

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

MĀCĪBU VADĪBAS SISTĒMA “ACADEMICFLOW”

Autori:

Arina Fokina af21043,

Artjoms Mačulis am21169,

Anastasija Saveika as21297,

Jelizaveta Zabanova jz21055,

Nikita Trofimovs nt21006,

Valērijs Šermanis vs21153,

Viktorija Ivanova vi21012.

RĪGA 2024

ANOTĀCIJA

Kursa grupas darba “AcademicFlow” mērķis ir izstrādāt un dokumentēt lietojumprogrammu, kas nodrošina studentiem un pasniedzējiem ērtu studiju procesa pārvaldību. Sistēmas lietotājiem tiek piedāvāta iespēja piekļūt savam kursu sarakstam, apskatīt nodevumus un iesniegt failus, kā arī apskatīt savus datus.

Administratoriem ir personīgais administrācijas centrs, kur viņi pārvalda visus sistēmas lietotājus, kursus, programmas un autorizācijas datus, norīkot studentus programmai, norīkot studentus un pasniedzējus kursiem, kā arī pievienot kursus programmai.

Sistēma ir izstrādāta “ASP.NET” vidē, izmantojot HTML5, CSS, JavaScript, un C# valodā. Programmatūras dokumentācijā tika detalizēti aprakstītas programmatūras funkcionālās un nefunkcionālās prasības, izveidoti funkciju un saskarņu projektējumi, veikta testēšana un novērtēta projekta organizācija.

Atslēgas vārdi: tīmekļa vietne, JavaScript, SQL, C#, API, ASP.NET

ABSTRACT

The goal of the course group work "AcademicFlow" is to develop and document an application that provides students and teachers with convenient management of the study process. System users are offered the ability to access their course list, view assignments and submit files, and view their data.

Administrators have a personal administration center where they manage all system users, courses, programs and authorization data, assign students to a program, assign students and instructors to courses, and add courses to a program.

The system is developed in the ASP.NET environment using HTML5, CSS, JavaScript, and C#. The software documentation detailed the functional and non-functional requirements of the software, created feature and interface designs, performed testing, and assessed the project organization.

Keywords: Website, JavaScript, SQL, C#, API, ASP.NET

SATURA RADĪTĀJS

ANOTĀCIJA	2
ABSTRACT.....	3
APZĪMĒJUMU SARAKSTS	7
IEVADS	8
Nolūks	8
Pasūtītājs.....	8
Darbības sfēra.....	8
Saistība ar citiem dokumentiem	8
Pārskats.....	8
1. VISPĀRĒJAIS APRAKSTS	10
1.1 Esošā stāvokļa apraksts	10
1.2 Pasūtītājs.....	10
1.3 Produkta perspektīva	10
1.4 Darījumprasības	10
1.5 Sistēmas lietotāji	11
1.6 Vispārējie ierobežojumi	13
1.7 Pieņēmumi un atkarības	14
2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA.....	15
2.1 Funkcionālās prasības	15
2.2 Paziņojumu saraksts	16
2.3 Funkciju sadalījums pa moduļiem.....	18
2.3.1 Autorizācijas modulis	19
2.3.2 Kursu modulis.....	21

2.3.3 Lietotāju modulis	24
2.3.4 Administratora modulis	25
2.4 Nefunkcionālās prasības.....	31
2.4.1 Lietotāja saskarnes prasības.....	31
2.4.2 Pieejamības prasības.....	31
2.4.3 Drošības prasības	31
3. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	33
3.1 Datu bāzes projektējums	33
3.1.1 Loģiskais datu bāzes modelis	33
3.1.2 Fiziskais datu bāzes modelis.....	34
3.2 Funkciju projektējums	35
3.2.1 Atkarību diagrammas projektējums.....	35
3.2.2 Kļāšu diagrammu projektējums.....	36
3.2.3 Funkcijas LOG_IN projektējums	38
3.2.4 Funkcijas AM_03(1) projektējums.....	39
3.3 Lietotāja saskarņu projektējums.....	40
3.3.1 Lietotāja autorizācijas ekrāns	40
3.3.2 Administratora centrs	41
3.3.3 Lietotāja profils.....	43
4. Programmatūras testēšana.....	46
4.1 Testēšanas žurnāls	46
5. Projekta organizācija.....	50
5.1 Darbietilpības novērtējums	50
5.2 Projekta kvalitātes nodrošināšana	51
5.3 Konfigurāciju pārvaldība.....	51
Rezultāti	53
Secinājumi.....	54

IZMANTOTĀ LITERATŪRA UN AVOTI	55
-------------------------------------	----

APZĪMĒJUMU SARAKSTS

JQuery – ātra, neliela un funkciju bagāta JavaScript bibliotēka. Tā padara lietas, piemēram, HTML dokumentu pārlūkošanu un manipulāciju, notikumu apstrādi, animācijas un “Ajax” daudz vienkāršākas ar viegli lietojamu API, kas darbojas daudzās pārlūkprogrammās.

Blazor – “.NET” lietotāja saskarnes sastāvdaļa, kas atbalsta gan servera pusē datu apstrādi, gan klienta interaktivitāti vienotā programmēšanas modelī.

DRY – “Neatkārto sevi” (Don't Repeat Yourself) kodēšanas princips. Katram zināšanu elementam jābūt vienam, viennozīmīgam, autoritatīvam pārstāvībā sistēmā.

DPD – datu plūsmu diagramma.

ER – (Entity-Relationship) Datu bāzes modelis, izmanto ER diagrammas, lai vizuāli attēlotu datu bāzes struktūru un attiecības starp dažādām entitātēm.

GIT – bezmaksas atvērtā koda versiju pārvaldības sistēma.

ID – unikāls tabulu ieraksta identifikators datu bāzē.

JavaScript – skriptu valoda, kas ļauj tīmekļa izstrādātājiem veidot interaktīvas vietnes..

JQuery – ātra, neliela un funkciju bagāta JavaScript bibliotēka. Tā padara lietas, piemēram, HTML dokumentu pārlūkošanu un manipulāciju, notikumu apstrādi, animācijas un “Ajax” daudz vienkāršākas ar viegli lietojamu API, kas darbojas daudzās pārlūkprogrammās. MVC – Modeļa-skata-kontroliera (Model View Controller) programmatūras projektējuma struktūra. Izmanto, lai atdalītu lietotāja saskarni (skatu), datus (modeli) un lietojumprogrammas loģiku (kontrolieri).

Razor Pages – piedāvā vienkāršu, tīru un vieglu veidu, kā radīt dinamisku tīmekļa saturu, izmantojot HTML un C# valodas.

SQL Server – “Microsoft SQL Server” ir relāciju datu bāzu pārvaldības sistēma (RDBMS).

W3C – Tīmekļa standarti. Tie ir pamatplāni tam, kā implementēt pārlūkprogrammas, blogus, grafikas redaktorus, meklētājus un daudzus citus programmēšanas risinājumus, kas nodrošina mūsu pieredzi tīmeklī.

CRUD – Create, Read, Update, Delete

IEVADS

Nolūks

Kursa darba nolūks ir izstrādāt sistēmu, kas nodrošina efektīvu veidu kā pārvaldīt un apstrādāt informāciju saistībā ar izglītības procesu. Tas piedāvā plašas iespējas administratoriem, pasniedzējiem un studentiem, aprakstot dokumentācijā sistēmas darījumprasības, testēšanas piemērus un projektējumu tīmekļa vietnei.

Pasūtītājs

Sistēma ir izstrādāta “DatZ4019-LV : Objektorientētā programmēšana” kursa ietvaros kā grupas darbs. Sistēma un sistēmas dokumentācija izveidoti **no jauna** tieši priekš šī kursa grupas darba.

Darbības sfēra

Sistēma ir paredzēta izglītības iestādēm, kuri vēlas pārvaldīt savus kursus un programmas, studentus un pasniedzējus. Tīmekļa vietne piedāvā lietotājiem ērtu un efektīvu veidu, kā pārvaldīt un apstrādāt informāciju saistībā ar izglītības procesu.

Sistēma ir izstrādāta lietotājiem saprotamā veidā, lai pēc iespējas vienkāršotu datu apmaiņas procesu. Studenti var apskatīt savas atzīmes un pievienot failus. Pasniedzējiem ir iespēja pievienot nodevumus un likt atzīmes studentiem. Lai sistēma būtu pieejama un saprotama lielākam lietotāju skaitam, tīmekļa vietne ir izstrādāta angļu valodā.

Saistība ar citiem dokumentiem

Dokumenta izveidē un noformēšanā tiek ievērotas standarta LV 68:1996 “Programmatūras prasību specifikācijas ceļvedis” [3] un LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai” [4] rekomendācijas un prasības.

Pārskats

Šis dokuments sastāv no 5 nodaļām: no programmatūras vispārēja apraksta, prasību specifikācijas apraksta, programmatūras projektējuma apraksta, programmatūras testēšanas dokumentācijas, projekta organizācijas apraksta.

1. VISPĀRĒJAIS APRAKSTS

1.1 Esošā stāvokļa apraksts

Gandrīz katrai universitātei 2024. gadā pastāv sava elektroniskā mācību vide, kas ir pieejama tikai noteiktās universitātes studentiem un pasniedzējiem, piemēram, e-studijas, ortus. Viena no pasaulē populārākajām mācību vadības sistēmām ir Moodle. Rīks ar kura palīdzību ir iespējams pilnībā organizēt mācību procesu, mācību materiālu pārvaldību, apmeklējumu, vērtēšanu un pat mācību materiālu izstrādi. Mūsu komanda nolēma izveidot sistēmu, kas ir līdzīga Moodle rīkam, bet ar jaunu dizainu, svarīgāko pēc komandas domām funkcionalitāti un pieejamu nevis publiski, bet tikai koda īpašniekiem.

1.2 Pasūtītājs

Sistēmas izstrāde un dokumentācijas apraksts ir veidoti pēc kursa komandu dalībnieku personīgas iniciatīvas objektorientētas programmēšanas kursa ietvaros.

1.3 Produkta perspektīva

Sistēma “AcademicFlow” ir piemērota studentiem, pasniedzējiem un administratoriem, lai varētu vieglāk un efektīvāk pārvaldīt izglītības procesu. Mūsu izstrādātu sistēmu nākotnē var papildināt ar papildus funkcionalitāti, piemēram, studenta un pasniedzēja lekciju sarakstu ar laikiem, meklēšanu pēc atslēgvārdiem pa visu tīmekļa vietni, piedāvāt tīmekļa vietni citās valodās un citas funkcijas, kuras vēlēsies sistēmas koda un dokumentācijas īpašnieks vai pasūtītājs.

1.4 Darījumprasības

“AcademicFlow” nodrošinās sekojošas funkcijas reģistrētiem lietotājiem kā studenti:

- Lietotāja pieslēgšanās sistēmai
- Lietotāja izrakstīšanās no sistēmas
- Lietotāja reģistrēšanās
- Vērtējumu apskatīšana
- Kursa datu apskatīšana

- Nodevumu pārvaldīšana (CRUD)
- Personīgo datu apskatīšana
- E-pasta un telefona numura rediģēšana

“AcademicFlow” nodrošinās sekojošas funkcijas reģistrētiem lietotājiem kā pasniedzēji:

- Lietotāja pieslēgšanās sistēmai
- Lietotāja izrakstīšanās no sistēmas
- Lietotāja reģistrēšanās
- Kursa datu apskatīšana
- Nodevumu pārvaldīšana (CRUD)
- Studentu atzīmju pārvaldīšana (CRUD)
- Personīgo datu apskatīšana
- E-pasta un telefona numura rediģēšana

“AcademicFlow” nodrošinās sekojošas funkcijas reģistrētiem lietotājiem kā administratori:

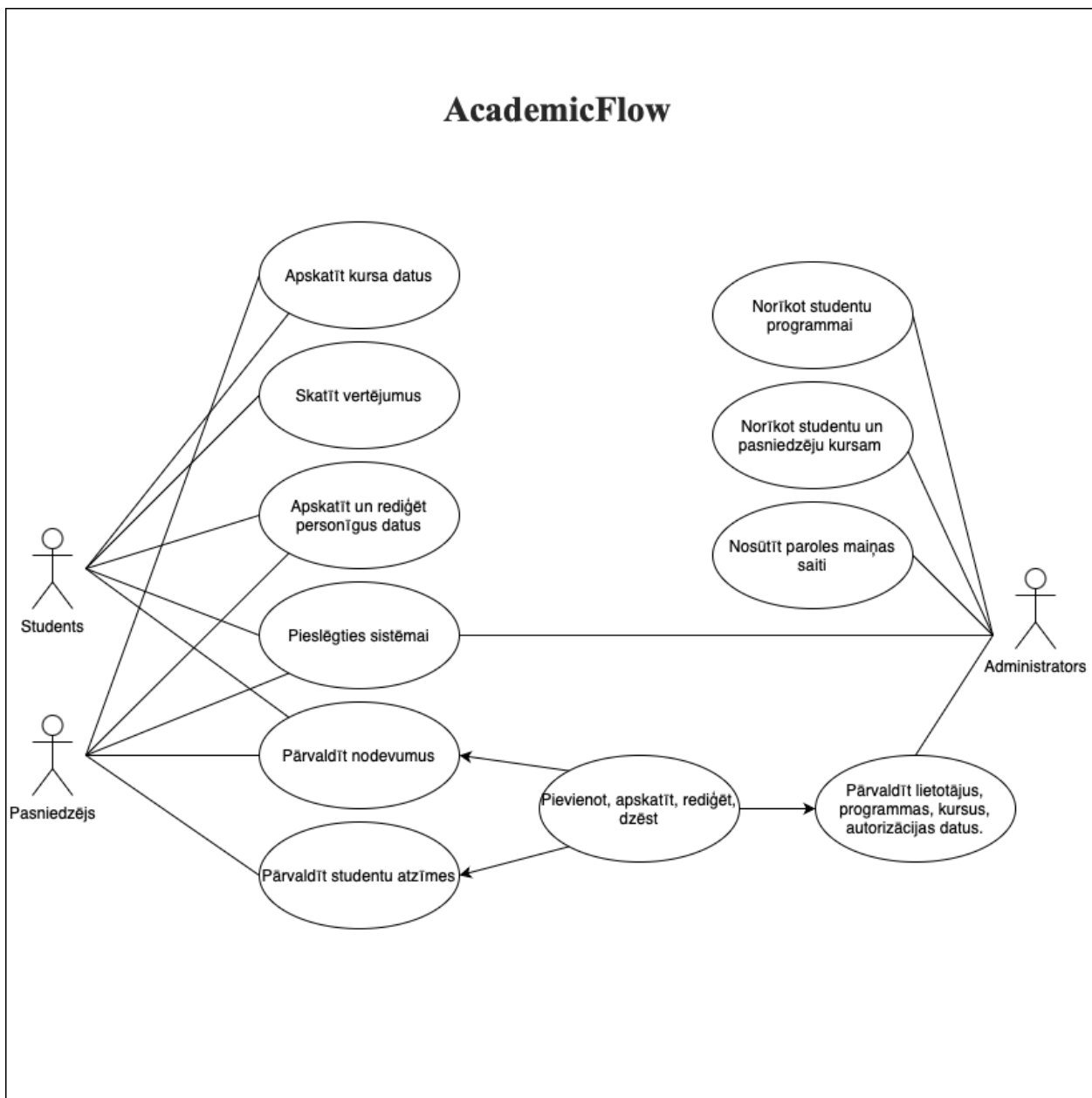
- Pieslēgšanās sistēmai
- Izrakstīšanās no sistēmas
- Studentu norīkošana programmai
- Studentu un pasniedzēju norīkošana kursam
- Paroles maiņas saites nosūtīšana
- Lietotāju, programmu, kursu, autorizācijas datu pārvaldīšana (CRUD).

1.5 Sistēmas lietotāji

Tīmekļa vietne nav publiski pieejama. Lietotājs reģistrējas izmantojot saiti, kuru nosūta administrators.

Tabula 1.1. Lietotāju grupu apraksts

Lietotāju grupa	Apraksts
Administrators	Spēj apskatīt, pievienot, dzēst un rediģēt visus sistēmas lietotājus, programmas, kursus, autorizācijas datus, var sūtīt paroles maiņas saiti.
Pasniedzējs	Iespēja apskatīt kursa datus, apskatīt un rediģēt personīgus datus. Spēj apskatīt, pievienot, dzēst un rediģēt nodevumus un studentu atzīmes.
Students	Var apskatīt, pievienot, dzēst un rediģēt nodevumus, apskatīt un rediģēt personīgus datus, skatīt vērtējumus un apskatīt kursa datus.



1.1.att. “AcademicFlow” lietošanas gadījuma diagramma

1.6 Vispārējie ierobežojumi

Lai izmantotu šo sistēmu, nepieciešama ierīce (dators, telefons vai planšetdators), ar kuru var piekļūt internetam un kurā ir uzstādīta pārlūkprogramma. Darbs ar šo sistēmu ir paredzēts, izmantojot norādītās vai jaunākās pārlūkprogrammu versijas.

Tabula 1.2. Pārlūkprogrammas versijas sistēmas korektai darbībai

Pārlūkprogramma	Pēdējā stabilā pārlūkprogrammas versija
Google Chrome (Windows, macOS, Linux)	120.0.6099.129/130
Google Chrome (Android)	120.0.6099.144
Google Chrome (iOS)	120.0.6099.119
Edge (Windows, macOS, Linux)	120.0.2210.91
Edge (iOS)	120.2210.105
Firefox (Standard Release, iOS, Android)	121.0
Opera (Desktop)	106.0.4998.12
Opera (Android)	76.2.4027.73374

Tabulā ir sniegtas sistēmas pārbaudes minimālās pārlūkprogrammas prasības.

1.7 Pieņēmumi un atkarības

Lai pilnvērtīgi strādātu, sistēmai ir nepieciešams būt atbilstošai W3C standartiem – rakstot HTML un CSS kodu, tiek nodrošināta pārbaude uz atbilstību uz noteiktajiem standartiem. Sistēma ir izstrādāta darbībai un pārbaudīta, izmantojot iepriekš norādītās pārlūkprogrammas, kuras ir minētas 1.6 nodaļā "Vispārējie ierobežojumi". Citu pārlūkprogrammu izmantošana nav noteikti aizliegta, tomēr var rasties situācijas, kurās pilnīga sistēmas funkcionalitāte nav garantēta.

2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

Šajā sadaļā ir detalizēti aprakstītas konkrētas prasības un to paredzētie rezultāti.

2.1 Funkcionālās prasības

Tabulā 2.1. ir uzskaitītas funkcijas, kas ir sadalītas pa moduļiem. Ar “X” tiek atzīmētā lietotāju grupa, kurai ir piekļuve dotajai funkcijai.

Tabula 2.1. Funkciju apkopojums pa moduļiem

Modulis	Funkcija	Lietotāju grupas		
		Students	Pasniedzējs	Administrators
Autorizācijas modulis	Pieslēgšanās sistēmai	X	X	X
	Izrakstīšanās no sistēmas	X	X	X
	Lietotāja reģistrēšanās	X	X	
Administratora modulis	Studentu norīkošana programmai			X
	Studentu un pasniedzēju norīkošana kursam			X
	Paroles atiestatīšana			X
	Lietotājus programmu, kursu, autorizācijas datu pārvaldīšana (CRUD)			X
	Lomas piešķiršana lietotājam			X
Kursu modulis	Kursa datu apskatīšana	X	X	
	Vērtējumu apskatīšana	X		

	Nodevumu pārvaldīšana (CRUD)	X	X	
	Studentu atzīmju pārvaldīšana (CRUD)		X	
Lietotāju modulis	Personīgo datu apskatīšana	X	X	
	E-pasta un telefona numura rediģēšana	X	X	

2.2 Paziņojumu saraksts

Tabula 2.2. Paziņojumu saraksts administratora centrā

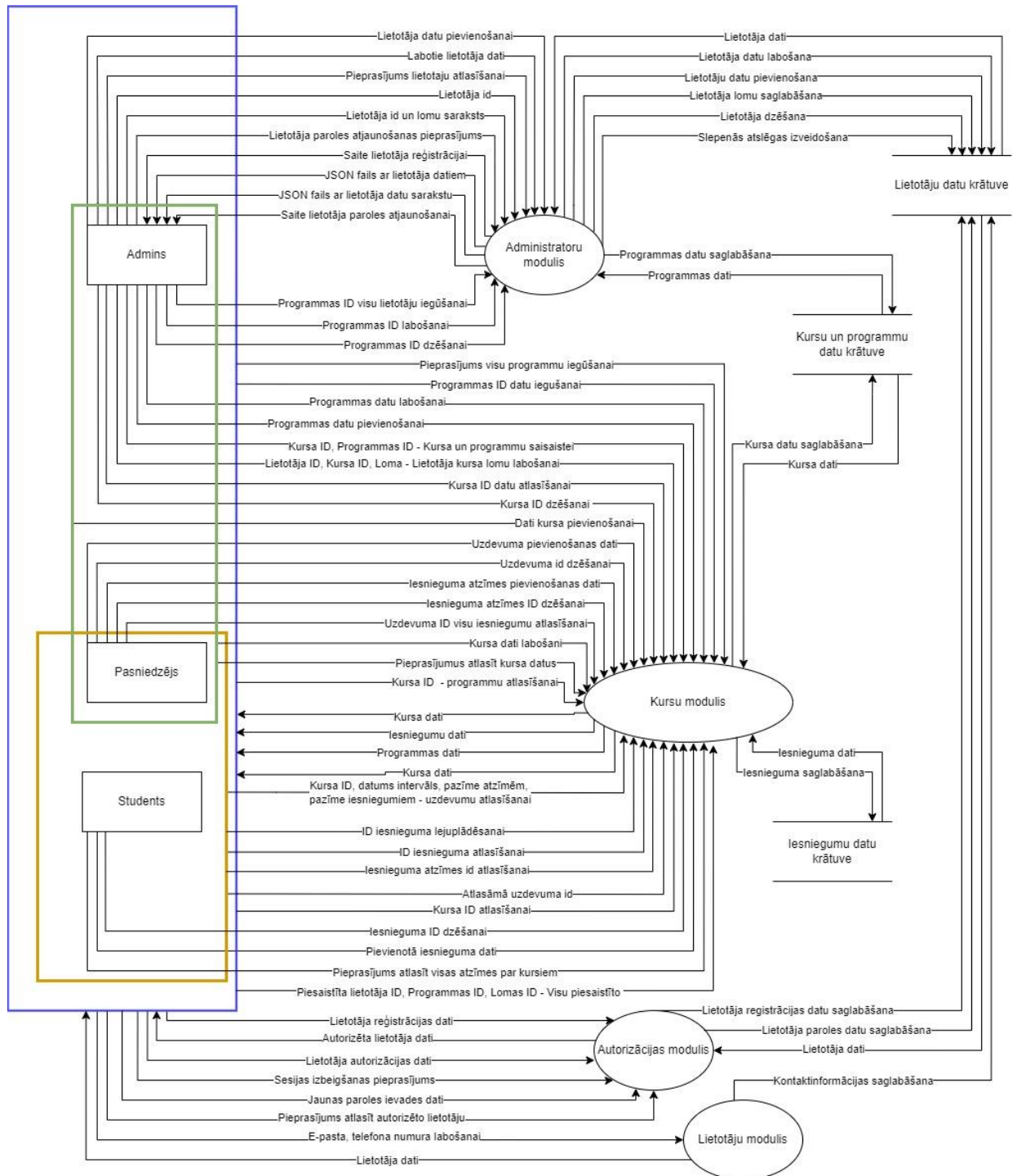
Administratora paziņojuma identifikators	Paziņojuma teksts
API_01	“User is assigned”
API_02	“User is not assigned”
API_03	“User is added”
API_04	“User is not added”
API_05	“User is changed”
API_06	“User is not changed”
API_07	“User is deleted”
API_08	“User is not deleted”
API_09	“Role is changed”
API_10	“Role is not changed”
API_11	“Course is added”
API_12	“Course is not added”
API_13	“Course is edited”

API_14	“Course is not edited”
API_15	“Course is deleted”
API_16	“Course is not deleted”
API_17	“Program is added”
API_18	“Program is not added”
API_19	“Program is edited”
API_20	“Program is not edited”
API_21	“Program is deleted”
API_22	“Program is not deleted”
API_23	“Password is reset”
API_24	“Password is not reset”

Tabula 2.3. Paziņojumu saraksts no lietotāja puses

Lietotāja paziņojuma identifikators	Paziņojuma teksts
LPI_01	“User with this username or password does not exist”
LPI_02	“Something went wrong”
LPI_03	“Profile info is changed”
LPI_04	“Profile info is not changed”
LPI_05	“Assignment is added”
LPI_06	“Assignment is not added”

2.3 Funkciju sadalījums pa moduļiem



2.1.att. 1. līmeņa datu plūsmas diagramma

1.līmeņa datu plūsmas diagrammā attēlā 2.1.att. ir redzams, ka sistēma sastāv no 4 moduļiem un 3 datu krātuvēm. Administratoru modulis ļauj administratoram pārvaldīt lietotājus, programmas, kursus un autorizācijas datus. Kursu modulis nodrošina kursa apraksta izveidi, nodevumu un atzīmju pārvaldīšanu. Autorizācijas modulis atbild par lietotāja pieslēgšanos savam kontam, atteikšanos no tā un registrēšanās. Lietotāja modulis ļauj lietotājam apskatīt savus personīgus datus un rediģēt e-pastu un telefona numuru.

2.3.1 Autorizācijas modulis

Tabula 2.4. Pieslēgšanās sistēmai funkcijas apraksts

Nosaukums	Pieslēgšanās sistēmai
Identifikators	LOG_IN
Mērķis	
	Funkcija ir paredzēta, lai lietotājs piekļūtu savam kontam.
Piekluve	
	Piekluve funkcijai ir neautorizētam lietotājam, kuram ir autorizācijas dati datubāzē.
Ievade	
	Lietotājs ievada savu 1. lietotājvārdu – obligāti; 2. paroli – obligāti. Un nospiež pogu “Log in”.
Apstrāde	
	Tiek pārbaudīts, vai visi lauki ir aizpildīti. Dati tiek nosūtīti uz serveri un salīdzināti ar datu bāzē glabātiem autorizācijas datiem.
Izvade	
	1. Ja lietotājvārds un/vai parole tika ievadīti nepareizi, tiek izvadīts kļūdas paziņojums LPI_01. 2. Ja dati tika ievadīti pareizi, lietotājs ir ielogots savā kontā.

Tabula 2.5. Izrakstīšanās no sistēmas funkcijas apraksts

Nosaukums	Izrakstīšanās no sistēmas
Identifikators	LOG_OUT
Mērķis	
	Funkcija ir paredzēta, lai lietotājs izietu no sava konta.
Pieklūve	
	Pieklūve funkcijai ir autorizētam lietotājam.
Ievade	
	Lietotājs nospiež pogu “Sign out”.
Apstrāde	
	Lietotājs ir izmests no sistēmas.
Izvade	
	Lietotājs tiek pārdresēts uz autorizācijas sākumlapu.

Tabula 2.6. Lietotāja reģistrēšanās funkcijas apraksts

Nosaukums	Lietotāja reģistrēšanās
Identifikators	AM_01
Mērķis	
	Funkcija, lai lietotājs varētu izdomāt savu lietotājvārdu un paroli.
Pieklūve	
	Pieklūve ir neautorizētam lietotājam, kuram administrators nosūtīja reģistrēšanas saiti.
Ievade	
	Lietotājs ievada: <ul style="list-style-type: none"> 1. Lietotājvārdu - obligāti; 2. Paroli - obligāti Nospiež pogu “Sign up”.
Apstrāde	
	Dati tiek nosūtīti uz datu bāzi.
Izvade	
	Lietotājs tiek pārdresēts uz savu kontu.

2.3.2 Kursu modulis

Tabula 2.7. Studentu atzīmju pārvaldīšanās funkcijas apraksts

Nosaukums	Studentu atzīmju pārvaldīšana (CRUD)
Identifikators	KR_01
Mērķis	
Funkcija, lai pasniedzējs varētu pievienot, apskatīt, rediģēt un dzēst atzīmes studentiem.	
Piekluve	
Tikai pasniedzējam.	
Ievade	
<ol style="list-style-type: none">1. Lai pievienot atzīmi: Atver sadaļu “Assignments”, nospiež uz nodevuma nosaukuma, nospiež uz pogas “View” pie attiecīga studenta nodevuma un aizpilda atzīmes un komentāra laukus. Nospiež uz pogas “Grade an assignment”.2. Lai apskatīt atzīmi: Atver sadaļu “Assignments” un nospiež uz nodevuma nosaukumu.3. Lai rediģēt (dzēst) atzīmi: Atver sadaļu “Assignments”, nospiež uz nodevuma nosaukuma, nospiež uz pogas “View” pie attiecīga studenta nodevuma un ievada jaunu atzīmi un komentāru. Nospiež uz pogas “Grade an assignment”.	
Apstrāde	
<ol style="list-style-type: none">1. Dati tiek nosūtīti uz datu bāzi.2. No datu bāzes tiek iegūti dati par nodevumu.3. Dati tiek atjaunināti datu bāzē.	
Izvade	
<ol style="list-style-type: none">1. Atzīme ir redzama pie apraksta par studenta nodevumu.2. Lietotājs redz atzīmi par attiecīgo nodevumu sarakstā.3. Lietotājs redz jauno atzīmi pie attiecīga nodevuma.	

Tabula 2.8. Kurša datu apskatīšanās funkcijas apraksts

Nosaukums	Kurša datu apskatīšana
Identifikators	KR_02
Mērķis	

Funkcija ļauj apskatīt informāciju par kursu.
Piekluve
Autorizētam lietotājam.
Ievade
Lietotājs nospiež un sadaļas “Courses” un nospiež uz pogu “Learn more” kursam, kuru vēlās apskatīt.
Apstrāde
Tiek iegūti dati par kursu no datu bāzes.
Izvade
Lietotājs redz kursa attēlu, nosaukumu, kredītpunktu skaitu, aprakstu, pasniedzēja vārdu un uzvārdu, atzīmi par kursu un nodevumu sarakstu.

Tabula 2.9. Nodevumu pārvaldīšanās funkcijas apraksts

Nosaukums	Nodevumu pārvaldīšana (CRUD)
Identifikators	KR_03
Mērķis	
Funkcija ir paredzēta, lai studenti varētu pievienot failus.	
Piekluve	
Tikai studentam.	
Ievade	
<ol style="list-style-type: none"> 1. Lai ielādētu failu, lietotājam jānospiež uz sadaļas “Assignments”, jānospiež uz nodevuma nosaukumu no tabulas un jānospiež uz pogu “Add new assignment entry”. Tad jāievada nodevuma uzdevuma ID un augšupielādē failu no sava datora. Tad nospiež pogu “Add assignment entry”. 2. Lai rediģētu (nodzēst) failu, jānospiež uz sadaļas “Assignments”, jānospiež uz nodevuma nosaukumu no tabulas un jānospiež uz pogu “View” pie attiecīga nodevuma un jānospiež uz pogas “Delete assignment entry”. Lai apstiprinātu dzēšanu jānospiež “ok”. 3. Lai apskatītu visus savus nodevumus, lietotājam jānospiež uz sadaļas “Assignments”, jānospiež uz nodevuma nosaukumu no tabulas. 	

Apstrāde
<ol style="list-style-type: none"> 1. Dati tiek nosūtīti uz datu bāzi. 2. Nodevums tiek dzēsts no datu bāzes. 3. No datu bāzes tiek iegūti dati par nodevumu un dati par augšupielādēto failu.
Izvade
<ol style="list-style-type: none"> 1. Ja tika norādīts uzdevuma ID un pievienots fails, tad tiek izvadīts paziņojums LPI_05. Citādi tiek izvadīts kļūdas paziņojums LPI_06. 2. Lietotājs tiek pāradresēts uz sadaļu “Assignments”. Nodevums nav redzams “Your submitted assignment” sadaļā. 3. Lietotājs redz nodevuma nosaukumu, uzdevuma aprakstu, autoru, nodevuma vārdu, pēdējo modifikācijas datumu, nodošanas termiņu - un datus par augšupielādēto failu – ID, izveides laiku, lietotāja vārdu un uzvārdu, failu, atzīmi un pogu “View”.

Tabula 2.10. Vērtējumu apskatīšanās funkcijas apraksts

Nosaukums	Vērtējumu apskatīšana
Identifikators	KR_04
Mērķis	
	Funkcija ļauj studentam apskatīt savus vērtējumus.
Piekluve	
	Autorizētam lietotājam ar lomu – students.
Ievade	
	Lietotājs nospiež uz sadaļu “Courses” nospiež pogu “Learn more” uz kursu, kura atzīmi vēlās apskatīt.
Apstrāde	
	Tiek iegūti dati par kursu no datu bāzes.
Izvade	
	Lietotājs redz savu atzīmi par izvēlēto kursu.

2.3.3 Lietotāju modulis

Tabula 2.11. E-pasta un telefona numura rediģēšanās funkcijas apraksts

Nosaukums	E-pasta un telefona numura rediģēšana
Identifikators	LM_01
Mērķis	
Funkcija ir izmantota, lai lietotāji varētu paši rediģēt savu e-pastu un telefona numuru.	
Piekluve	
Funkcija pieejama autorizētiem studentiem un pasniedzējiem.	
Ievade	
Lietotājs nospiež uz pogas “Edit” pie attiecīgā lauka un maina e-pastu/telefona numuru. Tad nospiež pogu “Save”, lai apstiprinātu, vai pogu “Cancel”, lai atceltu.	
Apstrāde	
<ol style="list-style-type: none">1. Ja tika nospiesta poga “Save”, tad e-pasts/telefona numurs tiek atjaunināts datubāzē.2. Ja tika nospiesta poga “Cancel” e-pasts/telefona numurs netiek mainīts.	
Izvade	
Tiek izvadīts paziņojums LPI_03 un lietotājs redz jaunu e-pastu/telefona numuru savā kontā. Ja notika kļūda, tiek izvadīts kļūdas paziņojums LPI_04.	

Tabula 2.12. Personīgo datu apskatīšanās funkcijas apraksts

Nosaukums	Personīgo datu apskatīšana
Identifikators	LM_02
Mērķis	
Funkcija paredzēta lietotājam personīgo datu apskatīšanai.	
Piekluve	
Autorizētam studentam un pasniedzējam.	

Ievade
Lietotājs nospiež uz sadaļas “Profile”.
Apstrāde
No datu bāzes tiek iegūti autorizēta lietotāja dati.
Izvade
Lietotājs redz savu vārdu, uzvārdu, lomu, personas kodu, e-pastu, telefona numuru un vecumu.

2.3.4 Administratora modulis

Tabula 2.13. Studentu norīkošanās programmai funkcijas apraksts

Nosaukums	Studentu norīkošana programmai
Identifikators	AM_01
Mērķis	
Funkcija ļauj studentu pievienot noteiktai programmai.	
Pieklūve	
Tikai administratoram.	
Ievade	
Administrators atver sadaļu “User manager”, nospiež uz lietotāju no saraksta, izvēlās programmu no nolaižamas izvēlnes un nospiež pogu “Assign to program”.	
Apstrāde	
Dati tiek nosūtīti uz datu bāzi.	
Izvade	
Lietotājs tiek norīkots uz programmu un tiek izvadīts paziņojums API_01. Ja notika kļūda, tad tiek izvadīts paziņojums API_02.	

Tabula 2.14. Studentu un pasniedzēju norīkošanās kursam funkcijas apraksts

Nosaukums	Studentu un pasniedzēju norīkošana kursam
Identifikators	AM_02
Mērķis	
	Funkcija ļauj pievienot studentu un pasniedzēju noteiktajam kursam.
Piekluve	
	Tikai administratoram.
Ievade	
	Administrators izvēlas kursa nosaukumu no izvēlnes saraksta un atzīmē tabulā no visiem lietotājiem tos, kurus vēlas norīkot izvēlētajam kursam. Nospiež uz pogu “Assign to Course”.
Apstrāde	
	Dati tiek nosūtīti uz datu bāzi.
Izvade	
	Izvēlētais kurss tiek atspoguļots lietotāja kursu sarakstā, kad viņš ir autorizēts savā kontā.

Tabula 2.15. Lietotāja pārvaldīšanās funkcijas apraksts

Nosaukums	Lietotāju pārvaldīšana (CRUD)
Identifikators	AM_03
Mērķis	
	Funkcija ļauj administratoram pievienot, dzēst, apskatīt un rediģēt lietotājus.
Piekluve	
	Tikai administratoram.
Ievade	
	<p>1. Lai pievienot lietotāju: Atver sadaļu ar nosaukumu “Users manager” un aizpilda attiecīgus ievadlaukus:</p> <ol style="list-style-type: none"> 1. Vārdu - obligāti; 2. Uzvārdu - obligāti; 3. Personas kodu - obligāti;

<ol style="list-style-type: none"> 4. E-pastu - neobligāti; 5. Telefona numuru - neobligāti; 6. Vecumu - neobligāti - vesels skaitlis <p>Nospiež pogu “Add User”.</p> <p>2. Lai apskatīt lietotāju datus, atver sadaļu ar nosaukumu “Users manager”.</p> <p>3. Lai rediģēt lietotāja datus, atver sadaļu ar nosaukumu “Users manager”, nospiež uz rediģēšanas ikonas pie attiecīgā lietotāja un aizpilda visus ievadlaukus zem formas “Edit user”</p> <p>4. Lai dzēst lietotāju, atver sadaļu ar nosaukumu “Users manager”, nospiež uz rediģēšanas ikonas pie attiecīgā lietotāja un nospiež uz pogu “Delete user”.</p>
<p>Apstrāde</p> <ol style="list-style-type: none"> 1. Tiek pārbaudīts, vai visi lauki tika aizpildīti. Dati tiek nosūtīti uz serveri, kur lietotājam tiek pievienots unikāls identifikācijas numurs un lietotājs tiek saglabāts datubāzē. Tiek ģenerēta saite lietotāja reģistrēšanai. 2. No datu bāzes tiek iegūti visu lietotāju dati. 3. Dati tiek nosūtīti uz datu bāzi. 4. Datubāzē lauka “isDeleted” vērtība kļūst 0.
<p>Izvade</p> <ol style="list-style-type: none"> 1. Ja tika aizpildīti visi lauki un ievērotas prasības, tiek izvadīt paziņojums API_03, lietotājs tiek veiksmīgi saglabāts datubāzē un ir redzams visu lietotāju sarakstā. Reģistrēšanas saite ir pievienota lietotāju datu bāzei. Ja tika ievadīts personas kods, kas jau eksistē datu bāzē, tiek izvadīts kļūdas paziņojums API_04. 2. Administrators redz tabulu ar visu lietotāju datiem. 3. Ja tika ievadīti visi lauki tiek izvadīts paziņojums API_05 un lietotāja dati tiek atjaunināti. Citādi, tiek izvadīts kļūdas paziņojums API_06. 4. Tiek izvadīts paziņojums API_07 un lietotājs vairs nav redzams visu lietotāju sarakstā. Citādi, tiek izvadīts kļūdas paziņojums API_08.

Tabula 2.16. Lomas piešķiršanās lietotājam funkcijas apraksts

Nosaukums	Lomas piešķiršana lietotājam
Identifikators	AM_04
Mērķis	
Funkcija ļauj administratoram piešķirt vai rediģēt lomu(-s) lietotājam.	
Piekluve	
Tikai administratoram.	
Ievade	
Administrators atver sadaļu “Users manager”, nospiež uz rediģēšanas ikonas pie attiecīgā lietotāja, ar ķeksīti atzīmē lomas un nospiež “Add role”.	
Apstrāde	
Dati tiek nosūtīti uz datu bāzi.	
Izvade	
Lietotājam ir piešķirta loma(-s).	

Tabula 2.17. Paroles atiestatīšanās funkcijas apraksts

Nosaukums	Paroles atiestatīšana
Identifikators	AM_05
Mērķis	
Funkcija ļauj atiestatīt lietotājam paroli, gadījumā, ja viņš ir to aizmirsis.	
Piekluve	
Saites nosūtīšana - tikai administratoram; paroles maiņa ejot pa saiti - lietotājam, kuram tika nosūtīta šī saite.	
Ievade	
Administrators atver sadaļu “Users manager”, nospiež uz rediģēšanas ikonas pie attiecīgā lietotāja un nospiež uz pogas “Reset password”.	
Apstrāde	
Tiek ģenerēta paroles atiestatīšanas saite.	
Izvade	
Tiek izvadīts paziņojums API_23 un izvadīta ģenerētas saites adrese. Administrators nosūta saiti lietotājam, kur lietotājs ievada jaunu paroli un nospiež “Sign up”.	

Tabula 2.18. Kursa pārvaldīšanas funkcijas apraksts

Nosaukums	Kursa pārvaldīšana (CRUD)
Identifikators	AM_06
Mērķis	
Funkcija ļauj pievienot, dzēst, apskatīt un rediģēt kursu.	
Piekluve	
Tikai administratoram.	
Ievade	
<p>1. Lai pievienot jauno kursu, atver sadaļu “Courses manager” un obligāti ievada:</p> <ol style="list-style-type: none"> 1. Vārdu; 2. Aprakstu; 3. Kredītpunktus - vesels skaitlis; 4. Publisko ID; 5. Attēla URL - neobligāti. <p>Nospiež pogu “Add Course”.</p> <p>2. Kursu apskatīšanai atver sadaļu “Courses manager”.</p> <p>3. Lai rediģētu kursu atver sadaļu “Courses manager”, nospiež uz rediģēšanas ikonas pie attiecīga kursa, aizpilda visus ievadlaukus un nospiež uz pogas “Edit course”.</p> <p>4. Lai dzēst kursu atver sadaļu “Courses manager”, nospiež uz rediģēšanas ikonas pie attiecīga kursa un nospiež uz pogas “Delete course”.</p>	
Apstrāde	
<ol style="list-style-type: none"> 1. Dati tiek nosūtīti uz datu bāzi. 2. No datu bāzes tiek iegūti visi kursi 3. Dati tiek nosūtīti uz datu bāzi. 4. Kurss ir dzēsts no datu bāzes. 	
Izvade	
<ol style="list-style-type: none"> 1. Ja tika aizpildīti visi lauki tiek izvadīts paziņojums API_11. Citādi tiek izvadīts kļūdas paziņojums API_12. 2. Administrators redz kursu sarakstu ar datiem. 	

3. Ja tika ievadīti visi lauki tiek izvadīts paziņojums API_13 un kursa dati tiek atjaunināti. Citādi, tiek izvadīts kļūdas paziņojums API_14.
4. Tiek izvadīts paziņojums API_15 un kurss vairs nav redzams visu kursu sarakstā. Ja radījās kļūda, tiek izvadīts kļūdas paziņojums API_16.

Tabula 2.19. Programmu pārvaldīšanās funkcijas apraksts

Nosaukums	Programmu pārvaldīšana (CRUD)
Identifikators	AM_07
Mērķis	
Funkcija ir paredzēta lai varētu pievienot, dzēst, apskatīt un rediģēt programmas.	
Piekluve	
Tikai administratoram.	
Ievade	
1. Lai pievienot jauno programmu, atver sadaļu “Programs manager” un obligāti ievada: <ol style="list-style-type: none"> Nosaukums - obligāti; Semestris - obligāti - vesels skaitlis; Nospiež pogu “Add Program”.	
2. Programmas apskatīšanai atver sadaļu “Programs manager”.	
3. Lai rediģētu programmu atver sadaļu “Programs manager”, nospiež uz rediģēšanas ikonas pie attiecīgas programmas, aizpilda visus ievadlaukus un nospiež uz pogas “Edit program”.	
4. Lai dzēst programmu atver sadaļu “Programs manager”, nospiež uz rediģēšanas ikonas pie attiecīgas programmas un nospiež uz pogas “Delete program”.	
Apstrāde	
<ol style="list-style-type: none"> Dati tiek nosūtīti uz datu bāzi. No datu bāzes tiek iegūti visas programmas. Dati tiek nosūtīti uz datu bāzi. Programma ir dzēsta no datu bāzes. 	
Izvade	

1. Ja tika aizpildīti visi lauki tiek izvadīts paziņojums API_17. Citādi tiek izvadīts kļūdas paziņojums API_18.
2. Administrators redz programmas sarakstu ar datiem.
3. Ja tika ievadīti visi lauki tiek izvadīts paziņojums API_19 un programmas dati tiek atjaunināti. Citādi tiek izvadīts kļūdas paziņojums API_20.
4. Programma vairs nav redzama visu programmu sarakstā un tiek izvadīts paziņojums API_21. Ja radījās kļūda tad tiek izvadīts kļūdas paziņojums API_22.

2.4 Nefunkcionālās prasības

Sistēmas kvalitatīvai darbībai ir nepieciešams atbalstīt arī nefunkcionālas prasības, kuras ir aprakstītas šajā nodaļā.

2.4.1 Lietotāja saskarnes prasības

- Sistēmai jābūt lietotājiem viegli izmantojamai un intuitīvai.
- Sistēmas ievadlaukiem, pogām un attēliem ir jābūt skaidriem un lietotājām saprotamiem.
- Sistēmas ekrāniem un sastāvdaļām jāpielāgojas ierīču izmēriem un jāatspoguļo atsaucīgie elementu izmēri.
- Sistēmai jāatpazīst lietotāju pieskārienus un klikšķus.
- Kļūdu paziņojumiem jābūt skaidriem un īsiem, lai lietotājs saprastu savu nepareizu darbību un izlabotu.

2.4.2 Pieejamības prasības

- Tīmekļa vietnei jābūt pieejamai dažādos interneta pārlūkos - Google Chrome, Mozilla Firefox, Microsoft Edge, Safari.
- Sistēmai jāstrādā tādās operētājsistēmās kā Windows, Linux, macOS.
- Sistēmas pārvaldei ir pieeja tam, kuram ir izstrādes kods un piekļuve datubāzei "AcademicFlow".

2.4.3 Drošības prasības

- Dati ir jāuzglabā datubāzē.
- Tiesības atjaunināt paroli ir tikai administratoram, kas nosūta saiti lietotājam, lai viņš ievadītu jaunu paroli.
- Parolēm ir jāuzglabājas šifrētā veidā autorizācijas datu krātuvē.
- Lietotājam autorizējoties vietnē, lietotājvārdam un parolei nav jābūt redzamiem pārlūkprogrammas rīkos. Tos ir jāšifrē un jānodod serverim, kas atšifrēs datus atpakaļ teksta formātā, lai salīdzinātu ar datubāzē glabātiem datiem.
- Datu pārraidei jābūt kodētai pēc SSL standarta, izmantojot HTTPS protokola palīdzību.
- Ir jānodrošina, lai neautorizēts lietotājs nevarētu piekļūt savam vai cita darbinieka kontam, ejot pa tā saites adresi, nospiežot pārlūkprogrammas pogu “Atpakaļ” vai pārvietojoties pa pārlūkprogrammas vēsturi.

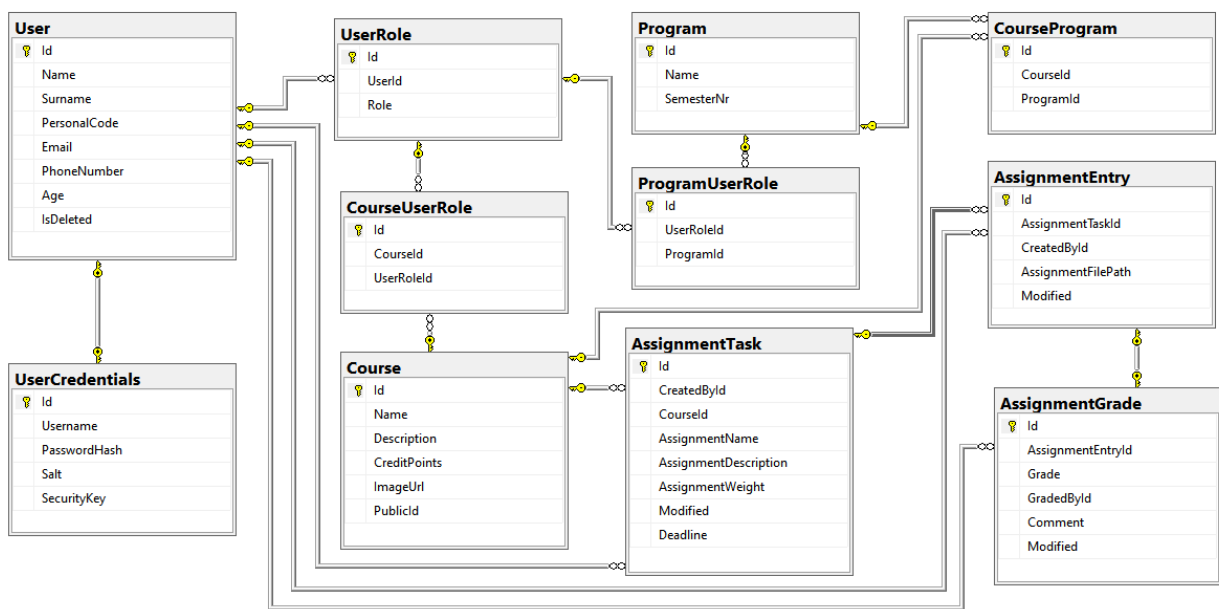
3. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

Darbs izstrādāts “ASP.NET” vidē, kas izmanto kopā HTML5, CSS, JavaScript, un C# valodu sistēmas izstrādei, lai pēc vajadzības būtu arī iespēja piepildīt sistēmu ar jaunām funkcijām un vienlaikus veidot drošu un gatavu lielumam lietotāju skaitam programmatūru. “ASP.NET” arī atbalsta MVC programmatūras dizaina modeļa izmantošanu, kas bija izmantots failu strukturēšanai sistēmas izstrādē kopā ar “Razor Pages”, kas ļauj veidot kvalitatīvu programmkodu pēc “DRY” principa un “Blazor” - izmantojot C# valodu interaktīvai datu veidošanai lietotājiem. Galvenokārt tika izmantota JavaScript scriptu valoda sistēmas funkciju programmēšanā, izmantojot gatavas bibliotēkas - jQuery un jQuery-UI - tika izmantots lietotāja saskarnes izstrādei, nodrošinot dinamisku un lietotājam draudzīgu pieredzi. Šī kombinācija ļauj ātri un efektīvi izstrādāt atsaucīgu saskarni, uzlabojot lietotāju pieredzi.

Datu glābšanai un mijiedarbībai tiek izvēlēts “Microsoft SQL Server”, jo tā sniedza stabilu un drošu datu glabāšanas risinājumu, kas labi integrējas ar “ASP.NET” tehnoloģijām [15].

3.1 Datu bāzes projektējums

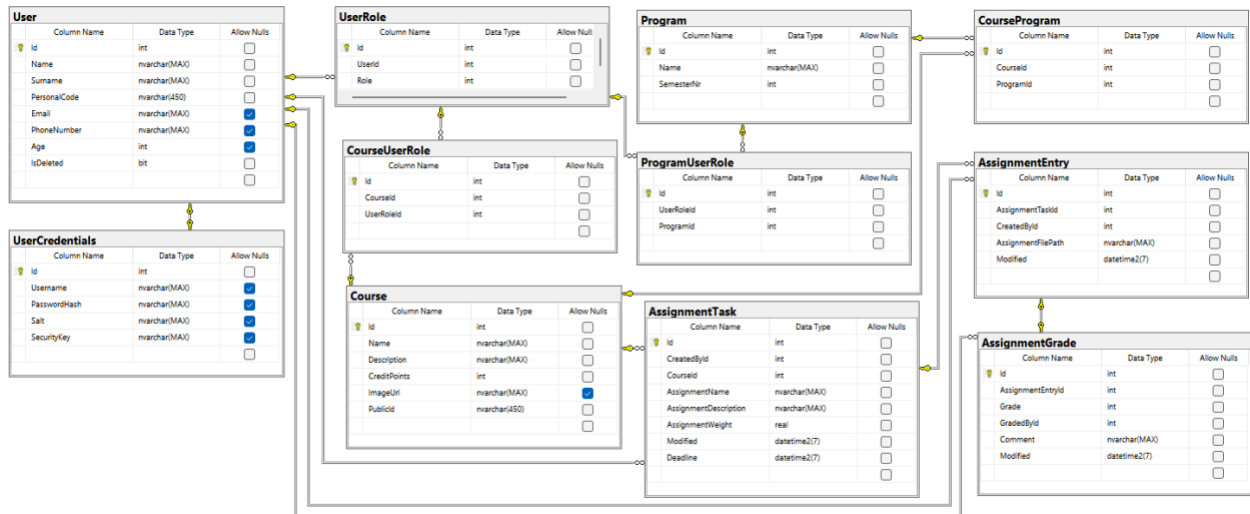
3.1.1 Loģiskais datu bāzes modelis



3.1.att. Loģiskais datu bāzes modelis

Loģiskā datubāzes modelī ir redzamas izveidotas 11 entītijas un to atribūti.

3.1.2 Fiziskais datu bāzes modelis

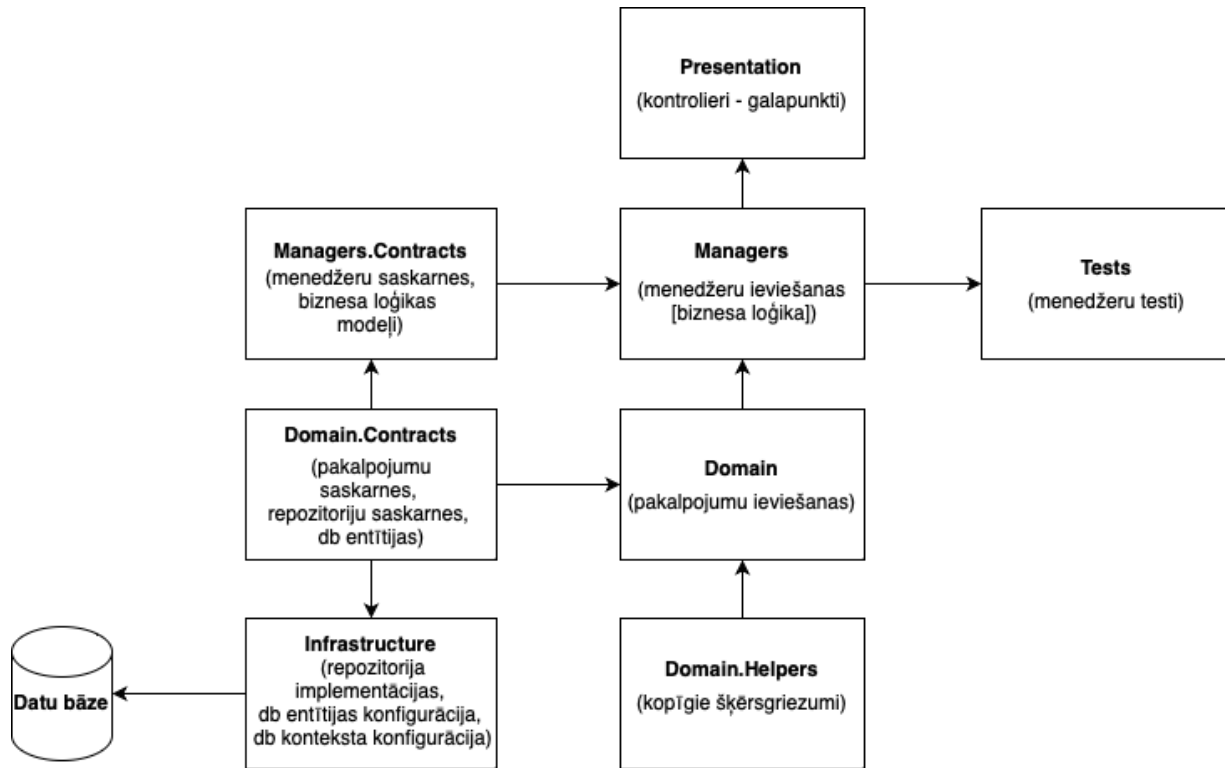


3.2.att. Fiziskais datu bāzes modelis

Fiziskā datubāzes modelī var redzēt katras tabulas atribūta datu tipu un to obligātumu.

3.2 Funkciju projektējums

3.2.1 Atkarību diagrammas projektējums



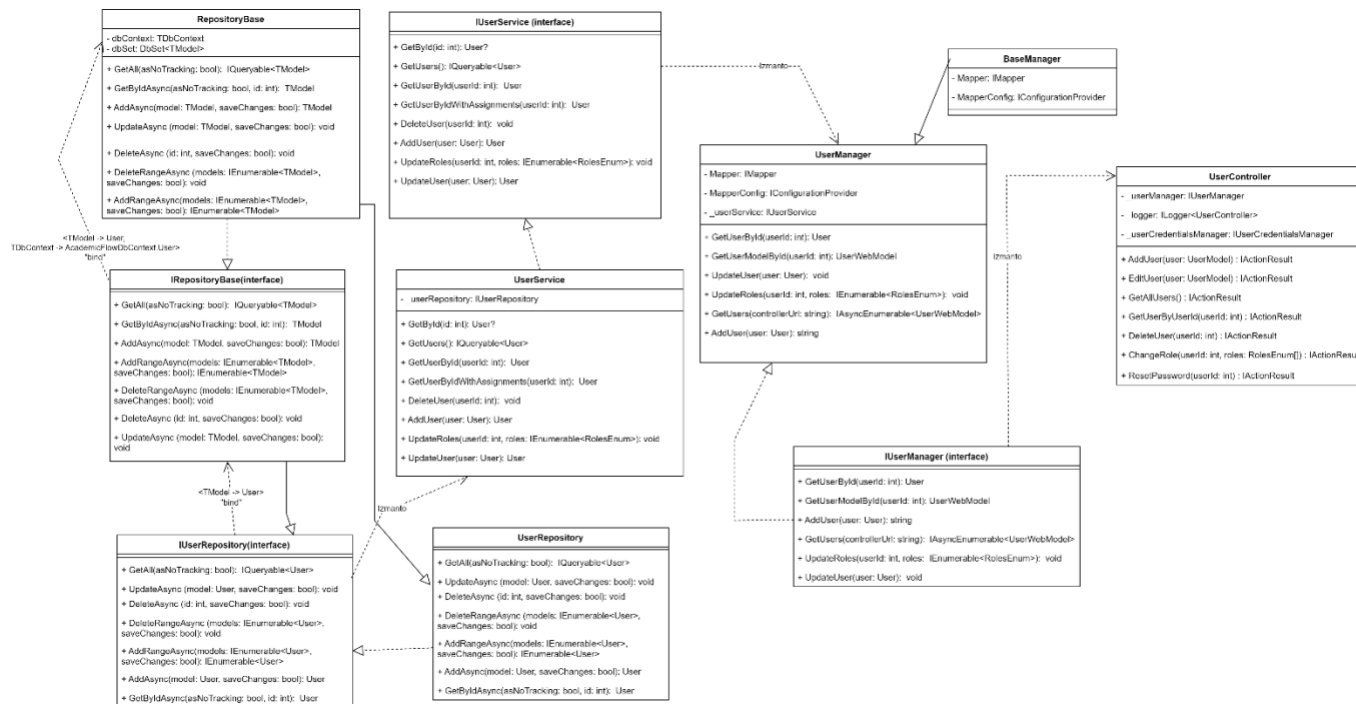
3.3.att. Atkarību diagramma

Repository: Datu bāzes piekļuves slānis. Katrai db entītijai tiek izmantots atsevišķs repozitorijs.

Service: Tiek izmantots, lai entītijām pievienotu pielāgotu loģiku, piemēram, mainītu modificēto parametru katru reizi, kad entītijā tiek atjaunināta vai pārbaudīta, vai tā ir derīga. Var injicēt vairākas krātuves.

Manager: Tiek izmantots, lai pievienotu biznesa loģiku. Var injicēt vairākus pakalpojumus.

3.2.2 Kļau diagrammu projektējums

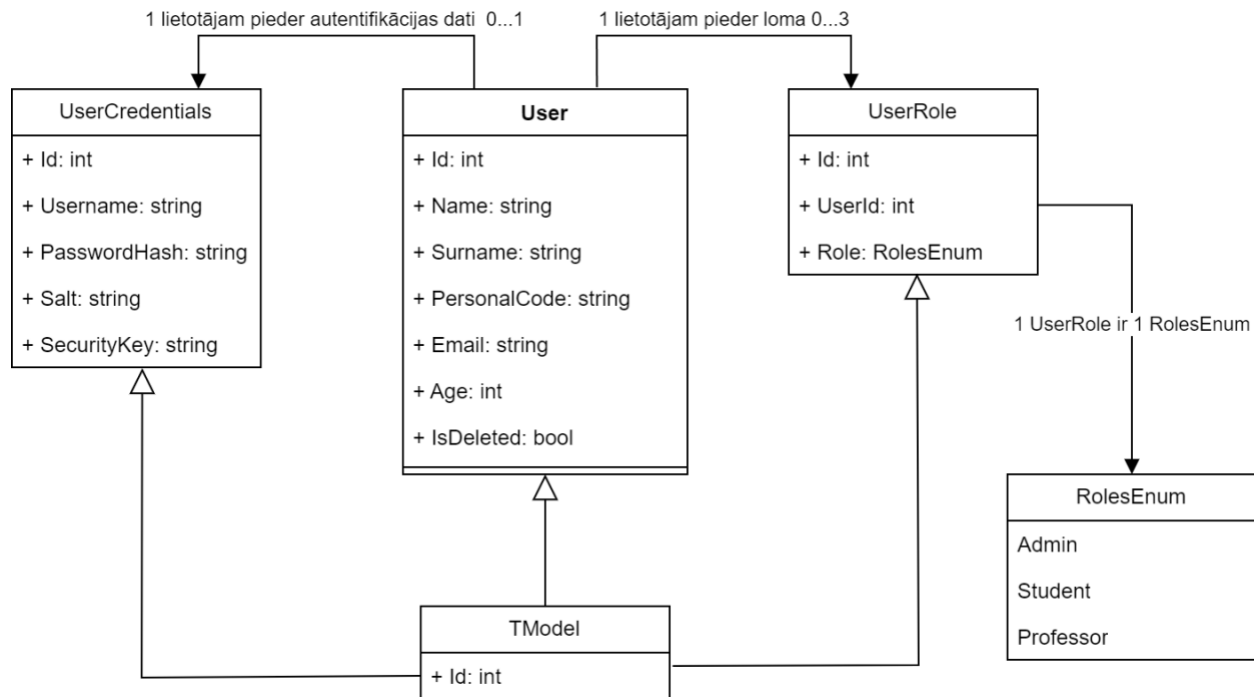


3.4.att. Kļau diagramma

3.4.attēlā kļau diagramma apraksta kontroliera ieviešanu, izmantojot piemēru "UserController". Projektam ir četri slāņi:

- **Prezentācijas slānis:** Šajā slānī atrodas kontrolleri, piemēram, UserController.
- **Aplikācijas slānis:** Šajā slānī atrodas menedžeri un to saskarnes. Diagrammā ir redzami IUserManager un UserManager, kā arī BaseManager. Visa biznesa loģika ir izvietota šajā slānī.
- **Domēna slānis:** Šis slānis nodrošina datus menedžeriem. Tajā tiek glabāti servisi un to saskarnes, kā arī repozitoriju saskarnes. Domēna līmenis var tikt izmantots jebkurā projekta daļā. Šeit ietilpst IUserService, IUserRepository un UserService.
- **Infrastruktūras slānis:** Šis ir zemākais slānis, kurā ievieš piekļuves metodes datu bāzē, piemēram, UserRepository, kā arī bāzes repozitorijs, kas kalpo kā vecāks visiem pārējiem

repozitorijiem. Šāda slāņu arhitektūra veicina koda modularitāti, uzturējamību un mērogojamību.

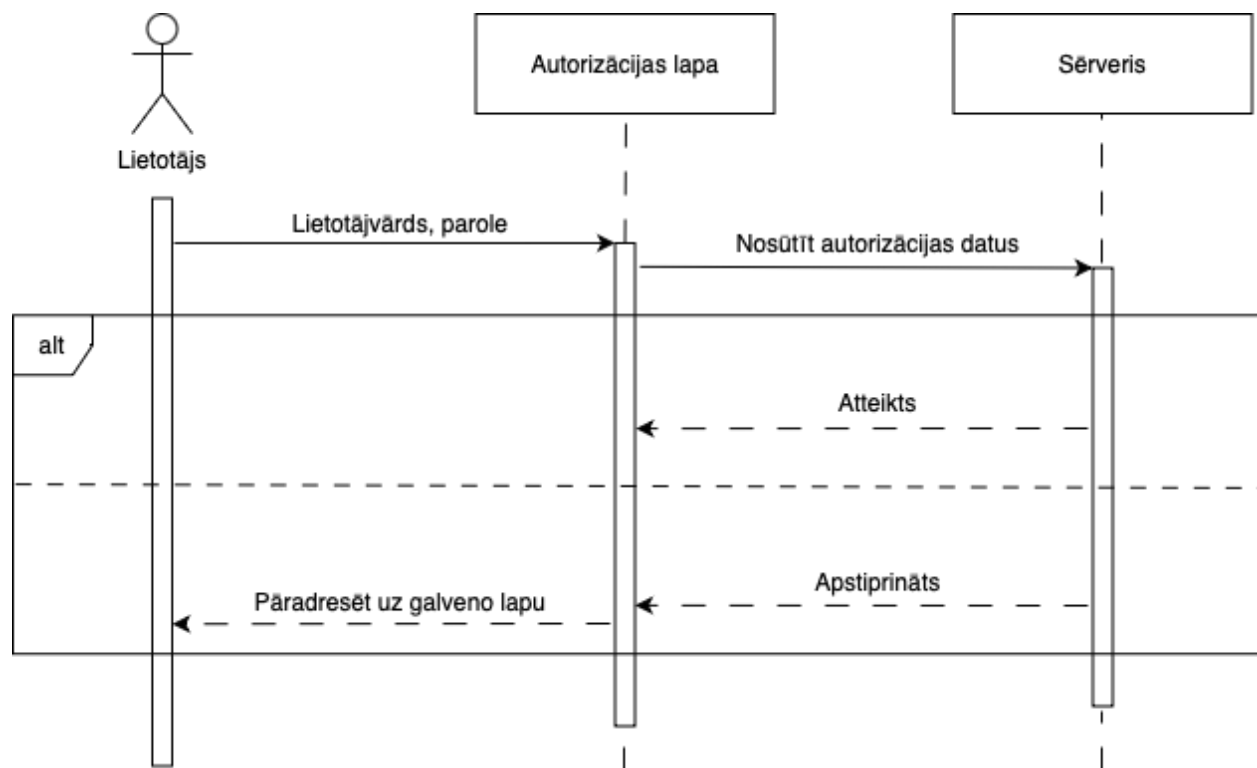


3.5.att. "User class dependencies" klašu diagramma

3.5. attēlā diagrammā parādīta Lietotāja klase un no tās atkarīgās klases.

- 1) Katram lietotājam var būt autorizācijas datu kopa.
- 2) Katram lietotājam var būt vairākas lomas, bet ne vairāk kā trīs.
- 3) Lietotāju lomas tiek norādītas, izmantojot uzskaitījumu (**RolesEnum**), kas var būt administrators, students vai profesors.
- 4) **TModel** klase ir bāzes modeļa klase.

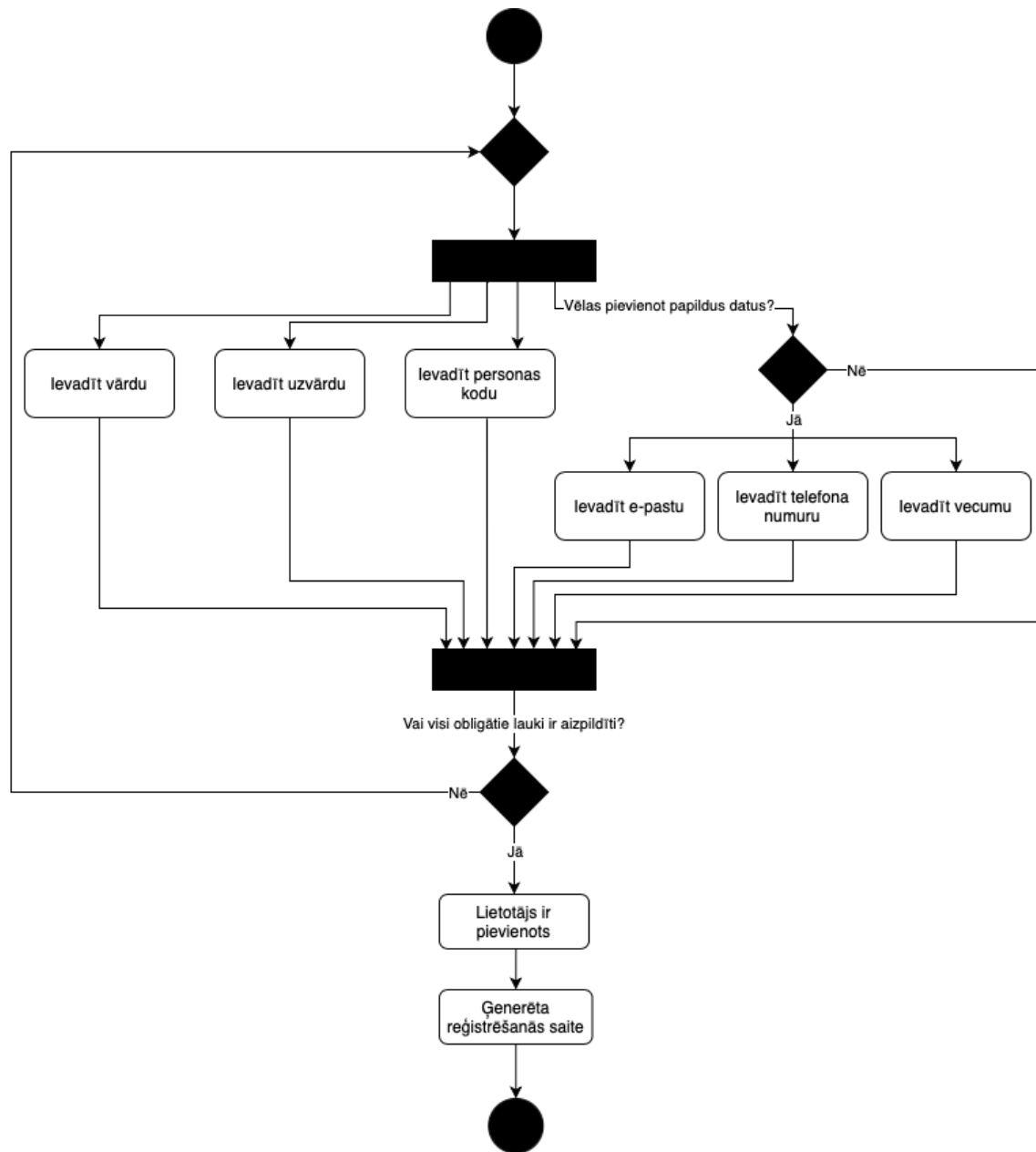
3.2.3 Funkcijas LOG_IN projektējums



3.6.att. Funkcijas “Pieslēgšanās sistēmai” secību diagramma

Autorizācijas lapā tiek ievadīti lietotājvārds un parole un tad autorizācijas dati tiek nosūtīti uz serveri. Ja dati bija ievadīti nepareizi, tad ir atteikta pieslēgšanās sistēmai. Ja dati tiek ievadīti pareizi, tad serveris apstiprina un pāradresē uz galveno lapu.

3.2.4 Funkcijas AM_03(1) projektējums



3.7.att. Funkcijas “Lietotāja pievienošana” aktivitāšu diagramma

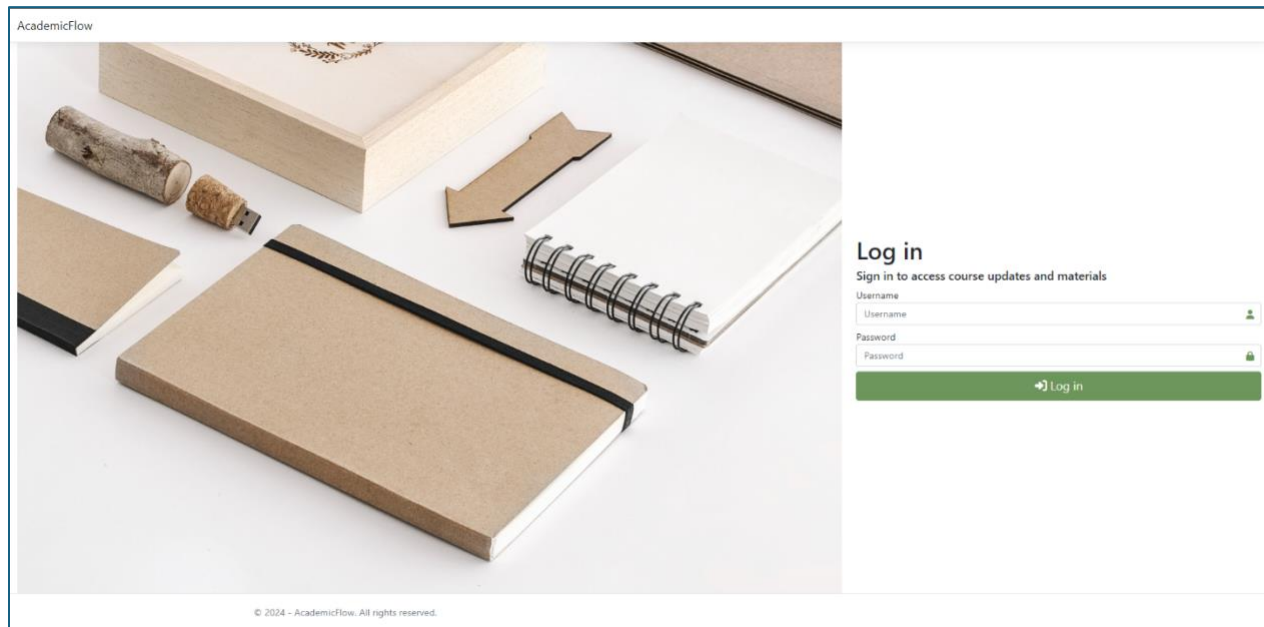
Administratoram obligāti jāievada jauna lietotāja vārdu, uzvārdu un personas kodu. Ja vēlas tad var pievienot papildus datus: e-pasts, telefona numurs, vecums. Tad iet pārbaude vai visi obligātie lauki ir aizpildīti. Ja visi lauki ir aizpildīti, tad lietotājs tiek pievienots un uzģenerēta saite lietotāja reģistrēšanai. Citādi administratoram vajag aizpildīt atlikušos laukus.

3.3 Lietotāja saskarņu projektējums

Šajā sadaļā tiek aprakstītas lietotāju saskarņu projektējums un tiek apkopoti saskarņu ekrāni.

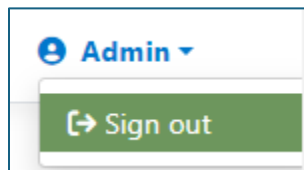
3.3.1 Lietotāja autorizācijas ekrāns

“AcademicFlow” pilnvērtīgai izmantošanai visiem lietotājiem ir nepieciešams autorizēties ar lietotāja vārdu un paroli.



3.8.att. Lietotāja autorizācijas ekrāns

Lai lietotājam atslēgtos no konta ir novietots atslēgšanas izvēlnes rīks galvenē zem lietotāja vārda:



3.3.2 Administratora centrs

Administratora centrā var tikt pievienota un rediģēta informācija par lietotājiem, kursiem un programmām.

Lietotāja rediģēšanas sadaļā var pievienot jaunus lietotājus “Add New User” kolonnā. “Assign Users to Course” sadaļā var izvēlēties pieejamus kursus, izvēlēties lietotāja līmeni un, izvēlējoties no lietotāju saraksta vajadzīgus lietotāju tie tiks pievienoti kursam, nospiežot pogu “Assign to course”. Arī ir iespēja pievienot lietotājus programmai: “Assign Users to Program” sadaļā, izvēlējoties programmu no saraksta, izvēlējoties no lietotāju saraksta vajadzīgus lietotāju tie tiks pievienoti kursam, nospiežot pogu “Assign to program”. Pievienojot kursu lietotājam tiks automātiski pievienoti visi kursi šajā programmā.

(Svarīgi! Kursam var būt pievienoti tikai studenti!)

The screenshot shows the 'AcademicFlow Admin Center' interface. On the left is a sidebar with 'Admin Center' and 'Users Manager' (selected). The main area is titled 'Users Manager' and contains three sections: 'Add New User' with form fields for Name, Surname, Username, Personal Code, Email, Phone Number, and Age; 'Assign Users to Course' with a course selector (DatZ3050: Algoritmu teorija) and a role selector (Student); and 'Assign Users to Program' with a program selector (6: Programinženierija). Below these is a table titled 'All Users' showing a list of users with columns for ID, Name, Surname, Personal Code, Email, Phone Number, Age, Registration Link, Role, and Actions. The table lists 6 users, including Admin, Sergey Ozols, Alina Kasparina, Marija Jisonova, Nikola Tesla, and Renat Ligovins.

ID	Name	Surname	Personal Code	Email	Phone Number	Age	Registration Link	Role	Actions
1	Admin	Admin	000000-00000	adm@adm.adm	0000000	-1		Admin,Student,Professor	
2	Sergey	Ozols	071299-10786	s.ozol@gmail.com	25262398	22		Professor	
3	Alina	Kasparina	150998-20785	alina.kasparina@inbox.lv	26859764	23		Student	
4	Marija	Jisonova	211101-21548	mjison@gmail.com	23658965	22		Student	
5	Nikola	Tesla	180697-15148	t.nikola@inbox.lv	26548978	28		Student	
6	Renat	Ligovins	160800-	lig.ren@ren.com	26559856	24		Student	

3.9.att. Administratora centra lietotāja rediģēšanas sadaļa

AcademicFlow Admin Center

Admin Center

Users Manager

Courses manager

Programs manager

Courses manager

Add Course

Name

Enter course name

Description

Enter course description

Credit Points

Enter course credit points

Public Id

Enter course public id

Image Url

Enter course image url

Add Course

All Courses

10 entries per page

Showing 1 to 3 of 3 entries

ID	Course Title	CP	Course ID	Course Description	Image	Actions
1	Algoritmu teorija	2	DatZ3050	Šis nav kurss par to, kā taisīt labus algoritmus, bet par to, ka dažiem uzdevumiem algoritms NEEKSISTĒ.	herolmg.svg	
2	Objektorientētā programmēšana	2	DatZ4019-LV	Kursa "Objektorientētā programmēšana" mērķis ir pielietot objektorientēto pieeju gan uzdevumu risinājuma projektēšanā, gan realizēšanā.	herolmg.svg	
7	test2	4	test2	test2		

10 entries per page

Showing 1 to 3 of 3 entries

3.10.att. Administratora centra kursa rediģēšanas sadaļa

Nospiežot uz pildspalvas ikonas tiks atvērts lietotāja rediģēšanas logs.

(Svarīgi! Lai rediģētu lietotāju ir nepieciešams aizpildīt visas ievadlaukus!)

AcademicFlow Admin Center

Admin Center

Users Manager

Courses manager

Programs manager

Programs Manager

Add Program

Name

Enter program name

Semester Number

Enter program number

Add Program

Assign Programs to Course

Select course:

Assign to course

All Programs

10 entries per page

Showing 1 to 8 of 8 entries

ID	Title	Semester Number	Actions
1	test	2	
4	Programminženierija	1	
5	Programminženierija	2	
6	Programminženierija	3	

3.11.att. Administratora centra programmas rediģēšanas sadaļa

Kursa rediģēšanas sadaļā var pievienot kursus “Add course” kolonnā. “All courses” kolonnā var apskatīties visus pieejāmus kursus.

3.3.3 Lietotāja profils

Hello, Alina Kaspariņa!

Alina Kaspariņa
Student

Personal code:

E-mail: [Edit](#)


Phone number: [Edit](#)


Age:


AcademicFlow Courses Assignments Profile Hello, Līga Sakmouča! [Līga](#)


[Home](#)
[My courses](#)
[Assignments](#)


Student courses


**Algoritmu teorija**
DatZ3050
Šis nav kurss par to, kā taistīties labus algoritmus, bet par to, ka dažiem uzdevumiem algoritms NEEKSISTĒ.
[Learn more](#)

**Objektorientētā programmēšana**
DatZ4019-LV
Kursa "Objektorientētā programmēšana" mērķis ir pielietot objektorientēto pieeju gan uzdevumu risinājuma projektēšanā, gan realizēšanā.
[Learn more](#)

**Datu bāzu praktiskums**
DatZ3048
Apgūt darbu ar MS SQL Server Management Studio
[Learn more](#)

**Timekļa dizaina pamati**
DatZ1000-LV
Studiju kursa mērķis ir teorētisko pamatzināšanu un praktisko iemaņu apguvi timekļa vizuālā dizaina veidošanā
[Learn more](#)

**Matemātikas pamatjēdzieni**
Mate4005
Kursa mērķis: iepazīstināt ar dažādu matemātikas nozaru pamatjēdzieniem, to savstarpējām sakārnēm un lietojumiem.
[Learn more](#)

**Datorprogrammu testēšanas pamati**
TKI-Istqb1



Algoritmu teorija

2 KP

Šis nav kurss par to, kā taisīt labus algoritmus, bet par to, ka dažiem uzdevumiem algoritms NEEKSISTĒ.

Grade

Grade: 10/10

Assignments

Id	Title	Description	Deadline
1	New important task	Very important!	05.06.2024 10:00:00
2	New	New	05.06.2024 11:53:00

- Home
- My courses
- Assignments

Student assignments

List of student assignments.

Student courses

Algoritmu teorija

Id	Title	Description	Deadline
1	New important task	Very important!	05.06.2024 10:00:00
2	New	New	05.06.2024 11:53:00

Objektorientētā programēšana

No assignments available.

Datu bāzu prakstisms

Id	Title	Description	Deadline
3	New Task	New	05.06.2024 23:44:00

Time's dīcaina cāmati

No assignments available.

Matemātikas cāmati

No assignments available.

New

Task: "New"

Author: Name Surname

Assignment Weight: 0.5

Last modified: 03.06.2024 23:29:36

Deadline: 05.06.2024 11:53:00

Your submitted assignment

Id	Created	User	File	Grade	Action
2	03.06.2024 23:30:59	Liga Sakimoviča	Untitled Diagram.drawio.png	10/10	View

Your Assignment

Download assignment: [Untitled Diagram.drawio.png](#)

Created: 03.06.2024 23:30:59

Grade

Grade: 10/10

Comment: "Viss ir labi!"

Delete assignment entry

4. PROGRAMMATŪRAS TESTĒŠANA

Sistēmas programmatūras testēšanas process bija veikts ar katru pabeigto moduļa veidošanu, pārbaudot arī saistītas ar moduļu sistēmas sastāvdaļas. Sistēmas izstrādājuma testēšanas process notika no 30.04.2024 līdz 05.06.2024 testējot katru jauno funkcionalitāti un pārbaudot sistēmas kopējo darbību.

Testēšanas process bija sadalīts pa moduļiem, un tad atsevišķi pa funkcijām, izmantojot vienībtestēšanu kā galveno testēšanas principu, veicot manuālu testēšanu.

Testēšanas laikā kopā ar vispārīgo moduļa un funkciju pārbaudījumu bija veikta nefunkcionālo prasību paralēla testēšana un pārbaude.

4.1 Testēšanas žurnāls

Testēšanas žurnāls tiek sadalīts pa moduļiem, aprakstot testējamo informāciju kopā ar sagaidītiem testēšanas rezultātiem.

Tabula 4.1. Testēšanas rezultātu apraksts

Manager	Datums	Testa nosaukums	Sagaidāmais rezultāts	Testēšanas rezultāts
UserManager	31.05.2024 (Atkārtoti: 05.06.2024)	GetUsers_ShouldReturnUsers()	Tiek iegūti visi lietotāji no datubāzes	Neizpildās
	31.05.2024 (Atkārtoti: 05.06.2024)	AddUser_ShouldAddUser_IfEverythingFine()	Lietotājs ir pievienots	Izpildās
	31.05.2024 (Atkārtoti: 05.06.2024)	AddUser_ShouldReturnException_IfPersonalCodeIsNotUnique()	Lietotājs nav pievienots	Izpildās
	31.05.2024 (Atkārtoti: 05.06.2024)	DeleteUser_ShouldReturnOk()	Lietotāja vērtība isDelete stājas 0	Izpildās
	31.05.2024 (Atkārtoti: 05.06.2024)	UpdateUser_ShouldReturnOk()	Lietotāja dati ir atjaunināti	Izpildās
ProgramManager	03.06.2024 (Atkārtoti: 05.06.2024)	AddProgramAsync_ShouldReturnInt()	Programma ir pievienota	Izpildās
	03.06.2024 (Atkārtoti: 05.06.2024)	DeleteProgramAsync_ShouldReturnOk()	Programma ir dzēsta	Izpildās

	03.06.2024 (Atkārtoti: 05.06.2024)	GetProgramById Async_ShouldR eturnProgram()	Ir atgriezti dati par noteikto programmu	Izpildās
	03.06.2024 (Atkārtoti: 05.06.2024)	UpdateProgram Async_ShouldR eturnOk()	Programmas dati ir atjaunināti	Izpildās
	03.06.2024 (Atkārtoti: 05.06.2024)	GetProgramTabl eItemList_Shoul dReturnIEnumer able()	Atgriež sarakstu, kas satur datus par programmām	Izpildās
	03.06.2024 (Atkārtoti: 05.06.2024)	GetProgramUser s_ShouldReturnI Enumerable()	Atgriež programmas lietotāju sarakstu	Izpildās
CourseManager	05.06.2024	AddCourseAsyn c_ShouldReturnI nt()	Kurss ir pievienots	Izpildās
	05.06.2024	DeleteCourseAs ync_ShouldRetu rnOk()	Kurss ir dzēsts	Izpildās
	05.06.2024	GetCourseById Async_ShouldR eturnCourse()	Ir atgriezti dati par noteikto kursu	Izpildās
	05.06.2024	UpdateCourseAs ync_ShouldRetu rnOk()	Kursa dati ir atjaunināti	Izpildās
	05.06.2024	GetCourseTableI temList_Should	Atgriež sarakstu, kas satur datus par kursiem	Izpildās

		ReturnIEnumerableable()		
	05.06.2024	GetCourseUsers_ShouldReturnIEnumerableable()	Atgriež kursa lietotāju sarakstu	Izpildās
	05.06.2024	GetCoursePrograms_ShouldReturnIEnumerableable()	Atgriež sarakstu, kas satur programmas, kas ir saistītas ar konkrētu kursu	Izpildās

5. PROJEKTA ORGANIZĀCIJA

Sistēmas izstrāde tika veikta grupā, apkopojot un noformulējot no pamata darījumprasības, funkcijas aprakstu, kopējus cēloņus un mērķi sistēmas izstrādē, veidojot arī vizuāla izskata dizaina variācijas, testējot izstrādāto produktu un dokumentējot veiktās darbības.

Ņemot vērā sistēmas izstrādes plānoto sarežģītību un prasības, kā arī izstrādes iepriekš minētas darbības procesa aprakstu, sistēmas izstrādei tika pieņemts izmantot “Agile” programmēšanas izstrādes paņēmieni, veidojot izmaiņas jau izstrādātam posmam izstrādes procesā, piedāvājot pielāgojamību. Sistēmas izstrādē tika sadalīta lomās - 3 cilvēki bija atbildīgie par backend, 3 par frontend un 1 par dokumentācijas rakstīšanu. Lielākais sistēmas funkcionalitātes apspriedums bija veikts dizaina un struktūras plānošanas laikā.

Sistēmas testēšana bija veikta pēc katra pabeigta sistēmas sastāvdaļas (moduļa) izstrādes, pēc tam veikta sistēmas atklūdošana un kopēja sistēmas testēšana ar visiem saistītiem ar konkrēto moduļu sistēmas sastāvdaļām.

Sistēmas izstrādes darba sākumā, pēc saņemtiem funkcijas definīcijām, bija veikta aptuvena darba plānošana, pēc kura bija izveidots izstrādes procesa metode un darba struktūra.

5.1 Darbietilpības novērtējums

Sistēmas izstrādes darbietilpības novērtējumā lielu lomu ņēma darba autoru pieredze, strādājot ar “ASP.NET” izstrādājumu izveidošanu.

Tabula 5.1. Koda rindiņu novērtējums pa moduļiem tabula

Izstrādāta moduļa nosaukums	Aptuvenais rindiņu skaits
Controllers	999
Managers	656
Tests	638
Services	179
Contracts	419
Kopā	2891

Sistēmas izstrādē piedalījās 7 programmētāji. Darba stundu ieguldījums rēķinot ar 40 stundām no katra dalībnieka ir 280 stundas kopā.

QSM sniegtie vēsturiskie dati tika izmantoti, lai pārbaudītu darbietilpības novērtējumu. Biznesa sistēmu ieviešanas statistika liecina, ka aptuveni 965 stundu projektam lieluma mediāna ir 1889 rindas. $965/280 = 3,45$ reizes. Sistēmas izstrādē jābūt $1889/3,45 = 548$ rindas.

Pamatojoties uz faktiem un reāliem darbietilpības datiem no līdzīgiem projektu scenārijiem mēs pamatojam, ka sistēmas izstrādē ir sasniegts vajadzīgs stundu skaits.

5.2 Projekta kvalitātes nodrošināšana

Sistēmas kvalitātes nodrošināšana tiek panākta, izmantojot dažādas metodoloģijas un principus. Viens no būtiskākajiem aspektiem ir MVC programmatūras projektējuma struktūra. Kas atdala lietotāja saskarni, datus un lietojumprogrammas loģiku (kontrolieri), kas efektīvi organizē kodu, sadalot to trīs loģiskās sastāvdaļās, nodrošinot labāku saprotamību, uzturēšanu un paplašināmību, veicinot sistēmas ilgtspējīgu attīstību.

Principa "DRY" ievērošana ir vēl viens svarīgs elements, kas bija izmantots kvalitātes nodrošināšanā. Tas nozīmē, ka izstrādājot kodu, jāizvairās no koda dublēšanas, jo tā var izraisīt nepieciešamību veikt izmaiņas vairākos vietējos punktos, kas palielina kļūdu risku sistēmas apvienoto darbību. Tā vietā tiek veidoti atkārtoti izmantojami komponenti, funkcijas un klasi. Ar šo uzdevumu tika izmantoti "ASP.NET" vides izstrādes aspekti.

Koda izstrādē tika lietots vairāk abstrakciju, lai būtu mazāk koda rindu. Kods bija īsi un precīzi komentēts, lai ir ērtāk izprast kodu gan esošajiem izstrādātājiem, gan tiem, kuri strādās ar kodu turpmāk.

Kopumā, šie principi un prakses veicina sistēmas kvalitātes uzlabošanu, nodrošinot efektīvu un viegli izstrādājamu programmatūru.

5.3 Konfigurāciju pārvaldība

Sistēmas izstrādes konfigurāciju pārvaldībai tika izmantota "Git" pārvaldības sistēma, kas bija atjaunota ar katru funkcionalitātes pievienošanu. Lai vienlaikus efektīvi atklūdotu funkcijas, neietekmējot avota kodu, projekts tika sazarots, un pabeidzot noteikto funkciju, izmaiņas tika sapludinātas projekta avota kodā.

Ar “Git” versiju pārvaldību bija atvieglots sistēmas izstrādes atklūdošanas procedūra, veidojot nepamanāmus no pirmā skata loģikas kļūdas, kas ļauj efektīvi pārvaldīt un monitorēt konfigurāciju izmaiņas.

Sistēmas faili tika strukturēti tā, lai sistēma būtu viegli uzstādāma jaunajā vidē, kā arī tiešsaistes datu glābšana tika izvēlēta drošības dēļ.

Sistēmas dokumentācija tika saglabāta tiešsaistē koplietojamā Microsoft Word dokumentā, kas bija pieejams tikai komandas dalībniekiem.

REZULTĀTI

Kursa grupas darba ietvaros tika izstrādāta funkcionējoša tīmekļa vietne, kuru var pilnvērtīgi lietot, ka arī izveidota sistēmas dokumentācija, kas labi apraksta un shematiski apzīmē vairākas sistēmas galvenās funkcijas, prasības un izstrādes gaitu.

Sistēmas izstrādes procesā tika pastiprinātas zināšanas darbam ar “ASP.NET” izstrādes vidi, tostarp ar iekļautam vidē funkcionalitātēm, kā “Razor Pages”. Tika veidotas stipras zināšanas darba izstrādes procesa plānošanā, pielietojot izmantotas tehnoloģijas lietojumprogrammas izstrādē, kas palīdzes ar nākotnes projektu plānošanā un izstrādē ar līdzīgiem tehnoloģijām. Pienākumu sadalījums pa komandas dalībniekiem attīstīja organizatoriskas, komunikācijas un laika plānošanas prasmes.

Tika pastiprinātas zināšanas konfigurāciju pārvaldībā, izmantojot pirmkoda bezmaksas atvērtā koda versiju pārvaldības sistēmu “Git”. Bija viegli saglabāt izstrādes strādājošu versiju, veidojot lokālas izmaiņas un saglabājot tikai pārbaudīto funkcionalitāti. Arī veikt versijas atgriešanu, ja tika veikti lielie sistēmas labojumi un bija konstatētas nesavienojamas funkciju darbības ar korektu sistēmas darbību, bija viegli un ātri, izmantojot “Git”.

SECINĀJUMI

Sistēmas izstrāde veidoja nepieciešamas prasmes grupas darbam ar sistēmas izstrādi, kas vērtīgā ar 40 * 7 cilvēkmēnešiem paveiktu darbu, kas sevī ietver darba plānošanu un darba procesa organizēšanu, un paveikta darba dokumentēšanu. Šīs prasmes ir nepieciešamas kvalitatīvas sistēmas izstrādē.

Kopumā, izstrādājot šo dokumentāciju un lietojumprogrammatūru grupas darbam, autori ieguva būtiskas prasmes un pieredzi. Darbā pie sistēmas izveides grupas dalībnieki attīstījusi programmēšanas, dizaina un datu bāzu pārvaldības prasmes.

Dokumentācijas izstrādē autori papildināja prasmes strukturēt informāciju loģiski, izmantot diagrammas un rakstīt skaidrus un saprotamus aprakstus. Šī pieredze ir bagātinājusi autoru profesionālās prasmes un iedvesmojusi turpināt pilnveidoties jomā. Īpaši svarīga bija spēja organizēt darbu, un veikt efektīvu problēmas risināšanas procesu.

Kopumā, šī projekta un dokumentācijas izstrāde bijusi izaicinoša un apmierinoša pieredze, kas papildināja mūsu izglītību un sagatavoja profesionālajai izaugsmei nākotnē.

IZMANTOTĀ LITERATŪRA UN AVOTI

Grāmatu avoti:

-“LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis”.

Pieejams: No “DatZ2072-LV : Programminženierija” kursa avotiem.

-“LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai”.

Pieejams: No “DatZ2072-LV : Programminženierija” kursa avotiem.

Tiešsaistes avoti:

-GIT, GIT. About” [Tiešsaiste]. Pieejams: <https://git-scm.com/about>.

-JavaScript, “MDN WEB Docx. JavaScript” [Tiešsaiste]. Pieejams:

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>.

-JQuery, “jQuery. About” [Tiešsaiste]. Pieejams: <https://jquery.com/>.

-jQuery UI, “jQuery UI. About” [Tiešsaiste]. Pieejams: <https://jqueryui.com/about/>.

-MVC, “Overview of ASP.NET Core MVC” [Tiešsaiste]. Pieejams:

<https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0>

-QSM, “Software Project Performance Benchmark Tables” [Tiešsaiste]. Pieejams:

<https://www.qsm.com/resources/qsm-benchmark-tables>.

-Resursu pakas, “Provide localized resources for languages and cultures in an ASP.NET Core app” [Tiešsaiste]. Pieejams: [https://learn.microsoft.com/en-](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/localization/provide-resources?view=aspnetcore-8.0)

[us/aspnet/core/fundamentals/localization/provide-resources?view=aspnetcore-8.0](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/localization/provide-resources?view=aspnetcore-8.0).

-SQL Server, “What is SQL Server?” [Tiešsaiste]. Pieejams: [https://learn.microsoft.com/en-](https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16)

[us/sql/sql-server/what-is-sql-server?view=sql-server-ver16](https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16)

-SQL Server un ASP, “How to connect a .NET 6 Web App to SQL Server” [Tiešsaiste].
Pieejams: [https://medium.com/@mickyangelod/connecting-net-6-web-app-to-sql-server-](https://medium.com/@mickyangelod/connecting-net-6-web-app-to-sql-server-fccf26ad0aed)

[fccf26ad0aed](https://medium.com/@mickyangelod/connecting-net-6-web-app-to-sql-server-fccf26ad0aed).

-W3C, “W3C Web Standards” [Tiešsaiste]. Pieejams: <https://www.w3.org/standards/about/>.