

LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE



SKOLĒNU SEKMJU ANALĪZES PLATFORMAS IZSTRĀDE

KVALIFIKĀCIJAS DARBS

Autors: **Matīss Zalāns**

Studenta apliecības numurs: mz15012

Darba vadītājs: B. E-kom. Ģirts Ļaudaks

RĪGA 2017

ANOTĀCIJA

Kvalifikācijas darbā “Skolēnu sekmju analīzes platformas izstrāde” ir aprakstīta platformas moduļu izstrāde. Šī kvalifikācijas darba mērķis ir izstrādāt Ekool sistēmas sekmju analīzes platformu, kurā lietotāji var aplūkot tiem ievadītās atzīmes, to detaļas un statistikas attēlojumu diagrammās. Izstrādātā platforma ļauj lietotājam ātri un ērti aplūkot savas atzīmes un dažādos griezumos salīdzināt tās ar klasesbiedriem. Produkts ir paredzēts integrēšanai Ekool sistēmā un šobrīd ir pieejams testa vidē.

Projekta sākumā prasības netika stingri definētas, līdz ar to platforma tika izstrādāta ievērojot spējās izstrādes vadlīnijas. Darba ietvaros tika izstrādāta platformas servera puses funkcionalitāte. Programmas moduļi tika izstrādāti PHP programmēšanas valodā, par pamatu izmantojot “Model-View-Controller” strukturēšanas principu Laravel ietvarā.

Atslēgas vārdi: Laravel, Spējā izstrāde, Analīze, Statistika, PostgreSQL, MVC

ABSTRACT

Student grade analysis platform development

Qualification work “Student grade analysis platform development” describes platform’s modules development. The purpose of qualification work is to develop grade analysis platform for Ekool system, where users can see them registered grades, their details and representation of statistical charts. The platform is designed for integration in Ekool system and is now available in test environment.

At the beginning of the project requirements were not strictly defined, so the platform was developed in accordance with agile development guidelines. In work time was developed platform’s server-side functionality. Program modules were developed in PHP programming language on the basis of “Model-View-Controller” structuring principle using Laravel framework.

Keywords: Laravel, Agile Development, Analysis, Statistics, PostgreSQL, MVC

SATURS

APZĪMĒJUMU SARAKSTS	7
1. IEVADS	8
2. PROJEKTĒJUMA PRASĪBU APRAKSTS	9
2.1. Ievads	9
2.1.1. Nolūks	9
2.1.2. Darbības sfēra	9
2.1.3. Definīcijas, akronīmi un saīsinājumi	9
2.1.4. Saistība ar citiem dokumentiem	9
2.1.5. Pārskats	9
2.2. Vispārējais apraksts	10
2.2.1. Produkta perspektīva	10
2.2.2. Produkta funkcijas	10
2.2.3. Sistēmas lietotāji	10
2.2.4. Lietotāja raksturiezīmes	11
2.2.5. Vispārējie ierobežojumi	11
2.2.6. Pieņēmumi un atkarības	11
3. LIETOTĀJSTĀSTI	12
3.1. Ievads	12
3.1.1. Nodaļas nolūks	12
3.2. Lietotājstāstu sadalījums pa iterācijām	13
3.2.1. 0. iterācija	13
3.2.2. 1. iterācija – lietotāja identificēšana, autentificēšana un autorizācija	13
3.2.3. 2. iterācija – sekmju attēlošana, sekmju attēlošana noteiktā priekšmetā	16
3.2.4. 3. iterācija – sekmes informācijas attēlošana	17
3.2.5. 4. iterācija – sekmju reitinga un dinamikas attēlošana	19
3.2.6. 5. iterācija – lokalizācijas maiņa (valodas maiņa)	22
4. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	23
4.1. Dekompozīcijas apraksts	23
4.1.1. Sistēmas arhitektūra	23
4.1.2. Modeļu dekompozīcija	24
4.1.3. Kontrolieru apraksts	24
4.1.4. Repozitoriju apraksts	25
4.2. Atkarību apraksts	27
4.3. Datu bāzes tabulu apraksts	27

4.3.1. Tabula "person"	28
4.3.2. Tabula "role"	28
4.3.3. Tabula "premium_role"	29
4.3.4. Tabula "premium_recurring"	29
4.3.5. Tabula "grade_type"	30
4.3.6. Tabula "term"	30
4.3.7. Tabula "parallel_identifier"	30
4.3.8. Tabula "grade_event"	30
4.3.9. Tabula "class_level"	31
4.3.10. Tabula "todo"	31
4.3.11. Tabula "periods_group_class_level"	32
4.3.12. Tabula "grade_value"	32
4.3.13. Tabula "journal_event"	33
4.3.14. Tabula "study_year"	33
4.3.15. Tabula "journal"	33
4.3.16. Tabula "course"	34
4.3.17. Tabula "curriculum"	34
4.3.18. Tabula "subject"	34
4.3.19. Tabula "periods_group"	35
4.4. Datu plūsmu diagrammas	35
4.4.1. 0. līmeņa datu plūsmu diagramma	35
4.4.2. 1. līmeņa datu plūsmu diagramma	36
4.4.3. Autorizācijas un validācijas modulis - 2. līmeņa datu plūsmas diagramma	37
4.4.4. Lietotāja modulis - 2. līmeņa datu plūsmas diagramma	38
4.4.5. Lokalizācijas modulis - 2. līmeņa datu plūsmas diagramma	39
4.4.6. Periodu modulis - 2. līmeņa datu plūsmas diagramma.....	40
4.4.7. Priekšmetu modulis - 2. līmeņa datu plūsmas diagramma	41
4.4.8. Atzīmju modulis - 2. līmeņa datu plūsmas diagramma	42
5. TESTĒŠANAS DOKUMENTĀCIJA	43
5.1. Ievads	43
5.2. Akcepttesti	43
6. PROJEKTA PARVALDĪBA.....	49
6.1. Projekta organizācija	49
6.2. Kvalitātes nodrošināšana	49
6.3. Konfigurāciju pārvaldība	50
7. DARBIETPLĪBAS NOVĒRTĒJUMS	51
8. LIETOTĀJA EKRĀNA FORMU APRAKSTS.....	54

SECINĀJUMI.....	60
IZMANTOTĀ LITERATŪRA UN AVOTI.....	61
PIELIKUMS.....	62

APZĪMĒJUMU SARAKSTS

PHP – atklātā pirmkoda skriptu valoda, kura sākotnēji bija paredzēta servera puses lietojumos dinamisku tīmekļa lapu ģenerēšanai.

Laravel – PHP ietvars.

HTML – iezīmēšanas valoda, kas ir izstrādāta tīmekļa lappušu un citas pārlūkprogrammā attēlojamās informācijas glabāšanai.

CSS – īpaša stila lapas valoda, ko lieto, lai aprakstītu izskatu iezīmēšanas valodā veidotiem dokumentiem.

JavaScript – skriptu valoda, kas balstīta uz prototipu koncepta. Valoda pamatā ir pazīstama pēc to izmantošanas tīmekļa vietnēs izpildīšanai uz klienta datora.

jQuery – starpplatformu JavaScript bibliotēka, kas izstrādāta, lai vienkāršotu HTML klientpuses skriptošanu.

jQuery SnackBar – jQuery spraudnis.

Bootstrap – atvērtā koda saskarnes ietvars tīmekļa vietņu dizaina un funkcionalitātes paplašināšanai.

Moment JS – JavaScript bibliotēka.

Knockout JS – JavaScript bibliotēka, kas realizē Model-View-View-Model arhitektūru.

Chartist JS – JavaScript bibliotēka.

SCRUM – iteratīvs spējās izstrādes paveids.

gitLab – tīmeklī bāzēta konfigurācijas sistēma.

camelCase – programmatūras koda noformēšanas prakse.

snake_case – programmatūras koda noformēšanas prakse.

NginX – viegls tīmekļa serveris, starpniekserveris un e-pasta starpniekserveris.

PostgreSQL – bezmaksas objektu relāciju datubāzes pārvaldības sistēma.

iFrame – iezīmēšanas valodas metode.

1. IEVADS

Mūsdienās visas skolas jau cenšas izmantot tehnoloģiju sniegtās iespējas, lai daudz efektīvāk un ātrāk organizētu skolas darbu un saskarsmi ar skolēniem. Skolas sāk izmantot elektroniskās dienasgrāmatas un skolu sistēmas, tādējādi visai skolai, skolotājiem, skolēniem un vecākiem nepieciešamā informācija tiek uzskaitīta sistēmā, lai to nepieciešamības gadījumā izmantotu atbilstošajai situācijai. Skolēni var aplūkot savu dienasgrāmatu un savas sekmes, bet ne visās elektroniskajās skolu sistēmās ir lietotājam ērta, pašsaprotama un draudzīga saskarne.

Galvenie mērķi bija izpētīt eKool datu bāzi un ar atbilstošajām iespējām un datu bāzes informācijas specifiku izveidot lietotājam ērtu un draudzīgu sekmju analīzes platformu.

Tīmekļa platforma “Grade Analytics” ir izstrādāta Igaunijas elektroniskajai skolu sistēmai Ekool. Izstrādātā platforma tiks integrēta Ekool sistēmā. Platforma ļauj lietotājam aplūkot Ekool datu bāzē esošo informāciju par ievietotajām sekmēm, kuras ir saistītas ar autorizēto lietotāju. Lietotājs var aplūkot savas sekmes priekšmetos, lietotāju sekmes, kuri arī ir saņēmuši sekmes kopīgajā uzdevumā. Lietotājs arī var aplūkot savu bērnu sekmes.

Platformas izstrādē piedalījās projekta vadītājs, kurš veicināja projekta virzību un iesaistījās lietotājistāstu izveidē, projekta vadītāja asistents, kurš konsultēja tehniskajos jautājumos un veica koda apskatu, un lietotāju saskarnes izstrādātājs.

Platformas izstrādē tika izmantotas spējas izstrādes metodes vadlīnijas, jo spējā izstrāde ir piemērotāka šim projektam un arī ir mūsdienīgāka. Spējā izstrāde ļāva programmatūru izstrādāt efektīvāk un īsākā laika periodā, kā arī tas ļāva mainīt un papildināt funkcionalitāti izstrādes laikā.

Versijkontrolei tika izmantota gitLab konfigurācijas sistēma.

Izmantotās tehnoloģijas servera pusē: PHP un Laravel ietvars, SQL.

Izmantotās tehnoloģijas saskarnes pusē: HTML, CSS, Javascript, jQuery, jQuery Sackbar, Bootstrap, Moment JS, Knockout JS, Chartist JS.

2. PROJEKTĒJUMA PRASĪBU APRAKSTS

2.1. Ievads

2.1.1. Nolūks

Izstrādātā programmatūras prasību specifikācija ir paredzēta izstrādājamās platformas, “Grade Analytics”, programmatūras prasību aprakstīšanai, kurā ir apkopotas moduļu prasības, kuras ir sastādītas pēc programmatūras pasūtītāja un izstrādātāja veidotajiem lietotājstāstiem. Šis dokuments ir savstarpēja vienošanās starp pasūtītāju un izpildītāju un ir paredzēts platformas izstrādātājiem platformas izstrādē un uzturēšanā.

2.1.2. Darbības sfēra

Platforma “Grade Analytics” ir Igaunijas elektroniskās skolu sistēmas Ekool sistēmā integrējama platforma, kurai ir piekļuve Ekool datu bāzē esošajiem datiem. Platforma veic skolēnu sekmju analīzi, statistiku un rezultātu attēlošanu. Platforma atvieglo skolēnam aplūkot sev izliktās atzīmes.

2.1.3. Definīcijas, akronīmi un saīsinājumi

Lielākā daļa no šajā dokumentā sastopamo specifisko informācijas tehnoloģiju definīciju un akronīmu ir pieminētas un paskaidrotas šī dokumenta sadaļā “Apzīmējumu saraksts”.

2.1.4. Saistība ar citiem dokumentiem

Šis dokuments ir izstrādāts pamatojoties uz programmatūras izstrādes standarta “LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis” [1. avots] ieteikumiem.

2.1.5. Pārskats

Turpmākajās sadaļās tiks apskatītas izstrādājamās platformas vispārējais apraksts, kurā tiks aprakstīti vispārējie faktori, kuri iedarbojas uz izstrādājamo produktu un tā prasībām. Tiks uzskaitīti lietotājstāsti, apskatīti datu plūsmu diagrammas, datu bāzu projektējumi un akcepttestu rezultāti.

2.2. Vispārējais apraksts

2.2.1. Produkta perspektīva

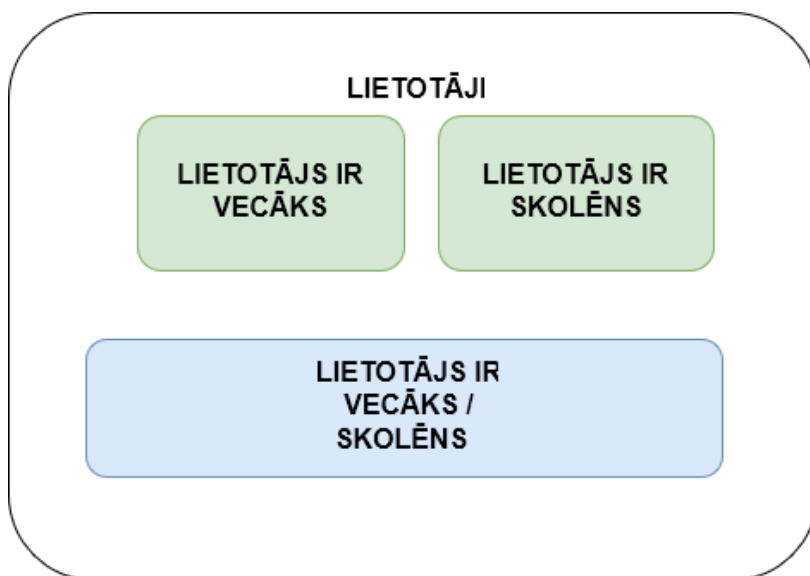
Platforma ir atkarīga, tas nozīmē, ka platforma ir kādas citas sistēmas sastāvdaļa. Platforma tiek izstrādāta ārpus Ekool, bet tiks integrēta Ekool sistēmā un tā izmantos Ekool datu bāzi un sesijas datus, lai pilnvērtīgi veiktu savu funkcionalitāti. Izmantojot Ekool datu bāzes datus, platformas funkcionalitāti ir iespējams papildināt – iekļaut kavējumu analīzi, integrēt dienasgrāmatu, paziņojumus utt.

2.2.2. Produkta funkcijas

Programmatūras produktam pēc pieprasījuma jāapstrādā no datu bāzes saņemtie dati, kuri diagrammas veidā jāattēlo atbilstošajam lietotājam. Lietotājs ir spējīgs aplūkot savu un savu bērnu sekmes un to statistiku. Vispārīgs funkcionalitātes apraksts ir redzams dokumenta sadaļā “Lietotājistāsti”.

2.2.3. Sistēmas lietotāji

Platformas funkcionalitāte ir paredzēta Ekool sistēmas datu bāzē reģistrētiem lietotājiem. Izmantot funkcionalitāti varēs lietotāji, kuri būs veikuši maksājumus. Lietotāji, kuriem arī bērni būs reģistrēti Ekool datu bāzē, varēs aplūkot viņu bērniem pieejamo funkcionalitāti. Vecāki arī var būt skolēni. Visas lietotāju grupas ir attēlotas shēmā (sk. 2.1. att.).



2.1. att. Lietotāju grupas

2.2.4. Lietotāja raksturierzīmes

Platformas lietotājiem nav nepieciešamas īpašas zināšanas. Lietotājiem ir jābūt pamatprasmēm darbā ar datoru - spēja lietot pelīti, klaviatūru un izmantot tīmekļa pārlūkprogrammu. Lietotāja saskarne ir viegli saprotama un intuitīva.

2.2.5. Vispārējie ierobežojumi

Platformas darbināšanai ir nepieciešams interneta savienojums un interneta pārlūkprogramma. Sistēmai jāatbalsta visas populārākās pārlūkprogrammu versijas. Lietotājam nav nepieciešams uzstādīt produktu kā atsevišķu programmatūru, bet jāvar pieslēgties tīmekļa vietnei no populārākajām operētājsistēmām. Platformas lietotājam ir jābūt reģistrētam Ekool sistēmas datu bāzē, lietotājam ir jābūt aktīvam maksājumam. Platforma nenodrošina jauna lietotāja reģistrāciju.

2.2.6. Pieņēmumi un atkarības

Sistēmai jābūt pieejamai visas darbadienas laikā. Sistēmai jādarbojas uz populārākajām operētājsistēmām. Sistēmai vienlaicīgi jāvar pieslēgties vairākiem lietotājiem. Platformas darbībai ir nepieciešams nepārtraukts interneta savienojums. Lietotāja pārlūkprogrammai ir nepieciešams JavaScript atbalsts. Platformas darbībai ir nepieciešams NginX serveris ar uzstādītu PHP.

3. LIETOTĀJSTĀSTI

3.1. Ievads

3.1.1. Nodaļas nolūks

Programmatūra tika izstrādāta izmantojot spējas izstrādes vadlīnijas, tāpēc prasību specifikācija netika veidota, bet tā vietā tika apkopoti lietotājstāsti, kur katrs lietotājstāsts ir novērtēts ar sarežģītības punktiem, kur noteicošais ir uzdevumos ielikto punktu savstarpējā attiecība, nevis skaitlis. Vissarežģītākais un laikietilpīgākais uzdevums ir novērtēts ar vislielāko punktu skaitu. Lietotājstāstu nolūks ir aprakstīt izstrādājamās “Grade Analytics” platformas prasības. Šeit, izmantojot lietotājstāstus, ir definēta platformas funkcionalitāte.

Lietotājstāsti tika apkopoti 5 iterācijās :

0. iterācija – nepieciešamā gatavošanās programmatūras izstrādei, jaunu zināšanu apgūšana un esošo nostiprināšana.

1. iterācija – lietotāja identificēšana, autentificēšana un autorizācija.

2. iterācija – sekmju attēlošana, sekmju attēlošana noteiktā priekšmetā.

3. iterācija – sekmes informācijas attēlošana.

4. iterācija – sekmju reitinga un dinamikas attēlošana.

5. iterācija – lokalizācijas maiņa (valodas maiņa).

Iterāciju aptuveno notikšanas ilgumu un sākuma, beigu laiku var redzēt attēlā (sk. 3.1. att.). Iterācijas nebija strikti atdalītas ar sākuma un beigu datumu, jo regulāri nācās atgriezties uz jau realizētām funkcijām tās papildināt, vai sākt veidot funkcijas no jaunas iterācijas, iepriekšējo vēl līdz galam nepabeidzot.



3.1. att. Iterāciju secība un aptuvenais notikšanas laiks

3.2. Lietotājstāstu sadalījums pa iterācijām.

3.2.1. 0. iterācija

0. iterācija ir nepieciešama, lai kvalitatīvāk sagatavotos izstrādājamās platformas izveidei. Šī iterācija ir viena no vissarežģītākajām, jo ir jānostiprina teorētiskās un praktiskās zināšanas, kā arī jāiegūst jaunas zināšanas. Lielu darba ieguldījumu prasa esošās situācijas analīze, kurā tiek veikta lietotājstāstu izveide un pilnveidošana. Tiek meklētas platformas funkcionalitātes realizēšanai nepieciešamās metodes un tehnoloģijas. Tiek iegūts pamats projekta organizēšanas struktūrai. Ļoti svarīga ir precīza datu bāzes analīze.

3.1. tabula. 0. iterācija

Lietotājstāsts	Uzdevumi	Sarežģītības punkti
Darba vides sagatavošana.	<ul style="list-style-type: none">• Uzstādīt nepieciešamo programmatūru.• Uzstādīt lokālo serveri.• Iegūt piekļuvi datu bāzei.	5
Darbam nepieciešamo zināšanu apgūšana.	<ul style="list-style-type: none">• Analizēt datu bāzi.• Nostiprināt Laravel teorētiskās un praktiskās zināšanas.• Nostiprināt SQL teorētiskās un praktiskās zināšanas.	21

3.2.2. 1. iterācija – lietotāja identificēšana, autentificēšana un autorizācija.

1. iterācijā tiek izveidota pirmā funkcionalitāte, tiek izmantota Ekool datu bāze un sistēma, lai izveidotu jaunu funkcionalitāti, kura ļauj izmantot datu bāzi lietotāja autorizācijai, šajā iterācijā ir ļoti nozīmīgs 0. iterācijā paveiktais.

3.2. tabula. 1. iterācija

Lietotājstāsta NR un lietotājstāsts	Uzdevumi	Sarežģītības punkti
1. Kā lietotājs vēlos autorizēties platformā.	<ul style="list-style-type: none">• Izmantojot datu bāzes lietotāju datus, nodrošināt autorizēšanos platformā.	8

	<p>Pieslēdzoties Ekool sistēmai, ģenerējas sesijas dati. Atverot Ekool sistēmā esošo saiti, lietotājam tiek sniegta šīs platformas funkcionalitāte, kura ir integrēta Ekool sistēmā ar iFrame palīdzību. Platforma saņem no sesijas Ekool sistēmā autorizētā lietotāja id, kuru izmanto kā autorizācijas rīku “Grade Analytics” platformā, izmantojot Ekool datu bāzi. Tiek veikta pārbaude vai lietotājs eksistē datu bāzē. Katram pieprasījumam tiek veikta datu eksistences pārbaude un lietotāja piekļuves atļaujas pārbaude pieprasītajiem datiem – pārbauda vai dati ir pieejami lietotājam, pirms veikt datu apstrādi. Visās datu atlasēs turpmāk par pamatu izmanto autorizēto lietotāja profilu. No kura izriet visi pieejamie dati.</p>	
2. Kā lietotājs vēlos aplūkot savu profila pamatinformāciju.	<ul style="list-style-type: none"> Atlasīt un apstrādāt lietotāja profila pamatinformāciju. <p>Izmantojot autorizēto profilu, datu bāzē tiek atlasīta informācija par lietotāju. Tiek atlasīts vārds, uzvārds, klases. Tiek veikta datu eksistences pārbaude, pēc kuras seko datu apstrāde nepieciešamajā struktūrā, lai to JSON formātā atgrieztu uz saskarnes vidi. Ja lietotājs neeksistē, tiek atgriezts kļūdas paziņojums.</p>	2
3. Kā lietotājs vēlos aplūkot man pieejamās klases un izvēlēties kādu no klasēm	<ul style="list-style-type: none"> Atlasīt un apstrādāt visas lietotāja klases, klases nosaukumu, paralēlklases nosaukumu. 	3

tālākai informācijas atlasei.	Izmantojot lietotāju, tiek atlasītas lietotāja klases, ja lietotājam sistēmā ir reģistrēti arī viņa bērni, tad atlasa arī lietotāja bērnu informāciju un ciklā tiek atlasītas katra bērna klases un atlasītie dati apstrādāti. Datus apstrādā, lai iegūtu pareizu JSON struktūru. Ja netiek atlasītas bērnu klases, tad bērni nerādās, tāpat arī ar lietotāju. Ja netiek atlasītas ne lietotāja klases, ne bērnu klases, tad tiek atgriezts kļūdas paziņojums.	
4. Kā lietotājs vēlos aplūkot un izvēlēties kādu no periodiem tālākai informācijas atlasei.	<ul style="list-style-type: none"> Atlasīt un apstrādāt klasei pieejamos periodus. <p>Izmantojot lietotāju, tiek atlasīti aktīvās klases mācību periodi. Atvēšanas brīdī tiek atlasīti periodi tai klasei, kura ir pirmā klašu sarakstā, kuru iepriekš atiež cita funkcija. Tālāk tiek atlasīti periodi lietotāja izvēlētajai klasei. Aktīvs periods, ir tas, kurš šajā brīdī vēl nav beidzies, ja visi periodi ir beigušies, tad aktīvais periods, ir pirmais periods aktīvajā klasē.</p>	2
5. Kā lietotājs vēlos aplūkot savu bērnu profilus, lai varētu aplūkot viņiem pieejamo informāciju.	<ul style="list-style-type: none"> Atlasīt lietotāja bērnu profilus. <p>Bērnus atlasa, izmantojot lietotāju. Bērni tiek atgriezti iepriekš izsauktā funkcijā. Lietotājs var izvēlēties bērna klasi no pieejamo klašu saraksta. Izvēloties klasi, tālāk var aplūkot bērnam pieejamos datus.</p>	5

3.2.3. 2. iterācija – sekmju attēlošana, sekmju attēlošana noteiktā priekšmetā.

2. iterācijā tiek izstrādātas vienas no svarīgākajām platformas funkcijām – sekmju aplūkošana un analīze.

3.3. tabula. 2. iterācija

Lietotārstāsts	Uzdevumi	Sarežģītības punkti
6. Kā lietotājs vēlos aplūkot savas jaunākās sekmes.	<ul style="list-style-type: none"> Atlasīt jaunākās sekmes. 	3
	Izmantojot lietotāju, tiek atlasītas lietotājam reģistrētās atzīmes, kuras ir pēc izlikšanas datuma ir sakārtotas pēc izlikšanas datuma dilstošā secībā.	
	<ul style="list-style-type: none"> Apstrādāt atlasītās sekmes. 	
	Atlasītās sekmes tiek apstrādātas, lai atgrieztu pareizu JSON struktūru. Ja netiek atlasīta neviena atzīme, tad tiek atgriezts kļūdas paziņojums.	
7. Kā lietotājs vēlos aplūkot sekmes izvēlētajā priekšmetā.	<ul style="list-style-type: none"> Atlasīt visus lietotāja atlasītās klases priekšmetus. 	5
	Izmantojot lietotāju, tam reģistrēto studiju plānu un priekšmetu grupu tiek atlasīti lietotājam reģistrētie mācību priekšmeti.	
	<ul style="list-style-type: none"> Atlasīt visas sekmes lietotāja izvēlētajai klasei un priekšmetam 	
	Izmantojot lietotāju, periodu un priekšmetu. Tiek veikta lietotājam reģistrēto atzīmju atlase, kuras ir izliktas saistītajā priekšmetā. Tiek atlasītas atzīmes ar atzīmju tipu, kurš nav gala atzīmes tips.	
	<ul style="list-style-type: none"> Apstrādāt atlasītos datus. 	

	Iegūtos datus apstrādā un atgriež JSON failu. Ja netiek atrastas atzīmes, tiek atgriezts kļūdas paziņojums.	
8. Kā lietotājs vēlos aplūkot gala atzīmes, izvēlētajā klasē.	<ul style="list-style-type: none"> Atlasīt visas lietotāja sekmes, kuras ir izliktas kādam periodam vai ir kāda cita veida gala atzīme. 	5
	Līdzīgi, kā 7. lietotājstāsta sekmju atlasē, bet lietotājs ir izvēlējis mācību gadu. Tiek atlasītas atzīmes ar atzīmes tipu, kurš gala atzīmes tips.	
	<ul style="list-style-type: none"> Apstrādāt atlasītās atzīmes. 	
	Papildus tiek atlasīti lietotāja klasei reģistrētie mācību periodi. Ar ciklu palīdzību gala atzīmes tiek piesaistītas periodam, kuram atzīme ir izlikta, bet atlikušās atzīmes tiek attēlotas, kā perioda atzīmes vai gala atzīmes tips, ja atzīme nav reģistrēta periodam. Visas gala atzīmes ir reģistrētas kādam priekšmetam, bet nav obligāti reģistrētas periodam. Ja netiek atrasta neviena gala atzīme, tad tiek atgriezts kļūdas paziņojums.	

3.2.4. 3. iterācija – sekmes informācijas attēlošana.

3. iterācijā tiek izstrādāta funkcionalitāte, kura veic izlikto atzīmju informācijas attēlošanu, pārējo klases biedru atzīmes.

3.4. tabula. 3. iterācija

Lietotājstāsts	Uzdevumi	Sarežģītības punkti
9. Kā lietotājs vēlos aplūkot savas sekmes detaļas.	<ul style="list-style-type: none"> Atlasīt sekmes informāciju. 	3
	Izmantojot lietotāju un atzīmi, tiek atlasīta atzīmes informācija, ja atzīme netiek atrasta, tiek atgriezts kļūdas paziņojums.	

	<ul style="list-style-type: none"> • Apstrādāt atlasīto informāciju. 	
	Tiek pārbaudīts vai atlasītā atzīme ir gala atzīme. Tiek atlasīta nepieciešamā informācija un izveidots pareizs JSON fails, kurš tiek atgriezts lietotāja saskarnei.	
10. Kā lietotājs vēlos aplūkot visas sekmes, kuras izliktas uzdevumā.	<ul style="list-style-type: none"> • Atlasīt visas sekmes saistītajā uzdevumā. 	13
	Izmantojot iepriekš atlasīto atzīmi, tiek atlasīts ieraksts mājas uzdevumu vai klases uzdevumu tabulā, atkarīgs no tā, kur atzīme ir reģistrēta. Ja nekur nav reģistrēta, tad pārējo skolēnu atzīmes netiek atlasītas un tiek attēlota tikai lietotāja atzīmes informācija bez diagrammas.	
	<ul style="list-style-type: none"> • Apstrādāt atlasītās sekmes. 	
	Atlasa nepieciešamos atzīmju datus – vērtību, nosaukumu, id.	
	<ul style="list-style-type: none"> • Izmantojot atlasītās sekmes, izveidot diagrammas iestatījumus. 	
	Tiek veikti pārbaudījumi, kuros nosaka kāda ir diagrammas maksimāla vērtība, kuros datos kā diagrammas vērtību izmantot atzīmes vērtību vai nosaukumu.	
	<ul style="list-style-type: none"> • Izmantojot apstrādātos datus, izveidot diagrammu. 	
	Ciklā tiek izveidoti diagrammas attēlojamie dati nepieciešamajā struktūrā, kā arī pārējie diagrammai nepieciešamie iestatījumi. Ja tiek atlasīta viena atzīme, tad attēlo tikai atzīmes informāciju bez diagrammas.	

3.2.5. 4. iterācija – sekmju reitinga un dinamikas attēlošana.

4. iterācijā tiek realizētas vienas no sarežģītākajām un svarīgākajām funkcijām – atzīmju atlase, analīze, statistikas izveide un datu vizuāla attēlošana.

3.5. tabula. 4. iterācija

Lietotārstāsts	Uzdevumi	Sarežģītības punkti
11. Kā lietotājs vēlos aplūkot savas klases reitingu izvēlētajā priekšmetā un periodā.	<ul style="list-style-type: none"> Atlasīt perioda ietvaros sekmes izvēlētajā periodā, kuras drīkst izmantot aprēķinos. 	8
	Izmantojot periodu, priekšmetu un lietotāju tiek atlasītas atzīmes, kuras ir izliktas attiecīgajā periodā un priekšmetā. Atzīmes ir lietotāja klases biedru atzīmes – ir kopīga skola, mācību gads, klases līmenis un paralēlklases identifikators.	
	<ul style="list-style-type: none"> Grupēt datus pa lietotājiem. 	
	Atlasē dati tiek grupēti pa lietotājiem.	
	<ul style="list-style-type: none"> Izmantojot atlasītās sekmes izveidot diagrammas iestatījumus. 	
	Atzīmes tiek pārveidotas uz 100 punktu skalu.	
	<ul style="list-style-type: none"> Izmantojot apstrādātos datus, izveidot diagrammu. 	
	Izmantojot grupētās lietotāju atzīmes, tiek aprēķināta vidējā atzīme lietotājam. Tiek izveidota diagramma un pareizā struktūrā atgriezti dati. Ja nevienam netika atrastas atzīmes, tad atgriež kļūdas paziņojumu, ja sekmes ir saņēmusi pārējā klase, bet ne lietotājs, tad lietotājs arī var aplūkot klases atzīmes. Ja atzīmes ir saņēmis tikai lietotājs, tad diagrammā attēlo tikai lietotāja datus.	

12. Kā lietotājs vēlos aplūkot savas klases reitingu izvēlētajā periodā.	<ul style="list-style-type: none"> Atlasīt sekmes izvēlētajā periodā, kuras drīkst izmantot aprēķinos. 	13
	Līdzīgi iepriekšējam lietotājistāstam. Sekmes tiek atlasītas visos priekšmetos perioda ietvaros.	
	<ul style="list-style-type: none"> Grupēt datus pa lietotājiem un priekšmetiem, un aprēķināt vidējo atzīmi. 	
	Sāukumā katram lietotājam aprēķina vidējo atzīmi katrā priekšmetā un tad aprēķina vidējo atzīmi no priekšmetos izrēķinātajām vidējām atzīmēm. Šī funkcija ir sarežģītāka un lēnāka par iepriekšējā lietotājistāstā minēto sekmju grupēšanu un apstrādi.	
	<ul style="list-style-type: none"> Izmantojot atlasītās sekmes izveidot diagrammas iestatījumus. 	
	Līdzīgi iepriekšējam lietotājistāstam.	
	<ul style="list-style-type: none"> Izmantojot apstrādātos datus, izveidot diagrammu. 	
	Līdzīgi iepriekšējam lietotājistāstam.	
13. Kā lietotājs vēlos aplūkot savas klases sekmju dinamiku izvēlētajā priekšmetā un periodā.	<ul style="list-style-type: none"> Atlasīt sekmes izvēlētajā periodā, kuras drīkst izmantot aprēķinos. 	13
	Līdzīgi iepriekšējā lietotājistāsta datu atlasei.	
	<ul style="list-style-type: none"> Grupēt datus pa lietotājiem un mēnešiem un aprēķināt vidējo atzīmi. 	
	Atlasītie dati ir grupēti pa mēnešiem katram lietotājam. Tiek aprēķināta vidēja atzīme katram lietotājam katrā mēnesī. Ja tiek izvēlēts mācību gads, tad grupēšanā vidējo atzīmi izrēķina katram mācību periodam.	

	<ul style="list-style-type: none"> Izmantojot atlasītās sekmes izveidot diagrammas iestatījumus. 	
	Līdzīgi iepriekšējam lietotājstāstam.	
	<ul style="list-style-type: none"> Izmantojot apstrādātos datus, izveidot diagrammu. Izmantojot apstrādātos datus, izveidot diagrammu. 	
	Līdzīgi iepriekšējam lietotājstāstam. Ja lietotājs nav saņēmis atzīmes kādā mēnesī, tad diagrammā tiek attēlota nulle, tāpat arī ja nav saņēmusi klase.	
14. Kā lietotājs vēlos aplūkot savas klases sekmju dinamiku izvēlētajā priekšmetā un mācību gadā.	<ul style="list-style-type: none"> Atlasīt sekmes izvēlētajā periodā, kuras drīkst izmantot aprēķinos. 	13
	Līdzīgi iepriekšējam lietotājstāstam. Sekmes tiek atlasītas noteiktam mācību priekšmetam visā mācību gada laikā.	
	<ul style="list-style-type: none"> Grupēt datus pa lietotājiem un periodiem, un aprēķināt vidējo atzīmi. 	
	Grupēšana tiek veikta pa mācību periodiem. Vidējā atzīme tiek aprēķināta periodam.	
	<ul style="list-style-type: none"> Izmantojot atlasītās sekmes izveidot diagrammas iestatījumus. 	
	Līdzīgi iepriekšējam lietotājstāstam.	
	<ul style="list-style-type: none"> Izmantojot apstrādātos datus, izveidot diagrammu. 	
	Līdzīgi iepriekšējam lietotājstāstam. Rezultātā lietotājam attēlo vidējo atzīmi katrā periodā, kā arī klases atzīmes.	

3.2.6. 5. iterācija – lokalizācijas maiņa (valodas maiņa).

5. iterācijā tiek veikta mazsvarīgas funkcionalitātes realizēšana. Lietotājam saistošākas saskarnes papildināšana.

3.6. tabula. 5. iterācija

Lietotājistāsts	Uzdevumi	Sarežģītības punkti
15. Kā lietotājs vēlos, lai informācija tiktu pasniegta manam profilam uzstādītajā valodā.	<ul style="list-style-type: none"> Noteikt lietotāja lokalizācijas valodu un to izmantot tulkojumiem. 	5
	Izmantojot lietotāju, tiek atlasīta lietotāja izvēlēta valoda. Ja lietotājs nav savam profilam izvēlējis valodu, tad tiek automātiski izvēlēta un iestatīta noklusētā igauņu valoda. Tulkojumus platformā iestata ar Laravel iebūvētajām funkcijām.	
	<ul style="list-style-type: none"> Noteikt tulkojamās tekstus. 	
	Platformā tika meklēti visi attēlojamie teksti un paziņojumi, kurus ir nepieciešams tulkot un tika sastādīts tulkojamā teksta saraksts.	
	<ul style="list-style-type: none"> Sasaistīt tulkojamās tekstus un tulkojumus. 	
	Nedaudz tika veikta tulkošana. Sistēmā var aplūkot trīs dažādu valodu tulkojumus – angļu, krievu un igauņu valodā.	
	<ul style="list-style-type: none"> Ievietot tulkojumus lietotāju saskarnē. 	
	Izmantojot Laravel iebūvētās funkcijas, tika sasaistīti tulkojumi ar lietotāja saskarni, ja nezināmu iemeslu dēļ netiek atrasts tulkojums, tad tiek automātiski izmantots angļu valodas tulkojums.	

4.1.2. Modeļu dekompozīcija

Programmatūras moduļi ir izstrādāti, izmantojot PHP programmēšanas valodas Laravel ietvaru. Moduļu izstrāde ir veikta pēc MVC pieejas, izmantojot repozitoriju, kurš veic datu atlasīšanu un risina iekšējās funkcijas, bet kontrolieris veic maršrutam tieši piesaistītās lielās pamatfunkcijas. Papildus tiek izmantota starpprocesu pārbaude (middleware), kura saņem sesijas datus un autorizē lietotāju. Tiek izmantota arī maršruta sasaistīšana, ar kuras palīdzību tiek veikta autorizētā lietotāja piekļuves pārbaude pieprasītajiem datiem. Kā arī tiek izmantota autorizācijas polise, kura kontrolē lietotāja autorizācijas pārbaudi.

4.1.3. Kontrolieru apraksts

Tabulā tiek uzskaitīti platformā izmantotie kontrolieri servera pusē, kuri ir saistīti ar maršruta pieprasījumiem.

4.1. tabula. Kontrolieru aprakstu tabula

Kontroliera nosaukums	Apraksts
CourseController	<ul style="list-style-type: none">• Atgriež lietotājam pieejamos mācību priekšmetus un to datus.
GradeEventController	<ul style="list-style-type: none">• Visapjomīgākais kontrolieris, kurā tiek veikta visu sekmju analīze un statistika. Apstrādātos datus atgriež lietotājam.• Atzīmju un to datu attēlošana.• Klases rezultātu attēlošana noteiktā uzdevumā.• Atzīmju attēlošana noteiktā priekšmetā.• Gala atzīmju noteikšana un attēlošana noteiktā priekšmetā.• Klases reitinga attēlošana noteiktā priekšmetā un periodā vai visā mācību gadā.• Klases sekmju dinamikas attēlošana noteiktā priekšmetā un periodā vai visā mācību gadā.
RoleController	<ul style="list-style-type: none">• Atgriež lietotājam viņa datus.• Atgriež lietotājam pieejamās klases, lomas un to datus.• Atgriež lietotājam viņa bērnus un viņu datus.

TermController	<ul style="list-style-type: none"> • Atgriež lietotājam pieejamos mācību periodus (semestrus, trimestrus, periodus, ceturkšņus u.c.) un to datus. • Atgriež lietotājam pieejamo mācību gadu un tā datus.
----------------	--

4.1.4. Repozitoriju apraksts

Tabulā ir attēloti repozitoriji, kuri veic datu atlasu un iekšējās funkcijas, kuras tiek izsauktas kontrolierī, maršrutu sasaistīšanā, starpprocesu pārbaudē.

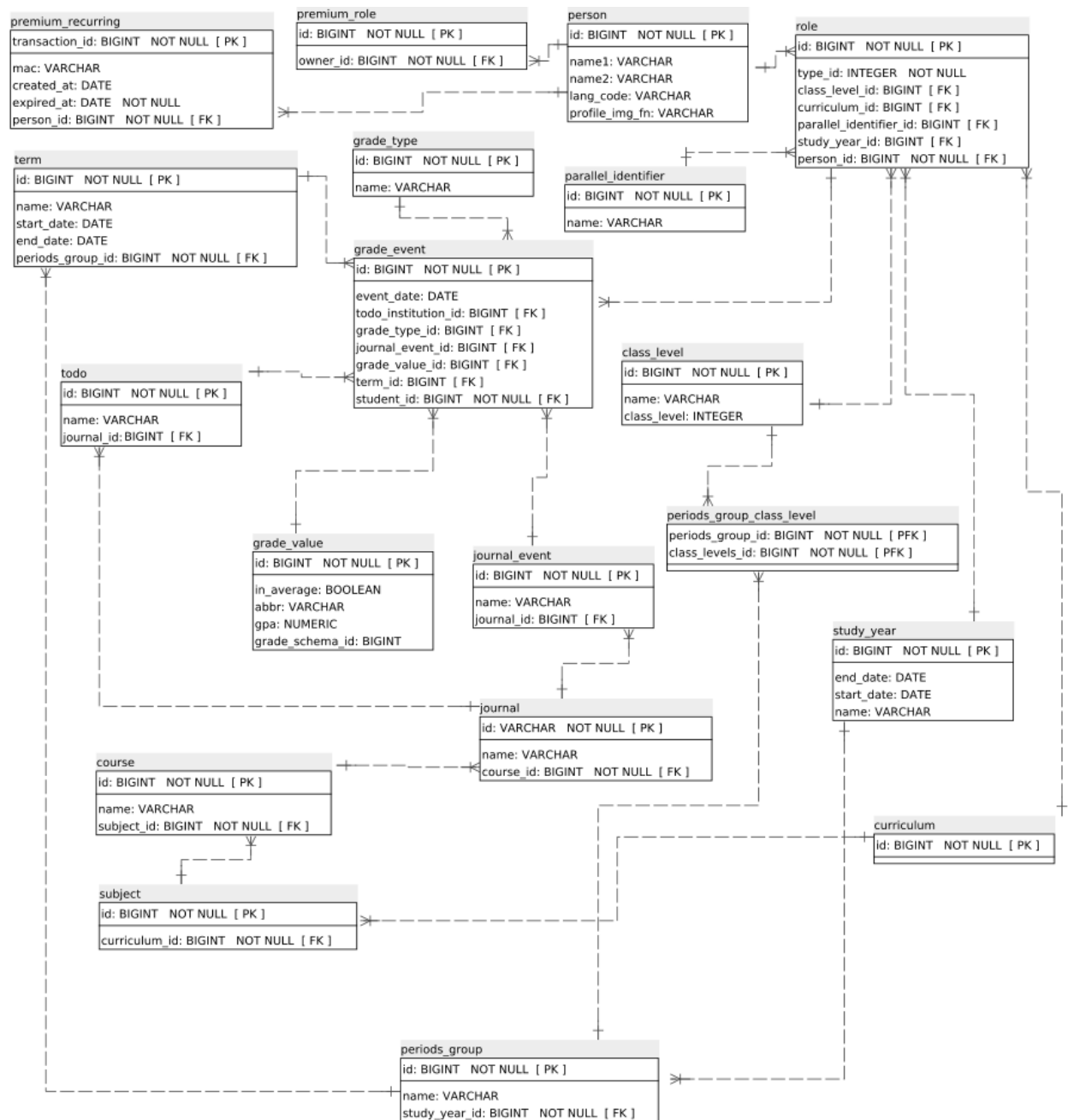
4.2. tabula. Repozitoriju aprakstu tabula

Repozitorija nosaukums	Apraksts
CourseRepository	<ul style="list-style-type: none"> • Atlasa lietotāja mācību priekšmetus.
GradeEventRepository	<p>Visapjomīgākais repozitorijs. Tiek veikta sekmju atlase un apstrāde, lai atgrieztu datus pareizā struktūrā.</p> <p>Atlasa:</p> <ul style="list-style-type: none"> • Dažāda veida sekmes ar dažādiem nosacījumiem. • Sekmes datus. • Nepieciešamās sekmes un datus diagramma izveidošanai. <p>Funkcijas:</p> <ul style="list-style-type: none"> • Nepieciešamo datu atlase. • Pārbauda vai pieprasītie dati ir autorizētajam lietotājam saistīti. • Sadala atlasītās atzīmes parastajās atzīmēs un gala atzīmēs. • Izveido diagrammas iestatījumus un saturu, kas ir viens no svarīgākajiem un sarežģītākajiem uzdevumiem. • Izveido diagrammu klases reitingam un klases dinamikai. • Sagrupē rezultātus pa vietām.

PersonRepository	<ul style="list-style-type: none"> • Atlasa lietotāja datus. • Pārbauda vai pieprasījumā padotā pirmā persona ir pieprasījumā padotās otrās personas bērns.
PremiumRoleRepository	<ul style="list-style-type: none"> • Atlasa lietotāja maksājumus (lietotāja, vecāku, bērnu, ģimenes locekļu maksājumus). • Veic pārbaudi vai kāds no maksājumiem ir atrasts.
RoleRepository	<p>Atlasa:</p> <ul style="list-style-type: none"> • Lietotāja lomas. • Lietotāja bērnus. • Lietotāja bērnu klases. <p>Pārbauda vai pieprasījumā padotā klase un lietotājs ir saistīti.</p>
StudyYearRepository	<ul style="list-style-type: none"> • Atlasa lietotāja mācību gadu.
TermRepository	<ul style="list-style-type: none"> • Atlasa lietotāja periodus.

4.2. Atkarību apraksts

Shēmā ir attēlotas izmantoto tabulu attiecības un nepieciešamie lauki.



4.2. att. Datu bāzes relāciju modelis

4.3. Datu bāzes tabulu apraksts

Datu bāzes relāciju shēmā ir attēlotas tikai funkcionalitātes veikšanai nepieciešamās tabulas. Shēmā attēlotas tabulu attiecības un programmai nepieciešamie ieraksti. Tiek attēloti

tikai nepieciešamie tabulu lauki un tabulas. Kopā izstrādes laikā datu bāzē bija 676 tabulas, bet lielākā daļa no tabulām nav nepieciešamas un tikai neliela daļa ir saistīta ar lietotājiem, skolēniem un atzīmēm. Kā arī tabulas saturēja laukus, kuri nebija nepieciešami platformas izstrādei.

Ar “X” tabulu laukos “Obligāts” ir atzīmēti lauki, kuri nedrīkst būt tukši.

4.3.1. Tabula “person”

Lietotāju tabula, kura tiek izmantota lietotāju datu atlasīšanai un autorizācijai.

4.3. tabula. **Tabula - person**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
name1	VARCHAR(255)		Lietotāja pirmais vārds.
name2	VARCHAR(255)		Lietotāja otrais vārds.
lang_code	VARCHAR(255)		Lietotāja iestatītā valoda.
profile_img_fn	VARCHAR(255)		Maršruts lietotāja profila bildei.

4.3.2. Tabula “role”

Lietotāju skolas lomu tabula. Vienam lietotājam var būt vairākas skolas lomas. Vienai skolas lomai ir piesaistīts viens mācību gads, skola, klases līmenis un paralēlklase, kā arī loma – skolēns, vecāks, skolotājs, direktors un citas mazsvarīgākas lomas. Katru mācību gadu lietotājam tiek piešķirta jauna skolas loma, kur skola var palikt nemainīga. Lietotājs var mācīties vienā klasē vairākus gadus, bet skolas lomas arī tiek piešķirtas jaunas, kā arī skolas loma tiek piešķirta, ja tiek mainīta skola.

4.4. tabula. **Tabula - role**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
type_id	INTEGER	X	Lietotāja lomas tips.
class_level_id	BIGINT		Ārējā atslēga referencējas uz tabulu “class_level”.
curriculum_id	BIGINT		Ārējā atslēga referencējas uz tabulu “curriculum”.

parallel_identifier_id	BIGINT		Ārējā atslēga referencējas uz tabulu “parallel_identifier”.
study_year_id	BIGINT		Ārējā atslēga referencējas uz tabulu “study_year”.
person_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “person”.

4.3.3. Tabula “premium_role”

Tabula uzskaita abonementa maksājumus un to lietotāju, kurš ir veicis maksājumu, un lietotāju, kuram pienākas maksājums (kāds ir samaksājis lietotāja vietā, samaksājis ir lietotāja vecāks, lietotāja bērns, bērna vecāks vai radnieks).

4.5. tabula. Tabula – premium_role

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
owner_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “person”.

4.3.4. Tabula “premium_recurring”

Tabulā uzskaitīta maksājumu informācija – maksājuma termiņi, lietotāji, kuri veica maksājumu.

4.6. tabula. Tabula – premium_recurring

Lauka nosaukums	Datu tips	Obligāts	Apraksts
transaction_id	BIGINT	X	Primārā atslēga.
mac	TEXT		Unikālais maksājuma kods.
created_at	TIMESTAMP		Pakalpojuma apmaksas sākuma datums.
expired_at	TIMESTAMP	X	Pakalpojuma apmaksas izbeigšanās datums.
person_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “person”.

4.3.5. Tabula “grade_type”

Tabulā tiek uzskaitīt visi izliekamo atzīmju tipi.

4.7. tabula. Tabula – grade_type

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
name	VARCHAR(255)		Atzīmes tipa nosaukums.

4.3.6. Tabula “term”

Tabulā ir apkopoti mācību periodi, kuri ir arī sasaistīti ar mācību gadu. Periodiem ir nosaukums, perioda sākuma un beigu datums.

4.8. tabula. Tabula – term

Lauka nosaukums	Datu tips	Obligāts	Apraksts
Id	BIGINT	X	Primārā atslēga.
name	VARCHAR(255)		Mācību perioda nosaukums.
start_date	DATE		Mācību perioda sākuma datums.
end_date	DATE		Mācību perioda beigu datums.
periods_group_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “periods_group”.

4.3.7. Tabula “parallel_identifier”

Tabula satur paralēlklāšu paplašinājumu nosaukumus.

4.9. tabula. Tabula – parallel_identifier

Lauka nosaukums	Datu tips	Obligāts	Apraksts
Id	BIGINT	X	Primārā atslēga.
name	VARCHAR(255)		Paralēlklases paplašinājuma nosaukums.

4.3.8. Tabula “grade_event”

Tabula ir viena no visapjomīgākajām, jo satur visas izliktās atzīmes. Katrai atzīmei ir piesaistīts lietotājs, kuram šī atzīme ir izlikta. Atzīmes tiek piešķirtas skolēniem, bet tās izliek pasniedzēji. Tabula apkopo svarīgāko informāciju par atzīmi – izlikšanas datumu, saistīto

uzdevumu, kurā izlikta atzīme, saistīto periodu, ja atzīme ir perioda (viena no gala atzīmēm) atzīme, saistīto atzīmes tipu, saistīto atzīmes vērtību un apzīmējumu.

4.10. tabula. **Tabula – grade_event**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
event_date	DATE		Atzīmes izlikšanas datums.
todo_institution_id	BIGINT		Ārējā atslēga referencējas uz tabulu “todo”.
grade_type_id	BIGINT		Ārējā atslēga referencējas uz tabulu “grade_type”.
journal_event_id	BIGINT		Ārējā atslēga referencējas uz tabulu “journal_event”.
grade_value_id	BIGINT		Ārējā atslēga referencējas uz tabulu “grade_value”.
term_id	BIGINT		Ārējā atslēga referencējas uz tabulu “term”.
student_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “role”.

4.3.9. Tabula “class_level”

Tabula satur klašu līmeņus.

4.11. tabula. **Tabula – class_level**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
name	VARCHAR(255)		Klases nosaukums.
class_level	INTEGER		Klases līmenis.

4.3.10. Tabula “todo”

Tabula satur informāciju par skolēniem uzdotajiem uzdevumiem.

4.12. tabula. **Tabula – todo**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.

name	VARCHAR(255)		Mājas uzdevuma nosaukums.
journal_id	BIGINT		Ārējā atslēga referencējas uz tabulu “journal”.

4.3.11. Tabula “periods_group_class_level”

Tabula sasaista divas tabula attiecībā - daudz pret daudz.

4.13. tabula. **Tabula – periods_group_class_level**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
periods_group_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “periods_group”.
class_levels_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “class_level”.

4.3.12. Tabula “grade_value”

Tabulā ir uzskaitītas visas atzīmju nepieciešamās detaļas – skaitliskās vērtības, atzīmes nosaukums, atzīmju skala, kurā atzīme ir iekļauta un lauks kurš nosaka, vai atzīmi izmanto vidējās atzīmes aprēķinā.

4.14. tabula. **Tabula – grade_value**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
in_average	BOOLEAN		Lauks nosaka, vai šo atzīmi izmanto vidējās atzīmes aprēķinos.
abbr	VARCHAR(255)		Atzīmes vērtības vizuālais attēlojums.
gpa	NUMERIC(6,2)		Atzīmes skaitliskā vērtība.
grade_schema_id	BIGINT		Ārējā atslēga referencējas uz tabulu “grade_schema”. Tabula nav iekļauta shēmā, bet lauks tiek izmantots grupēšanā, lai sadalītu atzīmes pa atzīmju skalām.

4.3.13. Tabula “journal_event”

Tabula satur informāciju par skolēniem notiekošajām mācību stundām (“notikumiem”), kuras var būt saistītas ar izliktajām atzīmēm.

4.15. tabula. **Tabula – journal_event**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
name	VARCHAR(255)		Klases uzdevuma nosaukums.
journal_id	BIGINT		Ārējā atslēga referencējas uz tabulu “journal”.

4.3.14. Tabula “study_year”

Tabulā ir reģistrēti visi mācību gadi ar galveno informāciju par mācību gadu – mācību gada nosaukumu, mācību gada sākuma un beigu datumu.

4.16. tabula. **Tabula – study_year**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
end_date	DATE		Mācību gada beigas.
startdate	DATE		Mācību gada sākums.
name	VARCHAR(255)		Mācību gada nosaukums.

4.3.15. Tabula “journal”

Tabulā apkopotas mācību stundas klasēm, kurām ir piesaistīti kursi (“mācību priekšmeti”).

4.17. tabula. **Tabula – journal**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
name	VARCHAR(255)		Žurnāla priekšmeta nosaukums.
course_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “course”.

4.3.16. Tabula “course”

Tabulā ir apkopoti kursi, kuri platformā tiek pieņemti kā mācību priekšmeti, jo to nosaukums un pārējās detaļas atbilst mācību priekšmetiem. Šie kursi ir apkopoti mācību priekšmetu grupās tabulā “subject”.

4.18. tabula. **Tabula – course**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
name	VARCHAR(255)		Kursa/mācību priekšmeta nosaukums.
subject_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “subject”.

4.3.17. Tabula “curriculum”

Tabulā ir uzskaitīti studiju plāni. Skolēnam ir reģistrēts studiju plāns. Studiju plānam ir piesaistīti mācību priekšmeti.

4.19. tabula. **Tabula – curriculum**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.

4.3.18. Tabula “subject”

Tabulā tiek uzskaitītas mācību priekšmetu grupas.

4.20. tabula. **Tabula – subject**

Lauka nosaukums	Datu tips	Obligāts	Apraksts
id	BIGINT	X	Primārā atslēga.
curriculum_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “curriculum”.

4.3.19. Tabula “periods_group”

Tabulā ir reģistrētas mācību periodu grupas, kuras ir piesaistītas periodiem un studiju gadam.

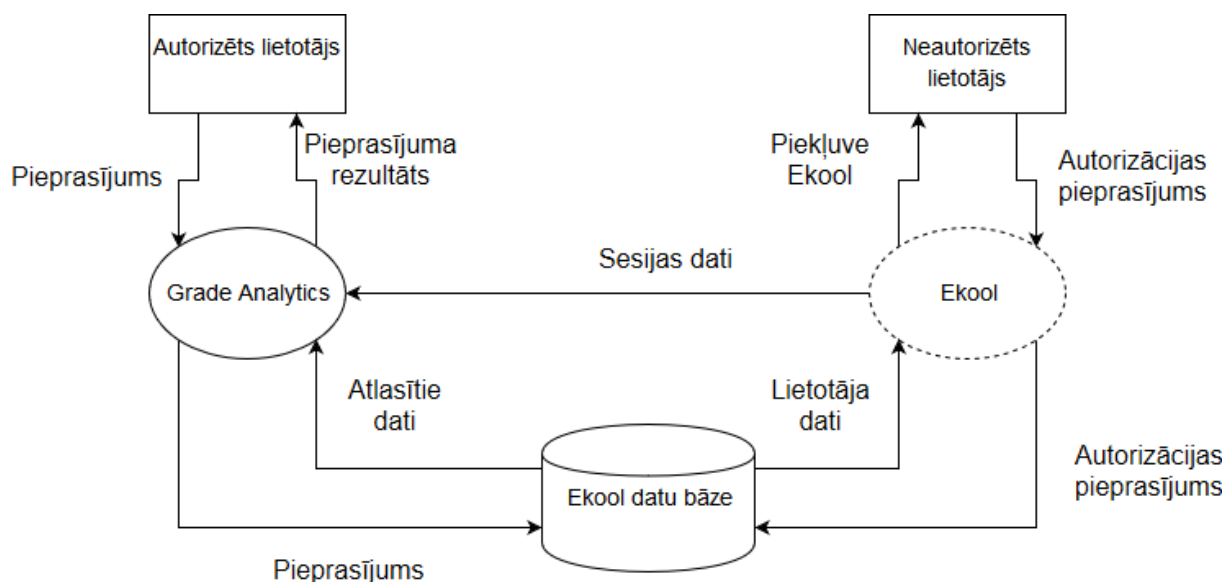
4.21. tabula. Tabula – periods_group

Lauka nosaukums	Datu tips	Obligāts	Apraksts
Id	BIGINT	X	Primārā atslēga.
Name	VARCHAR(255)		Periodu grupas nosaukums.
study_year_id	BIGINT	X	Ārējā atslēga referencējas uz tabulu “study_year”.

4.4. Datu plūsmu diagrammas

4.4.1. 0. līmeņa datu plūsmu diagramma

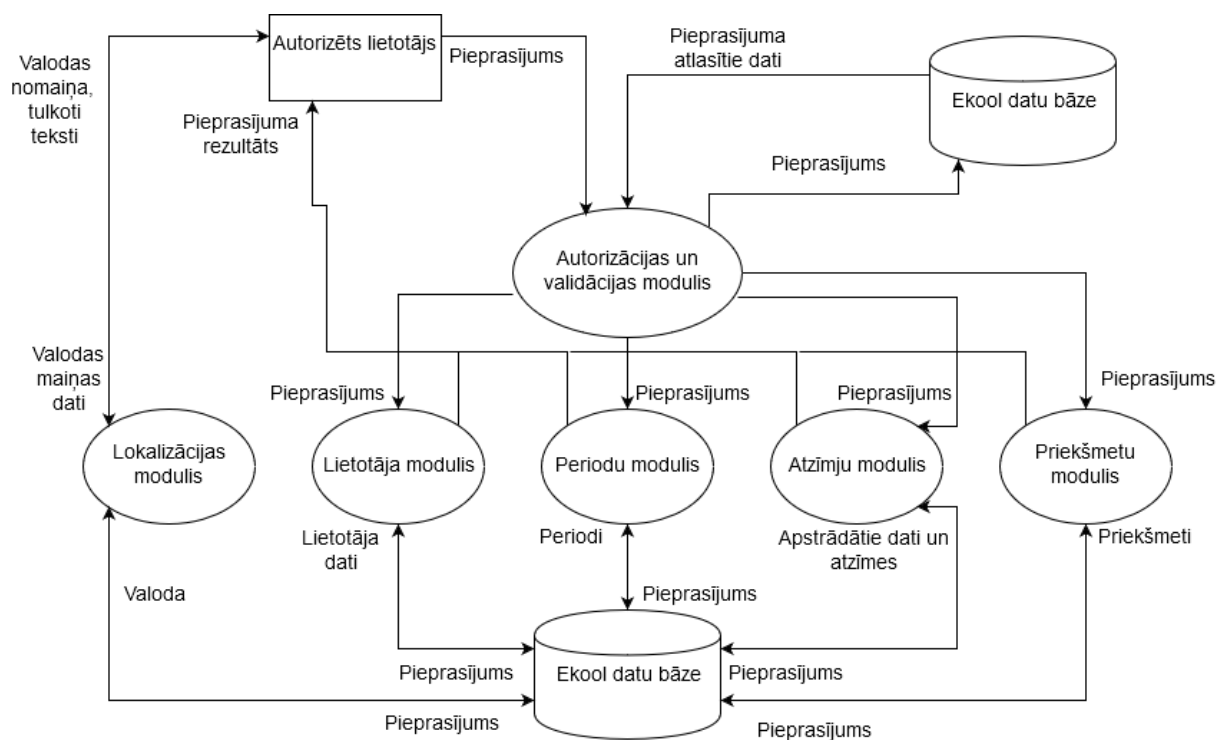
0. līmeņa datu plūsmas diagrammā var redzēt to, ka lietotājam ir jāautorizējas Ekool sistēmā. Pēc autorizēšanās lietotājs var atvērt “Grade Analytics” platformu, kur platforma no Ekool saņem sesijas datus un ar tiem veic lietotāja autorizāciju platformā un dod piekļuvi lietotājam. Platforma izmanto to pašu datu bāzi, kuru izmanto Ekool sistēma.



4.2. att. 0. līmeņa datu plūsmas diagramma

4.4.2. 1. līmeņa datu plūsmu diagramma

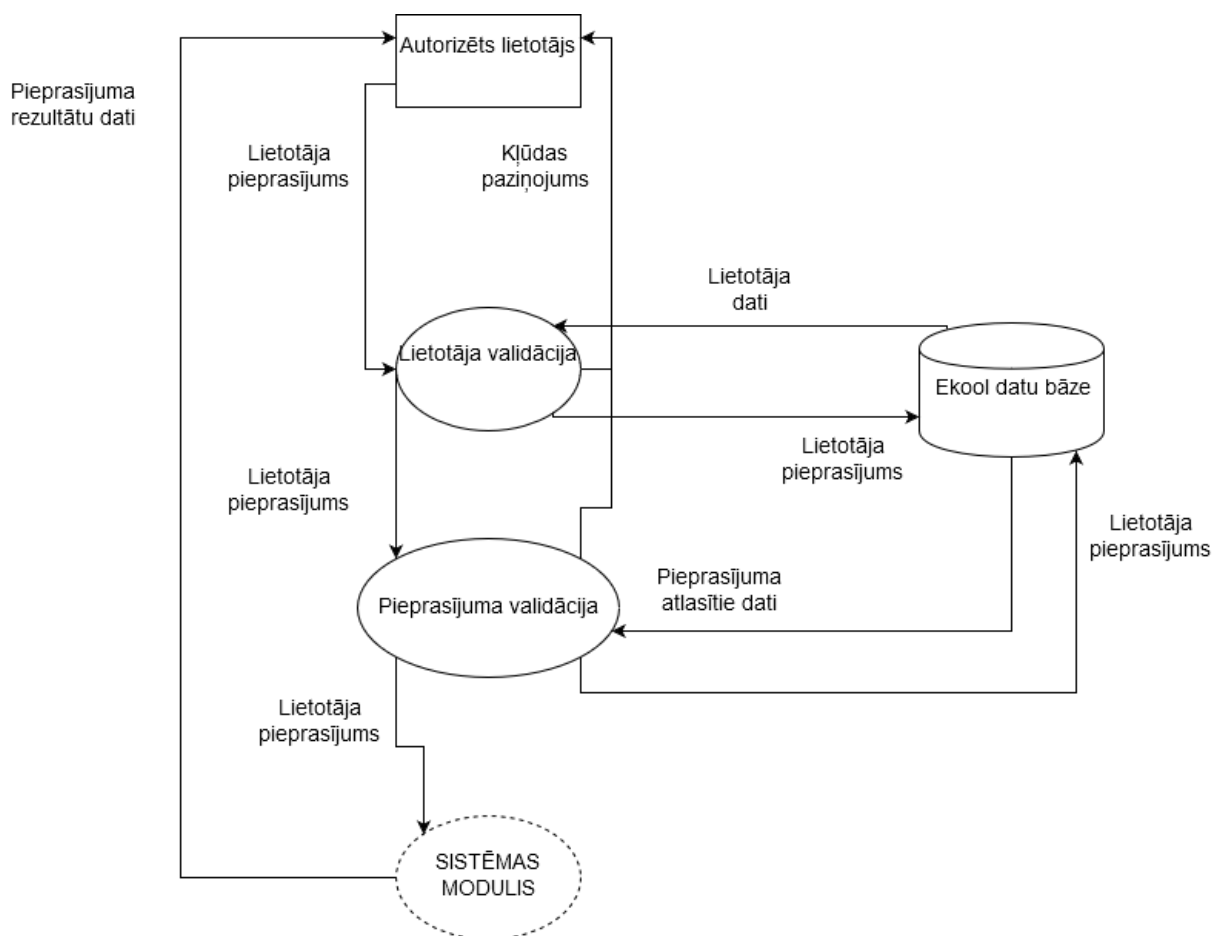
1. datu plūsmas diagrammā attēloti sistēmas galvenie funkcionālie moduļi un datu plūsmas uz un no moduļiem. Diagrammā redzams, ka lietotāja visiem pieprasījumiem vispirms ir jāiziet cauri “Autorizācijas un validācijas moduļim”, kurā lietotājs tiek autorizēts katrā pieprasījumā un tiek veikta datu validācija.



4.3. att. 1. līmeņa datu plūsmas diagramma

4.4.3. Autorizācijas un validācijas modulis - 2. līmeņa datu plūsmas diagramma

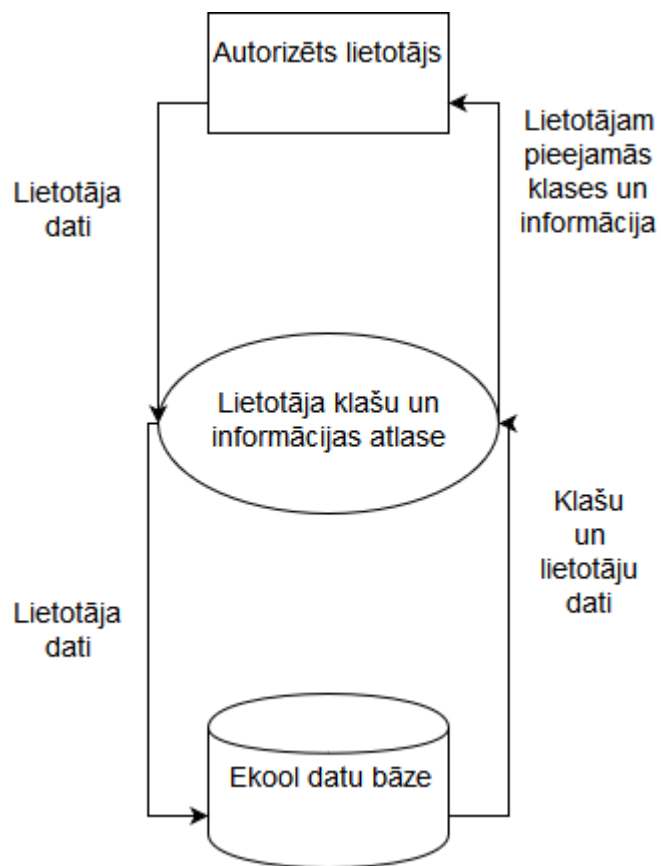
Autorizācijas modulis saņem visus lietotāja pieprasījuma datus. Modulis pārbauda vai lietotājs eksistē, vai lietotājs ir samaksājis abonementu. Tālāk tiek veikta pārējo pieprasījuma datu validācija, kur tiek pārbaudīts vai pieprasītie dati eksistē, kā arī vai pieprasītie dati ir savā starpā saistīti. Šī moduļa funkcionalitāte nodrošina datu drošību – lietotājam tiek atgriezti viņam piederošie dati, tas ir, lietotājs nevar apskatīt svešu lietotāju atzīmes un pārējos datus.



4.4. att. Autorizācijas un validācijas modulis

4.4.4. Lietotāja modulis - 2. līmeņa datu plūsmas diagramma

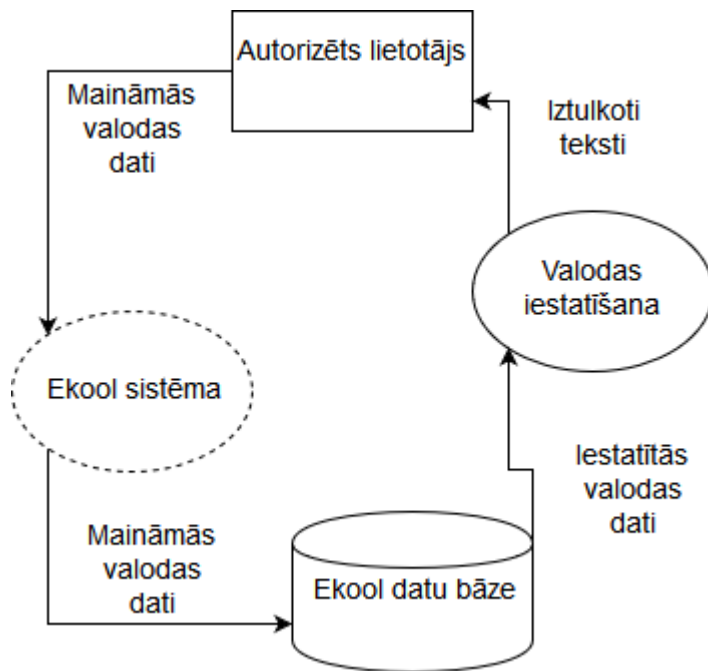
Modulis veic lietotājam piederošo pamatinformācijas datu atlasī. Tiek atlasītas lietotājam pieejamās klases.



4.5. att. Lietotāja modulis

4.4.5. Lokalizācijas modulis - 2. līmeņa datu plūsmas diagramma

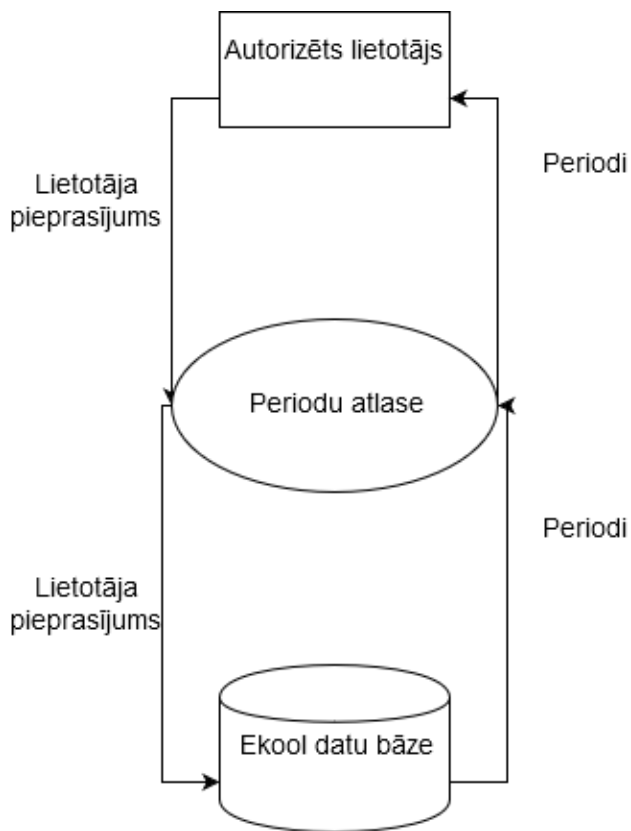
Lokalizācijas modulis nodrošina valodas maiņu. Lietotājs platformā nevar nomainīt valodu, tāpēc tiek attēlota arī Ekool sistēma, kurā lietotājs nomaina valodu, kuru pēc tam ar sesijas datu saņemšanas palīdzību varēs iegūt izvēlēto valodu no Ekool datu bāzes. Šī valoda tiek izmantota arī platformā. Kurā katrai sistēmas valodai ir definēti teksti sava valodā.



4.6. att. Lokalizācijas modulis

4.4.6. Periodu modulis - 2. līmeņa datu plūsmas diagramma

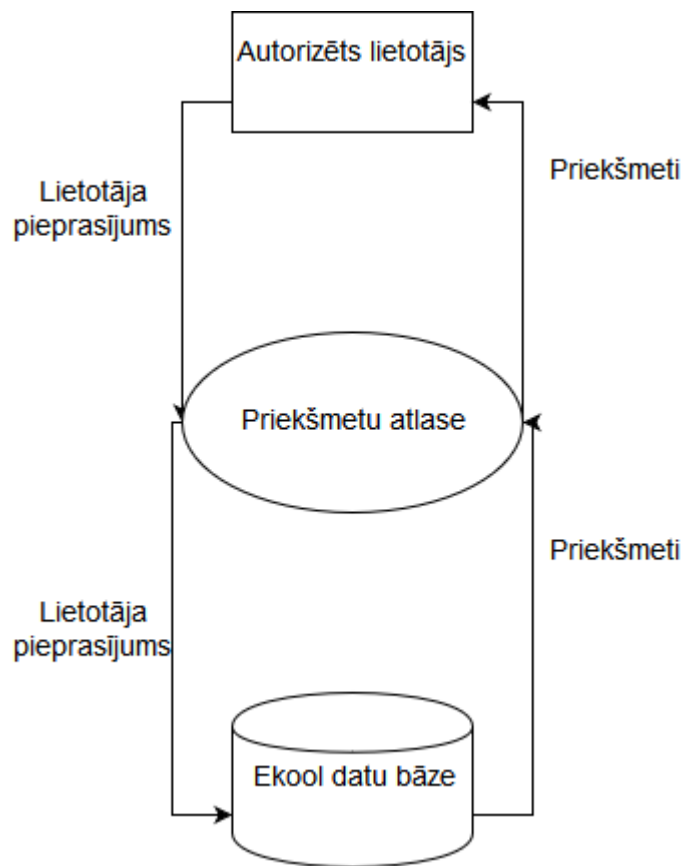
Periodu moduļa funkcionalitāte ir ļoti triviāla, tajā tiek atlasīti periodi un mācību gadi.



4.7. att. Periodu modulis

4.4.7. Priekšmetu modulis - 2. līmeņa datu plūsmas diagramma

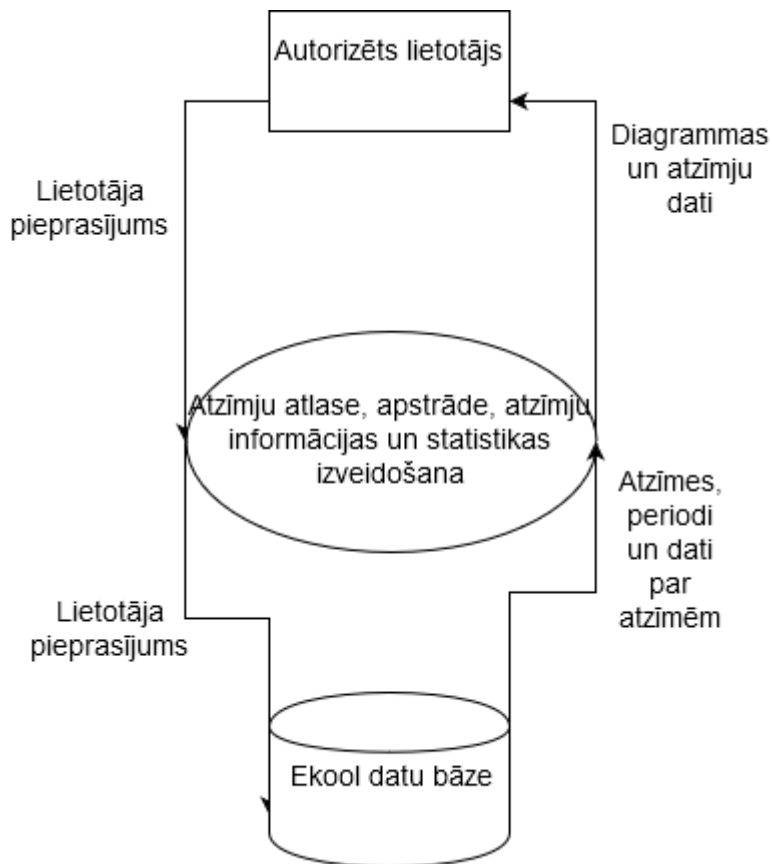
Priekšmetu moduļa funkcionalitāte ir ļoti triviāla, tajā tiek atlasīti mācību priekšmeti.



4.8. att. Priekšmetu modulis

4.4.8. Atzīmju modulis - 2. līmeņa datu plūsmas diagramma

Atzīmju moduļa funkcionalitāte ir visdaudzveidīgākā un sarežģītākā. Tajā tiek atlasītas atzīmes, periodi, mācību gadi un pārējie nepieciešamie dati, kuri tiek izmantoti atzīmju informācijas attēlošanai, statistikas diagrammu izveidošanai un dinamiskas diagrammas izveidošanai.



4.9. att. Atzīmju modulis

5. TESTĒŠANAS DOKUMENTĀCIJA

5.1. Ievads

Testēšana netika veikta izmantojot automātisko vienībtestēšanu. Testēšanas procesā izmantoju manuālo testēšanu ar Baltās kastes (“White-Box”) metodi, kurā es, regulāri veicot izmaiņas, veicu funkciju testēšanu – izvēlējos dažādus ievaddatus un analizēju izvaddatus vai programmas kļūdas ziņojumus. Testēšana netika veikta katras iterācijas beigās, bet regulāri veicot izmaiņas un uzlabojumus līdz vēlamā funkcionalitāte tika sasniegta.

5.2. Akcepttesti

Akcepttesti paredzēti, lai veiktu programmatūras vēlamās funkcionalitātes pārbaudi. Akcepttestos tiek pārbaudīts, vai lietotārstāsts ir pareizi realizēts. Akcepttestos parādīti tikai izstrādes beigās veiktie testi – 5. iterācijas beigās veiktie testi. Neliela manuālā testēšana tika veikta regulāri, tādēļ tiks veikta tikai pēdējo testu uzskaitē, kurā testēti visi lietotārstāsti. Lietotārstāstu testos tika izmantoti dažādi lietotāji ar dažādām atzīmēm, lietotāji ar un bez bērniem, lietotāji bez klasēm, bez atzīmēm. Tika atrasti visdažādākie lietotāji un veikta manuālā testēšana, pārbaudot pieejamās funkcionalitātes.

5.1. tabula. 1. lietotārstāsta akcepttesta rezultāti

Lietotārstāsts: Kā lietotājs vēlos autorizēties platformā.	
Izpilde: Atveru Ekool mājaslapu, atrodu sadaļu “Grade Analytics” un autorizējos platformā.	Rezultātu analīze: Uz šo brīdi lietotājs nevar autorizēties platformā, jo platforma vēl nav integrēta Ekool sistēmā. Integrēšanu un tās testēšanu veic Ekool komanda, kura to vēl nav izdarījusi. Patreiz platformā autorizētais lietotājs ir cieti iekodēts, lietotāju var mainīt tikai veicot izmaiņas kodā.

5.2. tabula. 2. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot savu profila pamatinformāciju.	
Izpilde: Atveru platformu un redzu savu vai sava bērna vārdu un klasi. Varu izvēlēties savas un savu bērnu klases.	Rezultātu analīze: Darbības izpildās. Nepieciešamā informācija attēlojas. Atverot lietotāju, kuram nav klašu un arī lietotāja bērnam nav klašu, tiek atgriezts kļūdu paziņojums.

5.3. tabula. 3. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot man pieejamās klases un izvēlēties kādu no klasēm tālākai informācijas atlasei.	
Izpilde: Platformā izvēlos klases. Izvēloties klasi, tā kļūst par aktīvo klasi informācijas atlasei.	Rezultātu analīze: Darbības izpildās. Izvēloties dažādas klases, attēlojas tai klasei saistošie dati.

5.4. tabula. 4. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot un izvēlēties kādu no periodiem tālākai informācijas atlasei.	
Izpilde: Atlasītajai klasei izvēlos kādu no saistošajiem periodiem, kurš kļūst par aktīvo periodu informācijas atlasei.	Rezultātu analīze: Darbības izpildās. Izvēloties dažādus periodus, attēlojas tajā periodā ievietotās atzīmes.

5.5. tabula. 5. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot savu bērnu profilus, lai varētu aplūkot viņiem pieejamo informāciju.	
Izpilde: Izmantojot profilu, kura bērniem ir reģistrētas klases, redzu savu bērnu klases un izvēlos klasi informācijas atlasei – redzu saviem bērniem izliktās atzīmes, klases periodus un priekšmetus.	Rezultātu analīze: Darbības izpildās. Es varu izvēlēties savu bērnu klases, attēlojas bērniem redzamā informācija.

5.6. tabula. 6. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot savas jaunākās sekmes.	
<p>Izpilde: Atverot platformu, redzu aktīvās klases jaunākās atzīmes. Mainot klasi, redzu katras klases jaunākās atzīmes.</p>	<p>Rezultātu analīze: Darbības izpildās. Katrai klasei attēlojas tajā klasē reģistrētas jaunākās atzīmes. Rādās līdz 10 jaunākajām atzīmēm. Lietotājam, kuram nav atzīmju tiek atgriezts kļūdu paziņojums.</p>

5.7. tabula. 7. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot sekmes izvēlētā priekšmetā.	
<p>Izpilde: Atveru atzīmju sadaļu “Grades”, kurā izvēlos priekšmetu, kurā ir izliktas atzīmes. Izvēlos atzīmi izliktajā priekšmetā.</p>	<p>Rezultātu analīze: Darbības izpildās. Atverot atzīmju sadaļu attēlojas lietotājam reģistrētie mācību priekšmeti. Ja lietotājam nav reģistrēti mācību priekšmeti, tad tiek atgriezts kļūdu paziņojums. Izvēloties priekšmetu, attēlojas priekšmetā izliktais atzīmes. Ja priekšmetā nav izliktas atzīmes, tad tiek atgriezts kļūdu paziņojums.</p>

5.8. tabula. 8. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot gala atzīmes, izvēlētajā klasē.	
<p>Izpilde: Periodu sarakstā izvēlos mācību gadu, atveru atzīmju sadaļu “Grades”, izvēlos priekšmetu.</p>	<p>Rezultātu analīze: Darbības izpildās. Izvēloties mācību gadu periodu sarakstā, atzīmju sadaļā parādās periodos izliktais atzīmes. Ja kādā periodā netiek izlikta atzīme, tad šis periods netiek attēlots, rāda tikai izliktais atzīmes. Ja lietotājam nav izliktas gala atzīmes izvēlētajā priekšmetā, tad tiek atgriezts kļūdu paziņojums.</p>

5.9. tabula. 9. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot savas sekmes detaļas.	
<p>Izpilde: Izvēlos izliktās sekmes, gan jaunāko sekmju sarakstā, gan atzīmju sadaļā. Izvēlos gan parastās atzīmes, gan gala atzīmes.</p>	<p>Rezultātu analīze: Darbības izpildās. Izvēloties atzīmi, tiek attēlotas atzīmes detaļas. Atzīmes izlikšanas datums, atzīmes tips un parastajām atzīmēm arī uzdevuma detaļas. Ja kāda no atzīmes detaļām nav pieejama, tad tajā vietā tiek attēlots tukšums.</p>

5.10. tabula. 10. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot visas sekmes, kuras izliktas uzdevumā.	
<p>Izpilde: Izvēlos kādu no atzīmēm un aplūkoju pārējiem skolēniem izliktos rezultātus.</p>	<p>Rezultātu analīze: Darbības izpildās. Izvēloties atzīmi, zem atzīmes detaļām parādās diagramma ar pārējo skolēnu rezultātiem dotajā uzdevumā. Tiek attēlota stabiņu diagramma vai arī apļa diagramma. Tiek attēlots skolēnu rezultāts un skolēnu skaits, kas saņēma rezultātu. Diagramma neattēlojas, ja pārējie skolēni nav saņēmuši atzīmi dotajā uzdevumā.</p>

5.11. tabula. 11. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos aplūkot savas klases reitingu izvēlētajā priekšmetā un periodā.	
<p>Izpilde: Izvēlos periodu, izvēlos reitingu sadaļu "Rating" un izvēlos mācību priekšmetu.</p>	<p>Rezultātu analīze: Darbības izpildās. Izvēloties klasi, tiek attēlota diagramma ar klases reitingu, kurā norādītas vidējās skolēnu atzīmes izvēlētajā periodā. Ja nevienam skolēnam nav izliktas atzīmes, tad diagramma netiek rādīta. Ja atzīmes ir izliktas pārējai klasei, bet ne lietotājam, tad diagramma arī tiek attēlota, bet lietotājs netiek izcelts.</p>

5.12. tabula. 12. lietotājistāsta akcepttesta rezultāti

Lietotājistāsts: Kā lietotājs vēlos aplūkot savas klases reitingu izvēlētajā periodā.	
<p>Izpilde: Izvēlos periodu, izvēlos reitingu sadaļu “Rating” un kopējā reitinga sadaļu “Class Ratings”.</p>	<p>Rezultātu analīze: Darbības izpildās. Izvēloties kopējā reitinga sadaļu, tiek attēlota diagramma ar vidējo atzīmi no visos priekšmetos iegūtajām vidējām atzīmēm. Ja neviens nav saņēmis atzīmes, tad diagramma netiek rādīta. Ja atzīmes ir izliktas pārējai klasei, bet ne lietotājam, tad diagramma arī tiek attēlota, bet lietotājs netiek izcelts.</p>

5.13. tabula. 13. lietotājistāsta akcepttesta rezultāti

Lietotājistāsts: Kā lietotājs vēlos aplūkot savas klases sekmju dinamiku izvēlētajā priekšmetā un periodā.	
<p>Izpilde: Izvēlos periodu, izvēlos reitingu sadaļu “Rating”, izvēlos priekšmetu un dinamikas sadaļu “Dynamic”.</p>	<p>Rezultātu analīze: Darbības izpildās. Izvēloties dinamikas sadaļu, tiek attēlota līnijdiagramma ar atzīmju izmaiņām pa mēnešiem. Ja neviens nav saņēmis atzīmes, tad diagramma netiek rādīta. Ja atzīmes ir izliktas pārējai klasei, bet ne lietotājam, tad diagramma arī tiek attēlota, bet lietotāja vērtība parādās kā nulle.</p>

5.14. tabula. 14. lietotājistāsta akcepttesta rezultāti

Lietotājistāsts: Kā lietotājs vēlos aplūkot savas klases sekmju dinamiku izvēlētajā priekšmetā un mācību gadā.	
<p>Izpilde: Izvēlos mācību gadu, izvēlos reitingu sadaļu “Rating”, izvēlos priekšmetu un dinamikas sadaļu “Dynamic”.</p>	<p>Rezultātu analīze: Darbības izpildās. Izvēloties dinamikas sadaļu, tiek attēlota līnijdiagramma ar atzīmju izmaiņām pa periodiem. Ja neviens nav saņēmis atzīmes, tad diagramma netiek rādīta. Ja atzīmes ir izliktas pārējai klasei, bet ne lietotājam, tad diagramma arī tiek attēlota, bet lietotāja vērtība parādās kā nulle.</p>

5.15. tabula. 15. lietotājstāsta akcepttesta rezultāti

Lietotājstāsts: Kā lietotājs vēlos, lai informācija tiktu pasniegt manam profilam uzstādītajā valodā.

<p>Izpilde: Nomainu Ekool sistēmā valodu. Autorizējos “Grade Analytics” un redzu valodas izmaiņas.</p>	<p>Rezultātu analīze: Uz doto brīdi lietotājs nevar autorizēties platformā, izmantojot Ekool sistēmu, jo platforma vēl nav integrēta Ekool sistēmā. Valodas maiņa notiek, bet nevar notestēt vai tikko nomainīta valoda Ekool sistēmā liks nomainīt valodu arī integrējamajā platformā.</p>
---	--

6. PROJEKTA PARVALDĪBA

6.1. Projekta organizācija

Projekta izstrādē tika izmantotas spējās izstrādes vadlīnijas, jo projekta sākumā nebija stingri definētas funkcionālās prasības. Funkcionālās prasības laika gaitā tika apkopotas lietotājistāstu veidā. Ūdenskrituma metode būtu bijusi neefektīva, jo izstrādes procesā mainījās funkciju ievaddati, izvaddati, mainījās funkcionalitāte. Izmantojot ūdenskrituma modeli, būtu jāiegulda ļoti daudz darba laika dokumentācijas labošanā, lai saglabātu trasējamību.

Projekta pirmās dienas tika pavadītas sagatavojot darba vidi, uzstādot nepieciešamo programmatūru, uzstādot lokālo serveri un iegūstot pieeju Ekool datu bāzei.

Tālāk pēc nelielas datu bāzes apraksta saņemšanas notika aktīva datu bāzes analīze un pētīšana, kura ilga apmēram nedēļu. Tika veikta kontaktēšanās ar Ekool pārstāvjiem, lai precizētu neskaidrības datu bāzes jautājumos. Izmantojot minimālo aprakstu, tika atrastas produkta funkcionalitātes realizēšanai nepieciešamās tabulas, tika izveidots relāciju modelis.

Produkta izstrādes process tika sadalīts vairākās iterācijās, kuras raksturo produkta sadaļas un realizētā funkcionalitāte.

Izstrāde tika sākta ar datu bāzes analīzi, datu atlasu rakstīšanu un analīzi, lai izveidotu pareizas datu atlasas, kuri atgriež nepieciešamos datus. Izmantojot iepriekš uzrakstītās datu atlasas, tie tika uzrakstīti Laravel ietvara kodā. Tālāk šie dati tika apstrādāti un atgriezti uz lietotāju saskarni. Tika izstrādāta lietotāja piekļuves validācija, kurā tika pārbaudīts vai lietotājs ir autorizēts atbilstošajam pieprasījumam.

Izstrādes virzība uz priekšu tika panākta regulāri kontaktējoties ar projekta vadītāju. Pāris reizes notika arī kontaktēšanās ar Ekool sistēmas uzturēšanas komandu. Dažas reizes notika SCRUM sapulces – tika apkopots padarītais, ieplānotie uzdevumi un radušās sarežģītības. Produkta izstrādes laikā uzņēmuma programmētājs veica produkta programmas PHP koda apskati, tika uzklauti ieteikumi un veikti labojumi. SQL koda apskati veica Ekool sistēmas datu bāzes pārzinis Ekool uzņēmumā, pēc apskates arī tika veikti nepieciešamie labojumi.

6.2. Kvalitātes nodrošināšana

Izstrādājamā platforma tika izstrādāta Laravel ietvarā, vadoties pēc MVC (*Model – View – Controller*) izstrādes principiem. MVC priekšrocība ir ērta programmatūras koda uzturēšana un organizācija. Tika izmantota PHP valoda, Laravel ietvars, PostgreSQL. Lietotāju saskarnē

izstrādātājs izmantoja HTML, CSS, JavaScript jQuery, jQuery SnackBar, Bootstrap, Moment JS, Knockout JS, Chartist JS.

Kods tika komentēts angļu valodā, tika ievēroti koda noformēšanas nosacījumi – koda noformējuma stils tika ievērots visā kodā. Funkciju nosaukumi tika veidoti aprakstoši, lai vieglāk varētu uztvert to darbību un nesajauktu ar citām funkcijām. Funkciju nosaukumi tika veidoti saskaņā ar “*camelCase*” nosacījumiem, bet lielākā daļa mainīgo – “*snake_case*”. SQL injekcijas briesmu kodā nav, jo datu bāzes dati netiek atlasīti pēc lietotāja ievadlauku datiem. Maršrutam saistītās funkcijas tika rakstītas kontrolieros, bet iekšējās funkcijas un datu bāzes datu atlase tika veikta repozitorijos. Pēc strādājošas funkcijas izveides vai labojumiem izmaiņas tika nodotas konfigurācijas pārvaldības sistēmai.

6.3. Konfigurāciju pārvaldība

Programmkoda versiju kontrolei tika izmantota gitLab konfigurāciju pārvaldes sistēma. Kods tika regulāri augšupielādēts gitLab, augšupielādēšana tika komentēta, lai izsekotu koda izmaiņām laika gaitā. gitLab vide nodrošināja ērtu programmatūras koda salīdzināšanu ar koda vecākām versijām, tāpēc kodā iepriekš veiktās izmaiņas bija viegli pārvaldāmas.

gitLab pakalpojums nodrošināja:

- Informāciju par koda izmaiņām.
- Piekļuvi iepriekšējām koda versijām.
- Atgriezties uz iepriekšējām koda versijām.
- Piekļuvi kodam no vairākiem datoriem.

7. DARBIETIPLĪBAS NOVĒRTĒJUMS

Lai noskaidrotu nepieciešamos ieguldāmos laika resursus, darbietilpības novērtējums netika veikts izstrādes sākumā, bet izstrādes laikā. Viens no populārākajiem spējās izstrādes darbietilpības novērtēšanas veidiem ir izmantot funkcijpunktu vai sarežģītības punktu piešķiršanu lietotājtāstiem. Programmatūrā lietotās tehnoloģijas man bija nedaudz pazīstamas, tas traucē iegūt precīzu darbietilpības novērtējumu. Tā kā programmatūras izstrādē neesmu pieredzējis, tad paļauties uz standartizētām aprēķinu metodēm nebija vēlams. Plānošana tika veikta uz manu iegūto nelielo pieredzi un pieredzējušu programmētāju novērtējuma. Iterācijām nepieciešamie sarežģītības punkti tika piešķirti pēc koda uzrakstīšanas, lai raksturotu ieguldītā laika un darba sarežģītības sadalījumu. Darbā ieguldītais laiks tika plānots intuitīvi, nepieciešamā funkcionalitāte tika papildināta un labota pēc nepieciešamības. Projekts pēc izstrādes tiks papildināts un uzlabots.

Vērtēšanā izmantotie sarežģītības punkti nav funkcijpunkti, bet beigās aprēķinot visus ieguldītos punktus un ieguldīto laiku ir iespējams aprēķināt katra sarežģītības punkta vērtību ieguldītās koda rindiņās un darba stundās. Sarežģītības punktos tika izmantota Fibonači punktu skala. Ar punktiem 2, 3, 5 tika novērtēti viegli un ātri realizējami lietotājtāsti. Ar punktiem 8, 13 grūtāki un laikietilpīgāki lietotājtāsti, bet ar 21 punktu tika novērtēts vislaikietilpīgākais process – sagatavošanās un analīze bija vislaikietilpīgākā, bet no sarežģītības viedokļa šis process nebija vissarežģītākais un neprasa lielu programmētāja pieredzi.

Katras iterācijas darbietilpības novērtējums:

7.1. tabula. Iterāciju darbietilpības novērtējums

Iterācija	Kopējais sarežģītības punktu skaits
0. iterācija	26
1. iterācija	20
2. iterācija	13
3. iterācija	16
4. iterācija	47
5. iterācija	5

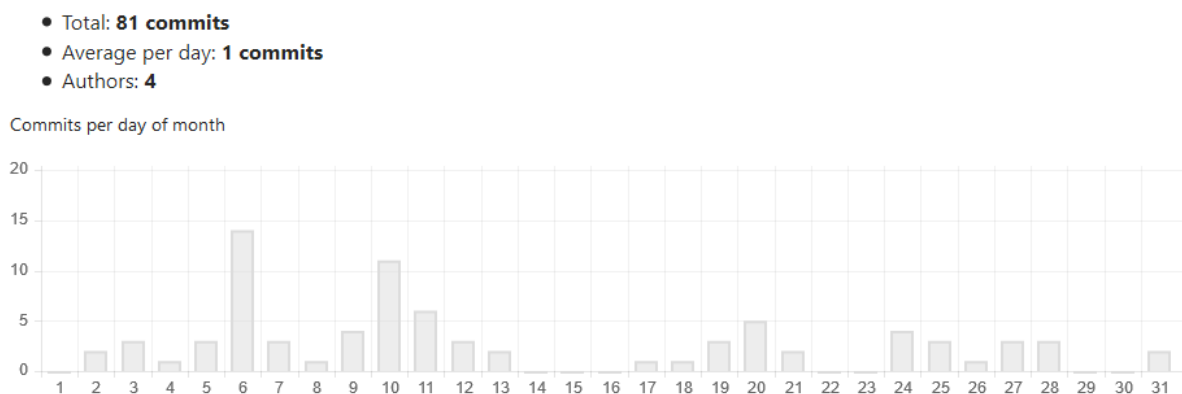
Kopā tika ieguldīti 127 sarežģītības punkti. Šie punkti tika ieguldīti apmēram 60 darbadienās. No šīm 60 dienām apmēram 40 tika ieguldītas projekta izstrādē, bet 20 –

sagatavošanās darbam un situācijas analīzei. Nedēļā vidēji tika ieguldīti 10 līdz 11 sarežģītības punkti.

Projekta izstrādē apmēram tika uzrakstītas 3100 koda rindiņas, kurās arī tika uzskatītas komentāru rindiņas un daudzi tukšumi un pārneseņi jaunās rindiņās, kuri ir paredzēti koda lasāmībai un struktūrai. Netika ieskaitītas izveidoto modeļu rindiņas, jo modeļos definētās metodes un modeļu savstarpējās atkarības praktiski netika izmantotas, kā arī netika iekļautas tulkojumu rindiņas un mainītais lietotāja saskarnes kods, kurš tika mainīts tulkojumu vajadzībām.

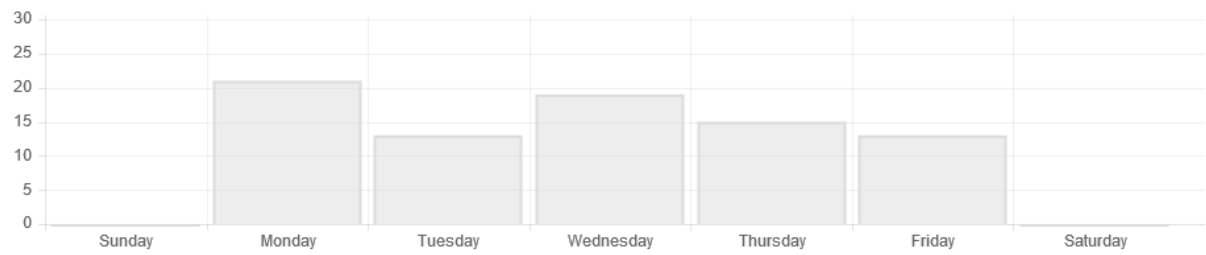
Izstrādes ātrums nebija konstants, tas attēlojas arī konfigurācijas pārvaldības sistēmas augšupielādēs (sk. 7.1., 7.2., 7.3. att.). Ieguldītais darbs un izstrādes laiks lietotājstāstos bija dažāds. Visvairāk augšupielādes tika veiktas mēnešu 6. datumos, pirmdienās, trešdienās un pulksten 16:00. Vairāk augšupielādes tika veiktas dienas otrajā dienas pusē, augšupielādējot dienas laikā paveikto darbu.

Izmantojot COCOMO Basic darbietilpības kalkulatoru, tika iegūti apmēram 7 personmēneši, kas ir daudz vairāk nekā reāli ieguldītais laiks. COCOMO II kalkulators deva līdzīgus rezultātus, tāpēc arī šī metode netika izmantota darbietilpības prognozēšanā pirms darba uzsākšanas, jo pastāv ļoti daudz dažādu aspektu, kuri ietekmē darba ražīgumu – pieredze, psiholoģiskais stāvoklis, pieļautās kļūdas, jaunu izmaiņu ieviešana un daudzi citi aspekti. Darbietilpības plānošana notika intuitīvi un paļaujoties uz pieredzējušu programmētāju norādījumiem.



7.1. att. Augšupielādes mēnešu datumos

Commits per weekday



7.2. att. Augšupielādes nedēļas dienās

Commits per day hour (UTC)

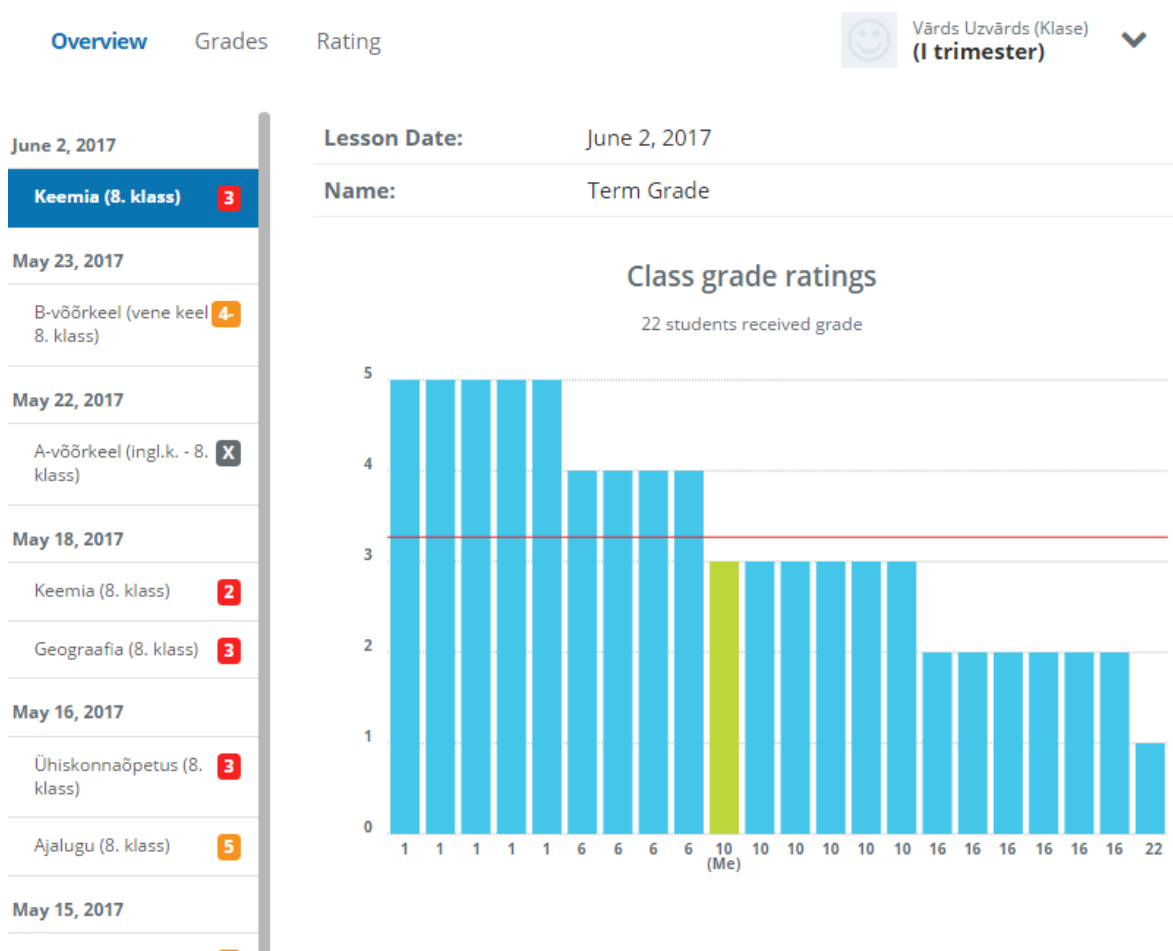


7.3. att. Augšupielādes darbadienas laikā

8. LIETOTĀJA EKRĀNA FORMU APRAKSTS

Šajā nodaļā ir aprakstītas dažas no “Grade Analytics” platformas ekrāna formām jeb lietotāja saskarnēm. Lietotāja saskarnes izstrādāja cits izstrādātājs. Katras lapas augšpusē atrodas lietotāja izvēlnes logs kurš sastāv no galvenajām sadaļām, kurās tiek attēlotas jaunākās atzīmes, visas izliktās atzīmes, reitings un lietotāja sadaļa, kurā attēlota aktīvā klase un periods.


Attēlā redzams sākuma skats (sk 8.1.att.). Šī lietotāja saskarne ir redzama pēc platformas atvēršanas. Augšējā labajā stūrī ir redzama lietotāja aktīvā izvēlētā klase ar personas vārdu un uzvārdu. Augšējā kreisajā stūrī ir redzama galvenā izvēlne, kurā lietotājs var izvēlēties sadaļas – Overview (Sākuma pārskats ar jaunākajām atzīmēm), Grades (Visas lietotājam izliktās atzīmes) un Rating (Atzīmju reitings). Labajā malā tiek uzskaitītas jaunākās atzīmes ar izlikšanas datumu. Ekrāna vidēja daļā parādās izvēlētās jaunākās atzīmes detaļas un diagramma. Lietotājs var mainīt klasi un pārējo informāciju uzklikšķinot uz bultiņas augšējā labajā ekrāna stūrī.





8.1. att. Sākuma skats


Atverot klašu izvēlnes logu var redzēt klašu un periodu sarakstu, kurā var izvēlēties gan savas klases, gan savu bērnu klases, kā arī klasei pieejamos periodus un mācību gadu, lai atlasītu vēlamos datus. Sākumā tiek attēlotas līdz sešām klasēm. Uzklikšķinot uz lauka “View All”, tiek attēlotas visas lietotājam pieejamās klases. Lai atkal rādītu mazāk klases, tad jāuzklikšķina uz lauka “View Less”.







Overview
Grades
Rating


Vārds Uzvārds (Klase)
(I trimester)




Classes


Periods

 Ilo Rätsep (2B)	I trimester
 Emma-Heidi Rätsep (8A)	II trimester
 Ilo Rätsep (3B)	III trimester
 Ilo Rätsep (1B)	Täiendav õppetöö
 Emma-Heidi Rätsep (7A)	2016/2017
 Emma-Heidi Rätsep (6A)	

View All

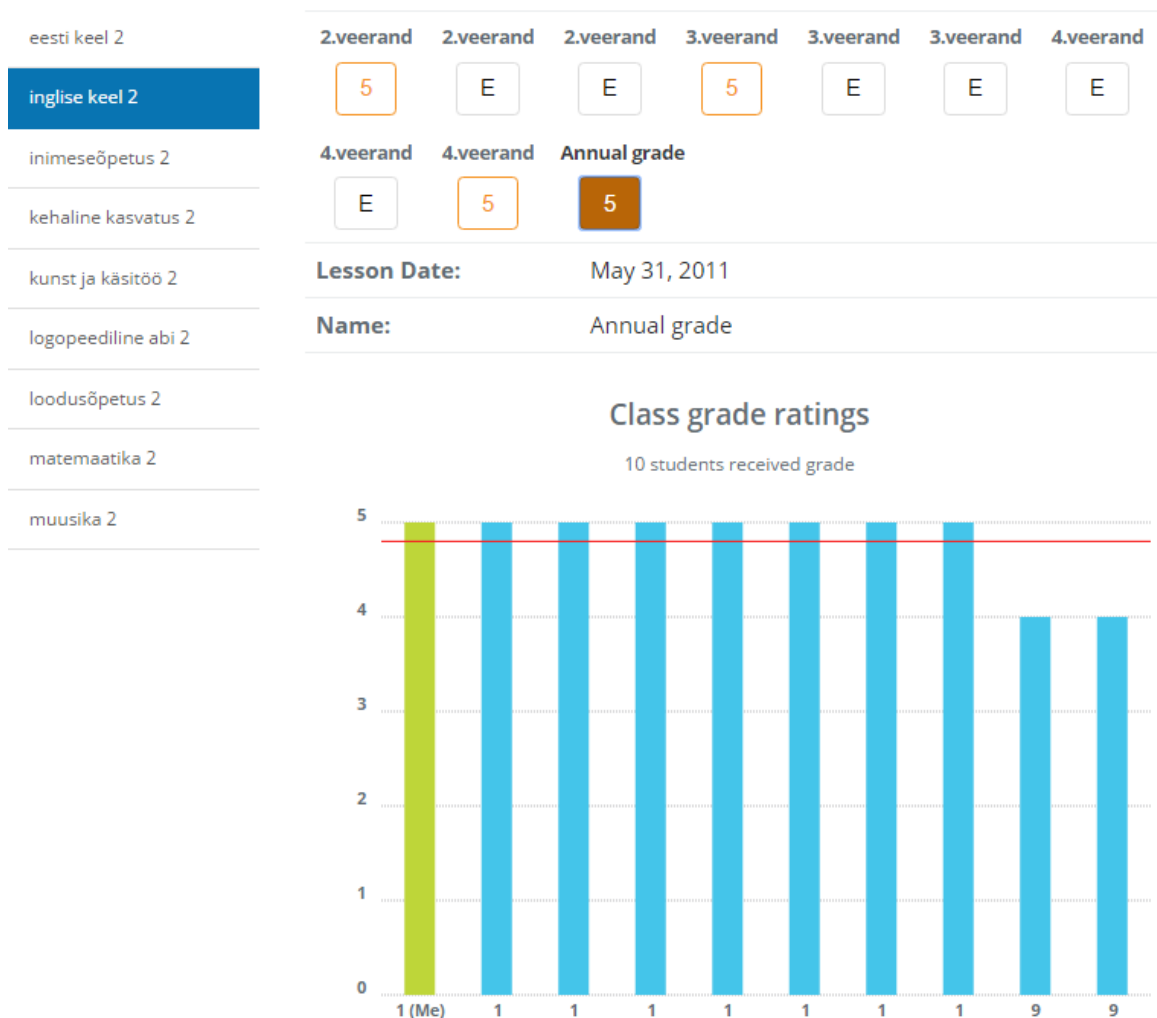
8.2. att. Klašu un periodu izvēlnes logs

Atzīmju sadaļā kreisajā lapas malā tiek attēloti lietotājam reģistrētie mācību priekšmeti. Lietotājs var izvēlēties priekšmetus un apskatīt visas izliktās atzīmes izvēlētajā periodā un mācību priekšmetā. Izvēloties priekšmetu, lapas augšpusē parādās atlasītās atzīmes. Lietotājs uzklikšķinot uz atzīmes var aplūkot atzīmes detaļas un statistikas diagrammu. Attēlā lietotājam attēlota apla diagramma (sk. 8.3. att.).



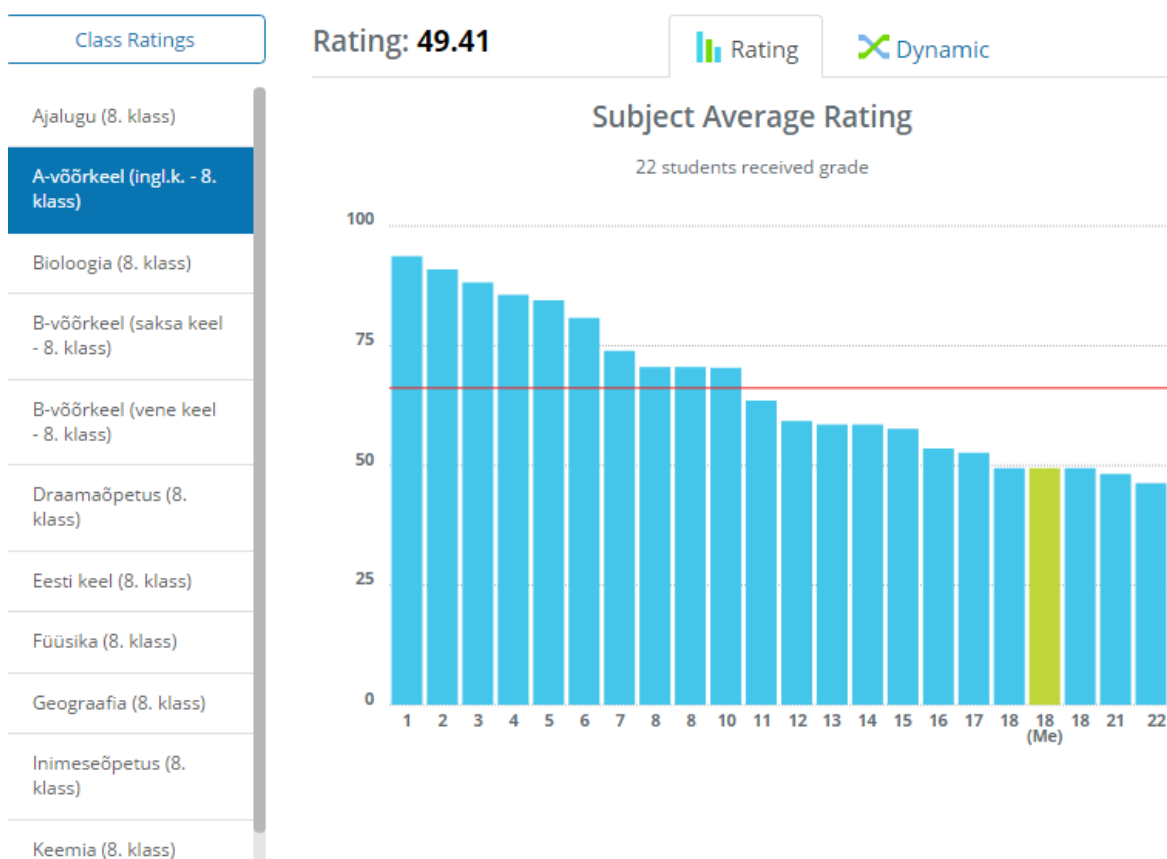
8.3. att. “Grades” - Atzīmju sadaļa

Izvēloties lietotāja izvēlnē mācību gadu, visu atzīmju sadaļā tiek attēloti visi lietotājam reģistrētie mācību priekšmeti. Lietotājam izvēloties mācību priekšmetu, lapas augšējā daļā parādās gala atzīmes. Gala atzīmes ir lietotājam izliktās atzīmes, kuras ir piešķirtas periodiem vai arī cita veida gala atzīmes, piemēram kursa atzīmes, ikgadējās atzīmes, eksāmenu atzīmes.



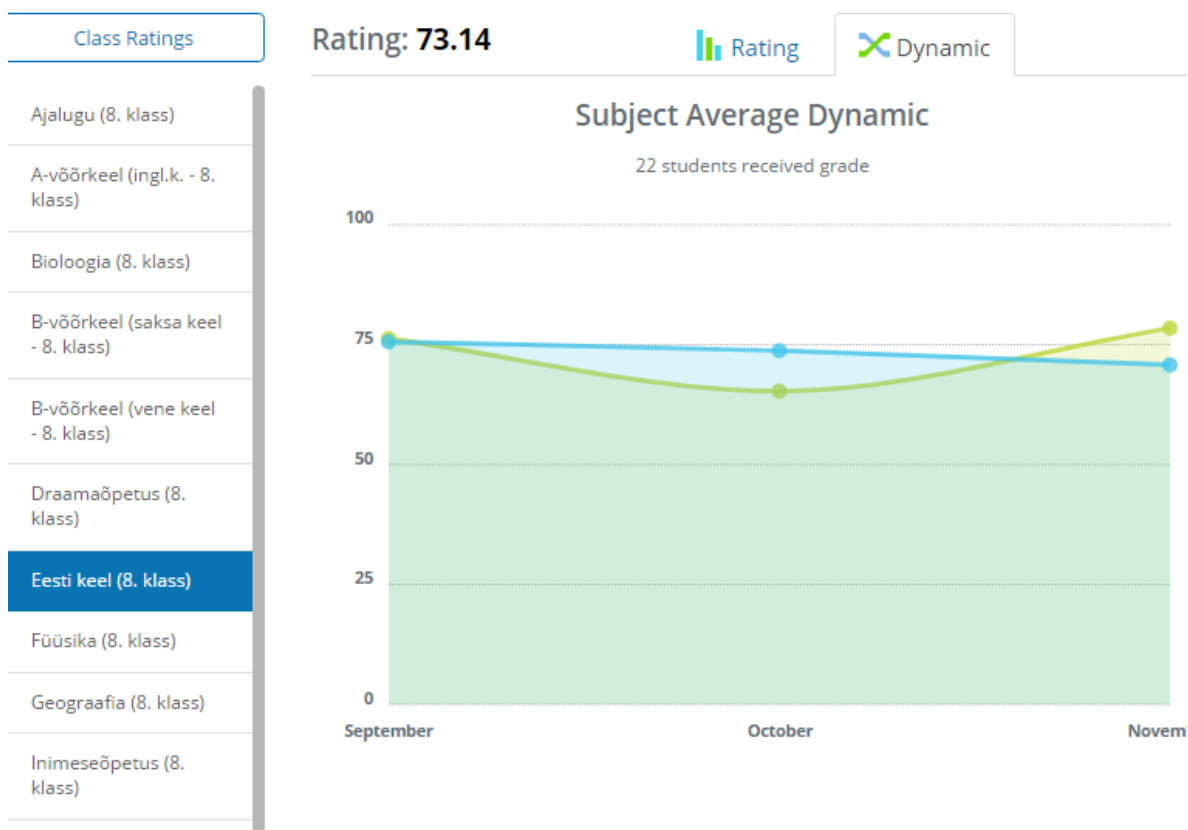
8.4. att. "Grades" – Atzīmju sadaļa ar izvēlētu mācību gadu

Reitingu sadaļā lietotājam kreisajā lapas malā tiek uzskaitīti pieejamie mācību priekšmeti. Lietotājam izvēloties mācību priekšmetu, lapas vidusdaļā tiek attēlots klases reitings izvēlētajā mācību priekšmetā un periodā. Izvēloties sadaļu “Class Ratings”, lietotājs redz reitingu, kurš tiek aprēķināts visos mācību priekšmetos. Diagrammā tiek izcelts lietotāja stabiņš un arī lietotāja sniegums 100 punktu skalā.



8.5. att. “Rating” - Reitingu sadaļa

Atzīmju dinamikas sadaļā lietotājs var aplūkot atzīmju izmaiņas laika gaitā. Lietotājs reitingu sadaļā izvēlas mācību priekšmetu, kur pēc tam virs klases reitinga var izvēlēties sadaļu “Rating”, kura jau ir atvērta, vai arī sadaļu “Dynamic”, kurā ir attēlota dinamika. Ja lietotājs ir atlasījis periodu, tad viņam dinamikā atzīmes tiek aprēķinātas un parādītas pa mēnešiem, bet ja lietotājs ir izvēlējis mācību gadu, tad atzīmes tiek aprēķinātas un attēlotas periodos.



8.6. att. “Dynamic” - atzīmju dinamikas sadaļa

SECINĀJUMI

Projekta izstrādes sākumā trešā daļa laika tika atvēlēta platformā izmantojamās datu bāzes analīzei un izstrādes vides sagatavošanai. Lai gan autoram pieredze darbā ar PHP un Laravel ietvaru ir neliela, praktiski visi lietotājtāsti tika realizēti. Funkcionalitāte tika izveidota visiem lietotājtāstiem, bet visus lietotājtāstus nebija iespējams notestēt produkcijas vidē. Neskatoties uz to, ka platforma joprojām tiek uzlabota, jau tagad ir izveidota prasībām atbilstoša programmatūra – izveidota tīmeklī bāzēta skolēnu sekmju analīzes platforma.

Izstrādājot šo projektu autors nostiprināja zināšanas PHP valodas un Laravel ietvara izmantošanā. Tika arī iegūtas jaunas zināšanas un apgūtas jaunas programmēšanas metodes. Realizējot programmas biznesa loģiku un veicot datu bāzes analīzi, tika uzlabotas analizēšanas spējas. Tika apgūta labas programmēšanas prakse un spēja optimizēt koda ātrdarbību. Izstrādes laikā autors konstatēja, ka Laravel ietvars ir ļoti daudzfunkcionāls un noderīgs turpmāko projektu izstrādē. Ietvars ir ļoti parocīgs, lai strukturētu sistēmas funkcionālo darbību un sistēmas failus. Ietvara funkcionalitāti var viegli apgūt lasot dokumentāciju un darbojoties praktiski. Iebūvētās funkcijas nodrošina failu sasaisti ar datu bāzi, kešatmiņu un citiem servisiem. Autors iesaka izmantot šo ietvaru jaunu projektu izstrādei.

Viens no svarīgākajiem ieguvumiem ir spēja izstrādāt kodu patstāvīgi, komunicēt ar vadītāju, realizēt nepieciešamo funkcionalitāti, sadarboties ar citiem izstrādātājiem, tai skaitā lietotāju saskarnes izstrādātāju. Uzlabota spēja komentēt un paskaidrot savu kodu koda apskatītājam, lai pēc tam veiktu izmaiņas un optimizētu kodu. Tika iegūta vērtīga pieredze, izstrādē izmantojot spējas izstrādes vadlīnijas. Tika iegūta pieredze koda kvalitātes uzlabošanā pēc koda apskates.

Nākotnes plānos ir turpināt “Grade Analytics” izstrādi. Ir paredzēts to integrēt Ekool sistēmā, kas līdz šim netika paspēts, jo sadarbība neritēja tik raiti. Nākotnē platformas kods tiks pārstrukturēts, tiks uzlabota ātrdarbība un paplašināta funkcionalitāte, kā arī tiks meklētas kļūdas kodā.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. Nr. LVS 68:1996. Informācijas tehnoloģija - Programminženierija – Programmatūras prasību specifikācijas ceļvedis. 1996. Pieejams: <http://estudijas.lu.lv/mod/resource/view.php?id=222577> (skatīts 16.05.2017.)
2. Laravel 5.4v dokumentācija. Pieejams: <https://laravel.com/docs/5.4>
3. PHP dokumentācija. Pieejams: <http://php.net/docs.php>
4. PostgreSQL dokumentācija. Pieejams: <https://www.postgresql.org/docs/>
5. Akadēmiskā terminu vārdnīca “AkadTerm”. Pieejams: <http://termini.lza.lv/term.php>
6. COCOMO Basic darbietilpības kalkulators. Pieejams: <http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/kutcher/kutcher.html>
7. COCOMO II darbietilpības kalkulators. Pieejams: <http://csse.usc.edu/tools/cocomoii.php>
8. NginX dokumentācija. Pieejams: <https://nginx.org/en/docs/>

PIELIKUMS

Pielikumā ir attēloti programmatūras kodā fragmenti.

Vistriviālākā kontroliera saturs. CourseController – mācību priekšmetu kontrolieris, atgriež lietotāja mācību priekšmetus.

```
class CourseController extends Controller
{
    protected $courses;

    public function __construct(CourseRepository $courses, ResponseFactory
$response)
    {
        $this->courses = $courses;
        $this->response = $response;
    }
    /**
     * USES USERS DATA TO RETURN COURSES USER IS REGISTERED TO
     **/
    public function getCourses($person, $role)
    {
        if($role == false)
        {
            return $this->response->json([
                'error' => __('error_messages.not_authorized'),
                'success' => false
            ]);
        }
        /**
         * SELECTING COURSES
         **/
        $courses = $this->courses->getCourses($role->id);

        if(!count($courses))
        {
            return $this->response->json([
                'error' => __('error_messages.subjects_not_found'),
                'success' => false
            ]);
        }
        foreach ($courses as $course)
        {
            $data[] = $course;
        }

        return $this->response->json([
            'data' => $data,
            'success' => true
        ]);
    }
}
```

Viena no maršruta ceļa sasaistīšanas RouteServiceProvider metodēm.

```
Route::bind('study_year', function($value) {
    $name = 'study_year_'. $value;
    /**
     * SELECTING STUDY YEAR
     **/
    $study_year = Cache::remember($name, 10, function ()
use($value)
    {
        return StudyYear::find($value);
    });
    if($study_year)
    {
        return $study_year;
    }
    abort(404, __('error_messages.study_year_not_found'));
});
```

Viens no datu bāzes datu atlasēm GradeRepository repozitorijā.

```
return DB::table('role')
->join(DB::raw("
(SELECT
    study_year_id,
    COALESCE(parallel_identifier_id, '1') AS parralel,
    class_level_id
FROM
    role
WHERE
    id = '". $role_id. "' AS r2"), function($join){
    $join->on('role.study_year_id', '=', 'r2.study_year_id');
    $join->on('role.class_level_id', '=', 'r2.class_level_id');
})
->join('grade_event', 'role.id', '=', 'grade_event.student_id')
->join('grade_value', 'grade_event.grade_value_id', '=', 'grade_value.id')
->leftJoin('journal_event', 'grade_event.journal_event_id', '=',
'journal_event.id')
->leftJoin('todo', 'grade_event.todo_institution_id', '=', 'todo.id')
->leftJoin('journal', 'journal_event.journal_id', '=', 'journal.id')
->leftJoin('journal AS jj', 'todo.journal_id', '=', 'jj.id')
->leftJoin('course', 'journal.course_id', '=', 'course.id')
->leftJoin('course AS cc', 'jj.course_id', '=', 'cc.id')
->where('grade_value.in_average', true)
->where('role.type_id', '2')
->whereRaw("COALESCE(role.parallel_identifier_id, '1') =
(
SELECT
    COALESCE(role.parallel_identifier_id, '1') AS parallel_2
FROM
    role
WHERE
    id = '". $role_id. "'
) ")
->whereRaw("grade_event.event_date BETWEEN
(SELECT
    start_date
FROM
    term
WHERE
```

```

        id = '". $term_id. "')
AND
(SELECT
    end_date
FROM
    term
WHERE
    id = '". $term_id. "')
->where(function ($query) use ($course_id){
    return $query->where('course.id', $course_id)
    ->orWhere('cc.id', $course_id);})
->groupBy('role.id', 'grade_value.grade_schema_id')
->get([DB::raw('ROUND(AVG(grade_value.gpa), 2) AS value'),
DB::raw('COUNT(grade_value.gpa) AS count'), 'role.id',
'grade_value.grade_schema_id']);

```

Viena no GradeEventRepository metodēm, kura masīva datus sakārto dilstošā secībā un atzīmju vērtībām piešķir vietas pēc kārtas.

```

public function sortAndAddPlacesToCompareSingleGradeChart($chart_data)
{
    arsort($chart_data);
    $place = 1;
    $counter = 1;
    $grade_tmp = null;
    foreach ($chart_data as $chart_item)
    {
        $tmpgrade = null;
        if($chart_item['value'] != $grade_tmp)
        {
            $place = $counter;
            $grade_tmp = $chart_item['value'];
        }
        $tmpgrade['place'] = $place;
        $tmpgrade['value'] = $chart_item['value'];
        $tmpgrade['name'] = $chart_item['name'];
        $tmpgrade['me'] = $chart_item['me'];
        $counter++;
        $chart[] = $tmpgrade;
    }

    return $chart;
}

```


Kvalifikācijas darbs „**Skolēnu sekmju analīzes platformas izstrāde**” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: **Matīss Zalāns** _____ .05.2017.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: **B. E-kom. Ģirts Ļaudaks** _____ .05.2017.

Recenzents: **M.dat. Agnis Škuškovniks**

Darbs iesniegts 29.05.2017.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: **Darja Solodovņikova** _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2017. prot. Nr. _____

Komisijas sekretārs(-e): _____