

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

BIBLIOTĒKU SISTĒMAS KLIENTU PORTĀLS
KVALIFIKĀCIJAS DARBS

Autors: **Edgars Bērziņš**

Studenta apliecības Nr.: eb12150

Darba vadītājs: M.dat. Jānis Plūme

RĪGA 2016

ANOTĀCIJA

„Bibliotēkas sistēmas klientu portāls” ir tīmeklī bāzēts portāls, kurš nodrošina uzņēmuma „Bibliotēkas Informācijas sistēmas ALISE” klientiem ērtu un drošu informācijas iegūšanu un ir viegli administrējams portāls.

Projekts ir izstrādāts pēc spējas (agile) izstrādes metodes. Portāla izstrādē ir izmantota tīmekļa lietojumprogrammu sistēma *MVC*, kura pamatā izmanto *ASP.NET* tīmekļa vietņu programmēšanas valodu.

Atslēgvārdi: Portāls, administrēšana, *MVC*, *ASP.Net*, *BIS Alise*.

ABSTRACT

"The library system customer portal" is a web-based portal, which provides the company's "Library Information system ALISE" convenient and reliable obtaining of information for customers and portal is easy to administrable.

The project is the result of the ability (agile) development techniques. The portal is built on *MVC* web application framework, which is based on using *ASP.NET* website programming language.

Keywords: Portal, administration, MVC, ASP.Net, BIS Alice.

SATURA RĀDĪTĀJS

ANOTĀCIJA.....	2
ABSTRACT	3
SATURA RĀDĪTĀJS	4
APZĪMĒJUMU SARAKSTS.....	6
IEVADS.....	7
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	8
1.1. Ievads.....	8
1.1.1 Nolūks.....	8
1.1.2 Darbības sfēra	8
1.1.3 Definīcijas un saīsinājumi.....	8
1.1.4 Saistība ar citiem dokumentiem.....	8
1.1.5 Pārskats	8
1.2. Vispārējais apraksts	9
1.2.1. Produkta perspektīva	9
1.2.2. Produkta funkcijas.....	9
1.2.3. Lietotāja raksturozīmes.....	10
1.2.4. Vispārējie ierobežojumi	10
1.2.5. Pieņēmumi un atkarības	10
1.3. Funkcionālās prasības	11
1.3.1. Administratora sistēma	11
1.3.2. Klientu sistēma.....	22
1.3.3. Ārējās saskarnes prasības	26
1.3.4. Drošība	27
2. PROGRAMMATŪRAS PROJEKTĒJUMU APRAKTS.....	28
2.1. Ievads.....	28
2.1.1. Nolūks.....	28
2.1.2. Darbības sfēra	28
2.1.3. Definīcijas un saīsinājumi:.....	28
2.1.4. Saistība ar citiem dokumentiem.....	28
2.2. Dekompozīcijas apraksts	28
2.2.1. Moduļu dekompozīcija	28
2.2.2. Datu dekompozīcija	35
2.3. Atkarības apraksts	35
2.3.1. Datu atkarības.....	35
2.5. Detalizētais projektējums.....	36

2.5.1.	Datu detalizēts projektējums	36
2.6.	Konfigurēšanas parametri	39
2.6.1.	Microsoft IIS 10 un .NET 4.5 konfigurācija.....	39
3.	TESTĒŠANAS DOKUMENTĀCIJA.....	40
3.1.	Ievads.....	40
3.1.1.	Nolūks.....	40
3.2.	Testēšanas žurnāls	40
3.2.1.	Testēšanas plāns	40
3.2.2.	Testēšanas rezultāti.....	41
3.2.3.	Testēšanas kļūdas.....	46
3.2.4.	Testēšanas kopsavilkums	46
4.	PROJEKTA ORGANIZĀCIJA.....	47
5.	KVALITĀTES NODROŠINĀŠANA	48
6.	KONFIGURĀCIJAS PĀRVALDĪBA.....	49
7.	DARBIETILPĪBAS NOVĒRTĒJUMS	50
8.	SECINĀJUMI	51
9.	IZMANTOTĀ LITERATŪRA UN AVOTI	52
	PIELIKUMI	53

APZĪMĒJUMU SARAKSTS

HTML – (HyperText Markup Language) – hiperteksta iezīmēšanas valoda, kas izstrādāta tīmekļa lapušu un citas pārlūkprogrammā attēlojamais informācijas glabāšanai.

IIS – atvērtā pirmkoda tīmekļa serveris.

MS SQL – relāciju datubāžu vadības sistēma.

LINQ – integrēto pieprasījumu valoda.

ASP.NET MVC 5 WEB APPLICATION – Uz ASP.NET bāzēts tīmekļa vietņu šablons.

MVC – modulis-skats-kontrolieris.

C# – programmēšanas valoda.

Jira – tiešsaistes pieteikumu reģistrs.

TinyMCE.Jquery – Javascript bibliotēka, kas paredzēta teksta labošanai.

IEVADS

Šajā darbā ir aprakstīta „Bibliotēku sistēmas klientu portāls” projekta izstrāde. Portāls ir veidots Bibliotēkas Informācijas sistēmas ALISE (turpmāk BIS Alise) klientiem.

Portāla galvenais mērķis jeb uzdevums ir ļaut klientiem viegli piekļūt nepieciešamajiem failiem un aplūkot BIS Alise aktualitātes, kā arī ļaut administratoriem viegli pārvaldīt pār lietotājiem, bibliotēkām un ērti labot portāla saturu.

Šo kvalifikācijas darba tēmu autors izvēlējās, lai pārbaudītu un pielietotu līdzšinējās zināšanas darbā ar ASP.Net, MVC, CSS, HTML, Java, SQL, LINQ programmēšanas valodām, strādājot ar reālu informācijas pārvaldes sistēmu.

Sistēmas izveidē tika izmantotas iepriekšminētās tīmekļa programmēšanas valodas – ASP.Net, MVC, CSS, HTML, Java, SQL, LINQ.

Izstrādātais kvalifikācijas darbs satur dokumentāciju, kurā ietilpst programmatūras prasību specifikācija, programmatūras projektējuma apraksts, testēšanas žurnāls, konfigurācijas pārvaldība, projekta organizācija, kvalitātes nodrošināšana, darbietilpības novērtējums, kā arī kodu paraugs apskatāms pielikumā.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1 Nolūks

Programmatūras prasību specifikācija (PPS) ir paredzēta, lai skaidri un nepārprotami aprakstītu izstrādājamās programmatūras „Bibliotēku sistēmas klientu portāls” prasību aprakstīšanai. Pēc PPS tiks izstrādāta programmatūra.

1.1.2 Darbības sfēra

„Bibliotēku sistēmas klientu portāls” ļaus uzņēmuma klientiem meklēt bibliotēkas, sekot līdzi aktualitātēm un iegūt nepieciešamos dokumentus, failus. Lai varētu lejupielādēt failus, lietotājiem nepieciešams pieteikties portālā.

1.1.3 Definīcijas un saīsinājumi

Saīsinājums	Skaidrojums
BIS Alise	Bibliotēku informācijas sistēma ALISE ir Latvijā izplatītākais bibliotēku automatizācijas risinājums
DB	Datu bāze
HTML	Hiperteksta iezīmēšanas valoda
PPS	Programmatūras prasību specifikācija
SQL	Strukturētā pieprasījumu valoda (Structured Query Language)

1.1.4 Saistība ar citiem dokumentiem

Programmatūras prasību specifikācija ir kvalifikācijas darba „Bibliotēku sistēmas klientu portāls” sastāvdaļa.

Dokumenta izveidošanā tika izmantots standarts LVS 68:1996 „Programmatūras prasību specifikācijas ceļvedis” [1.].

1.1.5 Pārskats

Šis dokuments sastāv no 3 nodaļām.

1. nodaļā ir dots ievads un definīcijas.
2. nodaļā ir dots vispārīgs apraksts, esošās situācijas un nākotnes mērķu izklāsts.
3. nodaļā ir norādīts detalizēts tehnisko prasību apraksts, kuras jāievēro izstrādājot programmatūru.

1.2. Vispārējais apraksts

1.2.1. Produkta perspektīva

„Bibliotēku sistēmas klientu portāls” ir neatkarīgs portāls. Tam nav saistības ar citām datorsistēmām, taču nākotnē to var savienot ar kādu citu datorsistēmu, lai vieglāk pārvaldītu portālu. Lai gan portāls ir veidots Bibliotēkas Informācijas sistēmas ALISE klientu informācijai, tas nav saistīts ar ALISES sistēmu.

Portālu var pārlūkot populārākajās pārlūkprogrammās. Portāls pārlūkojams no galda datoriem, klēpj datoriem, viedtālruniem, planšetdatoriem.

1.2.2. Produkta funkcijas

Šajā nodaļa ir iespējams iepazīties ar portāla galvenajiem funkciju moduļiem un to aprakstiem.

1. Administratoru sistēma – nodrošina pamatfunkcijas portāla satura rediģēšanai, bibliotēku un lietotāju pārvaldību.
2. Klientu sistēma – nodrošina klientiem pieeju failiem, bibliotēku meklēšanu un aktuālo kursu, semināru jaunumiem. Lai piekļūtu failiem klientam vajag pieteikties portālā.

Lietotāju kontam ir piešķirtas lomas un atkarībā no tām, lietotājam ir pieejama konkrēta funkcionalitāte:

Neautorizēta klienta funkcija:

- meklēt bibliotēku;
- atjaunot paroli;
- apskatīt aktualitātes un informāciju portālā.

Autorizēta klienta funkcijas:

- meklēt bibliotēku;
- apskatīt aktualitātes un informāciju portālā;
- nomainīt paroli;
- lejupielādēt vai apskatīt dokumentus, failus.

Administrators funkcijas:

- portāla satura rediģēšana;
- apskatīt lietotāju sarakstu;
- lietotāju atbloķēšana;
- lietotāju dzēšana;
- lietotāju datu rediģēšana
- apskatīt bibliotēku sarakstu;
- bibliotēku pievienošana;
- bibliotēku dzēšana;
- bibliotēku datu rediģēšana;
- failu augšupielāde.

1.2.3. Lietotāja raksturierzīmes

Klientam, lai lietu portālu būs nepieciešams dators ar Interneta pieslēgumu. Ir priekšzināšanas darbā ar tīmekļa pārlūkprogrammu. Ikdienas darbā lieto datoru.

Portāla administratoram arī nepieciešamas priekšzināšanas darbā ar tīmekļa pārlūkprogrammu, kā arī pamat zināšanas par to, kā darbojas „Bibliotēkas sistēmas klientu portāls”. Papildus ir nepieciešamas HTML un SQL valodas pamat zināšanas.

1.2.4. Vispārējie ierobežojumi

Portāla darbībai būs nepieciešams tīmekļa servera programmatūra un datu bāzes vadības sistēma, kurai jānodrošina SQL pieprasījumus datu apstrādei.

Portāla „Bibliotēkas sistēmas klientu portāls” lietotājiem būs nepieciešama pieeja tīmekļa pārlūkprogrammai ar Interneta pieslēgumu.

1.2.5. Pieņēmumi un atkarības

„Bibliotēkas sistēmas klientu portāls” programmatūrai ir jāstrādā uz Google Chrome, Safari, Mozilla FireFox un Internet Explorer pārlūkprogrammām.

1.3. Funkcionālās prasības

1.3.1. Administratora sistēma

1.3.1.1. Lietotāju pārskats

1.1. tabula

Lietotāju pārskats

Identifikators	AS.LP
Mērķis	
Šīs funkcijas mērķis ir ļaut administratoram pārskatīt portālā izveidotos lietotājus un to datus.	
Ievaddati	
Funkcija tiek inicializēta no izvēlnes 'Lietotāji'.	
Apstrāde	
Pārbauda, vai lietotājam ir administratora tiesības.	
Izvaddati	
<ol style="list-style-type: none">1. Lietotāja dati (<i>E-pasts, E-pasts apstiprināts, Tālruna numurs, Nevar pieslēgties līdz, Atslēgšana, Neveiksmīgas pieslēgšanās reizes</i>) tiek attēloti sarakstā ar meklēšanas iespēju.2. Attēlota forma jauna lietotāja reģistrēšanai.3. Attēlota poga 'Dzēst', kas nodrošina lietotāja dzēšanu.4. Attēlota poga 'Labot', kas nodrošina lietotāja datu labošanu.	
Kļūdu ziņojumi	
Nav	
Saistītas funkcijas	
AS.LR, AS.LDL, KS.LA	

Lietotāju reģistrēšana

Identifikators	AS.LR
Mērķis	
Šī funkcijas mērķis ir ļaut administratoram pievienot jaunu lietotāju.	
Ievaddati	
<ol style="list-style-type: none"> 1. Funkcija ir pieejama AS.LP funkcijas logā. 2. <i>E-pasts</i>, obligāts, simbolu virkne, kas var sastāvēt līdz 255 simboliem, unikāla starp lietotājiem, sastāv no derīgas e-pasta adreses 'user@server.domain' 3. <i>Parole</i>, obligāts, simbolu virkne, reģistrjūtīga, jā sastāv no vismaz 8 simboliem, tai skaitā, vismaz viens lielais un mazais burts, viens cipars un viens simbols. 4. <i>Atkārtota parole</i>, obligāts, simbolu virkne, reģistrjūtīga, jā sakrīt ar trešo apstrādes soli. 	
Apstrāde	
<ol style="list-style-type: none"> 1. Pārbauda, vai visi obligātie ievadlauki satur datus. Ja kāds ievadlauks nesatur datus, tad tiek attēlots attiecīgs paziņojums. 2. Pārbauda, vai ievadlauks '<i>E-pasts</i>' nav jau reģistrēts sistēmā. Ja ievadlauks jau ir reģistrēts sistēmā, tad tiek attēlots attiecīgs paziņojums. 3. Pārbauda, vai ievadlauks '<i>Parole</i>' sakrīt ar '<i>Atkārtota parole</i>'. Ja nesakrīt vai nav ievadīta pareiza parole, tad tiek attēlots attiecīgs paziņojums 4. Pārbauda, vai lietotājam ir administratora tiesības. 5. Ja visi ievadlauki aizpildīti pareizi un paroles sakrīt, tad informācija tiek saglabāta datubāzē un izsūtīts e-pasts par profila apstiprināšanu. 	
Izvaddati	
Attēlo AS.LP funkcijas logu.	
Kļūdu ziņojumi	
<ol style="list-style-type: none"> 1. E-pasts jau ir aizņemts 2. Paroles nesakrīt 	
Saistītas funkcijas	
AS.LR, AS.BP, AS.LDL, AS.BDL, KS.LA	

Lietotāju datu labošana

Identifikators	AS.LDL
Mērķis	
Šī funkcijas mērķis ir ļaut administratoriem labot lietotāju datus.	
Ievaddati	
Funkcija ir pieejama AS.LP funkcijas logā spiežot pogu 'Labot'	
Apstrāde	
Pārbauda, vai lietotājam ir administratora tiesības.	
Izvaddati	
<ol style="list-style-type: none"> 1. Attēlo izvēlētā lietotāja datus (<i>E-pasts, E-pasts apstiprināts, Tālruna numurs, Nevar pieslēgties līdz, Atslēgšana, Neveiksmīgas pieslēgšanās reizes</i>). 2. Attēlota poga 'Saglabāt', kas ļauj saglabāt veiktās izmaiņas datu bāzē. 3. Attēlota poga 'Uz sarakstu', kas atgriez funkcijas AS.LP logu. 	
Kļūdu ziņojumi	
Nav	
Saistītas funkcijas	
AS.LP, KS.LA	

Lietotāju dzēšana

Identifikators	AS.LD
Mērķis	
Šī funkcijas mērķis ir ļaut administratoram dzēst lietotāju.	
Ievaddati	
<ol style="list-style-type: none"> 1. Funkcija ir pieejama AS.LP funkcijas logā. 2. Attēlo funkcijas AS.LDL datus 	
Apstrāde	
<ol style="list-style-type: none"> 1. Pārbauda, vai lietotājam ir administratora tiesības. 2. Pārbauda, vai ir bibliotēka ar lietotāja e-pastu. Ja eksistē tāda bibliotēka ar atlasītā lietotāja e-pastu, tad attēlo attiecīgo ziņojumu. 	
Izvaddati	
Attēlo AS.LP funkcijas logu.	
Kļūdu ziņojumi	
Nedrīkst dzēst lietotāju ar piesaistītu bibliotēku.	
Saistītas funkcijas	
AS.LP, AS.BP, AS.BDL, KS.LA	

Bibliotēku pārskats

Identifikators	AS.BP
Mērķis	
Šīs funkcijas mērķis ir ļaut administratoram pārskatīt portālā izveidotās bibliotēkas un to datus.	
Ievaddati	
Funkcija tiek inicializēta no izvēlnes 'Bibliotēkas'.	
Apstrāde	
Pārbauda, vai lietotājam ir administratora tiesības.	
Izvaddati	
<ol style="list-style-type: none"> 1. Bibliotēku dati (<i>Nosaukums, WebPac adrese, Veids, Novads, Novads</i>) tiek attēloti sarakstā ar meklēšanas iespēju. 2. Attēlota forma jaunas bibliotēkas pievienošanai. 3. Attēlota poga 'Dzēst', kas nodrošina bibliotēkas dzēšanu. 4. Attēlota poga 'Labot', kas nodrošina bibliotēkas datu labošanu. 	
Kļūdu ziņojumi	
Nav	
Saistītas funkcijas	
AS.BPI, AS.BDL, AS.BD, KS.LA	

Bibliotēku pievienošana

Identifikators	AS.BPI
Mērķis	
Šī funkcijas mērķis ir ļaut administratoram pievienot jaunu bibliotēku.	
Ievaddati	
<p>Funkcija ir pieejama AS.BP funkcijas logā.</p> <ol style="list-style-type: none"> 1. <i>Nosaukums</i>, obligāts, simbolu virkne. 2. <i>Novads</i>, obligāts, simbolu virkne, jāizvēlas viens no dotajiem. 3. <i>Veida</i>, obligāts, simbolu virkne, jāizvēlas viens no dotajiem. 4. <i>WebPac adrese</i>, obligāts, simbolu virkne, jāievada bibliotēkas <i>WebPac adrese</i>. 5. <i>E-pasts</i>, obligāts, simbolu virkne, jāievada e-pasts no esoša lietotāja 	
Apstrāde	
<ol style="list-style-type: none"> 1. Pārbauda, vai lietotājam ir administratora tiesības. 2. Pārbauda, vai visi obligātie ievadlauki satur datus. Ja kāds ievadlauks nesatur datus, tad tiek attēlots attiecīgs paziņojums. 3. Ja visi obligātie ievadlauki ir pareizi, tad dati tiek saglabāti datu bāzē. 	
Izvaddati	
Attēlo AS.BP funkcijas logu.	
Kļūdu ziņojumi	
Nav aizpildīti visi lauki	
Saistītas funkcijas	
AS.LR, AS.BP, AS.LDL, AS.BDL, AS.LP, KS.LA	

Bibliotēku datu labošana

Identifikators	AS.BDL
Mērķis	
Šī funkcijas mērķis ir ļaut administratoriem labot bibliotēku datus.	
Ievaddati	
<p>Funkcija ir pieejama AS.BP funkcijas logā spiežot pogu 'Labot'</p> <ol style="list-style-type: none"> 1. Attēlo izvēlētas bibliotēkas datus (<i>Nosaukums, Veids, Novads, WebPac adrese, E-pasts</i>). 2. Attēlota poga 'Saglabāt', kas ļauj saglabāt veiktās izmaiņas datu bāzē. 3. Attēlota poga 'Uz sarakstu', kas atgriez funkcijas AS.BP logu. 	
Apstrāde	
<ol style="list-style-type: none"> 1. Pārbauda, vai visi obligātie ievadlauki satur datus. Ja kāds ievadlauks nesatur datus, tad tiek attēlots attiecīgs paziņojums. 2. Ja visi obligātie ievadlauki ir pareizi, tad dati tiek saglabāti datu bāzē. 3. Pārbauda, vai lietotājam ir administratora tiesības. 	
Izvaddati	
Attēlo funkcijas AS.BP logu.	
Kļūdu ziņojumi	
Nav aizpildīti visi lauki	
Saistītas funkcijas	
AS.LP, AS.BP, KS.LA	

Bibliotēku dzēšana

Identifikators	AS.BD
Mērķis	
Šī funkcijas mērķis ir ļaut administratoram dzēst bibliotēku.	
Ievaddati	
<ol style="list-style-type: none"> 1. Funkcija ir pieejama AS.LP funkcijas logā. 2. Attēlo funkcijas AS.BDL datus 	
Apstrāde	
Pārbauda, vai lietotājam ir administratora tiesības.	
Izvaddati	
Attēlo AS.LP funkcijas logu.	
Kļūdu ziņojumi	
Nav	
Saistītas funkcijas	
AS.LP, AS.BP, AS.LDL, KS.LA	

Failu augšupielāde

Identifikators	AS.FA
Mērķis	
Šī funkcijas mērķis ir ļaut administratoram augšupielādēt failus	
Ievaddati	
<p>Funkcija tiek inicializēta no izvēlnes 'Failu augšupielāde'.</p> <ol style="list-style-type: none"> 1. Tiek attēlots failu pārļūks ar izvēli pārļūkot failus un tos atlasīt. 2. Attēlota poga 'Augšupielādēt', kas nodrošina lietotājam augšupielādēt failu serverī. 	
Apstrāde	
Pārbauda, vai lietotājam ir administratora tiesības.	
Izvaddati	
Pēc veiksmīgas augšupielādes attēlo AS.FA funkcijas logu, ja augšupielāde neizdevās, tad tiek attēlots attiecīgs paziņojums.	
Kļūdu ziņojumi	
Neizdevās augšupielādēt failu.	
Saistītas funkcijas	
AS.SL, KS.LA	

Satura labošana

Identifikators	AS.SL
Mērķis	
Šī funkcijas mērķis ir ļaut administratoram labot portāla lapu saturus.	
Ievaddati	
<ol style="list-style-type: none"> 1. Funkcija tiek inicializēta no pogas 'Labot', kas ir pieejama katrā portāla lapā. 2. Attēlots teksta labotājs ar iespējam labot portāla lapu tekstu saturus, kas konvertē uzrakstīto tekstu HTML valodā. 3. Attēlota poga 'Saglabāt', kas nodrošina lietotājam saglabāt portāla lapas saturu datu bāzē. 	
Apstrāde	
Pārbauda, vai lietotājam ir administratora tiesības.	
Izvaddati	
Attēlo funkcijas AS.SP logu	
Kļūdu ziņojumi	
Nav	
Saistītas funkcijas	
AS.SP, KS.LA	

Satura pārskats

Identifikators	AS.SP
Mērķis	
Šīs funkcijas mērķis ir ļaut administratoram pārskatīt portālā izveidotos saturus un to datus.	
Ievaddati	
Funkcija tiek inicializēta no izvēlnes 'Lapu saturs'.	
Apstrāde	
Pārbauda, vai lietotājam ir administratora tiesības.	
Izvaddati	
<ol style="list-style-type: none"> 1. Lapu saturu dati (<i>Id, Autors, Izveidots/Labots, Lapa</i>) tiek attēloti sarakstā. 2. Attēlota poga 'Labot', kas nodrošina lapas satura labošanu. 	
Kļūdu ziņojumi	
Nav	
Saistītas funkcijas	
AS.SL, KS.LA	

1.3.2. Klientu sistēma

1.3.2.1. Lietotāju autorizēšana

1.12. tabula

Lietotāju autorizēšana

Identifikators	KS.LA
Mērķis	Šī funkcijas ļauj reģistrētiem lietotājiem pieteikties sistēmā ar lietotāja lomai atbilstošajām funkcijām.
Ievaddati	<ol style="list-style-type: none">1. <i>E-pasts</i>, obligāts, pareiza formāta simbolu virkne.2. <i>Parole</i>, obligāts, no 8 līdz 128 simbolu virkne, reģistrjūtīga.3. Attēlota poga '<i>Atjaunot paroli</i>', kas nodrošina lietotājam atjaunot paroli.
Apstrāde	<ol style="list-style-type: none">1. Pārbauda, vai eksistē tāds lietotājs ar konkrēto e-pastu.2. Pārbauda, vai atrastā parole sakrīt ar ievadīto paroli.3. Pārbauda, vai lietotājam ir apstiprināts profils. Attiecīgi tik attēlots attiecīgs paziņojums un izsūtīts atkārtota vēstule uz attiecīgo e-pastu.4. Ja lietotājs ir atrasts, profils apstiprināts un paroles sakrīt, tad tiek veikta autorizēšana.5. Ja lietotājs ir 5 reizes neveiksmīgi ievadījis paroli, tad lietotājs tiek atslēgts no sistēmas uz 30 minūtēm.
Izvaddati	Lietotājam tiek atvērta viņam paredzētā portāla vide.
Kļūdu ziņojumi	Nepareizs e-pasts un/vai parole.
Saistītas funkcijas	KS.AP, AS.LDL, KS.LAK

Lietotāju atteikšanās no sistēmas

Identifikators	KS.LANS
Mērķis	
Šīs funkcijas mērķis ir autentificētiem lietotājiem atteikties no sistēmas.	
Ievaddati	
Nav	
Apstrāde	
Nodzēš sesijas datus.	
Izvaddati	
Lietotājs tiek pārdresēts uz sākumlapu.	
Kļūdu ziņojumi	
Nav	
Saisītas funkcijas	
Nav	

Atjaunot paroli

Identifikators	KS.AP
Mērķis	
Šīs funkcijas mērķis ir neautorizētam lietotājam atjaunot paroli.	
Ievaddati	
<i>E-pasts</i> , obligāts, pareiza formāta simbolu virkne.	
Apstrāde	
<ol style="list-style-type: none"> 1. Pārbauda, vai eksistē tāds e-pasts datu bāzē. 2. Funkcija uzģenerē saites adresi konkrētajam lietotājam ar lietotāja <i>Id</i> vēstulē ar e-pasta servera starpniecību, kas pāradresē uz paroles nomaiņas lapu. 	
Izvaddati	
Veiksmīgas paroles atjaunošanas gadījumā, lietotājs tiek pāradresēts uz KS.LA logu.	
Kļūdu ziņojumi	
Nepareizs e-pasts.	
Saistītas funkcijas	
KS.LA	

Nomainīt paroli

Identifikators	KS.NP
Mērķis	
Šīs funkcijas mērķis ir ļaut autorizētam lietotājam nomainīt paroli.	
Ievaddati	
<ol style="list-style-type: none"> 1. <i>Pašreizējā parole</i>, obligāts, no 8 līdz 128 simboliem. 2. <i>Jaunā parole</i>, obligāts, no 8 līdz 128 simboliem. 3. <i>Apstipriniet paroli</i>, obligāts, no 8 līdz 128 simboliem. 	
Apstrāde	
<ol style="list-style-type: none"> 1. Pārbauda, vai <i>Jaunā parole</i> sakrīt ar <i>Apstipriniet paroli</i>. 2. Parole tiek saglabāta datu bāzē. 	
Izvaddati	
Veiksmīgas paroles nomainīšanas gadījumā, lietotājs tiek pāradresēts uz sākumu.	
Kļūdu ziņojumi	
Nepareiza parole, Paroles nesakrīt	
Saistītas funkcijas	
KS.LA	

Lietotāja aktivizācija

Identifikators	Identifikators
Mērķis	Šīs funkcijas mērķis ir aktivizēt reģistrētu lietotāju.
Ievaddati	Automātiski ģenerēts aktivizācijas kods.
Apstrāde	<ol style="list-style-type: none"> 1. Funkcija uzģenerē saiti ar kodu, kas tiek nosūtīts vēstulē ar lietotāja <i>Id</i> uz attiecīgo lietotāja e-pastu. 2. Kad lietotājs atver saiti, tad datu bāzes tabulā tiek nomainīts lauks uz aktivizētu un lietotājs var pieteikties sistēmā.
Izvaddati	Ja lietotājs veiksmīgi atvēris saiti, tad tiek pārdresēts uz funkcijas KS.LA logu.
Kļūdu ziņojumi	Kods nesakrīt.
Saistītas funkcijas	KS.LA

1.3.3. Ārējās saskarnes prasības**1.3.3.1. Lietotāja saskarne**

Sistēmas komunikācija ar lietotāju notiks ar tīmekļa lapu palīdzību, tāpēc lietotājam būs nepieciešama pārlūkprogramma.

1.3.3.2. Aparatūras saskarne

Lai pārlūkoto portālu, nepieciešams dators vai viedtālrunis ar Interneta pieslēgumu.

1.3.3.3. Programmatūras saskarne

Portālam nepieciešams tīmekļa serveris un datubāzes sistēma.

1.3.3.4. Sakaru saskarne

Portāla lietotājiem un sistēmas tīmekļa serverim nepieciešams Interneta pieslēgums.

1.3.4. Drošība

Reizi nedēļā tiks veikta sistēmas datu bāzes rezerves kopēšana. Drošību pret SQL injekcijām nodrošina LINQ vaicājumi datu bāzei. Sistēmai būs parole, kas neatļaus neautorizētiem lietotājiem piekļūt lietotāju datiem. Parole glabāsies šifrētā veidā ar savu unikālu drošības vērtību.

2. PROGRAMMATŪRAS PROJEKTĒUMU APRAKTS

2.1. Ievads

2.1.1. Nolūks

Dokuments apraksta un apkopo portāla sistēmas detalizēto moduļu projektējumu. Dokuments ir paredzēts portāla sistēmas izstrādātājiem un uzturētājiem kā palīgmateriāls.

2.1.2. Darbības sfēra

Izstrādātās sistēmas „Bibliotēku sistēmas klientu portāls” mērķis ir atvieglot darbu portāla administratoriem, ļaujot ērti pārvaldīt pār sistēmas lietotājiem, bibliotēkām un lapu saturiem.

2.1.3. Definīcijas un saīsinājumi:

- Administrators – sistēmas lietotājs, kuram ir pieejamas visas funkcijas;
- Klients – sistēmas lietotājs ar ierobežotām funkcijām;
- Entity Framework – atvērta pirmkoda objekt relāciju kartēšanas sistēma;
- ASP.NET MVC 5 WEB APPLICATION – uz ASP.NET valodas bāzēts tīmekļu šablons;
- MVC (Model-View-Controller) – modulis-skats-kontrolieris.

2.1.4. Saistība ar citiem dokumentiem

Par pamatu programmatūras projektējuma apraksta dokumenta izstrādē ir ņemts Latvijas valsts standarts LVS 72:1996 „Ieteicamā prakse programmatūras projektējuma aprakstīšanai” [2.]

2.2. Dekompozīcijas apraksts

2.2.1. Moduļu dekompozīcija

Par pamatu moduļu izstrādei izmantots uz *.NET* programmēšanas valodas izstrādāta tīmekļu platforma *ASP.NET MVC 5 WEB APPLICATION*. Šī platforma izmanto MVC pieeju, kas atsevišķi atdala datu bāzi, datu apstrādi un prezentāciju. Datu bāzes apstrādei tiek izmantots *Entity Framework*, kas automātiski ģenerē modeļus datu bāzēm.

2.2.1.1. Saknes modulis

Nolūks: Galvenais modulis, kas nodrošina sadarbību starp pārējiem moduļiem un attēlo nepieciešamo lapu informāciju

2.1. tabula

Saknes modulis

Kontrolieris	Funkcijas	Trasējamība
HomeController.cs	Index – parāda portāla sākuma lapas skatu	
	Edit – parāda satura labošanas skatu	AS.SL
Skati	Nolūks	Trasējamība
Index.cshtml	sākuma lapas saturs	
Edit.cshtml	satura labošanas skats	
_Layout.cshtml	ielādē visus HTML bloku saturus	

2.2.1.2. Saturu modulis

Nolūks: Nodrošina portāla lapu informāciju parādīšanu, un ļauj administratoriem labot informāciju.

Izmantots: <https://www.nuget.org/packages/TinyMCE.JQuery/>

2.2. tabula

Saturu modulis

Kontrolieris	Funkcijas	Trasējamība
ContentsController.cs	Pages – parāda portāla saturu pārskatu	AS.SP
	Delete – parāda lapas dzēšanas skatu	
	Edit – parāda lapas labošanas skatu	AS.SL
	AboutTieto – parāda attiecīgo skatu	
	AboutUs – parāda attiecīgo skatu	
	Alise – parāda attiecīgo skatu	
	Contact – parāda attiecīgo skatu	
	Contracts – parāda attiecīgo skatu	
	CurrentEvents – parāda attiecīgo skatu	
	DocList – parāda attiecīgo skatu	

	Downloads – parāda attiecīgo skatu	
	History – parāda attiecīgo skatu	
	Jira – parāda attiecīgo skatu	
	Methodical – parāda attiecīgo skatu	
	ProductsAndCourses – parāda attiecīgo skatu	
	Seminars – parāda attiecīgo skatu	
	Services – parāda attiecīgo skatu	
	SiteMap – parāda attiecīgo skatu	
	TechnicalSupport – parāda attiecīgo skatu	
	Users – parāda attiecīgo skatu	
Skati	Nolūks	Trasējāmība
Pages.cshtml	saturu pārskata skats	AS.SP
Edit.cshtml	satura labošanas skats	AS.SL
AboutTieto.cshtml	portāla satura skats	
AboutUs.cshtml	portāla satura skats	
Alise.cshtml	portāla satura skats	
Contact.cshtml	portāla satura skats	
Contracts.cshtml	portāla satura skats	
CurrentEvents.cshtml	portāla satura skats	
DocList.cshtml	portāla satura skats	
Downloads.cshtml	portāla satura skats	
History.cshtml	portāla satura skats	
Jira.cshtml	portāla satura skats	
Methodical.cshtml	portāla satura skats	
ProductsAndCourses.cshtml	portāla satura skats	
Seminars.cshtml	portāla satura skats	
Services.cshtml	portāla satura skats	
SiteMap.cshtml	portāla satura skats	
TechnicalSupport.cshtml	portāla satura skats	

Users.cshtml	portāla satura skats	
SearchLayout.cshtml	ielādē visus HTML bloku saturus priekš satura pārskata skata	
EditLayout.cshtml	ielādē visus HTML bloku saturus priekš labošanas skata	

2.2.1.3. Profila modulis

Nolūks: Ļaut lietotājiem pieteikties portālā, nomainīt paroli un administratoriem veikt nepieciešamās funkcijas.

2.3. tabula

Saturu modulis

Kontrolieris	Funkcijas	Trasējamība
AccountController.cs	Login – parāda portāla pieteikšanās skatu	KS.LA
	Register – parāda portāla reģistrācijas skatu	AS.LR
	ForgotPassword – parāda paroles atjaunošanas skatu	KS.AP
	ForgotPasswordConfirmation – parāda paroles apstiprināšanas skatu	KS.AP
	ConfirmEmail – parāda e-pasta apstiprināšanas skatu	KS.LAK
	LogOff – parāda attiecīgo skatu	KS.LANS
Skati	Nolūks	Trasējamība
Login.cshtml	portāla pieteikšanās skats	
Register.cshtml	portāla reģistrēšanās skats	
ForgotPassword.cshtml	atjaunot paroli skats	
ForgotPasswordConfirmation.cshtml	paroles atjaunošanas apstiprināšanas skats	
ConfirmEmail.cshtml	e-pasta apstiprināšanas skats	

2.2.1.4. Lietotāju modulis

Nolūks: Ļaut administratoriem veikt nepieciešamās funkcijas ar lietotājiem.

2.4. tabula

Lietotāju modulis

Kontrolieris	Funkcijas	Trasejamība
AspNetUsersController.cs	Index – parāda portāla lietotāju pārskatu skatu	AS.LP
	Delete – parāda lietotāja dzēšanas skatu	AS.LD
	Edit – parāda lietotāja datu labošanas skatu	AS.LDL
	Details – parāda lietotāja informācijas skatu	
Skati	Nolūks	Trasejamība
Index.cshtml	portāla lietotāju pārskata skats ar meklēšanas iespēju	
Delete.cshtml	portāla lietotāja dzēšanas skats	
Edit.cshtml	portāla lietotāju datu labošanas skats	
Details.cshtml	paroles lietotāju informācijas skats	

2.2.1.5. Bibliotēku modulis

Nolūks: Ļaut administratoriem veikt nepieciešamās funkcijas ar bibliotēkām.

2.5. tabula

Bibliotēku modulis

Kontrolieris	Funkcijas	Trasējamība
CustomersController.cs	Index – parāda bibliotēku pārskatu skatu	AS.BP
	Delete – parāda bibliotēku dzēšanas skatu	AS.BD
	Edit – parāda bibliotēku datu labošanas skatu	AS.BDL
	Details – parāda bibliotēku informācijas skatu	
	Create – parāda bibliotēkas izveidošanas skatu	AS.BPI
Skati	Nolūks	Trasējamība
Index.cshtml	bibliotēku pārskata skats ar meklēšanas iespēju	
Delete.cshtml	bibliotēku dzēšanas skats	
Edit.cshtml	bibliotēku datu labošanas skats	
Details.cshtml	bibliotēku informācijas skats	
Create.cshtml	bibliotēku izveidošanas skats	

2.2.1.6. Administratoru modulis

Nolūks: Ļaut administratoriem pārskatīt tiem paredzētās funkcijas.

2.6. tabula

Administratoru modulis

Kontrolieris	Funkcijas	Trasējamība
AdminController.cs	Index – parāda administratoru funkciju skatu	
Skati	Nolūks	Trasējamība
Index.cshtml	administratoru funkciju skats	

2.2.1.7. Failu augšupielādes modulis

Nolūks: Ļaut administratoriem augšupielādēt failus, lai pēc tam tos varētu ievietot portālā.

2.7. tabula

Failu augšupielādes modulis

Kontrolieris	Funkcijas	Trasējamība
FileUploadController.cs	Index – augšupielāde failus un parāda skatu	AS.FA
Skati	Nolūks	Trasējamība
Index.cshtml	failu augšupielādes skats	

2.2.1.8. Failu krātuves modulis

Nolūks: Ļaut administratoriem pārskatīt augšupielādētos failus.

2.8. tabula

Failu krātuves modulis

Kontrolieris	Funkcijas	Trasējamība
FilesController.cs	Index – parāda augšupielādēto failu skats	
Skati	Nolūks	Trasējamība
Index.cshtml	augšupielādēto failu skats	

2.2.1.9. Bibliotēku meklēšanas modulis

Nolūks: Ļaut jebkurai meklēt bibliotēkas.

2.9. tabula

Bibliotēku meklēšanas modulis

Kontrolieris	Funkcijas	Trasējamība
LibraryController.cs	Index – parāda bibliotēku meklēšanas skatu	
	Search – bibliotēku meklēšanas funkcija	
Skati	Nolūks	Trasējamība
Index.cshtml	bibliotēku meklēšanas skats	

2.2.2. Datu dekompozīcija

Dati glabāsies vienā MS SQL datubāzē. Datu bāze sastāv no 7 tabulām:

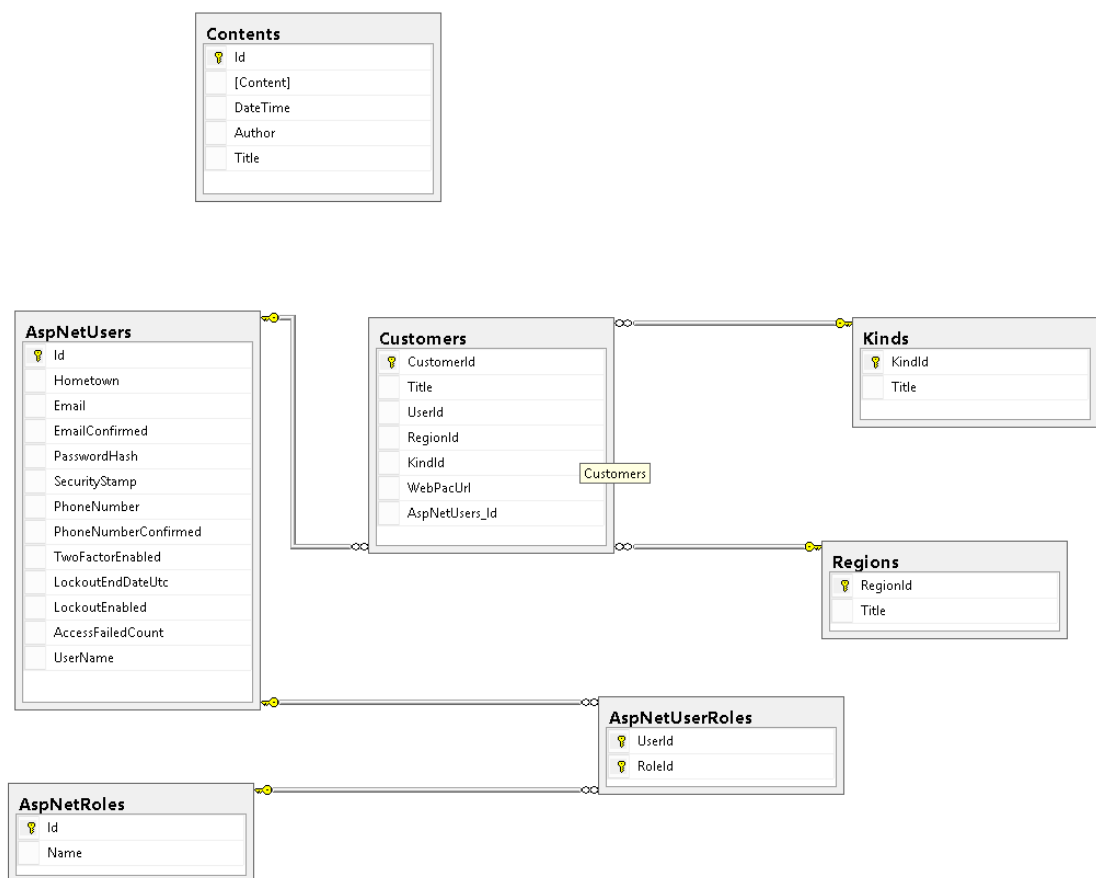
1. AspNetRole – lomu dati;
2. AspNetUserRoles – lomām piesaistīto lietotāju dati;
3. AspNetUsers – lietotāju dati;
4. Contents – portāla saturu dati;
5. Customers – bibliotēku dati;
6. Kinds – bibliotēku veidu dati;
7. Regions – bibliotēku novadu dati.

2.3. Atkarības apraksts

2.3.1. Datu atkarības

2.3.1.1. Fiziskais datubāzes modelis

Fizisko datu bāzes modeli, kurā attēlotas datu atkarības skatīt 2.1 attēlu.



2.4. att. Fiziskais datu bāzes modelis

2.5. Detalizētais projektējums

2.5.1. Datu detalizēts projektējums

2.5.1.1. Tabula „AspNetRoles”

Nolūks: Tabula paredzēta lietotāju lomu uzglabāšanai.

2.10. tabula

AspNetRoles

Lauks	Tips	Atslēgas	Null
Id	nvarchar (128)	PK	-
Name	nvarchar(256)	-	-

2.5.1.2. Tabula „AspNetUserRoles”

Nolūks: Tabula paredzēta lomām piesaistīto lietotāju datu uzglabāšanai.

2.11. tabula

AspNetUserRoles

Lauks	Tips	Atslēgas	Null
UserId	nvarchar (128)	FK	-
RoleId	nvarchar (128)	FK	-

2.5.1.3. Tabula „AspNetUsers”

Nolūks: Tabula paredzēta lietotāju datu uzglabāšanai.

2.12. tabula

AspNetUsers

Lauks	Tips	Atslēgas	Null
Id	nvarchar(128)	PK	-
Homwtown	nvarchar(256)	-	+
Email	nvarchar(256)	-	+
EmailConfirmed	Bit	-	-
PassowrdHash	nvarchar(MAX)	-	+
SecurityStamp	nvarchar(MAX)	-	+
PhoneNumber	nvarchar(MAX)	-	+
PhoneNumberConfimed	bit	-	-
TwoFactorEnabled	bit	-	-
LockoutEndDatUt	datetime	-	+
LockoutEnabled	bit	-	-
AccessFailedCount	int	-	-

2.5.1.4. Tabula „Contents”

Nolūks: Tabula paredzēta portāla satura uzglabāšanai.

2.13. tabula

Contents

Lauks	Tips	Atslēgas	Null
Id	int	PK	-
Content	nvarchar(MAX)	-	-
DateTime	datetime	-	+
Author	nvarchar(256)	-	-
Title	nvarchar(50)	-	+

2.5.1.5. Tabula „Customers”

Nolūks: Tabula paredzēta bibliotēku datu uzglabāšanai.

2.14. tabula

Customers

Lauks	Tips	Atslēgas	Null
CustomerId	int	PK	-
Title	nvarchar(256)	-	-
RegionId	int	FK	+
KindId	int	FK	+
WebPacUrl	nvarchar(100)	-	+
AspNetUsers_Id	nvarchar(128)	FK	+

2.5.1.6. Tabula „Kinds”

Nolūks: Tabula paredzēta bibliotēku veidu uzglabāšanai.

2.15. tabula

Kinds

Lauks	Tips	Atslēgas	Null
KindId	int	PK	-
Title	nvarchar(182)	-	-

2.5.1.7. Tabula „Regions”

Nolūks: Tabula paredzēta bibliotēku novadu uzglabāšanai.

2.16. tabula

Regions

Lauks	Tips	Atslēgas	Null
RegionId	Int	PK	-
Title	nvarchar(150)	-	-

2.6. Konfigurēšanas parametri

2.6.1. Microsoft IIS 10 un .NET 4.5 konfigurācija.

Lai strādātu vēstuļu sūtīšanas funkcijas, nepieciešams konfigurēt Web.config failu, kurā jānomaina *IP* un *PORT* uz attiecīgām vērtībām.

```
<system.net>
  <mailSettings>
    <smtp from="Alise.BIS@tieto.com" deliveryMethod="Network">
      <network host="IP" port="PORTS" defaultCredentials="false" />
    </smtp>
  </mailSettings>
</system.net>
```

Lai sistēma piekļūtu datu bāzei, vajag nomainīt *NOSAUKUMS* un *DATUBĀZE* uz atbilstošu servera un datubāzes nosaukumu.

```
<add name="DefaultConnection"
connectionString="Data Source=NOSAUKUMS; Initial Catalog=DATUBĀZE; Integrated Security=true"
providerName="System.Data.SqlClient" />
<add name="BisAliseEntities"
connectionString="metadata=res://*/DataModel.ModelBIS.csdl|res://*/DataModel.ModelBIS.ssdl|res://*/DataModel.
ModelBIS.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=NOSAUKUMS;initial
catalog= DATUBĀZE;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;;"
providerName="System.Data.EntityClient" />
<add name="BisAliseEntities3"
connectionString="metadata=res://*/Entities.Model1.csdl|res://*/Entities.Model1.ssdl|res://*/Entities.Model1.msl;provi
der=System.Data.SqlClient;provider connection string=&quot;data source=NOSAUKUMS;initial catalog=
DATUBĀZE;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;;"
providerName="System.Data.SqlClient" />
```

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Ievads

3.1.1. Nolūks

Dokumenta nolūks ir veikt „Bibliotēkas sistēmas klientu portāls” testēšanu un apkopot to rezultātus. Testēšanas dokumentācijā katram testam ir savs numurs, veicamā darbībā, vēlamais rezultāts un iegūtais rezultāts.

Veicot testēšanu, tika izmantots „Bibliotēkas sistēmas klientu portāls” programmatūras prasību specifikācijas dokuments.

3.2. Testēšanas žurnāls

3.2.1. Testēšanas plāns

Veicot testēšanu jāpārbauda kopējā sistēmas darbība un sistēmas moduļi. Testēšanas gaitā jāpārbauda, vai izpildās visas programmatūras prasību specifikāciju prasības.

Veicot testus jāpārbauda šādas sistēmas prasības:

- Datu ievade un izvade, datu apstrāde;
- Paziņojumu izvade atbilstoši programmatūras prasību specifikācijai;
- Funkciju izsaukšana atbilstoši lietotāju lomām;
- Funkciju izpildes rezultāts atbilstoši programmatūras prasību specifikācijai.

3.2.2. Testēšanas rezultāti

3.2.2.1. Administratoru sistēma

3.1. tabula

Lietotāju reģistrācija

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
1.	Aizpilda visus ievadlaukus pareizi.	Izveido jaunu lietotāju, un tiek izsūtīta vēstule uz norādīto e-pastu profila apstiprināšanai.	Izpildās
2.	Atstāj visus ievadlaukus tukšus.	Paziņo, ka nav aizpildīti visi ievadlauki.	Izpildās
3.	Atstāj lauku 'E-pasts' tukšu.	Paziņo, ka ievadlauks 'E-pasts' nav aizpildīts.	Izpildās
4.	Atstāj lauku 'Parole' tukšu.	Paziņo, ka ievadlauks 'Parole' ir obligāts.	Izpildās
5.	Atstāj lauku 'Apstipriniet paroli' tukšu.	Paziņo, ka ievadlauks 'Apstipriniet paroli' ir obligāts.	Izpildās
6.	Ievada pareiza formāta datus laukā 'Parole', bet laukā 'Apstipriniet paroli' ievada citādākus datus.	Paziņo, ka paroles nesakrīt.	Izpildās
7.	Ievada E-pastu, kas jau ir reģistrēts sistēmā.	Paziņo, ka E-pasts jau ir aizņemts.	Izpildās

3.2. tabula

Lietotāju pārskats

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
8.	Izsauc funkciju ar saiti 'AspNetUsers'.	Atver attiecīgo skatu.	Izpildās

Lietotāju datu labošana

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
9.	Izsauc funkciju konkrētajam lietotājam no lietotāju pārskata skata ar saiti 'Edit'.	Atver attiecīgā lietotāja datu labošanas pārskatu.	Izpildās
10.	Atstāj visus ievadlaukus tukšus.	Paziņo, ka lauks 'E-pasts' ir obligāts.	Kļūda izlabota
11.	Aizpilda visus ievadlaukus pareizi.	Tiek saglabātas veiktās izmaiņas attiecīgajam lietotājam.	Izpildās

Lietotāju dzēšana

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
12.	Mēģina izdzēst lietotāju, kura e-pasts ir piesaistīts kādai bibliotēkai.	Paziņo, ka nevar dzēst lietotāju.	Kļūda izlabota
13.	Mēģina izdzēst lietotāju, kura e-pasts nav piesaistīts kādai bibliotēkai.	Tiek izdzēst lietotājs no sistēmas un attēlo lietotāju pārskatu skatu.	Izpildās

Bibliotēku pārskats

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
14.	Izsauc funkciju ar saiti 'Customers'.	Atver attiecīgo skatu.	Izpildās

Bibliotēku pievienošana

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
15.	Aizpilda visus ievadlaukus pareizi.	Izveido jaunu bibliotēku saglabājot datus datu bāzē.	Izpildās
16.	Atstāj visus ievadlaukus tukšus.	Paziņo, ka nav aizpildīti visi obligātie ievadlauki.	Kļūda izlabota
17.	Atstāj lauku 'E-pasts' tukšu.	Paziņo, ka ievadlauks 'E-pasts' nav aizpildīts.	Kļūda izlabota
18.	Atstāj lauku 'Nosaukums' tukšu.	Paziņo, ka ievadlauks 'Nosaukums' ir obligāts.	Kļūda izlabota
19.	Aizpilda lauku 'E-pasts' ar neeksistējošu lietotāja e-pastu.	Paziņo, ka ievadlauks tāds e-pasts neeksistē.	Kļūda izlabota

Bibliotēku datu labošana

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
20.	Izsauc funkciju konkrētajai bibliotēkai no bibliotēku pārskata skata ar saiti 'Edit'.	Atver attiecīgā bibliotēku datu labošanas pārskatu.	Izpildās
21.	Atstāj visus ievadlaukus tukšus.	Paziņo, ka lauks 'E-pasts' un 'Nosaukums' ir obligāts.	Kļūda izlabota
22.	Aizpilda visus ievadlaukus pareizi.	Tiek saglabātas veiktās izmaiņas attiecīgajai bibliotēkai.	Izpildās
23.	Aizpilda lauku 'E-pasts' ar neeksistējošu lietotāju e-pastu.	Paziņo, ka tāds e-pasts neeksistē.	Kļūda izlabota

Bibliotēku dzēšana

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
24.	Mēģina izdzēst konkrētu bibliotēku.	Tiek izdzēsta bibliotēka no sistēmas un attēlo bibliotēku pārskatu skatu.	Izpildās

Failu augšupielāde

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
25.	Mēģina augšupielādēt failu.	Faili tiek veiksmīgi augšupielādēti serverī.	Izpildās
26.	Nepievieno nevienu failu un mēģina augšupielādēt.	Paziņo, ka neizdevās augšupielādēt failu.	Kļūda izlabota

Satura labošana

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
27.	Portāla satura labošana ir pieejama portāla lapās spiežot pogu 'Labot'.	Atveras teksta labotājs ar konkrētās portāla lapas saturu.	Izpildās
28.	Tiek izlabots teksts saturā.	Saglabā veiktās izmaiņas datu bāzē.	Izpildās

Satura pārskats

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
29.	Izsauc funkciju ar saiti 'Contents'.	Atvera attiecīgo skatu	Izpildās

Lietotāju autorizēšana

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
30.	Ievada pareizu e-pastu un paroli.	Klients tiek autorizēts sistēmā.	Izpildās
31.	Ievada nepareizu lauku 'E-pasts' vai nepareizu lauku 'Parole'.	Tiek paziņots, ka ievadīts nepareizs e-pasts un/vai parole.	Izpildās
32.	Atstāj visus ievadlaukus tukšus.	Paziņo, ka nav aizpildīti visi obligātie ievadlauki.	Izpildās
33.	Ievada e-pastu, kas nav apstiprināts.	Tiek paziņots, ka vajag apstiprināt e-pastu.	Izpildās

Lietotāju atteikšanās no sistēmas

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
34.	Nodzēš sesijas datus.	Lietotājs tiek pārdresēts uz sākumlapu.	Izpildās

Paroles atjaunošana

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
35.	Ievada neeksistējošu e-pastu.	Tiek izsūtīta vēstule uz e-pastu.	Izpildās
36.	Ievada pareizu e-pastu.	Tiek izsūtīta vēstule uz norādīto e-pastu, par paroles atjaunošanu.	Izpildās
37.	Paroles maiņa.	Atverot norādītājā e-pastā saiti uz paroles nomaiņu tiek atvērts skats ar paroles maiņu.	Izpildās

Lietotāja aktivizācija

Nr.	Veicamā darbība	Vēlamais rezultāts	Iegūtais rezultāts
38.	Veicot lietotāja reģistrāciju, tiek izsūtīta vēstule uz norādīto e-pastu.	Norādītājā e-pastā atverot saiti, tiek nomainīts datu bāzes tabulās ieraksts 'EmailConfirmed' un lietotājs var pieteikties sistēmā.	Izpildās

3.2.3. Testēšanas kļūdas

Nr.	Skaidrojums
10.	Netika attēlots paziņojums, par neizpildītiem ievadlaukiem lietotāju datu labošanā.
12.	Netika attēlots paziņojums, ka nevar izdzēst lietotāju.
16.	Netika attēlots paziņojums, par neizpildītiem ievadlaukiem.
17.	Netika attēlots paziņojums, par neizpildītu lauku 'E-pasts'.
18.	Netika attēlots paziņojums, par neizpildītu lauku 'Nosaukums'.
19.	Netika attēlots paziņojums, par neeksistējošu e-pastu.
21.	Netika attēlots paziņojums, par neizpildītiem ievadlaukiem.
23.	Netika attēlots paziņojums, par neeksistējošu e-pastu.
26.	Netika attēlots paziņojums par neveiksmīgu failu augšupielādi.

3.2.4. Testēšanas kopsavilkums

Testu laikā radušās kļūdas tika analizētas un izlabotas. Tika pārbaudītas administratoru tiesības un tika secināts, ka visas administratoru funkcijas ir pieejamas tikai administratoriem un ir atbilstošas programmatūras prasību specifikācijai.

4. PROJEKTA ORGANIZĀCIJA

Portāla izstrādi veica šī kvalifikācijas darba autors, kas veicas nepieciešamās darbības – projekta dokumentācijas rakstīšana, programmēšana. Konsultāciju sniedza kolēģi un darba vadītājs. Projekta pārvaldības sistēmai tika izmantota *Team Foundation Server*. Projekts ir izstrādāts uz *Microsoft Visual Studio 2015* vides izmantojot *C#* programmēšanas valodu un *ASP.NET MVC 5 Web Application*, kas papildus izmanto *EntityFramework*. Darba organizācijai tika izmantota spējas izstrāde *Agile*. Izstrādātā sistēma ir viegli lietojama arī mobilajās ierīcēs, atšķirība no iepriekš izveidotā portāla, kas nebija mobilu ierīču “draudzīgs”.

Projekta sākuma tikai plānots izveidot jaunu projektu un pārnest vecā portāla saturu bez funkcionalitātēm uz jauno projektu. Pēc tā tika izlemts, ka jāievieš portāla funkcionalitāte, jo klientiem tas ir nepieciešams. Pēc katras funkcijas izstrādes, tā tika testēta un tad nodota produkcijai. Izstrādes laikā, nācās saskarties ar vairākām lietām, kas nebija paredzētas, tāpēc tika funkcionalitāte tika mainīta. Pēc portāla pamat funkcionalitātes izstrādes tika izveidota programmatūras prasību specifikācija.

Portāla izskats tika veidots, izmantojot *CSS* un *Bootstrap* bibliotēkās, kā arī satura labotājam tika izmantota jau gatava Javascript bibliotēka – *TiniMCE.Jquery*.

5. KVALITĀTES NODROŠINĀŠANA

Portāla ir veidots pēc Programmatūras prasību specifikācijas un Programmatūras projektējuma apraksta. Abi dokumenti ir izstrādāti atbilstoši Latvijas Valsts standartiem.

Par projekta pamat šablonu tiek izmantota *ASP.NET MVC 5* „modulis-skats-kontrolieris” metode. Koda līnijas ir komentētas. Katra funkcija ir komentēta un mainīgajiem tika izvēlēti loģiski nosaukumi pēc to pielietošanas jēgas.

Testēšanas žurnāla piemēri tika izstrādāti pamatojoties pēc programmatūras prasību specifikācijas. Visas radušās kļūdas tika izlabotas un testētas vēlreiz. Papildus portāls tika testēts dažādās interneta pārlūkprogrammās, lai nodrošinātu pēc iespējas vienādu darbību un attēlošanu.

6. KONFIGURĀCIJAS PĀRVALDĪBA

Portāla „*Bibliotēkas sistēmas klientu portāls*” izstrādei tika izmantots *Team Foundation Server* pārvaldības rīks, kas ir iebūvēts *Microsoft Visual Studio 2015*, kurš veiksmīgi nodrošināja koda izmaiņu atcelšanu, veikto izmaiņu vēstures pārskatu, koda līniju izmaiņu salīdzināšanu. Papildus tika izmantota *Jira* pieteikumu reģistru sistēma, kas ļāva darba kolēģiem pieteikt izstrādāt jaunas funkcijas vai izlabot atrastās kļūdas. Viss padarītais tika testēts lokālajā vidē, pēc tam veiktās izmaiņas tika publicētas produkcijas vidē ar *Team Foundation Server* un *Remote Desktop Connection* palīdzību.

7. DARBIETILPĪBAS NOVĒRTĒJUMS

Lai aprēķinātu darbietilpības novērtējumu, tika izmantots funkciju punktu un COCOMO metodes kalkulators.

Aptuvenais rindiņu skaits kodā ir 3600. Iegūtais funkciju punktu skaits bija 80. Projekta darbietilpība bija 3 līdz 4 personmēneši, taču aprēķinot koda rindiņu skaitu ar funkciju punktu metodi tika iegūts, ka aptuvenais rindu skaits ir 4320. Aprēķinot to ar COCOMO metodi, iegūtais personmēnešu skaits ir 14,7.

8. SECINĀJUMI

Kvalifikācijas darbā tika izstrādāts Bibliotēku sistēmas klientu portāls, programmatūras prasību specifikācija, programmatūras projektējuma apraksts, testēšanas dokumentācija. Sistēma var pieteikties tikai reģistrēti lietotāji un veikt nepieciešamās darbības.

Portāla izstrādes laikā, darba autors ieguva pieredzi darbā ar C# programmēšanas valodu, kā arī zināšanas par *Microsoft Visual Studio* izmantošanu un *ASP.Net MVC* metodi.

Sistēmas izstrādes laikā autors secināja, ka ir svarīgi izmantot vairākus informācijas avotus, lai iegūtu nepieciešamās zināšanas precīzai un darbaspējīgai sistēmas izveidei.

Lai izstrādātu sistēmas ALISE klientu portālu, ir nepieciešama *Microsoft Visual Studio* programma, *Microsoft SQL* datu bāzes pārvaldības sistēma, tiešsaistes reģistra *Jira* uzņēmuma *Tieto Latvia* konts; šo visu nodrošināja *Tieto Latvia* uzņēmums.

9. IZMANTOTĀ LITERATŪRA UN AVOTI

1. Latvijas valsts standarts LVS 68:1996 - „Programmatūras prasību specifikācijas ceļvedis”
2. Latvijas valsts standarts LVS 72:1996 - „Ieteicamā prakse programmatūras projektējuma aprakstīšanai”
3. LZA Terminoloģijas komisija [tiešsaiste – pārbaudīt 24.05.2016]
Pieejams: <http://termini.lza.lv/>
4. COCOMO metodes kalkulators [tiešsaiste – pārbaudīt 24.05.2016]
Pieejams: <http://csse.usc.edu/tools/COCOMOII.php>
5. Funkciju punktu kalkulators [tiešsaiste – pārbaudīt 24.05.2016]
Pieejams: <http://groups.engin.umd.umich.edu/CIS/course>
6. .NET lietotāja rokasgrāmata [tiešsaiste – pārbaudīt 24.05.2016]
Pieejams: <http://www.asp.net/>
7. Tiešsaistes pieteikumu reģistrs [tiešsaistē – pārbaudīts 24.05.2016]

PIELIKUMI

Administratoru funkcijas ar lietotājiem

```
namespace BISAlise2.Controllers
{
    [Authorize(Roles = "Administrators")]
    public class AspNetUsersController : Controller
    {

        /// <summary>
        /// Database entity instance
        /// </summary>
        private BisAliseEntities db = new BisAliseEntities();

        /// <summary>
        /// Meklēšanas funkcija
        /// </summary>
        /// <param name="searchString"></param>
        /// <returns>View with list</returns>
        public ActionResult Index(string searchString)
        {
            var users = from m in db.AspNetUsers
                        select m;
            if (!String.IsNullOrEmpty(searchString))
            {
                users = users.Where(s => s.Email.Contains(searchString))
                    .OrderBy(s => s.Email);
            }

            return View(users.ToList());
        }

        /// <summary>
        /// AutoComplete
        /// </summary>
        /// <param name="term"></param>
        /// <returns></returns>
        public JsonResult GetUsers(string term)
        {
            BisAliseEntities db = new BisAliseEntities();
            List<string> users;
            users = db.AspNetUsers.Where(x => x.Email.Contains(term))
                .Select(p => p.Email).ToList();
            return Json(users, JsonRequestBehavior.AllowGet);
        }

        /// <summary>
        /// GET: AspNetUsers
        /// </summary>
        /// <param name="id"></param>
        /// <returns></returns>
        public ActionResult Details(string id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            AspNetUsers aspNetUsers = db.AspNetUsers.Find(id);
            if (aspNetUsers == null)
            {
                return HttpNotFound();
            }
        }
    }
}
```

```

        return View(aspNetUsers);
    }

    /// <summary>
    /// GET: AspNetUsers/Create
    /// </summary>
    /// <returns>View</returns>
    public ActionResult Create()
    {
        return View();
    }

    /// <summary>
    /// POST: AspNetUsers/Create
    /// </summary>
    /// <param name="aspNetUsers"></param>
    /// <returns>View</returns>
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
    "Id,Hometown,Email,EmailConfirmed>PasswordHash,SecurityStamp,PhoneNumber,PhoneNumberConfirmed,TwoFac
    torEnabled,LockoutEndDateUtc,LockoutEnabled,AccessFailedCount,UserName")] AspNetUsers aspNetUsers)
    {
        if (ModelState.IsValid)
        {
            db.AspNetUsers.Add(aspNetUsers);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        return View(aspNetUsers);
    }

    /// <summary>
    /// GET: AspNetUsers/Edit
    /// </summary>
    /// <param name="id"></param>
    /// <returns>View</returns>
    public ActionResult Edit(string id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        AspNetUsers aspNetUsers = db.AspNetUsers.Find(id);
        if (aspNetUsers == null)
        {
            return HttpNotFound();
        }
        return View(aspNetUsers);
    }

    /// <summary>
    /// POST: AspNetUsers/Edit
    /// </summary>
    /// <param name="aspNetUsers"></param>
    /// <returns>Redirect to Index</returns>
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
    "Id,Hometown,Email,EmailConfirmed>PasswordHash,SecurityStamp,PhoneNumber,PhoneNumberConfirmed,TwoFac
    torEnabled,LockoutEndDateUtc,LockoutEnabled,AccessFailedCount,UserName")] AspNetUsers aspNetUsers)
    {
        if (ModelState.IsValid)
        {

```

```

        aspNetUsers.UserName = aspNetUsers.Email;
        db.Entry(aspNetUsers).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(aspNetUsers);
}

/// <summary>
/// GET: AspNetUsers/Delete
/// </summary>
/// <param name="id"></param>
/// <returns>View</returns>
public ActionResult Delete(string id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    AspNetUsers aspNetUsers = db.AspNetUsers.Find(id);
    if (aspNetUsers == null)
    {
        return HttpNotFound();
    }
    return View(aspNetUsers);
}

/// <summary>
/// GEET :AspetUsers/DeleteDenied
/// </summary>
/// <param name="id"></param>
/// <returns>View</returns>
public ActionResult DeleteDenied(string id)
{
    return View();
}

/// <summary>
/// POST: AspNetUsers/Delete
/// </summary>
/// <param name="id"></param>
/// <returns>Redirect to DeleteDenied or Index</returns>
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(string id)
{
    AspNetUsers aspNetUsers = db.AspNetUsers.Find(id);
    var usr = db.Customers.Where(x => x.AspNetUsers_Id == aspNetUsers.Id).FirstOrDefault();
    if (usr != null)
    {
        aspNetUsers.Email = usr.AspNetUsers_Id;
        if (aspNetUsers.Email == usr.AspNetUsers_Id)
        {
            return RedirectToAction("DeleteDenied");
        }
    }
    db.AspNetUsers.Remove(aspNetUsers);
    db.SaveChanges();
    return RedirectToAction("Index");
}
}
}

```

Administrātoru funkcijas ar bibliotēkām

```
namespace BISAlise2.Controllers
{
    [Authorize(Roles = "Administrators")]
    public class CustomersController : Controller
    {
        /// <summary>
        /// Database entity instance
        /// </summary>
        private BisAliseEntities db = new BisAliseEntities();

        /// <summary>
        /// GET: Customers
        /// </summary>
        /// <param name="searchString"></param>
        /// <returns>View with list</returns>
        public ActionResult Index(string searchString)
        {
            var customers = from m in db.Customers
                            select m;
            if (!String.IsNullOrEmpty(searchString))
            {
                customers = customers.Where(s => s.Title.Contains(searchString))
                    .OrderBy(s => s.Title);
            }

            return View(customers.ToList());
        }

        /// <summary>
        /// GET: Customers/Details
        /// </summary>
        /// <param name="id"></param>
        /// <returns>Customers</returns>
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Customers customers = db.Customers.Find(id);
            if (customers == null)
            {
                return HttpNotFound();
            }
            return View(customers);
        }

        /// <summary>
        /// GET: Customers/Create
        /// </summary>
        /// <returns>View</returns>
        public ActionResult Create()
        {
            ViewBag.KindId = new SelectList(db.Kinds, "KindId", "Title");
            ViewBag.RegionId = new SelectList(db.Regions, "RegionId", "Title");
            ViewBag.UserId = new SelectList(db.Users, "UserId", "Email");
            return View();
        }

        /// <summary>
        /// POST: Customers/Create
    }
}
```

```

/// </summary>
/// <param name="customers"></param>
/// <returns>Redirect to Index</returns>
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"CustomerId,Title,UserId,RegionId,KindId,WebPacUrl,AspNetUsers_Id")] Customers customers)
{
    if (ModelState.IsValid)
    {
        var usr = db.AspNetUsers.Where(x => x.Email == customers.AspNetUsers_Id).FirstOrDefault();
        customers.AspNetUsers_Id = usr.Id;

        db.Customers.Add(customers);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.KindId = new SelectList(db.Kinds, "KindId", "Title", customers.KindId);
    ViewBag.RegionId = new SelectList(db.Regions, "RegionId", "Title", customers.RegionId);
    ViewBag.UserId = new SelectList(db.Users, "UserId", "Email", customers.UserId);
    ViewBag.AspNetUsers_Id = new SelectList(db.AspNetUsers, "Email", customers.AspNetUsers_Id);
    return View(customers);
}

/// <summary>
/// GET: Customers/Edit
/// </summary>
/// <param name="id"></param>
/// <returns>View</returns>
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Customers customers = db.Customers.Find(id);
    if (customers == null)
    {
        return HttpNotFound();
    }

    //lai no aspNetuser_id dabūtu epastu, ko ielikt e-pasts laukā
    var usr = db.AspNetUsers.Where(x => x.Id == customers.AspNetUsers_Id).FirstOrDefault();
    customers.AspNetUsers_Id = usr.Email;

    ViewBag.KindId = new SelectList(db.Kinds, "KindId", "Title", customers.KindId);
    ViewBag.RegionId = new SelectList(db.Regions.OrderBy(x => x.Title), "RegionId", "Title",
customers.RegionId);
    ViewBag.UserId = new SelectList(db.Users, "UserId", "Email", customers.UserId);
    ViewBag.AspNetUsers_Id = new SelectList(db.AspNetUsers, "Id", "Email", customers.AspNetUsers_Id);
    return View(customers);
}

/// <summary>
/// POST: Customers/Edit
/// </summary>
/// <param name="customers"></param>
/// <returns>Redirect to Index</returns>
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"CustomerId,Title,UserId,RegionId,KindId,WebPacUrl,AspNetUsers_Id")] Customers customers)
{

```

```

if (ModelState.IsValid)
{
    //lai no jaunā email izgūt aspnetuser_id, un ieglabātu to customers tabulā attiecīgajai bibliotēkai
    var usr = db.AspNetUsers.Where(x => x.Email == customers.AspNetUsers_Id).FirstOrDefault();
    customers.AspNetUsers_Id = usr.Id;

    db.Entry(customers).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}
ViewBag.KindId = new SelectList(db.Kinds, "KindId", "Title", customers.KindId);
ViewBag.RegionId = new SelectList(db.Regions, "RegionId", "Title", customers.RegionId);
ViewBag.UserId = new SelectList(db.Users, "UserId", "Email", customers.UserId);
ViewBag.AspNetUsers_Id = new SelectList(db.AspNetUsers, "Id", "Email", customers.AspNetUsers_Id);
return View(customers);
}

/// <summary>
/// AutoComplete funkcija
/// </summary>
/// <param name="term"></param>
/// <returns>Json list</returns>
public JsonResult GetEmail(string term)
{
    BisAliseEntities db = new BisAliseEntities();
    List<string> users;
    users = db.AspNetUsers.Where(x => x.Email.Contains(term))
        .Select(p => p.Email).ToList();
    return Json(users, JsonRequestBehavior.AllowGet);
}

/// <summary>
/// GET: Customers/Delete
/// </summary>
/// <param name="id"></param>
/// <returns></returns>
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Customers customers = db.Customers.Find(id);
    if (customers == null)
    {
        return HttpNotFound();
    }
    return View(customers);
}

/// <summary>
/// POST: Customers/Delete
/// </summary>
/// <param name="id"></param>
/// <returns>Redirect to Index</returns>
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Customers customers = db.Customers.Find(id);
    db.Customers.Remove(customers);
    db.SaveChanges();
    return RedirectToAction("Index");
}

```

```

/// <summary>
/// AutoComplete funkcija
/// </summary>
/// <param name="term"></param>
/// <returns>Json list</returns>
public JsonResult GetLibrary(string term)
{
    BisAliseEntities db = new BisAliseEntities();
    List<string> customers;
    customers = db.Customers.Where(x => x.Title.Contains(term))
        .Select(p => p.Title).ToList();
    return Json(customers, JsonRequestBehavior.AllowGet);
}
}
}

```

Administratoru funkcijas ar portāla saturu

```

namespace BISAlise2.Controllers
{
    public class ContentsController : Controller
    {
        /// <summary>
        /// Database entity instance
        /// </summary>
        private BisAliseEntities3 db = new BisAliseEntities3();

        // GET: Contact
        public ActionResult Contact()
        {
            return View(db.Contents.Where(c => c.Id == 1).ToList());
        }

        // GET: AboutUs
        public ActionResult AboutUs()
        {
            return View(db.Contents.Where(c => c.Id == 2).ToList());
        }

        // GET: AboutTieto
        public ActionResult AboutTieto()
        {
            return View(db.Contents.Where(c => c.Id == 3).ToList());
        }

        // GET: History
        public ActionResult History()
        {
            return View(db.Contents.Where(c => c.Id == 4).ToList());
        }

        // GET: Users
        public ActionResult Users()
        {
            return View(db.Contents.Where(c => c.Id == 5).ToList());
        }

        // GET: TechnicalSupport
        public ActionResult TechnicalSupport()
        {
            return View(db.Contents.Where(c => c.Id == 6).ToList());
        }

        // GET: Jira
        public ActionResult Jira()

```

```

{
    return View(db.Contents.Where(c => c.Id == 7).ToList());
}

// GET: ProductsAndCourses
public ActionResult ProductsAndCourses()
{
    return View(db.Contents.Where(c => c.Id == 8).ToList());
}

// GET: Alise
public ActionResult Alise()
{
    return View(db.Contents.Where(c => c.Id == 9).ToList());
}

// GET: Services
public ActionResult Services()
{
    return View(db.Contents.Where(c => c.Id == 10).ToList());
}

// GET: CurrentEvents
public ActionResult CurrentEvents()
{
    return View(db.Contents.Where(c => c.Id == 11).ToList());
}

// GET: SiteMap
public ActionResult SiteMap()
{
    return View(db.Contents.Where(c => c.Id == 13).ToList());
}

// GET: Seminars
public ActionResult Seminars()
{
    return View(db.Contents.Where(c => c.Id == 15).ToList());
}

// GET: Methodical
public ActionResult Methodical()
{
    return View(db.Contents.Where(c => c.Id == 16).ToList());
}

// GET: Contracts
public ActionResult Contracts()
{
    return View(db.Contents.Where(c => c.Id == 17).ToList());
}

// GET: Downloads
public ActionResult Downloads()
{
    return View(db.Contents.Where(c => c.Id == 18).ToList());
}

// GET: DocList
public ActionResult DocList()
{
    return View(db.Contents.Where(c => c.Id == 19).ToList());
}

//Portāla saturu pārskats tikai administratoriem

```

```

/// <summary>
/// GET: Pages
/// </summary>
/// <returns>View with list of pages</returns>
[Authorize(Roles = "Administrators")]
public ActionResult Pages()
{
    return View(db.Contents.OrderByDescending(c => c.DateTime).ToList());
}

/// <summary>
/// GET: Contents/Details
/// </summary>
/// <param name="id"></param>
/// <returns>View</returns>
[Authorize(Roles = "Administrators")]
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Contents contents = db.Contents.Find(id);
    if (contents == null)
    {
        return HttpNotFound();
    }
    return View(contents);
}

/// <summary>
/// GET: Contents/Create
/// </summary>
/// <returns>View</returns>
[Authorize(Roles = "Administrators")]
public ActionResult Create()
{
    return View();
}

/// <summary>
/// POST: Contents/Create
/// </summary>
/// <param name="contents"></param>
/// <returns>Redirect to Pages</returns>
[Authorize(Roles = "Administrators")]
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,Content,DateTime,Author,Title")] Contents contents)
{
    if (ModelState.IsValid)
    {
        contents.DateTime = DateTime.Now;
        contents.Author = User.Identity.GetUserName();
        db.Contents.Add(contents);
        db.SaveChanges();
        return RedirectToAction("Pages");
    }

    return View(contents);
}

/// <summary>
/// GET: Contents/Edit

```

```

/// </summary>
/// <param name="id"></param>
/// <returns>View</returns>
[Authorize(Roles = "Administrators")]
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Contents contents = db.Contents.Find(id);
    if (contents == null)
    {
        return HttpNotFound();
    }
    return View(contents);
}

/// <summary>
/// POST: Contents/Edit
/// </summary>
/// <param name="contents"></param>
/// <returns>Redirect to Pages</returns>
[Authorize(Roles = "Administrators")]
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "Id,Content,DateTime,Author,Title")] Contents contents)
{
    if (ModelState.IsValid)
    {
        contents.Author = User.Identity.GetUserName();
        contents.DateTime = DateTime.Now;

        db.Entry(contents).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Pages");
    }
    return View(contents);
}

/// <summary>
/// GET: Contents/Delete
/// </summary>
/// <param name="id"></param>
/// <returns>View</returns>
[Authorize(Roles = "Administrators")]
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Contents contents = db.Contents.Find(id);
    if (contents == null)
    {
        return HttpNotFound();
    }
    return View(contents);
}

/// <summary>
/// POST: Contents/Delete
/// </summary>
/// <param name="id"></param>
/// <returns>Redirect to Pages</returns>

```

```
[Authorize(Roles = "Administrators")]
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Contents contents = db.Contents.Find(id);
    db.Contents.Remove(contents);
    db.SaveChanges();
    return RedirectToAction("Pages");
}
}
```

Kvalifikācijas darbs „*Bibliotēku sistēmas klientu portāls*” 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: *Edgars Bērziņš* _____ **27.05.2016.**

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *M. Dat. Jānis Plūme* _____ **27.05.2016.**

Recenzents: *M. Dat. Inguna Pede*

Darbs iesniegts 30.05.2016.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2016. prot. Nr. _____

Komisijas sekretārs(-e): _____