**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**

[Ievads 6](#_Toc443386027)

[Saīsinājumu un nosacīto apzīmējumu 7](#_Toc443386028)

[1. Informācijas sistēma 8](#_Toc443386029)

[1.1. Informācijas sistēmas jēdzines 8](#_Toc443386030)

[1.2. Informācijas sistēmas izstrādes dzīves cikls 9](#_Toc443386031)

[1.3. Informāciju sistēmas izstrādes modeļi 11](#_Toc443386032)

[2. Datu bāzes un saskarnes risinājumi 14](#_Toc443386033)

[2.1. Datu bāzes risinājumi 14](#_Toc443386034)

[2.1.1. Datubāzes veidošanas alternatīvas 14](#_Toc443386035)

[2.1.2. MySQL piedāvātās iespējas 14](#_Toc443386036)

[2.2. Saskarnes veidošanas risinājumi 18](#_Toc443386037)

[2.2.1. Saskarnes veidošanas populārākās alternatīvas 18](#_Toc443386038)

[2.2.2. Izmantotās tehnoloģijas pamatojums 18](#_Toc443386039)

[3. Mācību centra vadības sistēmas projektēšana 18](#_Toc443386040)

[3.1. Mācību centra vadības sistēmas prasības 18](#_Toc443386041)

[3.2. Sistēmas datu struktūras projektējums 18](#_Toc443386042)

[3.3. Saskarnes projektējums 18](#_Toc443386043)

[4. Mācību centra vadības sistēmas izstrāde 18](#_Toc443386044)

[4.1. MySQL datu bāzes izstrāde 18](#_Toc443386045)

[4.1.1. Sistēmas datu bāze 18](#_Toc443386046)

[4.1.2. Sistēmas datu bāzes tabulas 18](#_Toc443386047)

[4.2. Saskarnes izstrāde 18](#_Toc443386048)

[4.2.1. WEB aplikācijas izstrāde 18](#_Toc443386049)

[Secinājumi un priekšlikumi 19](#_Toc443386050)

[Pielikums 20](#_Toc443386051)

[Galvojums 22](#_Toc443386052)

# 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 WEB aplikāciju 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 WEB aplikāciju izmantojot izvēlēto tehnoloģiju un saistīt aplikāciju 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

# 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. [2] 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. [3]

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. [3]

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. [2]

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. [2]

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.

Pirmās četras komponentes ir daļas no vispārējas informācijas tehnoloģiju organizācijā. Procedūras ir ļoti specifiska informācija, kas vajadzīga, lai atbildētu uz konkrētu jautājumu. [2]

## 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 [5]:

* 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. [6]

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

## Informāciju sistēmas izstrādes modeļi

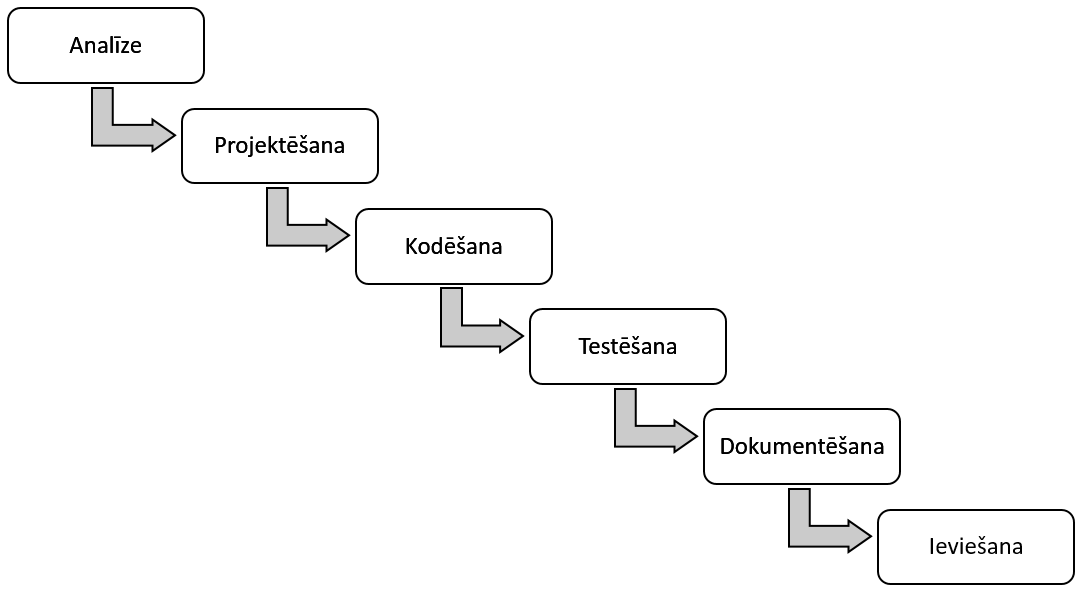
Izstrādes modeļi ir dažādi procesi un metodes, kas tiek izmantoti sistēmas izstrādē atkarībā no projekta mērķiem un veicamajiem uzdevumiem. Ir vairāki izstrādes modeļi, kas ir izstrādāti, lai sasniegtu dažādus nepieciešamos mērķus. Modeļi precizē dažādus procesa posmus un kārtību kādā tie tiks veikti [7].

Ir dažādi programmatūras izstrādes modeļi, tie ir šādi:

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

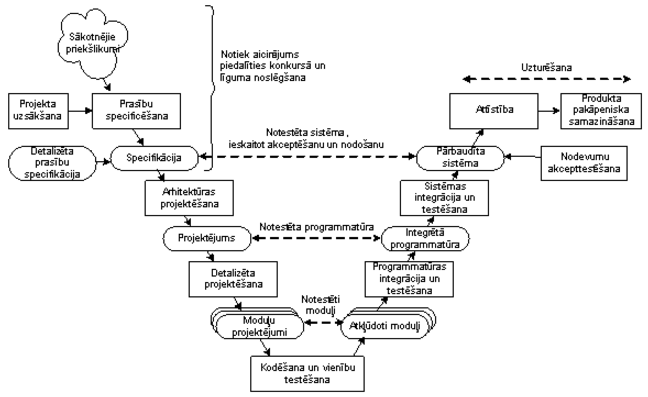
Autors savā darbā apskatīs trīs visbiežāk izmantotos sistēmas izstrādes modeļus, kas ir ūdenskrituma modeli, V modeli un Agile modeli. Pēc modeļu apskatīšanas un aprakstīšanas tiks veikta to salīdzināšana, lai noskaidrotu, kuru no modeļiem, kādās situācijās izmantot.

* 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 [8].



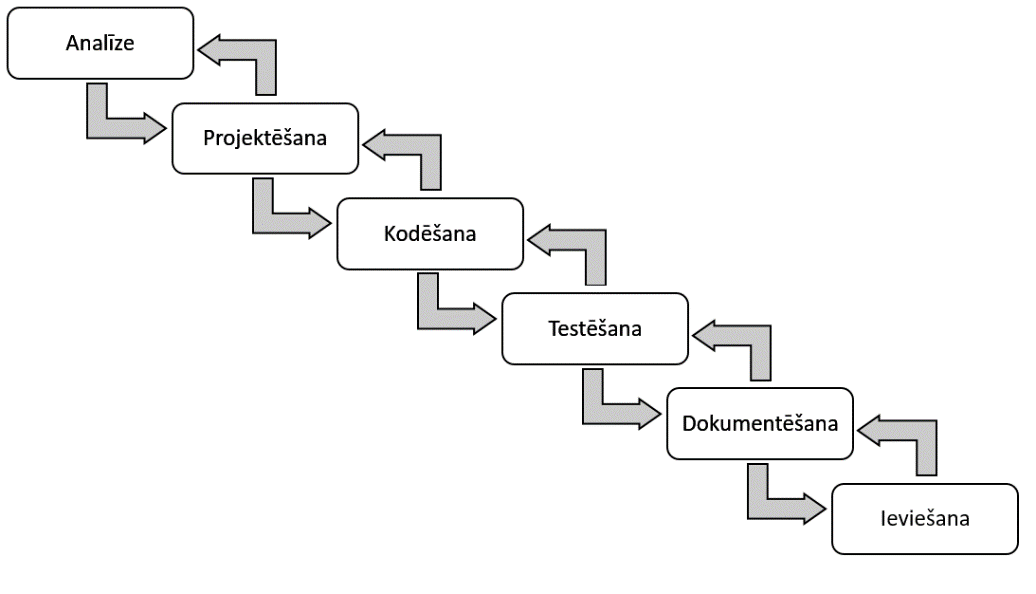
1.2 att. Ūdenskrituma modelis

* 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 [8].



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. [9]



1.4 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 veidošans 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 [10].

### Datu bāzes veidošanas alternatīvas

Datubāzes ir pamats mūsdienu uzņēmējdarbībā. Gan publiskiem sektoriem, gan augstu tehnoloģiju nozarēm, infrastruktūra uzņēmumā ir 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.

2.1 tabula

**MySQL, PostgreSQL, Oracle, SQLite un MS Access datu bāžu salīdzinājums** [11]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 aplikācijām un tāpiedāvā lielisku izmaksu/ ieguvumu attiecību. Ne tikai tāpēc ka MySQL ir bezmaksas, bet arī tāpēc, ka tai ir pieticīgas prasības uz aparatūru. [12]

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. [13]
* Plaša aplikāciju 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. [14]

## Saskarnes veidošanas risinājumi

### Saskarnes veidošanas populārākās alternatīvas

### Izmantotās tehnoloģijas pamatojums

# Mācību centra vadības sistēmas projektēšana

## Mācību centra vadības sistēmas prasības

## Sistēmas datu struktūras projektējums

## Saskarnes projektējums

# Mācību centra vadības sistēmas izstrāde

Pamatojoties uz iepriekšējā nodaļa veikto MCVS projektēšanu, šajā nodaļā autors apraksta pašas sistēmas izstrādi.

## MySQL datu bāzes izstrāde

Šajā nodaļā tiek aprakstīta un izanalizēta izveidotā mācību centra vadības sistēmas datu bāze.

### Sistēmas datu bāze

### Sistēmas datu bāzes tabulas

## Saskarnes izstrāde

### WEB aplikācijas izstrāde

# Secinājumi un priekšlikumi

# Izmantotās literatūras un avotu saraksts

# Pielikums

# 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)