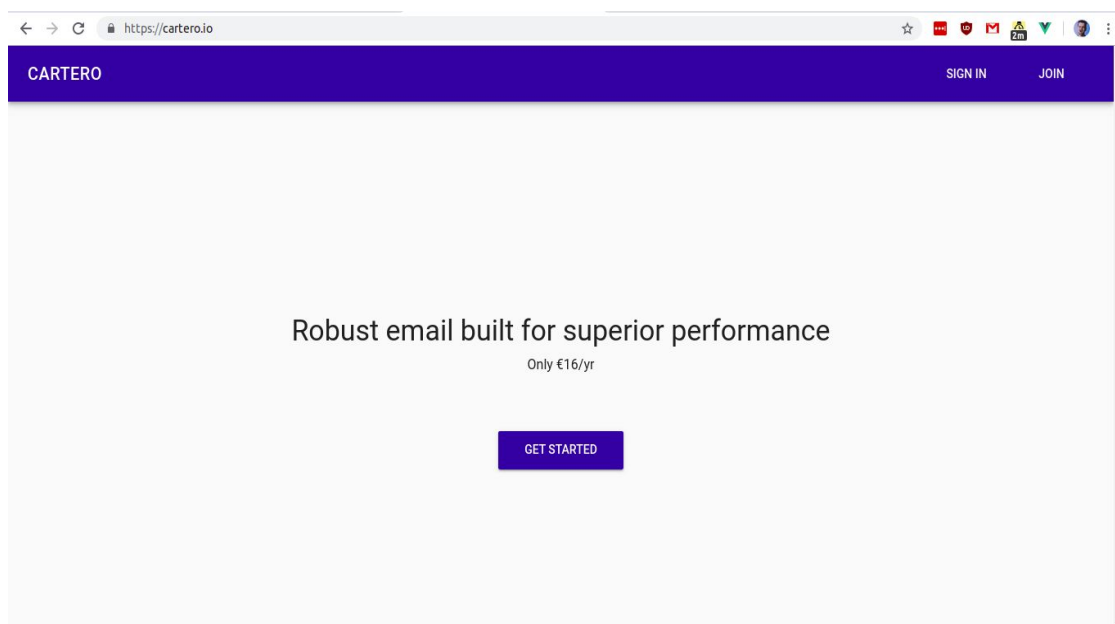
Atslēgas	Nosaukums	Datu tips	Obligāts	Apraksts
FK	Rēķins_ID	Integer autoincrement	Jā	Rēķins ID
FK	Pasūtījums_ID	Integer autoincrement	Jā	Pasūtījuma ID

3. LIETOTĀJU SASKARNES PROJEKTĒJUMS

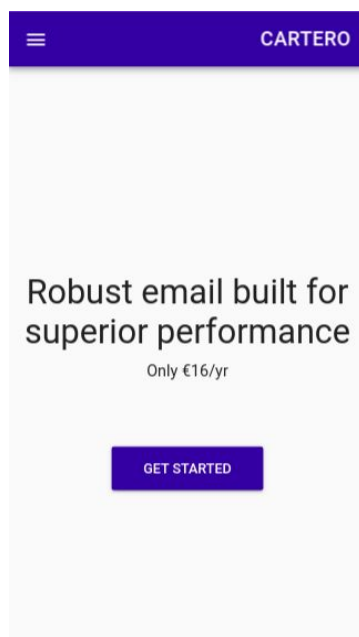
Lietotāju saskarņu izveidei tika izmantots “Vue.js” ietvars ar papildinājumu “Vuetify” . Ar to ir iespējams izveidot acīm tīkamu un responsīvu sistēmas izskatu, pie tam paveicot to diezgan īsā laika periodā. Katrā piemērā tiek pievienots saskarnes izskats uz liela un maza ekrāna.

3.1. Sākumlapā

Attēlos nr. 3.1. un 3.2 ir redzama sistēmas sākumlapā, kad lietotājs tikko ieiet mājaslapā.



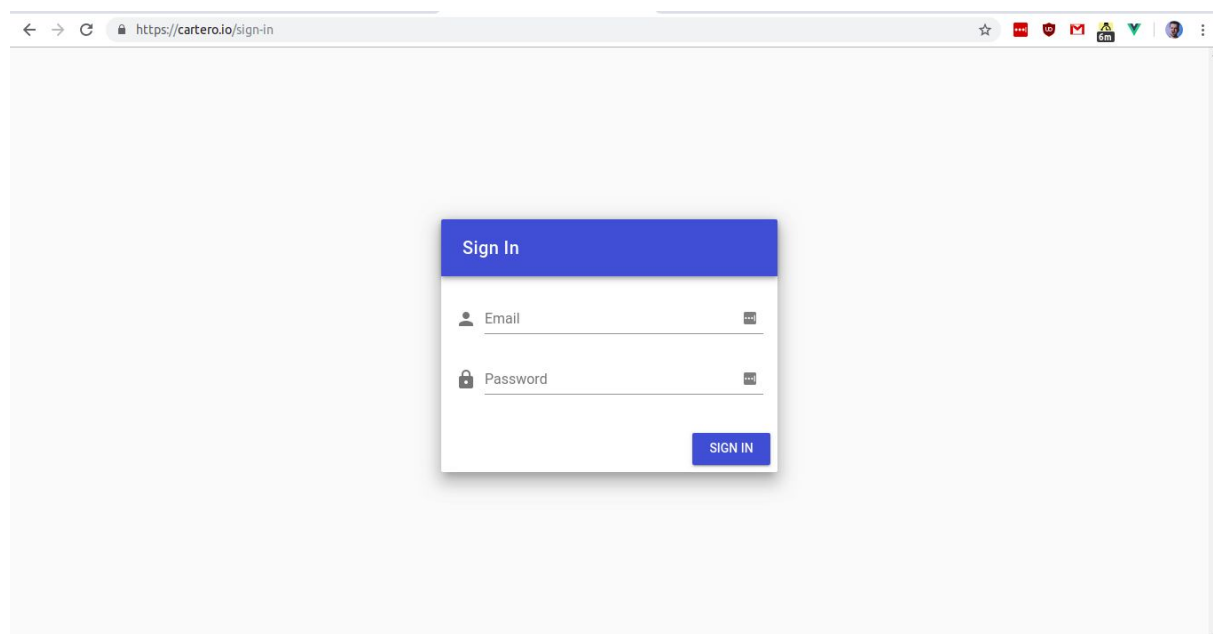
3.1. att. Sistēmas sākumlapa uz liela ekrāna



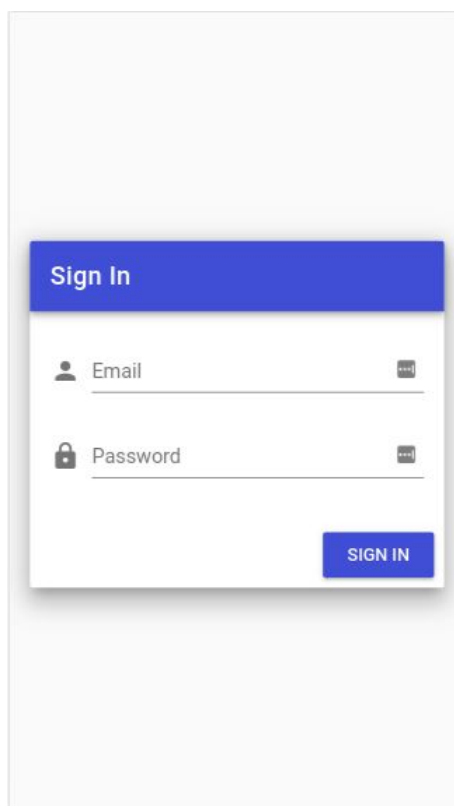
3.2. att. Sistēmas sākulapā uz maza ekrāna

3.2. Lietotāju autentificēšanās un pievienošanās lapas

Attēlos nr. 3.3. un 3.4. ir redzama sistēmas autentificēšanās lapa.

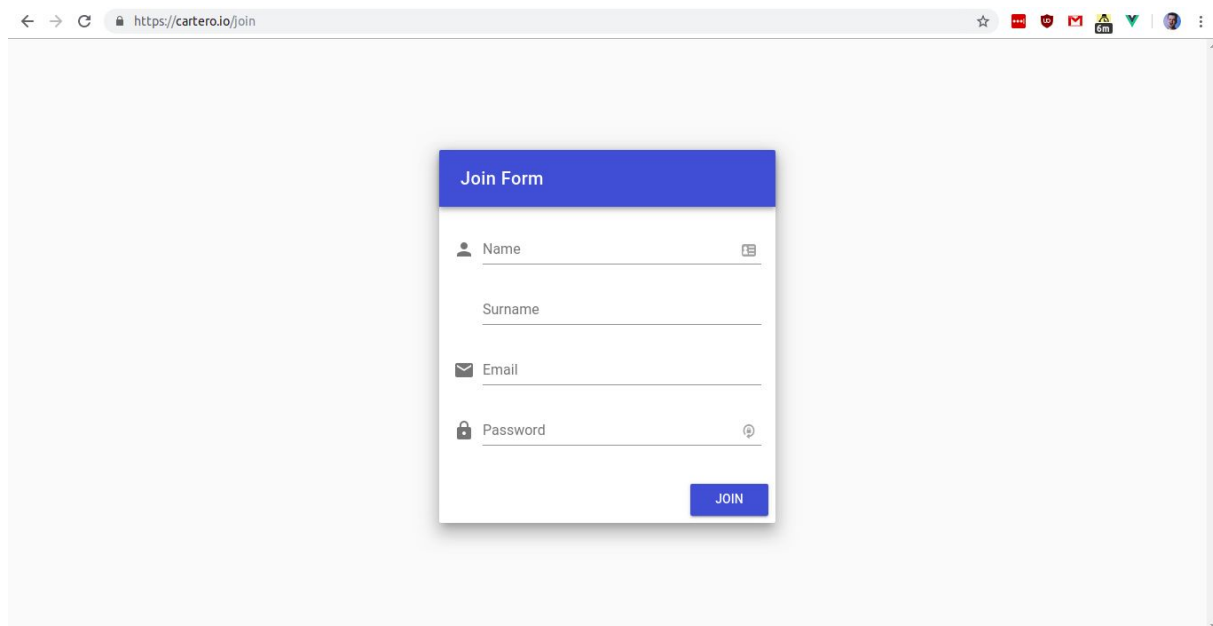


3.3. att. Autentificēšanās lapa uz lieliem ekrāniem

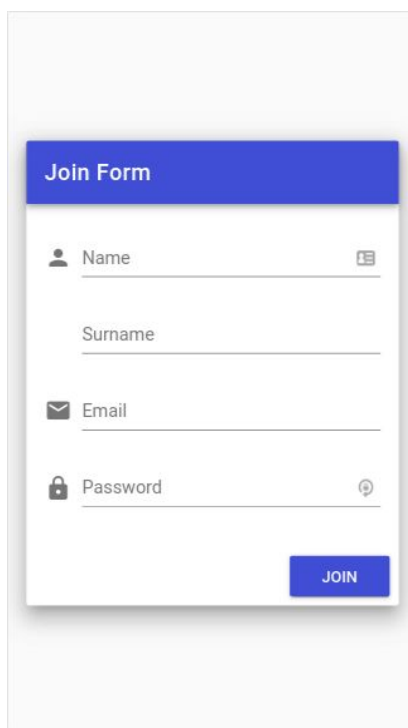


3.4. att. Autenticēšanās lapa uz mazajiem ekrāniem

Attēlos nr. 3.5. un 3.6. ir redzama sistēmas lietotāju pievienošanās lapa



3.5. att. Pievienošanās lapa uz lieliem ekrāniem

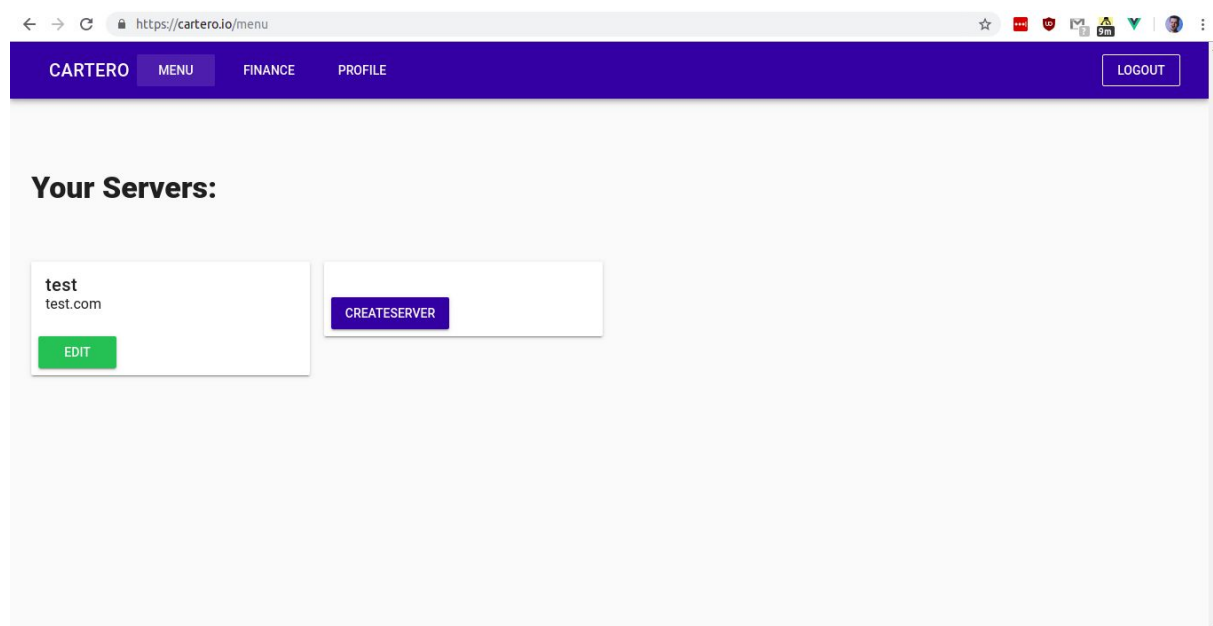


A mobile-optimized registration form titled "Join Form". It features four input fields: "Name" with a person icon, "Surname", "Email" with an envelope icon, and "Password" with a lock icon and a toggle for visibility. A blue "JOIN" button is positioned at the bottom right of the form.

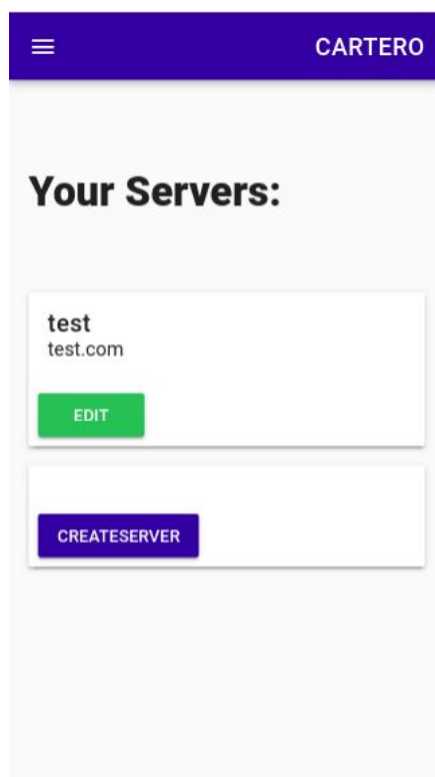
3.6. att. Pievienošanās lapa uz mazajiem ekrāniem

3.3 Lietotāja galvenais ekrāns

Attēlos nr. 3.7. un 3.8. ir redzama sistēmas galvenā lapa, kad lietotājs autentificējas mājaslapā.



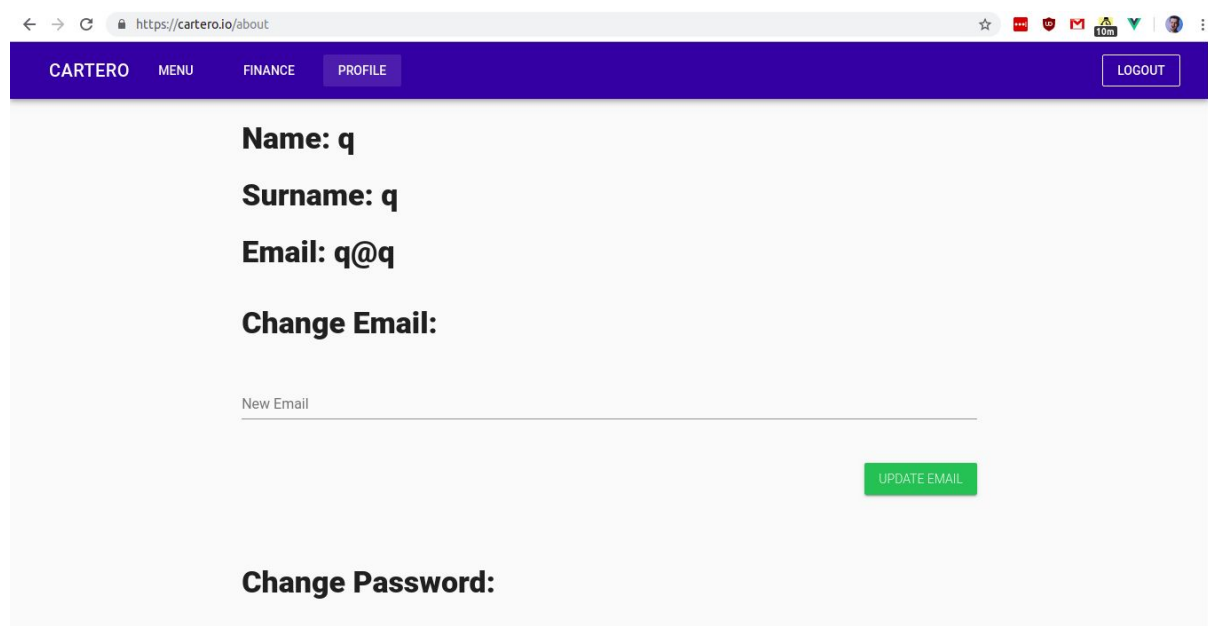
3.7. att. Galvenā lapa uz lieliem ekrāniem



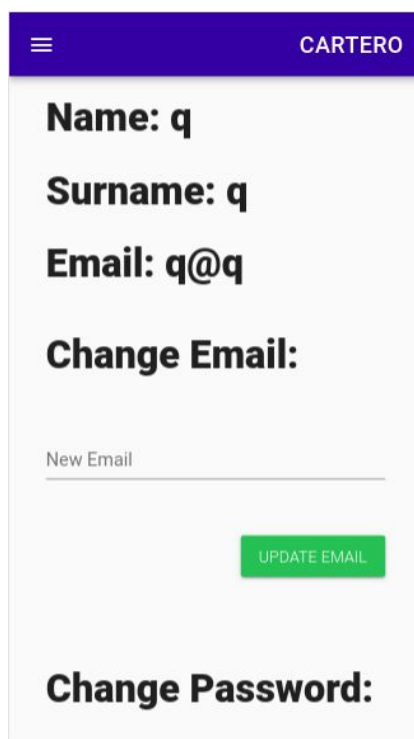
3.8. att. Galvenā lapa uz mazajiem ekrāniem

3.4. Lietotāja profils

Attēlos nr. 3.9. un 3.10. ir redzama lietotāju profila lapa.



3.9. att. Lietotāju profilu lapa uz lieliem ekrāniem



3.10. att. Lietotāju profila lapa uz mazajiem ekrāniem

4. TESTĒŠANA

Testēšana notiks manuāli, lai varētu pārbaudīt, cik ērti ir izmantot sistēmu un vai vēlāk nebūs nepieciešamība kaut ko mainīt.

Testēšanas dokumentā testi ir sadalīti vispirms pa moduļiem.

Tiek pārbaudītas PPS aprakstītās funkcijas.

4.1 Lietotāju moduļa funkciju testēšana

4.1. tabula Lietotāju moduļa funkciju testi

Nr.	Testa mērķis	Funkcijas ID	Sagaidāmais rezultāts
1.	Lietotājs apskata savu profilu	LM-01	Lietotājs redzēs profila skatu
2.	Lietotājs mēģina nomainīt paroli, visus datus ievadot pareizi	LM-02	Parole tika nomainīta
3.	Lietotājs mēģina nomainīt paroli, ievadot veco paroli nepareizi		Parole netika nomainīta un tiek paziņots, ka radusies kļūda
4.	Lietotājs mēģina nomainīt paroli, bet jaunā parole nesakrīt ar atkārtoto jauno paroli		Parole netika nomainīta un tiek paziņots, ka radusies kļūda
5.	Lietotājs mēģina ieiet sistēmā, ievadot datus nepareizi	LM-03	Lietotājs nevar ieiet sistēmā
6.	Lietotājs mēģina ieiet sistēmā, ievadot datus pareizi		Lietotājam parādās sistēmas galvenā lapa
7.	Lietotājs mēģina iziet no sistēmas	LM-04	Lietotājam tiek parādīta sākumlapa
8.	Visi obligātie dati tiek ievadīti pareizi reģistrācijas formā	LM-05	Lietotājs tiek pierēģistrēts sistēmā
9.	Reģistrācijas formā netiek ievadīti visi nepieciešamie dati		Lietotājam netiek atļauta reģistrācija, parādās kļūdu paziņojumi
10.	Reģistrācijas formā tiek nepareizi ievadīts e-pasts un pārsniegts atļautais simbolu skaits		Lietotājam netiek atļauta reģistrācija, parādās kļūdu paziņojumi
11.	Lietotājs izdzēš savu profilu, ievadot pareizo paroli	LM-06	Lietotāju profils tiek dzēsts
12.	Lietotājs izdzēš savu profilu, ievadot nepareizu paroli		Profils netika dzēsts un tiek paziņots, ka radusies kļūda
13.	Lietotājs ievada jaunu e-pastu pareizi.	LM-07	E-pasts tika atjaunots un saglabāts datubāzē.

14.	Lietotājs ievada jaunu e-pastu nepareizi.		E-pasts netika nomainīts un tiek paziņots, ka radusies kļūda
-----	---	--	--

4.2. tabula Lietotāju moduļa funkciju testēšanas datumi un rezultāti

Testa nr.	Testa datums	Rezultāts
1.	28.03	Fail
2.	28.03	Fail
3.	01.04	Fail
4.	01.04	OK
5.	03.04	OK
6.	08.04	FAIL
7.	08.04	OK
8.	01.05	FAIL
9.	01.05	OK
10.	13.05	OK
11.	16.05	OK

4.2 Finanšu moduļa funkciju testēšana

4.3. tabula Finanšu moduļa funkciju testi

Nr	Testa mērķis	Funkcijas ID	Sagaidāms rezultāts
1.	Lietotājs mēģina atvērt finanšu skatu	FM-01	Visi rēķini tiek radīti
2.	Lietotājs mēģina apmaksāt rēķinu	FM-02	Rēķins tiek apmaksāts

4.4. tabula Finanšu moduļa funkciju testēšanas datumi un rezultāti

Testa nr.	Testa datums	Rezultāts
1.	01.05	FAIL
2.	01.05	OK
3.	13.05	FAIL
4.	13.05	FAIL
5.	13.05	OK
6.	16.05	OK
7.	16.05	OK

4.3 Serveru moduļa funkciju testēšana

4.5. tabula Serveru moduļa funkciju testi

Nr	Testa mērķis	Funkcijas ID	Sagaidāms rezultāts
1.	Visi obligātie dati tiek pareizi ievadīti formā	SM-01	Tiek izveidots serveris
2.	Formā netiek ievadīti visi nepieciešamie dati		Serveris netiek izveidots, parādās kļūdu paziņojumi

3.	Formā tiek nepareizi ievadīts domēns un pārsniegts atļautais simbolu skaits		Serveris netiek izveidots, parādās kļūdu paziņojumi
4.	Tiek ievadīts jauns servera nosaukums	SM-02	Serveris tiek saglabāts ar jauno nosaukumu
5.	Lietotājs izdzēš serveri	SM-03	Serveris tiek dzēsts no datubāzes
6.	Lietotājs apskata serveri	SM-04	Parādās servera informācija

4.6. tabula Serveru moduļa funkciju testēšanas datumi un rezultāti

Testa nr.	Testa datums	Rezultāts
1.	08.04	FAIL
2.	08.04	FAIL
3.	08.04	OK
4.	01.05	FAIL
5.	01.05	OK
6.	13.05	OK
7.	16.05	OK

5. PROJEKTA PĀRVALDĪBA

5.1. Projekta organizācija

Darbu izstrādāja viens cilvēks, kas arī ir šī darba autors. Projekta izstrādei nepieciešamā funkcionalitāte bija nedefinēta no pasūtītāja puses un saplānota. Tas ļāva ātrāk un konstruktīvāk strādāt pie sistēmas izstrādes. Sistēma tika izstrādāta pakāpeniski. No sākuma tika pabeigts viens modulis, tad tika testētas visas funkcijas šajā modulī, lai ātrāk atrastu problēmas un izlabotu tās. Tikai pēc moduļa testēšanas beigšanas, tika uzsākts darbs pie nākamā moduļa.

5.2. Kvalitātes nodrošināšana

Pirmkodā kvalitāte tika nodrošināta, veicot sekojošās darbības:

- izstrādājot programmas pirmkodu, tika ievēroti Go, JavaScript un HTML programmēšanas un noformēšanas standarti;
- patstāvīgi tika veiktas pirmkoda atkārtotas pārbaudes;
- pirmkods tika izstrādāts vienotā un saprotamā stilā;
- mainīgo nosaukumi tika veidoti saprotami.

Dokumentācijas kvalitāte tika nodrošināta ievērojot LV 68:1996 “Programmatūras prasību specifikācijas ceļvedis” [1]

5.3. Konfigurāciju pārvaldība

Programmatūras versiju kontrolei tika izmantots pirmkoda versiju pārvaldības rīks “Git”. Projekts tika glabāts “GitLab” versiju kontroles servisā. Izstrādes laikā Git tika uzturēts galvenais zars, kura mērķis uzturēt programmatūras pirmkoda galveno un pēdējo kopiju. Rīks Git dod nepieciešamības gadījumā iespēju atgriezties pie vecākām versijām.

Sistēma tika izstrādāta pa moduļiem. Pirmkods tika pievienots galvenajam repozitorijam pēc katrām lielām izmaiņām.

Notika arī dokumentācijas regulāra dublikācija. Dokumentācijas glabāšanai tika izmantots “Google Drive”. Tas ļāva operatīvi saglabāt visas dokumentācijas izmaiņas un nepieciešamības gadījumā strādāt ar to no cita datora.

5.4. Darbietilpības novērtēšana

Darbietilpības novērtējuma izveidē palīdzēja darba vadītājs, ar kuru kopā tika izvērtēta sistēmas prototipa darbietilpība. Lai novērtētu darbietilpību, tika izmantota „eksperta metode”. Katram darbam tika aprēķināts optimistiskais, reālistiskais un pesimistiskais novērtējums. Pēc tam, izmantojot darbietilpības aprēķināšanas formulu, tika aprēķināts cilvēkmēnešu skaits. Rakstot darbietilpības novērtējumu, moduļu izstrādē ietilpa gan koda rakstīšanā, gan uzrakstīto funkciju testēšana. Tabulā 5.1. redzams sastādītais darbietilpības novērtējums.

5.1.tabula Darbietilpības novērtēšana

Darbības	Novērtējums (dienās)			
	Optimistiskais	Reālistiskais	Pesimistiskais	Kopējais
Plānošana un dokumentācijas rakstīšanā	15	17	19	17
Lietotāju modulis	8	10	13	10.1667
Finanšu modulis	8	11	14	11
Serveru modulis	9	13	15	12.66667
Lietotāju saskarne	11	13	15	13
Kopā	51	64	76	63.83337

Tā ka vienā mēnesī ir 20 darba dienas, mēs varam saskaitīt, ka 63.8337 dalot ar 20 būs 3.191, kas ir nedaudz vairāk par 3 cilvēkmēnešiem, kas atbilst kvalifikācijas darba prasībām. Sistēmas izstrāde aizņēma vairāk laika, jo bija nepieciešams apgūt pilnīgi jaunas zināšanas, piemēram, apgūstot Go programmēšanas valodu. Neskatoties uz to darbs tika realizēts laikā.

NOBEIGUMS

Kvalifikācijas darba ietvaros tika izstrādāts E-pasta pakalpojumu nodrošināšanas platformas “cartero.io” prototips. Prototips tika izveidots atbilstoši specifikācijai un prasībām, kuras ir aprakstītas kvalifikācijas darbā. Sistēma tika izstrādāta, izmantojot “Vue.js” un “Echo” ietvarus. Pielikumos ir pieejami daži uzrakstītā pirmkoda fragmenti.

Izstrādes laikā tika apgūtas dažādas jaunas programmēšanas pieejas, valodas, ietvari un tehnoloģijas. Izstrādājot sistēmas prototipu, tika iegūtas jaunas zināšanas par tīmekļa lietotņu izstrādes ietvaru “Vue.js”[3] un tā papildinājumu “Vuetify” [4], serveru lietotņu izstrādes ietvaru ”Echo” [5], par programmēšanas valodu Go[6] un par sistēmas konteinerizēšanas pieeju un tehnoloģijām, konkrēti Docker[7]. Tika secināts, ka labi sagatavots plāns atvieglo sistēmas programmēšanu.

Darba kvalitāti varētu uzlabot, vairāk laika atvēlot testu un konkrētas funkcijas izveidei, jo var nākties saskarties ar neparedzētām problēmām vai, autora gadījuma, zināšanu trūkumu.

Nobeigumā var konstatēt, ka prototips spēj parādīt un nodrošināt sistēmas galvenās funkcijas. Līdz ar to var konstatēt, ka darbā nospraustais mērķis ir sasniegts.

IZMANTOTIE AVOTI

1. LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis.
[tiešsaiste – pārbaudīts 25.05.2019]
Pieejams: <https://estudijas.lu.lv/mod/resource/view.php?id=131427>
2. LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai.
[tiešsaiste – pārbaudīts 25.05.2019]
Pieejams: <https://estudijas.lu.lv/mod/resource/view.php?id=131428>
3. JavaScript valodā izveidots ietvars tīmekļa lietotnēm “Vue.js” dokumentācija
Pieejams: <https://vuejs.org/v2/guide/>
4. “Vue.js” papildinājuma “Vuify” dokumentācija.
Pieejams: <https://vuifyjs.com/en/getting-started/quick-start>
5. Tīmekļa serveru izstrādes ietvare ”Echo” dokumentācija
Pieejams: <https://echo.labstack.com/guide>
6. Kursi par programmēšanas valodu Go
Pieejams: https://www.youtube.com/watch?v=9Pk7xAT_aCU&list=PLrCZzMib1e9q-X5V9pTM6J0AemRWseM7I&index=1
7. Docker dokumentācija
Pieejams: <https://docs.docker.com/install/linux/docker-ce/ubuntu/>

PIELIKUMI

1.pielikums

Lietotāja profila skata projektējums

```
<template>
  <v-container fill-height fluid grid-list-xl>
    <v-layout justify-center wrap>
      <v-flex xs12 md8>
        <v-form>
          <v-container py-0>
            <v-layout wrap>
              <v-flex xs12 class="text-xs-left display-1
font-weight-black ">Name: {{userInfo.data.name}}</v-flex>
              <v-flex xs12 class="text-xs-left display-1
font-weight-black ">Surname: {{userInfo.data.surname}}</v-flex>
              <v-flex xs12 class="text-xs-left display-1
font-weight-black ">Email: {{userInfo.data.email}}</v-flex>
            </v-layout>
          </v-container>
        </v-form>
      </v-flex>
      <v-flex xs12 md8>
        <v-form>
          <v-container py-0>
            <v-layout wrap>
              <v-flex xs12 class="text-xs-left display-1
font-weight-black my-3">Change Email:</v-flex>
              <v-flex xs12 md12>
                <v-text-field class="purple-input" label="New Email"/>
              </v-flex>
              <v-flex xs12 text-xs-right>
                <v-btn class="mx-0 font-weight-light" color="success"
@click="submit">Update Email</v-btn>
              </v-flex>
            </v-layout>
          </v-container>
        </v-form>
      </v-flex>
    </v-flex>
  </v-container>

```

```

    <v-form>
      <v-container py-0>
        <v-layout wrap>
          <v-flex xs12 class="text-xs-left display-1
font-weight-black my-5">Change Password:</v-flex>
          <v-flex xs12 md12>
            <v-text-field type="password" class="purple-input"
name="old_1" label="Old Password"/>
          </v-flex>
          <v-flex xs12 md12>
            <v-text-field type="password" class="purple-input"
name="old_2" label="New Password"/>
          </v-flex>
          <v-flex xs12 md12>
            <v-text-field type="password" class="purple-input"
name="password" label="New Password Again"/>
          </v-flex>
          <v-flex xs12 text-xs-right>
            <v-btn class="mx-0 font-weight-light" color="success"
@click="updatePassword">Update Password</v-btn>
          </v-flex>
        </v-layout>
      </v-container>
    </v-form>
  </v-flex>
  <v-flex xs12 md8>
    <v-form>
      <v-container py-0 v-model="valid" lazy-validation>
        <v-layout wrap>
          <v-flex xs12 class="text-xs-left display-1
font-weight-black my-5">Delete Account:</v-flex>
          <v-flex xs12>
            <v-text-field type="password" class="purple-input"
name="deletepassword"
label="Please Insert Yout Password To Delete Account"
v-model="password"
:rules="passwordRules"/>
          </v-flex>
          <v-flex xs12 text-xs-right>

```

```
        <v-btn class="mx-0 font-weight-light" color="warning"
@click="del" :disabled="!valid">DELETE ACCOUNT</v-btn>
    </v-flex>
</v-layout>
</v-container>
</v-form>
</v-flex>
</v-layout>
</v-container>
</template>
```

```
package models

import (
    "encoding/json"
)

type User struct {
    User_ID    int    `json:"user_id"`
    Name       string `json:"name"`
    Surname    string `json:"surname"`
    Email      string `json:"email"`
    Phone      string `json:"phone"`
    Password   string `json:"password"`
}

type Order struct {
    Order_ID    int    `json:"order_id"`
    Name        string `json:"name"`
    Domain      string `json:"domain"`
    Create_Date int    `json:"create_date"`
    User_ID     int    `json:"user_id"`
}

type Invoice struct {
    Invoice_ID    int    `json:"invoice_id"`
    Number       string `json:"number"`
    Date         int    `json:"date"`
    Sum          int    `json:"sum"`
    Status       string `json:"status"`
}

type Login struct {
    Email       string `json:"email"`
    Password    string `json:"password"`
}

type OrderInvoice struct {
    Order_ID    int    `json:"order_id"`
```

```
Invoice_ID int `json:"invoice_id"`  
}
```

```
version: '3'

services:

  proxy:
    image: traefik:1.7.11
    container_name: cartero_proxy
    command: --api --docker
    restart: always
    ports:
      - 80:80
      - 443:443
      - 8081:8081
    volumes:
      - ./proxy/traefik.toml:/etc/traefik/traefik.toml
      - ./proxy/cartero.htpasswd:/etc/traefik/cartero.htpasswd
      - ./proxy/acme:/etc/traefik/acme
      - /var/run/docker.sock:/var/run/docker.sock
    labels:
      - traefik.enable=false
    networks:
      - frontend
      - backend

  backend:
    build: backend
    image: cartero_backend
    container_name: cartero_backend
    restart: always
    labels:
      - traefik.backend=golang
      - traefik.frontend.rule=Host:cartero.io;PathPrefixStrip:/api
      - traefik.docker.network=frontend
    volumes:
      - ./backend/cartero:/go/src/github.com/cartero
    networks:
      - frontend
      - backend
```

```
frontend:
  image: nginx:mainline-alpine
  container_name: cartero_frontend
  restart: always
  labels:
    - traefik.backend=www
    - traefik.frontend.rule=Host:cartero.io,www.cartero.io
    - traefik.port=80
    - traefik.docker.network=frontend
  networks:
    - frontend
    - backend
```

```
frontend_test:
  build: frontend
  image: cartero_frontend
  container_name: cartero_frontend_test
  restart: always
  volumes:
    - ./frontend/vue_app:/vue_app
  tty: true
  stdin_open: true
  environment:
    - CHOKIDAR_USEPOLLING=true
  labels:
    - traefik.backend=www
    - traefik.frontend.rule=Host:cartero.io,www.cartero.io
    - traefik.port=8080
    - traefik.docker.network=frontend
  networks:
    - frontend
    - backend
```

```
db:
  image: postgres
  container_name: cartero_postgress
  restart: always
  environment:
    - POSTGRES_PASSWORD=cartero
```

```
    - POSTGRES_USER=cartero
labels:
    - traefik.enable=false
networks:
    - backend

db_admin:
  image: dpage/pgadmin4
  container_name: cartero_postgress_admin
  restart: always
  environment:
    - PGADMIN_DEFAULT_EMAIL=pgadmin4@pgadmin.org
    - PGADMIN_DEFAULT_PASSWORD=admin
  labels:
    - traefik.backend=pma
    - traefik.frontend.rule=Host:pma.cartero.io
    - traefik.port=80
    - traefik.docker.network=frontend
  networks:
    - frontend
    - backend

networks:
  frontend:
  backend:
    driver: bridge
    internal: true
```

Kvalifikācijas darbs „E-pasta pakalpojumu nodrošināšanas platformas “cartero.io” prototipa izstrāde” 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: Georgijs Pliska _____ .05.2019.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs/a: Dr.dat. Reiholds Zviedris _____ .05.2019.

Recenzents: Kārlis Gobleja

Darbs iesniegts 27.05.2019.

Kvalifikācijas darbu pārbaudījumu komisijas sekretāre: Darja Solodovņikova _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2019. prot. Nr. _____

Komisijas sekretārs(-e): _____