**Ventspils Augstskola**

**Informācijas tehnoloģiju fakultāte**

**BAKALAURA DARBS**

**MĀCĪBU CENTRA VADĪBAS SISTĒMAS IZSTRĀDE**

Autors Ventspils Augstskolas

Informācijas tehnoloģiju fakultātes

bakalaura studiju programmas

„Datorzinātnes”

3. kursa students

**Arnis Freimanis**

Matr.nr. 13020020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(paraksts)

Fakultātes dekāns asoc.prof., Dr. math. Gaļina Hiļķeviča

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(paraksts)

Zinātniskais vadītājs \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ieņemamais amats, zinātniskais nosaukums, vārds, uzvārds)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(paraksts)

Recenzents \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ieņemamais amats, zinātniskais nosaukums, vārds, uzvārds)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(paraksts)

Ventspils

2016

# Anotācija

**Darba nosaukums:** Mācību centra vadības sistēmas izstrāde.

**Darba autors:** Arnis Freimanis

**Darba vadītājs:** dr. math.Oskars Jansons

**Darba apjoms:** 90 lpp., 10 tabulas, 9 attēli, 40 bibliogr norādes, 6 pielikumi.

**Atslēgas vārdi:** attēlu segmentācija, algoritmi

Bakalaura darbā ir apskatīti vairāki eksistējošie attēlu segmentācijas metožu algoritmi un šo metožu trūkumi. Papildus aprakstīti dažādi attēlu uzlabošanas paņēmieni, kuri var būt pielietoti attēlu segmentācijas rezultāta kvalitātes palielināšanai.

Jaunums segmentācijas metodes izstrādāšanas mērķiem tiek piedāvāti vairāki algoritmi, tai skaitā arī darba autora patstāvīgi izstrādātie segmentācijas algoritmi un attēlu uzlabošanas paņēmieni. Algoritmu izpētei izveidota programma, kas ļauj salīdzināt piedāvātās metodes, un veikta šo algoritmu analīze.

Darbā ir izstrādāta jauna attēlu segmentācijas metode, kas izmanto gan attēlu uzlabošanas, gan izstrādātos algoritmus. Jaunās metodes darbības pētīšanai un novērtēšanai izstrādāta programma, kas būtiski atvieglo šo procesu.

# Anotācija (ENG)

**Darba nosaukums:** Mācību centra vadības sistēmas izstrāde.

**Darba autors:** Arnis Freimanis

**Darba vadītājs:**

**Darba apjoms:**

**Atslēgas vārdi:**

# Anotācija (RU)

**Darba nosaukums:** Mācību centra vadības sistēmas izstrāde.

**Darba autors:** Arnis Freimanis

**Darba vadītājs:**

**Darba apjoms:**

**Atslēgas vārdi:**

**SATURS**

[Anotācija 2](#_Toc449442989)

[Anotācija (ENG) 3](#_Toc449442990)

[Anotācija (RU) 4](#_Toc449442991)

[Ievads 7](#_Toc449442992)

[Saīsinājumu un nosacīto apzīmējumu 8](#_Toc449442993)

[1. Informācijas sistēma 9](#_Toc449442994)

[1.1. Informācijas sistēmas jēdziens 9](#_Toc449442995)

[1.2. Informācijas sistēmas izstrādes dzīves cikls 10](#_Toc449442996)

[1.3. Informāciju sistēmas izstrādes modeļi 11](#_Toc449442997)

[2. Datu bāzes un saskarnes risinājumi 16](#_Toc449442998)

[2.1. Datu bāzes veidošanas risinājumi 16](#_Toc449442999)

[2.1.1. Datu bāzes veidošanas alternatīvas 16](#_Toc449443000)

[2.1.2. Izvēlētās tehnoloģijas piedāvātās iespējas 19](#_Toc449443001)

[2.2. Saskarnes veidošanas risinājumi 20](#_Toc449443002)

[2.2.1. Saskarnes veidošanas populārākās alternatīvas 21](#_Toc449443003)

[2.2.2. Izmantotās tehnoloģijas pamatojums 22](#_Toc449443004)

[3. Mācību centra vadības sistēmas projektēšana 23](#_Toc449443005)

[3.1. Mācību centra vadības sistēmas prasības 23](#_Toc449443006)

[3.2. Lietošanas gadījumu diagramma 23](#_Toc449443007)

[3.3. Virsraksts 25](#_Toc449443008)

[4. Mācību centra vadības sistēmas izstrāde 26](#_Toc449443009)

[4.1. MySQL datu bāzes izstrāde 26](#_Toc449443010)

[4.1.1. Sistēmas datu bāze 26](#_Toc449443011)

[4.1.2. Sistēmas datu bāzes tabulas 27](#_Toc449443012)

[4.2. Saskarnes izstrāde 30](#_Toc449443013)

[4.2.1. Autorizācija un profila lapa 30](#_Toc449443014)

[Secinājumi un priekšlikumi 32](#_Toc449443015)

[Izmantotās literatūras un avotu saraksts 33](#_Toc449443016)

[Pielikums 34](#_Toc449443017)

[Galvojums 35](#_Toc449443018)

# Ievads

**Tēmas aktualitātes pamatojums**

Uzņēmumu un organizāciju vadītājiem, kuri ir atbildīgi par IT infrastruktūras darbību un attīstību, nereti nākas meklēt risinājumus specifisku uzdevumu veikšanai, jo tirgū nav pieejami gatavi produkti, kas pilnībā nodrošinātu konkrēto problēmu risinājumu. Tāpēc visefektīvākais risinājums ir tieši konkrētajam klientam izstrādāta informācijas sistēma, kas atbilst visām uzņēmuma vai organizācijas īpašajām prasībām, nodrošinot tās efektīvu darbību un sniedzot būtisku atbalstu uzņēmuma vai organizācijas vadībai. [1]

Izstrādātā sistēma atvieglos darbu un ļaus ekonomēt laiku mācību grupu plānošanā un komandējumu organizēšanā. Sistēmā tiks glabāta informācija par darbinieku veiktajām apmācībām, viņu zināšanām, iegūtajiem sertifikātiem, par pašu darbinieku, kursiem, pasniedzējiem un telpu noslogojumu.

**Darba mērķis**

Darba mērķis ir projektēt un izstrādāt uzņēmuma mācību centra vadības sistēmu (saīsināti – MCVS), kuras galvenie uzdevumi ir mācību grupu plānošana un komandējumu organizēšana.

**Darba mērķa sasniegšanai nepieciešamie uzdevumi**

1. Izpētīt un aprakstīt svarīgāko literatūru par informācijas sistēmas izstrādi, dzīves ciliem un tās pazīstamākajiem izstrādes modeļiem.
2. Izpētīt informāciju par dažādām datu bāžu izstrādes tehnoloģijām, aprakstīt to priekšrocības un īpatnības. Papildus aprakstīt datu bāzes izstrādei izvēlēto tehnoloģiju un tās īpašības.
3. Izpētīt un apgūt tīmeklī bāzētu lietotņu izstrādes tehnoloģijas, aprakstīt svarīgāko informāciju, to priekšrocības un īpašības, kā arī aprakstīt sistēmas izstrādei izvēlēto tehnoloģiju.
4. Iepazīties un aprakstīt sistēmas prasības.
5. Pamatojoties uz iegūto informāciju par sistēmu, projektēt un izstrādāt datu bāzi, izmantojot izvēlēto tehnoloģiju.
6. Aprakstīt izveidotās datu bāzes struktūru, tabulas un saišu nozīmi
7. Izstrādāt MCVS lietotājam draudzīgu tīmeklī bāzētu lietotni izmantojot izvēlēto tehnoloģiju un saistīt lietotni ar izveidoto datu bāzi.
8. Sagatavot sistēmas dokumentāciju.
9. Sagatavot testpiemērus un veikt testēšanu.

# Saīsinājumu un nosacīto apzīmējumu

Darbā izmantoto saīsinājumu un nosacīto apzīmējumu sarakstu noformē uz atsevišķas lapas. Ja saīsinājumu un nosacīto apzīmējumu ir mazāk nekā 10, tos var paskaidrot arī tekstā.

IS

B2B

DBPS

Agile

MCVS

RDBMS

ACID

OS

CGI

# Informācijas sistēma

## Informācijas sistēmas jēdziens

Informāciju sistēma ir programmatūra, kas mums palīdz organizēt un analizēt datus. Tā palīdz atbildēt uz jautājumiem un risināt problēmas, kas attiecas uz organizācijas attīstību. [1] IS ir integrēts komponenšu kopums informācijas savākšanai, uzglabāšanai un nodrošināšanai ar informāciju, zināšanām un digitāliem produktiem. Biznesa uzņēmumi un citas organizācijas paļaujas uz informāciju sistēmām, lai veiktu un vadītu savas darbības, mijiedarbojoties ar saviem klientiem, piegādātājiem un konkurētu tirgū. Informāciju sistēmas tiek izmantotas lai vadītu starptautisku organizāciju piegāžu ķēdes un elektroniskos tirgus. Piemēram, korporācijas izmanto informāciju sistēmas, lai sagatavotu finanšu pārskatus, lai pārvaldītu savus cilvēkresursus un sasniegtu savus potenciālos klientus ar tiešsaistes reklāmām. [2]

Daudzi lielie uzņēmumi ir pilnībā uzbūvēti uz informācijas sistēmām. Pazīstamākie šāda veida uzņēmumi ir “eBay”, kas ir izveidojis lielāko izsoles tirgu; “Amazone”, kas attīsta elektronisko tirdzniecības centru un sniedz mākoņa pakalpojumus; “Alibaba”, kas piedāvā “B2B” e-tirgu; un Google, meklētājprogrammas kompānija, kas lielāko daļu savus ienākumus gūst lieto ievietošanas interneta resursos. Tā pat arī publiskā sektora iestādes izmanto informācijas sistēmas, lai izmaksu ziņā lētāk sniegtu pakalpojumus iedzīvotājiem. Kā arī digitālās preces, piemēram, elektroniskās grāmatas, video produktus, programmatūras un tiešsaistes pakalpojumus, piemēram, spēles un sociālos tīklus arī pārvalda ar informācijas sistēmām. [2]

Ir daži vispārīgi informāciju sistēmu veidi, piemēram, datu bāzes pārvaldības sistēma (DBPS) ir programmatūras un datu apvienojums, kas ļauj organizēt un analizēt datus. DBPS programmatūra parasti nav paredzēta darbam ar konkrētu organizāciju vai konkrēta veida analīzi, drīzāk, tā ir universālas informāciju sistēmas. Cits piemērs ir elektroniskās izklājlapas, tas ir instruments, vai veiktu pamata datu analīzi, pamatojoties uz formulām, kas nosaka attiecības starp datiem. Piemēram, lietotāji izklājlapu var izmantot, lai aprēķinātu kādu datu vidējo vērtību, vai kāda noteikta laika posma vērtību. [1]

Turpretim ir vairākas specializētas informācijas sistēmas, kas ir īpaši izstrādātas, lai atbalstītu kādu konkrētu procesu organizācijā lai veiktu ļoti specifiskas analīzes uzdevumus. Piemēram uzņēmuma resursu plānošana ir informācijas sistēma, ko izmanto, lai integrētu visu vadības iekšējo un ārējo informāciju par uzņēmumu. Vēl viens piemērs ir ģeogrāfiskās informāciju sistēmas, kas tiek izmantotas, lai pārvaldītu un analizētu visa veida ģeogrāfiskos datus. [1]

Lai gan informāciju sistēmas var atšķirties, pēc to pielietojuma, tās parasti satur šādas sastāvdaļās:

* *Aparatūra*: uz datoru balstītas informāciju sistēmas izmanto datortehniku, procesorus, monitorus, klaviatūras, printerus un tamlīdzīgi.
* *Programmatūra*: tās ir programmas, ko izmanto lai organizētu, apstrādātu un analizētu datus.
* *Datu bāze*: informācijas sistēmas darbam ar datiem, organizēta tabulās un failos.
* *Tīkls*: dažādiem elementiem ir jābūt savienotiem savā starpā, lai vairāki lietotāji varētu izmantot sistēmu vienlaicīgi.
* *Procedūras*: tās apraksta, kādā secībā specifiski dati tiek apstrādāti un analizēti, lai saņemtu atbildes uz jautājumiem, kādus izstrādātā sistēma uzdod.

Pirmās četras komponentes ir daļas no vispārējas informācijas tehnoloģiju organizācijā. Procedūras ir ļoti specifiska informācija, kas vajadzīga, lai atbildētu uz konkrētu jautājumu. [1]

## Informācijas sistēmas izstrādes dzīves cikls

Informācijas sistēmas izstrādes dzīves cikls ir viss programmatūras pastāvēšanas laiks - no tās izstrādāšanas sākuma līdz brīdim, kad tā ir zaudējusi savu praktisko vērtību. Programmatūras dzīves cikla galvenās fāzes ir projektēšana, izstrādāšana, testēšana, nodošana ekspluatācijā, uzturēšana un, iespējams, arī modernizēšana. [4]

“IS netop vienā mirklī, un tai ir savs dzīves cikls, kas tiek aprakstīts ar dzīves cikla modeļiem. Jau kopš 70. gadu sākuma par klasiskām sistēmas izstrādes dzīves cikla sastāvdaļām tiek uzskatītas šādas fāzes” [3]:

* *Plānošana, priekš izpēte* - tā ir izmeklēšana, ko veic sistēmu analītiķis, lai uzzinātu kādas ir galvenās problēmas saistībā ar jauno, vai jau esošo sistēmu, un, vai tas ir tehniski iespējams un rentabli, lai atrisinātu šīs problēmas ar datoru balstītu risinājumu palīdzību.
* *Analīze* – sistēmas analīzes mērķis ir noteikt, kur ir problēma, lai salabotu vai izveidotu jaunu sistēmu. Šis solis sadala sistēmas analīzi vairākos mazos posmos, kas ir: noskaidrot, kas ir izveidots, iesaistīt lietotājus un uzklausīt viņu viedokli, definēt konkrētas prasības un mērķus.
* *Projektēšana* – šajā posmā tiek identificēti un novērtēti iespējamie sistēmas izstrādes risinājumi un atrasts labākais risinājums, lai izstrādātu sistēmu. Kā arī tiek izvērtētas dizaina specifikācijas un izstrādāta sistēma, kas sevī ietver: ievadi, izvadi, datu glabāšanu, lietotāju interfeisu, drošības procedūras, dublēšanas un atjaunošanas procedūras.
* *Ieviešana* – šis posms ietver, sistēmas uzstādīšanu, lai tā atbilstu visām dizaina specifikācijām. Testēšanu, kas jāveic izmantojot testēšanas plānu, lai pārliecinātos, ka visas sistēmas daļas darbojas pareizi ar normāliem un ārkārtīgi kļūdainiem datiem.
* *Atbalsts* – jauno informācijas sistēmu tās lietošanas laikā var būt nepieciešams mainīt vai uzlabot un šis posms, kas sevī ietver: izmaiņu ieviešana, kādas ir nepieciešamas lietotājiem, ja testēšanas laikā ir atrastas kļūdas, to labošana un uzlabojumi, kas nepieciešami, jau tad, kad sistēma aktīvi tiek lietota. [4]

. att. Informāciju sistēmas dzīves cikls (adaptēts no [6])

## Informāciju sistēmas izstrādes modeļi

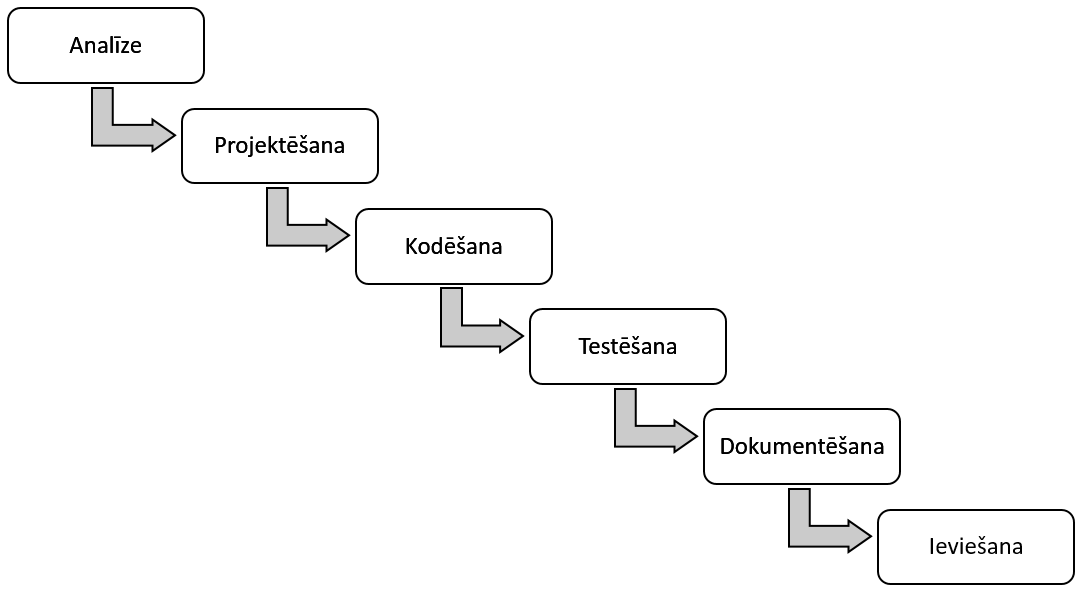
Izstrādes modeļi ir dažādi procesi un metodes, kas tiek izmantoti sistēmas izstrādē atkarībā no projekta mērķiem un veicamajiem uzdevumiem. Ir vairāki izstrādes modeļi, kas ir izstrādāti, lai sasniegtu dažādus nepieciešamos mērķus. Modeļi precizē dažādus procesa posmus un kārtību kādā tie tiks veikti [5].

Ir dažādi programmatūras izstrādes modeļi, tie ir šādi:

* *Ūdenskrituma modelis*
* *V modelis*
* *RAD modelis*
* *Agile modelis*
* *Spirālveida modelis*
* *Iteratīvais modelis*
* *Pakāpeniskais modelis*

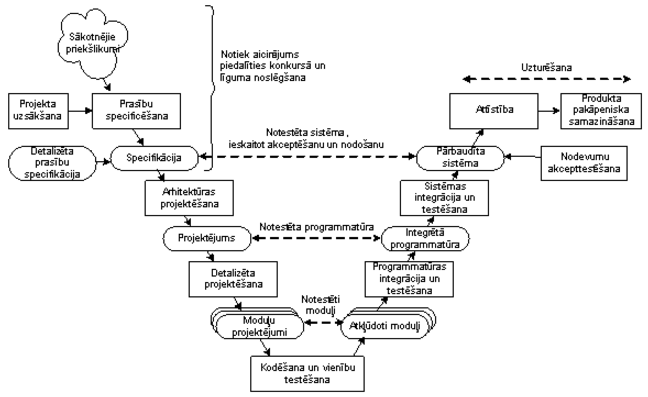
Autors savā darbā apskatīs trīs visbiežāk izmantotos sistēmas izstrādes modeļus, kas ir ūdenskrituma modeli, V modeli un Agile modeli. Pēc modeļu apskatīšanas un aprakstīšanas tiks veikta to salīdzināšana, lai noskaidrotu, kuru no modeļiem, kādās situācijās izmantot.

* *Ūdenskrituma modelis* – “Ūdenskrituma modelī sistēmas izstrāde ir sadalīta vairākās secīgās aktivitātēs vai fāzēs, kuras katras aktivitātes rezultāts tiek izmantots par ieejas informāciju nākamajai aktivitātei. Mūsdienās par ūdenskrituma modeli sauc jebkuru izstrādes metodoloģiju ar secīgām aktivitātēm neatkarībā no fāžu skaita un specifikas. Šim modelim ir virkne pozitīvu īpašību. Katras aktivitātes beigās tiek pielietota izveidot produktu verifikācija un validācija, kas palīdz izveidot kvalitatīvu produktu. Gadījumos, kad pasūtītāja prasības ir labi zināmas un saprotamas, ūdenskrituma modelis ir tas ar kura palīdzību visātrāk un kvalitatīvāk būs iespējams sasniegt uzstādīto mērķi. Viens no ūdenskrituma modeļa trūkumiem ir uzturēšanas fāzes neadekvāta pielietošana. Tā tiek uztverta kā atsevišķa fāze, it kā tai būtu atsevišķa ieejas un izejas informācija. Ūdenskrituma modelī netiek aprakstīta būtiskas uzturēšanas fāzes atšķirības no pārējām aktivitātēm. Jāpatur prātā, ka lielākā daļa darbietilpības, kura tiek patērēta programmatūras produkta dzīves cikla laikā, tiek patērēta uz uzturēšanas aktivitātēm.” [6]



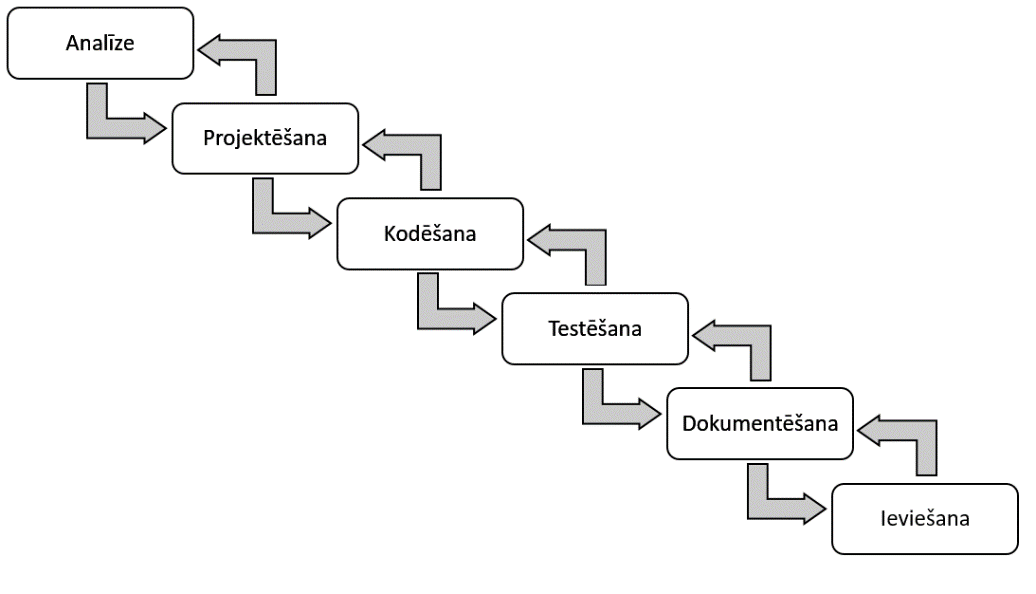
. att. Ūdenskrituma modelis

* *V modelis* – “šajā modelī, kurš ir ūdenskrituma modeļa variācija, visas aktivitātes, fāzes ir sakārtotas burta ‘V’ formā. Šajā diagrammā kreisais V zars parāda progresu sākot ar analīzi un projektēšanu līdz programmēšanai un pieaugošam sistēmas sadalījumam komponentēs. Labais V zars attēlo progresējošu sastāvdaļu salikšanu un testēšanu, beidzoties ar programmatūras produkta piegādi. Šī modeļa svarīgā īpašība ir dažādu fāžu savstarpējas atbilstības parādīšana. Piemēram, patstāvīgas programmas vai moduļi tiek testēti pret patstāvīgo moduļu projektējumiem, integrētā sistēmai tiek veikta sistēm-testēšana pret sistēmas projektējumu un visbeidzot gala sistēmai gala lietotāji veic akcept testēšanu pret sākotnējām prasībām, noformētām prasību specifikācijā. Gadījumos, kad projektā tiek pieaicināti ārējie apakškontraktori, šis modelis nodrošina skaidru izstrādes darba uzdevumu definēšanu un nodevum pārbaudi.” [6]



1.3 att. "V" modelis

* *“Agile” modelis* – spējās attīstības modelis ir iteratīvs un pakāpenisku procesu modelis ar uzsvaru uz procesu pielāgošanās spējām un klientu apmierinātību, apvienojumā ar ātru programmatūras izstrādes procesu. “Agile” metode sadala produkta attīstību vairākos mazos iteratīvos izstrādes procesos. Katra iterācija parasti ilgst, vienu līdz trīs nedēļās. Katra iterācija sevī ietver savstarpējas funkcionālas grupas, kas strādā vienlaicīgi dažādās jomās, piemēram, plānošana, prasību analīze, projektēšana, programmēšanā, vienību pārbaudes un akceptēšanas testēšana. [7]



. att. "Agile" modelis

Visi trīs izstrādes modeļi ir plaši pazīstami un tiem ir savi pozitīvās un negatīvās īpašības, lai izstrādātu nepieciešamo sistēmu šim darbam, autors nolēma izpētīt un izanalizēt kurš no modeļiem būtu piemērotāks.

Ūdenskrituma modelis ir vienkāršs, viegli saprotams un izmantojams. To ir viegli lietot, jo skaidri ir noteikti katra posma konkrētie uzdevumi un pārbaudes procesi. Šajā modelī posmi tiek apstrādāti un pabeigti vienā laikā, posmi nepārklājas. Ūdenskrituma modelis labi strādā arī maziem projektiem, kuru prasības ir ļoti labi saprotamas.

V modelis arī ir vienkāršs un viegli lietojams. Testēšanas darbības, piemēram, plānošana un testu projektēšana notiek krietni pirms programmēšanas, kas ietaupa daudz laika. Tā radot iespēju gūt labākus panākumus nekā ar ūdenskrituma modeli. Veidojot šos testus arī kļūdas tiek atklātas agrākās stadijās un tos ir iespējams ātrāk novērst. Tā pat kā ūdenskrituma modelis arī V modelis labi strādā arī mazajos projektos, kuriem prasības ir viegli saprotamas.

“Agile” modelis spēj pārsteigt klientu ar ātru, nepārtrauktu izstrādes stadijā esošas programmatūras piegādi. Klienti, izstrādātāji un testētāji patstāvīgi sadarbojas viens ar otru. Strādājošas programmatūras veiktie uzlabojumi tiek klientam atrādīti biežāk, nekā citos modeļos. Lielu lomu spēlē klienta un izstrādātāja komunikācija izstrādes laikā. Modelis nodrošina tehnisku izcilību un labu dizainu. Bet tas ir vairāk piemērots, kad sistēmas prasības izstrādes sākumā vēl nav zināmas.

Autors sistēmu izstrādās pēc ūdenskrituma modeļa, jo pirms izstrādes ir noskaidrotas un definētas sistēmas prasības. Izstrādājamā sistēma nav liels projekts, tajā iesaistās tikai autors un procesu mijiedarbošanās nav nepieciešama tāpēc autors uzskata ka ūdenskrituma modelis ir piemērotākais darbā aprakstītās sistēmas izstrādei.

# Datu bāzes un saskarnes risinājumi

## Datu bāzes veidošanas risinājumi

Izstrādājamās sistēmas datus ir nepieciešams uzglabāt datu bāzē, lai to izdarītu vispirms ir nepieciešams apskatīt datu bāzu izstrādes alternatīvas un plašāk apskatīt un aprakstīt kādu no alternatīvām, kas tiks izmantota sistēmas izstrādē.

“Datu bāze ir savstarpēji saistītu objektu kopums, kas ar speciālas pārvaldības sistēmas starpniecību organizēts tā, lai nodrošinātu ērtu informācijas ieguvi, atlasi un kārtošanu. Datu bāzes plaši izmanto ražošanā, pārvaldē, tirdzniecībā utt. Piemēram, ir datu bāzes, kurās apkopota informācija par Latvijas iedzīvotājiem, uzņēmumiem, komunālajiem pakalpojumiem, pirkstu nospiedumiem, atsevišķu uzņēmumu darbību. Datu bāzei ir jānodrošina trīs galvenās funkcijas:

* datu ievadīšanu, rediģēšanu un papildināšanu;
* uzkrātās informācijas kārtošanu, atlasi un apskati;
* pārskatu sastādīšanu par datu bāzē ietverto informāciju.

Datu bāzes veido un lieto, izmantojot datu bāzes pārvaldības sistēmas jeb datu bāzes lietotnes, kas organizē datus datu bāzē, nodrošinot to uzglabāšanu, atlasi un drošību.” [8]

### Datu bāzes veidošanas alternatīvas

Datubāzes ir pamats mūsdienu uzņēmējdarbībā. Gan publiskiem sektoriem, gan augstu tehnoloģiju nozarēm, infrastruktūra uzņēmumā nav tik nozīmīga, kā dati, to uzglabāšana, analīze, darījumi spēlē lielu lomu. Lai autors izvēlētos kādu datu bāžu pārvaldības sistēmu izmantot izstrādājot savu sistēmu, sākotnēji salīdzina piecas populārākās alternatīvas: MySQL, PostgreSQL, Oracle, SQLite, MS Access.

. tabula

**MySQL, PostgreSQL, Oracle, SQLite un MS Access datu bāžu salīdzinājums** [9]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Datu bāzes nosaukums | MySQL | PostgreSQL | Oracle | SQLite | MS Access |
| Apraksts | Plaši izmanto RDBMS | Plaši izmanto RDBMS | Plaši izmanto RDBMS | Plaši izmanto RDBMS | Microsoft Access apvieno “backend” RDBMS (JET / ACE Engine) ar GUI “frontend” datu manipulācijām un vaicājumiem. |
| Datubāzes modelis | Relāciju datubāzes pārvaldību sistēma | Relāciju datubāzes pārvaldību sistēma | Relāciju datubāzes pārvaldību sistēma | Relāciju datubāzes pārvaldību sistēma | Relāciju datubāzes pārvaldību sistēma |
| Izstrādātājs | Oracle | PostgreSQL Globālā Izdtrādātāju Grupa | Oracle | Dwayne Richard Hipp | Microsoft |
| Izlaišanas gads | 1995 | 1989 | 1980 | 2000 | 1992 |
| Izstrādes valodas | C un C++ | C | C un C++ | C | C++ |
| Servera operētājsistēmas | FreeBSD, Linux, OS X, Solaris, Windows | FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows | AIX, HP-UX, Linux, OS X, Solaris, Windows, z/OS | Bez servera | Bez servera |
| Licence | Atvērta koda | Atvērta koda | Komerciāla | Atvērta koda | Komerciāla |
| Atbalstītās programmēšanas valodas | Ada, C, C#, C++, D, Eiffel, Erlang, Haskell, Java, Objective -C, OCaml, Perl, PHP, Phyton, Ruby, Scheme, TCL | .Net, C, C++, Java, Perl, Python, Tcl | C, C++, C#, Clojure, Cobol, Eifell, Erlang, Fortran, Groovy, Haskell, Java, JavaScript, Lisp, Objective C, OCalm, Perl, PHP, Phyton, R, Ruby, Scala, Tcl, Visual Basic | Actionscript, Ada, Basic, C, C#, C++, D, Delphin, Forth, Fortran, Haskell, Java, JavaScript, Lisp, Lua, MatLab, Objective-c, OCaml, Perl, PHP, PL/SQL, Python, R, Ruby, Scala, Scheme, Smaltalk, Tcl | C, C#, C++, Java, VBA, Visual Basic.NET |
| Servisa puses skripti | Jā | Lietotāja definētas funkcijas | PL/SQL | Nē | Jā |
| Trigeri | Jā | Jā | Jā | Jā | Jā |
| Ārējās atslēgas | Jā | Jā | Jā | Jā | Jā |
| Transakciju koncepts | ACID | ACID | ACID | ACID | ACID |
| Iepriekš definēti datu tipi | Jā | Jā | Jā | Jā | Jā |
| Sekundāri indeksi | Jā | Jā | Jā | Jā | Jā |
| Datu shēmas | Jā | Jā | Jā | Jā | Jā |

Salīdzinot visas piecas izvēlētās datu bāžu alternatīvas var secināt :

1. Visām apskatītajām datu bāzēm modelis ir relāciju datu bāzes pārvaldības sistēma.
2. SQLite un MS Access ir datu bāzes kurām nav nepieciešams serveris, kas liecina, ka tās ir daudz vienkāršāka tipa datubāzes.
3. Kā arī MS Access un Oracle nav atvērta koda datu bāzes un tām ir pieejamas tikai nepilnas testa versijas. Pēc autora domām SQLite, Ms Access un Oracle nebūs pietiekami atbilstošas, lai izstrādātu nepieciešamo sistēmu. Tā kā sistēma sevī uzglabās daudz informācijas par darbiniekiem, telpām un kursiem autors uzskata, ka SQLite nespēs sniegt ātru piekļuvi datiem un Oracle un MS Access piedāvātās bezmaksas versijas arī nenodrošinās nepieciešamo funkcionalitāti.
4. PostgreSQL tradicionāli koncentrējas uz uzticamību, integrētiem datiem un integrētām izstrādātāja veidotām funkcijām. Tai ir ārkārtīgi sarežģīta vaicājumu plānošana, kas spēj savienot salīdzinoši lielu skaitu tabulas.
5. Savukārt MySQL, pretēji, tradicionāli koncentrējas uz tīmekļa vietnēm, kurās ir iespējams apskatīt daudz informācijas, kas parasti ir izstrādātas ar PHP palīdzību. Galvenās darbības notiek ar vienkāršiem vaicājumiem un šī alternatīva nodrošina ātru datu apstrādi un piekļuvi tiem.

### Izvēlētās tehnoloģijas piedāvātās iespējas

Lai izstrādātu sistēmas datu bāzi autors apskatot un izpētot dažādas datu bāžu tehnoloģijas ir izvēlējies savā darbā izmantot MySQL. MySQL ir ļoti spējīga klientu un servera relāciju datu bāze. Tā ir pietiekami droša un stabila dažāda veida lietotnēm un tā piedāvā lielisku izmaksu/ ieguvumu attiecību. Ne tikai tāpēc ka MySQL ir bezmaksas, bet arī tāpēc, ka tai ir pieticīgas prasības uz aparatūru. [10]

MySQL dēļ mazā izmēra un ātrumā ir ideāls datu bāzu risinājums tīmekļa vietnēm, tā priekšrocības ir šādas:

* *Viegli lietot*: lai gan pamat zināšanas par SQL ir nepieciešamas un lielākajai daļai relāciju datu bāžu būs tieši tāda pati prasības, bet tas vienalga neliecina par to ka MySQL ir sarežģīti lietot, tieši pretēji to ir ļoti viegli izmantot. Ar tikai dažiem, pa visam vienkāršiem, SQL vaicājumiem ir iespējam veidot mijiedarbību ar MySQL datu bāzi.
* *Droša*: MySQL iekļauj spēcīgu datu drošību, kas aizsargā konfidenciālos datus no iebrucējiem. Tiesības var tik iestatītas, lai atļautu dažas vai visas privilēģijas dažādiem lietotājiem. Visas paroles tiek šifrētas .
* *Nav dārga:* MySQL ir iekļauts bez maksas NetWare® 6.5 un to ir iespējam lejupielādēt bez maksas no MySQL tīmekļa vietnes.
* *Ātra*: liela loma ir ātrums, lai to nodrošinātu MySQL izstrādātāji pieņēma lēmumu piedāvāt mazāk funkcijas nekā citi lielākie datu bāžu konkurenti, piemēram, “Sybase” un “Oracle”. Tomēr, neskatoties uz to, ka MySQL piedāvā mazāk funkciju nekā konkurenti, tā joprojām sniedz visas tās funkcijas, ko pieprasa lielākā daļa datu bāžu izstrādātāji.
* *Pielāgojama*: MySQL var apstrādāt gandrīz jebkuru datu apjomu, līdz pat 5 000 000 rindiņu vai pat vairāk. Noklusējumā datu bāzes faila lielums ir ierobežots aptuveni līdz 4 GB, bet to ir iespējams mainīt, palielinot līdz pat 8 TB.
* *Ļoti labi pārvalda atmiņu*: MySQL serveris ir rūpīgi izstrādāts un testēts, lai novērstu atmiņas noplūdes.
* Strādā uz dažādām operētājsistēmām: MySQL atbalsta daudz dažādas operētājsistēmas, to skaitā Windows, Linux, UNIX, OS/2, Solaris, FreeBSD un citas. [11]
* *Plaša lietotņu izstrāde:* Līdzīgi, kā ar operētājsistēmām, MySQL atbalsta dažādas programmēšanas valodas, to skaitā C, C++, Java, PHP, Perl un citas. MySQL ir ļoti draudzīga ar PHP valodu, ar programmēšanas valodu, kas ir visaugstāk novērtētā programmēšanas valoda priekš WEB izstrādes un pielāgošanas. [12]

## Saskarnes veidošanas risinājumi

Lai visi uzņēmuma darbinieki varētu iegūt informāciju par sev tuvākajām mācībām un komandējumiem jaunajai sistēmai ir jāspēj pieslēgties jebkurā dienā un diennakts laikā, tāpēc ir nepieciešams izstrādāt tīmeklī bāzētu lietotni. Lai to panāktu vispirms ir nepieciešams apskatīt dažādas tīmekļu bāzētu lietotņu tehnoloģiju alternatīvas, kā arī aprakstīt, salīdzināt tās un aprakstīt izvēlēto alternatīvu.

Tīmeklī bāzēta lietotne ir jebkura lietotne, kas lieto interneta pārlūka programmu, lai nodrošinātu klienta piekļuvi tai. Tā var būt pa visam vienkāršs piemēram, kā ziņojumu dēlis, vai kā viesu grāmata. Tīmeklī bāzētas lietotnes priekšrocība ir, ka izstrādātājam nav jāuztraucas par to kādu datoru lieto lietotājs un to kāds tam ir procesors. Tai skaitā nav jādomā par to kādu operētājsistēmu un interneta pārlūka programmu lietots lietotājs. Tas nozīmē, ka izstrādātā lietotne vienādi darbosies dažādos apstākļos.

### Saskarnes veidošanas populārākās alternatīvas

Eksistē daudz un dažādu tehnoloģiju ko izmantot tīmeklī bāzētas lietotnes izstrādē, lai autors noskaidrotu, kura būtu piemērotākā viņā lietotnei, autors apskatīs un salīdzinās četrus iespējamos variantus, kas ir: Perl, ASP, PHP, Java.

.. tabula

**Perl, ASP, PHP un Java tehnoloģiju salīdzinājums** [13][14][15]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nosaukums | Perl | ASP | PHP | Java |
| Licence | Artistic License un GNU General Public License | MS EULA | PHP licence v3.01 | GNU General Public License un Java Community Process |
| Izstrādātājs | Larry Wall | Microsoft | PHP Community | Sun Microsystems |
| Koda saderība dažādās OS | Daļēja | Daļēja | Jā | Jā |
| Koda pieejamība | Atvērta koda | Aizvērta koda | Atvērta koda | Atvērta koda |
| Klienta atbalsts | Jā | Jā | Nē | Jā |
| Programmēšanas valodas | Perl | C#, VB.Net, Jscript | PHP | Java |
| Datu bāze | Jebkura | Jebkura, bet perfekti sader ar Microsoft SQL Server | Jebkura, bet perfekti sader ar MySQL | Jebkura |
| Viegli apgūt | Daļēji | Daļēji | Jā | Nē |
| Rekomendē lielām sistēmām | Nē | Jā | Nē | Jā |
| Tehnoloģijas attīstība | Ļoti ātra, bet tehnoloģija ir piesārņots un nesakārtots | Ļoti ātri | Ar ierobežotu ātrumu | Ļoti ātra, plaši pazīstama pasaulē |
| Atgriezeniska savienojamība | Viegli savienojams, bet dēļ nesakārtotības iespējamas problēmas | Vecāks kods strādās uz jaunām versijām, nebūs nepieciešamas nekādas izmaiņas | Jaunākajam kodam būs nepieciešamas nelielas izmaiņas, lai saderētu ar vecāku | Vecāks kods strādās uz jaunākas versijas bez izmaiņām |
| Koda attīstība | Ātrāka un saprotamāka mazos projektos | Ātra | Ātrāka un efektīvāka mazos projektos | Ātra |

Apskatot un salīdzinot visas četras servera puses programmēšanas tehnoloģijas var secināt:

1. Visas apskatītās tehnoloģijas ir iespējams saistīt ar jebkuru datu bāžu tehnoloģiju. Kā arī tika iegūta informācija ka PHP lieliski sader ar MySQL un ASP ar Microsoft SQL Servers, kas liecina par to, ka PHP varētu būt piemērotākā no tehnoloģijām, jo iepriekšējā sadaļā tika izvēlēta MySQL tehnoloģija.
2. Visvienkāršāk apgūstamā tehnoloģija ir PHP, pēc kuras sarežģītības ziņā grūtāk apgūstamas ir Perl un ASP, kā arī no visām četrām izvēlētajām tehnoloģijām viss sarežģītākā ir Java.
3. Tika noskaidrots, ka gan Perl, gan PHP nav paredzēts lielu sistēmu izstrādei, jo šīs tehnoloģijas pie lielāku sistēmu izstrādes kļūst neefektīvas un grūtāk lietojamas izstrādātājam, bet mazām sistēmām šīs tehnoloģijas sniegs ātrākus rezultātus.
4. Perl ir nobriedusi, standart platformu valoda, bet to ir grūti saprast.
5. ASP ir Microsoft spēcīga programmēšanas valoda, kura ir vairāk bāzēta uz Windows struktūru un var nesaderēt ar visām operētājsistēmām
6. PHP ir standart platformu valoda, kas ir atvērta koda alternatīva ar daudz iebūvētām funkcijām
7. Java ir spēka un elastīguma ziņā galīgā valoda, kura ir piemērotāka nopietniem un pieredzējušiem programmētājiem.

### Izmantotās tehnoloģijas pamatojums

Lai izstrādātu sistēmas servera puses funkcionalitāti autors apskatot un izpētot dažādas tehnoloģijas ir izvēlējies savā darbā izmantot PHP, kā arī lai saskarne lietotājam būtu patīkamāka un vieglāk uztverama apvienot to ar HTML un CSS.

Relatīvi PHP ir jaunpienācējs servera puses attīstības jomā, bet tas ir ieguvis lielu atsaucību no programmētājiem dēļ savas sintakses un visaptverošo funkciju bibliotēku. PHP salīdzinoši vienkāršā sintakse un tas, ka tā ir bezmaksas tehnoloģija ir padarījusi sevi pamanāmāku un pievilcīgāku citu tehnoloģiju lietotājiem, kā piemēram ASP tehnoloģijas piekritējiem. [14]

Starp serveru puses skriptu programmētājiem PHP tiek dēvēts par “Perl slepkavu” . Tādēļ ka tīmekļa bāzētu lietotņu izstrādē ar PHP palīdzību ir iespējams izdarīt gandrīz visu, ko var izdarīt ar Perl palīdzību, un parasti to var izdarīt daudz ātrāk un vienkāršāk. PHP ir ļoti spējīga valoda, ar ļoti daudz iebūvētām funkcijām ir iespējams izdarīt daudz dažādu lietu, to skaitā izsekot lietotāju sesijas, lai radītu dinamiskas grafikas. Nav nepieciešams instalēt modeļus, vai pirkt komerciālas pievienojumprogrammas, viss jau ir integrēts iekš PHP. Šai valodai ir tikai viena negatīva iezīme, tā ir tāda, ka to ir salīdzinoši grūti pievienot nestandarta funkcijas, kas nav tās iebūvētās funkcijas. [14]

PHP ir bezmaksas, standart platformas, atvērtā koda programmatūra. To ir viegli apvienot ar visiem lielākajiem WEB serveriem un pazīstamākajām operētājsistēmām. Viena negatīvā iezīme ir tāda, ka nav oficiāla atbalsta visām problēmām ar kādām var saskarties strādājot ar PHP tehnoloģiju. Tā pat, arī PHP turpina attīstīties, izlaižot jaunas versijas, kurās var saskarties ar dažādām problēmām. Bet neskatoties uz to PHP valodai ir daudz piekritēju, kā arī daudz forumu, kuros ir iespējams atrast atbildes uz viss dažādākajiem jautājumiem. Un pašreizēja PHP versija ir diezgan stabila un uzticama lietotājiem. [14]

Tehnoloģijas priekšrocības:

* Pietiekami viegli apgūt, īpaši izstrādātājiem ar priekšzināšanām programmēšanas valodās C/C++, Java vai Perl.
* Aktīva lietotāju kopiena, kas ir gatava palīdzēt visās PHP saistītās problēmās
* Tā ir bezmaksas
* Viss ir iebūvēt PHP valodā, nav nepieciešamas speciālas papildinājumi

Tehnoloģijas trūkumi:

* Šī valoda ir izstrādāta tā, lai būtu programmētājiem draudzīga, kas diemžēl padara to mazliet mazāk draudzīgu cilvēkiem, kas ikdienā nesaskaras ar programmēšanu

# Mācību centra vadības sistēmas projektēšana

## Mācību centra vadības sistēmas prasības

Pirms “Mācību centra vadības sistēmas” izstrādes ir nepieciešams noskaidrot kādas funkcijas ir jānodrošina jaunajai sistēmai, lai to izdarītu autors konsultējās ar pasūtītāju un apkopoja galvenās sistēmas prasības, kas ir šādas:

1. Jaunajai informāciju sistēmai jānodrošina lietotāju piekļuve izmantojot tikai pārlūkprogrammu, bez lieku failu lejupielādes un papildus programmatūras.
2. Izstrādātajai sistēmai ir jānodrošina trīs dažādu lietotāju piekļuve ar dažādām funkcionālām iespējām, katrai lietotāju grupai.
3. Autorizējoties kā pasniedzējam vai administratoram, ir jādod iespēja izveidot jaunus kursus, pievienot darbiniekus, studentus kursiem, jānodrošina dažāda veida informācijas meklēšana, piemēram, auditorijas, personu meklēšana.
4. Jaunajā sistēma no vizuālā un funkcionālā skata punkta maksimāli jāpielāgo pēc pasūtītāja noteiktā un tai ir jānodrošina viegla saskarne un jābūt “draudzīgai lietotājiem”.
5. Jaunajai sistēmai ir jānodrošina datubāze kurā tiks glabāta informācija par personām, mācību kursiem, grupām un auditorijām.
6. Sistēmai ir jānodrošina iespēja veidot mācību grupas un pievienot studentus tām jebkurā laikā, kā arī veidojot jaunus kursus, jānodrošina iespēja ar kursa mācību dokumentu pievienošanu.
7. Autorizējoties kā administratoram jānodrošina iespēja labot esošo informāciju un izveidot jaunus lietotājus, auditorijas un kursu.
8. Varbūt tev ir vēl kāda ideja, ko šeit var pielikt ……………………………… …………………………………………………………………………………
9. ……………………………………………………………………………………………………………………………………………………………………
10. ……………………………………………………………………………………………………………………………………………………………………

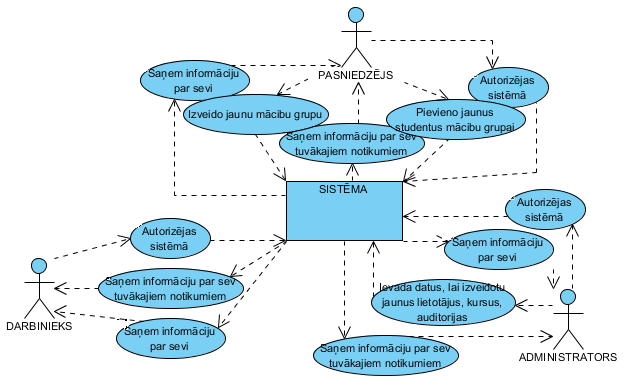
## Lietošanas gadījumu diagramma

“Lietošanas gadījumu pieejas nolūks ir specificēt visus projektējamās IS funkcionālos pakalpojumus, kuri ir nepieciešami tās lietotājiem darbā ar sistēmu, kā arī parādīt, kā atsevišķi sistēmas lietotāji un citas sistēmas mijiedarbojas ar projektējamo IS. Dotā pieeja nodrošina vienkāršu, no tehnoloģijas neatkarīgu veidu sistēmas funkciju atspoguļošanai un to darbību specificēšanai. Izstrādājot lietošanas gadījumu diagrammu ir jāsaprot divi jēdzieni „lietošanas gadījums” un „aktieris”:

* *lietošanas gadījums* – atsevišķs sistēmas funkcionāls pakalpojums, kuru izmanto aktieris, lai sasniegtu noteikto rezultātu. Katrs atsevišķais lietošanas gadījums sevī ietver no viena līdz vairākiem lietošanas scenārijiem (scenārijs ir lietošanas gadījuma specifiskā instance). Scenārijs ir nodefinēto darbību apraksts, kuru izpilda sistēma dialogā ar aktieri un ko izpilda IS lietotājs (jeb aktieris), strādājot ar attiecīgo lietošanas gadījuma scenāriju
* *aktieris* – IS vai tās daļas lietotājs (cilvēks, loma, cita IS vai tehniskā ierīce), kas atrodas ārpus sistēmas funkcionālās robežas un ir iesaistīts lietošanas gadījuma darbībā un mijiedarbojas ar sistēmu caur konkrētu lietošanas gadījumu konkrēta mērķa sasniegšanai. Aktieris ir ārējais elements attiecībā pret IS, bet tas mijiedarbojas ar sistēmu, līdz ar to aktieri definē IS darbības robežas vai sfēru. Tā, analizējot aktierus, var precīzi uzzināt, kas mijiedarbosies ar projektējamo IS. Katrs sistēmas lietošanas gadījums lietošanas gadījumu pieejā jāatbilst konkrēta aktiera mērķiem, un ja nav zināmi visi sistēmas aktieri, tad pastāv risks, ka nav nodefinēti visi nepieciešami lietošanas gadījumi. Tāpēc pareiza aktiera izvēle ir pamats pareizu IS prasību iegūšanai.” [16]

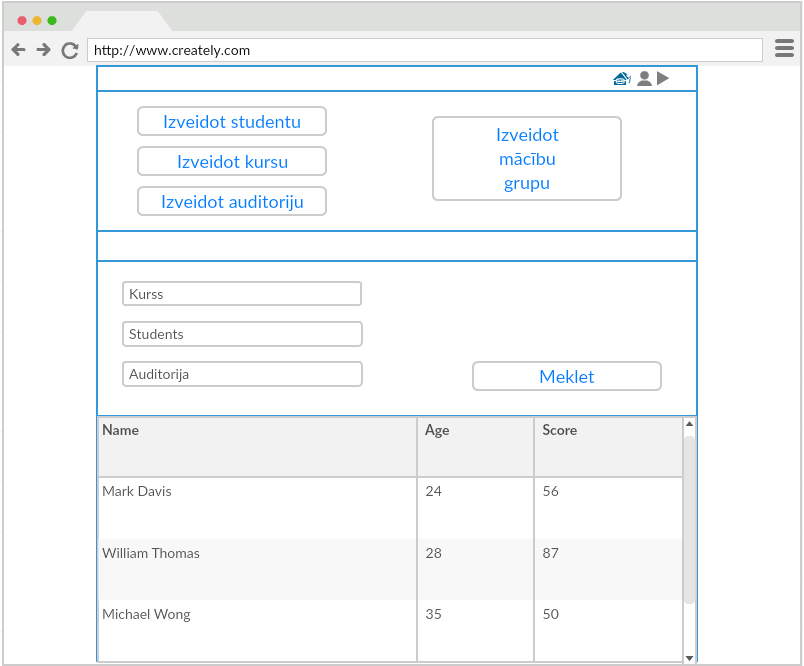
Lietošanas gadījuma diagrammas tiek veidotas izmantojot “Visual Paradigm for UML”. Šo diagrammu izstrādes rīku autors izvēlas, jo ar to jau ir strādājis studiju laikā un tas ir bezmaksas rīks. Rīks piedāvā daudz dažādu iespēju, pastāv iespēja ģenerēt kodu uz dažādām programmēšanas valodām, var veidot dažādas UML diagrammas, sākot ar klašu diagrammu un beidzot ar lietotāju gadījumu diagrammām. Tā pat ir iespējams attēlot datus entītiju saišu diagrammu. Visu diagrammu veidošana lietotājam ir viegli saprotama un ērti lietojama.

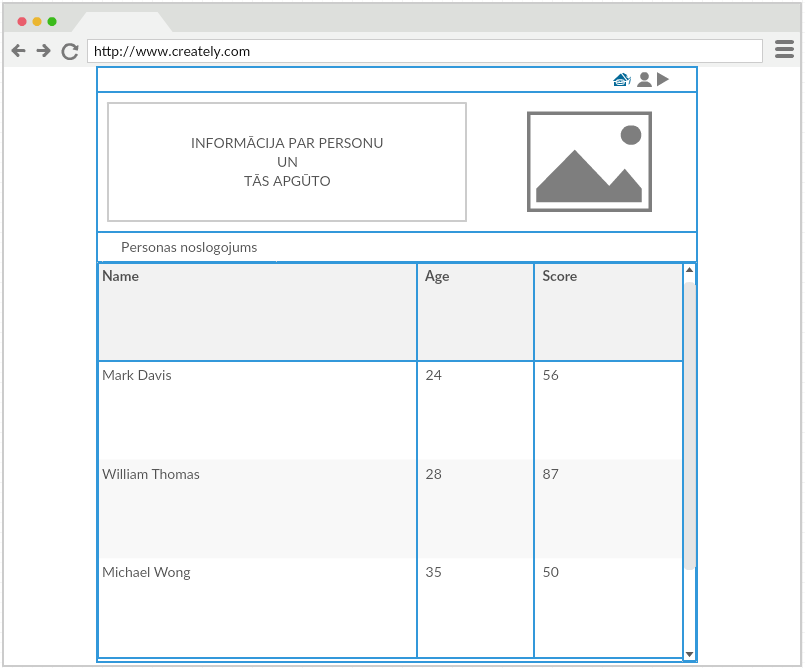
Apskatot izstrādāto lietošanas gadījuma diagrammu (skat. 3.1.att.) ir iespējams redzēt kādas darbības būs iespējams veikt dažādu prioritāšu lietotājiem. Piemēram, darbiniekam būs iespējams tikai autorizēties sistēmā un apskatīt informāciju par sevi un sev tuvāk esošajiem notikumiem.



3.1. att. Lietošanas gadījuma diagramma

## Virsraksts





# Mācību centra vadības sistēmas izstrāde

## MySQL datu bāzes izstrāde

### Sistēmas datu bāze

Pamatojoties uz iepriekšējā sadaļā iegūto informāciju, kas ir sistēmas prasības un pārskatot lietošanas gadījumu diagrammas, autors ir izvēlējies relāciju datu bāzes modeli datu glabāšanai. Sistēma tiek izstrādāta no pašiem pamatiem tāpēc tās datubāze pagaidām sastāv tikai no deviņām tabulām, bet strādājot ar to ilglaicīgi, visticamāk nāksies pievienot klāt vēl citas tabulas (skat. 4.1.att.). Datu bāzes tabulas var sadalīt divās daļās, kas savstarpēji ir saistītas.

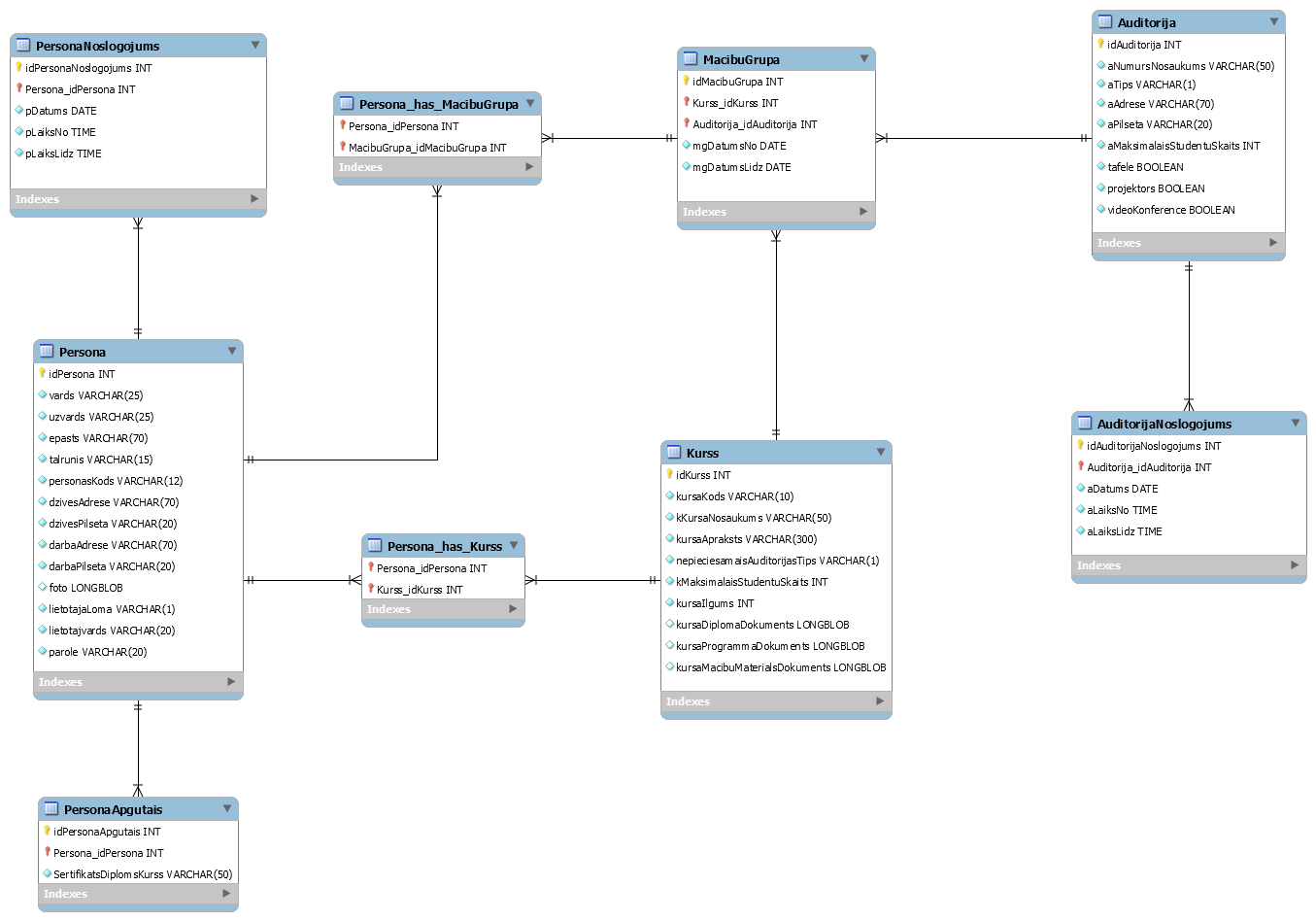
Pirmā daļa, glabā informāciju par sistēmas lietotājiem, gan administratoriem, gan darbiniekiem un tajā ietilpst trim tabulas:

* *PersonaNoslogojums*
* *Persona*
* *PersonaApgutais*

Turklāt otrā daļa glabā informāciju par mācību grupu, kādi ir kursi ko apgūt kādās telpās tas ir iespējams un šī daļa sastāv no četrām tabulām:

* *MacibuGrupa*
* *Kurss*
* *Auditorija*
* *AuditorijaNoslogojums*

Apskatot relāciju datu bāzes modeli (skat. 4.1.att.) varam redzēt, ka visas saites starp tabulām ir viens pret daudziem. Tabulām *PersonaNoslogojums* un *PersonaApgutais* ar tabulu Persona veidojas šāda saite, jo viena persona var būt apguvusi vairākas lietas un arī vienai personai pie noslogojuma var būt vairāki darāmi darbi. Tā pat arī tabulai *MacibuGrupa* ir divas citas saites ar tabulām, Kurss un Auditorija, jo vienai mācību grupai var attiecīgi piesaistīt vienu auditoriju un vienu apgūstamo kursu. Vēl šāda saite veidojas ar tabulām Auditorija un *AuditorijaNoslogojums*, jo viena auditorija var būt aizņemta dažādos datumos un laikos. Kā arī apskatot šo datu bāzes modeli ir viegli novērot, ka ir izveidotas divas starp tabulas, kur veidojās saite daudzi pret daudziem un arī šīs divas starp tabulas ir atlikušās divas datu bāzes modeļa tabulas, kas ir *Persona*\_*has*\_*MacibuGrupa* un *Persona*\_*has*\_*Kurss*. Šīs tabulas ir nepieciešamas, jo viena persona var mācīties vairākus kursus un vienu un to pašu kursu var apgūt vairākas personas, kā arī viena persona var būt vairākās mācību grupās un vienā mācību grupā noteikti būs vairākas personas.



.. att. Relāciju datu bāzes modelis

(Varbūt vajag atsevišķā horizontālā lapā?)

### Sistēmas datu bāzes tabulas

Visas datu bāzes tabulas ir apskatāmas 4.1.attēlā, tālāk šajā sadaļā tiks aprakstītas visas tabulas un paskaidrota to nepieciešamība sistēmā:

* *Persona* ir datu bāzes tabula, kurai ir svarīga loma sistēmā, tā glabā informāciju, gan par darbiniekiem, gan par administratoriem un bez šīs tabulas nebūtu iespējama autorizācija sistēmā. Kā redzams relāciju datubāzes modelī, katrai personai tiek automātiski piešķirts unikāls identifikators, kas ir *idPersona* ar datu tipu INT (veseli skaitļi). Reģistrējot jaunu lietotāju ir jānorāda lietotāja vārds un uzvārds, kas glabājas laukos ar nosaukumiem *vards* un *uzvards,* abu šo lauku datu tipi ir VARCHAR (simbolu virkne) un ir ierobežoti līdz 25 rakstzīmēm. Tālāk ir nepieciešams norādīt lietotāja e-pastu, tālruni un personas kodu, kas glabājas zem laukiem *epasts*, *talrunis*, *parsonasKods*, visiem šiem laukiem datu tips ir uzstādīts kā VARCHAR ar ierobežotiem rakstzīmju garumiem, attiecīgi e-pastam 70 rakstzīmes, tālrunim 15 un personas kodam 12. Papildus reģistrējot jaunu lietotāju nepieciešams norādīt dzīves vietas un darba vietas adreses, kuru datu tips arī ir VARCHAR un to garums ir ierobežots līdz 70 rakstzīmēm, kā arī ir jānorāda gan dzīvesvietas gan darba vietas pilsēta kuru datu tips ir VARCHAR un ir ierobežots līdz 20 rakstzīmēm.

Visi lauki ir obligāti aizpildāmi, izņemot vienu, kas ir lauks *foto* kurā ir nepieciešams pievienot lietotāja pašportreta attēlu, šī lauka datu tips ir LONGBLOB (liela izmēru faili) un ja attēls netiek pievienots tad lauks tiek aizpildīts ar noklusējuma vērtību NULL.

Svarīgākā loma ir pēdējiem tabulas trim laukiem, kas ir *lietotajaLoma*, *lietotajvards* un *parole*. Lauks *lietotajaLoma* ir ar datu tipu VARCHAR un atbalsta tikai viena simbola ievadi, kas nosaka vai lietotājs ir pasniedzējs, darbinieks vai administrators. Pēdējie divi lauki ir priekš lietotājvārda un paroles, to abu datu tipi ir VARCHAR un ir ierobežoti līdz 20 rakstzīmēm.

* *PersonaNoslogojums* datubāzes tabula satur tikai četrus laukus, šī tabula glabā informāciju par to kura persona ir kādos datumos ir aizņemta. Pirmie divi unikālie tabulas lauki ir unikālie identifikatori, *idPersonaNoslogojums* ir šīs tabulas identifikators, bet *Persona\_idPersona* ir ārējā atslēga no tabulas Persona, abu šo lauku datu tipi ir INT, kas palīdz saistīt šo tabulu ar tabulu *Persona.* Kā ar šī tabu satur datuma lauku, *pDatums*, kura datu tips ir DATE un divus laika laukus, kas ir *pLaiksNo* un *pLaiksLīdz*, kuru datu tipi ir TIME un kas norāda no cikiem līdz cikiem persona konkrētajā datumā ir aizņemta.
* *PersonaApgutais* ir datu bāzes tabula kas satur tikai trīs laukus, divi no tiem tā pat kā iepriekšējā aprakstītajā tabulā ir unikālie identifikatori viens ir šīs tabulas identifikators, kas ir *idPersonaApgūtais* un otrs ir ārējā atslēga no tabulas *Persona* un šis identifikators ir *Persona*\_*idPersona*, abu lauku datu tipi ir INT. Trešais šīs tabulas lauks ir *SertifikātsDiplomsKurss*, kurā tiek glabāta informācija ko darbinieks ir apguvis un šī lauka datu tips ir VARCHAR un tas ir ierobežots līdz 50 rakstzīmēm.
* *MacibuGrupa* šī tabula ir tik pat nozīmīga kā tabula *Persona*, jo tā sevī glabā svarīgāko informāciju par mācību grupu, kuru veidošana ir sistēmas galvenais uzdevums. Šī datubāzes tabula sevī ietver sešus laukus no kuriem pirmais, *idMacibuGrupa*, ir unikālais identifikators un nākamie divi lauki ir ārējās atslēgas no tabulas *Kurss* un tabulas *Auditorija*, lai noteiktu, ko apgūs šīs mācību grupas darbinieki un kurā auditorijā tas notiks. Tālāk ir obligāti aizpildāmie lauki, kas ir *mGrupasNosaukums*, kura datu tips ir VARCHAR un ir ierobežots līdz 45 rakstzīmēm, kā arī pēdējie divi lauki *mgDatumsNo*, *mgDatumsLidz*, kuru datu tips ir DATE un ir nepieciešami, lai attēlotu cik ilgas būs mācības.
* *Kurss* ir datu bāzes tabula, kurā tiek glabāta visa nozīmīgākā informācija par apgūstamajiem kursiem, šī tabula sastāv no desmit laukiem. Pirmais lauks, kā jau visās iepriekšējās tabulās arī ir unikālā identifikatora lauks un tas ir *idKurss*, kura datu tips ir INT. Sistēmā kursu var atpazīt divejādi pēc tā nosaukuma vai koda, tāpēc arī datu bāze ir divi VARCHAR ierobežotu rakstzīmju datu tipa lauki *kursaKods* un *kKursaNosaukums*. Tālāk seko lauks *kursaApraksts* kura datu tips ir VARCHAR un ir ierobežots līdz 300 rakstzīmēm, paredzēts, lai glabātu īsu aprakstu par to kas kursā tiks apgūts. Nākamie trīs lauki glabā vispārīgāku informāciju, kas ir kāda auditorija būs nepieciešama (*nepieciesamaisAuditorijasTips*, VARCHAR), maksimālais studentu skaits kursā (*kMaksimālaisStudentuSkaits*, INT) un cik ilgs būs kurss, skaitot nedēļas (*kursaIlgums*, INT). Pēdējo trīs lauku datu tips ir LONGBLOB un ir paredzēti dažādu dokumentu glabāšanai, šie lauki ir *kursaDiplomaDokuments*, *kursaProgrammaDokuments*, *kursaMacibuMaterialsDokuments*.
* *Auditorija* ir datu bāzes tabula kurā tiek glabāta visa nozīmīgākā informācija par esošajām auditorijām, telpām kurās var notikt mācības, šī tabula sastāv no deviņiem laukiem. Tā pat kā iepriekšējās tabulās pirmais lauks ir primārā atslēga, jeb unikālais identifikators, tas ir *idAuditorija* un tā datu tips ir INT. Lai vieglāk atpazītu auditoriju tai ir nepieciešams norādīt nosaukumu vai telpas numuru, kas glabājas laukā *aNumursNosaukums* un ir ar datu tipu VARCHAR un ierobežots līdz 50 rakstzīmēm. Vēl ir nepieciešams norādīt auditorijas adresi un pilsētu (*aAdrese*, ar datu tipu VARCHAR un ierobežots līdz 70 rakstzīmēm, *aPilsēta* ar datu tipu VARCHAR un ierobežots līdz 20 rakstzīmēm). Atlikušie četri lauki ir vienas rakstzīmes lauki un ir nepieciešami lai noteiktu vai auditorija ir dator auditorija vai nē (*aTips*), vai auditorijā ir pieejams projektors (projektors), tāfele (*tafele*) un vai ir iespējama video konference (*videoKonferene*).
* *AuditorijasNoslogojums* datu bāzes tabula ir ļoti līdzīga tabulai *PersonaNoslogojums*, tajā tiek glabāta informācija par to kuros datumos un no cikiem līdz cikiem ir aizņemta auditorija, lai to nodrošinātu tabula satur piecus laukus. Pirmais ir unikālais identifikators *idAuditorijaNoslogojums*, kura datu tips ir INT un otrais lauks ir ārējā atslēga no tabulas Auditorija, kas nosaka kura auditorija būs aizņemta. Nākamais lauks ir *aDatums*, kura datu tips ir DATE un pēdējie divi lauki ir *aLaiksNo* un *aLaiksLidz*, kuru datu tips ir TIME.
* *Persona*\_*has*\_*MacibuGrupa* datu bāzes tabula ir starp tabula, kas veidojas starp tabulām *Persona* un *MacibuGrupa*, tā satur trīs obligāti aizpildāmos laukus, divi no tiem ir ārējās atslēgas no iepriekš minētajām tabulām un pēdējais lauks ir *vaiIrPasniedzējs*, kas norādīs vai mācību grupai pievienotā persona ir pasniedzējs vai nē, šim laukam datu tips ir VARCHAR un ir ierobežots ar vienu simbolu.
* *Persona*\_*has*\_*Kurss* datu bāzes tabula ir starp tabula, kas veidojas starp tabulām *Persona* un *Kurss*, šī tabula satur tikai divus laukus un abi šie lauki ir ārējās atslēgas no iepriekš minētajām tabulām,

## Saskarnes izstrāde

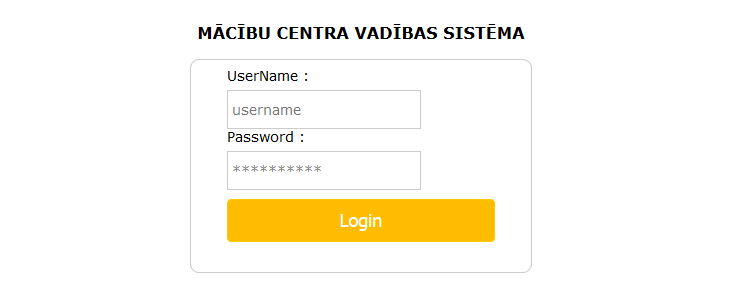
(Šo vēl nelasi un nemaz nemēģini labot)

(Tad man vēl ir no krista tāda testpiemēru tabula un arī tabula kur mēs rakstījām ko labot, taka labojumu un papildinājumu dokuments, kur man tos likt klāt, šijā sadaļā pie is izstrādes vai tie kaut kā iet pielikumā?)

Pamatojoties uz iepriekšējā sadaļā iegūto informāciju, par sistēmas prasībām un apkopojot to, jaunās sistēmas nepieciešamās funkcijas var sadalīt četrās lielās grupās:

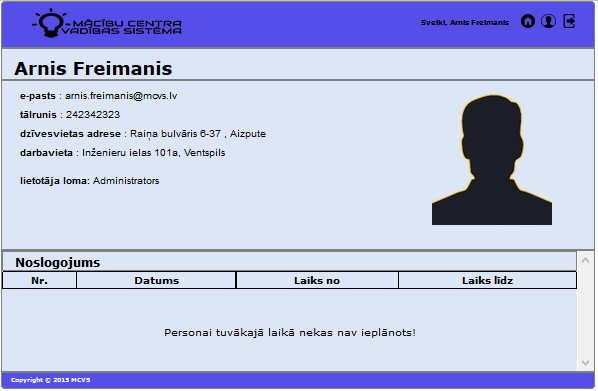
1. Autorizācija un profila lapa
2. Jaunas auditorijas, personas un kursa pievienošana
3. Mācību grupas plānošana
4. Dažādas informācijas meklēšana

### Autorizācija un profila lapa



6.1 att. Lietotnes autorizācijas logi

Pēc autorizācijas uzņēmuma darbinieks, ne sistēmas administrators nonāk savā profila lapā, kurā var izlasīt informāciju par sevi un par sev tuvāk ieplānotajām mācībā, bet protams arī administratoram ir pieejama šāda lapa, tikai viņam nākas spiest uz ikonu labajā augšējajā stūrī, cilvēka silueta izskatā. Lietotājam ir ļoti ierobežotas tiesības un tas ir arī viss ko viņš var apskatīt sistēmā.



6.2 att.Lietotāja un Administratora profila lapa

## Testēšana un kļūdu dokumentēšana

Sistēmas izstrādes laikā nākas saskarties ar dažādām kļūdām, vai nonākt pie jaunām idejām ko un kā vajadzētu uzlabot, tāpēc izstrādājot MCVS tika izveidots dokuments, jeb tabula, kurā tika reģistrēti visi labojumi un nepieciešamie papildinājumi. Šī dokumenta galvenais nolūks bija, lai varētu sekot līdzi mazsvarīgākiem darbiem, tādiem, ko ir iespējams aizmirst un kā dēļ sistēma būtu nepilnīga (skatīt 2. pielikumu).

Līdzīgi, kā ar kļūdu reģistrēšanu, arī tika izveidots dokuments testēšanai, kurā tika izveidoti visām sistēmas situācijām testpiemēri un veicamās pārbaudes. Šādi testpiemēri ir nepieciešami, lai izpētītu sistēmas funkcionalitāti un noteiktu, kādas problēmas ir dažādos procesos. Sistēmas testēšana parasti ir nepieciešama pirms un pēc tam, kad sistēma ir ieviesta. Šo procesu parasti veic testētāji un tas viens no ūdenskrituma modeļa posmiem. Testētāji parasti cenšas “salauzt sistēmu”, ievadot datus, kas var izraisīt sistēmas darbības traucējumus vai atgriezt nepareizu informāciju. Gan šo pozitīvo, gan negatīvo rezultātu notestēšana soli pa soli ir aprakstīta testēšanas tabulā (skatīt 1. pielikumu), kā arī autors pats ir veicis visus testpiemērus, lai pārliecinātos par sistēmas funkcionalitāti un tie visi ir bijuši veiksmīgi, kas liecina par to, ka sistēma strādā, bez kļūdām.

# Secinājumi un priekšlikumi

# Izmantotās literatūras un avotu saraksts

# Bibliogrāfija

|  |  |
| --- | --- |
| [1] | P. Zandbergen, «http://study.com,» [Tiešsaiste]. Available: http://study.com/academy/lesson/what-are-information-systems-definition-types-quiz.html. |
| [2] | «http://www.britannica.com/topic/information-system,» [Tiešsaiste]. |
| [3] | I. Solovjovs, «http://estudijas.lu.lv/pluginfile.php/234612/mod\_resource/content/1/II%20semestris/2.IS\_dzives\_cikli.htm,» [Tiešsaiste]. |
| [4] | «http://revisionworld.com/gcse-revision/ict/system-life-cycle,» [Tiešsaiste]. |
| [5] | «http://istqbexamcertification.com/what-are-the-software-development-models/,» [Tiešsaiste]. |
| [6] | «http://estudijas.lu.lv/file.php/3582/I%20semestris/Kospekti/1.Iss%20parskats%20par%20kursu.htm,» [Tiešsaiste]. |
| [7] | «http://www.tutorialspoint.com/sdlc/sdlc\_agile\_model.htm,» [Tiešsaiste]. |
| [8] | «http://www.uzdevumi.lv/p/informatika/10-klase/datu-bazu-veidosana-un-lietosana-11093/re-b679a107-97b0-445c-be2e-3a1574bee6f4,» [Tiešsaiste]. |
| [9] | «http://db-engines.com/en/ranking,» [Tiešsaiste]. |
| [10] | «http://searchitchannel.techtarget.com/feature/MySQL-alternatives,» [Tiešsaiste]. |
| [11] | «https://www.novell.com/documentation/nw65/web\_mysql\_nw/data/aj5bj52.html,» [Tiešsaiste]. |
| [12] | «http://raddyx.com/mysql,» [Tiešsaiste]. |
| [13] | «http://www.codeproject.com/Articles/102854/PHP-and-ASP-NET-A-Feature-List,» [Tiešsaiste]. |
| [14] | «https://www.quora.com/What-are-the-alternatives-of-PHP-if-theres-any,» [Tiešsaiste]. |
| [15] | «http://www.tutorialspoint.com/perl/perl\_introduction.htm,» [Tiešsaiste]. |
| [16] | «http://stpk.cs.rtu.lv/sites/all/files/stpk/materiali/ISM/MetodologijaISP/LGOP.htm,» [Tiešsaiste]. |
| [17] | «http://www.exigenservices.lv/,» http://www.exigenservices.lv/pakalpojumi-un-risinajumi/sistemu-izstrade. [Tiešsaiste]. |
| [18] | L. Z. akadēmija, «http://termini.lza.lv/term.php?term=programmat%C5%ABras%20dz%C4%ABves%20cikls&list=dz%C4%ABves%20cikls&lang=LV,» [Tiešsaiste]. |
| [19] | «http://revisionworld.com/gcse-revision/ict/system-life-cycle,» [Tiešsaiste]. |
| [20] | «http://estudijas.lu.lv/file.php/3582/I%20semestris/Kospekti/1.Iss%20parskats%20par%20kursu.htm,» [Tiešsaiste]. |

# Pielikumi

*Pielikums Nr. 1*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr.** | **ID** | **Nosaukums** | **IS lapa** | **Izpildes soļi** | **Sagaidāmais rezultāts** | **Komentāri** |
| 1 | 1.1. | Autorizēšanās sistēmā | index.php | 1. Jāatver sistēmas sākuma lapa 2. Jāievada korekts lietotājvārds 3. Paroles lauks jāatstāj tukšs 4. Jānospiež poga "Ienākt sistēmā" | Lietotājs paliek index.php lapā, kurā parādās ziņojums "Nav ievadīts lietotājvārds un / vai parole". | Testpiemērs veiksmīgs. |
| 2 | 1.2. | Autorizēšanās sistēmā | index.php | 1. Jāatver sistēmas sākuma lapa 2. Lietotājvārda lauks jāatstāj tukšs 3. Jāievada korekta parole 4. Jānospiež poga "Ienākt sistēmā" | Lietotājs paliek index.php lapā, kurā parādās ziņojums "Nav ievadīts lietotājvārds un / vai parole". | Testpiemērs veiksmīgs. |
| 3 | 1.3. | Autorizēšanās sistēmā | index.php | 1. Jāatver sistēmas sākuma lapa 2. Lietotājvārda lauks jāatstāj tukšs 3. Paroles lauks jāatstāj tukšs 4. Jānospiež poga "Ienākt sistēmā" | Lietotājs paliek index.php lapā, kurā parādās ziņojums "Nav ievadīts lietotājvārds un / vai parole". | Testpiemērs veiksmīgs. |
| 4 | 1.4. | Autorizēšanās sistēmā | index.php | 1. Jāatver sistēmas sākuma lapa 2. Jāievada korekts lietotājvārds 3. Jāievada nekorekta parole 4. Jānospiež poga "Ienākt sistēmā" | Lietotājs paliek index.php lapā, kurā parādās ziņojums "Nepareizi ievadīts lietotājvārds un / vai parole". | Testpiemērs veiksmīgs. |
| 5 | 1.5. | Autorizēšanās sistēmā | index.php | 1. Jāatver sistēmas sākuma lapa 2. Jāievada nekorekts lietotājvārds 3. Jāievada korekta parole 4. Jānospiež poga "Ienākt sistēmā" | Lietotājs paliek index.php lapā, kurā parādās ziņojums "Nepareizi ievadīts lietotājvārds un / vai parole". | Testpiemērs veiksmīgs. |
| 6 | 1.6. | Autorizēšanās sistēmā | index.php | 1. Jāatver sistēmas sākuma lapa 2. Jāievada korekts lietotājvārds 3. Jāievada korekta parole 4. Jānospiež poga "Ienākt sistēmā" | Lietotājs tiek pārvirzīts uz main.php lapu. | Testpiemērs veiksmīgs. |
| 7 | 2.1. | Navigācija uz galveno lapu | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenās lapas galvenē jānospiež poga "Saite uz galveno lapu" | Lietotājs tiek pārvirzīts uz main.php lapu. | Testpiemērs veiksmīgs. |
| 8 | 2.2. | Navigācija uz profila lapu | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenās lapas galvenē jānospiež poga "Saite uz profila lapu" | Lietotājs tiek pārvirzīts uz user-page.php lapu. | Testpiemērs veiksmīgs. |
| 9 | 2.3. | Izlogošanās no sistēmas | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenās lapas galvenē jānospiež poga "Izlogoties no sistēmas" | Lietotājs tiek pārvirzīts uz index.php lapu. | Testpiemērs veiksmīgs. |
| 10 | 2.4. | Navigācija uz jauna lietotāja pievienošanas lapu | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu lietotāju" | Lietotājs tiek pārvirzīts uz newUser.php lapu. | Testpiemērs veiksmīgs. |
| 11 | 2.5. | Navigācija uz jauna kursa pievienošanas lapu | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu kursu" | Lietotājs tiek pārvirzīts uz newCourse.php lapu. | Testpiemērs veiksmīgs. |
| 12 | 2.6. | Navigācija uz jaunas auditorijas pievienošanas lapu | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu auditoriju" | Lietotājs tiek pārvirzīts uz newRoom.php lapu. | Testpiemērs veiksmīgs. |
| 13 | 2.7. | Navigācija uz jaunas mācību grupas plānošanas lapu | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Izveidot jaunu mācību grupu" | Lietotājs tiek pārvirzīts uz groupPlanning.php lapu. | Testpiemērs veiksmīgs. |
| 14 | 2.8. | Navigācija uz pievienot studentus mācību grupai lapu | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot studentus mācību grupai" | Lietotājs tiek pārvirzīts uz addStudent.php lapu. | Testpiemērs veiksmīgs. |
| 15 | 2.9. | Lietotāja meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana neaizpildot nevienu no meklēšanas laukiem 4. Jānospiež poga "Meklēt lietotāju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 16 | 2.10. | Lietotāja meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana, aizpildot lietotāja vārda lauku ar informāciju, kas sistēmas datubāzē nav atrodama 4. Jānospiež poga "Meklēt lietotāju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 17 | 2.11. | Lietotāja meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana, aizpildot lietotāja uzvārda lauku ar informāciju, kas sistēmas datubāzē nav atrodama 4. Jānospiež poga "Meklēt lietotāju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 18 | 2.12. | Lietotāja meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana, aizpildot lietotāja personas koda lauku ar informāciju, kas sistēmas datubāzē nav atrodama 4. Jānospiež poga "Meklēt lietotāju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 19 | 2.13. | Lietotāja meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana, aizpildot vismaz 2 lietotāja meklēšanas laukus ar informāciju, kas sistēmas datubāzē nav atrodama 4. Jānospiež poga "Meklēt lietotāju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 20 | 2.14. | Lietotāja meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana, aizpildot lietotāja vārda meklēšanas lauku ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt lietotāju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar atrastajiem meklēšanas rezultātiem | Testpiemērs veiksmīgs. |
| 21 | 2.15. | Lietotāja meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana, aizpildot lietotāja uzvārda meklēšanas lauku ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt lietotāju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar atrastajiem meklēšanas rezultātiem | Testpiemērs veiksmīgs. |
| 22 | 2.16. | Lietotāja meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana, aizpildot lietotāja personas koda meklēšanas lauku ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt lietotāju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar atrastajiem meklēšanas rezultātiem | Testpiemērs veiksmīgs. |
| 23 | 2.17. | Lietotāja meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana, aizpildot vismaz 2 no lietotāja meklēšanas laukiem ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt lietotāju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar atrastajiem meklēšanas rezultātiem | Testpiemērs veiksmīgs. |
| 24 | 2.18. | Kursa meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic kursa meklēšana neaizpildot nevienu no meklēšanas laukiem 4. Jānospiež poga "Meklēt kursu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 25 | 2.19. | Kursa meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic kursa meklēšana, aizpildot kursa koda lauku ar informāciju, kas sistēmas datubāzē nav atrodama 4. Jānospiež poga "Meklēt kursu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 26 | 2.20. | Kursa meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic kursa meklēšana, aizpildot kursa nosaukuma lauku ar informāciju, kas sistēmas datubāzē nav atrodama 4. Jānospiež poga "Meklēt kursu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 27 | 2.21. | Kursa meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic kursa meklēšana, aizpildot abus kursa meklēšanas laukus ar informāciju, kas sistēmas datubāzē nav atrodama 4. Jānospiež poga "Meklēt kursu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 28 | 2.22. | Kursa meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic kursa meklēšana, aizpildot kursa koda lauku ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt kursu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar atrastajiem meklēšanas rezultātiem | Testpiemērs veiksmīgs. |
| 29 | 2.23. | Kursa meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic kursa meklēšana, aizpildot kursa nosaukuma lauku ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt kursu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar atrastajiem meklēšanas rezultātiem | Testpiemērs veiksmīgs. |
| 30 | 2.24. | Kursa meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic kursa meklēšana, aizpildot abus kursa meklēšanas laukus ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt kursu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar atrastajiem meklēšanas rezultātiem | Testpiemērs veiksmīgs. |
| 31 | 2.25. | Auditorijas meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic auditorijas meklēšana neaizpildot nevienu no meklēšanas laukiem 4. Jānospiež poga "Meklēt auditoriju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 32 | 2.26. | Auditorijas meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic auditorijas meklēšana, aizpildot auditorijas numura vai nosaukuma lauku ar informāciju, kas sistēmas datubāzē nav atrodama 4. Jānospiež poga "Meklēt auditoriju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 33 | 2.27. | Auditorijas meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic auditorijas meklēšana, aizpildot auditorijas numura vai nosaukuma lauku ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt auditoriju" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar atrastajiem meklēšanas rezultātiem | Testpiemērs veiksmīgs. |
| 34 | 2.28. | Mācību grupas meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic mācību grupas meklēšana neaizpildot nevienu no meklēšanas laukiem 4. Jānospiež poga "Meklēt mācību grupu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 35 | 2.29. | Mācību grupas meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic mācību grupas meklēšana, aizpildot mācību grupas nosaukuma lauku ar informāciju, kas sistēmas datubāzē nav atrodama 4. Jānospiež poga "Meklēt mācību grupu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 36 | 2.30. | Mācību grupas meklēšana | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic mācību grupas meklēšana, aizpildot mācību grupas nosaukuma lauku ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt mācību grupu" | Zem meklēšanas laukiem tiek izvadīta rezultātu tabula ar atrastajiem meklēšanas rezultātiem | Testpiemērs veiksmīgs. |
| 37 | 2.31. | Lietotāja datu apskatīšana pēc to meklēšanas | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic lietotāja meklēšana, aizpildot lietotāja meklēšanas lauku (-us) ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt lietotāju" 5. Jāizvēlas rezultātu tabulā kādu no lietotājiem nospiežot pogu "Dati", kas atrodas kolonnā "Uz profila lapu" | Tiek atvērta sistēmas lietotāju profila lapa ar konkrētā lietotāja datiem | Testpiemērs veiksmīgs. |
| 38 | 2.32. | Kursa datu apskatīšana pēc to meklēšanas | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic kursa meklēšana, aizpildot kursa meklēšanas lauku (-us) ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt kursu" 5. Jāizvēlas rezultātu tabulā kādu no kursiem nospiežot pogu "Dati", kas atrodas kolonnā "Uz kursa lapu" | Tiek atvērta sistēmas kursa lapa ar konkrētā mācību kursa datiem | Testpiemērs veiksmīgs. |
| 39 | 2.33. | Auditorijas datu apskatīšana pēc to meklēšanas | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic auditorijas meklēšana, aizpildot auditorijas meklēšanas lauku ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt Auditoriju" 5. Jāizvēlas rezultātu tabulā kādu no auditorijām nospiežot pogu "Dati", kas atrodas kolonnā "Uz auditorijas lapu" | Tiek atvērta sistēmas auditorijas lapa ar konkrētās auditorijas datiem | Testpiemērs veiksmīgs. |
| 40 | 2.34. | Mācību grupas datu apskatīšana pēc to meklēšanas | main.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jāveic mācību grupas meklēšana, aizpildot mācību kursa meklēšanas lauku ar informāciju, kas sistēmas datubāzē ir atrodama 4. Jānospiež poga "Meklēt mācību grupu" 5. Jāizvēlas rezultātu tabulā kādu no mācību grupām nospiežot pogu "Dati", kas atrodas kolonnā "Uz mācību grupas lapu" | Tiek atvērta sistēmas mācību grupas lapa ar konkrētās mācību grupas datiem | Testpiemērs veiksmīgs. |
| 41 | 3.1. | Jauna lietotāja pievienošana | newUser.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu lietotāju" 4. Jāatstāj visi aizpildāmie lietotāja lauki tukši 5. Jānospiež poga "Izveidot lietotāju" | Jauns lietotājs datubāzē netiek pievienots. Sistēma attēlo to pašu lapu newUser.php un tās apakšā informatīvu tekstu "Kāds no aizpildāmajiem laukiem nav aizpildīts!" | Testpiemērs veiksmīgs. |
| 42 | 3.2. | Jauna lietotāja pievienošana | newUser.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu lietotāju" 4. Jāaizpilda daži, bet ne visi no obligāti aizpildāmajiem laukiem 5. Jānospiež poga "Izveidot lietotāju" | Jauns lietotājs datubāzē netiek pievienots. Sistēma attēlo to pašu lapu newUser.php un tās apakšā informatīvu tekstu "Kāds no aizpildāmajiem laukiem nav aizpildīts!" | Testpiemērs veiksmīgs. |
| 43 | 3.3. | Jauna lietotāja pievienošana | newUser.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu lietotāju" 4. Jāaizpilda visi aizpildāmie lauki ar korektu lietotāja informāciju 5. Jānospiež poga "Izveidot lietotāju" | Jauns lietotājs datubāzē tiek pievienots. Sistēma attēlo to pašu lapu newUser.php un tās apakšā informatīvu tekstu "Persona veiksmīgi pievienota datubāzei!" | Testpiemērs veiksmīgs. |
| 44 | 4.1. | Jauna kursa pievienošana | newCourse .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu kursu" 4. Jāatstāj visi aizpildāmie kursa lauki tukši 5. Jānospiež poga "Izveidot kursu" | Jauns kurss datubāzē netiek pievienots. Sistēma attēlo to pašu lapu newCourse.php un tās apakšā informatīvu tekstu "Kāds no aizpildāmajiem laukiem nav aizpildīts!" | Testpiemērs veiksmīgs. |
| 45 | 4.2. | Jauna kursa pievienošana | newCourse .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu kursu" 4. Jāaizpilda daži, bet ne visi no obligāti aizpildāmajiem laukiem 5. Jānospiež poga "Izveidot kursu" | Jauns kurss datubāzē netiek pievienots. Sistēma attēlo to pašu lapu newCourse.php un tās apakšā informatīvu tekstu "Kāds no aizpildāmajiem laukiem nav aizpildīts!" | Testpiemērs veiksmīgs. |
| 46 | 4.3. | Jauna kursa pievienošana | newCourse .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu kursu" 4. Jāaizpilda visi aizpildāmie lauki ar korektu kursa informāciju 5. Jānospiež poga "Izveidot kursu" | Jauns kurss datubāzē tiek pievienots. Sistēma attēlo to pašu lapu newCourse.php un tās apakšā informatīvu tekstu "Kurss veiksmīgi pievienots datubāzei!" | Testpiemērs veiksmīgs. |
| 47 | 5.1. | Jaunas auditorijas pievienošana | newRoom.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu auditoriju" 4. Jāatstāj visi aizpildāmie auditorijas lauki tukši 5. Jānospiež poga "Izveidot auditoriju" | Jauna auditorija datubāzē netiek pievienota. Sistēma attēlo to pašu lapu newRoom.php un tās apakšā informatīvu tekstu "Kāds no aizpildāmajiem laukiem nav aizpildīts!" | Testpiemērs veiksmīgs. |
| 48 | 5.2. | Jaunas auditorijas pievienošana | newRoom.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu auditoriju" 4. Jāaizpilda daži, bet ne visi no obligāti aizpildāmajiem laukiem 5. Jānospiež poga "Izveidot auditoriju" | Jauna auditorija datubāzē netiek pievienota. Sistēma attēlo to pašu lapu newRoom.php un tās apakšā informatīvu tekstu "Kāds no aizpildāmajiem laukiem nav aizpildīts!" | Testpiemērs veiksmīgs. |
| 49 | 5.3. | Jaunas auditorijas pievienošana | newRoom.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot jaunu auditoriju" 4. Jāaizpilda visi aizpildāmie lauki ar korektu kursa informāciju 5. Jānospiež poga "Izveidot auditoriju" | Jauna auditorija datubāzē tiek pievienota. Sistēma attēlo to pašu lapu newRoom.php un tās apakšā informatīvu tekstu "Auditorija veiksmīgi pievienota datubāzei!" | Testpiemērs veiksmīgs. |
| 50 | 6.1. | Jaunas mācību grupas pievienošana | groupPlanning.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Izveidot jaunu mācību grupu" 4. Jāizvēlas kāds no piedāvātajiem mācību kursiem un jānospiež poga "Apstiprināt" | Zemāk esošajos lapas sarakstos "pasniedzēji" un "auditorijas" tiek ielasīta informācija no datubāzes. Izvēlētais mācību kurss tiek atzīmēts kā izvēlēts | Testpiemērs veiksmīgs. |
| 51 | 6.2. | Jaunas mācību grupas pievienošana | groupPlanning.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Izveidot jaunu mācību grupu" 4. Jāizvēlas kāds no piedāvātajiem mācību kursiem un jānospiež poga "Apstiprināt" 5. Jāizvēlas kāds no piedāvātājiem pasniedzējiem un jānospiež poga "Apstiprināt" | Iepriekš apstiprinātās lauku vērtības paliek nemainīgas. Papildus tiek apstiprināts izvēlētais mācību grupas pasniedzējs | Testpiemērs veiksmīgs. |
| 52 | 6.3. | Jaunas mācību grupas pievienošana | groupPlanning.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Izveidot jaunu mācību grupu" 4. Jāizvēlas kāds no piedāvātajiem mācību kursiem un jānospiež poga "Apstiprināt" 5. Jāizvēlas kāds no piedāvātājiem pasniedzējiem un jānospiež poga "Apstiprināt" 6. Jāizvēlas kāda no piedāvātajām auditorijām un jānospiež poga "Apstiprināt" | Iepriekš apstiprinātās lauku vērtības paliek nemainīgas. Papildus tiek apstiprināta izvēlētā mācību grupas auditorija | Testpiemērs veiksmīgs. |
| 53 | 6.4. | Jaunas mācību grupas pievienošana | groupPlanning.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Izveidot jaunu mācību grupu" 4. Jāizvēlas kāds no piedāvātajiem mācību kursiem un jānospiež poga "Apstiprināt" 5. Jāizvēlas kāds no piedāvātājiem pasniedzējiem un jānospiež poga "Apstiprināt" 6. Jāizvēlas kāda no piedāvātajām auditorijām un jānospiež poga "Apstiprināt" 7. Jāizvēlas mācību grupas sākuma un beigu datumi un jānospiež poga "Apstiprināt" | Iepriekš apstiprinātās lauku vērtības paliek nemainīgas. Papildus tiek apstiprināti mācību grupas sākuma un beigu datumi | Testpiemērs veiksmīgs. |
| 54 | 6.5. | Jaunas mācību grupas pievienošana | groupPlanning.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Izveidot jaunu mācību grupu" 4. Jāizvēlas kāds no piedāvātajiem mācību kursiem un jānospiež poga "Apstiprināt" 5. Jāizvēlas kāds no piedāvātājiem pasniedzējiem un jānospiež poga "Apstiprināt" 6. Jāizvēlas kāda no piedāvātajām auditorijām un jānospiež poga "Apstiprināt" 7. Jāizvēlas mācību grupas sākuma un beigu datumi un jānospiež poga "Apstiprināt" 8. Jānospiež poga "Atjaunot" | Visas iepriekš apstiprinātās lauku vērtības lapā tiek notīrītas un visi vērtību lauki, izņemot mācību kursa izvēles lauku, paliek tukši | Testpiemērs veiksmīgs. |
| 55 | 6.6. | Jaunas mācību grupas pievienošana | groupPlanning.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Izveidot jaunu mācību grupu" 4. Neizvēloties nekādas vērtības jānospiež kāda no lapā esošajām pogām "Apstiprināt" | Lietotājs paliek groupPlanning.php lapā, kur tam ir iespēja izveidot jaunu mācību grupu | Testpiemērs veiksmīgs. |
| 56 | 6.7. | Jaunas mācību grupas pievienošana | groupPlanning.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Izveidot jaunu mācību grupu" 4. Neizvēloties nekādas vērtības jānospiež poga "Atjaunot" | Lietotājs paliek groupPlanning.php lapā, kur tam ir iespēja izveidot jaunu mācību grupu | Testpiemērs veiksmīgs. |
| 57 | 6.8. | Jaunas mācību grupas pievienošana | groupPlanning.php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Izveidot jaunu mācību grupu" 4. Jāizvēlas kāds no piedāvātajiem mācību kursiem un jānospiež poga "Apstiprināt" 5. Jāizvēlas kāds no piedāvātājiem pasniedzējiem un jānospiež poga "Apstiprināt" 6. Jāizvēlas kāda no piedāvātajām auditorijām un jānospiež poga "Apstiprināt" 7. Jāizvēlas mācību grupas sākuma un beigu datumi un jānospiež poga "Apstiprināt" 8. Jānospiež poga "Izveidot mācību grupu" | Jauna mācību grupa datubāzē tiek pievienota. Sistēma attēlo to pašu lapu groupPlanning.php un tās apakšā informatīvu tekstu "Mācību grupa veiksmīgi pievienota datubāzei!" | Testpiemērs veiksmīgs. |
| 58 | 7.1. | Studentu pievienošana mācību grupai | addStudent .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot studentus mācību grupai" 4. Jānospiež poga "Atjaunot" | Sistēmas lapa tiek atjaunināta, ja iepriekš izvēlēta mācību grupa vai veikta studentu meklēšana - šis lapas saturs tiek notīrīts | Testpiemērs veiksmīgs. |
| 59 | 7.2. | Studentu pievienošana mācību grupai | addStudent .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot studentus mācību grupai" 4. Jānospiež poga "Meklēt", neievadot nekādu meklēšanas informāciju | Lapā parādās rezultātu tabula, kura nesatur nevienu ierakstu, taču satur paskaidrojošu tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 60 | 7.3. | Studentu pievienošana mācību grupai | addStudent .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot studentus mācību grupai" 4. Jāizvēlas kāda no izveidotajām mācību grupām un jānospiež poga "Apstiprināt" | Lapas mācību grupas laukā tiek apstiprināta izvēlētā mācību grupa | Testpiemērs veiksmīgs. |
| 61 | 7.4. | Studentu pievienošana mācību grupai | addStudent .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot studentus mācību grupai" 4. Jāizvēlas kāda no izveidotajām mācību grupām un jānospiež poga "Apstiprināt" 5. Jāveic studentu meklēšana, ievadot nekorektus studenta datus vienā no meklēšanas laukiem un jānospiež poga "Meklēt" | Lapas mācību grupas laukā tiek apstiprināta izvēlētā mācību grupa. Lapā parādās rezultātu tabula, kura nesatur nevienu ierakstu, taču satur paskaidrojošu tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 62 | 7.5. | Studentu pievienošana mācību grupai | addStudent .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot studentus mācību grupai" 4. Jāizvēlas kāda no izveidotajām mācību grupām un jānospiež poga "Apstiprināt" 5. Jāveic studentu meklēšana, ievadot nekorektus studenta datus vismaz 2 no meklēšanas laukiem un jānospiež poga "Meklēt" | Lapas mācību grupas laukā tiek apstiprināta izvēlētā mācību grupa. Lapā parādās rezultātu tabula, kura nesatur nevienu ierakstu, taču satur paskaidrojošu tekstu "Pēc šādiem meklēšanas kritērijiem datubāzē nav atrasts neviens ieraksts!" | Testpiemērs veiksmīgs. |
| 63 | 7.6. | Studentu pievienošana mācību grupai | addStudent .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot studentus mācību grupai" 4. Jāizvēlas kāda no izveidotajām mācību grupām un jānospiež poga "Apstiprināt" 5. Jāveic studentu meklēšana, ievadot studenta datus, kuri eksistē datubāzē, vienā no meklēšanas laukiem un jānospiež poga "Meklēt" | Lapas mācību grupas laukā tiek apstiprināta izvēlētā mācību grupa. Lapā parādās rezultātu tabula ar iepriekš meklēto studentu datiem | Testpiemērs veiksmīgs. |
| 64 | 7.7. | Studentu pievienošana mācību grupai | addStudent .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot studentus mācību grupai" 4. Jāizvēlas kāda no izveidotajām mācību grupām un jānospiež poga "Apstiprināt" 5. Jāveic studentu meklēšana, ievadot studenta datus, kuri eksistē datubāzē, vismaz 2 no meklēšanas laukiem un jānospiež poga "Meklēt" | Lapas mācību grupas laukā tiek apstiprināta izvēlētā mācību grupa. Lapā parādās rezultātu tabula ar iepriekš meklēto studentu datiem | Testpiemērs veiksmīgs. |
| 65 | 7.8. | Studentu pievienošana mācību grupai | addStudent .php | 1. Jāatver sistēmas sākuma lapa 2. Jāveic sekmīga autorizācija 3. Galvenajā lapā jānospiež poga "Pievienot studentus mācību grupai" 4. Jāizvēlas kāda no izveidotajām mācību grupām un jānospiež poga "Apstiprināt" 5. Jāveic studentu meklēšana, ievadot studenta datus kādā (-os) no meklēšanas laukiem un jānospiež poga "Meklēt" 6. Rezultātu tabulā jāizvēlas kāds no meklēšanas rezultātiem, nospiežot uz pogas "Pievienot" | Datubāzē tiek veikts ieraksts - izvēlētais students tiek pievienots konkrētajai mācību grupai. Sistēmas lapā parādās ziņojums "Students veiksmīgi pievienots mācību grupai" | Testpiemērs veiksmīgs. |

*Pielikums Nr. 2*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Datums** | **IS lapa** | **Apraksts** |
| 1 | 15.10.2015. | index.php | 1. Ievades lauka nosaukums un pats ievades lauks vienā rindā; 2. Ievades lauka un pogas nosaukumi - latviski. |
| 2 | 15.10.2015. | profile.php | 1. Pārsaukt lapu par main.php; 2. Uztaisīt linku "Mācību grupas plānošana"; 3. Meklēšanā - students un pasniedzējs pārvērst par vienu izvēlni - lietotājs; 4. Pārskatīt visā lapā mīkstinājuma zīmes (pogām, lauku nosaukumiem); 5. Pielikt home pogu (pēc klienta vēlmes); 6. Meklēšanas izkārtojums - meklēšanas ievades laukus vajadzētu līdzināt pēc vienas vertikālas līnijas. |
| 3 | 15.10.2015. | user-page.php | 1. Noņemt nost uzrakstu "TU ESI ADMINISTRATORS". |
| 4 | 15.10.2015. | newUser.php | 1. Ievades lauka nosaukums un pats ievades lauks vienā rindā; 2. Ievades laukus vajadzētu līdzināt pēc vienas vertikālas līnijas; 3. Lietotāja lomas sarakstu vajag vienādā lielumā ar ievades laukiem; 4. Pogu Pievienot vēlams centrēt; 5. Veicot pievienošanu jāuztaisa IF nosacījums - ja nav veiksmīgi pievienots - parādās kļūdas ziņojums, ja ir, parādās informatīvs ziņojums pēc kura lapa pāriet uz main.php; 6. Pareizrakstības kļūda: Perosnas kods --> Personas kods. |
| 5 | 15.10.2015. | newCourse.php | 1. Ievades lauka nosaukums un pats ievades lauks vienā rindā; 2. Ievades laukus vajadzētu līdzināt pēc vienas vertikālas līnijas; 3. Pogu Pievienot vēlams centrēt; 4. Veicot pievienošanu jāuztaisa IF nosacījums - ja nav veiksmīgi pievienots - parādās kļūdas ziņojums, ja ir, parādās informatīvs ziņojums pēc kura lapa pāriet uz main.php. |
| 6 | 15.10.2015. | newRoom.php | 1. Ievades lauka nosaukums un pats ievades lauks vienā rindā; 2. Ievades laukus un checkbox izvēlnes vajadzētu līdzināt pēc vienas vertikālas līnijas; 3. Pogu Pievienot vēlams centrēt; 4. Veicot pievienošanu jāuztaisa IF nosacījums - ja nav veiksmīgi pievienots - parādās kļūdas ziņojums, ja ir, parādās informatīvs ziņojums pēc kura lapa pāriet uz main.php. |
| 7 | 02.11.2015. | newRoom.php | 1. Jāpārskata pievienošanas forma sistēmā - DB ir jauni lauki un tie atšķiras no IS laukiem; 2. Pēc pogas "Pievienot" ieraksts DB netiek pievienots. |
| 8 | 02.11.2015. | index.php | 1. Pēc autorizācijas vajag atvērt atbilstošo lapu katrai lomai. |
| 9 | 02.11.2015. | header.php footer.php | 1. Uzlikt tumši zilu vai violetu fonu. |
| 10 | 03.11.2015. | main.php | 1. Meklēšanas rezultātus attēlot tāpat kā groupPlanning.php lapā. |
| 11 | 04.11.2015. | header.php | 1. Pievienot sistēmas galvenē logo un favikonu. |
| 12 | 06.11.2015. | Visas sistēmas lapas | 1. Sataisīt mīkstinājumzīmes. |
| 13 | 09.11.2015. | main.php | 1. Meklēšanu uztaisīt bez radiobutton pogām tā, kad visi meklēšanas lauki ir redzami uzreiz, tie ir sakārtoti 3 rindās un katras rindas beigās, labajā pusē ir neliela meklēšanas poga. |
| 14 | 09.11.2015. | groupPlanning.php | 1. Veidojot mācību grupu, pēc lauka "Izvēlieties mācību kursu", pievienot jaunu teksta ievades lauku "Ievadiet mācību grupas nosaukumu", jāveic papildinājumi arī datubāzē. |
| 15 | 12.11.2015. | header.php | 1. Pievienot teksta lauku ar informāciju "Sveiki, <Vārds Uzvārds>". |
| 16 | 12.11.2015. | user-page.php | 1. Papildus lauks - Lietotāja loma: Lietotājs / Pasniedzējs / Administrators; 2. Ja lietotājam nav bilde, ievietot kādu lietišķu noklusējuma bildi. |
| 17 | 12.11.2015. | main.php | 1. Pārbaudīt piekļuvi sistēmai bez iepriekšējas autorizācijas. |
| 18 | 12.11.2015. | searchRoom.php | 1. Tafele / Projektors / Video konference informāciju attēlot ar "Ir / Nav"; 2. Kļūda vārdā (konforence > konference); 3. Auditorijas noslogojuma tabulā jābūt papildus laukam "Pasākuma nosaukums", papildus tajā jāielasa atbilstošā informācija. |
| 19 | 16.11.2015. | searchRoom.php searchCourse.php searchPerson.php | 1. Jāatveras atbilstošajām profila lapām. |
| 20 | 16.11.2015. | searchRoom.php searchCourse.php searchPerson.php | 1. Ja lietotājs nav pievienojis porfila bildi, nepieciešams attēlot noklusējuma bildi. |
| 21 | 16.11.2015. | user-page.php searchRoom.php searchCourse.php searchPerson.php | 1. Lauku "Lietotāja loma" vajag virs laukiem "apgūtie kursi, iegūtie diplomi, iegūtie sertifikāti, pasniedzamie kursi". |
| 22 | 23.11.2015. | newRoom.php | 1. Nodrošināt pogas "Pievienot" funkcionalitāti. |
| 23 | 23.11.2015. | user-pageForUser.php | 1. Lauku "Lietotāja loma" vajag virs laukiem "apgūtie kursi, iegūtie diplomi, iegūtie sertifikāti, pasniedzamie kursi". |
| 24 | 23.11.2015. | profile.php | 1. Ielogojoties ar lietotāju, kura tiesības atbilst pasniedzēja līmenim parādās lapa profile.php, kurā attēlots nenostilots lauks "Mācību GRUPU PLĀNOŠANA". |
| 25 | 23.11.2015. | index.php | 1. Lauks "Nav ievadīts lietotājvārds un / vai parole!" jānostilo, lai tas ir zem pogas "Ienākt sistēmā". |
| 26 | 25.11.2015. | index.php | 1. Jāizdala divas pogas "Jaunas mācību grupas izveidošana" un "Pievienot studentus mācību grupai". |
| 27 | 25.11.2015. | addStudent.php | 1. Studentu meklēšanas laukus savilkt ciešāk kopā vertikālā līmenī, atstājot aptuveni 5-10 px atstarpi starp tiem. |
| 28 | 27.11.2015. | addStudent.php | 1. Nodrošināt pogas "Pievienot" funkcionalitāti. |

# Galvojums

Ar šo es, Arnis Freimanis, galvoju, ka bakalaura darbs ir izpildīts patstāvīgi, konsultējoties ar darba vadītāju. No svešiem pirmavotiem ņemtā informācija ir norādīta ar atsaucēm, dati un definējumi ir uzrādīti darbā. Šis darbs tādā vai citādā veidā nav nekad iesniegts nevienai citai pārbaudījumu komisijai.

Esmu informēts (-a), ka mans bakalaura darbs tiks ievietots un apstrādāts Vienotajā datorizētajā plaģiāta kontroles sistēmā plāģiāta kontroles nolūkos.

20\_\_.gada\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(paraksts)