

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

Orientēšanās sacensību portāls

Kvalifikācijas darbs

Autors: Ieva Rukšāne

Studenta apliecības nr.: ir09106

Darba vadītājs: Dainis Dosbergs

RĪGA, 2011

Anotācija

Kvalifikācijas darbā ir aprakstīta tīmekļa programmēšanas valodās realizēta tīmekļa vietne, kas ir paredzēta kā Orientēšanās sacensību portāls. Vietne tika izstrādāta, lai sportistiem būtu vienota vieta, kur meklēt un iegūt informāciju, pieteikties sacensībām. Portāls atvieglos komunikāciju sportistu starpā un ļaus ātrāk un efektīvāk izplatīt informāciju.

Portālu var apmeklēt jebkurš, taču reģistrētiem lietotājiem ir plašāka piekļuve portāla sadaļām.

Abstract

In this document has been described a homepage, which could be used as Orienteering competition web site. The page was created for athletes, so they could have a common place for information. This site could ease communication between athletes un could let them spread information much more effective.

The web site is free for everyone, but registered users have greater access to site's functions.

Saturs

| | |
|---|----|
| Ievads | 7 |
| 1. Programmatūras prasību specifikācija | 8 |
| 1.1. Ievads | 8 |
| 1.1.1. Dokumenta nolūks | 8 |
| 1.1.2. Darbības sfēra | 8 |
| 1.1.4. Saistība ar citiem dokumentiem..... | 8 |
| 1.2. Neformāls apraksts | 9 |
| 1.2.1. Produkta perspektīva | 9 |
| 1.2.2. Produkta funkcijas..... | 9 |
| 1.2.3. Lietotāja raksturiežmes..... | 9 |
| 1.2.4. Vispārējie ierobežojumi | 9 |
| 1.3. Funkcionālās prasības | 10 |
| 1.3.1. Lietotāja reģistrācija..... | 10 |
| 1.3.2. Lietotāja datu apskatīšana un rediģēšana | 11 |
| 1.3.3. Lietotāja statusa maiņa | 12 |
| 1.3.4. Lietotāja dzēšana..... | 13 |
| 1.3.5. Autorizācija | 14 |
| 1.3.6. Raksta pievienošana..... | 15 |
| 1.3.7. Komentāra pievienošana..... | 16 |
| 1.3.8. Komentāra dzēšana..... | 16 |
| 1.3.9. Raksta dzēšana..... | 17 |
| 1.3.10. Jaunu sacensību pievienošana un rediģēšana..... | 17 |
| 1.3.11. Pieteikšanās sacensībām | 19 |
| 1.3.12. Pieteikto dalībnieku saraksta apskatīšana | 20 |
| 1.3.13. Failu pievienošana..... | 20 |
| 1.3.14. Failu dzēšana..... | 21 |
| 1.4. Ārējās saskarnes prasības | 21 |
| 1.4.1. Lietotāja saskarne..... | 21 |
| 1.4.2. Programmatūras saskarne..... | 22 |
| 1.5. Nefunkcionālās prasības | 22 |
| 1.5.1. Drošība..... | 22 |
| 2. Programmatūras projektējuma apraksts | 23 |
| 2.1. Ievads | 23 |
| 2.1.1. Nolūks..... | 23 |
| 2.1.2. Darbības sfēra | 23 |

| | | |
|----------|--|----|
| 2.1.3. | Definīcijas..... | 23 |
| 2.2. | Saistība ar citiem dokumentiem | 24 |
| 2.3. | Dekompozīcijas apraksts | 24 |
| 2.3.1. | Moduļu dekompozīcija | 24 |
| 2.3.2. | Datu dekompozīcija..... | 24 |
| 2.4. | Detalizētais projektējums..... | 25 |
| 2.4.1. | Moduļu detalizēts projektējums | 25 |
| 2.4.1.1. | WELCOME kontrolieris..... | 25 |
| 2.4.1.2 | REGISTER kontrolieris | 26 |
| 2.4.1.3. | COMPETITIONS kontrolieris | 28 |
| 2.4.1.4. | PICTURE kontrolieris | 29 |
| 2.4.1.5. | ARTICLES kontrolieris..... | 29 |
| 2.4.1.6. | ADMIN kontrolieris | 32 |
| 2.4.2. | Datu detalizētais projektējums | 35 |
| 2.4.2.1. | Tabulas „Articles” detalizējums..... | 35 |
| 2.4.2.2. | Tabulas „ci_sessions” detalizējums | 36 |
| 2.4.2.3. | Tabulas „comments” detalizējums..... | 36 |
| 2.4.2.4. | Tabulas „competitions” detalizējums | 37 |
| 2.4.2.5. | Tabulas „picture” detalizējums | 37 |
| 2.4.2.6. | Tabulas „registered_users” detalizējums | 38 |
| 2.4.2.7. | Starptabulas „comp_users” detalizējums..... | 38 |
| 3. | Testēšanas dokumentācija..... | 39 |
| 3.1. | Testēšanas metodika | 39 |
| 3.2. | Funkciju testēšana | 39 |
| 3.2.1. | Lietotāja reģistrācija | 39 |
| 3.2.2. | Lietotāja datu apskatīšana un rediģēšana | 40 |
| 3.2.3. | Lietotāja statusa maiņa | 40 |
| 3.2.4. | Lietotāja dzēšana | 40 |
| 3.2.5. | Autorizācija | 41 |
| 3.2.6. | Raksta pievienošana..... | 42 |
| 3.2.7. | Komentāra pievienošana..... | 42 |
| 3.2.8. | Komentāra dzēšana..... | 42 |
| 3.2.9. | Raksta dzēšana..... | 42 |
| 3.2.10. | Jaunu sacensību pievienošana | 43 |
| 3.2.11. | Pieteikšanās sacensībām | 43 |
| 3.2.12. | Pieteikto dalībnieku saraksta apskatīšana..... | 44 |
| 3.2.13. | Failu pievienošana..... | 44 |

| | | |
|---------|--|----|
| 3.2.14. | Failu dzēšana..... | 45 |
| 4. | Programmatūras projekta organizācija | 46 |
| 5. | Kvalitātes nodrošināšana | 47 |
| 6. | Konfigurāciju pārvaldība | 48 |
| 7. | Darbietilpības novērtējums..... | 49 |
| 8. | Secinājumi..... | 50 |
| 9. | Izmantotā literatūra..... | 51 |
| 10. | Programmatūras pirmkoda paraugs..... | 52 |

Ievads

Taku orientēšanās sports ir samērā jauns Latvijā, tādēļ vēl nav izveidojušās pastāvīgas interneta vietnes, kurās būtu apkopota visa sportistiem nepieciešamā informācija. Lielākoties saziņa notiek, izmantojot e-pastus. Tā kā tas ir nepraktiski un apgrūtinoši gadījumos, kad informācijas ir tiešām daudz un tā vēl ir jāšķiro, tika izlemts par vienota portāla izveidi, kas atvieglotu ne tikai organizatoriskos jautājumus, bet arī uzlabotu savstarpējo komunikāciju.

Mērķis ir izstrādāt pagaidu sistēmu, ko laika gaitā varētu uzlabot un pilnveidot.

1. Programmatūras prasību specifikācija

1.1. Ievads

1.1.1. Dokumenta nolūks

Šī dokumenta nolūks ir viennozīmīgi un precīzi aprakstīt tīmekļa vietnes „Orientēšanās sacensību portāls” prasības.

Dokuments ir paredzēts sistēmas pasūtītājiem, lai specificētu viņu prasības, kā arī programmatūras izstrādātājiem, kas, ievērojot šo dokumentu, izstrādās sistēmu.

1.1.2. Darbības sfēra

Šajā dokumentā ir definētas „Orientēšanās sacensību portāla” prasības. Programmatūras produkta mērķis ir atvieglot Taku Orientēšanās sportistu saziņu, informācijas apmaiņu. Programmprodukts ir paredzēts, lai dotu iespēju sportistiem sazināties, uzzināt aktuālos jaunumus, kā arī pieteikties sacensībām (pēc tam arī apskatīt rezultātus) ērti lietojamā un viegli saprotamā interneta vietnē.

1.1.4. Saistība ar citiem dokumentiem

Programmatūras prasību specifikācija ir kvalifikācijas darba „Orientēšanās sacensību portāls” daļa. Uz šī dokumenta pamata tiks izstrādāts programmatūras projektējuma apraksts.

Dokuments izstrādāts saskaņā ar LVS 68:1996 *”Programmatūras prasību specifikācijas ceļvedis”*.

1.2. Neformāls apraksts

1.2.1. Produkta perspektīva

Orientēšanās sacensību portāls ir tīmekļa vietne, kas tiks izmantota informācijas apmaiņai starp sportistiem.

1.2.2. Produkta funkcijas

Orientēšanās sacensību portāls nodrošina šādas funkcijas:

- Jaunu lietotāju reģistrāciju – katrs portāla apmeklētājs var izveidot sev lietotāja kontu, kas nodrošina lietotājam plašākas iespējas portālā (funkcija USER.1);
- Lietotāju datu apskatīšanu un rediģēšanu – katrs lietotājs var apskatīt savus datus, kā arī mainīt daļu no tiem (funkcija USER.2);
- Jaunumu pievienošanu un dzēšanu – reģistrētiem lietotājiem ar attiecīgām privilēģijām ir iespēja pievienot portālam jaunus rakstus (funkcijas SITE.2, SITE.5);
- Komentāru pievienošanu un dzēšanu – ikviens portāla apmeklētājs var komentēt portālā esošos rakstus un bildes (funkcijas SITE.3, SITE.4);
- Sacensību informācijas pievienošanu, rediģēšanu un papildināšanu – reģistrētiem lietotājiem ar attiecīgām privilēģijām ir iespēja pievienot portālā informāciju par sacensībām, kā arī apskatīt pievienoto informāciju, labot to (funkcija COMP.1);
- Reģistrēšanos sacensībām – ikviens portāla apmeklētājs var pieteikties sacensībām, aizpildot pieteikšanās anketu (funkcija COMP.2);
- Failu pievienošanu un dzēšanu – portālā reģistrētie lietotāji var augšupielādēt failus, bildes, kā arī dzēst tos (funkcijas FILE.1, FILE.2).

1.2.3. Lietotāja raksturiezīmes

Lai izmantotu Orientēšanās sacensību portāla sniegtās iespējas, ikvienam tā apmeklētājam nepieciešamas pamatzināšanas darbā ar datoru un tīmekļa vietnēm. Vietnes apmeklētājam jābūt pieejamam interneta pieslēgumam, kā arī tīmekļa pārlūkprogrammai. Lietotāja profila izveidei apmeklētājam nepieciešama aktīva e-pasta kastīte.

Portāla administratoriem būtu vēlams pārzināt sistēmas uzbūvi.

1.2.4. Vispārējie ierobežojumi

Programma jāizstrādā PHP programmēšanas valodā, informācija jāglabā SQL datu bāzē, kurai jābūt *utf-8* kodējuma atbalstam.

1.3. Funkcionālās prasības

1.3.1. Lietotāja reģistrācija

Identifikators: USER.1

Funkcijas mērķis:

Orientēšanās sacensību portāla apmeklētāja datu saglabāšana portālam piesaistītajā datu bāzē, kā arī unikāla indentifikatora piešķiršana.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|------------------|----------|----------------------|---|
| Vārds | Jā | Teksta ievades lauks | |
| Uzvārds | Jā | Teksta ievades lauks | |
| Lietotājvārds | Jā | Teksta ievades lauks | Nav reģistrjūtīgs; Lietotājvārda garumam jābūt ne īsākam par simboliem un ne garākam par 20 |
| Parole | Jā | Teksta ievades lauks | Ievadītie simboli tiek aizstāti ar punktiņiem („•”). |
| Atkārtota parole | Jā | Teksta ievades lauks | Ievadītie simboli tiek aizstāti ar punktiņiem („•”); Jāsakrīt ar lauku „Parole”. |
| E-pasta adrese | Jā | Teksta ievades lauks | |
| Grupa | Jā | Nolaižamais saraksts | Iespējamie varianti: <ul style="list-style-type: none">• PE – paralimpiskā grupa;• AE – atklātā grupa; |
| Dzimšanas gads | Nē | Teksta ievades lauks | Vēlams norādīt, ja lietotājs jaunāks par 18gadiem; Ievadītajai vērtībai jābūt skaitliskai; |
| Klubs | Nē | Teksta ievades lauks | |

Ievaddatu apstrāde:

Funkcija pieejama ikvienam tīmekļa vietnes apmeklētājam. Tā paredz attiecīgas norādes obligātajiem laukiem, kā arī paskaidrojošo informāciju pie laukiem, kur tā varētu būt nepieciešama.

Funkcija paredz pārbaudi, vai, apstiprinot reģistrāciju, ir visi obligātie lauki aizpildīti korekti, kā arī tā pārbauda, vai sistēmā jau nav reģistrēts lietotājs ar apmeklētājam vēlamo lietotājvārdu. Ja šāds lietotājvārds jau ir reģistrēts datu bāzē, apmeklētājam tiek parādīts attiecīgs kļūdas paziņojums un dota

iespēja mēģināt pierēģistrēties vēlreiz. Ievadītā informācija visos laukos paliek apmeklētājam redzama, taču dati netiek saglabāti datu bāzē.

Ja kļūda netiek konstatēta, dati tiek saglabāti datu bāzē, lietotājam tiek piešķirts statuss – reģistrēts lietotājs. Funkcija paredz lietotāja paroles drošu saglabāšanu. Apmeklētājam tiek parādīts ziņojums par veiksmīgu reģistrāciju, kā arī lūgums to apstiprināt. Funkcija paredz automātisku reģistrācijas apstiprinājuma e-pasta nosūtīšanu uz jaunā lietotāja norādīto e-pasta adresi.

Funkcija paredz lietotāja konta aktivizēšanu brīdī, kad apmeklēta uz norādīto e-pasta adresi nosūtītā tīmekļa vietne.

Izvaddati:

Paziņojums par veiksmīgu/neveiksmīgu reģistrāciju.

1.3.2. Lietotāja datu apskatīšana un rediģēšana

Identifikators: USER.2

Funkcijas mērķis:

Reģistrēta lietotāja datu apskatīšana un ierobežota rediģēšana.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|-----------------|----------|----------------------|---|
| Vārds | Jā | Teksta ievades lauks | Ievades laukā redzams datu bāzē saglabātais lietotāja vārds; nav rediģējams |
| Uzvārds | Jā | Teksta ievades lauks | Ievades laukā redzams datu bāzē saglabātais lietotāja uzvārds; nav rediģējams |
| Lietotājvārds | Jā | Teksta ievades lauks | Ievades laukā redzams lietotājvārds, ar kādu lietotājs reģistrējies sistēmā; nav rediģējams |
| E-pasts | Jā | Teksta ievades lauks | Ievades laukā redzama datu bāzē saglabātā lietotāja e-pasta adrese |
| Grupa | Jā | Nolaižamais saraksts | Atlases laukā redzama pirmā saraksta izvēle „PE”, taču blakus norādīta datu bāzē saglabātā lietotāja grupa Iespējamās izvēles: PE, AE, B |
| Dzimšanas gads | Nē | Teksta ievades lauks | Ievades laukā redzama datu bāzē saglabātais lietotāja dzimšanas gads; nav rediģējams |

| | | | |
|-------|----|----------------------|---|
| Klubs | Nē | Teksta ievades lauks | Ievades laukā redzams datu bāzē saglabātais lietotāja klubs |
|-------|----|----------------------|---|

Ievaddatu apstrāde:

Funkcija paredzēta tīmekļa vietnē reģistrētiem lietotājiem. Tā paredz iespēju apskatīt visus autorizēta lietotāja datus (izņemot paroli), kā arī ļauj rediģēt šādus laukus – e-pasta adresi, grupu un klubu.

Datu rediģēšanas gadījumā, tiek pārbaudīts, vai ir aizpildīti visi obligāti aizpildāmie lauki, kā arī vai ievadītie dati ir korekti. Kļūdas gadījumā tiek uzrādīts attiecīgs paziņojums. Veiksmīgas datu maiņas gadījumā jaunie dati jā saglabā datu bāzē veco vietā. Lietotājam tiek parādīts paziņojums par veiksmīgu datu labošanu, kā arī labotā profila informācija.

Izvaddati:

Autentificētā lietotāja dati.

Paziņojums par veiksmīgu/neveiksmīgu datu rediģēšanu.

1.3.3. Lietotāja statusa maiņa

Identifikators: USER.3

Funkcijas mērķis:

Reģistrēta lietotāja statusa maiņa.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|-----------------|----------|----------------------|--|
| Vārds | Jā | Teksta ievades lauks | Ievades laukā redzams datu bāzē saglabātais lietotāja vārds; nav rediģējams. |
| Uzvārds | Jā | Teksta ievades lauks | Ievades laukā redzams datu bāzē saglabātais lietotāja uzvārds; nav rediģējams. |
| E-pasts | Jā | Teksta ievades lauks | Ievades laukā redzama datu bāzē saglabātā lietotāja e-pasta adrese; nav rediģējams. |
| Statuss | Jā | Nolaižamais saraksts | Atlases laukā redzama datu bāzē saglabātais lietotāja statuss; Iespējamās izvēles: <ul style="list-style-type: none"> • Parasts lietotājs; • Funkcionārs; |

| | | | |
|----------------|----|----------------------|---|
| Grupa | Jā | Nolaižamais saraksts | Atlases laukā redzama datu bāzē saglabātā lietotāja grupa; nav rediģējams |
| Klubs | Nē | Teksta ievades lauks | Ievades laukā redzams datu bāzē saglabātais lietotāja klubs; nav rediģējams |
| Dzimšanas gads | Nē | Teksta ievades lauks | Ievades laukā redzams datu bāzē saglabātais lietotāja dzimšanas gads; nav rediģējams |

Ievaddatu apstrāde:

Funkcija paredzēta tīmekļa vietnes administratoram. Tā parāda visus portālā reģistrētos lietotājus, paredz iespēju mainīt jebkura reģistrēta lietotāja statusu, tādējādi izmainot lietotāja piekļuves tiesības attiecīgām tīmekļa vietnes sadaļām.

Tīmekļa vietnes administrators var mainīt statusa esošo vērtību uz kādu no citām divām atlikušajām iespējām. Jaunais lietotāja statuss tiek saglabāts attiecīgā lietotāja datus datu bāzē.

Funkcija paredz e-pasta nosūtīšanu lietotājam, kura statuss ticis nomainīts.

Izvaddati:

Nav.

1.3.4. Lietotāja dzēšana

Identifikators: USER.4

Funkcijas mērķis:

Reģistrēta lietotāja dzēšana.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|------------------|----------|----------------------|----------|
| Dzēst lietotāju! | Jā | Poga | |
| Iemesls | Nē | Teksta ievades lauks | |
| Dzēst | Nē | Poga | |
| Atcelt | Nē | Poga | |

Ievaddatu apstrāde:

Funkcija paredzēta gan tīmekļa vietnes administratoram, gan reģistrētajiem lietotājiem. Tā paredz iespēju dzēst lietotāju no datu bāzes. Tīmekļa vietnes administrators var izdzēst jebkuru tīmekļa vietnē reģistrētu lietotāju. Reģistrēts lietotājs var dzēst savus datus no datu bāzes.

Pirms lietotāja datu dzēšanas no datu bāzes, ja to veic administrators, tiek dota iespēja ievadīt dzēšanas iemeslu. Tiek pieprasīts atkārtots apstiprinājums lietotāja dzēšanai. Ja šāds apstiprinājums tiek saņemts, uz attiecīgā lietotāja norādīto e-pastu tiek nosūtīta ziņa, ka lietotājs ticis dzēsts no portāla, kā arī, ja administrators uzrādījis, iemesls, kāpēc lietotājs dzēsts. Lietotāja personas dati datu bāzē tiek aizvietoti ar nederīgu informāciju: vārds „dzēsts”, uzvārds „lietotājs”, e-pasts „”, grupa „”, dzimšanas gads „”, klubs „”, lai nebūtu nepieciešamība dzēst lietotāja pievienotos rakstus / komentārus / sacensības / failus.

Ja apstiprinājums dzēšanai netiek saņemts, dati netiek dzēsti.

Tīmekļa vietnē reģistrēts lietotājs var pats izdzēst savus datus no datu bāzes. Tiek prasīts atkārtots apstiprinājums lietotāja dzēšanai. Ja šāds apstiprinājums tiek saņemts, uz attiecīgā lietotāja norādīto e-pastu tiek nosūtīta ziņa, ka lietotājs ticis dzēsts no portāla. Lietotāja dati datu bāzē tiek aizvietoti ar nederīgu informāciju, tāpat kā gadījumā, ja dzēšanu veic administrators. Lietotājam tiek parādīta tīmekļa vietne kā neautenticētam portāla apmeklētājam.

Izvaddati:

Nav.

1.3.5. Autorizācija

Identifikators: SITE.1

Funkcijas mērķis:

Reģistrēta lietotāja autorizācija tīmekļa vietnē. Paredzēta, lai identificētu portāla lietotāju.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|-----------------|----------|----------------------|---|
| Lietotājevārds | Jā | Teksta ievades lauks | Nav reģistrjūtīgs |
| Parole | Jā | Teksta ievades lauks | Lietotājam redzama kā punktiņu („•”) rinda; |

Ievaddatu apstrāde:

Funkcija paredzēta tīmekļa vietnes apmeklētājiem, kuri ir reģistrējušies portālā. Tā paredz autorizēšanos sistēmā.

Tīmekļa vietnes apmeklētājam attiecīgajos laukos jāievada lietotājevārds un parole. Funkcija pārbauda, vai lietotājs ar ievadīto lietotājevārdu eksistē, kā arī, ja eksistē, pārbauda, vai ievadītā parole ir vienāda ar to, kas saglabāta datu bāzē. Ja kāds no šiem nosacījumiem neizpildās, funkcija atgriež paziņojumu, ka autentifikācija nav bijusi veiksmīga.

Ja ievadītais lietotājevārds un parole ir korekti, notiek autentificēšanās sistēmā. Pēc veiksmīgas autentificēšanās tiek parādīta tīmekļa vietnes sāukmlapa ar attiecīgajām izvēlnēm, atkarībā no lietotājam

piešķirtā statusa.

Izvaddati:

Paziņojums par neveiksmīgu autorizācijas mēģinājumu.

1.3.6. Raksta pievienošana

Identifikators: SITE.2

Funkcijas mērķis:

Pievienot jaunu rakstu.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|-------------------------------------|----------|----------------------|--|
| Raksta nosaukums | Jā | Teksta ievades lauks | |
| Sadaļa | Jā | Nolaižamais saraksts | Iespējamie varianti: <ul style="list-style-type: none">• Ziņas;• Interesanti; |
| Neliels apraksts vai ievads rakstam | Jā | Teksta ievades lauks | Raksta daļa, kas redzama pirms raksta atvēršanas |
| Raksts | Jā | Teksta ievades lauks | |
| Tips | Jā | Nolaižamais saraksts | Nosaka raksta pieejamību; Iespējamie varianti: <ul style="list-style-type: none">• Saglabāt kā melnrakstu,• Publicēt visiem, |

Ievaddatu apstrāde:

Funkcija paredzēta autentificētiem tīmekļa vietnes apmeklētājiem. Tā paredz iespēju publicēt portālā rakstus.

Funkcija pārbauda, vai visi obligātie lauki ir aizpildīti; ja tā nav, tiek izvadīts attiecīgs kļūdas paziņojums. Ja dati ievadīti korekti, saglabā rakstu datu bāzē un pievieno tam saglabāšanas datumu un laiku, kā arī fiksē raksta autoru. Pēc veiksmīgas raksta saglabāšanas, lietotājam tiek parādīts attiecīgs ziņojums.

Izvaddati:

Paziņojums par veiksmīgu raksta saglabāšanu.

1.3.7. Komentāra pievienošana

Identifikators: SITE.3

Funkcijas mērķis:

Pievienot jaunu komentāru.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|-----------------|----------|----------------------|--|
| Autors | Jā | Teksta ievades lauks | Ja lietotājs ir autorizējies sistēmā, teksta ievades laukā redzams lietotāja |
| Komentārs | Jā | Teksta ievades lauks | |

Ievaddatu apstrāde:

Funkcija paredzēta ikvienam portāla apmeklētājam. Tā paredz iespēju komentēt portālā esošos rakstus, bildes.

Funkcija pārbauda, vai visi obligātie lauki ir aizpildīti, ja nav izvada attiecīgu paziņojumu.

Funkcija saglabā komentāru datu bāzē, pievieno tam saglabāšanas datumu un laiku, kā arī piesaista komentāru rakstam / bildei. Ja komentāra pievienošana bijusi veiksmīga, tiek parādīts attiecīgs paziņojums.

Izvaddati:

Paziņojums par kļūdu.

Paziņojums par veiksmīgu komentāra saglabāšanu.

1.3.8. Komentāra dzēšana

Identifikators: SITE.4

Funkcijas mērķis:

Dzēst komentāru.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|------------------|----------|---------------|----------|
| Dzēst komentāru! | Jā | Poga | |

Ievaddatu apstrāde:

Funkcija paredzēta tīmekļa vietnes administratoram. Tā paredz iespēju dzēst pievienotos komentārus. Pirms komentāra dzēšanas netiek pieprasīts atkārtots apstiprinājums. Pēc komentāra dzēšanas, tiek parādīts attiecīgs ziņojums.

Izvaddati:

Paziņojums par veiksmīgu komentāra dzēšanu.

1.3.9. Raksta dzēšana

Identifikators: SITE.5

Funkcijas mērķis:

Dzēst rakstu, kopā ar komentāriem (ja tādi rakstam ir).

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|-----------------|----------|---------------|----------|
| Dzēst rakstu | Jā | Poga | |
| Dzēst | Nē | Poga | |
| Atcelt | Nē | Poga | |

Ievaddatu apstrāde:

Funkcija paredzēta gan tīmekļa vietnes administratoram, gan reģistrētajiem lietotājiem ar funkcionāra statusu. Tā paredz iespēju dzēst rakstus no datu bāzes.

Pirms raksta dzēšanas no datu bāzes, tiek prasīts atkārtots apstiprinājums dzēšanai. Ja šāds apstiprinājums tiek saņemts, funkcija pārbauda, vai konkrētajam rakstam nav pievienoti komentāri. Ja komentāri ir, tie tiek izdzēsti. Pēc tam tiek dzēsts arī raksts, lietotājam tiek parādīts ziņojums par veiksmīgu raksta dzēšanu.

Ja apstiprinājums netiek saņemts, funkcija no jauna ielādē attiecīgo portāla sadaļu.

Izvaddati:

Paziņojums par veiksmīgu raksta dzēšanu.

1.3.10. Jaunu sacensību pievienošana un rediģēšana

Identifikators: COMP.1

Funkcijas mērķis:

Pievienot portālam jaunas sacensības ar visu tām nepieciešamo informāciju, kā arī šīs informācijas rediģēšanu.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|-----------------|----------|---------------|----------|
|-----------------|----------|---------------|----------|

| | | | |
|--------------------------------------|----|----------------------|--|
| Sacensību nosaukums | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |
| Laiks | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |
| Vieta | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |
| Sacensību programma | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |
| Dalībnieku grupas | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |
| Starta kārtība un distanču parametri | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |
| Rezultāti | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |
| Kartes | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |
| Pieteikumi | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |
| Sacensības rīko | Jā | Teksta ievades lauks | Sacensību rediģēšanas gadījumā, teksta ievades laukā redzama datu bāzē saglabātā informācija |

Ievaddatu apstrāde:

Funkcija paredzēta reģistrētiem lietotājiem ar funkcionāra statusu. Tā paredz iespēju pievienot jaunas sacensības ar visu tām nepieciešamo informāciju.

Funkcija paredz pārbaudi, vai, pievienojot jaunas / rediģējot jau pievienotas sacensības, visi obligātie lauki ir aizpildīti. Ja kāds no laukiem nav aizpildīts, funkcionāram tiek uzrādīts attiecīgs kļūdas paziņojums. Ievadītā informācija visos laukos paliek apmeklētājam redzama, taču dati netiek saglabāti datu bāzē. Ja kļūda netiek konstatēta, dati tiek saglabāti datu bāzē.

Tiek izvadīts paziņojums par veiksmīgu sacensību pievienošanu.

Izvaddati:

Paziņojums par nekorekti aizpildītiem laukiem.

Paziņojums par veiksmīgu sacensību pievienošanu.

1.3.11. Pieteikšanās sacensībām

Identifikators: COMP.2

Funkcijas mērķis:

Nodrošināt lietotājiem iespēju pieteikties sacensībām.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|-----------------|----------|----------------------|---|
| Vārds | Jā | Teksta ievades lauks | Ja lietotājs autorizējies sistēmā, jau aizpildīts |
| Uzvārds | Jā | Teksta ievades lauks | Ja lietotājs autorizējies sistēmā, jau aizpildīts |
| Klubs | Nē | Teksta ievades lauks | Ja lietotājs autorizējies sistēmā, jau aizpildīts |
| Dzimšanas gads | Nē | Teksta ievades lauks | Vēlams norādīt, ja dalībnieks jaunāks par 18 gadiem; Ja lietotājs autorizējies sistēmā un ir |
| Grupa | Jā | Nolaižamais saraksts | Iespējamie varianti: <ul style="list-style-type: none">• PE• AE |
| Stūmējs | Nē | Izvēles rūtiņa | Opcija pieejama tikai tad, ja dalībnieks startē paralimpiskajā (PE) grupā |

Ievaddatu apstrāde:

Funkcija paredzēta ikvienam lapas apmeklētājam. Tā paredz iespēju pieteikties sacensībām, izmantojot tīmekļa vietnē piejamo pieteikšanās anketu.

Ja lapas apmeklētājs ir autorizējies sistēmā, anketa tiek aizpildīta ar datu bāzē saglabātajiem lietotāja datiem.

Ja lapas apmeklētājs nav autentificējies sistēmā, anketa jāaizpilda pašam.

Funkcija pārbauda, vai formā ir aizpildīti visi obligātie lauki. Tiek prasīta pieteikšanās apstiprināšana. Ja tāda tiek saņemta, un ievades dati ir korekti, tīmekļa vietnes apmeklētājs tiek reģistrēts kā attiecīgo sacensību dalībnieks, tiek izvadīts attiecīgs ziņojums par veiksmīgu pieteikšanos. Ja ievadītie dati nav korekti, tiek izvadīts attiecīgs kļūdas paziņojums. Ievadītā informācija visos laukos paliek apmeklētājam redzama, taču dati netiek saglabāti datu bāzē.

Izvaddati:

Kļūdas paziņojums par nekorektiem ievades.

Paziņojums par veiksmīgu pieteikšanos sacensībām.

1.3.12. Pieteikto dalībnieku saraksta apskatīšana

Identifikators: COMP.3

Funkcijas mērķis:

Ļaut tīmekļa vietnes apmeklētājam apskatīt konkrētajām sacensībām pieteikušos dalībniekus.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|---------------------------------|----------|---------------|----------|
| Apskatīt pieteiktos dalībniekus | Jā | Poga | |

Ievaddatu apstrāde:

Funkcija paredzēta ikvienam portāla apmeklētājam. Tā paredz iespēju jebkurām sacensībām apskatīties pieteikto dalībnieku sarakstu.

Ja funkciju izsauc autentificējies lietotājs, kurš ir pieteicies konkrētajām sacensībām, funkcija izceļ lietotāju dalībnieku sarakstā. Ja dalībnieks nav pieteicies konkrētajām sacensībām, dalībnieku sarakstu iespējams apskatīt tāpat kā neautentificētam lapas apmeklētājam – bez izceltiem dalībniekiem. Funkcija nodrošina dalībnieku saraksta parādīšanu.

Izvaddati:

Sacensību dalībnieku saraksts.

1.3.13. Failu pievienošana

Identifikators: FILE.1

Funkcijas mērķis:

Ļaut failu augšupielādi tīmekļa vietnē.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|---------------------|----------|----------------------|---|
| Faili | Jā | Faila izvēles poga | Lietotājs izvēlas, kuru failu augšupielādēt |
| Galerijas nosaukums | Jā | Teksta ievades lauks | |

Ievaddatu apstrāde:

Funkcija paredzēta reģistrētiem tīmekļa vietnes lietotājiem. Tā paredz iespēju augšupielādēt failus. Funkcija pārbauda, vai aizpildīts galerijas nosaukuma lauks un norādīts augšupielādējamais fails. Ja ir, fails

tiek augšupielādēts. Ja nav, tiek parādīts attiecīgs paziņojums.

Ja augšupielāde kāda iemesla dēļ bijusi neveiksmīga, tiek parādīts kļūdas paziņojums. Ja augšupielāde bijusi veiksmīga, tiek parādīts attiecīgs paziņojums, informācija par failu un lietotāju tiek saglabāta datu bāzē.

Izvaddati:

Paziņojums par kļūdu, veicot faila augšupielādi.

Paziņojums par veiksmīgu faila augšupielādi.

1.3.14. Failu dzēšana

Identifikators: FILE.2

Funkcijas mērķis:

Dzēst augšupielādētos failus.

Ievaddati:

| Lauka nosaukums | Obligāts | Ievades forma | Piezīmes |
|-----------------|----------|---------------|----------|
| Dzēst failu! | Jā | Poga | |

Ievaddatu apstrāde:

Funkcija paredzēta gan tīmekļa vietnes administratoram, gan reģistrētajiem lietotājiem. Tā paredz iespēju dzēst tīmekļa vietnē augšupielādētos failus.

Tīmekļa vietnes administrators var izdzēst jebkuru tīmekļa vietnē pievienoto failu. Tīmekļa vietnē reģistrēts lietotājs var izdzēst sevis augšupielādētos failus. Pirms faila dzēšanas no datu bāzes, neatkarīgi no lietotāja statusa, tiek prasīts atkārtots apstiprinājums dzēšanai. Ja šāds apstiprinājums tiek saņemts, fails tiek dzēsts no servera, kā arī informācija par failu – no datu bāzes, administratoram/lietotājam tiek parādīts ziņojums par veiksmīgu faila dzēšanu.

Ja apstiprinājums dzēšanai netiek saņemts, fails netiek dzēsts.

Izvaddati:

Paziņojums par veiksmīgu faila dzēšanu.

1.4. Ārējās saskarnes prasības

1.4.1. Lietotāja saskarne

Lietotāja saskarnei jābūt vienkāršai un intuitīvai.

Lietotāja saskarnei var piekļūt, izmantojot interneta pārlūkprogrammu. Tā tiek veidota kā HTML

lietojumprogramma. Sistēmā ir nodrošinātas 3 lietotāju grupas – parastais lietotājs, funkcionārs un administrators. Katrai lietotāju grupai ir tai pielāgota saskarne ar tās statusam atbilstošajām funkcijām.

Portāla saturu var apskatīt arī neautorizējies portāla apmeklētājs, taču ar ierobežotām iespējām.

Dati galvenokārt tiek attēloti tabulu, sarakstu vai failu formā. Funkcionālo taustiņu nav.

Lietotāja saskarnei vizuāli jābūt iedalītai 5 blokos – augšējā, apakšējā, labajā, kreisajā un galvenajā. Augšējā daļa sastāv no izvēlnes, apakšējā daļa satur kontaktinformāciju, kreisais bloks ietver sevī apakšizvēlnes, savukārt labais – autorizēta lietotāja informāciju vai autorizācijas iespēju. Galvenajā blokā tiek parādīts attiecīgās portāla sadaļas saturs.

1.4.2. Programmatūras saskarne

Tiek izmantots tīmekļa serveris. Sistēmas pieeja datu bāzei tiek nodrošināta ar SQL servera palīdzību.

1.5. Nefunkcionālās prasības

1.5.1. Drošība

Jābūt nodrošinātai vienkāršai autentifikācijai un autorizācijai dažādiem lietotāju līmeņiem, lai ierobežotu lietotāju piekļuvi atsevišķām portāla sadaļām. Jābūt nodrošinātai parolu šifrēšanai.

Lietotāju datiem drīkst piekļūt tikai paši lietotāji, kā arī administrators.

Jābūt elementārai aizsardzībai pret SQL un XSS injekcijām.

2. Programmatūras projektējuma apraksts

2.1. Ievads

2.1.1. Nolūks

Orientēšanās sacensību portāla programmas projektējuma apraksta nolūks ir nodrošināt veiksmīgu sistēmas izstrādi atbilstoši programmatūras prasību specifikācijā izvirzītajām prasībām. Dokuments paredzēts sistēmas izstrādātājiem, lai atvieglotu programmatūras projektēšanu un izstrādi.

2.1.2. Darbības sfēra

Orientēšanās sacensību portāla sistēma paredzēta, lai atvieglotu saziņu starp sportistiem, izveidotu vienotu vietu, kurā noris diskusijas un plānošana, kā arī ir iespējams ērti pieteikties sacensībām, publicēt bildes.

Ar sistēmu strādās 3 tipu lietotāji:

- parasts lietotājs – var apskatīt portālā esošo informāciju (izņemot slēgtās diskusijas), kā arī pievienot rakstus, bildes, komentārus, pieteikties sacensībām;
- funkcionārs – var apskatīt visu portālā esošo informāciju (arī slēgtās diskusijas), kā arī pievienot sacensības, rakstus, bildes, komentārus, pieteikties sacensībām;
- administrators – pārvalda sistēmā pierēģistrētos lietotājus, kā arī portālam pievienoto informāciju – rakstus, bildes, komentārus;

kā arī sistēma pieejama brīvai lietošanai – portāla apmeklētājam nav jāautorizējas, lai apskatītu portālā publicētos rakstus, bildes, kā arī pievienotu komentārus.

2.1.3. Definīcijas

DB – datu bāze

ER modelis– *Entity Relationship* modelis; attēlo datubāzes entītiju savstarpējās atkarības

PPA – programmas projektējuma apraksts

2.2. Saistība ar citiem dokumentiem

Šis programmatūras projektējuma apraksts ir izstrādāts saskaņā ar Orientēšanās sacensību portāla programmatūras prasību specifikāciju, kā arī dokumenta izstrādē ņemti vērā Latvijas standarta 72:1996 ieteikumi.

2.3. Dekompozīcijas apraksts

Sistēma tiks izstrādāta ar *Model – View – Controller (MVC) Design Pattern* pieeju, kas nodrošinās programmas koda sadalīšanu 3 slāņos:

1. *Controller* (kontrolieris) – nodrošina darbības ar skatiem un modeļiem;
2. *Model* (modelis) – nodrošina darbības, kuras saistītas ar datu iegūšanu un glabāšanu datu bāzē;
3. *View* (skats) – nodrošina sistēmas informācijas parādīšanu portāla apmeklētājam.

2.3.1. Moduļu dekompozīcija

Sistēmā ir 6 galvenie moduļi, katrs modulis ir kontrolieris sistēmā:

1. modulis *WELCOME* – atver portāla galveno sadaļu, nodrošina lietotāju pieteikšanos / atteikšanos sistēmā;
2. modulis *REGISTER* – nodrošina jaunu lietotāju reģistrēšanos sistēmai, lietotāja profila informācijas apskatīšanu un rediģēšanu, lietotāja konta dzēšanu;
3. modulis *COMPETITIONS* – nodrošina ar sacensībām saistītās informācijas pievienošanu un rediģēšanu, kā arī ļauj lietotājam pieteikties sacensībām un apskatīt sacensībām pieteikto dalībnieku sarakstus;
4. modulis *PICTURE* – nodrošina failu augšupielādi / dzēšanu;
5. modulis *ARTICLES* – nodrošina rakstu publicēšanu un komentāru pievienošanu;
6. modulis *ADMIN* – nodrošina administratīvo funkciju izpildi portālā – lietotāju statusa maiņu, lietotāju kontu dzēšanu, komentāru un rakstu dzēšanu.

2.3.2. Datu dekompozīcija

Sistēma glabā datus vienā MSSQL datu bāzē. Tā sastāv no 6 tabulām un 1 starptabulas:

1. tabula „*Articles*” – glabā informāciju par portālā publicētajiem rakstiem;
2. tabula „*ci_sessions*” – glabā aktīvās sesijas datus;

3. tabula „*comments*” – glabā portālā pievienotos komentārus;
4. tabula „*competitions*” – glabā portālā pievienotās sacensības;
5. tabula „*picture*” – glabā portālā pievienoto bilžu informāciju;
6. tabula „*registered_users*” – glabā informāciju par portālā reģistrētajiem lietotājiem;
1. starptabula „*comp_users*” – glabā informāciju par sacensībām pieteiktajiem dalībniekiem.

2.4. Detalizētais projektējums

2.4.1. Moduļu detalizēts projektējums

2.4.1.1. WELCOME kontrolieris

WELCOME kontrolieris sevī ietver šādas funkcijas:

- galvenās portāla sadaļas atvēršana:

| | |
|------------------|--|
| Pieejas tiesības | Ikviens portāla apmeklētājs |
| Procesi | <ul style="list-style-type: none"> • Atverot portālu, tiek ielādēti galvenie portālu bloku veidojošie skati • Tiek noskaidrots, vai sistēmā ir kāds autorizējies lietotājs |

- Autorizēšanos sistēmā:

| | |
|------------------|---|
| Pieejas tiesības | Ikviens portāla apmeklētājs |
| Procesi | <ul style="list-style-type: none"> • Ievadītais lietotājvārds un parole tiek nosūtīti autentifikācijas modelim; • Autentifikācijas modulis no datu bāzes tabulas „<i>registered_users</i>” nolasa informāciju par lietotāju; • Lietotāja ievadītajai parolei tiek pievienots no lietotāja datiem nolasītā <i>salt</i> vērtība, šādā formā tiek šifrēta izmantojot <i>SHA1</i> šifrēšanas algoritmu; • Nošifrētā parole tiek salīdzināta ar to, kura glabājas lietotāja datos datu bāzē; • Ja paroles sakrīt, lietotājs ir autorizējies sistēmā: <ul style="list-style-type: none"> modelis kontrolierim atgriež lietotāja statusu; kontrolieris saglabā pašreizējās sesijas datus – autorizētā lietotāja identifikatoru, lietotājvārdu, statusu; kontrolieris no jauna ielādē galveno portāla sadaļu; • Ja paroles nesakrīt, autentifikācija nav izdevusies: <ul style="list-style-type: none"> modelis kontrolierim atgriež nekorektu lietotāja statusu; |

| | |
|--|--|
| | kontrolieris paziņo par neveiksmīgu autentifikāciju. |
|--|--|

- Atteikšanos no sistēmas;

| | |
|------------------|---|
| Pieejas tiesības | Autorizējies lietotājs |
| Procesi | <ul style="list-style-type: none"> • Kontrolieris pārtrauc aktīvo sesiju; • Kontrolieris ielādē lapas galveno sadaļu. |

2.4.1.2 REGISTER kontrolieris

REGISTER kontrolieris sevī ietver šādas funkcijas:

- Jaunu lietotāju reģistrāciju:

| Pieejas tiesības | Neautorizējies lietotājs |
|------------------|--|
| Apstrāde | <ul style="list-style-type: none"> • Kontrolieris pārbauda, vai ir kāda aktīva sesija; ja ir – no jauna ielādē portāla galveno sadaļu; • Kontrolieris ielādē skatus, kas parāda portāla navigācijas iespējas; • Kontrolieris ielādē reģistrācijas formu; • Nosūtot formas datus tiek pārbaudīts ievaddatu korektums – vai izvēlētais lietotājvārdā esošo simbolu skaits ir no 5 līdz 20 simboliem, vai ievadītās paroles sakrīt, vai ievadīta derīga e-pasta adrese, vai ievadītais dzimšanas gads ir skaitlis; • Ja ir nekorekti aizpildīti ievades lauki, pie katra nekorekti aizpildītā lauka <i>Codeigniter</i> iebūvēta validācijas funkcija pēc definētajiem noteikumiem, izvada attiecīgu kļūdas paziņojumu; • Ja visi ievadītie dati ir korekti, parole tiek šifrēta (izmantojot <i>SHA1</i> algoritmu), kontrolieris iegūst skatā ievadītās vērtības (ar „<i>POST</i>” metodes palīdzību); • Kontrolieris ielādē reģistrācijas moduli, kuram kā atribūtus padod no skata iegūtās vērtības; • Reģistrācijas modulis pārbauda, vai izvēlētais lietotājvārds jau nav ticis izmantots; • Ja lietotājvārds ir izmantots, reģistrācijas modulis reģistrācijas kontrolierim atgriež attiecīgu paziņojumu; formas ievades laukos ievadītā informācija ir saglabājusies (tiek izmantota <i>set_value</i> funkcija); • Ja izvēlētais lietotājvārds ir unikāls, reģistrācijas modulī tiek nejauši izvēlēta 6 |

| | |
|--|--|
| | <p>ciparu skaitliska vērtība;</p> <ul style="list-style-type: none"> • Vērtība tiek pievienota šifrētās paroles beigās, iegūtā simbolu virkne tiek atkārtoti šifrēta, izmantojot <i>SHA1</i> algoritmu; • Dati tiek salikti masīvā un saglabāti datu bāzes „<i>registered_users</i>” tabulā; • Tiek izsaukta reģistrācijas moduļa funkcija „<i>sendmail</i>”; • Izmantojot <i>Codeigniter</i> e-pasta bibliotēku, uz norādīto e-pasta adresi tiek nosūtīts e-pasts, ar kura palīdzību tiek aktivizēts lietotāja konts (reģistrācijas moduļa validācijas funkcija): <ul style="list-style-type: none"> no datu bāzes tiek tiek nolasīti lietotāja dati; tie tiek salīdzināti ar saitē esošo informāciju; ja dati sakrīt, datu bāzē tiek aktivizēts lietotāja konts (vērtība „<i>validated</i>” tiek uzstādīta uz 1); tiek izsaukta reģistrācijas kontroliera apstiprināšanas („<i>confirm</i>”) funkcija, kura lietotājam parāda lietotāja konta validācijas rezultātus; • Kontrolieris ielādē portāla labā un apakšējā bloka skatus. |
|--|--|

- Lietotāja profila informācijas apskatīšanu un rediģēšanu:

| Pieejas tiesības | Autorizējies lietotājs |
|------------------|---|
| Procesi | <ul style="list-style-type: none"> • Kontrolieris pārbauda, vai sistēmas lietotājs ir autorizējies; ja nav, apmeklētājam tiek ielādēta galvenā portāla sadaļa; • Izmantojot sesijas datus, tiek noskaidrots lietotāja identifikators; • Izmantojot lietotāja datu modeli „<i>user_model</i>” no datu bāzes tabulas „<i>registered_users</i>” tiek iegūta attiecīgā lietotāja profila informācija; • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris ielādē skatu, kurā iespējams mainīt lietotāja datus; • <i>Form_validation</i> bibliotēka nodrošina ievadīto datu validācijas pārbaudi, kā arī kļūdu ziņojumu parādīšanu; • Kad ievadīti korekti dati, kontrolieris ielādē lietotāja datu moduļa „<i>change_info</i>” funkciju, kura nomaina datu bāzē glabāto informāciju pret jaunajiem datiem; • Ja dati nomainīti bez kļūdām, kontrolieris ielādē lietotāja profila informācijas skatu „<i>userdata</i>”; • Kontrolieris ielādē portāla labā un apakšējā bloka skatus. |

- Lietotāja konta dzēšanu:

| Pieejas tiesības | Autorizējies lietotājs |
|------------------|---|
| Procesi | <ul style="list-style-type: none"> • Kontrolierī tiek izsaukta pārbaudes funkcija; • Kontrolieris pārbauda, vai sistēmas lietotājs ir autorizējies; ja nav, apmeklētājam tiek ielādēta galvenā portāla sadaļa; • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris ielādē skatu, kas pieprasa apstiprināt lietotāja konta dzēšanu; • Kontrolieris ielādē portāla labā un apakšējā bloka skatus; • Ja lietotāja dzēšana tiek apstiprināta (poga „Dzēst!”), kontrolieris izsauc lietotāja datu modeļa funkciju „delete_user”, kurai kā atribūts tiek padots autorizētā lietotāja identifikators; • Funkcija izveido masīvu ar dzēsta lietotāja datiem (vārds „dzēsts”, uzvārds „lietotājs”, lietotājavārds „”, e-pasts „”, grupa „”, klubs „”, dzimšanas gads „0” un lietotāja statuss „0”), kuri tiek saglabāti datu bāzes „registered_users” tabulā iepriekšējo datu vietā; • Kontrolieris ielādē Welcome kontroliera „logout” funkciju; • Ja lietotāja dzēšana neitiek apstiprināta (poga „Atcelt!”), tiek no jauna izsaukta reģistrācijas kontroliera lietotāja datu apskatīšanas un rediģēšanas funkcija. |

2.4.1.3. COMPETITIONS kontrolieris

COMPETITIONS kontrolieris sevī ietver šādas funkcijas:

- Sacensību informācijas parādīšanu:

| Pieejas tiesības | Ikviens portāla apmeklētājs |
|------------------|--|
| Procesi | <ul style="list-style-type: none"> • Kontrolieris ielādē sacensību informācijas modeļa („view_comp_model”) funkciju „get_comp”; • „Get_comp” funkcija no gatu bāzes tabulas „competitions” iegūst informāciju par reģistrētajām sacensībām, kā masīvu atgriež kontrolierim; • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris ielādē skatu, kas parāda informāciju par sacensībām, kā atribūtus padodot no sacensību informācijas modeļa iegūto masīvu; |

| | |
|--|---|
| | <ul style="list-style-type: none"> • Kontrolieris ielādē portāla labā un apakšējā bloka skatus; • Sacensību skats iegūtos datus attēlo tabulas veidā. |
|--|---|

- Sacensību informācijas pievienošanu un rediģēšanu:

| Pieejas tiesības | Funkcionāri |
|------------------|--|
| Procesi | <ul style="list-style-type: none"> • Kontrolieris no aktīvās sesijas datiem nolasa autorizētā lietotāja statusu – ja tas neatbilst funkcionāram, kontrolieris ielādē funkciju, kura atbild par sacensību informācijas parādīšanu; • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris ielādē skatu, kurā ir sacensību pievienošanas forma; • Kontrolieris ielādē portāla labā un apakšējā bloka skatus; • <i>Form_validation</i> bibliotēka nodrošina sacensību reģistrācijas formā ievadīto datu validācijas pārbaudi, kā arī kļūdu ziņojumu parādīšanu; • Sacensību informācijas pievienošanas forma ar korektiem ievades datiem izsauc kontroliera funkciju „<i>submit</i>”, kura atbild par sacensību informācijas saglabāšanu. • Kontrolieris iegūst no skata informāciju par sacensībām (izmantojot „<i>POST</i>” metodes palīdzību); • Kontrolieris ielādē sacensību pievienošanas moduļa („<i>add_comp_model</i>”) funkciju „<i>register_comp</i>” un kā atribūtus padod iegūto sacensību informāciju; • Funkcija „<i>register_comp</i>” no padotajiem parametriem izveido masīvu, kura datus saglabā datu bāzes tabulā „<i>competitions</i>”; • Kontrolieris izsauc funkciju „<i>register</i>”, kas ielādē portāla saturu, galvenajā daļā parādot paziņojumu par veiksmīgu sacensību reģistrāciju. |

2.4.1.4. PICTURE kontrolieris

PICTURE kontrolieris sevī ietver šādas funkcijas:

- Failu augšupielādi:
- Failu dzēšanu:

2.4.1.5. ARTICLES kontrolieris

ARTICLES kontrolieris sevī ietver šādas funkcijas:

- Visu rakstu saraksta parādīšanu:

| Pieejas tiesības | Ikviens portāla apmeklētājs |
|------------------|--|
| Procesi | <ul style="list-style-type: none"> • Kontrolieris ielādē rakstu skatīšanas modeļa funkciju „<i>get_articles</i>”; • Funkcija „<i>get_articles</i>” no datu bāzes tabulas „<i>articles</i>” iegūst informāciju par datu bāzē saglabātajiem rakstiem, datus kā masīvu atgriež kontrolierim; • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris ielādē skatu, kurš nodrošina rakstu parādīšanu, kā atribūtus padodot no rakstu skatīšanās modeļa iegūtos datus; • Kontrolieris ielādē portāla labā un apakšējā bloka skatus; • Raksta parādīšanas skats „<i>view</i>” nodrošina rakstu nosaukumu, autoru, pievienošanas laiku un raksta ievadu publicēšanu portālā dilstošā secībā, sākot no jaunākā. |

- Atsevišķa raksta parādīšanu:

| Pieejas tiesības | Ikviens portāla apmeklētājs |
|------------------|---|
| Procesi | <ul style="list-style-type: none"> • Rakstu parādīšanas skats „<i>view</i>” izsauc kontroliera metodi „<i>read</i>”, kas nodrošina pilna raksta publicēšanu; • Kontrolieris ar „<i>POST</i>” metodes palīdzību no skata iegūst raksta identifikatoru; • Kontrolieris ielādē rakstu skatīšanās modeļa funkciju „<i>read</i>”, padodot raksta identifikatoru kā parametru; • Funkcija „<i>read</i>” no datu bāzes tabulas „<i>articles</i>” iegūst informāciju par konkrēto rakstu, ko kā masīvu atgriež kontrolierim; • Kontrolieris ielādē rakstu skatīšanās modeļa funkciju „<i>comments</i>”, padodot rakst identifikatoru kā parametru; • Funkcija „<i>comments</i>” no datu bāzes tabulas „<i>comments</i>” atlasa komentārus, kuri pievienoti konkrētajam rakstam, informāciju kā masīvu atgriež kontrolierim; • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris ielādē skatu, kurš nodrošina raksta un komentāru parādīšanu, kā parametrus padodot no rakstu skatīšanās modeļa funkcijām iegūtos datus kā asociatīvo masīvu; • Kontrolieris ielādē portāla labā un apakšējā bloka skatus; |

- Jaunu rakstu pievienošanu:

| Pieejas tiesības | Autorizējies lietotājs |
|------------------|---|
| Procesi | <ul style="list-style-type: none"> • Kontrolieris pārbauda, vai sistēmas lietotājs ir autorizējies; ja nav, apmeklētājam tiek ielādēta portāla rakstu sadaļa; • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris izsauc lietotāja informācijas modeļa funkciju „<i>get_usertype</i>”, kā parametru padodot no aktīvās sesijas datiem iegūto lietotāja identifikatoru; • Funkcija „<i>get_usertype</i>” no datu bāzes tabulas „<i>registered_users</i>” nolasa un kontrolierim atgriež konkrētā lietotāja statusu; • Kontrolieris ielādē skatu, kurā ir jauna raksta pievienošanas forma; • „<i>Form_validation</i>” bibliotēka nodrošina jaunu rakstu pievienošanas formā ievadīto datu validācijas pārbaudi, kā arī kļūdu ziņojumu parādīšanu; • Kad ievadīti korekti dati, kontrolieris ar „<i>POST</i>” metodes palīdzību tos iegūst no jauna raksta pievienošanas formas; • Kontrolieris no aktīvās sesijas datiem iegūst autorizētā lietotāja identifikatoru, kā arī lietotājvārdu; • Kontrolieris ar funkcijas „<i>date</i>” palīdzību nosaka raksta pievienošanas laiku; • Kontrolieris izsauc raksta pievienošanas „<i>add</i>” funkciju, kā parametrus padodot no jauna raksta pievienošanas formas iegūtos datus, lietotāja datus, kā arī pievienošanas laiku; • Funkcija „<i>add</i>” datus sakārto masīvā un saglabā datu bāzes „<i>articles</i>” tabulā; • Kontrolieris ielādē skatu, kas parāda paziņojumu par veiksmīgu raksta pievienošanu; • Kontrolieris ielādē portāla labā un apakšējā bloka skatus. |

- Komentāru pievienošanu:

| Pieejas tiesības | Ikviens portāla apmeklētājs |
|------------------|---|
| Procesi | <ul style="list-style-type: none"> • Kontrolieris izmanto atsevišķa raksta parādīšanas funkciju; • Kontrolieris pārbauda, vai sistēmas lietotājs ir autorizējies; ja ir - no aktīvās sesijas datiem tiek iegūts autorizētā lietotāja lietotājvārds; • Kontrolieris kā atribūtu raksta parādīšanas skatam padod arī lietotājvārdu; • Padotais lietotājvārds tiek parādīts komentāra autora ievades laukā; • „<i>Form_validation</i>” bibliotēka nodrošina komentāra pievienošanas formā ievadīto datu validācijas pārbaudi, kā arī kļūdu ziņojumu parādīšanu; |

| | |
|--|---|
| | <ul style="list-style-type: none"> • Kontrolieris iegūst korektus datus no komentāra pievienošanas formas un padod tos kā atribūtus komentāra pievienošanas funkcijai; • Komentāra pievienošanas funkcija padotos datus sakārto masīvā un saglabā datu bāzes tabulā „<i>comments</i>”; • Kontrolieris izsauc rakstu saraksta parādīšanas funkciju. |
|--|---|

2.4.1.6. ADMIN kontrolieris

ADMIN kontrolieris sevī ietver šādas funkcijas:

- Lietotāju saraksta izveidi:

| Pieejas tiesības | Administrators |
|------------------|---|
| Procesi | <ul style="list-style-type: none"> • Kontrolieris no aktīvās sesijas datiem nolasa autorizētā lietotāja statusu – ja tas neatbilst administratoram, kontrolieris ielādē portāla galveno sadaļu; • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris ielādē lietotāja informācijas modeli, kurš no datu bāzes tabulas „<i>registered_users</i>” iegūst informāciju par visiem portālā reģistrētajiem lietotājiem, ko kā masīvu atgriež kontrolierim; • Kontrolieris ielādē skatu „<i>all_users</i>”, kam kā parametrs tiek padots asociētais (no lietotāja informācijas modeļa saņemtais) masīvs; • Skats padoto informāciju attēlo tabulas veidā, katram ierakstam pievienojot divas pogas – „<i>Mainīt statusu!</i>” un „<i>Dzēst lietotāju!</i>”; • Kontrolieris ielādē portāla labā un apakšējā bloka skatus; |

- Lietotāja statusa maiņu:

| Pieejas tiesības | Administrators |
|------------------|--|
| Procesi | <ul style="list-style-type: none"> • Kontrolieris no aktīvās sesijas datiem nolasa autorizētā lietotāja statusu – ja tas neatbilst administratoram, kontrolieris ielādē portāla galveno sadaļu; • Kontrolieris izmanto lietotāju saraksta izveides funkciju; • Uzklīšķinot uz pogas „<i>Mainīt statusu!</i>” tiek izsaukta kontroliera „<i>edit_user</i>” funkcija; • Funkcija ar „<i>POST</i>” metodes palīdzību no skata iegūst lietotāja identifikatoru; • Kontrolieris ielādē lietotāja informācijas modeli, kurš, izmantojot tam kā parametru padoto lietotāja identifikatoru, no datu bāzes tabulas |

| | |
|--|--|
| | <p>„<i>registered_users</i>” nolasa informāciju par attiecīgo lietotāju un kā masīvu atgriež to kontroliera „<i>edit_user</i>” funkcijai;</p> <ul style="list-style-type: none"> • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris ielādē skatu, kas parāda atsevišķi attiecīgā lietotāja profila informāciju; • Skatā iespējams mainīt vienu lauku – lietotāja statusu; • Skats izsauc kontroliera funkciju „<i>change_user_type</i>”; • Kontrolieris ielādē portāla labā un apakšējā bloka skatus; • Kontrolieris no aktīvās sesijas datiem nolasa autorizētā lietotāja statusu – ja tas neatbilst administratoram, kontrolieris ielādē portāla galveno sadaļu; • Funkcija ar „<i>POST</i>” metodes palīdzību no skata iegūst lietotāja statusu, identifikatoru un e-pasta adresi; • Kontrolieris ielādē lietotāja informācijas modeļa statusa maiņas funkciju, kurai kā parametri tiek padoti lietotāja statuss un identifikators; • Statusa maiņas funkcija saglabā jauno statusu datu bāzes tabulā „<i>registered_users</i>”; • Atkarībā no lietotāja statusa tiek izveidota e-pasta ziņa; • Kontrolieris izsauc funkciju „<i>sendmail</i>”, kurai kā parametrus padod lietotāja e-pasta adresi, e-pasta nosaukumu un saturu; • Funkcija „<i>sendmail</i>” lietotājam, kuram tika mainīts statuss, nosūta ziņu par statusa maiņu; • Kontrolieris izsauc lietotāju saraksta izveides funkciju. |
|--|--|

- Lietotāja konta dzēšanu:

| Pieejas tiesības | Administrators |
|------------------|--|
| Procesi | <ul style="list-style-type: none"> • Kontrolieris no aktīvās sesijas datiem nolasa autorizētā lietotāja statusu – ja tas neatbilst administratoram, kontrolieris ielādē portāla galveno sadaļu; • Kontrolieris izmanto lietotāju saraksta izveides funkciju; • Uzklikšķinot uz pogas „<i>Dzēst lietotāju!</i>” tiek piek pieprasīta atkārtota dzēšanas apstiprināšana; • Kontrolieris ielādē skatus, kas nodrošina portāla navigācijas iespējas; • Kontrolieris ielādē skatu, kas parāda atkārtotas dzēšanas apstiprināšanas paziņojumu kopā ar teksta ievades lauku; |

| | |
|--|---|
| | <ul style="list-style-type: none"> • Kontrolieris ielādē portāla labā un apakšējā bloka skatus; • Uzklīķinot uz pogas „Atcelt” tiek izsaukta lietotāju saraksta izveides funkciju; • Uzklīķinot uz pogas „Dzēst” tiek izsaukta lietotāja dzēšanas funkcija, par parametru padodot teksta ievades laukā esošo tekstu; • Funkcija ar „POST” metodes palīdzību no skata iegūst lietotāja identifiatoru un lietotāja e-pasta adresi; • Kontrolieris ielādē lietotāja informācijas modeļa lietotāja dzēšanas funkciju, kurai kā parametrs padots attiecīgā lietotāja identifikators; • Ja izsaucot lietotāja dzēšanas funkciju, tika norādīts arī dzēšanas iemesls, tas tiek iekļauts e-pasta tekstā; • Kontrolieris izsauc „sendmail” funkciju, kurai kā parametri tiek padoti lietotāja e-pasta adrese, e-pasta nosaukumu un tekstu; • Funkcija „sendmail” lietotājam, kuram tika mainīts statuss, nosūta ziņu par statusa maiņu; • Kontrolieris izsauc lietotāju saraksta izveides funkciju. |
|--|---|

- Komentāru dzēšanu:

| | |
|------------------|---|
| Pieejas tiesības | Administrators |
| Procesi | <ul style="list-style-type: none"> • Kontrolieris no aktīvās sesijas datiem nolasa autorizētā lietotāja statusu – ja tas neatbilst administratoram, kontrolieris ielādē portāla galveno sadaļu; • Kontrolieris izmanto „Articles” kontroliera rakstu lasīšanas funkciju; • Uzklīķinot uz pogas „Dzēst komentāru!” tiek izsaukta „admin” kontroliera „delete_comment” funkcija; • Funkcija ar „POST” metodes palīdzību no skata iegūst attiecīgā komentāra identifiatoru; • Kontrolieris ielādē raksta pievienošanas moduļa funkciju „delete_comment”, kurai kā parametrs tiek padots komentāra identifikators; • Funkcija izdzēš komentāra datus no datubāzes; • Kontrolieris izsauc „Articles” kontrolieri. |

- Rakstu dzēšanu:

| | |
|------------------|---|
| Pieejas tiesības | Administrators |
| Procesi | <ul style="list-style-type: none"> • Kontrolieris no aktīvās sesijas datiem nolasa autorizētā lietotāja statusu – ja |

| | |
|--|---|
| | <p>tas neatbilst administratoram, kontrolieris ielādē portāla galveno sadaļu;</p> <ul style="list-style-type: none"> • Kontrolieris izmanto „Articles” kontroliera rakstu saraksta parādīšanas funkciju; • Uzklikšķinot uz pogas „Dzēst rakstu” tiek izsaukta „admin” kontroliera „delete_article” funkcija; • Funkcija ar „POST” metodes palīdzību no skata iegūst attiecīgā raksta identifiatoru; • Kontrolieris ielādē raksta pievienošanas modeļa „comments” funkciju, kā parametru padodot raksta identifikatoru, kas kontrolierim atgriež datu masīvu ar attiecīgajam rakstam pievienotajiem komentāriem; • „Foreach” cikls nodrošina visa masīva caurskatīšanu, katrā iterācijā izsaucot raksta pievienošanas modeļa komentāra dzēšanas funkciju, kurai kā parametru padod attiecīgā komentāra identifikatoru; • Kontrolieris ielādē raksta pievienošanas moduļa raksta dzēšanas funkciju, kurai kā arguments tiek padots raksta identifikators; • Raksta dzēšanas funkcija izdzēš raksta datus no datu bāzes; • Kontrolieris izsauc „Articles” kontrolieri. |
|--|---|

2.4.2. Datu detalizētais projektējums

Datu bāzes tabulu aprakstā lauka nosaukums ar pasvītrojumu nozīmē PK, bet *slīprakstā* – FK.

Skaitlis, kas ierakstīts pie tipa iekavās norāda zīmju skaitu, piemēram, int(4) norāda, ka iespējamās vērtības ir no 1 līdz 9999.

2.4.2.1. Tabulas „Articles” detalizējums

| Lauks | Tips | Atribūti | Komentāri |
|-------------------|--------------|--------------|--|
| <u>Isn</u> | int(4) | Not Null, PK | Rakstam automātiski piešķirts unikāls identificējošs numurs |
| <i>author_isn</i> | int(4) | Not Null, FK | Raksta autora identifikators; Piesaistīts no tabulas „registered_users” |
| Time | datetime | Not Null | Norāda, kad raksts pievienots protālā |
| Title | varchar(200) | Not Null | Raksta nosaukums |
| category | int(2) | Not Null | Norāda, kurā sadaļā rakstu publicēt |
| State | int(1) | Not Null | Norāda raksta pieejamību |
| Picture | varchar(200) | | Rakstam pievienotās bildes atrašanās vieta |

| | | | |
|---------|------|----------|--|
| Text | text | Not Null | Raksts |
| text_pr | text | Not Null | Raksta daļa, kas tiks parādīta pirms raksta atvēršanas |

2.4.2.2. Tabulas „ci_sessions” detalizējums

| Lauks | Tips | Atribūti | Komentāri |
|-------------------|-------------|--------------|---|
| <u>session_id</u> | varchar(40) | Not Null, PK | Automātiski piešķirts unikāls sesijas identifikators |
| ip_address | varchar(16) | Not Null | Norāda lietotāja IP adresi |
| user_agent | varchar(50) | Not Null | Norāda pārlūkprogrammu, kuru lietotājs izmanto, lai apskatītu protālu |
| last_activity | int(10) | Not Null | |
| user_data | text | Not Null | Norāda šīs sesijas autorizētā lietotāja lietotājvārdu un identifikatoru, ar kādu tas reģistrēts sistēmā |

2.4.2.3. Tabulas „comments” detalizējums

| Lauks | Tips | Atribūti | Komentāri |
|-------------|-------------|--------------|--|
| <u>Isn</u> | int(4) | Not Null, PK | Komentāram automātiski piešķirts unikāls identificējošs numurs |
| Time | datetime | Not Null | Norāda, kad komentārs pievienots protālā |
| Comment | text | Not Null | Komentārs |
| article_isn | int(11) | Not Null, FK | Norāda, kuram rakstam pievienots komentārs; Piesaistīts no tabulas „articles” |
| Author | varchar(50) | Not Null | Komentāra autors |

2.4.2.4. Tabulas „competitions” detalizējums

| Lauks | Tips | Atribūti | Komentāri |
|----------------------|--------------|--------------|---|
| <u>Isn</u> | int(11) | Not Null | Sacensībām automātiski piešķirts unikāls identificējošs numurs |
| Name | varchar(50) | Not Null | Sacensību nosaukums |
| Time | varchar(100) | Not Null | Sacensību norises datums |
| Place | varchar(100) | Not Null | Sacensību norises vieta |
| Prog | text | Not Null | Sacensību programma |
| Group | text | Not Null | Dalībnieku grupas |
| Dist | text | Not Null | Starta kārtība un distanču parametri |
| Rez | text | Not Null | Rezultāti |
| Maps | text | Not Null | Informācija par sacensību kartēm |
| App | text | Not Null | Informācija par pieteikšanos sacensībām |
| Org | varchar(100) | Not Null | Informācija par sacensību organizatoriem |
| <i>comp_added_by</i> | int(4) | Not Null, FK | Norāda, kurš lietotājs pievienojis sacensības; Piesaistīts no tabulas „registered_users” |

2.4.2.5. Tabulas „picture” detalizējums

| Lauks | Tips | Atribūti | Komentāri |
|-----------------|--------------|--------------|--|
| <u>Isn</u> | int(4) | Not Null, PK | Bildei automātiski piešķirts unikāls identificējošs numurs |
| <i>user_isn</i> | int(4) | Not Null, FK | Lietotāja, kurš pievienojis bildi, identifikators |
| Gallery | varchar(40) | Not Null | Galerijas, kurā atrodas bilde, nosaukums |
| url | varchar(200) | Not Null | Norāde uz bildes atrašanās vietu serverī |

2.4.2.6. Tabulas „registered_users” detalizējums

| Lauks | Tips | Atribūti | Komentāri |
|------------|--------------|--------------|---|
| <u>Isn</u> | int(4) | Not Null, PK | Lietotājam automātiski piešķirts unikāls identificējošs numurs |
| User | varchar(40) | Not Null | Lietotājevārds, kuru izmantojot, lietotājs var autorizēties sistēmā |
| usertype | int(1) | Not Null | Lietotāja tips; nosaka lietotāja piekļuves tiesības portālā esošajai informācijai |
| password | varchar(255) | Not Null | Uzglabā šifrētu lietotāja paroli |
| Salt | varchar(255) | Not Null | Vērtība, kuru pievieno parolei pirms tā tiek šifrēta otro reizi |
| Email | varchar(255) | Not Null | Lietotāja e-pasta adrese |
| Validated | int(1) | Not Null | Norāda, vai lietotājs apstiprinājis reģistrāciju sistēmā |
| Name | varchar(40) | Not Null | Lietotāja vārds |
| Surname | varchar(40) | Not Null | Lietotāja uzvārds |
| Group | varchar(2) | | Grupa, kurā lietotājs vēlas startēt sacensībās |
| Club | varchar(20) | | Sporta klubs, kuru lietotājs vēlas pārstāvēt sacensībās |
| By | int(4) | | Lietotāja dzimšanas gads |

2.4.2.7. Starptabulas „comp_users” detalizējums

| Lauks | Tips | Atribūti | Komentāri |
|------------|---------|--------------|--|
| <u>Isn</u> | int(11) | Not Null, PK | Sasaistei starp sacensībām un reģistrētiem lietotājiem, kuri piesaka dalību sacensībās automātiski piešķirts unikāls identificējošs numurs |
| user_isn | int(4) | Not Null, FK | Norāda, kurš lietotājs pieteicies sacensībām; no tabulas „registered_users” |
| comp_isn | int(11) | Not Null, FK | Norāda, kurām sacensībām lietotājs pieteicies; no tabulas „competitions” |

3. Testēšanas dokumentācija

3.1. Testēšanas metodika

Šajā dokumentā ir aprakstīti „Orientēšanās sacenstību portāla” testēšanas rezultāti.

Testēšana norisinājās paralēli programmatūras izstrādāšanas procesam, tādēļ visas kļūdas tika laicīgi pamanītas un līdz ar to bija vieglāk un ātrāk novēršamas.

Pēc programmatūras izstrādes tika veikta atkārtota testēšana, pēc „Melnās kastes” principa.

3.2. Funkciju testēšana

3.2.1. Lietotāja reģistrācija

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|---|--|------------------|--------------------|
| Izvēlētais lietotājvārds ir īsāks par 5 simboliem | Tiek parādīts attiecīgs kļūdas paziņojums | Jā | |
| Izvēlētais lietotājvārds ir garāks par 20 simboliem | Tiek parādīts attiecīgs kļūdas paziņojums | Jā | |
| Ievadītā parole nesakrīt ar atkārtoti ievadīto paroli | Tiek parādīts attiecīgs kļūdas paziņojums | Jā | |
| Ievadītā e-pasta adrese neatbilst standartam (teksts@teksts.domēns) | Tiek parādīts attiecīgs kļūdas paziņojums | Jā | |
| Reģistrējoties iek ievadīts jau eksistējošs lietotājvārds | Tiek parādīts attiecīgs kļūdas paziņojums | Jā | |
| Dzimšanas gada ievades laukā netiek ievadīta skaitliska vērtība | Tiek parādīts attiecīgs kļūdas paziņojums | Jā | |
| Reģistrējoties ievadītie dati ir korekti | Tiek izveidots jauns lietotājs, informācija tiek saglabāta datu bāzē, tiek parādīts paziņojums par veiksmīgu reģistrāciju; uz lietotāja norādīto e-pasta adresi nosūtīts validācijas e-pasts | Jā | |

3.2.2. Lietotāja datu apskatīšana un rediģēšana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|---|--|------------------|--|
| Rediģējot datus netiek ievadīta e-pasta adrese | Tiek parādīts attiecīgs kļūdas paziņojums | Jā | |
| Ievadītā e-pasta adrese neatbilst standartam (teksts@teksts.domēns) | Tiek izvadīts attiecīgs kļūdas paziņojums | Jā | |
| Rediģējot datus ievadīta korekta informācija | Rediģētie dati tiek saglabāti datu bāzē iepriekšējo datu vietā; Tiek parādīti labotie dati un paziņojums par veiksmīgu datu rediģēšanu | Nē | Lietotājam tiek parādīti datu bāzē saglabātie dati, līdz ar to iznāk informācijas aizkave par vienu datu maiņu |

3.2.3. Lietotāja statusa maiņa

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|--------------------------------|---|------------------|--------------------|
| Tiek mainīts lietotāja statuss | Jaunais lietotāja statuss tiek saglabāts datu bāzē iepriekšējo datu vietā, uz lietotāja profilā norādīto e-pasta adresi tiek nosūtīts e-pasts par lietotāja statusa maiņu | Jā | |

3.2.4. Lietotāja dzēšana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|-------------------------------|---|------------------|--------------------|
| Tiek dzēsts lietotājs | Tiek pieprasīts atkārtots apstiprinājums lietotāja dzēšanai | Jā | |
| Netiek atkārtoti apstiprināta | Tiek parādīts visu lietotāju | Jā | |

| | | | |
|---|---|----|--|
| lietotāja dzēšana | saraksts | | |
| Tiek dzēsts lietotājs, neievadot dzēšanas iemeslu | Uz norādītā lietotāja e-pastu tiek nosūtīts paziņojums par konta dzēšanu; Lietotāja dati datu bāzē tiek aizstāti ar „dzēsts”, „lietotājs” un „”; Tiek no jauna parādīts lietotāju saraksts | Jā | |
| Tiek dzēsts lietotājs, ievadot dzēšanas iemeslu | Uz norādītā lietotāja e-pastu tiek nosūtīts paziņojums par konta dzēšanu un iemeslu, kāpēc lietotājs dzēsts; Lietotāja dati datu bāzē tiek aizstāti ar „dzēsts”, „lietotājs” un „”; Tiek no jauna parādīts lietotāju saraksts | Jā | |

3.2.5. Autorizācija

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|--|---|------------------|--------------------|
| Tiek ievadīts neeksistējošs lietotājvārds | Tiek parādīts kļūdas paziņojums par neveiksmīgu autentifikāciju | Jā | |
| Tiek ievadīt nekorekta parole | Tiek parādīts kļūdas paziņojums par neveiksmīgu autentifikāciju | Jā | |
| Tiek ievadīts korekti dati, bet lietotāja konts nav apstiprināts | Tiek parādīts kļūdas paziņojums par neveiksmīgu autentifikāciju | Jā | |
| Tiek ievadīti korekti pieslēgšanās dati | Lietotājs tiek autorizēts sistēmā; Tiek parādīta sistēmas galvenā lapa | Jā | |

3.2.6. Raksta pievienošana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|--|--|------------------|---------------------------------|
| Netiek aizpildīti visi obligātie lauki | Tiek parādīti attiecīgi kļūdu paziņojumi | Jā | |
| Ievadītie dati ir korekti | Informācija par rakstu kopā ar pievienošanas laiku un autora identifikatoru tiek saglabāta datu bāzē | Nē | Dati netiek saglabāti datu bāzē |

3.2.7. Komentāra pievienošana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|--|---|------------------|--|
| Netiek aizpildīti visi obligātie lauki | Tiek parādīts attiecīgs kļūdas paziņojums | Nē | Tiek parādīts nepareiz kļūdas paziņojums |
| Ievadītie dati ir korekti | Dati par komentāra autoru un komentārs kopā ar raksta / bildes identifikatoru un komentāra pievienošanas laiku tiek saglabāta datu bāzē | Nē | Komentārs netiek saglabāts datu bāzē |

3.2.8. Komentāra dzēšana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|--|---|------------------|--------------------|
| Tiek uzklikšķināts uz pogas „Dzēst komentāru!” | Komentārs tiek dzēsts no datu bāzes; Tiek no jauna parādīta attiecīgā protāla sadaļa | Jā | |

3.2.9. Raksta dzēšana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|---------|-----------------------|------------------|--------------------|
|---------|-----------------------|------------------|--------------------|

| | | | |
|---|---|----|--|
| Tiek pieprasīts dzēst rakstu | Tiek pieprasīts atkārots apstiprinājums dzēst rakstu | Jā | |
| Raksta dzēšana tiek atcelta | Tiek no jauna parādīta attiecīgā portāla sadaļa | Jā | |
| Raksta dzēšana tiek apstiprināta; rakstam nav komentāru | No datu bāzes tiek dzēsts raksts; Tiek no jauna parādīta attiecīgā portāla sadaļa | Jā | |
| Raksta dzēšana tiek apstiprināta; rakstam ir komentāri | No datu bāzes tiek dzēsti rakstam pievienotie komentāri, kā arī pats raksts; Tiek no jauna parādīta attiecīgā portāla sadaļa | Jā | |

3.2.10. Jaunu sacensību pievienošana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|--|---|------------------|--------------------|
| Netiek aizpildīti visi obligātie lauki | Tiek parādīts attiecīgs kļūdas paziņojums | Jā | |
| Ievadītie dati ir korekti | Informācija par sacensībām tiek saglabāta datu bāzē; Tiek parādīts paziņojums par veiksmīgu sacensību reģistrāciju | Jā | |

3.2.11. Pieteikšanās sacensībām

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|---|--|------------------|-----------------------------------|
| Netiek aizpildīti visi obligātie lauki (lietotājs no kāda lauka informāciju izdzēsis) | Tiek parādīts attiecīgs kļūdas paziņojums | Nē | Funkcionalitāte netika izstrādāta |
| Ievadītie dati ir korekti | Tiek prasīts pieteikšanās atkārtots apstiprinājums | Nē | Funkcionalitāte netika izstrādāta |
| Ievadītie dati ir korekti un pieteikšanās tiek atkārtoti | Informācija par lietotāju un sacensībām tiek saglabāta datu bāzē | Nē | Funkcionalitāte netika izstrādāta |

| | | | |
|---|--|----|-----------------------------------|
| apstiprināta | bāzē; tiek parādīts paziņojums par veiksmīgu pieteikšanos sacensībām | | |
| Ievadītie dati ir korekti, bet pieteikšanās netiek atkārtoti apstiprināta | Tiek no jauna parādīts sacensību saraksts | Nē | Funkcionalitāte netika izstrādāta |

3.2.12. Pieteikto dalībnieku saraksta apskatīšana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|---|---|------------------|-----------------------------------|
| Autorizēts lietotājs | | | |
| Tiek nosūtīts pieprasījums parādīt sacensībām pieteikušos dalībniekus | Tiek parādīta tabula ar sacensībām pieteiktajiem dalībniekiem, kurā izcelts autorizētais lietotājs, ja viņš pieteicies sacensībām | Nē | Funkcionalitāte netika izstrādāta |
| Neautorizēts lietotājs | | | |
| Tiek nosūtīts pieprasījums parādīt sacensībām pieteikušos dalībniekus | Tiek parādīta tabula ar sacensībām pieteiktajiem dalībniekiem | Nē | Funkcionalitāte netika izstrādāta |

3.2.13. Failu pievienošana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|---|--|------------------|--|
| Netiek norādīts augšupielādes fails | Tiek parādīts attiecīgs kļūdas paziņojums | Nē | Netiek parādīts paziņojums |
| Netiek norādīts augšupielādes fails ar derīgu formātu | Tiek parādīts attiecīgs kļūdas paziņojums | Nē | Netiek parādīts paziņojums |
| Tiek ievadīti korekti dati | Fails tiek saglabāts uz servera mapē, datu bāzē tiek saglabāta informācija par faila atrašanās vietu, kā arī lietotāju | Nē | Fails tiek saglabāts uz servera mapē, bet informācija par failu datu bāzē nē |

3.2.14. Failu dzēšana

| Darbība | Sagaidāmais rezultāts | Darbojas korekti | Problēmas apraksts |
|---|---|------------------|-----------------------------------|
| Tiek nosūtīts pieprasījums dzēst failu | Tiek pieprasīts atkārtots apstiprinājums faila dzēšanai | Nē | Funkcionalitāte netika izstrādāta |
| Tiek nosūtīts pieprasījums dzēst failu, tas tiek atkārtoti apstiprināts | Faili tiek dzēsti no servera mapes, kā arī no datu bāzes tiek izdzēsta informācija par failu; Tiek parādīts paziņojums par veiksmīgu faila dzēšanu | Nē | Funkcionalitāte netika izstrādāta |
| Tiek nosūtīts pieprasījums dzēst failu, tas netiek atkārtoti apstiprināts | Tiek no jauna parādīta attiecīgā portāla sadaļa | Nē | Funkcionalitāte netika izstrādāta |

4. Programmatūras projekta organizācija

Sistēmas izstrādes sākumposmā tika izveidota programmatūras prasību specifikācija, kurā tika apkopota informācija par portālam nepieciešamajām funkcijām, ārējās saskarnes prasībām, kā arī drošību.

Projektu tika izvēlēts izstrādāt pēc ūdenskrituma metodes.

5. Kvalitātes nodrošināšana

Kvalitāte tika nodrošināta vispirms izstrādājot programmatūras prasību specifikāciju un programmatūras projektējuma aprakstu, kuru izstrādē tika ņemta vērā programmatūras prasību specifikācijas ceļvedis LVS 68:1996.

Tika pievērsta uzmanība koda strukturēšanai, kā arī tika veidoti saturīgi komentāri.

Paralēli izstrādes procesam tika testētas arī atsevišķas sistēmas daļas.

6. Konfigurāciju pārvaldība

Konfigurāciju pārvaldībā netika izmantoti papildus rīki. Kods un tā daļas periodiski tika saglabāti gan dažādās vietās datora cietajā diskā, gan arī uz ārējā datu nesēja. Kopijas tika veidotas, lai kļūdas gadījumā varētu atgriezties pie iepriekšējās versijas, kā arī lai nodrošinātos pret iespējamām tehniska rakstura problēmām.

7. Darbietilpības novērtējums

Darbietilpības novērtēšanā tika izmantots COCOMO Intermediate darbietilpības kalkulators.

| Programmatūras atribūti | |
|---|---|
| Nepieciešamā sistēmas atkarība | Datu zudumi sistēmas avārijas gadījumā nav lieli un ir viegli atjaunojami |
| Datu bāzes izmērs | Mazs (6tabulas un 1starptabula) |
| Produkta sarežģītība | Zema |
| Datora atribūti | |
| Izpildes laiks | Vidējs |
| Atmiņas ierobežojumi (aizņemtā vieta uz datu servera) | Zema |
| Virtuālās mašīnas nepastāvība | Zema |
| Datu apstrādes laiks | Ar nelielu aizkavi |
| Personāla atribūti | |
| Spēja analizēt | Vidēja |
| Iepriekšējā pieredze aplikāciju izstrādē | Zema |
| Programmēšanas spējas | Vidēja |
| Virtuālās mašīnas izmantošanas pieredze | Zema |
| Pieredze izmantotajā programmēšanas valodā | Zema |
| Projekta atribūti | |
| Mūsdienīgas programmēšanas metodes | Daļēji |
| Izmantotā programmatūra | Notepad++, phpMyAdmin, google chrome, wamp server, windows7 |
| Nepieciešamais izstrādes laiks | Kvalifikācijas darba izstrādes termiņi |

Prognozētais koda rindiņu skaits: 1500

Paredzētais projekta apjoms: 3,34 personmēneši

Reālais koda rindiņu skaits: 2093

Reālais izstrādes laiks: 3 mēneši

Reālais koda rindiņu skaits ir tik liels jo kods vairākās vietās atkārtojas (modeļu un skatu ielāde).

8. Secinājumi

Pirms programmatūras izstrādes sākšanas, ļoti svarīgi ir kārtīgi iepazīties ar programmatūras vidi, rīkiem un sintaksi.

Programmatūras kodēšana ir daudz vieglāka, ja pirms tam ir veikta programmatūras prasību specifikācijas un programmatūras projektējuma apraksta izstrāde, izplānota sistēmas uzbūve.

Kvalifikācijas darba mērķis tika gandrīz sasniegts, tomēr pēdējās testēšanas laikā atklājās nelielas kļūmes, kuras tā arī neizdevās novērst, jo pietrūka laika.

9. Izmantotā literatūra

1. LVS 68:1996 Informācijas tehnoloģija– Programminženierija – Programmatūras prasību specifikācijas ceļvedis
2. LVS 72:1996 Informācijas tehnoloģijas – Programminženierija – Ieīveicama prakse programmatūras projektējuma aprakstīšanai
3. LVS 70:1996 Informācijas tehnoloģijas – Programminženierija – Programmatūras testēšanas dokumentācija
4. http://www.codeignitor.com/user_guide/
5. <http://www.termini.lv/>
6. <http://net.tutsplus.com/sessions/codeigniter-from-scratch/>

10. Programmatūras pirmkoda paraugs

```
// administratora kontrolieris

<?php
class Admin extends Controller
{
    var $izvelne=array('vietas'=>array(
        "Jaunumi"=>"admin",
        "Sacensības"=>"competitions",
        "Bildes"=>"picture",
        "Diskusijas"=>"articles")); // augšējā navigācija

    var $kr_izvelne=array('vietas'=>array(
        "Lietotāju saraksts"=>"admin/user_list",
        ""=>"competitions")); // kreisā navigācija

function __construct()
{
    parent::Controller();
    $this->load->library('session');
    $this->load->helper('form');
    $this->load->helper('url');
}

//parāda galveno lapu
function index()
{
    if (!(($this->session->userdata("is_authenticated")) || ($this->session-
>userdata("usergroup")!=3))
        redirect(site_url('welcome'));
    else
    {
        // ielādē lapas saturu
        $this->load->view('kaste/htmlSakums');
        $this->load->view('navigacija/augsaNav',$this->izvelne);
        $this->load->view('navigacija/kreisiNav', $this->kr_izvelne);
        $this->load->view('kaste/kermenisSakums');
        $this->load->view('kermenis/welcome_message');
        $this->load->view('kermenis/sveiciens');
        $this->load->view('kaste/kermenisBeigas');
        $this->load->view('kaste/htmlBeigas');
    }
}

function user_list()
{
    if (!(($this->session->userdata("is_authenticated")) || ($this->session-
>userdata("usergroup")!=3))
```

```

        redirect(site_url('welcome'));
    else
    {
        $this->load->view('kaste/htmlSakums');
        $this->load->view('navigacija/augsaNav',$this->izvelne);
        $this->load->view('navigacija/kreisiNav', $this->kr_izvelne);
        $this->load->view('kaste/kermenisSakums');

        $this->load->model('user_model');
        $users = $this->user_model->get_all();
        $data ['query'] = $users;

        // parāda lietotāju sarakstu:
        $this->load->view('admin/all_users', $data);

        $this->load->view('kermenis/sveiciens');

        $this->load->view('kaste/kermenisBeigas');
        $this->load->view('kaste/htmlBeigas');
    }
}

function edit_user()
{
    if (!(($this->session->userdata("is_authenticated")) || ($this->session->userdata("usergroup")!=3)))
        redirect(site_url('welcome'));
    else
    {
        $user = $this->input->post('user_id');

        $this->load->model('user_model');
        $info = $this->user_model->get_info($user);
        $data ['query'] = $info;

        $this->load->view('kaste/htmlSakums');
        $this->load->view('navigacija/augsaNav',$this->izvelne);
        $this->load->view('navigacija/kreisiNav', $this->kr_izvelne);
        $this->load->view('kaste/kermenisSakums');

        // parāda atsevišķu lietotāju:
        $this->load->view('admin/edit_user', $data);

        $this->load->view('kermenis/sveiciens');
        $this->load->view('kaste/kermenisBeigas');
        $this->load->view('kaste/htmlBeigas');
    }
}

function change_user_type()
{

```

```

        if (!(($this->session->userdata("is_authenticated")) || ($this->session-
>userdata("usergroup")!=3))
            redirect(site_url('welcome'));
        else{
            $usertype = $this->input->post('usertype');
            $user=$this->input->post('user_id');
            $this->load->model('user_model');
            $email = $this->user_model->get_email($user);
            //$email = $this->input->post('email'); // kļūda

            $this->load->model("user_model");
            $this->user_model->change_info_admin($user, $usertype);

            $title="Lietotāja statusa maiņa vietnē localhost";
            $msg = 'Sveiki!
Jūsu lietotāja statuss nomainīts uz ';
            if ($usertype=='3')
            {
                $msg = $msg."Administrators"!;
            }
            else if ($usertype=='2')
            {
                $msg = $msg."Funkcionārs"!;
            }
            else
            {
                $msg = $msg."Parasts lietotājs"!;
            }
            $msg = $msg.'

Jauku dienu,
Vietnes localhost administrācija';
            if ($this->sendmail($email,$title,$msg)==true)
            redirect(site_url('admin/user_list'));
            else redirect('welcome');
        }
    }

    private function sendmail($mail, $title, $msg) {
        //todo: izveidot konstantes MAIL_USER, MAIL_PASSWORD, USER_NAME
        //MAIL_USER = "gardskakao";
        $config = Array(
            //konstantes definētas config/constants.php
            'protocol' => 'smtp',
            'smtp_host' => 'ssl://smtp.gmail.com',
            'smtp_port' => 465,
            'smtp_user' => MAIL_USER,
            'smtp_pass' => MAIL_PASSWORD,
        );
        $this->load->library('email', $config);
        $this->email->set_newline("\r\n");
    }
}

```

```

        $this->email->from(MAIL_USER, USER_NAME);
        $this->email->to($mail);
        $this->email->subject($title);
        $this->email->message($msg);

        if (!$this->email->send())
            return "Kļūda izsūtot e-pastu!";
        else
            return true;
    }

    function check()
    {
        if (!$this->session->userdata("is_authenticated") || ($this->session-
>userdata("usergroup")!=3))
            redirect(site_url('welcome'));
        else
        {
            $user = $this->input->post('user_id');

            $this->load->model("user_model");
            $info = $this->user_model->get_info($user);
            $data['query']=$info;

            $this->load->view('kaste/htmlSakums');
            $this->load->view('navigacija/augsaNav',$this->izvelne);
            $this->load->view('navigacija/kreisiNav', $this->kr_izvelne);
            $this->load->view('kaste/kermenisSakums');

            // parāda atsevišķu lietotāju:
            $this->load->view('admin/check', $data);

            $this->load->view('kermenis/sveiciens');
            $this->load->view('kaste/kermenisBeigas');
            $this->load->view('kaste/htmlBeigas');
        }
    }

    function delete_user()
    {
        if (!$this->session->userdata("is_authenticated") || ($this->session-
>userdata("usergroup")!=3))
            redirect(site_url('welcome'));
        else
        {
            $user = $this->input->post('user_id');
            $this->load->model('user_model');
            $email = $this->user_model->get_email($user);
            // $email = $this->input->post('email'); // KKAS NAV!!!!!!!!!!!!!!!!!!!!!!
            $reason = $this->input->post('reason');

```

```

        $this->load->model('user_model');
        $this->user_model->delete_user($user);

        $title="Lietotāja konta dzēšana vietnē localhost";
        $msg = 'Sveiki!
Jūs lietotāja konts vietnē localhost ir dzēsts!';
        if (trim($reason)=="")
        {
            $msg = $msg.'
Dzēšanas iemesls diemžēl nav norādīts.';
        }
        else
        {
            $msg = $msg.'
'. $reason;
        }
        $msg = $msg.'

Jauku dienu,
Vietnes localhost administrācija';
        if ($this->sendmail($email,$title,$msg)==true)
            redirect(site_url('admin/user_list'));
        else redirect('welcome');
    }
}

function delete_comment()
{
    if (!$this->session->userdata("is_authenticated") || ($this->session-
>userdata("usergroup")!=3))
        redirect(site_url('welcome'));
    else
    {
        $comment = $this->input->post('comment');
        $this->load->model('add_article_model');
        $this->add_article_model->delete_comment($comment);

        redirect(site_url('articles'));
    }
}

function delete_article()
{
    if (!$this->session->userdata("is_authenticated") || ($this->session-
>userdata("usergroup")!=3))
        redirect(site_url('welcome'));
    else
    {
        $article = $this->input->post('article_id');

```

```

        $this->load->model('view_articles_model');
        $data= $this->view_articles_model->comments($article);

        foreach ($data as $item)
        {
            $this->load->model('add_article_model');
            $this->add_article_model->delete_comment($item['isn']);
        }
        $this->load->model('add_article_model');
        $this->add_article_model->delete_article($article);

        redirect(site_url('articles'));
    }
}
}
}

```

/* End of file admin.php */

/* Location: ./system/application/controllers/admin.php */ */

// attiecīgie moduļi

// user_module

<?php

class user_model extends Model {

public function __construct()

{

parent::__construct();

\$this->load->database();

}

public function get_all()

{

\$query=\$this->db->get('registered_users');

return \$query->result_array();

}

//funkcija, kas atgriež sacensības

public function get_info(\$id)

{

//\$username = \$this->session->userdata('id');

\$query=\$this->db->get_where('registered_users', array('isn' => \$id));

return \$query->result_array();

}

public function change_info(\$id, \$email, \$group, \$club)

{

\$query = \$this->db->get_where('registered_users', array('isn' => \$id));

\$items = \$query->result();

\$usrrow = \$items[0];

\$data = array(

```

'email'=>$email,
'group'=>$group,
    'club'=>$club
    );

    $this->db->where('isn',$id);
    $this->db->update('registered_users', $data);
    return true;
}

public function change_info_admin($id, $usertype)
{
    $query = $this->db->get_where('registered_users', array('isn' => $id));
    $items = $query->result();
    $srow = $items[0];
    $data = array(
        'usertype'=>$usertype
    );
    $this->db->where('isn',$id);
    $this->db->update('registered_users', $data);
    return true;
}

public function delete_user($id)
{
    // $query = $this->db->get_where('registered_users', array('isn' => $id));
    // $items = $query->result();
    // $srow = $items[0];
    $data = array(
        'name' => "dzēsts",
        'surname'=> "lietotājs",
            'user' => "",
        'email'=> "",
        'group'=> "",
            'club'=> "",
            'by'=>"0",
            'usertype'=>0
    );

    $this->db->where('isn',$id);
    $this->db->update('registered_users', $data);
    return true;

}

public function get_mail($id)
{
    $query = $this->db->get_where('registered_users', array('isn' => $id));
    foreach ($query->result() as $row)
    {

```

```

        $name = $row->user;
    }
    return $name;
}

public function get_usertype($id)
{
    $query = $this->db->get_where('registered_users', array('isn' => $id));
    foreach ($query->result() as $row)
    {
        $type = $row->usertype;
    }
    return $type;
}

public function get_email($id)
{
    $query = $this->db->get_where('registered_users', array('isn' => $id));
    foreach ($query->result() as $row)
    {
        $mail = $row->email;
    }
    return $mail;
}

public function get_validated($user)
{
    $query = $this->db->get_where('registered_users', array('user' => $user));
    foreach ($query->result() as $row)
    {
        $val = $row->validated;
    }
    return $val;
}
}
?>

// add_article_model
<?php
class add_article_model extends Model {

    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function add($title, $category, $text_pr, $text, $state, $user, $time, $username)
    {
        $data = array(

```

```

'author_isn' => $user ,
'time' => $time,
'title' => $title ,
        'category' => $category,
        'state' => $state,
        'text' => $text,
        'text_pr' => $text_pr,
        'author' => $username
    );
    $this->db->insert('articles', $data);
    return true;
}

public function comment($comment, $article, $author, $time)
{
    $data = array(
        'author'=>$author,
        'time'=>$time,
        'comment'=>$comment,
        'article_isn'=>$article
    );
    $this->db->insert('comments',$data);
    return 1;
}

public function delete_comment($id)
{
    $this->db->delete('comments', array('isn' => $id));
    return 1;
}

public function delete_article($id)
{
    $this->db->delete('articles', array('isn' => $id));
    return 1;
}
}
?>

```

```

// view_articles_model
<?php
class view_articles_model extends Model {

    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }
}

```

//funkcija, kas atgriež rakstus

```

public function get_articles($status)
{
    if ($status==3 || $status==2)
    {
        $query=$this->db->get('articles');
    }
    else
    {
        $query=$this->db->get_where('articles', array('state' => '2'));
    }
    return $query->result_array();
}

public function read($id)
{
    $query=$this->db->get_where('articles', array('isn' => $id));
    return $query->result_array();
}

public function comments($id)
{
    $query=$this->db->get_where('comments', array('article_isn' => $id));
    return $query->result_array();
}

}
?>

// un skati
// edit_user
<div id="gietvars">
    <div id="galvenais">

        <?php if (isset($errmsg)){ ?><p class="error"><?php echo $errmsg ?></p><?php } ?>
        <p>

            <?php foreach($query as $row): echo form_open('admin/change_user_type');?>
            <table><tr>
            <input type="hidden" name="user_id" id="user_id" value="<?php echo $row['isn'];?>" />
            <td><label for="name">Vārds</label></td><td><input type="text" name="name"
            id="name" value="<?php echo $row['name'];?>" disabled="disabled"/></td>
            </tr><tr>
            <td><label for="surname">Uzvārds</label></td><td><input type="text" name="surname"
            id="surname" value="<?php echo $row['surname'];?>" disabled="disabled"/></td>
            </tr><tr>
            <td><label for="email">E-pasts</label></td><td><input type="text" name="email"
            id="email" value="<?php echo $row['email'];?>" disabled="disabled"/></td>
            </tr><tr>
            <td> <label for="usertype">Statuss</label></td><td>
            <?php if ($row['usertype']=='1'):?>
            <select name="usertype" id="usertype">

```

```

        <option value="1">Parasts lietotājs</option>
        <option value="2">Funkcionārs</option>
        <option value="3">Administrators</option>
    </select>
    <?php elseif ($row['usertype']=='2'):?>
    <select name="usertype" id="usertype">
        <option value="1">Parasts lietotājs</option>
        <option value="2" selected>Funkcionārs</option>
        <option value="3">Administrators</option>
    </select>
    <?php else: ?>
    <select name="usertype" id="usertype">
        <option value="1">Parasts lietotājs</option>
        <option value="2">Funkcionārs</option>
        <option value="3" selected>Administrators</option>
    </select>
    <?php endif;?>
</td></tr><tr><td><label for="group">Grupa</label></td><td>

    <?php if ($row['group']=='AE'):?>
        <select name ="group" value="" disabled="disabled">
            <option value ="PE">PE</option>
            <option value ="AE" selected>AE</option>
            <option value ="B" >B</option>
        </select>
    <?php elseif ($row['group']=='B'):?>
        <select name ="group" value="" disabled="disabled">
            <option value ="PE">PE</option>
            <option value ="AE" >AE</option>
            <option value ="B" selected>B</option>
        </select>
    <?php else: ?>
        <select name ="group" value="" disabled="disabled">
            <option value ="PE">PE</option>
            <option value ="AE">AE</option>
            <option value ="B" >B</option>
        </select>
    <?php endif;?>
</td></tr><tr>
    <td> <label for="club">Klubs</label></td><td><input type="text" name="club" id="club"
value="<?php echo $row['club']?>" disabled="disabled"/></td>
</tr><tr>
    <td> <label for="by">Dzimšanas gads</label></td><td><input type="text" name="by"
id="by" value="<?php echo $row['by']?>" disabled="disabled"/></td>
</tr> </p>
<tr><td></td><td>
    <input type="submit" value="Mainīt statusu!" /></td>
</tr></table>
<?php echo form_close()?>

<?php endforeach;?>

```

```

        <br />

    </div>
</div>

// all_users
<div id="gietvars">
    <div id="galvenais">
        <table>
            <?php if (isset($errmsg)){ ?><p class="error"><?php echo $errmsg ?></p><?php } ?>

            <?php foreach($query as $row): if ($row['usertype']!="0") { echo
form_open('admin/edit_user');?>
                <tr>
                    <td>
                        <?php echo $row['isn'];?>
                        <input type="hidden" name="user_id" value="<?php echo
$row['isn'];?>">
                    </td>
                    <td>
                        <?php echo $row['user'];?>
                    </td>
                    <td>
                        <?php echo $row['name'];?>
                    </td>
                    <td>
                        <?php echo $row['surname'];?>
                    </td>
                    <td>
                        <?php echo $row['email'];?>
                    </td>
                    <td>
                        <?php echo $row['usertype'];?>
                    </td>
                    <td>
                        <?php echo $row['group'];?>
                    </td>
                    <td>
                        <?php echo $row['club'];?>
                    </td>
                    <td>
                        <?php echo $row['by'];?>
                    </td>
                    <td>
                        <input type="submit" value="Mainīt statusu!" />
                    </td>
                </tr>
            <?php echo form_close(); ?>
            <?php echo form_open('admin/check');?>
                <td>

```

```

        <input type="hidden" name="user_id" id="user_id" value="<?php
echo $row['isn'];?>"/>
        <input type="hidden" name="type" id="type" value="user"/>
        <input type="submit" value="Dzēst lietotāju!" />
    </td>
    <?php echo form_close(); ?>
</tr>
<?php endforeach;?>
</table>
</div>
</div>

// check
<div id="gietvars">
    <div id="galvenais">
        <p class="error">Vai jūs tiešām vēlaties dzēst lietotāju? <br/><br/>
        <?php foreach ($query as $item):?>
            <?php echo $item['name'];?>
            <?php echo $item['surname'];?>
        <?php endforeach;?>
    </p>
    <?php echo form_open('admin/delete_user');?>
    <p>
    <label for="reason">Iemesls:</label>
    <textarea name="reason" rows='5' cols='70'></textarea>
    <br/>
    <input type="hidden" name="user_id" id="user_id" value="<?php echo
$item['isn'];?>"/>
    <input type="hidden" name="email" value="<?php echo $item['email'];?>">
    <input type="submit" value="Dzēst"/>
    </p>
    <?php echo form_close();?>
    <?php echo form_open('admin/user_list');?>
    <p>
    <input type="submit" value="Atcelt"/>
    </p>
    <?php echo form_close();?>
    </div>
</div>

```