

LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

Ģimenes ciltskoku veidošanas un pārvaldīšanas portāls

KVALIFIKĀCIJAS DARBS

Autors: Rolands Atvars

Studentu apliecības numurs: ra07010

Darba vadītājs: M.sc.dat. Krišs Rauhvargers

Rīga 2009

Anotācija

Šis kvalifikācijas darbs ir par sistēmu, kura atļauj jebkuram cilvēkam, kuram vien ir pieejams dators un interneta pieslēgums, izveidot savu ģimenes ciltskoku, kā arī uzturēt kontaktus ar savu ģimeni, radiem un draugiem un palīdzēt plānot savus un radu laikus un darbus. Lai gan ārzemēs šāda sistēma jau pastāv, Latvijas iedzīvotājiem, tā nav sevišķi piemērota, jo viņa ir angļu valodā un tur nav reģistrējušies daudz Latvijas iedzīvotāji. Ar sistēmu www.mesesam.lv var veikt tās pašas un vēl vairāk funkcijas tikai latviskā vidē ar latvisku auditoriju.

Abstract

This qualification work is about system that allows everyone who has a PC and internet connection to create a family tree and also helps to stay in touch with family, relatives and friends. The system is also very usefull to plan how to spend your and your relatives time and how to manage work. Although this kind of system exists in foreign countries it isn't very useful to latvians, because that system is in english and there aren't many latvians there. With www.mesesam.lv people will be able to do similar and more better things in latvian language interface and with latvians registered.

Saturs

SKAIDROJOŠĀ VĀRDNĪCA.....	8
IEVADS.....	9
1. DARBA MĒRĶI UN UZDEVUMI	9
2. DARBA STRUKTŪRA	9
PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	10
1. IEVADS.....	10
1.1. <i>Nolūks</i>	10
1.2. <i>Darbības sfēra</i>	10
1.3. <i>Saistība ar citiem dokumentiem</i>	10
2. VISPĀRĒJAIS APRAKSTS	10
2.1. <i>Produkta Perspektīva</i>	10
2.2. <i>Produkta funkcijas</i>	10
2.2.1. Lietotāja moduļa funkcijas.....	11
2.2.2. Komentāru moduļa funkcijas.....	11
2.2.3. Galerijas moduļa funkcijas.....	11
2.2.4. Ātro ziņu moduļa funkcijas	11
2.2.5. Ciltskoka moduļa funkcijas	12
2.2.6. Dienasgrāmatas moduļa funkcijas.....	12
2.3. <i>Lietotāja raksturiezīmes</i>	12
2.4. <i>Vispārējie ierobežojumi</i>	12
3. FUNKCIONĀLĀS PRASĪBAS	12
3.1. <i>Jauna lietotāja reģistrācija</i>	12
3.2. <i>Lietotāja iežurnālēšanās</i>	13
3.3. <i>Lietotāja atteikšanās</i>	13
3.4. <i>Lietotāja datu maiņa</i>	13
3.5. <i>Lietotāja paroles maiņa</i>	13
3.6. <i>Lietotāju meklēšana</i>	14
3.7. <i>Komentāra pievienošana</i>	14
3.8. <i>Komentāru lasīšana</i>	14
3.9. <i>Galerijas izveidošana</i>	14
3.10. <i>Bilžu pievienošana galerijā</i>	15
3.11. <i>Galerijas labošana</i>	15
3.12. <i>Galerijas apskate</i>	15
3.13. <i>Lietotāja galeriju saraksta apskate</i>	15
3.14. <i>Ātro ziņu pievienošana</i>	16
3.15. <i>Ātro ziņu dzēšana</i>	16
3.16. <i>Ātro ziņu apskate</i>	16
3.17. <i>Ciltskoka izveide</i>	16
3.18. <i>Nezināma ciltskoka locekļa pievienošana</i>	16
3.19. <i>Zināma ciltskoka locekļa pievienošana</i>	17
3.20. <i>Ciltskoka locekļa meklēšana pievienošanas brīdī</i>	17
3.21. <i>Dienasgrāmatas ieraksta pievienošana</i>	17
3.22. <i>Dienasgrāmatas ieraksta labošana</i>	18
3.23. <i>Dienasgrāmatu ierakstu apskate</i>	18
3.24. <i>Dienasgrāmatas ierakstu formatēšana</i>	18
4. NEFUNKCIONĀLĀS PRASĪBAS	19
4.1. <i>Veiktspējas prasības</i>	19
4.2. <i>Pieejamības prasības</i>	19

4.3.	<i>Datu drošība</i>	19
5.	ĀRĒJĀ SASKARNE	19
5.1.	<i>Lietotāja saskarne</i>	19
PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS		20
1.	IEVADS.....	20
1.1.	<i>Nolūks</i>	20
1.2.	<i>Darbības sfēra</i>	20
2.	DEKOMPOZĪCIJAS APRAKSTS	20
2.1.	<i>Moduļu dekompozīcija</i>	20
2.1.1.	Datu bāzes savienojuma modulis.....	20
2.1.2.	Attēlu modulis	21
2.1.3.	Galerijas modulis	21
2.1.4.	Komentāru modulis.....	21
2.1.5.	Grupu modulis	22
2.1.6.	Kreisā paneļa modulis	22
2.1.7.	Labā paneļa modulis	22
2.1.8.	XML modulis.....	22
2.1.9.	Ātro ziņu modulis.....	23
2.1.10.	Ciltskoka modulis	23
2.1.11.	Lietotāju modulis.....	23
2.1.12.	Dienasgrāmatu modulis.....	23
2.1.13.	Vadītāja modulis.....	24
2.2.	<i>Datu dekompozīcija</i>	24
2.2.1.	Users	24
2.2.2.	Relations.....	24
2.2.3.	Relation Members.....	24
2.2.4.	Postit.....	24
2.2.5.	Groups.....	24
2.2.6.	Group Members.....	24
2.2.7.	Gallery.....	24
2.2.8.	Gallery Pictures.....	24
2.2.9.	Gallery Comments.....	24
2.2.10.	Blog	25
2.2.11.	Blog Comments	25
3.	ATKARĪBAS APRAKSTS	25
3.1.	<i>Starpmodeļu un starpprocesu atkarības</i>	25
3.1.1.	0. Līmenis	25
3.1.2.	1. Līmenis	26
3.1.3.	2. Līmenis	27
3.2.	<i>Datu atkarības</i>	29
4.	DETALIZĒTAIS PROJEKTĒJUMS.....	29
4.1.	<i>Moduļu detalizēts projektējums</i>	29
4.1.1.	Datu bāzes savienojuma modulis.....	30
4.1.2.	Attēlu modulis	31
4.1.3.	Galerijas modulis	34
4.1.4.	Komentāru modulis.....	37
4.1.5.	Grupu modulis	39
4.1.6.	Ciltskoka modulis	41
4.2.	<i>Datu detalizētais projektējums</i>	42
4.2.1.	Users	42
4.2.2.	Relations.....	43
4.2.3.	Relation Members.....	43
4.2.4.	Postit.....	43
4.2.5.	Groups.....	44
4.2.6.	Group Members.....	44

4.2.7.	Gallery.....	44
4.2.8.	Gallery Pictures.....	44
4.2.9.	Blog.....	45
4.2.10.	Comments.....	45
TESTĒŠANAS DOKUMENTĀCIJA.....		46
1.	IEVADS.....	46
2.	TESTĒŠANAS REZULTĀTI.....	46
2.1.	<i>WEB lapas pirmā lapa neautorizētam lietotājam.....</i>	46
2.2.	<i>WEB lapas sākumlapa autorizētam lietotājam.....</i>	46
2.3.	<i>Lietotāja profila informācijas maiņas lapa.....</i>	48
2.4.	<i>Ciltskoka lapa.....</i>	50
2.5.	<i>Pievienot galeriju lapa.....</i>	52
2.6.	<i>Dienasgrāmatas pievienošanas lapa.....</i>	53
2.7.	<i>Galeriju apskates lapa.....</i>	54
2.8.	<i>Ātrās ziņas.....</i>	55
2.9.	<i>Manu grupu lapa.....</i>	56
2.10.	<i>Grupu saraksta lapa.....</i>	57
LIETOTĀJA DOKUMENTĀCIJA.....		58
1.	IEVADS.....	58
1.1.	<i>Auditorijas raksturojums.....</i>	58
1.2.	<i>Lietojamība.....</i>	58
1.3.	<i>Dokumenta nolūks.....</i>	58
1.4.	<i>Dokumenta lietošanas apraksts.....</i>	58
1.5.	<i>Problēmu ziņošana.....</i>	58
2.	INSTRUKCIJA SISTĒMAS LIETOTĀJIEM.....	58
2.1.	<i>Reģistrēšanās sistēmā.....</i>	58
2.1.1.	<i>Reģistrēšanās sistēmā kļūdu paziņojumi.....</i>	59
2.2.	<i>Iežurnalēšanās sistēmā.....</i>	60
2.2.1.	<i>Iežurnalēšanās sistēmā kļūdu paziņojumi.....</i>	60
2.3.	<i>Navigācija pa sadaļām.....</i>	60
2.4.	<i>Lietotāja profila informācijas maiņa.....</i>	61
2.4.1.	<i>Lietotāja profila informācijas maiņas kļūdu paziņojumi.....</i>	61
2.5.	<i>Lietotāja paroles maiņa.....</i>	62
2.5.1.	<i>Lietotāja paroles maiņas kļūdas paziņojumi.....</i>	62
2.6.	<i>Lietotāja profila bildes maiņa.....</i>	63
2.7.	<i>Jaunas galerijas pievienošana.....</i>	63
2.7.1.	<i>Jaunas galerijas pievienošanas kļūdu paziņojumi.....</i>	64
2.8.	<i>Galerijas bilžu augšupielādēšana.....</i>	64
2.9.	<i>Galeriju apskate.....</i>	65
2.10.	<i>Galerijas bilžu apskate.....</i>	66
2.11.	<i>Galerijas labošana.....</i>	66
2.12.	<i>Jaunas ātrās ziņas pievienošana.....</i>	67
2.13.	<i>Ātro ziņu dzēšana.....</i>	67
2.14.	<i>Ciltskoka izmantošana.....</i>	68
2.14.1.	<i>Sevis atrašana ciltskokā.....</i>	68
2.14.2.	<i>Jauna ciltskoka lietotāja pievienošana, ja viņš nav reģistrējies sistēmā.....</i>	68
2.14.3.	<i>Jauna ciltskoka lietotāja pievienošana, ja viņš ir reģistrējies sistēmā.....</i>	69
2.14.4.	<i>Ciltskoka locekļa nostādīšana centrā.....</i>	70
2.15.	<i>Jaunas grupas izveidošana.....</i>	70
2.15.1.	<i>Jaunas grupas izveidošana kļūdu paziņojumi.....</i>	70
2.16.	<i>Jaunu lietotāju pievienošana grupā.....</i>	71

2.17. Grupu uzaicinājumu apstiprināšana/noraidīšana	71
2.18. Grupas, kurās es esmu	72
PROJEKTA ORGANIZĀCIJA.....	73
KVALITĀTES NODROŠINĀŠANA.....	74
DARBIETILPĪBAS NOVĒRTĒJUMS	75
1. IZMANTOJOT COCOMO BASIC METODI	75
2. IZMANTOJOT COCOMO II METODI	75
3. SECINĀJUMI PAR IEGŪTAJIEM REZULTĀTIEM	76
PROGRAMMATŪRAS KODS.....	77
DOKUMENTĀRA LAPA.....	92

Skaidrojošā vārdnīca

Parsēšana – Sintaktiski analizēt doto ievades tekstu, lai izveidotu kādu struktūru

Iežurnālēšanās – lietotāja autorizēšanās sistēmā

Hostings – interneta mājaslapas izvietošana

Interneta pārlūkprogramma – programma ar kuru no datora var aplūkot internetu

Stila fails – fails, kurš atbild par to, kā tiks noformatēts un attēlots html marķēts kods

Model-View-Controller „Design Pattern” - koda rakstīšanas stils, kura mērķis ir panākt dizaina nodalīšanu no funkcionālās daļas

SQL Injections – SQL injekcija – ļaunprātīgi ievietots kods kādā no HTML ievades laukiem, lai izpildītu kaitīgu vaicājumu pret datu bāzes serveri

„INSERT” tipa SQL vaicājums – SQL vaicājums, kurš ievieto datus datu bāzē

base64 – MIME tipu pārsūtīšanas kodējums

hexadecimal color – krāsa, kas norādīta formā #000000

Ievads

Kvalifikācijas darba temats un sfēra tika izvēlēta balstoties uz studenta interesēm un augstskolā pasniegto priekšmetu „Timekļa Tehnoloģijas II”. Precīzi šī tēma tika izvēlēta tādēļ, ka tā ir iespēja izveidot ko jaunu un vēl neesošu Latvijas robežās, kā arī lieliska iespēja iemācīties ko jaunu un interesantu, kas noderētu turpmākajā dzīvē.

1. Darba mērķi un uzdevumi

Šī kvalifikācijas darba mērķis ir pielietot un pierādīt studenta zināšanas, kuras ir iegūtas mācoties Latvijas Universitātē un kuras tika papildinātās izejot praksi kompānijā „Exigen”.

2. Darba struktūra

Kvalifikācijas darbā ietilpst dokumenti un programmas kods, šajā darbā aprakstītajam projektam. Dokumentos ietilpst Programmatūras Prasību Specifikācija, Programmatūras Projektējuma Apraksts, Testēšanas Dokumentācija un Lietotāja Dokumentācija.

Programmas kods, kas pievienots kvalifikācijas darbam nesatur visu uzrakstīto kodu, jo tā apjoms ir pārāk liels, lai būtu jēga to ievietot šajā darbā, tā vietā ir ievietota sistēmas „sāls” – kods, kas nodarbojas ar ciltskoka apstrādāšanu.

Programmatūras Prasību Specifikācija

1. Ievads

1.1. Nolūks

Šī dokumenta mērķis ir aprakstīt www.mesesam.lv uzbūves specifiku, lai apzinātu visas programmatūras prasības, kā arī radītu pamatu detalizētam programmprodukta projektējumam. Dokumenta mērķauditorija ir www.mesesam.lv izstrādātāji un pasūtītāji.

1.2. Darbības sfēra

Mājas lapas nosaukums un tā adrese tīmeklī ir www.mesesam.lv un tā ir paredzēta, lai ikviens interesents varētu izveidot un, iespējams, apzināt savus radu rakstus (ciltskoku). Tāpat sistēmu www.mesesam.lv var izmantot kā ģimenes pārvaldīšanas rīku ar kura palīdzību var atstāt dažādus paziņojumus galvenajā lapā un dalīties ar draugiem publiskojot fotogrāfijas un ierakstus un uzzinot ko viņi šobrīd dara ar citu sociālo tīklu palīdzību.

Lai varētu izmantot sistēmu www.mesesam.lv ir nepieciešams reģistrēties. Visi pamata pakalpojumi ir paredzēti bezmaksas, taču sistēmā jāparedz iespēja bez liekām pūlēm pievienot arī dažādus maksas pakalpojumus, kuri varētu tikt ieviesti nākotnē.

1.3. Saistība ar citiem dokumentiem

Uz šī dokumenta pamata tiks veidots Programmatūras Projektējuma Apraksts, Testēšanas Dokumentācija un Lietotāja Dokumentācija.

2. Vispārējais apraksts

2.1. Produkta Perspektīva

www.mesesam.lv tiek plānota kā tīmekļa vietne, kur lietotāji var veidot un apskatīt savus radu rakstus tādējādi palīdzot sev, saviem bērniem, mazbērniem un vēl tālākas nākotnes atvasēm, apzināt savus senčus. Domājot par tagadni tiek paredzēta iespēja pārvaldīt savu ģimeni publicējot dažādus atgādinājumus, atvieglojot komunikāciju ar saviem ģimenes locekļiem, kā arī publiskot savas fotogrāfijas, lai ģimenes draugi un attālāki radi varētu apskatīties kā šobrīd draugiem/radiem iet un ko viņi šobrīd dara.

Katram atsevišķam lietotājam tiek paredzēta iespēja rakstīt dažādus ierakstus (emuārus vai dienasgrāmatas), publicēt bildes un apzināt savus radus. Lietotāji visas augstāk minētās darbības varēs veikt, ja vien viņiem ir interneta pieslēgums un interneta pārlūkprogramma. Visa lapas – lietotāja mijiedarbība notiks caur interneta pārlūkprogrammu.

2.2. Produkta funkcijas

Sistēmai jāuztur sevī vairāki atsevišķi moduļi, kur katram modulim jāietver sevī visas konkrētā moduļa funkcijas. Sistēmai jāuztur sekojošie moduļi:

- Lietotāja modulis
- Komentāru modulis

- Galerijas modulis
- Ātro ziņu modulis
- Ciltskoka modulis
- Dienasgrāmatu (emuāru) modulis

2.2.1. Lietotāja moduļa funkcijas

- Jauna lietotāja reģistrācija – www.mesesam.lv sniegtos pakalpojumus drīkst izmantot tikai reģistrēti lietotāji, tādējādi jāparedz iespēja lietotājus reģistrēt sistēmā.
- Lietotāja iezurnalēšanās – lai piekļūtu saviem datiem, lietotājam ir jāiežurnalējas sistēmā, izmantojot savu e-pastu un paroli.
- Lietotāja atteikšanās – lai samazinātu risku ar cita profila nozagšanu, jāparedz iespēja lietotājam iziet no sava konta
- Lietotāja datu maiņa – lai nodrošinātu pēc iespējas labāku cilvēku atrašanu un atpazīšanu tos meklējot jābūt iespējai izmainīt savus profila datus, lai tie atspoguļotu tieši konkrētā lietotāja būtību.
- Lietotāja paroles maiņa – ja lietotājs aizmirst vai vēlas nomainīt savu paroli, tad jāparedz iespēja to izmainīt, ja vien lietotājs var pierādīt, ka tas ir viņa profils
- Lietotāju meklēšana – lai varētu izveidot ciltskoku universālāku un noderīgāku, jāparedz iespēja meklēt lietotājus, kuri ir reģistrējušies www.mesesam.lv sistēmā.

2.2.2. Komentāru moduļa funkcijas

- Komentāra pievienošana – jābūt iespējai pievienot komentārus nepieciešamajās vietās, kā, piemēram, pie galerijas bildēm, tādēļ komentāru sistēmai jābūt portablai un viegli modificējamai vai viegli pievienojamai jebkurā vietā.
- Komentāru lasīšana – jāparedz iespēja lasīt iesūtītos komentārus konkrētā vietā

2.2.3. Galerijas moduļa funkcijas

- Galerijas izveidošana – jebkuram lietotājam jāvar pievienot galerija, lai varētu parādīt savu dzīvi citiem lietotājiem/ģimenes locekļiem
- Bilžu pievienošana galerijai – ikvienā, lietotāja izveidotā, galerijā jāvar pievienot lietotāja izvēlēts skaits bilžu.
- Galerijas labošana – ikvienas galerijas datus un bildes jāvar izlabot tam lietotājam, kurš to galeriju ir izveidojis
- Galerijas apskate – jebkuram lietotājam jāvar apskatīt jebkuru galeriju.
- Lietotāja galeriju saraksta apskate – jebkuram lietotājam jāvar apskatīt jebkura cita, sistēmā www.mesesam.lv, reģistrēta lietotāja galeriju saraksti ar iespēju atvērt tos.

2.2.4. Ātro ziņu moduļa funkcijas

- Ātro ziņu pievienošana – jebkuram lietotājam jāvar pievienot ātrā ziņa
- Ātro ziņu dzēšana – jebkuram lietotājam jāvar izdzēst savas un grupu, kurā lietotājs atrodas, lietotāju ātrās ziņas.
- Ātro ziņu apskate – lietotājam jāvar redzēt visas savas un visas grupu, kurā lietotājs atrodas, lietotāju ātrās ziņas.

2.2.5. Ciltskoka moduļa funkcijas

- Ciltskoka izveide – jebkuram reģistrētam lietotājam ir jābūt savam, vienam, ciltskokam
- Nezināma ciltskoka locekļa pievienošana – jābūt iespējai pievienot ciltskoka locekli ciltskokam, pat ja tāds lietotājs nav reģistrējies sistēmā www.mesesam.lv
- Zināma ciltskoka locekļa pievienošana – jābūt iespējai pievienot ciltskoka locekli, ņemot tā datus no kāda lietotāja sistēmā www.mesesam.lv
- Ciltskoka locekļa meklēšana pievienošanas brīdī – jābūt iespējai meklēt vai tāds lietotājs eksistē www.mesesam.lv sistēmā, pēc lietotāja ievadītajiem datiem

2.2.6. Dienasgrāmatas moduļa funkcijas

- Dienasgrāmatas ieraksta pievienošana – jebkuram lietotājam jāvar pievienot dienasgrāmatas ieraksts
- Dienasgrāmatas ieraksta labošana – jebkuram lietotājam jāvar izlabot savu dienasgrāmatu ieraksta datus.
- Dienasgrāmatu ierakstu apskate – jebkuram lietotājam jāvar apskatīt jebkura cita lietotāja dienasgrāmatu ieraksti.
- Dienasgrāmatu ierakstu formatēšana – pievienojot ierakstu jāvar to noformēt izmantojot lietotājam saprotamu veidu.

2.3. Lietotāja raksturiezīmes

Lietotājus varētu raksturot, kā modernus cilvēkus, kuriem nav svešs interneta jēdziens un viņi prot to izmantot. Standarta lietotājs, kurš apmeklēs un izmantos sistēmu www.mesesam.lv varētu interesēties par dažādiem sociālajiem tīkliem un/vai par savas ģimenes saknēm un viņam ir svarīgi tas, lai viņa bērni zinātu savus vecākus un tālākus radus.

2.4. Vispārējie ierobežojumi

- Sistēmas www.mesesam.lv ātrums ir lielā mērā atkarīgs no tā cik ātrs interneta pieslēgums ir sistēmas lietotājam
- Lai varētu izmantot sistēmu www.mesesam.lv lietotāja pārlūkam nepieciešams iespējot JavaScript atbalstu.

3. Funkcionālās prasības

3.1. Jauna lietotāja reģistrācija

Mērķis:

Jauna lietotāja pievienošana www.mesesam.lv sistēmā un lietotāja ievadīto datu saglabāšana datu bāzē.

Ievaddati:

- Vārds (līdz 150 simboliem, obligāts)
- Uzvārds (līdz 150 simboliem, obligāts)
- E-pasts (līdz 200 simboliem, obligāts, unikāls datu bāzē)

Apstrāde:

- Ievaddatu validācija (validācijas vadlīnijas paskaidrotas pie ievaddatiem)
- Unikālas paroles uzģenerēšana
- Datu saglabāšana datu bāzē

Izvadati:

Ja lietotāja ievadītie dati veiksmīgi iziet visus apstrādes posmus, tad lietotājam izvada paziņojumu, ka viņš ir veiksmīgi reģistrējies sistēmā, kā arī izvada viņam uzģenerēto paroli, lūdzot to saglabāt vai nomainīt.

Kļūdu paziņojumi:

- Visiem laukiem jābūt obligāti aizpildītiem
- Cilvēks ar šādu e-pastu jau ir reģistrējies

3.2. Lietotāja iežurnālēšanās

Mērķis:

Ielaist lietotāju sistēmā www.mesesam.lv un ļaut viņam manipulēt ar visiem saviem datiem.

Ievadati:

- E-pasts
- Parole

Apstrāde:

- Pārbauda vai lietotājam ar ievadīto e-pastu atbilst ievadītā parole
- Ja ievadītā parole atbilst ievadītajam e-pastam, tad uzstāda nepieciešamos sesijas mainīgos un ielaiž lietotāju sistēmā

Kļūdu paziņojumi:

- Kļūda e-pastā vai parolē. Lūdzu mēģiniet vēlreiz.

3.3. Lietotāja atteikšanās

Mērķis:

Jāatbrīvo visi lietotāja privātie dati, lai neviens cits nevarētu izmantot lietotāja profilu.

Apstrāde:

- Lietotāja sesijas mainīgo izdzēšana
- Lietotāja atgriešana sākuma lapā

3.4. Lietotāja datu maiņa

Mērķis:

Izmainīt lietotāja profila datus no vecajiem uz jaunajiem, lietotāja ievadītajiem, profila datiem.

Ievadati:

- Vārds (līdz 150 simboliem)
- Uzvārds (līdz 150 simboliem)
- Dzimšanas datums (formā dd/mm/yyyy)
- Īsumā par mani (līdz 250 simboliem)
- Mana biogrāfija

Apstrāde:

- Ievaddatu validācija (validācijas vadlīnijas aprakstītas pie ievaddatiem)
- Izmainīto lauku vērtības tiek saglabātas datu bāzē

Izvadati:

Ja visi apstrādes procesi iziet veiksmīgi, tad izvada paziņojumu, ka viss ir izdevies.

Kļūdu paziņojumi:

- Ievadītais dzimšanas datums nav pareizā formātā.

3.5. Lietotāja paroles maiņa

Mērķis:

Nomainīt lietotāja esošo paroli uz jauno, lietotāja ievadīto, paroli.

Ievadati:

- Vecā parole (obligāts)
- Jaunā parole (obligāts)

- Jaunā parole atkārtoti (obligāts)

Apstrāde:

- Ievaddatu validācija (validācijas vadlīnijas aprakstītas pie ievaddatiem)
- Jaunās paroles šifrēšana
- Vecās paroles nomaiņa ar jauno paroli datubāzē

Izvaddati:

Ja visi apstrādes procesi ir izieti veiksmīgi, tad izvada paziņojumu par to.

Kļūdu paziņojumi:

- Vecā parole nesakrīt ar ievadīto paroli
- Jaunās paroles nesakrīt

3.6. Lietotāju meklēšana

Mērķis:

Atrast lietotājus balstoties uz lietotāja ievadītajiem datiem.

Ievaddati:

- Vārds
- Uzvārds

Apstrāde:

- Lietotāju meklēšanas izsaukuma sastādīšana, balstoties uz ievadītajiem datiem
- Lietotāju meklēšana balstoties uz ievadītajiem datiem

Izvaddati:

Izvada visus tos lietotājus, kuru informācija sakrīt ar lietotāja ievadīto informāciju.

3.7. Komentāra pievienošana

Mērķis:

Pievienot komentāru, lietotāja izvēlētajā vietā ar lietotāja ievadītajiem datiem.

Ievaddati:

- Komentāra teksts

Apstrāde:

- Komentāra saglabāšana datu bāzē
- Saglabātā komentāra datu iegūšana no datu bāzes

Izvaddati:

Ievadītais komentārs noformētā veidā.

3.8. Komentāru lasīšana

Mērķis:

Izvadīt visus atbilstošos komentārus lietotāja esošajā vietā.

Apstrāde:

- Nepieciešamo komentāru atlasīšana balstoties no lietotāja atrašanās vietas
- Komentāru noformatēšana

Izvaddati:

Visi komentāri, kuri atbilda pieprasījuma kritērijiem.

3.9. Galerijas izveidošana

Mērķis:

Izveidot galeriju ar lietotāja ievadītajiem datiem.

Ievaddati:

- Nosaukums (līdz 200 simboliem, obligāts)
- Apraksts

Apstrāde:

- Ievaddatu validācija (validācijas vadlīnijas aprakstītas pie ievaddatiem)
- Galerijas saglabāšana datu bāzē

- Attēlu augšupielādes formas uzģenerēšana

Izvaddati:

Ja visi apstrādes procesi ir izieti veiksmīgi, izvada attēlu augšupielādes formu.

Kļūdu paziņojumi:

- Nav aizpildīta nosaukuma aile
- Nosaukums nedrīkst būt garāks par 200 simboliem

3.10. Bilžu pievienošana galerijā

Mērķis:

Pievienot (augšupielādēt) esošai lietotāja galerijai bildes.

Ievaddati:

- Augšupielādējamā bilde
- Bildes apraksts

Apstrāde:

- Augšupielādē lietotāja izvēlēto bildi uz servera
- Samazina augšupielādēto bildi, līdz maksimāli pieļaujamajam izmēram
- Izveido bildes sīkattēlu
- Uzģenerē bildes nosaukumu un piešķir to bildei
- Saglabā bildes nosaukumu un aprakstu datu bāzē, atbilstošajai galerijai

Izvaddati:

Ja visi apstrādes procesi izpildās veiksmīgi, tad parāda augšupielādētās bildes sīkattēlu un izvada paziņojumu par to, ka bilde ir veiksmīgi augšupielādēta.

3.11. Galerijas labošana

Mērķis:

Labot kādas lietotāja galerijas datus ar lietotāja ievadītajām vērtībām un augšupielādētajām bildēm.

Ievaddati:

- Bildes apraksts
- Augšupielādējamā bilde

Apstrāde:

- Identiska bilžu pievienošanai galerijā

Izvaddati:

Identiski bilžu pievienošanai galerijā.

3.12. Galerijas apskate

Mērķis:

Parādīt lietotājam kāda cita lietotāja galeriju.

Ievaddati:

- Izvēlētā galerija

Apstrāde:

- Nepieciešamās galerijas atlasīšana no datu bāzes
- Galerijas bilžu pareiza noformatēšana un attēlošana

Izvaddati:

Lietotāja izvēlētā galerija.

3.13. Lietotāja galeriju saraksta apskate

Mērķis:

Kāda lietotāja galerijas saraksta izvadīšana.

Ievaddati:

- Lietotājs, kura galerijas sarakstu vajag izvadīt

Apstrāde:

- Visu galerijas datu atlase izvēlētajam lietotājam
- Galerijas saraksta noformatēšana

Izvaddati:

Izvēlēta lietotāja galerijas saraksts.

3.14. Ātro ziņu pievienošana

Mērķis:

Pievienot jaunu ātro ziņu ar lietotāja izvēlētajiem datiem.

Ievaddati:

- Teksts (līdz 200 simboliem)
- Krāsa (6 simboli)

Apstrāde:

- Ievadīto datu saglabāšana datu bāzē
- Atlasa no datu bāzes jauno ātro ziņu un noformatē to

Izvaddati:

Izvada jauno ātro ziņu un izvada paziņojumu par veiksmīgu pievienošanu.

3.15. Ātro ziņu dzēšana

Mērķis:

Izdzēst savu, lietotāja izvēlētu, ātro ziņu, kā arī citu lietotāju (to, kuri atrodas grupās, kurās atrodas pats lietotājs) ātrās ziņas.

Ievaddati:

- Lietotāja izvēlēta ātrā ziņa

Apstrāde:

- Izdzēš atbilstošās ātrās ziņas ierakstu no datu bāzes
- Noņem ātro ziņu no ātro ziņu laukuma

Izvaddati:

Izvada paziņojumu par to, ka ir notikusi veiksmīga ātrās ziņas dzēšana

3.16. Ātro ziņu apskate

Mērķis:

Ļaut lietotājam apskatīt visas ātrās ziņas, kuras ir izveidojis vai nu pats lietotājs, vai arī kāds no lietotājiem, kurš atrodas grupā, kurā atrodas arī ātro ziņu apskates lietotājs.

Apstrāde:

- Visu ātro ziņu atlasīšana no ātro ziņu pieprasījuma lietotāja, kā arī visiem tiem lietotājiem, kuri atrodas grupās, kurās atrodas ātro ziņu pieprasījuma lietotājs
- Ātro ziņu noformatēšana

Izvaddati:

Izvada visas pieprasītās ziņas.

3.17. Ciltskoka izveide

Mērķis:

Izveidot laukumu, kur lietotājam veidot savu ciltskoku, bez jebkādas lietotāja iejaukšanās.

Apstrāde:

- Noklusēto vērtību piešķiršana, lai izveidotu laukumu ciltskoka veidošanai.

Izvaddati:

Laukums, kur lietotājs var sākt veidot savu ciltskoku.

3.18. Nezināma ciltskoka locekļa pievienošana

Mērķis:

Pievienot lietotāja ciltskokam locekli ar atbilstošajiem lietotāja ievadītajiem datiem, ja tāds nav reģistrējies sistēmā www.mesesam.lv

Ievaddati:

- Vārds (līdz 150 simboliem)
- Uzvārds (līdz 150 simboliem)
- Tips

Apstrāde:

- Atkarībā no izvēlēta tipa tiek izrēķināta vieta, kur vajadzētu novietoties ciltskoka locekļa laukumam
- Ja nepieciešams, pārbīda visus ciltskoka locekļa laukumus, lai būtu vieta jaunajam ciltskoka locekļa laukumam
- Tiek saglabāti ciltskoka locekļa dati datu bāzē
- Tiek uzģenerēts ciltskoka locekļa laukums

Izvaddati:

Izrēķinātajā vietā tiek parādīts ciltskoka locekļa laukums ar visiem lietotāja ievadītajiem datiem.

3.19. Zināma ciltskoka locekļa pievienošana

Mērķis:

Pievienot lietotāja ciltskokam locekli, kurš sistēmā www.mesesam.lv eksistē kā reģistrējies lietotājs un paņemt visus viņu nepieciešamos datus pie ciltskoka rādīšanas.

Ievaddati:

- Vārds (līdz 150 simboliem)
- Uzvārds (līdz 150 simboliem)
- Tips
- Izvēlētais lietotājs no saraksta pēc atlasītajiem datiem

Apstrāde:

- Tiek atlasīti visi lietotāji, kura dati atbilst lietotāja ievadītajiem datiem
- Pēc lietotāja izvēles tiek izveidots laukums izvēlētajam lietotājam
- Nepieciešamības gadījumā tiek pārbīdīti citi ciltskoka locekļi, lai būtu vieta jaunajam ciltskoka locekļa laukumam.
- Tiek saglabāti ciltskoka locekļa dati datu bāzē.
- Tiek uzģenerēts ciltskoka locekļa laukums.

Izvaddati:

Pēc vārda un uzvārda ievades tiek parādīti atlasītais saraksts ar lietotājiem pēc ievadītās informācijas. Kad tiek izvēlēts īstais lietotājs, viņa ciltskoka laukums tiek parādīts izrēķinātajā vietā.

3.20. Ciltskoka locekļa meklēšana pievienošanas brīdī

Mērķis:

Sameklēt visus atbilstošos lietotājus, kuri atbilst cita lietotāja ievadītajiem datiem.

Ievaddati:

- Vārds (līdz 150 simboliem)
- Uzvārds (līdz 150 simboliem)

Apstrāde:

- Tiek veikti visi nepieciešamie vaicājumi datu bāzei, lai atrastu visus lietotājus, kuri atbilst lietotāja ievadītajai informācijai
- Tiek uzģenerēts lietotāju saraksts, kuri atbilst lietotāja ievadītajai informācijai

Izvaddati:

Tiek parādīts saraksts ar lietotājiem, kuri atbilst lietotāja ievadītajiem datiem. Blakus atrasto lietotāju vārdiem un uzvārdiem atrodas podziņa, lai tos varētu pievienot savam ciltskokam.

3.21. Dienasgrāmatas ieraksta pievienošana

Mērķis:

Pievienot dienasgrāmatas ierakstu ar lietotāja ievadītajiem datiem.

Ievaddati:

- Virsraksts (līdz 150 simboliem, obligāts)
- Teksts (obligāts)

Apstrāde:

- Tiek validēti visi lietotāja ievadītie dati (validācijas vadlīnijas aprakstītas pie ievaddatiem)
- Lietotāja ievadītie dati tiek saglabāti datu bāzē

Izvaddati:

Tiek izvadīts lietotāja dienasgrāmatas ieraksts.

Kļūdu paziņojumi:

- Nav aizpildīti visi lauki
 - Ieraksta virsraksts nevar būt garāks par 150 simboliem
-

3.22. Dienasgrāmatas ieraksta labošana

Mērķis:

Atļaut lietotājam labot visus savus dienasgrāmatas ierakstus ar jauniem lietotāja ievaddatiem.

Ievaddati:

- Virsraksts
- Teksts

Apstrāde:

- Visi izmainītie lauki tiek pārrakstīti pāri vecajiem lietotāja ievadītajiem datiem

Izvaddati:

Tiek izvadīts paziņojums par to, ka ir izdevies izmainīt dienasgrāmatas datus.

3.23. Dienasgrāmatu ierakstu apskate

Mērķis:

Atļaut lietotājiem apskatīt jebkura cita un sevis paša izveidotus dienasgrāmatu ierakstus.

Ievaddati:

- Izvēlētais dienasgrāmatas ieraksts

Apstrāde:

- Tiek iegūti dati no datubāzes par atbilstošo dienasgrāmatas ierakstu
- Ievadītā informācija tiek formatēta

Izvaddati:

Tiek izvadīts lietotāja pieprasītais dienasgrāmatas ieraksts.

3.24. Dienasgrāmatas ierakstu formatēšana

Mērķis:

Atļaut lietotājam formatēt savu ievadīto tekstu ērtā un saprotamā veidā.

Ievaddati:

- Noteiktas komandas teksta formatēšanai

Apstrāde:

- Tiek parsēts teksts un formatēšanas komandas tiek aizstātas ar reāliem formatēšanas atslēgas vārdiem, kas izvadīti rādās formatēti

Izvaddati:

Tiek izvadīts atbilstoši formatēts lietotāja teksts.

4. Nefunkcionālās prasības

4.1. Veiktspējas prasības

- Sistēmai jāspēj nodrošināt vismaz 100 lietotāju vienlaicīgu darbu pie servera ar 100Mb/s pieslēgumu
- Sistēmai jāstrādā uz Internet Explorer (sākot no 7 versijas), Mozilla Firefox (sākot no 1.5 versijas), Opera (sākot no 7 versijas), Safari un Google Chrome pārlūkprogrammām. Dizaina atspoguļošana var minimāli mainīties starp dažādiem interneta pārlūkiem, ja vien tas netraucē informācijas uztveri un kopējam dizainam.
- Datu bāzei jāspēj uzturēt vismaz 100 000 ieraksti, bez veiktspējas problēmām.

4.2. Pieejamības prasības

- Sistēmai jābūt 99% pieejamai un strādājošai, ja vien to atļauj hostings.
- Izmaiņas drīkst veikt tikai laikā no 03:00 – 06:00 pēc GMT + 2 laika joslas un tā laikā nepieciešams izvietot atbilstošu paziņojumu mājas lapas sākumā

4.3. Datu drošība

- Parolēm datu bāzē jāglabājas šifrētā veidā, kas ir sarežģītāks par parastu paroles kriptēšanu.
- Datu bāzē nedrīkst glabāties sensitīva informācija par lietotāju.
- Lai piekļūtu lietotāja datiem nepieciešams ievadīt lietotāja paroli

5. Ārējā saskarne

5.1. Lietotāja saskarne

- Dizainam jābūt viegli maināmam un pielāgojamam, vienkārši ar stila failu nomaiņu
- Lapai jābūt sadalītai trīs kolonnās, no kurām vidējā ir galvenā un kura mainās atkarībā no lietotāja darbībām. Malējām kolonnām jāsaturs dati par lietotāju, navigāciju un jaunumiem.
- Lapā jāizmanto vienādas nokrāsas krāsas līdzīgiem objektiem.
- Jābūt iespējai uzreiz atšķirt informatīvus ziņojumus no kļūdu paziņojumiem un veiksmīgas darbības izpildes paziņojumiem.
- Jāizmanto dažādi efekti objektu un bloku parādīšanai un noslēpšanai. Kādus efektus izmantot un kādas tehnoloģijas izmantot tā veikšanai paliek izpildītāja ziņā
- Jāparedz pēc iespējas vairāk darbības veikt nepārlādējot lapu un atjaunot informāciju automātiski.

Programmatūras projektējuma apraksts

1. Ievads

1.1. Nolūks

Programmatūras projektējuma apraksta nolūks ir parādīt to, kā Programmatūras Prasību Specifikācijā definētās prasības tiks realizētas strādājošā projektā. Programmatūras Projektējuma Apraksts ir paredzēts programmētājiem un projekta vadītājiem.

1.2. Darbības sfēra

Mājas lapas nosaukums un tā adrese tīmeklī ir www.mesesam.lv un tā ir paredzēta, lai ikviens interesents varētu izveidot un, iespējams, apzināt savus radu rakstus (ciltskoku). Tāpat sistēmu www.mesesam.lv var izmantot kā ģimenes pārvaldīšanas rīku ar kura palīdzību var atstāt dažādus paziņojumus galvenajā lapā un dalīties ar draugiem publiskojot fotogrāfijas un ierakstus un uzzinot ko viņi šobrīd dara ar citu sociālo tīklu palīdzību.

Lai varētu izmantot sistēmu www.mesesam.lv ir nepieciešams reģistrēties. Visi pamata pakalpojumi ir paredzēti bezmaksas, taču sistēmā jāparedz iespēja bez liekām pūlēm pievienot arī dažādus maksas pakalpojumus, kuri varētu tikt ieviesti nākotnē.

2. Dekompozīcijas apraksts

2.1. Moduļu dekompozīcija

Vairāki moduļi sistēmā sastāv no vairākām daļām:

- Funkcionālās, servera puses daļas – tiek rakstīts PHP valodā un nodrošina datu iegūšanas, mijiedarbības un apstrādes
- Dizaina, klienta puses daļas – tiek rakstīts HTML/CSS marķējuma valodā, izmantojot JavaScript skriptēšanas valodu. Nodrošina sistēmas vizuālo izskatu un funkcionālās daļas padoto datu savietošanu pareizajos laukos
- Mijiedarbības, klienta puses daļas – tiek rakstīts JavaScript valodā izmantojot jQuery, JavaScript bibliotēku. Tehnoloģija tiek pazīta kā AJAX. Nodrošina klienta dialogu ar serveri, lai novērstu nepārtrauktas lapas pārlādes pie katras lietotāja darbības.

Augstāk minētais sadalījums tiek izmantots, lai nodrošinātu koda nošķiršanu no dizaina marķējuma un, lai nodrošinātu to, lai pie katras lietotāja darbības netiktu veikta lapas pārlāde. Sistēmas izstrādē tiek izmantots kvalifikācijas darba autora izdomāts/praktizēts failu strukturēšanas princips un moduļu mijiedarbības princips, kas līdzinās Model-View-Controller „Design Pattern”, tikai bez veidņu datņu izmantošanas dizaina daļā.

2.1.1. Datu bāzes savienojuma modulis

Nolūks:

Nodrošina savienojumu ar MySQL datu bāzes serveri, kā arī atvieglo dažādu SQL vaicājumu izveidošanu un drošību pret SQL Injections. Tā kā šis modulis ir abstrakts datu tips, tad bez citu moduļu palīdzības viņu izmantot nevar.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Nodrošina savienojumu ar MySQL datu bāzes serveri
- Atvieglo SQL vaicājumu izveidi
- Nodrošina sistēmu pret SQL injekcijām

Pakļautība:

Šim modelim pakļaujas visi tie moduļi, kuri savu datu iegūšanai un saglabāšanai izmanto MySQL datu bāzes serveri.

2.1.2. Attēlu modulis

Nolūks:

Nodrošina darbu ar attēliem.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Attēla izmēra maiņu
- Attēla saglabāšanu
- Attēlu dzēšanu

Pakļautība:

Šim modulim pakļaujas Galerijas modulis un Lietotāja modulis, jo viņiem ir vajadzīga bilžu apstrāde. Pēc būtības, modulim pakļaujas visi tie moduļi, kuri izmanto attēlus.

2.1.3. Galerijas modulis

Nolūks:

Nodrošina darbu ar galerijām.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Nodrošina galeriju izveidi
- Nodrošina galeriju labošanu
- Nodrošina galeriju rādīšanu

Atkarības:

Modulis ir pakļauts Datu bāzes savienojuma modulim, Attēlu modulim un Komentāru modulim.

2.1.4. Komentāru modulis

Nolūks:

Nodrošina jebkura veida komentāru pievienošanu.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Komentāru pievienošana
- Komentāru lasīšana
- Ērta un vienkārša komentāru atbalsta pievienošana jebkurā vēlamajā vietā

Atkarības:

Modulis ir pakļauts Datu bāzes savienojuma modulim.

2.1.5. Grupu modulis

Nolūks:

Nodrošina darbības ar grupām

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Grupu izveidošanu
- Pievienošanu grupām
- Piekļuves līmeņu sadale

Atkarības:

Modulis ir pakļauts Datu bāzes savienojumu modulim.

2.1.6. Kreisā paneļa modulis

Nolūks:

Nodrošina kreiso paneļu attēlošanu.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Navigācijas attēlošanu
- Profila informācijas attēlošanu

Atkarības:

Modulis ir atkarīgs no Datu bāzes savienojuma moduļa un lietotāja moduļa.

2.1.7. Labā paneļa modulis

Nolūks:

Nodrošina labās puses paneļu attēlošanu.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Meklēšanas paneļa attēlošana
- Grupu ielūgumu paneļa attēlošana

Atkarības:

Modulis ir atkarīgs no Grupu moduļa.

2.1.8. XML modulis

Nolūks:

Nodrošina dialogu starp lietotāju un serveri.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- XML failu ģenerēšanu
- XML failu automātisku ģenerēšanu
- XML failu izveide un atgriešana

Pakļautība:

Modulim ir pakļauta dizaina daļa un servera puses daļa, jo modulis nodrošina saskarsmi starp abiem.

2.1.9. Ātro ziņu modulis

Nolūks:

Nodrošina ātro ziņu izmantošanu.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Ātro ziņu pievienošana
- Ātro ziņu dzēšana
- Ātro ziņu parādīšana

Atkarība:

Modulis ir atkarīgs no Datu bāzes savienojuma moduļa un Grupu moduļa.

2.1.10. Ciltskoka modulis

Nolūks:

Nodrošina darbības ar ciltskoku.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Ciltskoka izveidošanu
- Ciltskoka locekļu pievienošanu

Pakļautība:

Modulim ir pakļauts Grupu modulis.

2.1.11. Lietotāju modulis

Nolūks:

Nodrošina darbu ar lietotājiem.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Lietotāju reģistrāciju
- Lietotāju iežurnālēšanu
- Lietotāju iziešanu
- Paroļu ģenerēšanu
- Profila datu labošanu
- Paroles maiņu
- Lietotāju meklēšanu

Atkarības:

Modulis ir atkarīgs no datu bāzes savienojuma moduļa.

2.1.12. Dienasgrāmatu modulis

Nolūks:

Nodrošina lietotāja darbu ar dienasgrāmatām.

Tips:

Modulis

Funkcija:

Modulis nodrošina vairākas funkcijas:

- Dienasgrāmatu pievienošanu
- Dienasgrāmatu labošanu
- Dienasgrāmatu apskati

-
- Dienasgrāmatu formatēšanu

Atkarība:

Modulis ir atkarīgs no Datu bāzes moduļa.

2.1.13. Vadītāja modulis

Nolūks:

Nodrošina POST un GET pieprasījumu apstrādi.

Tips:

Modulis

Funkcija:

Izsauc nepieciešamo moduļu funkcijas skatoties pēc POST un GET parametriem.

2.2. Datu dekompozīcija

2.2.1. Users

Tabula, kurā glabājas informācija par sistēmas www.mesesam.lv lietotājiem.

2.2.2. Relations

Tabula, kurā glabājas informācija par visām ciltskoka saistībām starp ciltskoka locekļiem. Satur norādes uz ciltskoka locekļu datiem.

2.2.3. Relation Members

Tabula, kurā glabājas informācija par ciltskoka locekļiem. Ja ciltskoka lietotājs ir reģistrējies sistēmā www.mesesam.lv, tad glabājas norāde uz viņa ierakstu Users tabulā.

2.2.4. Postit

Tabula, kurā glabājas informācija par ātro ziņu ierakstiem.

2.2.5. Groups

Tabula, kurā glabājas informācija par grupām un norāde uz Users tabulu grupas izveidotājam.

2.2.6. Group Members

Tabula, kurā glabājas informācija par grupu dalībniekiem. Satur norādi uz grupu un lietotāja ierakstu.

2.2.7. Gallery

Tabula, kurā glabājas informācija par galerijām. Satur norādi uz lietotāju, kuram galerija pieder.

2.2.8. Gallery Pictures

Tabula, kurā glabājas informācija par bildēm, kuras atrodas galerijās. Satur norādi uz galeriju, kurai pieder attēls.

2.2.9. Gallery Comments

Tabula, kurā glabājas informācija par komentāriem, kuri pievienoti galeriju bildēm. Satur norādi uz attēlu, kuram pievienots komentārs.

2.2.10. Blog

Tabula, kurā glabājas informācija par lietotāju dienasgrāmatām. Satur norādi uz lietotāju, kurš pievienoja dienasgrāmatu.

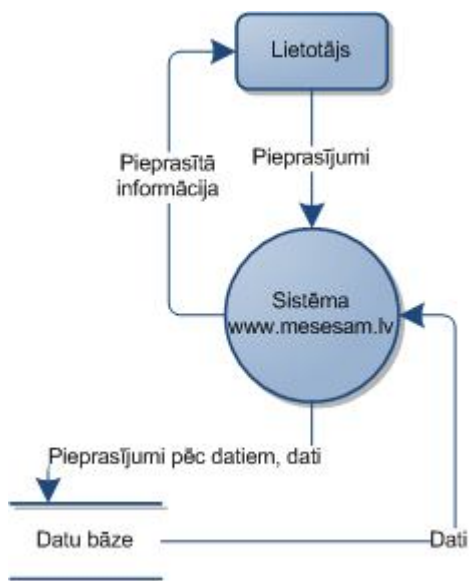
2.2.11. Blog Comments

Tabula, kurā glabājas informācija par komentāriem, kuri pievienoti dienasgrāmatu ierakstiem. Satur norādi uz dienasgrāmatas ierakstu, kuram pievienots komentārs.

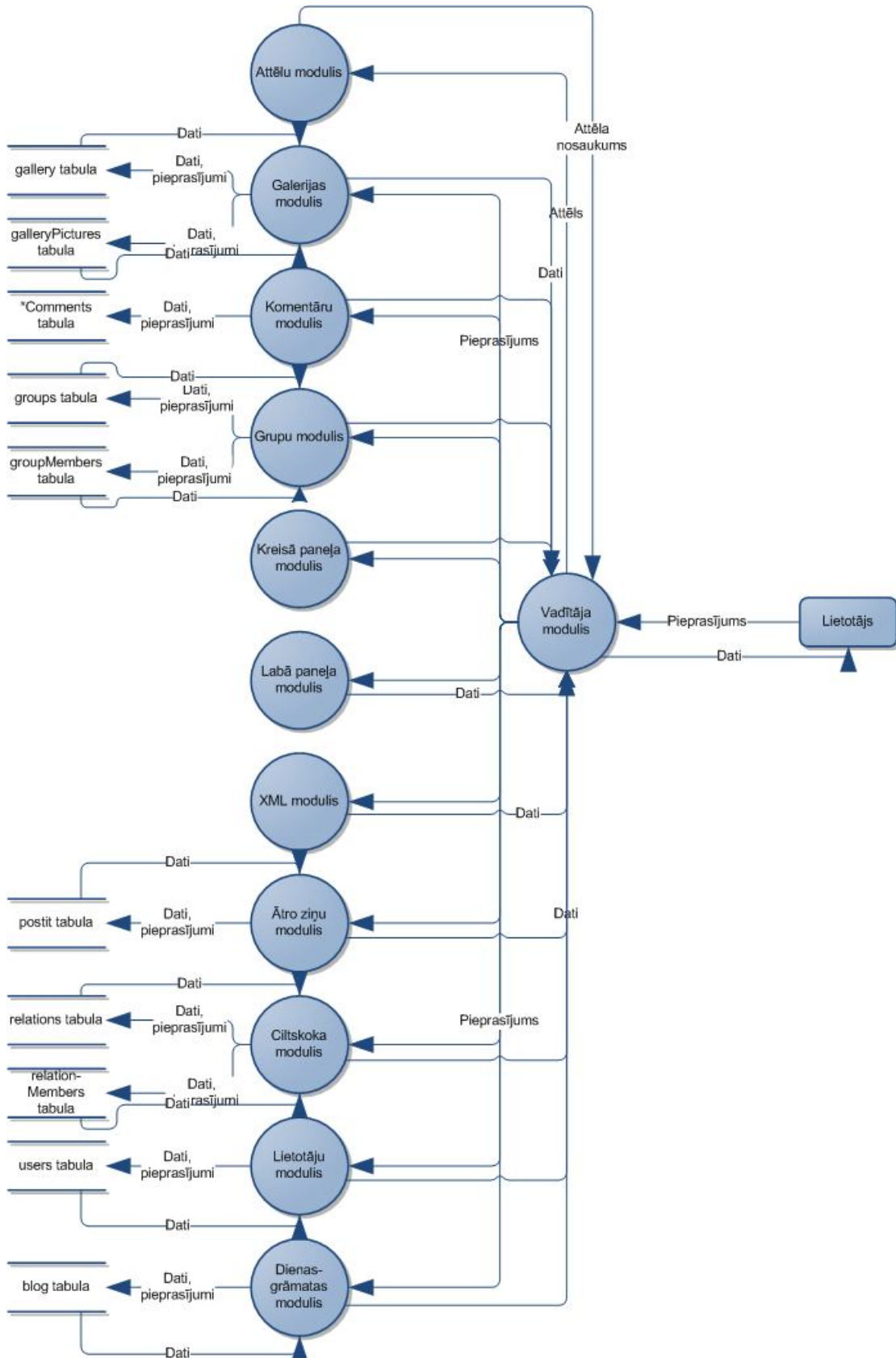
3. Atkarības apraksts

3.1. Starpmodeļu un starpprocesu atkarības

3.1.1. 0. līmenis

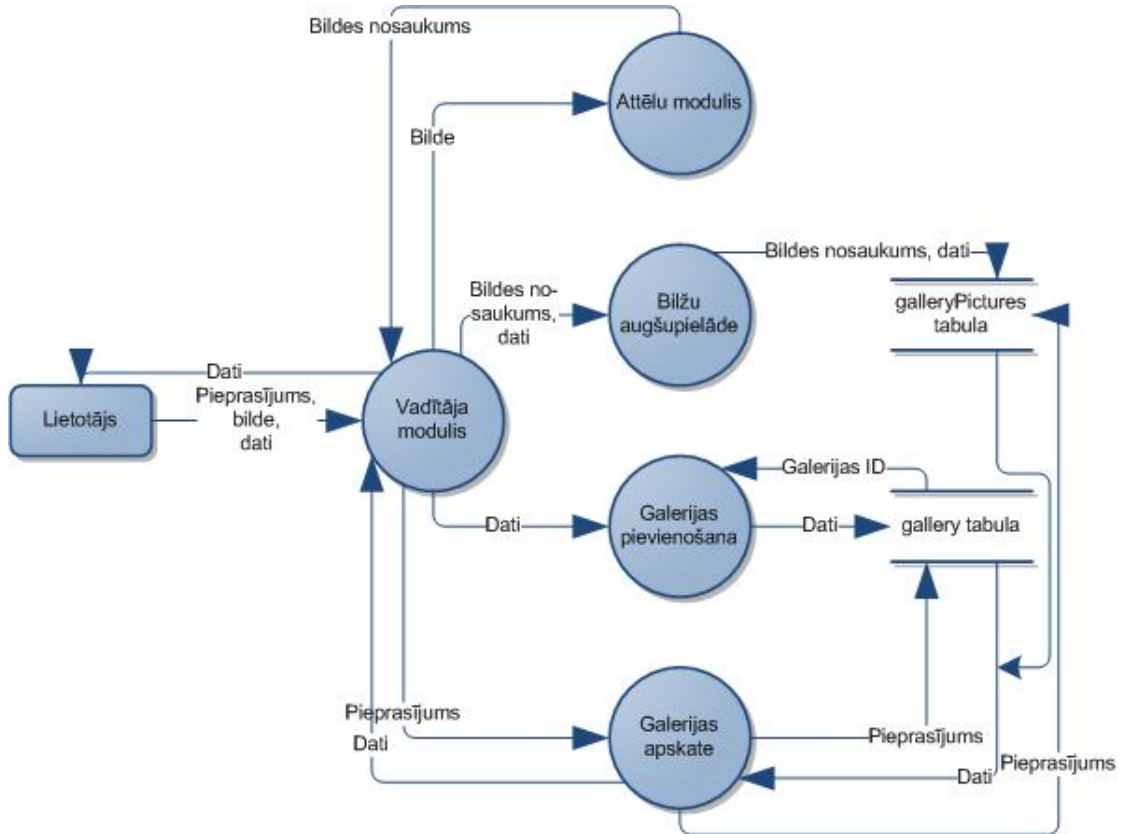


3.1.2. 1. Līmenis

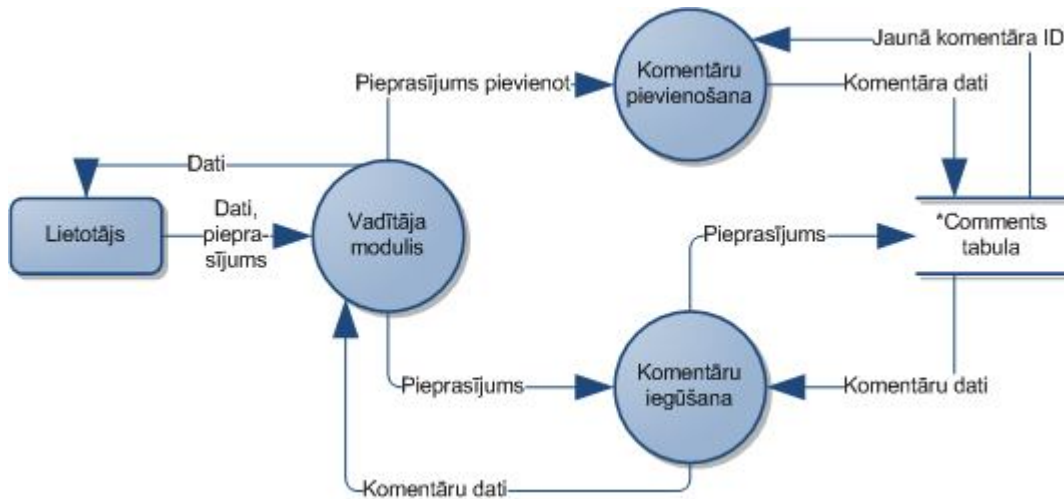


3.1.3. 2. Līmenis

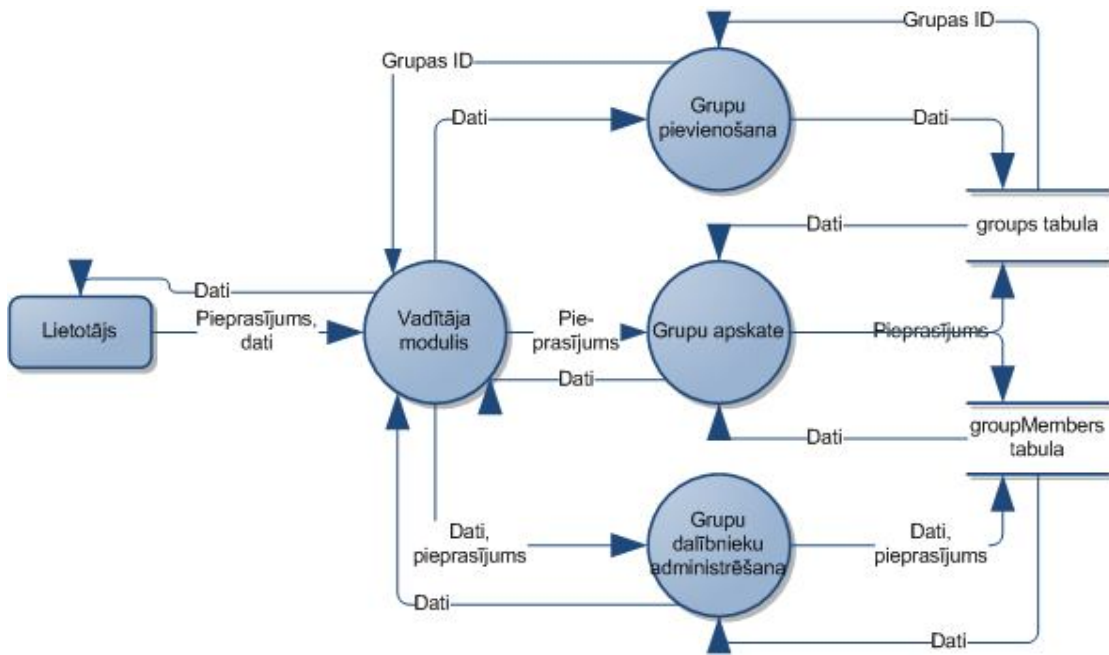
3.1.3.1. Galerijas modulis



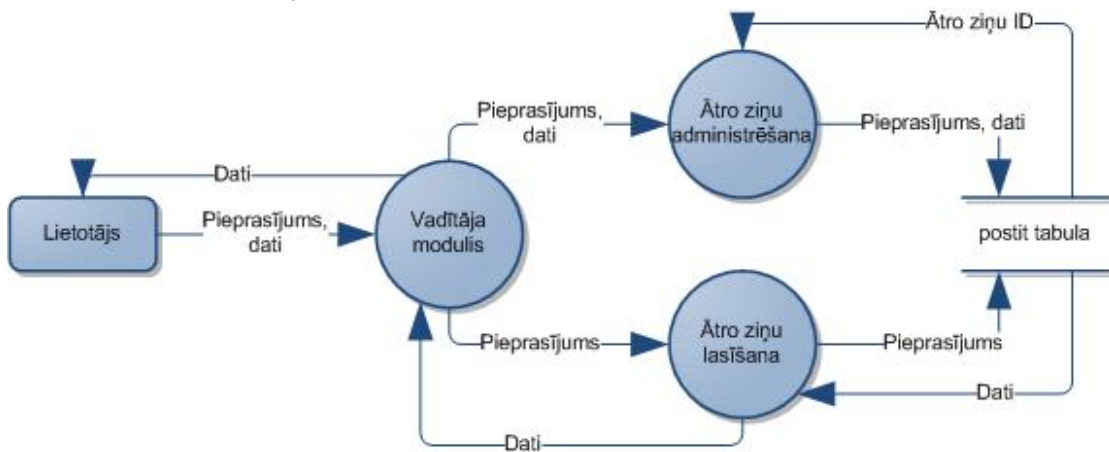
3.1.3.2. Komentāru modulis



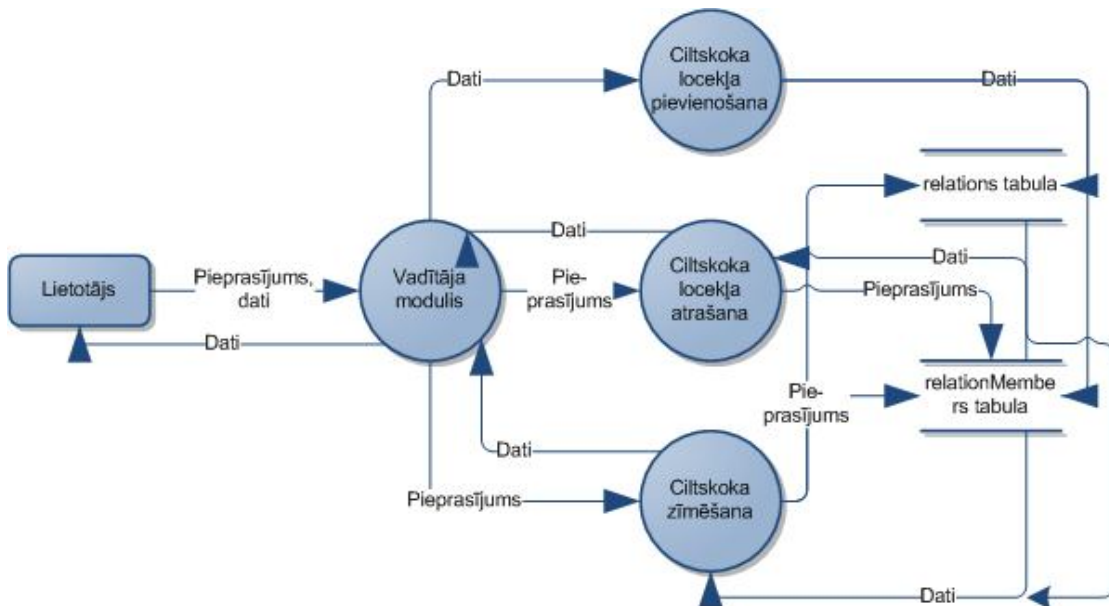
3.1.3.3. Grupu modulis



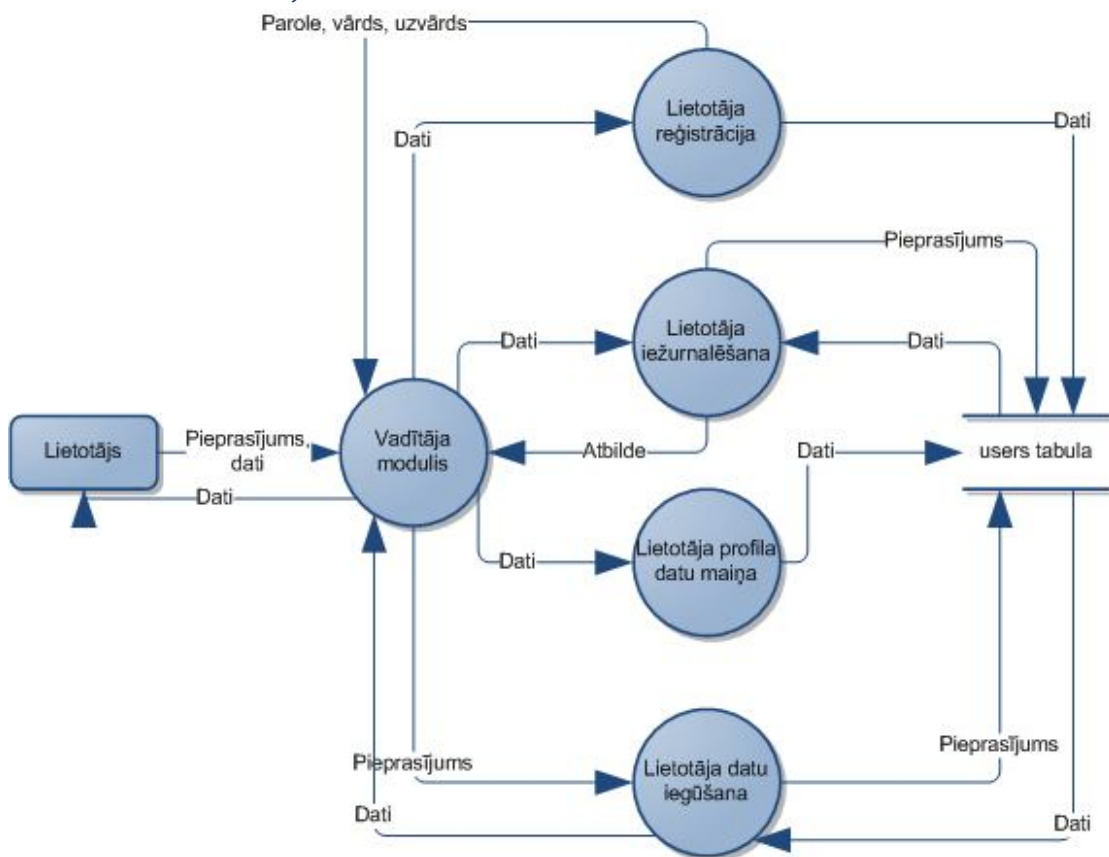
3.1.3.4. Ātro ziņu modulis



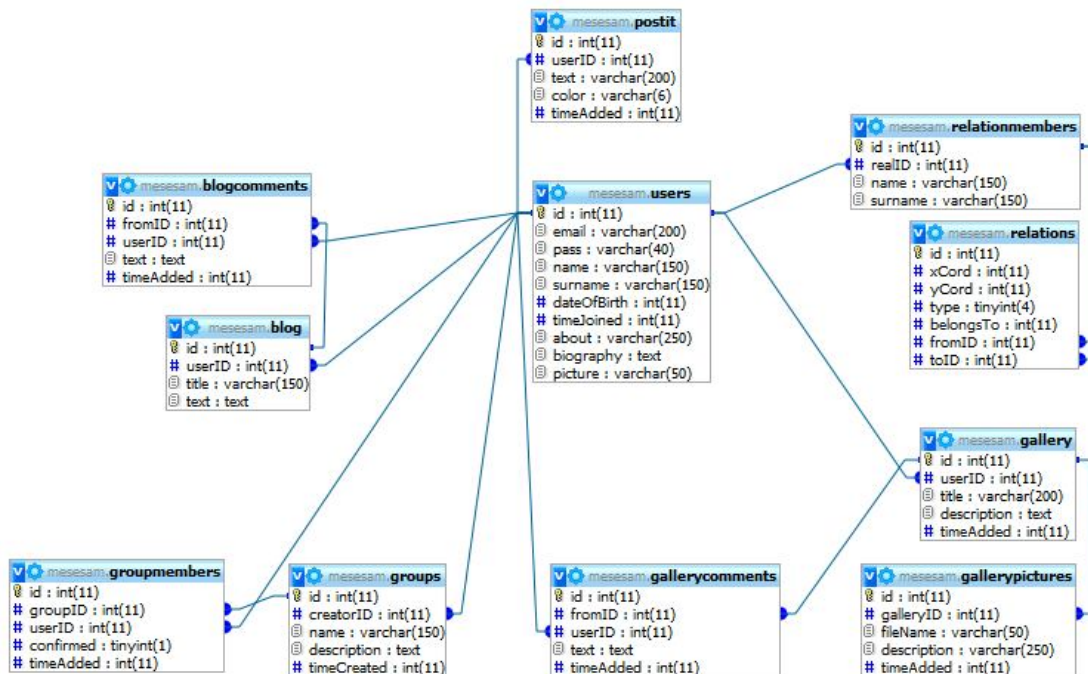
3.1.3.5. Ciltskoka modulis



3.1.3.6. Lietotāju modulis



3.2. Datu atkarības



4. Detalizētais projektējums

4.1. Moduļu detalizēts projektējums

Detalizēti aprakstītas tiek tikai metodes, kuras dara kādu svarīgu darbu un nav sevišķi triviālas. Visu metožu detalizēts izklāsts būtu pārāk ietilpīgs un laiku aizņemošs.

4.1.1. Datu bāzes savienojuma modulis

4.1.1.1. Metode „__construct”

Apraksts:

Metode sagatavo moduli ar noklusētajām vērtībām, kuras var mainīt tikai mainot pašu moduli.

Apstrāde:

Metode katram klases mainīgajam piešķir noklusētās vērtības, lai varētu atvērt savienojumu ar MySQL datu bāzes serveri. Mainīgie kuriem tiek piešķirtas vērtības:

- Server – MySQL servera adrese
- User - MySQL servera lietotājs
- Pass – MySQL servera lietotāja parole
- db – datu bāze, kurā glabājas sistēmas www.mesesam.lv tabulas
- Connection – atvērtais savienojums. Tiek piešķirta vērtība „NULL”, ja savienojums nav atvērts, pretējā gadījumā satur savienojuma identifikatoru.

4.1.1.2. Metode „open”

Apraksts:

Atver savienojumu ar MySQL serveri.

Apstrāde:

- Atver savienojumu (neveiksmes gadījumā izmet kļūdu)
- Uzstāda savienojuma simbolu kodējumu
- Izvēlas datubāzi

Kļūdas:

Ja tiek notverta kļūda pieslēdzoties serverim, tad izmet paskaidrojošu tekstu par to kādēļ tā ir noticis.

4.1.1.3. Metode „close”

Apraksts:

Aizver atvērto savienojumu ar MySQL serveri.

Apstrāde:

- Aizver esošo savienojumu ar MySQL serveri
- Connection mainīgajam uzstāda vērtību „NULL”

4.1.1.4. Metode „executeQuery”

Apraksts:

Izpilda doto SQL komandu (kas paredz datu atgriešanu) MySQL serverī un atgriež iegūtos datus.

Ievade:

- String cmdStr – izpildāmā komanda

Apstrāde:

- Pārbauda vai ir atvērtais savienojums, ja ir atvērts izmanto to, ja nav tad izsauc „open” metodi.
- Saglabā mainīgajā „result” izpildītās komandas rezultātu
- Ja savienojums metodes izsaukšanas brīdī nebija atvērts, tad aizver to, pretējā gadījumā patur to atvērtu.

Izvade:

Tiek atgriezts „result” tipa mainīgais, kurš satur iegūtos rezultātus par padoto SQL komandu.

Kļūdas:

Ja nespēj izpildīt SQL komandu, izmet atbilstošu kļūdas paziņojumu par kļūdas būtību.

4.1.1.5. Metode „executeUpdate”

Apraksts:

Izpilda doto SQL komandu (kura paredz datu maiņu kādā no tabulām) MySQL serverī.

Ievade:

- String cmdStr – izpildāmā komanda

Apstrāde:

- Pārbauda vai ir atvērts savienojums, ja ir atvērts izmanto to, ja nav tad izsauc „open” metodi.
- Izpilda doto komandu MySQL serverī
- Ja savienojums metodes izsaukšanas brīdī nebija atvērts, tad aizver to, pretējā gadījumā patur to atvērtu.

Izvade:

Ja izpildāmā komanda bija „INSERT” tipa, tad atgriež pievienotā rezultāta unikālo atslēgu.

Kļūdas:

Ja nespēj izpildīt SQL komandu, izmet atbilstošu kļūdas paziņojumu par kļūdas būtību.

4.1.1.6. Metode „escapeParams”

Apraksts:

Nodrošina, ka padotie parametri ir droši pret SQL injekcijām.

Ievade:

- Parametri, kurus nepieciešams nodrošināt

Apstrāde:

- Saglabā visus padotos parametrus masīvā
- Katram padotajam parametram nodrošina, ka viņš ir drošs un SQL komandā ierakstāms

Izvade:

Atgriež masīvu ar nodrošinātajiem parametriem.

4.1.1.7. Metode „buildInsert”

Apraksts:

Izveido „INSERT” tipa komandu no dotajiem parametriem.

Ievade:

- String table – tabulas nosaukums, kuram izpildīt „INSERT” tipa komandu
- Array variables – mainīgo nosaukumi
- Array values – mainīgo vērtības

Apstrāde:

- Izveido komandas karkasu
- Katram mainīgajam piešķir atbilstošu vērtību ierakstot to komandā
- Ja komandas vērtība nav *Integer* tipa, tad ieliek to pēdiņās

Izvade:

String cmdStr – izveidotā komanda

4.1.2. Attēlu modulis

4.1.2.1. Metode „__construct”

Apraksts:

Metode, kas piešķir klases mainīgajiem noklusētās vērtības, kuras atbild par attēlu izmēriem.

Ievade:

- *Integer bWidth* – lielās bildes platums (noklusētā vērtība 400)
- *Integer bHeight* – lielās bildes garums (noklusētā vērtība 500)
- *Integer tWidth* – sīkattēla platums (noklusētā vērtība 100)
- *Integer tHeight* – sīkattēla garums (noklusētā vērtība 100)

- *Integer btWidth* – lielā sīkattēla platums (noklusētā vērtība 200)
- *Integer btHeight* – lielā sīkattēla garums (noklusētā vērtība 150)

Apstrāde:

- Katram mainīgajam piešķir padoto vai noklusēto vērtību

4.1.2.2. Metode „uploadProfilePicture”

Apraksts:

Metode, kura saglabā visus nepieciešamos attēlus, kuri nepieciešami profila bildes attēlošanai jebkurā vietā.

Ievade:

- *Array file* – masīvs, kurš satur visus augšupielādētos failus
- *String controlName* – nosaukums zem kā glabājas bilde

Apstrāde:

- Tiek uzģenerēts faila nosaukums
- Tiek izsauktas funkcijas visu bildes izmēru saglabāšanai

Izvade:

Tiek atgriezts lielās bildes nosaukums.

4.1.2.3. Metode „uploadGalleryPicture”

Apraksts:

Metode, kas saglabā augšupielādēto galerijas bildi.

Ievaddati:

- *Array file* – augšupielādēto failu masīvs
- *String controlName* – augšupielādētā faila identifikators

Apstrāde:

- Tiek uzģenerēts faila nosaukums
- Tiek izsauktas funkcijas lielās un lielā sīkattēla bildes izmēru saglabāšanai

Izvade:

Tiek atgriezts lielās galerijas bildes nosaukums.

4.1.2.4. Metode „uploadBigPicture”

Apraksts:

Izveido lielu bildi no padotās attēla.

Ievaddati:

- *Image image* – augšupielādētā bilde
- *String name* – bildes uzģenerētais nosaukums

Apstrāde:

- Tiek atrasts bildes izmērs
- Tiek izskaitļots jaunais bildes izmērs saglabājot malu attiecību
- Tiek izveidots laukums, kur saglabāt bildi
- Vecā bilde tiek samazināta un saglabāta izveidotajā laukumā
- Tiek izsaukta bildes saglabāšanas funkcija „*savePicture*”
- Tiek izdzēsts sagatavotais laukums un padotā bilde

4.1.2.5. Metode „UploadBigThumb”

Apraksts:

Metode izveido lielu sīkattēlu no padotā attēla

Ievaddati:

Identiski ar ievaddatiem no funkcijas *uploadBigPicture*

Apstrāde:

Identiska kā metodē *uploadBigPicture*, tikai ar citiem attēla izmēriem.

4.1.2.6. Metode „uploadSmallThumb”

Apraksts:

Metode izveido mazu sīkattēlu no padotā attēla

Ievaddati:

Identiski ar ievaddatiem no funkcijas *uploadBigPicture*

Apstrāde:

Identiska kā metodē *uploadBigPicture*, tikai malu attiecība tiek pārveidota uz 1:1 un tiek izgriezts attēla centrs.

4.1.2.7. Metode „generateFileName”

Apraksts:

Tiek uzģenerēts unikāls attēla nosaukums.

Ievaddati:

- *String ext* – attēla paplašinājums (noklusētā vērtība „jpg”)
- *String prefix* – prefikss ko pievienot faila nosaukuma sākumā (noklusētā vērtība „”)

Apstrāde:

- Tiek uzģenerēts *String* tipa mainīgais balstoties uz pašreizējo laiku, lietotāja identifikatoru, prefiksu un paplašinājumu
- Daļa no faila nosaukuma tiek nokodēta ar *base64* algoritmu

Izvade:

Atgriež faila nosaukumu.

4.1.2.8. Metode „savePicture”

Apraksts:

Saglabā doto attēlu uz servera ar doto faila nosaukumu.

Ievade:

- *Image image* – saglabājamais attēls
- *String name* – attēla saglabājamais nosaukums
- *String/bool prefix* – prefikss, ko pievienots attēla sākumā (noklusētā vērtība „false”)

Apstrāde:

- Ja *prefix* nav vienāds ar *false*, tad faila nosaukumam pievienot prefiksu un saglabā bildi ar doto nosaukumu un prefiksu, pretējā gadījumā tikai ar doto bildes nosaukumu.

4.1.2.9. Metode „getExtention”

Apraksts:

Iegūst faila paplašinājumu.

Ievade:

- *String name* – faila nosaukums

Apstrāde:

Tiek iegūts faila paplašinājums.

Izvade:

Tiek atgriezts faila paplašinājums.

4.1.2.10. Metode „deleteProfileImages”

Apraksts:

Izdzēš visas profila bildes pēc dotā nosaukuma.

Ievade:

- *String fileName* – lielās bildes nosaukums

Apstrāde:

Tiek izdzēsta gan lielā bilde, gan arī lielais un mazais sīkattēls.

4.1.3. Galerijas modulis

4.1.3.1. Metode „drawAddGallery”

Apraksts:

Pievieno lapā galerijas pievienošanas formu.

Apstrāde:

Pievieno ģenerētajai lapai atbilstošo dizaina failu un javascript failu.

4.1.3.2. Metode „drawUploadGalleryPictureForm”

Apraksts:

Uzzīmē lapā bilžu augšupielādes formu.

Ievade:

- *Integer id* – galerijas identifikators

Apstrāde:

- Attēlo bilžu augšupielādes formu
- Izsauc *drawImageList* javascript funkciju

4.1.3.3. Metode „addGallery”

Apraksts:

Pievieno jaunu galeriju ar lietotāja ievadītajiem datiem.

Ievade:

- *String title* – galerijas nosaukums
- *String description* – galerijas apraksts

Apstrāde:

Ieraksta datu bāzē jaunu galerijas ierakstu ar padotajiem datiem

Izvade:

Atgriež jaunizveidotās galerijas identifikatoru

4.1.3.4. Metode „addGalleryPicture”

Apraksts:

Pievieno jaunu bildi galerijai.

Ievade:

- *Integer gid* – galerijas identifikators
- *String fileName* – attēla nosaukums
- *String description* – attēla apraksts

Apstrāde:

Ieraksta datu bāzē jaunās bildes ierakstu ar atbilstošajiem datiem.

Izvade:

Atgriež pievienotās bildes identifikatoru.

4.1.3.5. Metode „getGalleryList”

Apraksts:

Iegūst kāda lietotāja galeriju sarakstu.

Ievade:

- *Integer uid* – lietotāja identifikators

Apstrāde:

No datu bāzes iegūst galerijas identifikatorus un nosaukumus, kuri pieder padotajam lietotājam.

Izvade:

Atgriež *result* tipa mainīgo, kurš satur pieprasītās vērtības.

4.1.3.6. Metode „drawGalleryList”

Apraksts:

Uzzīmē lapā kāda lietotāja galeriju sarakstu.

Ievade:

- *Integer id* – lietotāja identifikators (noklusētā vērtība 0)

Apstrāde:

- Ja noklusētā vērtība ir 0, tad piešķir jaunu noklusēto vērtību, kura ir vienāda ar lietotāja identifikatoru, kurš pieprasa galeriju sarakstu.
- Iegūst nepieciešamos datus
- Pievieno lapā nepieciešamos dizaina failus

4.1.3.7. Metode „getGalleryMainPicture”

Apraksts:

Iegūst galerijas pirmo bildi.

Ievade:

- *Integer gid* – galerijas identifikators

Apstrāde:

No datu bāzes iegūst pirmo, dotās galerijas, bildi.

Izvade:

Atgriež bildes nosaukumu.

Kļūdas:

- Ja galerijā nav bilžu atgriež *false*

4.1.3.8. Metode „getGalleryPictureCount”

Apraksts:

Iegūst kādas galerijas pievienotu bilžu skaitu.

Ievade:

- *Integer gid* – galerijas identifikators

Apstrāde:

No datu bāzes iegūst padotās galerijas pievienoto bilžu skaitu.

Izvade:

Atgriež bilžu skaitu.

4.1.3.9. Metode „getPicture”

Apraksts:

Iegūst kādas galerijas bildi.

Ievade:

- *Integer gid* – galerijas identifikators
- *Integer nr* – bildes skaits pēc kārtas, kārtojot bildes pēc galerijas identifikatoriem augošā secībā

Apstrāde:

Iegūst no datu bāzēs bildes identifikatoru, nosaukumu un aprakstu.

Izvade:

Atgriež masīvu, kurš satur pieprasītās vērtības.

4.1.3.10. Metode „getGalleryData”

Apraksts:

Iegūst kādas galerijas datus.

Ievade:

- *Integer gid* – galerijas identifikators

Apstrāde:

No datu bāzes iegūst galerijas nosaukumu, aprakstu un izveides datumu.

Izvade:

Atgriež masīvu, kurš satur pieprasītās vērtības.

4.1.3.11. Metode „drawGalleryView”

Apraksts:

Uzzīmē lapā pieprasīto galeriju.

Ievade:

- *Integer gid* – galerijas identifikators

Apstrāde:

- Iegūst galerijas datus
- Izsauc javascript funkciju *showImage*
- Izsauc funkcijas, kas nodrošina komentāru atbalstu pie bildēm

4.1.3.12. Metode „getGalleryPictureList”

Apraksts:

Iegūst pieprasītās galerijas bilžu sarakstu.

Ievade:

- *Integer gid* – galerijas identifikators

Apstrāde:

No datu bāzes tiek iegūti visi bilžu nosaukumi, kuri pieder pieprasītajai galerijai

Izvade:

Atgriež masīvu ar bilžu nosaukumiem.

4.1.3.13. Metode „drawEditGallery”

Apraksts:

Uzzīmē galerijas attēlu augšupielādes formu kopā ar augšupielādētajām bildēm noteiktai galerijai.

Ievade:

- *Integer id* – galerijas identifikators

Apstrāde:

Pievieno lapā galerijas bilžu augšupielādes formu.

4.1.3.14. Metode „isAuthor”

Apraksts:

Noskaidro vai lietotājs ir galerijas autors.

Ievade:

- *Integer uid* – lietotāja identifikators
- *Integer gid* – galerijas identifikators

Apstrāde:

- Iegūst lietotāja identifikatoru no datu bāzes padotajai galerijai
- Salīdzina padoto lietotāja identifikatoru ar datu bāzes lietotāja identifikatoru

Izvade:

Ja lietotāju identifikatori sakrīt atgriež *true*, pretējā gadījumā atgriež *false*

4.1.3.15. Javascript metode „addGallery”

Apraksts:

Sūta POST datus ar AJAX palīdzību, lai pievienotu jaunu galeriju.

Apstrāde:

- Iegūst sūtāmos datus
- Nokonfigurē sūtīšanas parametrus
- Validē sūtāmo informāciju
- Sūta pieprasījumu

Izvade:

Atgriež informāciju no servera.

Kļūdas:

Kļūdas gadījumā izveido *alert* logu, kurā ir paziņots, ka ir radusies kļūda.

4.1.3.16. Javascript metode „validGalleryData”

Apraksts:

Validē galerijas informācijas datus.

Apstrāde:

- Iegūst validējamus datus
- Validē datus

Izvade:

Ja visi dati validējas veiksmīgi atgriež *true*, pretējā gadījumā atgriež *false*

Kļūdas:

Atkarībā no kļūdas, atbilstošajā kļūdu laukā ieraksta kļūdas informāciju.

4.1.3.17. Javascript metode „showImage”

Apraksts:

Sūta POST datus ar AJAX palīdzību, lai iegūtu galerijas bildi.

Ievade:

- *Integer nr* – kura bilde pēc kārtas jāņem
- *Integer gid* – galerijas identifikators

Apstrāde:

- Sūta POST pieprasījumu
- Saņemto XML failu atbilstoši parsē, lai iegūtu bildes informāciju
- Attēlo lapā jauno bildi un tās komentārus.

Kļūda:

Ja sūtīšanas laikā radās kļūda paziņo par to.

4.1.4. Komentāru modulis

4.1.4.1. Metode „addComment”

Apraksts:

Ieraksta datu bāzē komentāra datus.

Ievade:

- *String from* – vieta no kurienes nāk komentārs (galerija, dienasgrāmata utt.)
- *Integer fromID* – objekts, kuram tiks pievienots komentārs
- *String text* – komentāra teksts

Apstrāde:

Ieraksta komentāra datus datu bāzē atbilstošajā tabulā, vadoties pēc vietas no kurienes nāk komentārs.

Izvade:

Atgriež pievienotā komentāra identifikatoru.

4.1.4.2. Metode „getComments”

Apraksts:

Iegūst visus komentārus konkrētajam objektam.

Ievade:

- *String from* – vieta no kurienes nāk komentārs
- *Integer fromID* – objekta identifikators, kuram nepieciešams iegūt komentārus
- *Integer commentID* – komentāra identifikators (noklusētā vērtība 0)

Apstrāde:

legūst visus komentārus dotajam objektam, ja komentāra identifikators ir 0, pretējā gadījumā iegūst komentāru pēc dotā komentāra identifikatora.

Izvade:

Atgriež *result* tipa mainīgo, kurš satur nepieciešamos komentāru datus.

4.1.4.3. Metode „drawComments”

Apraksts:

Uzzīmē komentārus nepieciešamajā vietā.

Ievade:

- *String from* – Vieta, kurā zīmēt komentārus
- *Integer fromID* – objekta identifikators, kuram pieder komentāri
- *Integer commentID* – komentāra identifikators (noklusētā vērtība 0)

Apstrāde:

- Iegūst komentāru datus
 - Pievieno lapā komentāru dizaina failu
 - Katram komentāra tekstam un tā pievienošanas laikam izsauc funkcijas, kuras to apstrādā
-

4.1.4.4. Metode „formatSQL”

Apraksts:

Apstrādā SQL vaicājumu, lai iegūtu komentārus no pareizās tabulas.

Ievade:

- *String from* – tabulas prefikss, kurā atrodas komentāri
- *String sql* – SQL vaicājums

Apstrāde:

Visus vaicājuma apgabalus, kur ir vērtība *<from_name>* aizstāj ar tabulas prefiksu.

Izvade:

Atgriež apstrādātu SQL vaicājumu.

4.1.4.5. Metode „drawAddCommentForm”

Apraksts:

Uzzīmē komentāra pievienošanas formu.

Ievade:

- *String from* – Vieta, kur tiks pievienoti komentāri
- *Integer fromID* – objekta identifikators, kuram tiks pievienoti komentāri

Apstrāde:

Pievieno lapā komentāru pievienošanas formu

4.1.4.6. Metode „formatCommentDate”

Apraksts:

Apstrādā komentāra pievienošanas laiku.

Ievade:

- *Integer time* – laiks *posix time* formātā

Apstrāde:

Pārveido doto laiku tā lai tas rādītu cik sekundes/minūtes/stundas/dienas ir pagājušas kopš komentāra pievienošanas.

Izvade:

Atgriež jau noformatētu laiku.

4.1.4.7. Metode „formatCommentText”

Apraksts:

Noformatē komentāra tekstu, lai tas būtu gatavs attēlošanai.

Ievade:

- *String text* – komentāra teksts

Apstrāde:

Formatē komentāra tekstu (noņem liekās slīpsvītras un *new line* vietā ieliek `
` tagu.

Izvade:

Izvada apstrādātu komentāru tekstu.

4.1.4.8. Javascript metode „drawComments”

Apraksts:

Ar AJAX palīdzību nosūta POST parametrus, lai iegūtu komentārus atvērtajam objektam.

Ievade:

- *String from* – vieta no kurienes tiek pieprasīti komentāri
- *Integer fromID* – objekta identifikators, kuram tiek pieprasīti komentāri

Apstrāde:

- Tiek nosūtīts POST pieprasījums
- Saņemtie dati tiek parādīti komentāru vietā
- Tiek izsaukta funkcija, kura uzzīmē komentāru pievienošanas formu.

Kļūdas:

Paziņo, ka ir radusies kļūda, ja pieprasījums neizdodas.

4.1.4.9. Javascript metode „drawAddCommentForm”

Apraksts:

Pievieno lapā komentāra pievienošanas formu.

Ievade:

- *String from* – vieta no kurienes tiek pieprasīti komentāri
- *Integer fromID* – objekta identifikators, kuram tiek pieprasīti komentāri

Apstrāde:

- Tiek nosūtīts POST pieprasījums
- Saņemtie dati ir pievienošanas forma un tā tiek attēlota lapā.

Kļūdas:

Paziņo, ka ir radusies kļūda, ka pieprasījums neizdodas.

4.1.4.10. Javascript metode „addComment”

Apraksts:

Metode, kura ar AJAX palīdzību nosūta POST parametrus, kuros atrodas pievienojamā komentāra dati.

Ievade:

- *String from* – vieta no kurienes tiek pievienots komentārs
- *Integer fromID* – objekts, kuram tiek pievienots komentārs

Apstrāde:

- Iegūst nepieciešamās vērtības
- Uzģenerē parametrus
- Nosūta POST pieprasījumu
- Atbildē saņemtais teksts ir pievienotais komentārs noformatētā formā
- Pievienotais komentārs tiek attēlots lapā

Kļūdas:

Paziņo, ka ir radusies kļūda, ja pieprasījums neizdodas.

4.1.5. Grupu modulis

4.1.5.1. Metode „addGroup”

Apraksts:

Ieraksta datu bāzē jaunu grupas ierakstu.

Ievade:

- *String name* – grupas nosaukums
- *String description* – grupas apraksts

Apstrāde:

Ieraksta datubāzē grupas informāciju.

Izvade:

Atgriež grupas identifikatoru.

4.1.5.2. Metode „hasGroup”

Apraksts:

Pārbauda vai lietotājs ir kādas grupas izveidotājs.

Ievade:

- *Integer id* – lietotāja identifikators

Apstrāde:

Pārbauda vai grupu tabulā ir tāda grupa, kurai izveidotājs ir lietotājs ar padoto identifikatoru.

Izvade:

Izvada *true*, ja lietotājam ir grupa, pretējā gadījumā izvada *false*.

4.1.5.3. Metode „drawAddGroup”

Apraksts:

Uzzīmē lapā jaunas grupas izveides formu.

Apstrāde:

Pievieno lapai grupas izveides formu.

4.1.5.4. Metode „drawMyGroup”

Apraksts:

Uzzīmē lapā manas grupas informāciju un lietotājus.

Apstrāde:

Pievieno lapai manas grupas dizaina lapu.

4.1.5.5. Metode „getPosibleMembersList”

Apraksts:

Iegūst to lietotāju datus, kuri var tikt pievienoti konkrēta lietotāja grupā.

Ievade:

- *Integer id* – lietotāja identifikators

Apstrāde:

Iegūst to lietotāju datus, kuri atrodas konkrētā lietotāja ciltskokā.

Izvade:

Atgriež *result* tipa mainīgo, kurš satur pieprasītos datus.

4.1.5.6. Metode „drawPosibleMembersList”

Apraksts:

Uzzīmē iespējamo grupas lietotāju sarakstu.

Ievade:

- *Integer id* – lietotāja identifikators

Apstrāde:

- Iegūst lietotāju sarakstu
 - Pievieno lapā saraksta dizaina failu.
-

4.1.6. Ciltskoka modulis

4.1.6.1. Metode „__construct”

Apraksts:

Izveido ciltskoka moduli.

Ievade:

- *User user* – klases *User* objekts.

Apstrāde:

- Uzstāda privātajam mainīgajam *user* vērtību, kas padota.
- Izsauc vecāku klases konstruktoru

4.1.6.2. Metode „generateGrid”

Apraksts:

Uzģenerē koordinātu laukumu, kur zīmēsies ciltskoku locekļi.

Ievade:

- *Integer width* – koordinātu sistēmas x ass garums vienā virzienā
- *Integer height* – koordinātu sistēmas y ass garums vienā virzienā
- *Integer fromID* – lietotāja identifikators, kurš šobrīd atrodas centrā (noklusētā vērtība 0)

Apstrāde:

- Uzģenerē lapā koordinātu sistēmu izmantojot html tabulas
- Katram laukumam pārbauda vai viņš nav aizpildīts, ja ir aizpilda to
- Pašā centrā ievieto centra locekli (ja tā identifikators ir 0, tad tas ir iežurnalējošais lietotājs)

4.1.6.3. Metode „drawProfileField”

Apraksts:

Uzzīmē ciltskoka locekļa lauku.

Ievade:

- *Integer id* – ciltskoka locekļa identifikators (noklusētā vērtība 0)
- *Bool root* – vai ciltskoka loceklis šobrīd ir centrā (noklusētā vērtība *false*)
- *Integer xTmp* – x ass vērtība, ja tā neatbilst datu bāzē rakstītajai (noklusētā vērtība *false*)
- *Integer yTmp* – y ass vērtība, ja tā neatbilst datu bāzē rakstītajai (noklusētā vērtība *false*)

Apstrāde:

- Ja ciltskoka locekļa identifikators ir 0, tad zīmē lietotāja, kurš ir iežurnalējis, laukumu
- Pretējā gadījumā iegūst ciltskoka locekļa informāciju
- Ja asu koordinātu vērtības ir *false* iegūst tās no datu bāzes
- Iegūst ciltskoka locekli, kurš pievienoja konkrēto ciltskoka locekli
- Ja ciltskoka loceklis ir reģistrējies sistēmā, tad pārējos datus iegūst no lietotāja, pretējā gadījumā no ciltskoka locekļa
- Lapā uzģenerē ciltskoka locekļa lauku

4.1.6.4. Metode „addBlankField”

Apraksts:

Uzģenerē ciltskoka locekļa pievienošanas formu.

Ievade:

- *Integer fromX* – izsaukuma locekļa x ass vērtība
- *Integer fromY* – izsaukuma locekļa y ass vērtība
- *Integer fromID* – izsaukuma locekļa identifikators

- *Integer type* – saistības tips

Apstrāde:

- Iegūst formas zīmēšanas lauka koordinātas
- Uzģenerē pievienošanas formu

4.1.6.5. Metode „renderCords”

Apraksts:

Izskaitļo jauna ciltskoka lauka zīmēšanas koordinātas, balstoties uz izsaukuma locekļa koordinātām un jaunā ciltskoka locekļa saistības ar veco locekli tipu.

Ievade:

- *Integer fromX* – izsaukuma locekļa x ass vērtība
- *Integer fromY* – izsaukuma locekļa y ass vērtība
- *Integer fromID* – izsaukuma locekļa identifikators
- *Integer type* – saistības tips

Apstrāde:

- Ja tips ir tēvs, tad $x = \text{fromX} - 1$ – bērnu skaits un $y = \text{fromY} + 1$
- Ja tips ir māte, tad $x = \text{fromX} + 1$ un $y = \text{fromY} + 1$
- Ja tips ir māsa/brālis, tad $x = \text{fromX} - 1$ – brāļu/māsu skaits un $y = \text{fromY}$
- Ja tips ir vīrs/sieva, tad $x = \text{fromX} + 1$ + bērnu skaits
- Ja tips ir dēls/meita, tad $x = \text{fromX} + 1$ + brāļu/māsu skaits un $y = \text{fromY} - 1$

Izvade:

Atgriež masīvu ar x un y vērtībām

4.1.6.6. Metode „makeUncordRelationsXML”

Apraksts:

Pārveido zināmās saistības par cita veida saistībām balstoties uz jaunā ciltskoka centra attiecībām.

Ievade:

- *Integer id* – jaunā ciltskoka locekļa centra identifikators

Apstrāde:

- Iegūst ciltskoka locekļa identifikatoru un tipu, kurš izveidoja jauno ciltskoka centru
- Atkarībā no saistības tipa uzģenerē XML failu, kas pasaka koordinātes kur būs locekļi un to identifikatorus

Izvade:

Uzģenerētais XML fails

4.2. Datu detalizētais projektējums

4.2.1. Users

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Lietotāja identifikators
Email	Varchar(200)		Jā	Lietotāja e-pasts
Pass	Varchar(40)		Jā	Lietotāja parole (šifrēta pēc SHA1 algoritma)
Name	Varchar(150)		Jā	Lietotāja vārds
Surname	Varchar(150)		Jā	Lietotāja uzvārds
dateOfBirth	Int(11)		Nē	Lietotāja dzimšanas datums posix time formātā
timeJoined	Int(11)		Jā	Lietotāja reģistrācijas laiks

				posix time formātā
About	Varchar(250)		Nē	Neliels apraksts par lietotāju
Biography	Text		Nē	Lietotāja biogrāfija
Picture	Varchar(50)		Nē	Lietotāja profila bildes nosaukums

4.2.2. Relations

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Saistības identifikators
xCord	Int(11)		Jā	Ciltskoka lauka x koordināta
yCord	Int(11)		Jā	Ciltskoka lauka y koordināta
Type	Tinyint(4)		Jā	Saistības tips: <ul style="list-style-type: none"> • 1 – tēvs • 2 – māte • 3 – brālis • 4 – māsa • 5 – vīrs • 6 – sieva • 7 – dēls • 8 – meita
belongsTo	Int(11)	Foreign Key	Jā	Lietotāja identifikators, kura ciltskokā ir šī saistība
fromID	Int(11)	Foreign Key	Jā	Ciltskoka locekļa identifikators, no kura iziet relācija.
toID	Int(11)	Foreign Key	Jā	Ciltskoka locekļa identifikators, uz kuru iziet relācija.

4.2.3. Relation Members

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Ciltskoka locekļa identifikators
realID	Int(11)	Foreign Key	Nē	Lietotāja locekļa identifikators, kurš ir arī ciltskoka loceklis
Name	Varchar(150)		Nē	Ciltskoka locekļa vārds
Surname	Varchar(150)		Nē	Ciltskoka locekļa uzvārds

4.2.4. Postit

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Ātrās ziņas identifikators
userID	Int(11)	Foreign Key	Jā	Lietotājs, kurš pievienojis ātro ziņu
Text	Varchar(200)		Jā	Ātrās ziņas teksts
Color	Varchar(6)		Nē	Ātrās ziņas fona krāsa hexadecimal color formātā

timeAdded	Int(11)		Jā	Laiks, kad pievienota ātrā ziņa posix time formātā
-----------	---------	--	----	--

4.2.5. Groups

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Grupās identifikators
creatorID	Int(11)	Foreign Key	Jā	Lietotāja identifikators, kurš izveidojis grupu
Name	Varchar(150)		Jā	Grupās nosaukums
Description	Text		Nē	Grupās apraksts
timeCreated	Int(11)		Jā	Laiks, kad grupa izveidota posix time formātā

4.2.6. Group Members

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Grupās dalībnieka identifikators
groupID	Int(11)	Foreign Key	Jā	Grupās identifikators, kurā pieder dalībnieks
userID	Int(11)	Foreign Key	Jā	Lietotāja identifikators, kurš iekļaujas grupā
Confirmed	Tinyint(1)		Jā	Lietotāja apstiprinājums par iekļaušanos grupā. 0 – nepiekrīt 1 – piekrīt Noklusētā vērtība – 0
timeAdded	Int(11)		Jā	Laiks, kad lietotājs ir uzaicināts būt grupā

4.2.7. Gallery

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Galerijas identifikators
userID	Int(11)	Foreign Key	Jā	Lietotāja identifikators, kuram pieder galerija
Title	Varchar(200)		Jā	Galerijas nosaukums
Description	Text		Nē	Galerijas apraksts
timeAdded	Int(11)		Jā	Laiks, kad izveidota galerija posix time formātā

4.2.8. Gallery Pictures

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Galerijas bildes identifikators
galleryID	Int(11)	Foreign Key	Jā	Galerijas identifikators, kurai pieder bilde
fileName	Varchar(50)		Jā	Bildes nosaukums
Description	Varchar(250)		Nē	Bildes apraksts

timeAdded	Int(11)		Jā	Laiks, kad augšupielādēta bilde posix time formātā
-----------	---------	--	----	--

4.2.9. Blog

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Dienasgrāmatas identifikators
userID	Int(11)	Foreign Key	Jā	Lietotāja identifikators, kuram pieder dienasgrāmatas ieraksts
Title	Varchar(150)		Jā	Dienasgrāmatas ieraksta nosaukums
Text	Text		Jā	Dienasgrāmatas ieraksta teksts

4.2.10. Comments

Datu bāzē var būt vairākas komentāru tabulas un atkarībā no komentāru vietas (galerija, dienasgrāmatas, lietotāju profils) mainās tabulas nosaukuma prefikss pēc šādas sistēmas – *komentāru_vietaComments*, piemēram, *galleryComments*.

Tabulas struktūra saglabājas nemainīga atkarībā no vietas, kur tiek pievienoti komentāri, tādēļ detalizēti aprakstīts tiek tikai komentāru tabulas vispārīgais gadījums. Visu komentāru tabulu struktūras ir vienādas.

Kolonnas nosaukums	Datu tips	Atslēgas	Obligāts	Apraksts
Id	Int(11)	Primary Key	Jā	Komentāra identifikators
fromID	Int(11)	Foreign Key	Jā	Komentāra objekta identifikators
userID	Int(11)	Foreign key	Jā	Lietotāja identifikators, kurš pievienoja komentāru
Text	Text		Jā	Komentāra teksts
timeAdded	Int(11)		Jā	Laiks, kad pievienots komentārs posix time formātā.

Testēšanas dokumentācija

1. Ievads

Testēšanas dokumentācija satur informāciju par to kādi testi tika veikti, lai pārbaudītu sistēmas www.mesesam.lv darbību un kādi bija to rezultāti. Testēšanu veica kvalifikācijas darba autors.

Testējot netika pārbaudīta katra metode atsevišķi, bet lapas darbība kopumā pārbaudot darbības, ko varētu veikt standarta lietotājs.

2. Testēšanas rezultāti

Testēšana notika pa sadaļām, kuras apmeklētu standarta lietotājs.

2.1. WEB lapas pirmā lapa neautorizētam lietotājam

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
Jauna lietotāja reģistrācija ar pareiziem ievaddatiem.	Tiek pierēģistrēts jauns lietotājs un parādās par to informācija un jaunā lietotāja parole.	Jā		
Jauna lietotāja reģistrācija, ja nav aizpildīts kāds no laukiem.	Parādās paziņojums, ka nav aizpildīti visi lauki un lietotājs netiek pierēģistrēts.	Jā		
Jauna lietotāja reģistrācija, ja tiek ievadīts datu bāzē eksistējošs e-pasts.	Parādās paziņojums, ka lietotājs ar tādu e-pastu jau ir reģistrējies.	Jā		
Lietotāja autorizēšanās ar pareiziem ievaddatiem.	Lietotājs tiek ievests autorizēta lietotāja galvenajā lapā.	Jā		
Lietotāja autorizēšanās ar neeksistējošu e-pastu un nepareizu paroli	Parādās paziņojums – Kļūda e-pastā vai parolē. Lūdzu, mēģiniet vēlreiz!	Jā		

2.2. WEB lapas sākulapa autorizētam lietotājam

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
Sākulapas	Kreisā paneļa	Jā		

izkārtojums.	augšā atrodas informācija par lietotāju, apakšā navigācija pa sadaļām. Labā paneļa augšā atrodas meklēšana un zem tā grupu uzaicinājumi. Vidējā panelī atrodas ātrās ziņas.			
Navigācija pa sadaļām	Uzejot uz kādas no sadaļām iekrāsojas tās lauks. Uzspiežot uz tās atveras attiecīgā sadaļa.	Jā		
Informācija par lietotāju bez profila bildes.	Pašā augšā atrodas bilde, kura rādās visiem lietotājiem, kuriem nav profila bildes (cilvēka siluets bez galvas). Zem tās ir lietotāja vārds un uzvārds, tad „par mani” un tad „mana biogrāfija”	Jā		
Informācija par lietotāju ar profila bildi.	Pašā augšā atrodas lietotāja profila bildes lielais sīkattēls. Tālāk viss ir tāds pats kā lietotājam bez profila bildes.	Jā		
Lietotāju meklēšana pēc vārda un uzvārda.	Parādās lapa ar meklēšanas rezultātiem, kuri atbilst ievadītajiem kritērijiem (tika ievadīts Rolands Atvars, jāatrod lietotājs Rolands Atvars)	Jā		
Lietotāja meklēšana pēc vārda.	Tiek meklēts lietotājs pēc vārda <i>Rolands</i> . Jāparādās	Jā		

	rezultātu lapai ar atrastu lietotāju <i>Rolands Atvars</i>			
Lietotāja meklēšana pēc uzvārda.	Tiek meklēts lietotājs pēc uzvārda <i>Atvars</i> . Jāparādās rezultātu lapai ar atrastu lietotāju <i>Rolands Atvars</i> .	Jā		
Lietotāja meklēšana neievadot ievaddatus.	Jāparādās tukšai rezultātu lapai.	Jā		
Grupu uzaicinājumi bez uzaicinājumiem.	Jāparādās uzrakstam, ka nav neviena uzaicinājuma.	Jā		
Grupu uzaicinājumi ar uzaicinājumiem.	Parādās lietotāja vārds un uzvārds kā saite uz lietotāja profilu un ģimenes nosaukums, uz kuru aicina. Blakus atrodas „+” un „-” pogas.	Jā		
Grupu uzaicinājuma apstiprināšana.	Pazūd konkrētais uzaicinājums no uzaicinājumu saraksta un lietotāja lauks iekrāsojas zaļā krāsā (to redz lietotājs, kurš uzaicināja).	Jā		
Grupu uzaicinājuma apstiprināšana.	Pazūd konkrētais uzaicinājums no uzaicinājumu saraksta un lietotāja lauks paliek dzeltenā krāsā (to redz lietotājs, kurš uzaicināja)	Jā		

2.3. Lietotāja profila informācijas maiņas lapa

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
Lietotāja profila informācijas maiņa visos laukos ar	Tiek saglabātas izmaiņas un parādās	Jā		

pareiziem ievaddatiem.	paziņojums par veiksmīgu datu maiņu.			
Lietotāja profila informācijas maiņa vienā laukā ar pareiziem ievaddatiem.	Tiek saglabātas izmaiņas un parādās paziņojums par veiksmīgu datu maiņu.	Jā		
Lietotāja profila informācijas maiņa, ja izdzēš vārdu vai/un uzvārdu.	Neļauj saglabāt profila datu maiņu izvadot par to kļūdas paziņojumu.	Nē	Svarīgs	Ļauj saglabāt profila datu maiņu izdzēšot lietotāja vārdu vai uzvārdu.
Lietotāja profila informācijas maiņa, ja tiek ievadīts dzimšanas datums nepareizā formātā (23/12/19g0)	Neļauj saglabāt profila datu maiņu un izvada paziņojumu, ka ievadītais dzimšanas datums nav pareizā formā.	Jā		
Lietotāja paroles maiņa ievadot pareizu veco paroli un identiskas jaunās paroles.	Nomaina lietotāja paroli un izvada par to paziņojumu.	Jā		
Lietotāja paroles maiņa ievadot nepareizu veco paroli.	Izvada kļūdas paziņojumu, ka nepareiza vecā parole.	Nē	Mazsvarīgs	Izvada nevis kļūdas paziņojumu, bet vienkāršu paziņojumu.
Lietotāja paroles maiņa ievadot nevienādas jaunās paroles.	Izvada kļūdas paziņojumu, ka jaunās paroles nesakrīt.	Jā		
Lietotāja paroles maiņa neievadot jaunās paroles.	Neļauj nomainīt paroli.	Nē	Svarīgs	Saglabā lietotāja paroli kā tukšumu.
Lietotāja profila bildes maiņa ar pareizu attēlu.	Parādās animācija, kura attēlo, ka attēls augšupielādējas. Pēc tam parāda augšupielādēto bildi un paziņo, ka viss izdevās.	Jā		
Lietotāja profila bildes maiņa ar augšupielādētu .pdf failu.	Neļauj nomainīt profila bildi izvadot par to paziņojumu.	Nē	Vidēji svarīgs	Parāda paziņojumu, ka profila bilde nomainīta.

2.4. Ciltskoka lapa

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
Jauna ciltskoka locekļa pievienošana, kurš nav reģistrējies sistēmā.	Atkarībā no locekļa tipa attiecīgā laukā parāda locekļa pievienošanas formu un ievadot locekļa datus un spiežot „Pievienot” tajā pat laukā parāda ciltskoka locekļa datus.	Jā		
Jauna ciltskoka locekļa pievienošana, kurš ir reģistrējies sistēmā.	Atkarībā no locekļa tipa attiecīgā laukā parāda locekļa pievienošanas formu. Ievadot locekļa meklējamo informāciju un spiežot meklēt atrod visus lietotājus ar ievadītajiem datiem. Nospiežot „+” tajā pat laukā parāda lietotāja informāciju un pievieno viņu ciltskokam.	Jā		
Ciltskoka locekļu meklēšanas rezultātu funkcionalitāte.	Uzspiežot uz lietotāja profila un vārda jāatveras lietotāja profils.	Nē		Nekas nenotiek.
Jauna ciltskoka locekļa ar tipu brālis/māsa pievienošana, ja nav pievienots neviens cits brālis/māsa un tētis.	Blakus (pa kreisi) ciltskoka loceklim, kurš pievieno radnieku parādās pievienošanas forma	Jā		
Jauna ciltskoka locekļa ar tipu brālis/māsa pievienošana, ja ir	Blakus (pa kreisi) iepriekšēji pievienotam brālim/māsai	Jā		

pievienots jau cits brālis/māsa, bet nav pievienots tēvs.	parādās pievienošanas forma			
Jauna ciltskoka locekļa ar tipu/brālis māsa pievienošana, ja ir pievienots tēvs.	Blakus (pa kreisi) iepriekšēji pievienotam brālim/māsai (ja tādi ir) vai blakus loceklim, kuram tiek pievienots radnieks (ja nav brāļi/māsas) parādās pievienošanas forma. To apstiprinot parādās locekļa lauks un tēva lauks pabīdās vienu lauku pa kreisi.	Jā		
Jauna ciltskoka locekļa ar tipu tēvs pievienošana ja loceklim, kuram tiek pievienots tēvs ir māsas/brāļi.	Aiz pēdējā brāļa/māsas un vienu rindiņu uz augšu parādās pievienošanas forma.	Jā		
Jauna ciltskoka locekļa ar tipu tēvs pievienošana, ja loceklim, kuram tiek pievienots tēvs nav brāļi/māsas	Vienu rindiņu augstāk un vienu kolonnu pa kreisi parādās pievienošanas forma.	Jā		
Jauna ciltskoka locekļa ar tipu māte, pievienošana.	Vienu rindiņu augstāk un vienu kolonnu pa labi parādās pievienošanas forma.	Jā		
Jauna ciltskoka locekļa ar tipu sieva/vīrs pievienošana, ja loceklim, kurš pievieno šo tipu nav ciltskoka locekļi ar tipu dēls/meita	Blakus (pa labi) parādās pievienošanas forma.	Jā		
Jauna ciltskoka locekļa ar tipu sieva/vīrs pievienošana, ja loceklim, kurš pievieno šo tipu ir	Pēc pēdējā locekļa ar tipu dēls/meita un vienu rindiņu augstāk parādās pievienošanas	Jā		

ciltškoka locekļi ar tipu dēls/meita.	forma.			
Jauna ciltškoka locekļa ar tipu dēls/meita pievienošana, ja nav pievienots loceklis ar tipu vīrs/sieva.	Zem ciltškoka locekļa, kas pievieno jauno ciltškoka locekli un vienu kolonnu pa labi (vai pēc pēdējā pievienotā locekļa ar tipu dēls/meita) parādās pievienošanas forma.	Jā		
Jauna ciltškoka locekļa ar tipu dēls/meita pievienošana, ja ir pievienots loceklis ar tipu vīrs/sieva.	Zem ciltškoka locekļa un pēc pēdējā locekļa ar tipu dēls/meita parādās pievienošanas forma un to apstiprinot parādās locekļa lauks un vīra/sievas lauks pabīdās vienu lauku pa labi.	Jā		
Ciltškoka locekļa, kurš atrodas centrā nomainīa uz citu ciltškoka locekli.	Ciltškoks atjaunojas, centrā parādās izvēlētais ciltškoka loceklis. Viņam apkārt parādās visi tiešie radi.	Jā		

2.5. Pievienot galeriju lapa

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
Galerijas pievienošana ar pareiziem ievaddatiem.	Parādās bilžu augšupielādes forma.	Jā		
Galerijas pievienošana ar tukšu apraksta lauku.	Parādās bilžu augšupielādes forma	Jā		
Galerijas pievienošana ar tukšu nosaukuma lauku.	Parādās kļūdas paziņojums ar uzrakstu Nav aizpildīta nosaukuma aile!	Jā		

Galerijas pievienošana ar nosaukumu, kurš ir garāks par 200 simboliem.	Parādās kļūdas paziņojums ar uzrakstu Nosaukums nedrīkst būt garāks par 200 simboliem!	Jā		
Bildes augšupielāde ar pareiziem ievaddatiem.	Zem formas parādās augšupielādētās bildes lielais sīkattēls un parādās attiecīgs uzraksts.	Jā		
Bildes augšupielāde ar neaizpildītu apraksta lauku.	Zem formas parādās augšupielādētās bildes lielais sīkattēls un parādās attiecīgs uzraksts.	Jā		

2.6. Dienasgrāmatas pievienošanas lapa

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
Dienasgrāmatas pievienošana ievadot visus nepieciešamos datus (korektus)	Tiek pievienots dienasgrāmatas ieraksts un tiek atvērts jaunpievienotais ieraksts.	Jā		
Dienasgrāmatas ieraksta pievienošana neaizpildot virsraksta lauku.	Izvada kļūdas paziņojumu, kur rakstīts – Nav aizpildīti visi lauki!	Jā		
Dienasgrāmatas ieraksta pievienošana, ja nosaukums ir garāks par 150 simboliem.	Izvada kļūdas paziņojumu – leraksta virsraksts nevar būt garāks par 150 simboliem!	Jā		
Dienasgrāmatas ieraksta pievienošana korekti aizpildot nosaukuma lauku, bet neaizpildot teksta lauku.	Izvada kļūdas paziņojumu, kur rakstīts – Nav aizpildīti visi lauki!	Jā		

2.7. Galeriju apskates lapa

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
Galeriju atvēršana.	Uzspiežot uz kādas galerijas atveras izvēlēta galerija.	Jā		
Galerijas navigācijas pozīcija, ja atvērta pirmā bilde no vairākām.	Rādās saite (zem bildes) ar kuru var pāriet uz nākamo bildi. Tāpat rādās lauks uz bildes uz kura uzspiežot var pāriet uz nākamo bildi. Apakšā rādās, ka ir atvērta pirmā bilde no visa bilžu skaita.	Jā		
Galerijas navigācijas pozīcija, ja atvērta pēdējā bilde no vairākām.	Rādās saite (zem bildes) ar kuru var pāriet uz iepriekšējo bildi. Tāpat rādās lauks uz bildes uz kura uzspiežot var pāriet uz iepriekšējo bildi. Apakšā rādās, ka ir atvērta pēdējā bilde no visa bilžu skaita.	Jā		
Galerijas navigācijas pozīcija ja atvērta kāda no bildēm, kura nav ne pēdējā, ne pirmā.	Rādās saites (zem bildes) ar, kurām var pāriet uz nākamo un iepriekšējo bildi. Tāpat rādās divi lauki virs bildes ar kuriem var pāriet uz nākamo un iepriekšējo bildi. Un rādās arī kura bilde pēc kārtas ir atvērta no kopējo bilžu skaita.	Jā		
Galerijas navigācija, ja galerijā ir tikai viena bilde.	Nekādas galerijas navigācijas saites un pogas nerādās. Rāda tikai, ka atvērta viena bilde no vienas.	Jā		
Apraksta lauka	Apraksta lauks	Jā		

stāvoklis bildei, kurai nav apraksta.	nerādās.			
Galerijas izvēlnes stāvoklis, ja lietotājs skatās kādu no savām galerijām	Galerijas izvēlnē rādās saite ar uzrakstu „Labot galeriju”.	Jā		
Darbība nospiežot uz saites – <i>Labot galeriju</i>	Atveras bilžu augšupielādes forma ar augšupielādētajām bildēm.	Jā		
Komentāru pievienošana ar korektiem datiem.	Pievienotais komentārs parādās zem pēdējā pievienotā komentāra, vai arī virs komentāru pievienošanas lauka.	Jā		
Komentāru pievienošana ar tukšu komentāru lauku.	Jāizvada paziņojums par to, ka tukšs komentāra teksta lauks.	Nē	Vidēji svarīga	Atļauj pievienot komentāru bez teksta.

2.8. Ātrās ziņas

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
„Pievienot ziņu” saites nospiešana.	Atveras ātro ziņu pievienošanas forma	Jā		
Ātrās ziņas pievienošana bez krāsas lauka aizpildīšanas, bet ar korekti aizpildītu teksta lauku.	Pievienojas ātrā ziņa ar noklusēto krāsas lauku.	Jā		
Ātrās ziņas pievienošana ar korektiem datiem.	Pievienojas ātrā ziņa ar ievadīto krāsu fonā.	Jā		
Ātrās ziņas pievienošana ar korektu teksta lauku, bet nekorektu krāsas lauku.	Pievienojas ātrā ziņa un fonā rādās noklusētā krāsa	Nē	Mazsvarīga	Ātrā ziņa pievienojas, bet fonā rādās balta krāsa nevis svītraini balta.
Ātrās ziņas pievienošana ar tukšu teksta lauku.	Nepievienojas ātrā ziņa, izvadot kļūdas paziņojumu.	Nē	Mazsvarīga	Ātrā ziņa pievienojas ar tukšu teksta lauku.
Ātro ziņu dzēšana.	Nospiežot saiti „Dzēst” ātrā ziņa	Jā		

	izdzēšas.			
Ātro ziņu rādīšana, ja lietotājs nav nevienā grupā.	Rādās tikai lietotāja ātrās ziņas	Jā		
Ātro ziņu rādīšana, ja lietotājs ir kādā grupā.	Rādās visu to lietotāju ātrās ziņas, kuri atrodas vienā grupā ar konkrēto lietotāju.	Jā		

2.9. Manu grupu lapa

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
Manu grupu lapa, ja lietotājam nav izveidota grupa.	Rādās jaunas grupas pievienošanas forma.	Jā		
Jaunas grupas izveidošana, ja visi lauki tiek korekti aizpildīti.	Pievienojas grupa un parādās grupas lietotāju saraksts.	Jā		
Jaunas grupas izveidošana, ja ir korekti aizpildīts nosaukuma lauks, bet nav aizpildīts apraksta lauks.	Pievienojas grupa un parādās grupas lietotāju saraksts.	Jā		
Jaunas grupas izveidošana, ja nosaukuma lauks nav aizpildīts vai tas ir garāks par 150 simboliem.	Parādās kļūdas paziņojums – Nosaukumam jābūt aizpildītam un tas nedrīkst būt garāks par 150 simboliem	Jā		
Grupas lietotāju saraksta lietotāji.	Grupu lietotāju sarakstā rādās visi tie lietotāji, kuri ir iekļauti konkrētā lietotāja ciltskokā.	Jā		
Ielūguma, uz grupu, nosūtīšana.	Ielūgums tiek nosūtīts, saņēmēja lietotājam parādās jauns ielūgums, lietotāja krāsa lietotāju sarakstā iekrāsojas dzeltena.	Jā		
Ielūguma	Ja saņēmēja	Jā		

apstiprināšana	lietotājs apstiprina ielūgumu, viņa lauks lietotāju sarakstā iekrāsojas zaļš			
Lietotāja izdzēšana no grupas.	Lietotāja lauks iekrāsojas balts un viņš vairs neskaitās grupā.	Jā		

2.10. Grupu saraksta lapa

Testa apraksts	Sagaidāmais rezultāts	Tests iziets veiksmīgi	Svarīgums	Problēmas kopsavilkums
Saraksta attēlojums	Sarakstā rādās visas tās grupas, kurās ir lietotājs, izņemot viņa paša izveidoto grupu.	Jā		
Grupu saraksta izkārtojums	Katras grupas augšā rādās grupas izveidotājs, kura laukums ir iekrāsots viegli pelēkā krāsā.	Jā		
Saite uz lietotāja profilu.	Nospiežot uz katra lietotāja vārda vai uzvārda, atveras viņa profils.	Jā		
Palielināmā stikla darbība	Nospiežot uz palielināmā stikla atveras grupas apraksts.	Jā		
Palielināmā stikla atkārtota nospiešana.	Atkārtoti nospiežot uz palielināmā stikla grupas apraksts aizveras.	Nē	Mazsvarīgs	Grupas apraksts paliek atvērts
Izdzēšanās no grupas	Nospiežot sarkano krustiņu konkrētais lietotājs tiek izdzēsts no grupas.	Jā		

Lietotāja dokumentācija

1. Ievads

1.1. Auditorijas raksturojums

Lietotājus varētu raksturot, kā modernus cilvēkus, kuriem nav svešs interneta jēdziens un viņi prot to izmantot. Standarta lietotājs, kurš apmeklēs un izmantos sistēmu www.mesesam.lv varētu interesēties par dažādiem sociālajiem tīkliem un/vai par savas ģimenes saknēm un viņam ir svarīgi tas, lai viņa bērni zinātu savus vecākus un tālākus radus.

Šis dokumentācijas lasītājs ir cilvēks, kurš pilnībā vēlas izprast sistēmas www.mesesam.lv darbību vai cilvēks, kurš īpaši labi neorientējas interneta vidē vai arī ir saskāries ar kādām grūtībām, kuras vēlas novērst. Šo dokumentāciju var lasīt jebkurš lietotājs – gan tāds, kurš ir reģistrējies sistēmā, gan tāds, kurš vēl nav to izdarījis.

1.2. Lietojamība

Dokumentu vēlams izlasīt jebkuram, kurš vēlas pilnībā pārvaldīt sistēmu www.mesesam.lv. Ja lietotājs nav sevišķi zinošs par interneta tehnoloģijām un lapām, tad viņam vajadzētu izlasīt visu lietotāja dokumentāciju, pretējā gadījumā lietotājs var izlasīt tikai par to funkciju, kura viņam liekas interesanta vai nepieciešama.

1.3. Dokumenta nolūks

Dokuments paredzēts, lai palīdzētu lietotājiem izprast sistēmu www.mesesam.lv un ļautu tiem to pilnvērtīgi izmantot.

1.4. Dokumenta lietošanas apraksts

Dokumentu ieteicams izlasīti visu kopumā, ja ir vēlme izzināt sistēmu kopumā. Ja ir kāda neskaidrība, vajadzētu meklēt atbildi skatoties pēc nodaļu nosaukumiem.

1.5. Problēmu ziņošana

Ja pēc lietotāja dokumentācijas izlasīšanas joprojām ir kādi jautājumi par sistēmas www.mesesam.lv darbību vai arī ja Lietotāja Dokumentācijā pamanāt kādu kļūdu, tad droši rakstiet uz faq@mesesam.lv.

2. Instrukcija sistēmas lietotājiem

2.1. Reģistrēšanās sistēmā

Nēesi reģistrējies? **Izdari to tagad!**

Ilustrācija 1 - Saite uz reģistrēšanos

REĢISTRĒŠANĀS

VĀRDS:

UZVĀRDS:

E- PASTS:

Ilustrācija 2 – reģistrēšanās

Apsveicam **Lietotājs Dokumentētājs**! Jūs esat veiksmīgi piereģistrējies www.mesesam.lv.
 Jūsu īslaicīgā parole ir **ODNiNmUwNTAz**
 Nomainiet to līdzko jūs ieiesiet sistēmā.
 Lai ieietu sistēmā spiežat **šeit**

Ilustrācija 3 - veiksmīga reģistrācija

Lai nokļūtu uz reģistrācijas sekciju nepieciešams nospiegt uz saites „Izdari to tagad”, kura attēlota *Ilustrācija 1*. Tas jūs aizvedīs līdz reģistrācijas formai, kur jāievada savi dati, kuri ir prasīti (*Ilustrācija 2*). Jums būs nepieciešams savs e-pasts, kurš ir unikāls (nevienam citam lietotājam sistēmā tāds nevar būt).

Ja reģistrācija noritēs veiksmīgi parādīsies atbilstošs logs ar paziņojumu, ka esat veiksmīgi reģistrējies, kā arī jums tiks atklāta jūsu pagaidu parole (*Ilustrācija 3*).

2.1.1. Reģistrēšanās sistēmā kļūdu paziņojumi

Visiem laukiem jābūt obligāti aizpildītiem!

Ilustrācija 4 - reģistrēšanās kļūda

Cilvēks ar šādu e-pastu jau ir reģistrējies.

Ilustrācija 5 - reģistrēšanās kļūda

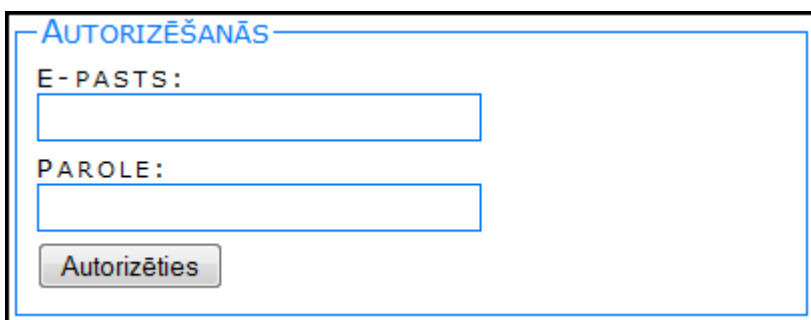
Ja reģistrējoties jums izmet kļūdu, tad vadoties pēc kļūdas paziņojuma būs nepieciešamība reģistrēties vēlreiz.

Ja reģistrācijas kļūda paziņo:

- Visiem laukiem jābūt obligāti aizpildītiem (*Ilustrācija 4*), tad jūs neesat aizpildījis kādu no laukiem (vārds, uzvārds, e-pasts). Jums nepieciešams aizpildīt visus laukus, lai reģistrētos

- Cilvēks ar šādu e-pastu jau ir reģistrējies (*Ilustrācija 5*), tad tas nozīmē, ka kāds jau ir reģistrējies sistēmā ar jūsu ierakstīto e-pastu. Jums būs nepieciešamība izvēlēties citu e-pastu, lai reģistrētos sistēmā.

2.2. Iežurnalēšanās sistēmā



AUTORIZĒŠANĀS

E- PASTS:

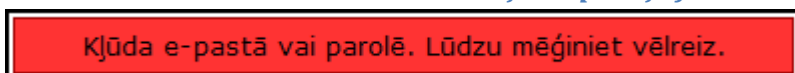
PAROLE:

Autorizēties

Ilustrācija 6 - iežurnalēšanās sistēmā

Lai iežurnalētos sistēmā, jums būs nepieciešams ievadīt savu e-pastu un paroli (*Ilustrācija 6*). Kad tas ir izdarīts, vajadzēs nospriest pogu – Autorizēties un ja e-pasts un parole būs pareizi, tad jūs tiksiet autorizēts ieiet sistēmā.

2.2.1. Iežurnalēšanās sistēmā kļūdu paziņojumi



Ilustrācija 7 - iežurnalēšanās kļūda

Ja jūs saskaraties ar kļūdas paziņojumu **Kļūda e-pastā vai parolē. Lūdzu, mēģiniet vēlreiz.**, tad pārliecinieties, vai jūsu ievadītais e-pasts un parole ir pareizi un mēģiniet tos ievadīt pa jaunu.

2.3. Navigācija pa sadaļām



Ilustrācija 8 - navigācija pa sadaļām

Navigēt pa sadaļām jūs varat izmantojot Navigācijas paneli, kurš atrodas kreisajā pusē (*Ilustrācija 8*). Lai pārietu uz citu kategoriju ir nepieciešams nospiegt atbilstošo saiti uz kuru jūs vēlaties pāriet un tā jūs aizvedīs tieši uz turieni.

2.4. Lietotāja profila informācijas maiņa

DATI PAR MANI

VĀRDS:

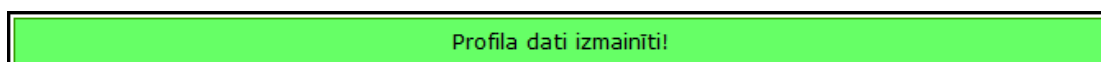
UZVĀRDS:

DZIMŠANAS DATUMS (FORMĀTĀ DD/MM/YYYY):

ĪSUMĀ PAR MANI:

MANA BIOGRĀFIJA:

Ilustrācija 9 - lietotāja datu maiņa

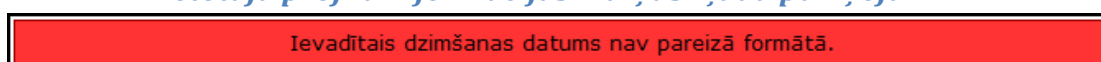


Ilustrācija 10 - veiksmīga profila informācijas maiņa

Lai mainītu savus profila datus nepieciešams aizpildīt formu „Dati par mani”. Kad visa nepieciešamā informācija ir uzrakstīta, tad nepieciešams nospiegt pogu „Labot datus”. Dzimšanas datumam jābūt formātā *dd/mm/yyyy*, kur *dd* ir diena, *mm* mēnesis un *yyyy* gads, kad jūs piedzimāt. Piemēram, ja esat dzimis 1970. Gada, 17. Maijā, tad jūsu dzimšanas datums nepieciešamajā formā būtu 17/05/1970.

Ja profila informācija tiks veiksmīgi nomainīta, pēc pogas nospiešanas, zem formas parādīsies atbilstošs ieraksts (*Ilustrācija 9*).

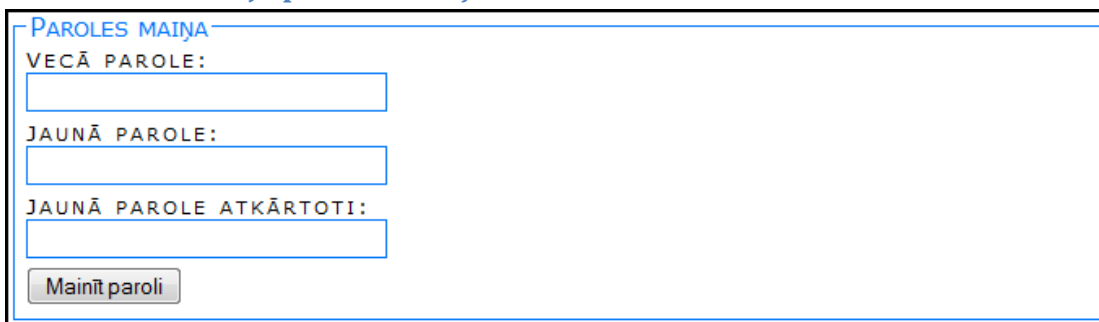
2.4.1. Lietotāja profila informācijas maiņas kļūdu paziņojumi



Ilustrācija 11 - profila informācijas maiņas kļūdu paziņojums

Ja mainot profila informāciju jums parādās sekojoša kļūda – **ievadītais dzimšanas datums nav pareizā formātā**, tad pārbaudiet vai jūsu ievadītais dzimšanas datums ir formā *dd/mm/yyyy* un mēģiniet izlabot profila datus vēlreiz.

2.5. Lietotāja paroles maiņa



PAROLES MAIŅA

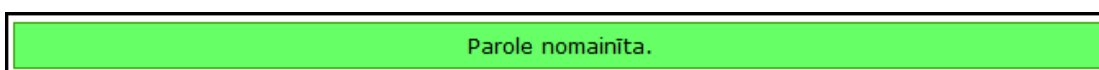
VECĀ PAROLE:

JAUNĀ PAROLE:

JAUNĀ PAROLE ATKĀRTOTI:

Mainīt paroli

Ilustrācija 12 - lietotāja paroles maiņas forma

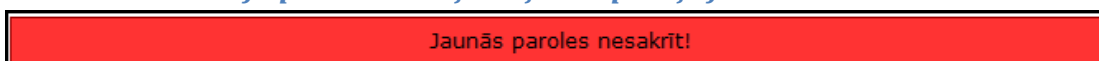


Parole nomainīta.

Ilustrācija 13 - veiksmīgas lietotāja paroles maiņas paziņojums

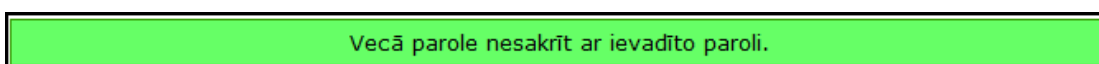
Lai nomainītu paroli, jums nepieciešams ierakstīt veco paroli un divas reizes ierakstīt jauno paroli paroles maiņas formā (*Ilustrācija 12*). Ja paroles maiņa noritēs veiksmīgi un jūsu ievadītie dati būs korekti, tad zem formas parādīsies atbilstošs paziņojums (*Ilustrācija 13*).

2.5.1. Lietotāja paroles maiņas kļūdas paziņojumi



Jaunās paroles nesakrīt!

Ilustrācija 14 - paroles maiņas kļūdu paziņojums



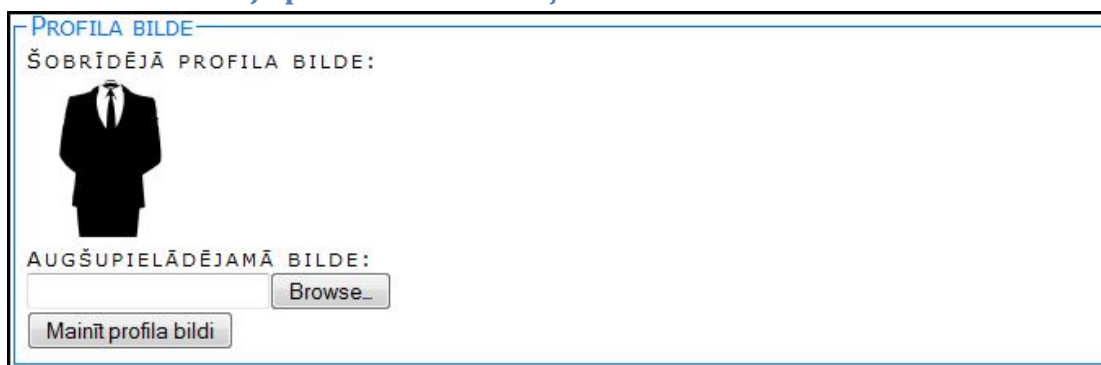
Vecā parole nesakrīt ar ievadīto paroli.

Ilustrācija 15 - paroles maiņas kļūdu paziņojums

Ja mainot paroli jums parādās kļūdu paziņojums tad jums vajag rīkoties atkarībā no kļūdu paziņojuma:

- **Jaunās paroles nesakrīt!** – ja parādās šāds kļūdas paziņojums, tad jūsu ievadītās jaunās paroles nav vienādas. Tām ir nepieciešams būt vienādām, lai būtu iespējams nomainīt paroli – pārliedzinieties, ka paroles ir vienādas.
- **Vecā parole nesakrīt ar ievadīto paroli** – tas nozīmē, ka jūsu ievadītā vecā parole nav tāda kāda tā ir saglabāta datu bāzē. Pārliedzinieties, ka esat pareizi ierakstījis veco paroli un mēģiniet vēlreiz.

2.6. Lietotāja profila bildes maiņa



The screenshot shows a web form titled "PROFILA BILDE". It contains two sections: "ŠOBRĪDĒJĀ PROFILA BILDE:" with a placeholder image of a suit, and "AUGŠUPIELĀDĒJAMĀ BILDE:" with an empty text input field, a "Browse..." button, and a "Mainīt profila bildi" button.

Ilustrācija 16 - lietotāja profila bildes maiņas forma

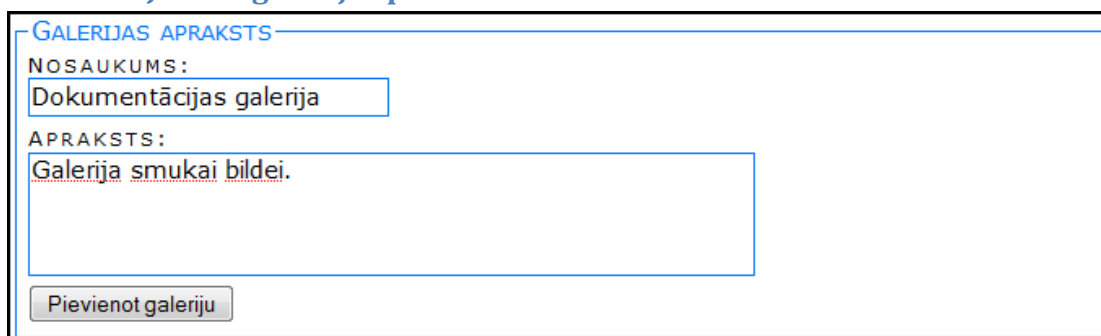


The screenshot shows the same form as in the previous image, but with a landscape photo in the "ŠOBRĪDĒJĀ PROFILA BILDE:" field. The "AUGŠUPIELĀDĒJAMĀ BILDE:" field now contains the file path "C:\Users\cLOWN\Pictur" and a "Browse..." button. A green bar at the bottom of the form displays the message "Profila bilde nomainīta".

Ilustrācija 17 - veiksmīga profila bildes maiņa

Lai nomainītu savu profila bildi, nepieciešams nospiest „Browse” pogu profila bildes maiņas formā (*Ilustrācija 16*) un norādīt uz jauno profila bildi. Sākumā tur rādīsies vecā profila bilde, taču veiksmīgas profila bildes maiņas gadījumā zem formas parādīsies attiecīgs uzraksts un nomainīsies attēls zem „Šobrīdējā profila bilde” lauka (*Ilustrācija 17*).

2.7. Jaunas galerijas pievienošana



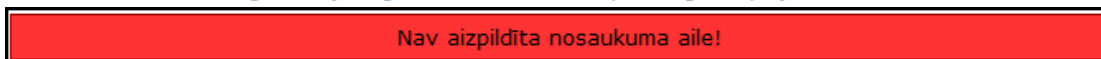
The screenshot shows a form titled "GALERIJAS APRAKSTS". It has two text input fields: "NOSAUKUMS:" with the value "Dokumentācijas galerija" and "APRAKSTS:" with the value "Galerija smukai bildei.". Below the fields is a "Pievienot galeriju" button.

Ilustrācija 18 - jaunas galerijas pievienošanas forma

Lai izveidotu jaunu galeriju nepieciešams aizpildīt jaunas galerijas pievienošanas formu (*Ilustrācija 18*). Ir jāieraksta nepieciešamā informācija (nosaukums ir obligāti aizpildāms, aprakstu var arī neaizpildīt, taču nosaukums nedrīkst būt garāks par 200 simboliem) un

jānospiež poga „Pievienot galeriju”. Ja galerija pievienosies veiksmīgi jūs tiksiet pārvests uz bilžu augšupielādes formu.

2.7.1. Jaunas galerijas pievienošanas kļūdu paziņojumi



Ilustrācija 19 - galerijas pievienošanas kļūdas paziņojums



Ilustrācija 20 - galerijas pievienošanas kļūdas paziņojums

Ja pievienojot galeriju parādās kļūdu paziņojums, tad nākamā rīcība jāizvērtē skatoties pēc kļūdu paziņojuma:

- **Nav aizpildīta nosaukuma aile!** – nozīmē, ka nav aizpildīta nosaukuma aile. Galeriju var pievienot tikai tad, ja ir aizpildīta nosaukuma aile. Aizpildiet to un mēģiniet pievienot galeriju vēlreiz. (*Ilustrācija 19*)
- **Nosaukums nedrīkst būt garāks par 200 simboliem** – nozīmē, ka ievadītais nosaukums ir garāks par 200 simboliem. Saīsiniet to līdz pieļaujamajam garumam un mēģiniet pievienot galeriju vēlreiz. (*Ilustrācija 20*)

2.8. Galerijas bilžu augšupielādēšana

BILŽU PIEVIENOŠANA

BILDE:

BILDES APRAKSTS:

Ilustrācija 21 - galerijas bilžu augšupielādēšana

BILŽU PIEVIENOŠANA

BILDE:

BILDES APRAKSTS:



Bilde veiksmīgi augšupielādēta

Ilustrācija 22 - veiksmīga galerijas bildes pievienošana

Lai pievienotu bildes savai galerijai nepieciešams aizpildīt galerijas bilžu pievienošanas formu – katrai bildei vienu reizi (*Ilustrācija 21*). Bilde apraksts nav obligāti jāaizpilda. Ja bilde veiksmīgi pievienosies galerijai parādīsies atbilstošs uzraksts zem formas un jūsu augšupielādētā bilde (*Ilustrācija 22*).

2.9. Galeriju apskate



Ilustrācija 23 - galeriju apskate

Lai apskatītu kādu galeriju, nepieciešams tikai uzspiest uz galerijas laukuma un tā automātiski atvērsies. (*Ilustrācija 23*)

2.10. Galerijas bilžu apskate



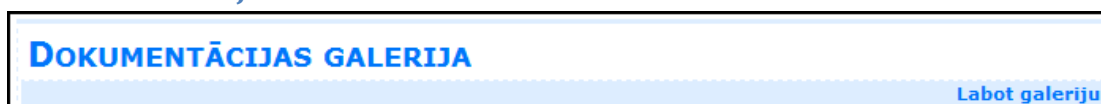
Ilustrācija 24 - galerijas bilžu apskate

Lai apstaigātu galerijas bildes var izvēlēties vienu no diviem veidiem:

- Izmantot bultiņas, kuras parādās, kad ar kursoru tiek uzbraukts uz bildes. Bultiņa pa kreisi jūs pārnesīs vienu bildi atpakaļ, turpretim bilde pa labi – uz priekšu.
- Izmantot „Iepriekšējā bilde” un „Nākamā bilde” saites, kuras dara tieši to, kas uz viņām rakstīts – pārnes jūs uz nākamo un iepriekšējo bildi.

Dzeltenajā laukumā rakstās bildes apraksts un starp bildes navigāciju jūs varat redzēt pie kuru bildi jūs šobrīd apskatāt un cik bildes ir kopā pieejamas konkrētajā galerijā. (Ilustrācija 24).

2.11. Galerijas labošana



Ilustrācija 25 - galerijas labošanas saite

Lai labotu kādu no savām galerijām, atveriet to un nospiediet uz saites „Labot galeriju” (Ilustrācija 25). Tas jūs aizvedīs uz bilžu augšupielādes formu, kur varēsiet pievienot savai galerijai vēl bildes.

2.12. Jaunas ātrās ziņas pievienošana

PIEVIEŅO JAUNU NOTIKUMU

ZIŅAS DATI

TEKSTS:
Mācos kā pievienot jaunu ziņu.

KRĀSA:

Pievienot

Ilustrācija 26 - jaunas ātrās ziņas pievienošanas forma

AŠĀS ZIŅAS

Pievienot ziņu

Lietotājs: Mācos kā pievienot jaunu ziņu.
Dzēst

Ziņa pievienota veiksmīgi!

Ilustrācija 27 - pievienota jauna ātrā ziņa

Lai pievienotu jaunu ātro ziņu nepieciešams aizpildīt ātrās ziņas pievienošanas formu. Lauks „Krāsa” nav obligāts un to var atstāt tukšu (*Ilustrācija 26*). Ja jaunā ziņa veiksmīgi pievienosies, tad parādīsies attiecīgs paziņojums un ātro ziņu laukā parādīsies jūsu ziņa (*Ilustrācija 27*).

2.13. Ātro ziņu dzēšana

Lietotājs: Mācos kā pievienot jaunu ziņu.
Dzēst

Ilustrācija 28 - ātro ziņu dzēšana

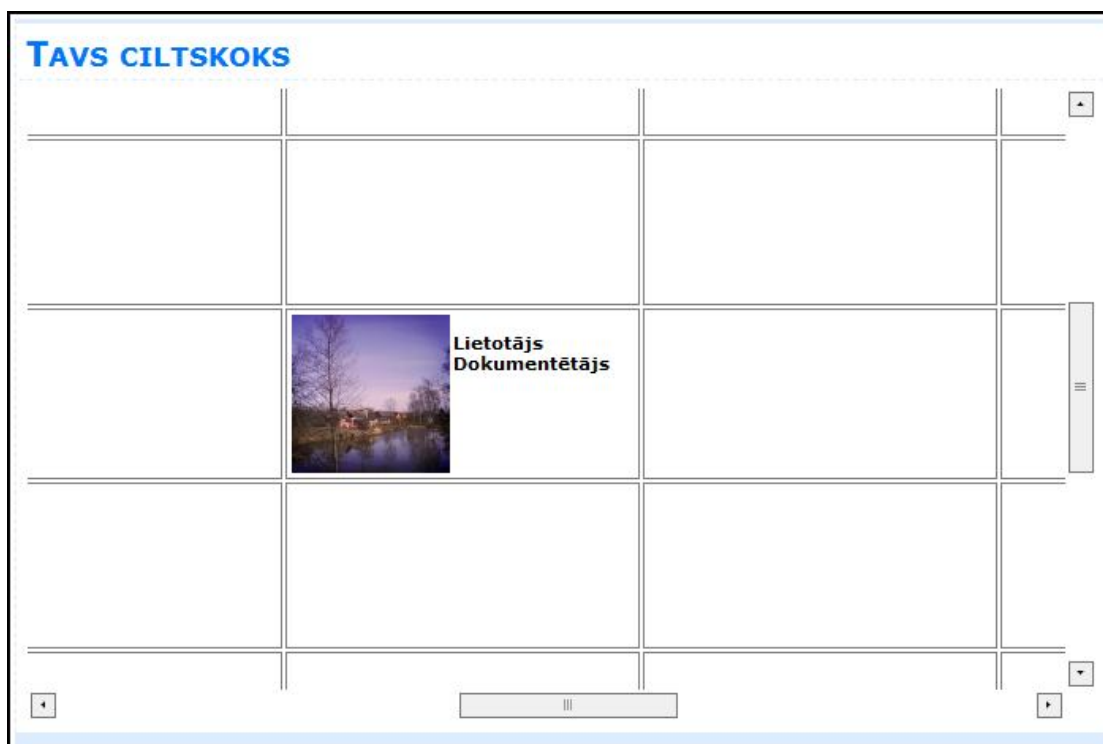
Ziņa veiksmīgi izdzēsta!

Ilustrācija 29 - veiksmīgi izdzēsta ātrā ziņa

Ja vēlaties izdzēst ātro ziņu, viss, kas nepieciešams ir nospiest saiti „Dzēst” (*Ilustrācija 28*) un jūsu ātrā ziņa tiks izdzēsta. Veiksmīgas izdzēšanas gadījumā parādīsies attiecīgs paziņojums (*Ilustrācija 29*).

2.14. Ciltskoka izmantošana

2.14.1. Sevis atrašana ciltskokā



Ilustrācija 30 - sevis atrašana ciltskokā

Lai atrastu sevi ciltskokā nepieciešams aiziet uz ciltskoka laukuma centru. Ciltskoka laukumu var uzskatīt kā koordināšu sistēmu, kur jūs esat punktā (0; 0) un tāpat jebkurš, kuram tiek pievienoti jaunu radi tiek uzskatīts kā centrs.

2.14.2. Jauna ciltskoka lietotāja pievienošana, ja viņš nav reģistrējies sistēmā



Ilustrācija 31 - jauna ciltskoka lietotāja pievienošana



Ilustrācija 32 - jauna ciltskoka lietotāja pievienošana

VĀRDS:

UZVĀRDS:

Pievienot Meklēt

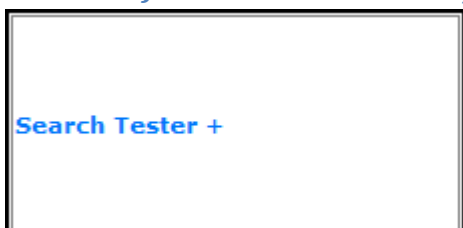
Ilustrācija 33 - jauna ciltskoka lietotāja pievienošana



Ilustrācija 34 - jauna ciltskoka lietotāja pievienošana

Ja vēlaties pievienot jaunu ciltskoka lietotāju, kurš nav reģistrējies sistēmā www.mesesam.lv, tad nepieciešams uzspiest uz savas bildes un parādīsies ciltskoka navigācija (Ilustrācija 31). Tālāk nepieciešams pievienot „+” pogu, kura nozīmē, ka vēlaties pievienot jaunu ciltskoka locekli. Tad vajadzēs izvēlēties kādu radu jūs vēlaties pievienot (Ilustrācija 32). Nospiežat uz radu tipa, kuru vēlaties pievienot un tas izveidos jaunu lauku pozīcijā, kura ir atkarīga no tā, kādu tipu jūs pievienojat. Jaunpievienotajā formā ir nepieciešams ievadīt ciltskoka locekļa vārdu un uzvārdu (Ilustrācija 33). Kad tas ir izdarīts vajag nospiegt pogu „Pievienot”. Ja ciltskoka loceklis pievienosies veiksmīgi formas vietā parādīsies jaunpievienotais ciltskoka loceklis.

2.14.3. Jauna ciltskoka lietotāja pievienošana, ja viņš ir reģistrējies sistēmā



Ilustrācija 35 - ciltskoka lietotājs, kurš ir reģistrējies sistēmā



Ilustrācija 36 - ciltskoka lietotājs, kurš ir reģistrēts sistēmā

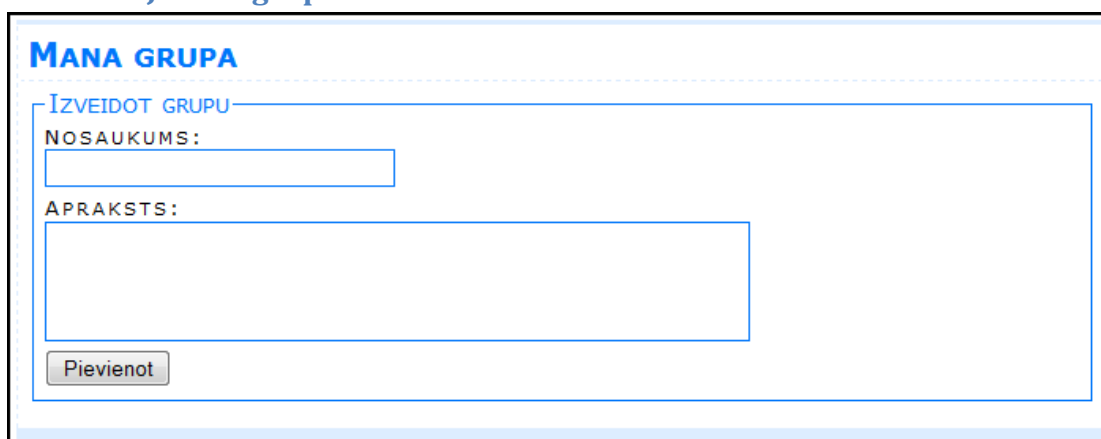
Ja vēlaties pievienot ciltskoka lietotāju, kurš ir reģistrējies sistēmā sekojiet līdzī padomiem, kuri doti punktā, ja pievienojat kādu, kurš nav reģistrējies sistēmā, līdz vietai, kur nonākat līdz pogas „Pievienot” spiešanai. Tā vietā spiežat pogu „Meklēt” un tajā pašā laukā

parādīsies visi lietotāji, kuri atbilst jūsu meklēšanas kritērijiem (*Ilustrācija 35*). Ja vēlaties apskatīt kāda lietotāja profilu nepieciešams nospiegt uz lietotāja vārda vai uzvārda. Ja vēlaties kādu lietotāju pievienot ciltskokam jāspiež „+” saite. Ja viss noritēs bez kļūdām, tad tajā pat laikā parādīsies jaunpievienotais ciltskoka loceklis, kura dati tiek ņemti no reāli eksistējoša sistēmas www.mesesam.lv lietotāja.

2.14.4. Ciltskoka locekļa nostādīšana centrā

Ja vēlaties pievienot kādam ciltskoka loceklim, kurš šobrīd nav centrā (atverot ciltskoku, jūs būsiet ciltskoka centrā) jaunu radu, to nepieciešams nostādīt centrā. Lai to izdarītu atveriet ciltskoka navigāciju (*Ilustrācija 31*) un nospiežat pogu „?”. Tas izvēlēs ciltskoka locekli nostādīs centrā parādot tā tiešos radniekus un atļaujot tam pievienot jaunus radus.

2.15. Jaunas grupas izveidošana



The screenshot shows a web form titled "MANA GRUPA" with a sub-header "IZVEIDOT GRUPU". Inside the form, there are two input fields: "NOSAUKUMS:" and "APRAKSTS:". Below these fields is a button labeled "Pievienot".

Ilustrācija 37 - jaunas grupas pievienošanas forma

Ja vēlaties izveidot savu grupu, tad nepieciešams iet uz sadaļu „Mana grupa” un ja jums jau nebūs kāda grupa izveidota, tad jums piedāvās vienu izveidot. Vajadzēs aizpildīt jaunas grupas izveidošanas formu (*Ilustrācija 37*) ar obligātu lauku „Nosaukums” un neobligātu lauku „Apraksts”. Kad nepieciešamā informācija ir ievadīta vajag nospiegt pogu „Pievienot” un ja ievaddatos nebūs kļūdu, tad lapa atjaunosies un jūs varēsiet pievienot grupā jaunus lietotājus.


2.15.1. Jaunas grupas izveidošana kļūdu paziņojumi

Nosaukumam jābūt aizpildītam un tas nedrīkst būt garāks par 150 simboliem!

Ilustrācija 38 - jaunas grupas pievienošanas kļūdas paziņojums

Ja pievienojot grupu sastopaties ar kļūdu paziņojumu – **Nosaukumam jābūt aizpildītam un tas nedrīkst būt garāks par 150 simboliem** (*Ilustrācija 38*) – tad pārbaudiet vai nosaukuma lauks nav tukšs vai tajā ierakstītā informācija nav garāka par 150 simboliem. Izlabojiet to un mēģiniet pievienot grupu vēlreiz.


2.16. Jaunu lietotāju pievienošana grupā

IETILPST	PROFILA BILDE	INFORMĀCIJA
<input type="checkbox"/>		<p>Search Tester</p> <p>Man garšo pienešas un es esmu izveidots tikai tādēļ, lai redzētu kā darbojas meklēšana :D</p>

Ilustrācija 39 - jaunu lietotāju pievienošana grupai

IETILPST	PROFILA BILDE	INFORMĀCIJA
<input checked="" type="checkbox"/>		<p>Search Tester</p> <p>Man garšo pienešas un es esmu izveidots tikai tādēļ, lai redzētu kā darbojas meklēšana :D</p>
<p>Uzaicinājums pievienoties jūsu grupai ir nosūtīts!</p>		

Ilustrācija 40 - jauns lietotājs uzaicināts grupai

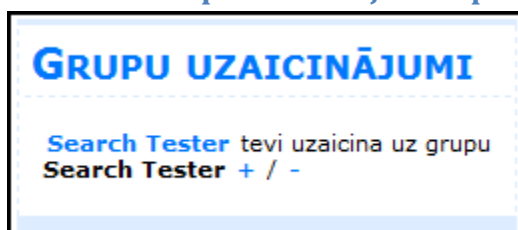
IETILPST	PROFILA BILDE	INFORMĀCIJA
<input checked="" type="checkbox"/>		<p>Search Tester</p> <p>Man garšo pienešas un es esmu izveidots tikai tādēļ, lai redzētu kā darbojas meklēšana :D</p>

Ilustrācija 41 - lietotājs pievienojies jūsu grupai

Lai pievienotu jaunu lietotāju savai grupai, jums vispirms vajag izveidot savu grupu. Ja tas ir izdarīts, tad pievienot jūs varat tos lietotājus, kuri ir jūsu ciltskokā. Visi, kas ir jūsu ciltskokā varēs lasīt jūsu ātrās ziņas un jūs varēsiet lasīt viņu ātrās ziņas.

Lai pievienotu jaunu lietotāju jums vajag ieķeksēt lauku pie atbilstošā lietotāja „Ietilpst” kolonnā (Ilustrācija 39). Kad tas ir izdarīts, parādīsies paziņojums par to un lietotāja lauks iekrāsosies dzeltenā krāsā (Ilustrācija 40). Tas nozīmē, ka uzaicinājums pievienoties jūsu grupai ir nosūtīts, un ja otrs lietotājs to apstiprināts, tad viņš pilntiesīgi skaitīsies jūsu grupā un viņa lauks iekrāsosies zaļā krāsā (Ilustrācija 41).

2.17. Grupu uzaicinājumu apstiprināšana/noraidīšana



Ilustrācija 42 - grupu uzaicinājums

Uzaicinājumus iesaistīties kādā grupā var apskatīt „Grupu uzaicinājumi” panelī. Uzspiežot uz lietotāja vārda vai uzvārda, kas jūs aicina, var apskatīt lietotāja profilu. Nospiežot „+” jūs piekrītat aicinājumam un iesaistāties grupā. Ja nospiežat „-”, tad atsakāties iesaistīties grupā. (Ilustrācija 42).

2.18. Grupas, kurās es esmu

PROFILA BILDE	INFORMĀCIJA PAR LIETOTĀJU
	Search Tester Man garšo pienešas un es esmu izveidots tikai tādēļ, lai redzētu kā darbojas meklēšana :D
	Rolands Atvars Esmu skaists un foršs! Un mīļš arī :) Un kruts!
	Lietotājs Dokumentētājs Es testēju lietotāja dokumentāciju.

Ilustrācija 43 - grupas kurās es esmu

Arī man gribās izveidot kādu grupiņu - te piegāsim varētu sanākt visi tester...

Ilustrācija 44 - sīkāka grupas informācija

Jūs varat apskatīties kādās grupās jūs šobrīd atrodaties. Pirmais grupas lietotājs vienmēr ir grupas dibinātājs (Ilustrācija 43). Ja vēlaties apskatīt sīkāku informāciju par grupu nospiežat uz palielināmā stikla un jums parādīsies sīkāka informācija par grupu (Ilustrācija 44).

Ja vēlaties iziet no grupas nospiežat sarkano krustiņu grupas augšējā daļā un tas jūs izdzēsīs no grupas.

Projekta organizācija

Sistēma www.mesesam.lv tika izstrādāta izmantojot ūdenskrituma modeli, tomēr jāņem vērā, ka izstrādātā sistēma vēl netiek skaitīta kā pabeigta un nav paredzēta publiskai rādīšanai. Sistēmas tapšanas posmi:

- Tika izpētītas citas alternatīvas sistēmas, kuras sniedz līdzīgu iespēju klāstu kā sistēma www.mesesam.lv
- Tika izdomāts sistēmas kopējais koncepts – kādam vajadzētu izskatīties dizainam, kādām vajadzētu būt iespējām sistēmā.
- Tika izdomāts kā sadalīt kopēju sistēmas konceptu daļās, jo visa sistēmas izstrāde prasītu pārāk daudz laika un tās apjoms būtu par lielu kvalifikācijas darbam
- Tika izstrādāta Programmatūras Prasību Specifikācija
- Tika izstrādāts Programmatūras Projektējuma Apraksts
- Tika sazīmētas sistēmas dizaina skices, izstrādāts dizains un uzprogrammēts sistēmas kods
- Tika veikta sistēmas testēšana
- Tika izstrādāta Lietotāja Dokumentācija

Visu projektu radīja (izdomāja, izprojektēja un uzprogrammēja) kvalifikācijas darba autors.

Kvalitātes nodrošināšana

Lai nodrošinātu, ka izveidotā sistēma darbotos kā paredzēts – tika veikta testēšana. Testēšana tika veikta no lietotāja skatu punkta izanalizējot iespējamās un tipiskās lietotāja „ceļus”, kā viņš varētu pētīt sistēmu un testējot katru viņa darbību. Visi neizdevušies testi tika uzskaitīti un tika meklēti iespējamie risinājumi, lai novērstu atrastās kļūdas. Svarīgākās kļūdas tika apskatītas pirmās. Ne visas sistēmas kļūdas tika novērstas, jo tā kā sistēma vēl nav paredzēta publiskai parādīšanai, tad tas nav tik svarīgi.

Darbietilpības novērtējums

1. Izmantojot COCOMO basic metodi

$$\text{Cilvēkmēneši} = a_b * (KLOC)^{b_b} = 2,4 * 4,3^{1,05} = 2,4 * 4,6 = 11,04$$

$$\text{Mēneši} = c_b(\text{Effort Applied})^{d_b} = 2,5 * 11,04^{0,38} = 2,5 * 2,5 = 6,25$$

2. Izmantojot COCOMO II metodi

Faktors	Vērtējums	Komentārs
Iepriekšējā pieredze šādu uzdevumu risināšanā (SF1)	3	Autoram ir pieredze php, html, css un javascript izmantošanā, bet nav pieredze jQuery bibliotēkas izmantošanā, kā arī nav bijusi saskarsme ar ciltskoku veidošanu un liela daudzuma AJAX tehnoloģiju izmantošanu vienā lapā.
Darbu ietekmē spēcīgi aparatūras, programmatūras, laika u.c. ierobežojumi (SF2)	2	Lai nodrošinātu ātru ciltskoku ģenerēšanu un operēšanu ar tiem, ir jāizstrādā salīdzinoši ātri algoritmi, kuri nosaka visas attiecības ciltskoku locekļu starpā.
Visas iespējamās atkāpes no standatrisinājumiem ir definētas un iekļautas plānos (SF3)	1	Iespējamās atkāpes ir labi definētas.
Izstrādātāju komanda (SF4)	0	Savstarpējās komunikācijas nav nepieciešamas, jo darbu projektu izstrādā tikai viens cilvēks.
Produkta izstrādes procesa organizācija. (SF5)	2	Produkta izstrādes process tika labi organizēts.

$$B = 1.01 + 0.01 * \sum_{j=1}^5 SF_j = 1.01 + 0.01 * (3 + 2 + 1 + 0 + 2) = 1.01 + 0.01 * 8$$

$$= 1.01 + 0.08 = 1.09$$

Faktors	Vērtējums	Komentārs
Izstrādātāju spējas (EM1)	1.1	Izstrādātāja spējas ir vidējas
Uzsvars uz drošumu un dokumentētību	1.2	Drošums šajā projektā nav svarīgs, turpretim dokumentācija ir ļoti nepieciešama.
Izstrādājamā produkta sarežģītība	1.4	Izveidot ciltskoka izveidošanas un tālākas personalizēšanas algoritmu nav triviāls uzdevums. Tāpat izveidot ērtu, no dizaina viedokļa un maksimāli efektīvi izmantot AJAX spējas prasa daudz plānošanas.
Datu bāzes izmērs	1	Gatava produkta datu bāze ir visnotaļ maza un aizpildījums ir tikai testa ieraksti, turpretim sākot lietot produktu, gaidāms liels ierakstu pieplūdums
Drošuma prasības = (1.2 + 1.4 + 1)3 (EM2)	1.2	
Atkārtotās pielietojamības prasības (EM3)	0.9	Produkts netiks atkārtoti izmantots.
Produkta apjoma ierobežojumi un	1.2	Produktam nevajadzētu dziļi pārsniegt 3

ātrdarbība. (EM4)		mēnešu darbu vienam cilvēkam. Kā arī ciltskoka ģenerēšanai un veidošanai nevajadzētu būt lēnai.
Izstrādes vides stabilitāte (EM5)	0.9	Izstrādes vide ir ļoti stabila.
Izstrādes kalendārais plānojums (EM6)	1	Kalendārais plānojums ir visnotaļ labi saplānots.

$$E = \prod_{i=1}^6 EM_i = 1.1 * 1.2 * 0.9 * 1.2 * 0.9 * 1 = 1.3$$

$$PM = 2.5 * \left(\frac{S}{1000}\right)^B * E = 2.5 * 4,3^{1,09} * 1.3 = 2.5 * 5 * 1.3 = 16,25$$

$$CM = 3 * PM^{(0.33+0.2(B-1.01))} * 1 = 3 * 16,25^{(0.33+0.2(1.09-1.01))} * 1 = 3 * 16.25^{0.3} * 1 = 3 * 2,3 * 1 = 6,9$$

B – izstrādes risku ietekme uz darbietilpību

E – izstrādes vides ietekme uz darbietilpību

PM – personmēneši

CM – kalendārie mēneši

3. Secinājumi par iegūtajiem rezultātiem

Iegūtie rezultāti īpaši neatbilst reāli pavadītajam laikam pie projekta. Basic COCOMO šoreiz deva pareizāku rezultātu, droši vien tādēļ, ka tas ir paredzēts mazākiem projektiem, kur nav daudz cilvēku, bet visi, kas ir pie projekta ir ar labāk zināšanām, kā arī šajā projektā bija.

Iespējams savu kļūdu procentu deva tas, ka koda rindiņu skaitā iekļaujas ne tikai PHP un JavaScript rakstītais kods, bet arī nedaudz HTML marķējuma. Tas ir darīts tādēļ, ka dizaina failos PHP un HTML ir samiksējušies kopā un tos atdalīt ir sarežģīti. Piedevām arī lapas noformēšana prasa savu laiku, tādēļ autors neuzskata, ka HTML marķējuma kodu nevajadzētu ieskaitīt, bet tas droši vien rada pārmērīgu lielos darbietilpības rezultātus. Ja koda rindiņu skaitā tiktu ieskaitīts arī CSS formatējuma kods, tad rezultāts būtu vēl nepareizāks.

Reāli izstrādājot projektu tika pavadīts apmēram uz pusi mazāks laiks nekā piedāvā COCOMO sniegtie rezultāti.

Programmatūras kods

jEdit - Relation.php

```
1 <?php
2 class Relation extends Connection
3 {
4     private $user; //stores classes User object
5
6     public function __construct($user)
7     {
8         $this->user = $user;
9         parent::__construct();
10    }
11
12    /*
13    Generates Relation grid based on given width and height.
14    Puts relationMembers user with fromID in center if fromID is not 0
15    Puts users profile info in center if fromID is 0
16    */
17    public function generateGrid($width, $height, $fromID = 0)
18    {
19        include(TMP_PATH . "relationGrid.php");
20    }
21
22    /*
23    Draws page relations page contents
24    */
25    public function drawRelationView()
26    {
27        include(TMP_PATH . "relationView.php");
28    }
29
30    /*
31    Gets relationMembers id who created relationMember with given id
32    */
33    public function getFromID($id)
34    {
35        $var = parent::escapeParams($id);
36        $cmdStr = "SELECT fromID FROM relations WHERE id = %d";
37        $cmdStr = sprintf($cmdStr, $var[0]);
38        $result = parent::executeQuery($cmdStr);
39        $row = mysql_fetch_row($result);
40        return $row[0];
41    }
42
43    /*
44    Generates relation members profile field with given id
45    If id is 0 info comes from users table
46    If root is true both coordinates are 0
47    If temporary coordinates are not false they come from relations table
48    */
49    public function drawProfileField($id = 0, $root = false, $xTmp = false,
50    $yTmp = false)
51    {
52        if($id == 0)
53        {
54            $row = $this->user->getRelationInfo($_SESSION["cUid"]);
55            $x = ($xTmp == false) ? 0 : $xTmp;
56            $y = ($yTmp == false) ? 0 : $yTmp;
```

```

56     $fromID = 0;
57 }
58 else
59 {
60     $row = $this->getMemberInfo($id);
61     $x = ($root) ? 0 : $row["xCord"];
62     $x = ($xTmp == false) ? $x : $xTmp;
63     $y = ($root) ? 0 : $row["yCord"];
64     $y = ($yTmp == false) ? $y : $yTmp;
65     $fromID = $this->getFromID($id);
66     $row = ($row["realID"] == 0) ? $row : $this->user-
>getRelationInfo($row["realID"]);
67 }
68 include(TMP_PATH . "relationPersonField.php");
69 }
70
71 /*
72 Draws menu where user can add new relatives
73 */
74 public function drawProfileFieldAddMenu($x, $y, $fromID)
75 {
76     include(TMP_PATH . "profileFieldAddMenu.php");
77 }
78
79 /*
80 Generates new relationMembers input form
81 */
82 public function addBlankField($fromX, $fromY, $fromID, $type)
83 {
84     list($x, $y) = $this->renderCords($fromX, $fromY, $fromID, $type);
85     echo $x . ";" . $y . "&&";
86     include(TMP_PATH . "blankFieldForm.php");
87 }
88
89 /*
90 Calculates new relationMembers coordinates from given relationMembers id
and
91 coordinates and new members type
92 */
93 public function renderCords($fromX, $fromY, $fromID, $type)
94 {
95     $x = 0; //x coordinate to new field
96     $y = 0; //y coordinate to new field
97
98     if(($fromX == 0) and ($fromY == 0))
99     {
100         switch($type)
101         {
102             /*
103             * case of father
104             */
105             case 1:
106                 $x = $fromX - 1;
107                 $y = $fromY + 1;
108                 $x -= $this->siblingCount($fromID);
109                 break;
110
111             /*
112             * case of mother
113             */
114             case 2:

```

```

115         $x = $fromX + 1;
116         $y = $fromY + 1;
117     break;
118
119     /*
120     * case of brother
121     */
122     case 3:
123
124     /*
125     * case of sister
126     */
127     case 4:
128         $x = $fromX - 1 - $this->siblingCount($fromID);
129         $y = $y;
130     break;
131
132     /*
133     * case of husband
134     */
135     case 5:
136
137     /*
138     * case of wife
139     */
140     case 6:
141         $x = $x + 1 + $this->childrenCount($fromID);
142         $y = $y;
143     break;
144
145     /*
146     * case of son
147     */
148     case 7:
149
150     /*
151     * case of daughter
152     */
153     case 8:
154         $x = $x + 1 + $this->childrenCount($fromID);
155         $y = $y - 1;
156     break;
157     }
158 }
159
160 return array($x, $y);
161 }
162
163 /*
164 Inserts new relationMember in Data Base
165 If type is brother/sister or son/daughter that checks and moves
166 father/wife/husband
167 field
168 */
169 public function insertNewMember($xCord, $yCord, $type, $belongsTo, $fromID,
170 $realID, $name, $surname)
171 {
172     $vars = array($xCord, $yCord, $type, $belongsTo, $fromID, $realID,
173 $name, $surname);
174     $variables = array("realID", "name", "surname");
175     $values = array($vars[5], $vars[6], $vars[7]);

```

```

173     $cmdStr = parent::buildInsert("relationMembers", $variables, $values);
174     $toID = parent::executeUpdate($cmdStr);
175     $variables = array("xCord", "yCord", "type", "belongsTo", "fromID",
176     "toID");
177     $values = array($vars[0], $vars[1], $vars[2], $vars[3], $vars[4],
178     $toID);
179     $cmdStr = parent::buildInsert("relations", $variables, $values);
180     $sid = parent::executeUpdate($cmdStr);
181     if($type == 3 or $type == 4)
182     {
183         $type = 1;
184         $mustMove = $this->mustMove($fromID, $type, $belongsTo);
185     }
186     elseif($type == 7 or $type == 8)
187     {
188         $type = 6;
189         $mustMove = $this->mustMove($fromID, $type, $belongsTo);
190     }
191     else
192     {
193         $mustMove = 0;
194     }
195     if(is_array($mustMove))
196     {
197         $this->moveMember($mustMove[0], $type);
198     }
199     return array($sid, $mustMove);
200 }
201
202 /*
203 Gets relationMembers info from given relations id
204 */
205 public function getMemberInfo($id)
206 {
207     $var = parent::escapeParams($id);
208     $cmdStr = "SELECT relationMembers.realID, relationMembers.name,
209     relationMembers.surname, relations.xCord, relations.yCord FROM
210     relations " .
211     "INNER JOIN relationMembers ON relations.toID = relationMembers.id
212     WHERE relations.id = %d";
213     $cmdStr = sprintf($cmdStr, $var[0]);
214     $result = parent::executeQuery($cmdStr);
215     $row = mysql_fetch_assoc($result);
216     return $row;
217 }
218
219 /*
220 Checks if cell with given coordinates is full
221 */
222 public function cellIsFilled($x, $y, $belongsTo, $fromID = 0)
223 {
224     $vars = parent::escapeParams($x, $y, $belongsTo, $fromID);
225     $cmdStr = "SELECT id FROM relations WHERE belongsTo = %d && xCord = %d &&
226     yCord = %d && fromID =
227     %d";
228     $cmdStr = sprintf($cmdStr, $vars[2], $vars[0], $vars[1], $vars[3]);
229     $result = parent::executeQuery($cmdStr);
230     if(mysql_num_rows($result) > 0)
231     {
232         $row = mysql_fetch_row($result);
233         return $row[0];
234     }
235 }

```

```

227     }
228     else
229     {
230         return 0;
231     }
232 }
233
234 /*
235 Gets sibling count for given relationMember
236 */
237 public function siblingCount($fromID)
238 {
239     $var = parent::escapeParams($fromID);
240     $cmdStr = "SELECT COUNT(id) FROM relations WHERE fromID = %d && belongsTo
= '%s' && (type = 3 ||
type = 4)";
241     $cmdStr = sprintf($cmdStr, $var[0], $_SESSION["cUid"]);
242     $result = parent::executeQuery($cmdStr);
243     $row = mysql_fetch_row($result);
244     return $row[0];
245 }
246
247 /*
248 Gets children count for given relationMember
249 */
250 public function childrenCount($fromID)
251 {
252     $var = parent::escapeParams($fromID);
253     $cmdStr = "SELECT COUNT(id) FROM relations WHERE fromID = %d && belongsTo
= '%s' && (type = 7 ||
type = 8)";
254     $cmdStr = sprintf($cmdStr, $var[0], $_SESSION["cUid"]);
255     $result = parent::executeQuery($cmdStr);
256     $row = mysql_fetch_row($result);
257     return $row[0];
258 }
259
260 /*
261 Draws search results in field with given coordinates
262 */
263 public function drawPersonSearchResults($name, $surname, $fromID, $type,
$x, $y)
264 {
265     $result = $this->user->searchUser($name, $surname);
266     include(TMP_PATH . "personSearchResults.php");
267 }
268
269 /*
270 Checks if relationMembers field with given id must be moved
271 */
272 public function mustMove($fromID, $type, $belongsTo)
273 {
274     $val = parent::escapeParams($fromID, $type, $belongsTo);
275     $cmdStr = "SELECT id, xCord, yCord FROM relations WHERE fromID = %d &&
belongsTo = %d && type = %d";
276     $cmdStr = sprintf($cmdStr, $val[0], $val[2], $val[1]);
277     $result = parent::executeQuery($cmdStr);
278     if(mysql_num_rows($result) == 0)
279     {
280         return 0;
281     }

```

```

282     else
283     {
284         $row = mysql_fetch_row($result);
285         return $row;
286     }
287 }
288
289 /*
290 Performs field moving
291 */
292 public function moveMember($id, $type)
293 {
294     $val = parent::escapeParams($id);
295     $action = ($type == 1) ? "xCord = xCord - 1" : "xCord = xCord + 1";
296     $cmdStr = "UPDATE relations SET " . $action . " WHERE id = %d";
297     $cmdStr = sprintf($cmdStr, $val[0]);
298     parent::executeUpdate($cmdStr);
299 }
300
301 /*
302 Gets coordinates for given relation id
303 */
304 public function getCords($id)
305 {
306     $val = parent::escapeParams($id);
307     $cmdStr = "SELECT xCord, yCord FROM relations WHERE id = %d";
308     $cmdStr = sprintf($cmdStr, $val[0]);
309     $result = parent::executeQuery($cmdStr);
310     $row = mysql_fetch_row($result);
311     return array($row[0], $row[1]);
312 }
313
314 /*
315 Gets siblings for relationMembers user with given id
316 */
317 public function getSiblings($fromID)
318 {
319     $val = parent::escapeParams($fromID);
320     $cmdStr = "SELECT toID FROM relations WHERE fromID = %d && (type = 3 ||
type = 4) && belongsTo = %d";
321     $cmdStr = sprintf($cmdStr, $val[0], $_SESSION["cUid"]);
322     $result = parent::executeQuery($cmdStr);
323     return $result;
324 }
325
326 /*
327 Gets parents for relationMembers user with given id
328 */
329 public function getParents($fromID)
330 {
331     $val = parent::escapeParams($fromID);
332     $cmdStr = "SELECT toID FROM relations WHERE fromID = %d && type = 2 &&
belongsTo = %d";
333     $cmdStr = sprintf($cmdStr, $val[0], $_SESSION["cUid"]);
334     $result = parent::executeQuery($cmdStr);
335     $row = mysql_fetch_row($result);
336     $mothersID = $row[0];
337     $cmdStr = "SELECT toID FROM relations WHERE fromID = %d && type = 1 &&
belongsTo = %d";
338     $cmdStr = sprintf($cmdStr, $val[0], $_SESSION["cUid"]);
339     $result = parent::executeQuery($cmdStr);

```

```

340     $row = mysql_fetch_row($result);
341     $fathersID = $row[0];
342     return array($fathersID, $mothersID);
343 }
344
345 /*
346 Gets children for given relation members user with given id
347 */
348 public function getChildren($fromID)
349 {
350     $val = parent::escapeParams($fromID);
351     $cmdStr = "SELECT toID FROM relations WHERE fromID = %d && (type = 7 ||
type = 8) && belongsTo = %d";
352     $cmdStr = sprintf($cmdStr, $var[0], $_SESSION["cUid"]);
353     $result = parent::executeQuery($cmdStr);
354     return $result;
355 }
356
357 /*
358 Gets partner for given relationMember with given id
359 */
360 public function getPartner($fromID)
361 {
362     $val = parent::escapeParams($fromID);
363     $cmdStr = "SELECT toID, type FROM relations WHERE fromID = %d && (type =
6 || type = 5) && belongsTo = %d";
364     $cmdStr = sprintf($cmdStr, $var[0], $_SESSION["cUid"]);
365     $result = parent::executeQuery($cmdStr);
366     $row = mysql_fetch_row($result);
367     return array($row[0], $row[1]);
368 }
369
370 /*
371 Gets who created given relationMember
372 */
373 public function getRootFromData($id)
374 {
375     $val = parent::escapeParams($id);
376     $cmdStr = "SELECT fromID, type FROM relations WHERE toID = %d &&
belongsTo = %d";
377     $cmdStr = sprintf($cmdStr, $val[0], $_SESSION["cUid"]);
378     $result = parent::executeQuery($cmdStr);
379     $row = mysql_fetch_assoc($result);
380     return array($row["fromID"], $row["type"]);
381 }
382
383 /*
384 Generates XML with re-calculated coordinates and relationMembers id`s for
385 new root relationMember
386 */
387 public function makeUncordRelationsXML($id)
388 {
389     $xml = "";
390     list($fromID, $fromType) = $this->getRootFromData($id);
391     //if type is parents it means that to new root it is children
392     if($fromType == 1 or $fromType == 2)
393     {
394         $x = 0;
395         $y = 0;
396         $siblingResult = $this->getSiblings($fromID);
397         $childrenCount = mysql_num_rows($siblingResult);

```

```

398 while($row = mysql_fetch_assoc($siblingResult))
399 {
400     if($row["toID"] == $fromID)
401     {
402         continue;
403     }
404     $x++;
405     $xml .= "<child>\n";
406     $xml .= "<id>" . $row["toID"] . "</id>\n";
407     $xml .= "<x>" . $x . "</x>\n";
408     $xml .= "<y>" . ($y - 1) . "</y>\n";
409     $xml .= "</child>\n";
410 }
411 $xml .= "<child>\n";
412 $xml .= "<id>" . $fromID . "</id>\n";
413 $xml .= "<x>" . ($x + 1) . "</x>\n";
414 $xml .= "<y>" . ($y - 1) . "</y>\n";
415 $xml .= "</child>\n";
416
417 $x = 0;
418 $y = 0;
419 list($fID, $mID) = $this->getParents($fromID);
420 $partnerID = ($fID == $id) ? $mID : $fID;
421 $xml .= "<partner>\n";
422 $xml .= "<id>" . $partnerID . "</id>\n";
423 $xml .= "<x>" . ($x + 2 + $childrenCount) . "</x>\n";
424 $xml .= "<y>" . $y . "</y>\n";
425 $xml .= "</partner>\n";
426 }
427 //if type is siblings that to new root they are also siblings
428 elseif($fromType == 3 or $fromType == 4)
429 {
430     $siblingsResult = $this->getSiblings($fromID);
431     $siblingsCount = mysql_num_rows($siblingsResult);
432     $x = 0;
433     $y = 0;
434     while($row = mysql_fetch_assoc($siblingsResult))
435     {
436         if($row["toID"] == $id)
437         {
438             continue;
439         }
440         $x--;
441         $xml .= "<sibling>\n";
442         $xml .= "<id>" . $row["toID"] . "</id>\n";
443         $xml .= "<x>" . $x . "</x>\n";
444         $xml .= "<y>" . $y . "</y>\n";
445         $xml .= "</sibling>\n";
446     }
447     $x--;
448     $xml .= "<sibling>\n";
449     $xml .= "<id>" . $fromID . "</id>\n";
450     $xml .= "<x>" . $x . "</x>\n";
451     $xml .= "<y>" . $y . "</y>\n";
452     $xml .= "</sibling>\n";
453
454     $x = 0;
455     $y = 0;
456     list($fID, $mID) = $this->getParents($fromID);
457     $xml .= "<father>\n";
458     $xml .= "<id>" . $fID . "</id>\n";

```

```

459     $xml .= "<x>" . ($x - 1 - $siblingsCount) . "</x>\n";
460     $xml .= "<y>" . ($y + 1) . "</y>\n";
461     $xml .= "</father>\n";
462     $xml .= "<mother>\n";
463     $xml .= "<id>" . $mID . "</id>\n";
464     $xml .= "<x>" . ($x + 1) . "</x>\n";
465     $xml .= "<y>" . ($y + 1) . "</y>\n";
466     $xml .= "</mother>\n";
467 }
468 //if type is wife or husband than now it is also wife or husband
469 elseif($fromType == 5 or $fromType == 6)
470 {
471     $childrenResult = $this->getChildren($fromID);
472     $childrenCount = mysql_num_rows($childrenResult);
473     $x = 0;
474     $y = 0;
475     while($row = mysql_fetch_assoc($childrenResult))
476     {
477         $x++;
478         $xml .= "<child>\n";
479         $xml .= "<id>" . $row["toID"] . "</id>\n";
480         $xml .= "<x>" . $x . "</x>\n";
481         $xml .= "<y>" . ($y - 1) . "</y>\n";
482         $xml .= "</child>\n";
483     }
484     $xml .= "<partner>\n";
485     $xml .= "<id>" . $fromID . "</id>\n";
486     $xml .= "<x>" . ($childrenCount + 1) . "</x>\n";
487     $xml .= "<y>" . $y . "</y>\n";
488     $xml .= "</partner>\n";
489 }
490 //if type is children than to new root type is parents
491 else
492 {
493     $x = 0;
494     $y = 0;
495     $childrenResult = $this->getChildren($fromID);
496     $childrenCount = mysql_num_rows($childrenResult);
497     while($row = mysql_fetch_assoc($childrenResult))
498     {
499         if($id == $row["toID"])
500         {
501             continue;
502         }
503         $x--;
504         $xml .= "<sibling>\n";
505         $xml .= "<id>" . $row["toID"] . "</id>\n";
506         $xml .= "<x>" . $x . "</x>\n";
507         $xml .= "<y>" . $y . "</y>\n";
508         $xml .= "</sibling>\n";
509     }
510
511     $x = 0;
512     $y = 0;
513     list($partnerID, $partnerType) = $this->getPartner($fromID);
514     if($partnerType == 2)
515     {
516         $motherID = $fromID;
517         $fatherID = $partnerID;
518     }
519     else

```

```

520     {
521         $motherID = $partnerID;
522         $fatherID = $fromID;
523     }
524     $xml .= "<mother>\n";
525     $xml .= "<id>" . $motherID . "</id>\n";
526     $xml .= "<x>" . ($x + 1) . "</x>\n";
527     $xml .= "<y>" . ($y + 1) . "</y>\n";
528     $xml .= "</mother>\n";
529
530     $xml .= "<father>\n";
531     $xml .= "<id>" . $fatherID . "</id>\n";
532     $xml .= "<x>" . ($x - $childrenCount) . "</x>\n";
533     $xml .= "<y>" . ($y + 1) . "</y>\n";
534     $xml .= "</father>\n";
535 }
536
537 return $xml;
538 }
539 }
540 ?>

```

jEdit - relationView.js

```

1  /*
2  Shows relationMembers profile menu
3  */
4  function showProfileMenu(x, y)
5  {
6      $('#profileFieldMenu_' + x + '_' + y).fadeIn('def');
7  }
8
9  /*
10 Hides relationMembers profile menu
11 */
12 function hideProfileMenu(x, y)
13 {
14     $('#profileFieldMenu_' + x + '_' + y).hide();
15 }
16
17 var visible = false;
18
19 function changeProfileMenuVisibility(x, y)
20 {
21     if(visible)
22     {
23         hideProfileMenu(x, y);
24         visible = false;
25     }
26     else
27     {
28         showProfileMenu(x, y);
29         visible = true;
30     }
31 }
32
33 /*
34 Changes image when mouse hovers over button
35 */
36 function imageHoverOn(image, name)
37 {

```

```

38     image.src = "./style/images/" + name + ".png";
39 }
40
41 /*
42 Changes image when mouse hovers off button
43 */
44 function imageHoverOff(image, name)
45 {
46     image.src = "./style/images/" + name + "_u.png";
47 }
48
49 /*
50 Sends AJAX request to draw new relationMembers add menu
51 */
52 function drawProfileFieldAddMenu(x, y, fromID)
53 {
54     $.ajax({
55         type: 'POST',
56         url: 'scr/drawProfileFieldAddMenu.htm',
57         data: 'fSubmit=submit&fX=' + x + '&fY=' + y + '&fFromID=' + fromID,
58         beforeSend: function() {
59             $('#personFieldInfo_' + x + '_' + y).hide();
60         },
61         success: function(data) {
62             $('#personFieldInfo_' + x + '_' + y).html(data);
63             $('#personFieldInfo_' + x + '_' + y).fadeIn('def');
64         },
65         failure: function(connObj, er, err) {
66             alert('Kļūda');
67         }
68     });
69 }
70
71 /*
72 Sends AJAX request to create new relationMember
73 */
74 function addMember(x, y, fromID, type)
75 {
76     $.ajax({
77         type: 'POST',
78         url: 'scr/addBlankField.htm',
79         data: 'fSubmit=submit&fX=' + x + '&fY=' + y + '&fType=' + type +
80 '&fFromID=' + fromID,
81         success: function(data) {
82             //alert(data);
83             var cords = getNewCords(data);
84             var cord = cords.split(';');
85             $('#cell_' + cord[0] + '_' + cord[1]).hide();
86             $('#cell_' + cord[0] + '_' + cord[1]).html(getNewHTML(data));
87             $('#cell_' + cord[0] + '_' + cord[1]).fadeIn('def');
88         },
89         failure: function(connObj, er, err) {
90             alert('Kļūda');
91         }
92     });
93 }
94 /*
95 Parses given data to get coordinates from data
96 */
97 function getNewCords(data)

```

```

98 {
99     var cords = data.split('&&');
100     return cords[0];
101 }
102
103 /*
104 Parses given data to extract html data from given data
105 */
106 function getNewHTML(data)
107 {
108     var d = data.split('&&');
109     return d[1];
110 }
111
112 /*
113 Sends AJAX request to insert new relationMember in Data Base
114 */
115 function insertNewMember(x, y, type, fromID, realID)
116 {
117     $.ajax({
118         type: 'POST',
119         url: 'scr/insertNewMember.htm',
120         data: 'fSubmit=submit&fX=' + x + '&fY=' + y + '&fType=' + type +
121             '&fFromID=' + fromID +
122             '&fName=' + $('#fName').val() + '&fSurname=' + $('#fSurname').val() +
123             '&fRealID=' + realID,
124         dataType: 'xml',
125         success: function(xml) {
126             $(xml).find('response').each(function () {
127                 var id = $(this).find('id').text();
128                 var mustMove = $(this).find('mustMove').text();
129                 if(mustMove != '0')
130                 {
131                     $(xml).find('response').each(function () {
132                         var xCord = $(this).find('xCord').text();
133                         var yCord = $(this).find('yCord').text();
134                         //alert('Must move from (' + xCord + '; ' + yCord + ')');
135                         moveItem(xCord, yCord, mustMove);
136                     });
137                 }
138                 drawProfileField(x, y, id);
139             });
140         },
141         failure: function(connObj, er, err) {
142             alert("Kļūda!");
143         }
144     });
145 }
146 /*
147 Sends AJAX request to get new coordinates so that it can move relationMembers
148 field
149 */
150 function moveItem(oldX, oldY, id)
151 {
152     $.ajax({
153         type: 'POST',
154         url: 'scr/getCords.htm',
155         data: 'fSubmit=submit&fId=' + id,
156         dataType: 'xml',
157         beforeSend: function() {

```

```

157     $('#cell_' + oldX + '_' + oldY).hide();
158     $('#cell_' + oldX + '_' + oldY).html('');
159 },
160 success:function(xml) {
161     $(xml).find('response').each(function () {
162         var xCord = $(this).find('xCord').text();
163         var yCord = $(this).find('yCord').text();
164         drawProfileField(xCord, yCord, id);
165         $('#cell_' + oldX + '_' + oldY).fadeIn('def');
166     });
167 },
168 failure:function(connObj, er, err) {
169     alert('Kļūda');
170 }
171 });
172 }
173
174 /*
175 Draws relationMembers profile field in given coordinates from given relation
176 members id
177 */
178 function drawProfileField(x, y, id)
179 {
180     $.ajax({
181         type: 'POST',
182         url: 'scr/drawProfileField.htm',
183         data: 'fSubmit=submit&fId=' + id + '&fX=' + x + '&fY=' + y,
184         beforeSend: function() {
185             $('#cell_' + x + '_' + y).hide();
186         },
187         success: function(data) {
188             $('#cell_' + x + '_' + y).html(data);
189             $('#cell_' + x + '_' + y).fadeIn('def');
190         },
191         failure: function(connObj, er, err) {
192             alert('Kļūda');
193         }
194     });
195 }
196
197 /*
198 Sends AJAX request to get search data from given input data
199 */
200 function searchMember(x, y, fromID, type)
201 {
202     $.ajax({
203         type: 'POST',
204         url: 'scr/searchProfileMember.htm',
205         data: 'fSubmit=submit&fName=' + $('#fName').val() + '&fSurname=' +
206             $('#fSurname').val() +
207             '&fFromID=' + fromID + '&fType=' + type + '&fX=' + x +
208             '&fY=' + y,
209         beforeSend: function() {
210             $('#cell_' + x + '_' + y).hide();
211         },
212         success: function(data) {
213             $('#cell_' + x + '_' + y).html(data);
214             $('#cell_' + x + '_' + y).fadeIn('def');
215         },
216         failure: function(connObj, er, err) {
217             alert("Kļūda!");

```

```

216     }
217   });
218 }
219
220 /*
221 Changes root and re-generates relation grid
222 */
223 function changeRoot(id)
224 {
225   $.ajax({
226     type: 'POST',
227     url: 'scr/changeRoot.htm',
228     data: 'fSubmit=submit&fID=' + id,
229     beforeSend: function() {
230       $('#relationContainer').hide();
231     },
232     success: function(data) {
233       $('#relationContainer').html(data);
234       $('#relationContainer').fadeIn('def');
235       drawNewRootsRelations(id);
236     },
237     failure: function(connObj, er, err) {
238       alert('Kļūda');
239     }
240   });
241 }
242
243 /*
244 Re-calculates new roots direct relatives coordinates
245 */
246 function drawNewRootsRelations(id)
247 {
248   $.ajax({
249     type: 'POST',
250     url: 'scr/drawRootsRelations.htm',
251     data: 'fSubmit=submit&fID=' + id,
252     dataType: 'xml',
253     complete: function(connObj, status) {
254       alert(connObj.responseText);
255       //alert(status);
256     },
257     success: function(xml) {
258       //alert(xml);
259       $(xml).find('response').each(function() {
260         $(this).find('sibling').each(function() {
261           var id = $(this).find('id').text();
262           var xCord = $(this).find('x').text();
263           var yCord = $(this).find('y').text();
264           drawProfileField(xCord, yCord, id);
265         });
266         $(this).find('father').each(function() {
267           var id = $(this).find('id').text();
268           var xCord = $(this).find('x').text();
269           var yCord = $(this).find('y').text();
270           drawProfileField(xCord, yCord, id);
271         });
272         $(this).find('mother').each(function() {
273           var id = $(this).find('id').text();
274           var xCord = $(this).find('x').text();
275           var yCord = $(this).find('y').text();

```

```

276         drawProfileField(xCord, yCord, id);
277     });
278     $(this).find('child').each(function() {
279         var id = $(this).find('id').text();
280         var xCord = $(this).find('x').text();
281         var yCord = $(this).find('y').text();
282         drawProfileField(xCord, yCord, id);
283     });
284     $(this).find('partner').each(function() {
285         var id = $(this).find('id').text();
286         var xCord = $(this).find('x').text();
287         var yCord = $(this).find('y').text();
288         drawProfileField(xCord, yCord, id);
289     });
290 });
291 },
292 failure:function(connObj, er, err) {
293     alert('Kļūda');
294 }
295 });
296 }

```

Dokumentāra lapa

Kvalifikācijas darbs „Ģimenes ciltskoku veidošanas un pārvaldīšanas portāls” izstrādāts Latvijas Universitātes Datorikas fakultātē

Ar savu parakstu apliecinu, ka kvalifikācijas darbs veikts patstāvīgi un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Rolands Atvars

Rekomendēju darbu aizstāvēšanai

Vadītājs: M.sc.dat. Krišs Rauhvargers

Recenzents: Juris Brants

Darbs iesniegts datorikas fakultātē 01.06.2009

Metodiķe: