

LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

TĪMEKĻA LIETOTNE SUDOKU MĪKLU RISINĀŠANAI

KVALIFIKĀCIJAS DARBS

Autors: **Reinis Šadinovs**

Studenta apliecības Nr. rs17059

Darba vadītājs: B.dat. Jānis Knets

RĪGA 2019

ANOTĀCIJA

Sudoku mīklu risināšanas lietotne “Sudoku App” ir tīmeklī bāzēta programmatūra, kas ļauj neregistrētiem un reģistrētiem lietotājiem risināt sudoku, kā arī, reģistrētiem lietotājiem, saglabāt savu risinājumu vēsturi ar to atrisināšanai patērēto laiku, turpināt jau iepriekš iesāktās mīklas un apskatīt primitīvu statistiku par lietotāja risinātajām sudoku mīklām.

Vietnes izstrāde veikta pēc Kanban spējās izstrādes ietvara. Programmatūra izmanto NodeJS tehnoloģiju ar express bibliotēku aizmugursistēmas izstrādē, un ReactJS tehnoloģiju lietotāja saskarnes izstrādē.

Atslēgas vārdi: Javascript, NodeJS, ReactJS, MongoDB, sudoku.

ABSTRACT

Web application for solving sudoku puzzles

Sudoku solving application “Sudoku App” is application based in web which lets registered and unregistered users to solve sudokus in different difficulty settings. Registered users also have an additional functionality such as automatic history saving for solved and started sudokus with time counter as well as continue to solve already started sudokus and look at simple statistics about their solved sudoku puzzles.

Development of application was done after Kanban framework of agile software development. Application uses NodeJs Javascript runtime with express library for server side development and ReactJs library for user interface development.

Keywords: Javascript, NodeJS, ReactJS, MongoDB, sudoku.

SATURS

APZĪMĒJUMU SARAKSTS	6
IEVADS	7
1. Programmatūras prasību specifikācija	8
1.1 Ievads	8
1.1.1 Nolūks	8
1.1.2 Darbības sfēra.....	8
1.1.3 Saistība ar citiem dokumentiem	8
1.1.4 Pārskats.....	8
1.2 Vispārējais apraksts	9
1.2.1 Produkta perspektīva	9
1.2.2 Produkta funkcijas.....	9
1.2.3 Lietotāja raksturiezīmes	11
1.2.4 Vispārējie ierobežojumi	11
1.2.5 Pieņēmumi un atkarības	11
1.3 Funkcionālās prasības	12
1.3.1 Aizmugursistēma.....	12
1.3.2 Lietotāja saskarne	28
1.4 Nefunkcionālās prasības	40
1.4.1 Valoda	40
1.4.2 Drošības prasības.....	40
1.4.3 Veiktspējas prasības	40
1.4.4 Uzturamība	40
1.4.5 Pieejamība	40
1.4.6 Atbalsts tīmekļa pārlūkprogrammām	40
2. Programmatūras projektējuma apraksts	41
2.1 Ievads	41
2.1.1 Nolūks	41
2.1.2 Darbības sfēra.....	41
2.1.3 Saistība ar citiem dokumentiem	41
2.2 Dekompozīcijas apraksts	41
2.2.1 Aizmugursistēma.....	41
2.2.2 Lietotāja saskarne	43
2.3 Datu bāzes projektējums	48
2.3.1 Datu bāzes kolekciju apraksts	48
3. Funkciju projektējums	51
3.1 0. līmeņa datu plūsmas diagramma.....	51

3.2	1. līmeņa datu plūsmas diagramma.....	51
3.3	2. līmeņa datu plūsmas diagramma.....	52
3.3.1	Autorizācijas modulis.....	52
3.3.2	Lietotāja modulis.....	53
3.3.3	Sudoku modulis.....	53
3.3.4	Vēstures modulis.....	54
3.3.5	Administratora modulis.....	54
4.	Testēšanas dokumentācija.....	55
4.1	Ievads.....	55
4.1.1	Apraksts.....	55
4.1.2	Saistība ar citiem dokumentiem.....	55
4.2	Testēšanas žurnāls.....	55
4.2.1	Aizmugursistēma.....	55
4.2.2	Lietotāja saskarne.....	61
5.	Projekta organizācija.....	74
6.	Kvalitātes nodrošināšana.....	75
7.	Konfigurāciju pārvaldība.....	76
8.	Darbietilpības novērtējums.....	77
9.	Secinājumi.....	78
10.	Izmantotā literatūra.....	79
11.	Pielikums.....	80

APZĪMĒJUMU SARAKSTS

Apzīmējums	Skaidrojums
Javascript	Augsta līmeņa skriptu valoda, ko izmanto tīmekļa vietņu veidošanā.
HTML	(Hyper Text Markup Language) Iezīmēšanas valoda, ko izmanto tīmekļa vietņu veidošanā.
CSS	(Cascading Style Sheets) Valoda, ko izmanto iezīmēšanas valodā veidotu dokumentu izskata aprakstīšanā.
JSON	(JavaScript Object Notation) ir datu apmaiņas formāts.
HTTP	(HyperText Transfer Protocol) Pieprasījuma un atbildes protokols starp klientiem un serveriem.
API	(Application programming interface) Lietojumprogrammas saskarne, kurai, padodot HTTP pieprasījumus, ir iespējams iegūt specifisku informāciju.
NodeJS	Javascript izpildlaiks, kas balstīts uz Chrome V8 Javascript programmas. [1.]
ReactJS	Javascript bibliotēka lietotāja saskarnes veidošanai. [2.]
JSX	(Javascript with XML) ir sintakses paplašinājums Javascript valodai. [3.]
MongoDB	Bez-relāciju datu bāze, kurā dati tiek uzglabāti BSON formātā (Binary JSON). [4.]
JWT	(JSON Web Token) Tas ir atvērtais standarts, kas definē kompaktu un neatkarīgu veidu, lai droši sūtītu informāciju starp iesaistītajām pusēm, JSON objekta formātā. [5.]
Kanban	Spējās izstrādes ietvars.
Redux	Prognozējams stāvokļu ietvars Javascript lietotnēm. [6.]

IEVADS

Tīmekļa lietotņu izstrādes tehnoloģijas nepārtraukti attīstās, tāpēc autors nolēma padziļināti apgūt vienas no šī brīža populārākajām tehnoloģijām - ReactJS un Express, ko apguva prakses laikā. Līdzšinējās tēmas izvēle neradīja problēmas, jo bija nepieciešams izstrādāt tādu vietni, kuras funkcionalitāte būtu sarežģītāka par vienkāršu tīmekļa vietni ar lietotāju autentifikāciju un primitīvu ievaddatu saglabāšanu un attēlošanu. Ideja par sudoku mīklu risināšanas lietotni uzreiz kļuva pievilcīga, jo autors, brīvajos dienas brīžos, mēdz tās risināt, kā arī mīklas risināšanas un pārbaudes funkcionalitātes ieviešanai nepieciešami jau salīdzinoši sarežģītāki algoritmi.

Kvalifikācijas darba mērķis ir 3 personmēnešu laikā izstrādāt sudoku mīklu risināšanas tīmekļa lietotni "Sudoku App", kas dotu iespēju lietotājiem risināt dažādu grūtības pakāpju sudoku mīklas un uzņemt laiku, cik ilgi tā tika risināta. Reģistrētiem lietotājiem iesāktie un atrisinātie sudoku tiks saglabāti, un būs pieejami turpināšanai vai apskatei, kā arī būs iespēja aplūkot savu progresu un nelielu statistiku.

Dokuments sastāv no Programmatūras Prasību Specifikācijas, kurā tiek aprakstītas funkcionālās daļas prasības, no Programmatūras Projektējuma Apraksta, kurā tiek veikts funkcionālo un nefunkcionālo prasību realizācijas projektējums, no zema līmeņa funkciju projektējuma, no testēšanas dokumentācijas, kurā tiek aprakstīti vienībtestēšanas rezultāti, kā arī projekta organizācijas, kvalitātes nodrošināšanas, konfigurāciju pārvaldības, darbietilpības novērtējuma, secinājumiem, izmantotās literatūras saraksta un pielikuma.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1 Ievads

1.1.1 Nolūks

Programmatūras prasību specifikācija (PPS) ir paredzēta tīmekļa lietotnes “Sudoku App” funkcionālās daļas prasību aprakstīšanai, lai atbilstoši specifikācijai tiktu izstrādāta aprakstītā programmatūra.

1.1.2 Darbības sfēra

Tīmekļa lietotne “Sudoku App” paredzēta sudoku mīklu risināšanai ar laika uzskaiti, risinājumu vēstures saglabāšanai un iesākto sudoku iespējama turpināšanai.

1.1.3 Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standarta LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis prasības”.

Dokuments ir saistīts ar programmatūras projektējuma aprakstu un testēšanas dokumentāciju.

1.1.4 Pārskats

Dokuments sastāv no 4 galvenajām daļām: ievada, vispārējā apraksta, funkcionālajām un nefunkcionālajām prasībām.

Ievads ietver nolūku, darbības sfēru, saistību ar citiem dokumentiem un dokumenta pārskatu.

Vispārējais apraksts ietver produkta perspektīvu, produkta funkcijas ar lietotāju piekļuvēm, lietotāja raksturiezīmes, sistēmas vispārējos ierobežojumus, kā arī pieņēmumus un atkarības.

Funkcionālās prasības sevī ietver aizmugursistēmas un lietotāja saskarnes moduļu funkcionālās prasības. Aizmugursistēmas ietvarā pastāv 5 moduļi, t.i., autorizācijas modelis, lietotāja modelis, sudoku modelis, vēstures modelis un administratora modulis, toties lietotāja saskarnē sastopami astoņi moduļi, t.i., pieteikšanās modulis, reģistrācijas modulis, pamata modulis, sudoku modulis, vēstures modulis, lietotāja profila modulis, administratora modulis un sudoku pievienošanas modulis.

Nefunkcionālās prasības sastāv no valodas prasībām, piemērojamības, drošības prasībām, veiktspējas prasībām, uzturamības, ierobežojumiem un pārlūkprogrammu atbalsta.

1.2 Vispārējais apraksts

1.2.1 Produkta perspektīva

Produkta perspektīva iekļauj minimālistisku un ērtu tīmekļa vietni sudoku mīklu risināšanai. Neregistrētiem lietotājiem tiek nodrošināta iespēja izvēlēties starp četrus grūtības pakāpju mīklām, tās risināt ar laika uzskaiti, pārbaudīt risinājumu un parādīt atbildi. Reģistrētiem lietotājiem tiek nodrošināta tāda pati funkcionalitāte, kā neregistrētiem lietotājiem, tikai ar papildinājumiem: vēstures saglabāšanu ar mīklas risinājuma iegūšanai atvēlēto, iesākto sudoku mīklu turpināšanu, lietotāja informācijas atjaunināšanu un nelielu statistiku. Administratoriem tiek nodrošināta iepriekš minētā funkcionalitāte un piekļuve administratora panelim ar vietnes statistiku un visiem lietotājiem, kurus iespējams dzēst, kā arī jaunu sudoku pievienošanas iespēja sistēmai.

1.2.2 Produkta funkcijas

1.2.2.1 Aizmugursistēma

Moduļa numurs	Modulis	Neautorizēts lietotājs	Autorizēts lietotājs	Moderators	Administrators
1.3.1.1	Autorizācijas modulis				
1.3.1.1.1	Lietotāja reģistrācija	x			
1.3.1.1.2	Lietotāja pieteikšanās	x	x	x	x
1.3.1.2	Lietotāja modulis				
1.3.1.2.1	Lietotāja informācijas iegūšana		x	x	x
1.3.1.2.2	Lietotāja informācijas atjaunošana		x	x	x
1.3.1.2.3	Visu lietotāju informācijas iegūšana			x	x
1.3.1.2.4	Lietotāja dzēšana no sistēmas			x	x
1.3.1.3	Sudoku modulis				
1.3.1.3.1	Visu sudoku iegūšana	x	x	x	x
1.3.1.3.2	Sudoku iegūšana pēc id	x	x	x	x
1.3.1.3.3	Visu sudoku iegūšana pēc grūtības pakāpes	x	x	x	x

1.3.1.3.4	Nejauši izvēlēta sudoku iegūšana pēc grūtības pakāpes	x	x	x	x
1.3.1.3.5	Nejauši izvēlēta sudoku iegūšana pēc grūtības pakāpes autorizētiem lietotājiem		x	x	x
1.3.1.3.6	Sudoku pievienošana			x	x
1.3.1.3.7	Sudoku atjaunināšana			x	x
1.3.1.3.8	Sudoku dzēšana			x	x
1.3.1.4	Vēstures modulis				
1.3.1.4.1	Visu lietotāju vēstures iegūšana			x	x
1.3.1.4.2	Lietotāja dalītās vēstures iegūšana		x	x	x
1.3.1.4.3	Vēstures ieraksta iegūšana		x	x	x
1.3.1.4.4	Vēstures ieraksta pievienošana		x	x	x
1.3.1.4.5	Lietotāja vēstures statistikas iegūšana		x	x	x
1.3.1.4.6	Lietotāja vēstures ieraksta esamības pārbaude		x	x	x
1.3.1.4.7	Vēstures ieraksta atjaunināšana		x	x	x
1.3.1.4.8	Vēstures ieraksta dzēšana			x	x
1.3.1.5	Administratora modulis				
1.3.1.5.1	Administratora paneļa informācijas iegūšana			x	x

1.2.2.2 Lietotāja saskarne

Moduļa numurs	Modulis	Neautorizēts lietotājs	Autorizēts lietotājs	Moderators	Administrators
1.3.2.1	Pieteikšanās modulis				
1.3.2.1.1	Lietotāja pieteikšanās	x			
1.3.2.2	Reģistrācijas modulis				
1.3.2.2.1	Lietotāja reģistrācija	x			
1.3.2.3	Pamata modulis				
1.3.2.3.1	Galvene	x	x	x	x
1.3.2.3.2	Grūtības pakāpes izvēlne	x	x	x	x
1.3.2.4	Sudoku modulis				
1.3.2.4.1	Sudoku ietvars	x	x	x	x
1.3.2.4.2	Sudoku laukums	x	x	x	x

1.3.2.5	Vēstures modulis				
1.3.2.5.1	Vēstures saraksts		x	x	x
1.3.2.5.2	Vēstures ieraksts		x	x	x
1.3.2.6	Lietotāja profila modulis				
1.3.2.6.1	Profila skats		x	x	x
1.3.2.6.2	Lietotāja rediģēšana		x	x	x
1.3.2.7	Administratora modulis				
1.3.2.7.1	Administratora panelis			x	x
1.3.2.7.2	Lietotāja informācija			x	x
1.3.2.8	Sudoku pievienošanas modulis				
1.3.2.8.1	Sudoku pievienošanas skats			x	x
1.3.2.8.2	Pievienošanas apstiprinājums			x	x

1.2.3 Lietotāja raksturiezīmes

Neregistrētajiem un reģistrētajiem lietotājiem ar līmeņiem “user” un “moderator” nepieciešama prasme darboties ar datoru un ar tām uzstādīto interneta pārlūkprogrammu.

Reģistrētajiem lietotājiem ar līmeni “admin”, kas būs atbildīgi par sistēmas izstrādi un uzturēšanu, nepieciešamas prasmes ar koda rediģēšanas lietotni, versiju kontroles pārvaldību, servera pārvaldības sistēmu un interneta pārlūkprogrammām.

1.2.4 Vispārējie ierobežojumi

Sistēmas darbības nodrošināšanai nepieciešams interneta pieslēgums. Sistēma izvietota tīmeklī, tādējādi tās darbībai nav nepieciešama instalēšana, bet to darbina izmantojot pārlūkprogrammu.

1.2.5 Pieņēmumi un atkarības

Sistēma sastāv no lietotāj saskarnes, tāpēc tās izmantošanai nepieciešama ierīce ar grafisko saskarni. Sistēmas darbībai nepieciešams interneta savienojums un jāizmanto pārlūkprogramma, kas atbalsta jaunākos HTML, CSS un JS standartus – tāpat kāda no populārākajām interneta pārlūkprogrammām.

1.3 Funkcionālās prasības

1.3.1 Aizmugursistēma

1.3.1.1 Autorizācijas modulis

1.3.1.1.1 Lietotāja reģistrācija

Ceļš:	POST: /api/v1/register			
Apraksts:	Funkcija nodrošina jaunu lietotāju reģistrāciju sistēmā.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājevārds	Rakstzīmju dati	x	Rakstzīmju virkne, kas nodrošinās lietotāja identificēšanu
	e-pasts	Rakstzīmju dati	x	Rakstzīmju virkne, kas nodrošinās lietotāja identificēšanu un saziņas veidu ar lietotāju.
	parole	Rakstzīmju dati	x	Rakstzīmju virkne, kas nodrošinās piekļuvi sistēmai konkrētam lietotājam.
	līmenis	Rakstzīmju dati	x	Lietotāja līmenis, lai nodrošinātu piekļuvi konkrētai funkcionalitātei.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none">- Lietotājvārda pārbaude vai šāds lietotājs jau neeksistē sistēmā.- E-pasta pārbaude vai šāds e-pasts jau netiek izmantots sistēmā un vai tas ir atbilstošs e-pasta standartam.- Paroles šifrēšana pirms tā tiek sūtīta uz datu bāzi.- Lietotāja līmenis pēc noklusējuma tiks iestatīts kā "user".			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar paziņojumu "Successfully registered" un tikko reģistrētā lietotāja datu objektu.		
	400	Tiek atgriezts kļūdas paziņojums "Please provide valid email address" gadījumā, ja lietotājs ir ievadījis e-pasta adresi ne šādā formā: lietotājs + "@" zīme + e-pasta servera domēna vārds.		
	400	Tiek atgriezts kļūdas paziņojums "User already exists!" gadījumā, ja jau pastāv lietotājs ar ievadīto lietotājvārdu.		
	400	Tiek atgriezts kļūdas paziņojums "Email is already in use!" gadījumā, ja ievadītais e-pasts jau pastāv sistēmas datu bāzē.		

1.3.1.1.2 Lietotāja pieteikšanās

Ceļš	POST: /api/v1/login			
Apraksts:	Funkcija nodrošina lietotāju pieteikšanos sistēmai.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājavārds	Rakstzīmju dati	x	Rakstzīmju virkne, kas nodrošina lietotāja identificēšanu
	parole	Rakstzīmju dati	x	Rakstzīmju virkne, kas nodrošina piekļuvi sistēmai konkrētam lietotājam.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Lietotāja atrašana datu bāzē pēc ievadītā lietotājavārda. - Ievadītās paroles un atrastā lietotāja paroles salīdzināšana. - JWT paraksta izveide, iekļaujot datus par lietotāju (lietotājavārdu, e-pastu, līmeni, pievienošanās datumu, atjaunināšanas datumu), tos šifrējot ar apslēpto atslēgu, un pievienojot paraksta derīguma laiku – 6 stundas. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar paziņojumu “Successfully logged in” un tikko izveidoto JWT parakstu.		
	400	Tiek atgriezts kļūdas paziņojums “Wrong user credentials!” gadījumā, ja lietotājs ir nepareizi norādījis lietotājavārdu un/vai paroli.		

1.3.1.2 Lietotāja modulis

1.3.1.2.1 Lietotāja informācijas iegūšana

Ceļš:	GET: /api/v1/users/self			
Apraksts:	Funkcija nodrošina lietotāja, kurš ir pieteicies sistēmā, informācijas atgriešanu.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	-			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar lietotāja datu objektu.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.2.2 Lietotāja informācijas atjaunošana

Ceļš:	PATCH: /api/v1/users/:userId			
Apraksts:	Funkcija nodrošina lietotāja informācijas atjaunošanu.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	userId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā lietotāju identifikators.
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
	lietotājvārds	Rakstzīmju dati		Rakstzīmju virkne, kas nodrošina lietotāja identificēšanu
	e-pasts	Rakstzīmju dati		Rakstzīmju virkne, kas nodrošina lietotāja identificēšanu un saziņas veidu ar lietotāju.
	parole	Rakstzīmju dati		Rakstzīmju virkne, kas nodrošina piekļuvi sistēmai konkrētam lietotājam.
	līmenis	Rakstzīmju dati		Lietotāja līmenis, lai nodrošinātu piekļuvi konkrētai funkcionalitātei.
Apstrāde:	<p>Datu apstrādē tiks veiktas šādas darbības:</p> <ul style="list-style-type: none"> - Pārbaude vai lietotāja informācijas atjaunotājs ir pats lietotājs vai administrators. - Pārbaude, kuri lauki ir norādīti ievaddatos, jo lietotāji nevēlas, lai, lauka nenorādīšanas gadījumā, informācija tiktu atjaunota uz tukšu vienību. - Lietotājvārda norādīšanas gadījumā, pārbaude vai šāds lietotājs jau netiek izmantots sistēmā. - E-pasta norādīšanas gadījumā, pārbaude vai šāds e-pasts jau netiek izmantots sistēmā. - Lietotāja līmeņa norādīšanas gadījumā, pārbaude vai lietotāju atjaunina administrators, jo pats lietotājs nedrīkst atjaunināt savu līmeni. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar tikko atjauninātā lietotāja datu objektu.		
	400	Tiek atgriezts kļūdas paziņojums "User not found!" gadījumā, ja netika atrasts lietotājs ar norādīto identifikatoru.		
	400	Tiek atgriezts kļūdas paziņojums "Username is already taken!" gadījumā, ja jau pastāv lietotājs ar ievadīto lietotājvārdu.		
	400	Tiek atgriezts kļūdas paziņojums "Email is already taken!" gadījumā, ja ievadītais e-pasts jau pastāv sistēmas datu bāzē.		
400	Tiek atgriezts kļūdas paziņojums "Only admin or user himself can update user information!" gadījumā, ja lietotāju mēģina			

		atjaunināt kāds cits lietotājs, kuram nepiemīt administratora līmenis.
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.

1.3.1.2.3 Visu lietotāju informācijas iegūšana

Ceļš:	GET: /api/v1/users			
Apraksts:	Funkcija nodrošina visu lietotāju informācijas iegūšanu.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: - Lietotāja līmeņa pārbaude vai tas atbilst moderatoram vai administratoram.			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar visu lietotāju informācijas objektu masīvu.		
	400	Tiek atgriezts kļūdas paziņojums “Only admin or moderator can get all users” gadījumā, ja lietotāju mēģina atjaunināt kāds cits lietotājs, kuram nepiemīt moderatora vai administratora līmenis.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.2.4 Lietotāja dzēšana no sistēmas

Ceļš:	DELETE: /api/v1/users/:userId			
Apraksts:	Funkcija nodrošina lietotāja informācijas un lietotājam piederošo vēstures ierakstu dzēšanu no sistēmas.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	userId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā lietotāju identifikators.
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.

Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Pārbaude vai lietotājs, kurš vēlas izdzēst norādīto lietotāju ir pats lietotājs vai administrators. - Lietotāja dzēšana un tam atbilstošo vēstures ierakstu dzēšana no sistēmas 	
Izvaddati:	Statuss	Dati/Skaidrojums
	200	Dati tiek padoti objekta veidā ar paziņojumu "User succesfully deleted!".
	400	Tiek atgriezts kļūdas paziņojums "User not found!" gadījumā, ja netika atrasts lietotājs ar norādīto identifikatoru.
	400	Tiek atgriezts kļūdas paziņojums "Only admin or user himself can delete user information!" gadījumā, ja lietotāju mēģina dzēst kāds cits lietotājs, kuram nepiemīt administratora līmenis.
	401	Tiek atgriezts kļūdas paziņojums "No token provided or token expired" gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.

1.3.1.3 Sudoku modulis

1.3.1.3.1 Visu sudoku iegūšana

Ceļš:	GET: /api/v1/sudoku			
Apraksts:	Funkcija nodrošina visu sudoku iegūšanu.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	-	-	-	-
Apstrāde:	-			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar visu iegūto sudoku objektu masīvu.		

1.3.1.3.2 Sudoku iegūšana pēc id

Ceļš:	GET: /api/v1/sudoku/:sudokuId			
Apraksts:	Funkcija nodrošina sudoku iegūšanu pēc norādītā identifikatora			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	sudokuId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā sudoku identifikators.
Apstrāde:	-			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar iegūtā sudoku objektu		
	400	Tiek atgriezts kļūdas paziņojums "Sudoku not found!" gadījumā, ja netika atrasts sudoku ar norādīto identifikatoru.		

1.3.1.3.3 Visu sudoku iegūšana pēc grūtības pakāpes

Ceļš:	GET: /api/v1/sudoku/difficulty/:difficulty			
Apraksts:	Funkcija nodrošina visu sudoku iegūšanu norādītajā grūtības pakāpē.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	grūtības pakāpe	Rakstzīmju dati	x	Rakstzīmju virkne, kas norāda sudoku grūtības pakāpi.
Apstrāde:	-			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar iegūto sudoku objektu masīvu.		
	400	Tiek atgriezts kļūdas paziņojums “Cannot find sudoku with given difficulty” gadījumā, ja netika atrasts sudoku ar norādīto grūtības pakāpi.		

1.3.1.3.4 Nejauši izvēlēta sudoku iegūšana pēc grūtības pakāpes

Ceļš:	GET: /api/v1/sudoku/random/difficulty/:difficulty			
Apraksts:	Funkcija nodrošina nejauši izvēlēta sudoku atgriešanu no norādītās grūtības pakāpes.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	grūtības pakāpe	Rakstzīmju dati	x	Rakstzīmju virkne, kas norāda sudoku grūtības pakāpi.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Iegūti visu atrasto sudoku objektu identifikatori un ievietoti vienā masīvā. - Iegūts nejauši izvēlēts skaitlis diapazonā no 0 līdz iepriekš izveidotā masīva garumam atņemot 1. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar sudoku objektu, kura indekss ir vienāds ar nejauši izvēlēto skaitli.		
	400	Tiek atgriezts kļūdas paziņojums “Cannot find sudoku with given difficulty” gadījumā, ja netika atrasts sudoku ar norādīto grūtības pakāpi.		

1.3.1.3.5 Nejauši izvēlēta sudoku iegūšana pēc grūtības pakāpes autorizētiem lietotājiem

Ceļš:	GET: /api/v1/sudoku/random/auth/difficulty/:difficulty			
Apraksts:	Funkcija nodrošina nejauši izvēlēta sudoku atgriešanu no norādītās grūtības pakāpes, ja lietotājs ir pieteicies sistēmā.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	grūtības pakāpe	Rakstzīmju dati	x	Rakstzīmju virkne, kas norāda sudoku grūtības pakāpi.
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	<p>Datu apstrādē tiks veiktas šādas darbības:</p> <ul style="list-style-type: none"> - Iegūti visu atrasto sudoku objektu identifikatori un ievietoti vienā masīvā. - Iegūti visi sudoku identifikatori no lietotāja vēstures ierakstiem un ievietoti vienā masīvā. - Salīdzināti abi iepriekš iegūtie masīvi, un visi identifikatori no pirmā masīva, ar visu atrasto sudoku identifikatoriem, kuri nav sastopami vēstures masīvā tiek ievietoti jaunā masīvā, kas reprezentē pieejamo sudoku mīklu identifikatorus - Iegūts nejauši izvēlēts skaitlis diapazonā no 0 līdz iepriekš izveidotā masīva garumam atņemot 1. - Tiek iegūts sudoku objekts, izmantojot sudoku identifikatoru no pieejamo identifikatoru masīva, kurā elementa indekss ir vienāds ar nejauši izvēlēto skaitli 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar gala rezultātā iegūto sudoku objektu.		
	400	Tiek atgriezts kļūdas paziņojums “Cannot find sudoku with given difficulty” gadījumā, ja netika atrasts sudoku ar norādīto grūtības pakāpi.		
	400	Tiek atgriezts kļūdas paziņojums “Cannot find user!” gadījumā, ja netika atrasts lietotājs.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.3.6 Sudoku pievienošana

Ceļš:	POST: /api/v1/sudoku			
Apraksts:	Funkcija nodrošina jaunas sudoku mīklas pievienošanu sistēmai.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
	grūtības pakāpe	Rakstzīmju dati	x	Rakstzīmju virkne, kas norāda sudoku grūtības pakāpi.
	sudoku	Rakstzīmju dati	x	Rakstzīmju virkne, kura atbilst sudoku mīklai, kas savienota vienā simbolu virknē pa rindām, sākot no augšas, aizstājot tukšās vietas ar "0".
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Pārbaude vai jaunā sudoku pievienotājs ir moderatora vai administratora līmenī. - Pārbaude vai ievadītā sudoku rakstzīmju virknes garums ir tieši 81. - Pārbaude vai ievadītais sudoku jau neeksistē datu bāzē. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar paziņojumu "Successfully added sudoku!" un jaunizveidotā sudoku objektu.		
	400	Tiek atgriezts kļūdas paziņojums "Only Admin or Moderator can add new sudoku!" gadījumā, ja sudoku mēģina pievienot lietotājs, kas nav moderatora vai administratora līmenī.		
	400	Tiek atgriezts kļūdas paziņojums "Sudoku length must be 81!" gadījumā, ja tiek mēģināts pievienot datu bāzei sudoku, kura garums nav vienāds ar 81.		
	400	Tiek atgriezts kļūdas paziņojums "Sudoku already exists!" gadījumā, ja datu bāzē jau pastāv šāds sudoku.		
	401	Tiek atgriezts kļūdas paziņojums "No token provided or token expired" gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.3.7 Sudoku atjaunināšana

Ceļš:	PATCH: /api/v1/sudoku/:sudokuId			
Apraksts:	Funkcija nodrošina norādītās sudoku mīklas atjaunināšanu sistēmā.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
	sudokuId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā sudoku identifikators.
	grūtības pakāpe	Rakstzīmju dati		Rakstzīmju virkne, kas norāda sudoku grūtības pakāpi.
	sudoku	Rakstzīmju dati		Rakstzīmju virkne, kura atbilst sudoku mīklai, kas savienota vienā simbolu virknē pa rindām, sākot no augšas, aizstājot tukšās vietas ar "0".
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Pārbaude vai sudoku informācijas atjaunotājs ir moderatora vai administratora līmenī. - Pārbaude vai ievadītā sudoku rakstzīmju virknes garums ir tieši 81, ja tas vispār tiek ievadīts. - Pārbaude vai ievadītais sudoku jau neeksistē datu bāzē. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar atjaunotā sudoku objektu.		
	400	Tiek atgriezts kļūdas paziņojums "Only Admin or Moderator can update sudoku!" gadījumā, ja sudoku mēģina atjaunināt lietotājs, kas nav moderatora vai administratora līmenī.		
	400	Tiek atgriezts kļūdas paziņojums "Sudoku not found!" gadījumā, ja nepastāv sudoku ar norādīto identifikatoru.		
	400	Tiek atgriezts kļūdas paziņojums "Sudoku length must be 81!" gadījumā, ja tiek mēģināts atjaunināt sudoku, kura garums nav vienāds ar 81.		
	400	Tiek atgriezts kļūdas paziņojums "Sudoku already exists!" gadījumā, ja datu bāzē jau pastāv šāds sudoku.		
401	Tiek atgriezts kļūdas paziņojums "No token provided or token expired" gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.			

1.3.1.3.8 Sudoku dzēšana

Ceļš:	DELETE: /api/v1/sudoku/:sudokuId			
Apraksts:	Funkcija nodrošina norādītā sudoku dzēšanu no sistēmas.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	sudokuId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā sudoku identifikators.
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Pārbaude vai lietotājs, kurš vēlas izdzēst norādīto sudoku ir moderatora vai administratora līmenī. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar paziņojumu “Sudoku succesfully deleted!”.		
	400	Tiek atgriezts kļūdas paziņojums “Sudoku not found!” gadījumā, ja netika atrasts sudoku ar norādīto identifikatoru.		
	400	Tiek atgriezts kļūdas paziņojums “Only Admin or Moderator can delete sudoku!” gadījumā, ja sudoku mēģina dzēst lietotājs, kuram nepiemīt moderatora vai administratora līmenis.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.4 Vēstures modulis

1.3.1.4.1 Visu lietotāju vēstures iegūšana

Ceļš:	GET: /api/v1/history			
Apraksts:	Funkcija nodrošina visu lietotāju vēstures ierakstu iegūšanu.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Pārbaude vai lietotājs, kurš vēlas iegūt visu lietotāju vēsturi ir moderatora vai administratora līmenī. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar visu iegūto vēstures ierakstu objektu masīvu.		

	400	Tiek atgriezts kļūdas paziņojums “Only Admin or Moderator can get all History entries!” gadījumā, ja visu lietotāju vēstures ierakstus mēģina iegūt lietotājs, kuram nepiemīt moderatora vai administratora līmenis.
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.

1.3.1.4.2 Lietotāja dalītās vēstures iegūšana

Ceļš:	GET: /api/v1/history/divided			
Apraksts:	Funkcija nodrošina lietotāja dalītās vēstures iegūšanu, kas sastāv no: <ul style="list-style-type: none"> - Vēstures ieraksti, kur sudoku vēl nav līdz galam atrisināts; - Vēstures ieraksti, kur sudoku ir atrisināts. 			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Iegūst masīvu ar vēstures ierakstiem, kur lietotājs nav pabeidzis sudoku. - Iegūst masīvu ar vēstures ierakstiem, kur lietotājs ir pabeidzis sudoku. - Abos masīvos tiek apskatīts katrs elements, un iegūti sudoku objekti attiecīgi šiem elementiem. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar izveidoto nepabeigtās vēstures ierakstu un sudoku objektu apvienoto masīvu, un pabeigtās vēstures ierakstu un sudoku objektu apvienoto masīvu.		
	400	Tiek atgriezts kļūdas paziņojums “History entries not found!” gadījumā, ja netika atrasti vēstures ieraksti.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.4.3 Vēstures ieraksta iegūšana

Ceļš:	GET: /api/v1/history/:historyId			
Apraksts:	Funkcija nodrošina lietotāja vēstures ieraksta iegūšanu pēc tā identifikatora.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	historyId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā vēstures ieraksta identifikators.

	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Pārbaude vai lietotājs, kurš vēlas iegūt norādītā vēstures ieraksta informāciju ir šī ieraksta autors vai lietotājs ir moderatora vai administratora līmenī. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar iegūto vēstures ieraksta objektu.		
	400	Tiek atgriezts kļūdas paziņojums “Only Admin/Moderator or history entry owner can see this information!” gadījumā, ja visu lietotāju vēstures ierakstus mēģina iegūt lietotājs, kurš nav norādītā vēstures ieraksta īpašnieks un kuram nepiemīt moderatora vai administratora līmenis.		
	400	Tiek atgriezts kļūdas paziņojums “History entry not found!” gadījumā, ja netiek atrasts vēstures ieraksts pēc norādītā identifikatora.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.4.4 Vēstures ieraksta pievienošana

Ceļš:	POST: /api/v1/history			
Apraksts:	Funkcija nodrošina lietotāja vēstures ieraksta pievienošanu.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
	sudokuId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā sudoku identifikators.
	atbilde	Rakstzīmju dati	x	Rakstzīmju virkne, kura atbilst sudoku mīklai, kas savienota vienā simbolu virknē pa rindām, sākot no augšas, aizstājot tukšās vietas ar “0”.
	laiks	Skaitliskie dati	x	Skaitlis sekundēs, kas norāda patērēto laiku sudoku risināšanai līdz saglabātajam pieturas punktam.

	izmantoja atbildi	Binārie dati		Patiess vai nepatiess, kas norāda vai lietotājs izmantoja atbildes parādīšanas funkcionalitāti. Pēc noklusējuma tiek iestatīts kā nepatiess.
	atrisināts	Binārie dati		Patiess vai nepatiess, kas norāda vai lietotājs ir pareizi atrisinājis sudoku mīklu. Pēc noklusējuma tiek iestatīts kā nepatiess.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: - Pārbaude vai ievadītās atbildes rakstzīmju virknes garums ir tieši 81.			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar paziņojumu “Successfully added history entry!” un pievienotā vēstures ieraksta objektu.		
	400	Tiek atgriezts kļūdas paziņojums “Answer length must be 81!” gadījumā, ja atbildes rakstzīmju virknes garums nav vienāds ar 81.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.4.5 Lietotāja vēstures statistikas iegūšana

Ceļš:	GET: /api/v1/history/sudoku/statistics			
Apraksts:	Funkcija nodrošina lietotāja vēstures statistikas iegūšanu.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: - Iegūts vēstures ierakstu skaits, kur lietotājs ir atrisinājis sudoku mīklu. - Iegūts vēstures ierakstu skaits, kur lietotājs ir sācis risināt un/vai atrisinājis sudoku mīklu. - Iegūts vēstures ierakstu skaits, kur lietotājs ir izmantojis sudoku atrisinājuma parādīšanas funkcionalitāti. - Iegūts visu datu bāzē eksistējošo sudoku skaits.			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar iepriekš iegūtajiem skaitļiem.		
	400	Tiek atgriezts kļūdas paziņojums “User not found!” gadījumā, ja netika atrasts lietotājs.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.4.6 Lietotāja vēstures ieraksta esamības pārbaude

Ceļš:	GET: /api/v1/history/sudoku/:sudokuId			
Apraksts:	Funkcija nodrošina pārbaudi vai lietotājam eksistē vēstures ieraksts ar norādīto sudoku indentifikatoru.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	sudokuId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā sudoku indentifikators.
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Pārbaude vai datu bāzē eksistē vēstures ieraksts ar norādītā lietotāja indentifikatoru un norādītā sudoku indentifikatoru. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar: <ul style="list-style-type: none"> - true, gadījumā, ja tika atrasts vēstures ieraksts, - false, gadījumā, ja netika atrasts vēstures ieraksts. 		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.4.7 Vēstures ieraksta atjaunināšana

Ceļš:	PATCH: /api/v1/history/:sudokuId			
Apraksts:	Funkcija nodrošina vēstures ieraksta atjaunināšanu pēc norādītā sudoku un lietotāja identifikatoriem vai vēstures ieraksta saglabāšanu, ja šādu identifikatoru savienojums datu bāzē vēl neeksistē.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
	sudokuId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā sudoku indentifikators.
	atbilde	Rakstzīmju dati	x	Rakstzīmju virkne, kura atbilst sudoku mīklai, kas savienota vienā simbolu virknē pa rindām, sākot no augšas, aizstājot tukšās vietas ar “0”.
	laiks	Skaitliskie dati	x	Skaitlis sekundēs, kas norāda patērēto laiku sudoku risināšanai līdz saglabātajam pieturas punktam.

	izmantoja atbildi	Binārie dati	x	Patiess vai nepatiess, kas norāda vai lietotājs izmantoja atbildes parādīšanas funkcionalitāti.
	atrisināts	Binārie dati	x	Patiess vai nepatiess, kas norāda vai lietotājs ir pareizi atrisinājis sudoku mīklu.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Meklē datu bāzē vēstures ierakstu, kas atbilst norādītajiem sudoku un lietotāja identifikatoriem. - Atbilstoši tam vai atrasts, vai neatrasts vēstures ieraksts, notiek sudoku ieraksta atjaunināšana vai izveidošana. - Pārbaude vai ievadītās atbildes rakstzīmju virknes garums ir tieši 81, ja tas vispār tiek ievadīts. - Pārbaude vai ievadītais sudoku jau neeksistē datu bāzē. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar atjaunotā vēstures ieraksta objektu.		
	201	Dati tiek padoti objekta veidā ar paziņojumu “Successfully added history entry!” un pievienotā vēstures ieraksta objektu gadījumā, ja šāds vēstures ieraksts vēl nepastāvēja datu bāzē, un tika izveidots jauns ieraksts.		
	400	Tiek atgriezts kļūdas paziņojums “Answer length must be 81!” gadījumā, ja tiek mēģināts atjaunināt vēstures ierakstu, kura atbildes garums nav vienāds ar 81.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.1.4.8 Vēstures ieraksta dzēšana

Ceļš:	DELETE: /api/v1/history/:historyId			
Apraksts:	Funkcija nodrošina norādītā vēstures ieraksta dzēšanu no sistēmas.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	historyId	Rakstzīmju dati	x	Rakstzīmju virkne, kas datu bāzē strādā kā vēstures ieraksta identifikators.
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	Datu apstrādē tiks veiktas šādas darbības: <ul style="list-style-type: none"> - Pārbaude vai lietotājs, kurš vēlas izdzēst norādīto vēstures ierakstu ir moderatora vai administratora līmenī. 			
Izvaddati:	Statuss	Dati/Skaidrojums		

	200	Dati tiek padoti objekta veidā ar paziņojumu “History entry succesfully deleted!”.
	400	Tiek atgriezts kļūdas paziņojums “History entry not found!” gadījumā, ja netika atrasts vēstures ieraksts ar norādīto identifikatoru.
	400	Tiek atgriezts kļūdas paziņojums “Only Admin or Moderator can delete history entry!” gadījumā, ja vēstures ierakstu mēģina dzēst lietotājs, kuram nepiemīt moderatora vai administratora līmenis.
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.

1.3.1.5 Administratora modulis

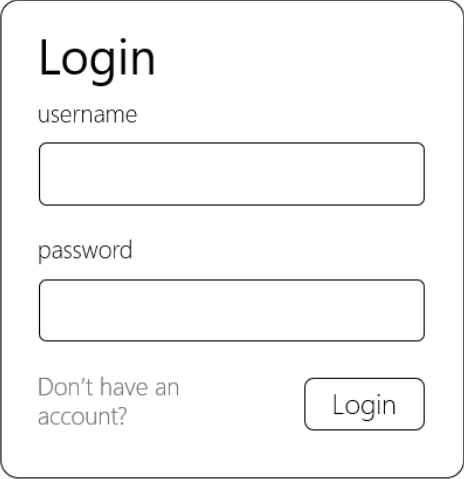
1.3.1.5.1 Administratora paneļa informācijas iegūšana

Ceļš:	GET: /api/v1/admin			
Apraksts:	Funkcija nodrošina administratora panelim nepieciešamās informācijas iegūšanu.			
Ievaddati:	Nosaukums	Datu tips	Obligāts	Skaidrojums
	lietotājs	JSON objekts	x	JSON objekts, kura ietvaros ir visa JWT parakstā iekļautā informācija.
Apstrāde:	<p>Datu apstrādē tiks veiktas šādas darbības:</p> <ul style="list-style-type: none"> - Pārbaude vai lietotājs, kurš vēlas iegūt visu lietotāju vēsturi ir administratora līmenī. - Iegūst visu sistēmā esošo lietotāju informāciju masīvā. - Iegūst visu lietotāju skaitu, kuri reģistrējušies pēdējās 7 dienās. - Iegūst visu sistēmā esošo sudoku skaitu. - Iegūst visu sistēmā atrodamo vēstures ierakstu skaitu, kuri skaitās kā atrisināti. - Iegūst visu sistēmā atrodamo vēstures ierakstu skaitu, kuros atzīmēts, ka lietotājs ir izmantojis atbildes parādīšanas funkcionalitāti. 			
Izvaddati:	Statuss	Dati/Skaidrojums		
	200	Dati tiek padoti objekta veidā ar visu iepriekš norādīto informāciju par sistēmu.		
	400	Tiek atgriezts kļūdas paziņojums “Only admin has access rights!” gadījumā, ja administratora panelim nepieciešamos datus vēlas iegūt lietotājs, kuram nav administratora līmenis.		
	401	Tiek atgriezts kļūdas paziņojums “No token provided or token expired” gadījumā, ja nav norādīts JWT paraksts vai arī tā derīguma termiņš ir beidzies.		

1.3.2 Lietotāja saskarne

1.3.2.1 Pieteikšanās modulis

1.3.2.1.1 Lietotāja pieteikšanās

Ceļš:	/login	Komponente:	Login
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	onLogin	Funkcija	Funkcija, kas nosūta POST pieprasījumu uz aizmugursistēmu un pievieno atgriezto JWT parakstu pārlūkprogrammas localStorage. Funkcija pieņem divus parametrus: lietotājvārdu un paroli.
Apraksts:	Pieteikšanās komponentē lietotājiem ir iespējams pieteikties sistēmā, izmantojot savu lietotājvārdu un paroli. Komponente sastāv no diviem ievades laukiem (lietotājvārda un paroles), linka uz reģistrācijas lapu, kā arī pieteikšanās pogas (skat. Attēlu 1.1).		
Izvaddati:	 <p>Attēls 1.1</p>		

1.3.2.2 Reģistrācijas modulis

1.3.2.2.1 Lietotāja reģistrācija




Ceļš:	/register	Komponente:	Register
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	onRegister	Funkcija	Funkcija, kas nosūta POST pieprasījumu uz aizmugursistēmu un novirza lietotāju uz pieteikšanās skatu.
Apraksts:	Reģistrācijas komponentē lietotājiem ir iespējams reģistrēties, izveidojot savu lietotājvārdu, paroli un norādot derīgu e-pastu. Komponente sastāv no trīs ievades laukiem (lietotājvārda, e-pasta un paroles), linka uz pieteikšanās lapu, kā arī reģistrēšanās pogas (skat. Attēlu 1.2).		

Izvaddati:	<div style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: auto;"> <h2 style="text-align: center;">Register</h2> <p>username</p> <input style="width: 100%;" type="text"/> <p>email</p> <input style="width: 100%;" type="text"/> <p>password</p> <input style="width: 100%;" type="password"/> <p style="text-align: center;"> <small>Already have an account?</small> <input style="margin-left: 20px;" type="button" value="Login"/> </p> </div> <p style="text-align: center;">Attēls 1.2</p>
-------------------	--

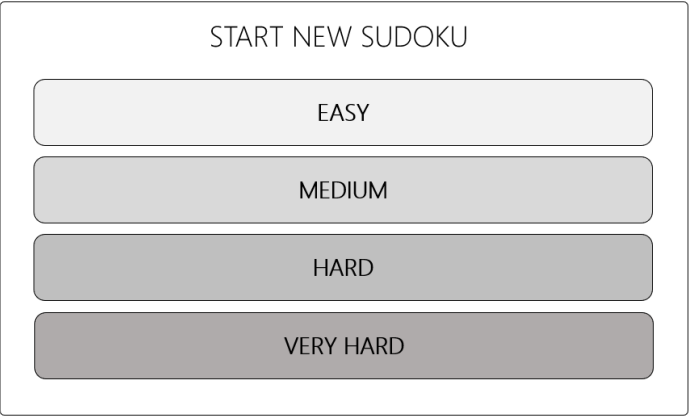
1.3.2.3 Pamata modulis

1.3.2.3.1 Galvene

Ceļš:	-	Komponente:	Header
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	onLogout	Funkcija	Funkcija, kas no pārlūkprogrammas localStorage izdzēš JWT atslēgu, tādējādi atvienojot lietotāju no sistēmas.
Apraksts:	<p>Galvenes komponentē lietotājiem ir iespējams:</p> <ul style="list-style-type: none"> - Pieteikties sistēmai, ja vēl tas nav izdarīts; - Atvērt izvēlni ar pieejamajām novirzīšanas iespējām; <p>Komponente sastāv no vietnes nosaukuma un :</p> <ul style="list-style-type: none"> - Pogas, kas novirza uz pieteikšanās lapu, ja lietotājs nav pieteicies sistēmā. (skat. Attēlu 1.3) - Lietotājvārda un izvēlnes atvēršanas pogas, ja lietotājs ir pieteicies sistēmā. (skat. Attēlu 1.4) - Lietotājvārda un izvēlnes, ja lietotājs ir pieteicies sistēmā un uzspiedis uz izvēlnes atvēršanas pogas, kurā sastopamas novirzīšanas iespējas un atteikšanās pogas. (skat. Attēlu 1.5) 		

Izvaddati:	 <p>Attēls 1.3</p>
	 <p>Attēls 1.4</p>
	 <p>Attēls 1.5</p>

1.3.2.3.2 Grūtības pakāpes izvēlne

Ceļš:	/	Komponente:	DifficultyList
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	onDifficulty-Select	Funkcija	Funkcija, kas nosūta pieprasījumu uz aizmugursistēmu, pieprasot vienu sudoku mīklu, kas izvēlēta atbilstoši izvēlētajai grūtības pakāpei un novirza lietotāju uz sudoku mīklas risināšanas skatu. Funkcija pieņem grūtības pakāpi, kā vienīgo parametru.
Apraksts:	Grūtības pakāpes izvēlnes komponentē reģistrētiem un neregistrētiem lietotājiem ir iespējams izvēlēties vienu no pieejamajām grūtības pakāpēm un saņemt attiecīgi vienu, pēc nejaušības principa izvēlētu, sudoku mīklu. Komponente sastāv no virsraksta un četrām grūtības pakāpes pogām (skat. Attēlu 1.6).		
Izvaddati:	 <p>Attēls 1.6</p>		

1.3.2.4 Sudoku modulis

1.3.2.4.1 Sudoku ietvars

Ceļš:	/sudoku	Komponente:	SudokuGrid
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	sudoku	Rakstzīmju dati	Rakstzīmju virkne 81 simbola garumā, kura atbilst sudoku mīklai, kas savienota vienā simbolu virknē pa rindām, sākot no augšas, aizstājot tukšās vietas ar "0".
	historyEntry	JSON objekts	Nav obligāts. Objekts satur informāciju par lietotāja iepriekš veikto risinājumu un tam patērēto laiku.
Apraksts:	Komponente sastāv no ietvara, kurā parādīsies sudoku laukuma komponente.		
Izvaddati:	Sudoku ietvars bez vizuāla noformējuma.		

1.3.2.4.2 Sudoku laukums

Ceļš:	/sudoku	Komponente:	SudokuGrid
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	sudoku	Rakstzīmju dati	Rakstzīmju virkne 81 simbola garumā, kura atbilst sudoku mīklai, kas savienota vienā simbolu virknē pa rindām, sākot no augšas, aizstājot tukšās vietas ar "0".
	historyEntry	JSON objekts	Nav obligāts. Objekts satur informāciju par lietotāja iepriekš veikto risinājumu un tam patērēto laiku.
	onChangeSave	Funkcija	Funkcija, kas nosūta pieprasījumu uz aizmugursistēmu (gadījumā, ja lietotājs ir pieteicies sistēmā), pieprasot atjaunināt vēstures ierakstu ar jaunāko lietotāja risinājumu. Funkcija pieņem divus parametrus: sudoku identifikatoru un JSON objektu, kas ietver lietotāja jaunāko risinājumu un tam patērēto laiku.

Apraksts: Sudoku laukuma komponentē reģistrētiem un neregistrētiem lietotājiem ir iespējams risināt sudoku mīklu, ievadot lauciņus 9x9 laukumā.

Reģistrētiem lietotājiem pie katra skaitļa ievades tiek veikta vēstures ieraksta atjaunināšana (ja tāds vēl neeksistē, tad tas tiek izveidots), kā arī veicot atrisinājuma pārbaudi ar pogu “Check”, tiek izsaukta atjaunināšana, tikai papildus atbildei un laikam tiek pievienot, ka sudoku ir atrisināts, ja pārbaude bija veiksmīga, kā arī nospiežot atbildes priekšā parādīšanas pogu “Solve”, vēstures ieraksta atjaunināšanas objektam tiek pievienots ieraksts, ka sudoku ir pabeigts, un ka tā atrisināšanai tika izmantota šī atrisinājuma parādīšanas funkcionalitāte.

Neregistrētiem lietotājiem šīs vēstures ieraksta atjaunināšanas darbības nenotiek.

Komponente sastāv no sudoku laukuma, kurā ir 9x9 režģis ar ievades laukiem skaitļu ievadīšanai, un atslēgtiem ievades laukiem, kur oriģināli atrodas jau dotie skaitļi. Zem sudoku laukuma atrodama sudoku pareizības pārbaudes poga, laika uzskaitītājs un atrisinājuma parādīšanas poga. (skat. Attēlu 1.7)

Izvaddati:

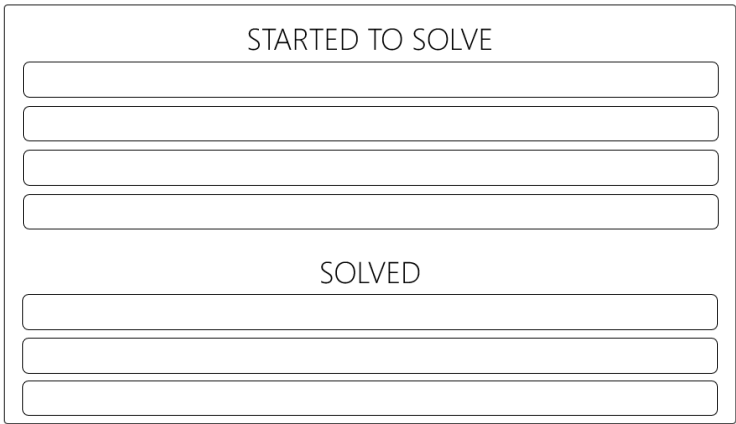
	3			5			4	
		8		1		5		
4	6						1	2
	7		5		2		8	
			6		3			
	4		1		9		3	
2	5						9	8
		1		2		6		
	8			6			2	

00:00

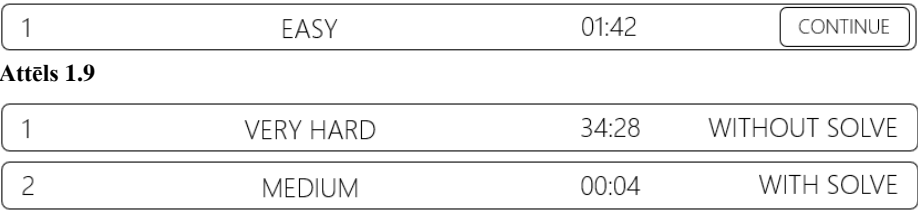
Attēls 1.7

1.3.2.5 Vēstures modulis

1.3.2.5.1 Vēstures saraksts

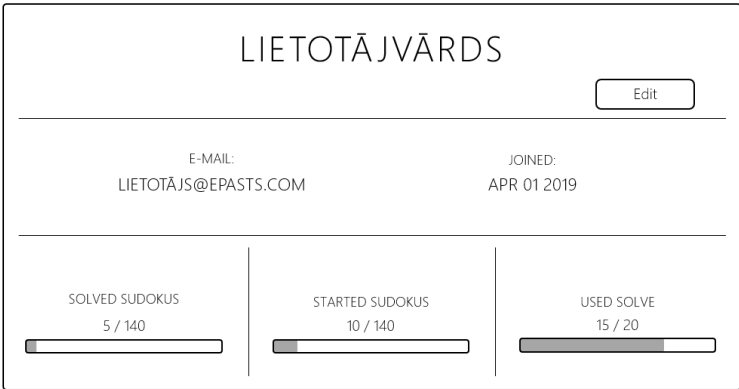
Ceļš:	/history	Komponente:	HistoryList
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	redirectTo-Sudoku	Funkcija	Funkcija, kas novirza lietotāju uz sudoku mīklas risināšanas skatu, norādot, ka tas tiek darīts no vēstures skata, tādējādi sudoku skatā informāciju uztvers savādāk. Funkcija pieņem divus parametrus: esošo sudoku un atrasto vēstures ierakstu.
	dividedHistory	JSON objekts	Objekts, kas sastāv no diviem masīviem: masīva ar nepabeigtajiem sudoku objektiem, un masīva ar atrisinātajiem sudoku objektiem.
Apraksts:	Vēstures saraksta komponentē reģistrētie lietotāji redzēs visus viņu vēstures ierakstus. Komponente sastāv no divām sadaļām: iesākts risināt, kurā parādīsies atbilstošie vēstures ieraksti no dividedHistory objekta un atrisināts, kurā parādīsies atbilstošie vēstures ieraksti no dividedHistory objekta. (skat. Attēlu 1.8)		
Izvaddati:	 <p>Attēls 1.8</p>		

1.3.2.5.2 Vēstures ieraksts

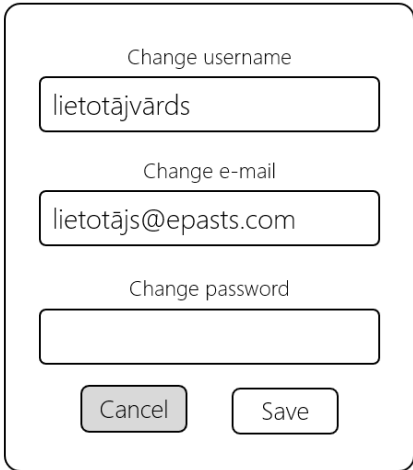
Ceļš:	/history	Komponente:	HistoryCard
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	onContinue	Funkcija	Funkcija, iegūst sudoku no datu bāzes un novirza lietotāju uz sudoku mīklas risināšanas skatu, norādot, ka tas tiek darīts no vēstures skata, tādējādi sudoku skatā informāciju uztvers savādāk. Funkcija pieņem vēstures ieraksta objektu kā vienīgo parametru.
	historyEntry	JSON objekts	Viena vēstures ieraksta objekts ar visu nepieciešamo informāciju, lai varētu norādīt grūtības pakāpi, laiku, vai izmantoja atbildes parādīšanas funkcionalitāti.
	index	Skaitliski dati	Kārtas skaitlis konkrētajam vēstures ierakstam.
Apraksts:	<p>Vēstures ieraksta komponentē reģistrētie lietotāji varēs apskatīt un turpināt iesāktos sudoku, kā arī apskatīt atrisinātos sudoku, un vai tie tika atrisināti lietotāja paša spēkiem vai ar atbildes priekšā parādīšanas funkcionalitāti.</p> <p>Komponentei ir divi iespējamie varianti, kā tā attēlosies:</p> <ul style="list-style-type: none"> - Ja vēstures ieraksta objektā norādīts, ka sudoku nav pabeigts, tad ierakstā parādīsies tā kārtas numurs, grūtības pakāpe, līdz šim aizņemtais laiks un poga, kuru uzspiežot, lietotājam būs iespēja turpināt konkrēto sudoku. (skat. Attēlu 1.9) - Ja vēstures ieraksta objektā norādīts, ka sudoku ir pabeigts, tad ierakstā parādīsies tā kārtas numurs, grūtības pakāpe, atrisināšanai patērētais laiks un ziņojums “WITH SOLVE”, ja lietotājs izmantoja atbildes priekšā parādīšanas funkcionalitāti vai “WITHOUT SOLVE”, ja lietotājs atrisināja sudoku mīklu paša spēkiem. (skat. Attēlu 1.10) 		
Izvaddati:	 <p>Attēls 1.9</p> <p>Attēls 1.10</p>		

1.3.2.6 Lietotāja profila modulis

1.3.2.6.1 Profila skats

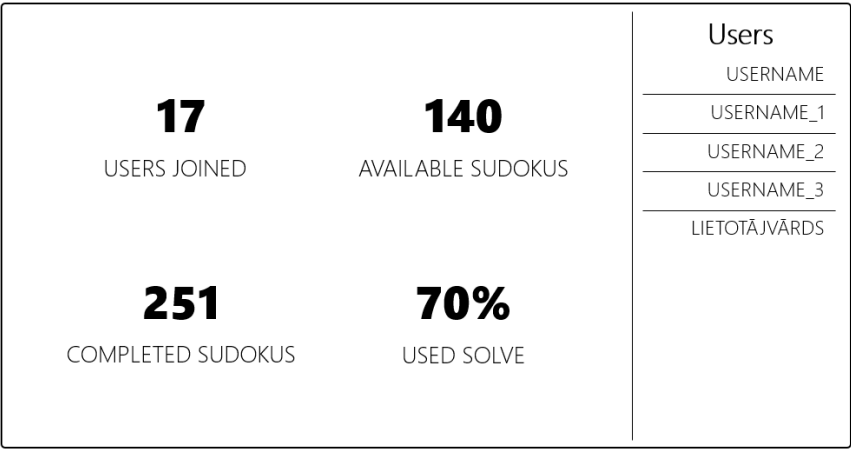
Ceļš:	/profile	Komponente:	ProfileCard
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	onUpdate	Funkcija	Funkcija, kas nosūta patch pieprasījumu uz datu bāzi, kas atjaunina lietotāja informāciju, un ja tiek mainīts lietotājvārds un/vai parole, tad novirza lietotāju uz pieteikšanās lapu. Funkcija pieņem trīs parametrus: lietotāja identifikatoru, objektu ar jauno lietotāja informāciju un mainīgo, kas norāda vai tika izmainīts lietotāja lietotājvārds vai parole, lai zinātu, kad novirzīt uz pieteikšanās skatu.
	user	JSON objekts	Lietotāja informācijas objekts.
	statistics	JSON objekts	Nepieciešamās statistikas objekts, kurā ietilpst izpildīto sudoku skaits, iesākto sudoku skaits, visu sudoku skaits, ar atbildes parādīšanas funkciju atrisināto sudoku skaits.
Apraksts:	Lietotāja profila komponentā lietotājam ir iespējams apskatīt informāciju par sevi (lietotājvārdu, e-pastu, datumu, kurā pievienojies sistēmai), to labot un redzēt nelielu statistiku par lietotāja: atrisināto sudoku skaitu pret visu sudoku skaitu, iesākto sudoku skaitu pret visu sudoku skaitu un atbildes parādīšanas funkcionalitātes izmantošanas reižu skaitu pret atrisināto sudoku skaitu. (skat. Attēlu 1.11)		
Izvaddati:	 <p>The screenshot shows a user profile interface. At the top, the name 'LIETOTĀJVĀRDS' is displayed with an 'Edit' button. Below this, the email 'LIETOTĀJS@EPASTS.COM' and the date 'JOINED: APR 01 2019' are shown. At the bottom, there are three progress bars: 'SOLVED SUDOKUS' (5 / 140), 'STARTED SUDOKUS' (10 / 140), and 'USED SOLVE' (15 / 20).</p>		
	Attēls 1.11		

1.3.2.6.2 Lietotāja rediģēšana


Ceļš:	/profile	Komponente:	UserEdit
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	onUserUpdate	Funkcija	Funkcija, kas aizver rediģēšanas logu un izsauc onUpdate funkciju lietotāja profila komponentē.
	user	JSON objekts	Lietotāja informācijas objekts.
	onClose	Funkcija	Funkcija, kas nodrošina rediģēšanas loga aizvēršanu gadījumā, ja tiek nospiesta atcelšanas poga.
Apraksts:	<p>Lietotāja informācijas rediģēšanas komponentē lietotājam ir iespēja rediģēt savu profila informāciju un to saglabāt.</p> <p>Komponentē redzami 3 ievades lauki, kur 2 no tiem ir jau aizpildīti ar esošu lietotājevārdu un e-pastu (tie joprojām ir rediģējami), atcelšanas pogu un saglabāšanas pogu. (skat. Attēlu 1.12)</p>		
Izvaddati:	 <p>Attēls 1.12</p>		

1.3.2.7 Administratora modulis

1.3.2.7.1 Administratora panelis

Ceļš:	/admin	Komponente:	AdminDashboard
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	onDeleteUser	Funkcija	Funkcija, kas izdzēš norādīto lietotāju un viņa vēsturi, un atjaunina datus adminDashboardData objektā.
	adminDashboardData	JSON objekts	Administratora panelim nepieciešamās informācijas objekts, kurš sastāv no: pēdējo 7 dienu laikā sistēmai pievienojušos lietotāju skaita, visu sistēmā pieejamo sudoku skaita, skaitu, cik reizes ir atrisināti sudoku, starp visiem lietotājiem un atbildes parādīšanas funkcionalitātes izmantošanas skaitu starp visiem lietotājiem.
Apraksts:	Administratora paneļa komponentē, lietotājam, kurš ir administratora līmenī, saskaras ar nelielu vietnes statistiku un visiem vietnes reģistrētajiem lietotājiem. Statistika iekļauj pēdējas 7 dienās reģistrējušos lietotāju skaitu, visu sistēmā pieejamo sudoku skaitu, skaitu, cik reizes ir atrisināti sudoku, starp visiem lietotājiem un atbildes parādīšanas funkcionalitātes izmantošanas procentu pret atrisināto sudoku skaitu, starp visiem lietotājiem. Lietotāju sarakstā ir redzami visi lietotāji sistēmā, kas sakārtoti pēc principa – jaunākie lietotāji augšpusē. Uz lietotājiem ir iespējams uzspiest, kas atver lietotāja informācijas komponenti. (skat. Attēlu 1.13)		
Izvaddati:	 <p>Attēls 1.13</p>		

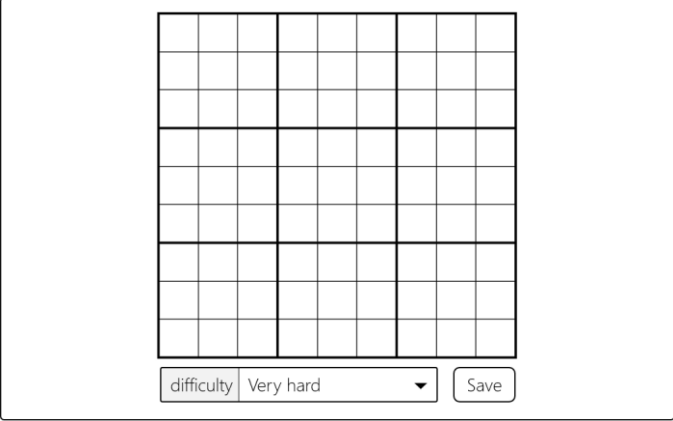
1.3.2.7.2 Lietotāja informācija

Ceļš:	/admin	Komponente:	UserCard
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	handleDelete	Funkcija	Funkcija, izsauc onDeleteUser administratora paneļa komponentē, kas izdzēš norādīto lietotāju un viņa vēsturi, un atjaunina datus adminDashboardData objektā un aizver atvērto lietotāja informācijas kartīti.
	onClose	Funkcija	Funkcija, kas nodrošina lietotāja informācijas kartītes aizvēršanu.
	user	JSON objekts	JSON objekts, kurā sastopama informācija par lietotāju, kurš tiek norādīts, atverot komponenti.
Apraksts:	<p>Lietotāja informācijas komponentē, lietotājam, kam ir administratora piekļuves tiesības, var apskatīt informāciju par izvēlēto lietotāju un to dzēst no sistēmas ar visu tā vēsturi.</p> <p>Komponentē sastopams lietotājvārds, lietotāja e-pasts, datums, kad lietotājs pievienojies sistēmai un divas pogas: aizvērt kartīti un dzēst lietotāju. (skat. Attēlu 1.14)</p>		
Izvaddati:	 <p>Attēls 1.14</p>		

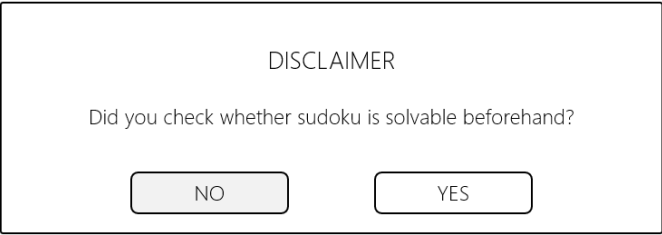
1.3.2.8 Sudoku pievienošanas modulis

1.3.2.8.1 Sudoku pievienošanas skats

Ceļš:	/add	Komponente:	SudokuAdd
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	onSave	Funkcija	Funkcija, kas nodrošina ievadītā sudoku pievienošanu datu bāzei.

Apraksts:	<p>Sudoku pievienošanas komponentē, lietotājiem, kam piešķirtas administratora tiesības, ir iespējams pievienot sidtēmai sudoku mīklu.</p> <p>Komponente sastāv no sudoku laukuma ar skaitļu ievades iespējām, grūtības pakāpes izvēlnes un no pogas, kas pārbauda vai ievadītie skaitļi nav atkārtoti sastopami pa līnijām, kolonnām vai 3x3 laukiem un atver SudokuAddApproval komponenti pirms notiek pievienošana datu bāzei. (skat. Attēlu 1.15)</p>
Izvad dati:	 <p>Attēls 1.15</p>

1.3.2.8.2 Pievienošanas apstiprinājums

Ceļš:	/add	Komponente:	SudokuAddApproval
Ievaddati:	Nosaukums	Datu tips	Skaidrojums
	handleSave	Funkcija	Funkcija, kas veic ievadīto skaitļu pārbaudes, sudoku pārveidošanu rakstzīmju virknē, saglabāšanu un apstiprināšanas kartītes aizvēršanu.
	onClose	Funkcija	Funkcija, kas nodrošina apstiprinājuma kartītes aizvēršanu.
Apraksts:	<p>Sudoku pievienošanas apstiprinājuma komponentē, lietotāji, kas ir administratora līmenī, saskarās ar sudkou pievienošanas datu bāzei apstiprināšanu, lai samazinātu nejauši pievienotu sudoku skaitu.</p> <p>Komponentē redzams paziņojums par to, ka lietotājs uzņemas atbildību par atrisināmu sudoku pievienošanu un divas pogas: nē poga, kas norāda, ka lietotājs nav iepriekš pārliecinājies par sudoku atrisinātību un jā poga, kas norāda, ka lietotājs ir pārliecinājies, ka sudoku ir atrisināms. (skat. Attēlu 1.16)</p>		
Izvad dati:	 <p>Attēls 1.16</p>		

1.4 Nefunkcionālās prasības

1.4.1 Valoda

Tīmekļa vietnes saturam jābūt angļu valodā.

1.4.2 Drošības prasības

Sistēmas datu bāzē paroles nepieciešams glabāt šifrētā veidā. Lietotāju vēstures glabāšanas funkcionalitātes un administrācijas rīku atdalīšanai tiek izmantoti lietotāju līmeņi.

1.4.3 Veiktspējas prasības

Sistēmas atbildes laikam uz pieprasījumiem nedrīkst pārsniegt 1 sekundi.

1.4.4 Uzturamība

Sistēmas izstrādātājs un uzturētājs var viegli modificēt vai papildināt tīmekļa vietni ar jaunu funkcionalitāti, atbilstoši prasību izmaiņām.

Sistēmas izstrādē tiek izmantots versiju kontroles rīks Git.

1.4.5 Pieejamība

Sistēmai jābūt pieejamai 24 stundas 7 dienas nedēļā, izņemot gadījumus, kad nepieciešams veikt sistēmas atjauninājumus. Ja vietnē tiek konstatēta kļūme un sistēmas kvalitatīva darbība tiek apdraudēta, tad tās pilnvērtīga darbība jānodrošina nedēļas laikā.

1.4.6 Atbalsts tīmekļa pārlūkprogrammām

Sistēmas darbība tiek atbalstīta populārākajās pārlūkprogrammās, kā, piemēram, Google Chrome, Mozilla Firefox, Safari un Microsoft Edge gan darbvirsmās, gan viedierīcēs.

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1 Ievads

2.1.1 Nolūks

Dokumentā tiek parādīts programmatūras prasību specifikācijā aprakstīto funkcionālo prasību un nefunkcionālo prasību realizācijas projektējums.

2.1.2 Darbības sfēra

Tīmekļa lietotne “Sudoku App” paredzēta sudoku mīklu risināšanai ar laika uzskaiti, risinājumu vēstures saglabāšanai un iesākto sudoku iespējamai turpināšanai.

2.1.3 Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standarta LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis prasības”.

Dokuments ir saistīts ar programmatūras prasību specifikāciju un testēšanas dokumentāciju.

2.2 Dekompozīcijas apraksts

2.2.1 Aizmugursistēma

Aizmugursistēma izstrādāta ar šādu struktūru: modeļiem un kontrolieriem. Modeļos tiek veiktas darbības ar datu bāzi - datu bāzes shēmas definēšana, datu iegūšana, saglabāšana, rediģēšana, dzēšana un citu pieprasījumu veidošana. Kontrolieros tiek veikta galvenā sistēmas loģika: apstrādāti dati un lietotāja pieprasījumi datu bāzei, lai tos nosūtītu/saņemtu caur modeļiem.

2.2.1.1 Autorizācijas modulis

Autorizācijas modulī tiek izmantoti:

- sudoku-app-api/controllers/authController – autorizācijas kontrolieris, kas nodrošina lietotāja pieslēgšanos un reģistrēšanos sistēmā.
- sudoku-app-api/models/UserModel – lietotāja modelis, kas nodrošina lietotāja saglabāšanas funkcionalitāti un lietotāja pieteikšanās pārbaudes ar datu bāzes ierakstiem.

2.2.1.2 Lietotāja modulis

Lietotāja modulī tiek izmantoti:

- sudoku-app-api/controllers/userController – lietotāju kontrolieris, kas nodrošina visu lietotāju iegūšanu, viena lietotāja informācijas iegūšanu, lietotāja informācijas rediģēšanu un lietotāja dzēšanu.
- sudoku-app-api/models/UserModel – lietotāja modelis, kas nodrošina lietotāju iegūšanu no datu bāzes, lietotāju rediģēšanu un dzēšanu.

2.2.1.3 Sudoku modulis

Sudoku modulī tiek izmantoti:

- sudoku-app-api/controllers/sudokuController – sudoku kontrolieris, kas nodrošina sudoku pievienošanu, visu sudoku iegūšanu, viena sudoku iegūšanu, visu sudoku pēc grūtības pakāpes iegūšanu, pēc nejaušības principa izvēlēta sudoku iegūšanu norādītajā grūtības pakāpē, pēc nejaušības principa izvēlēta sudoku iegūšanu norādītajā grūtības pakāpē autorizētiem lietotājiem, sudoku saglabāšanu un dzēšanu.
- sudoku-app-api/models/SudokuModel – sudoku modelis, kas nodrošina pie kontroliera minētās funkcionalitātes nepieciešamos datu bāzes pieprasījumus.
- sudoku-app-api/models/HistoryModel – vēstures modelis, kas nodrošina pēc nejaušības principa izvēlēta sudoku iegūšanu norādītajā grūtības pakāpē autorizētiem lietotājiem funkcionalitātē pārbaudi, lai norādītajam lietotājam neatgrieztu jau iesāktu sudoku mīklu.

2.2.1.4 Vēstures modulis

Vēstures modulī tiek izmantoti:

- sudoku-app-api/controllers/historyController – vēstures kontrolieris, kas nodrošina vēstures ieraksta pievienošanu, visu vēstures ierakstu iegūšanu, viena ieraksta iegūšanu, pārbaudi vai lietotājam eksistē vēstures ieraksts ar norādīto sudoku, dalītās vēstures iegūšanu ar atrisinātajiem un iesāktajiem sudoku, lietotāja vēstures ierakstu statistikas iegūšanu, vēstures ieraksta rediģēšanu un dzēšanu no sistēmas.
- sudoku-app-api/models/HistoryModel – vēstures modelis, kas nodrošina pie kontroliera minētās funkcionalitātes nepieciešamos datu bāzes pieprasījumus.

- sudoku-app-api/models/SudokuModel – sudoku modelis, kas nodrošina sudoku iegūšanu dalītās vēstures iegūšanas funkcionalitātē un visu sudoku skaita iegūšanu lietotāja vēstures ierakstu statistikas iegūšanas funkcionalitātē.

2.2.1.5 Administratora modulis

Administratora modulī tiek izmantoti:

- sudoku-app-api/controllers/adminController – administratora kontrolieris, kas nodrošina administratora paneliem nepieciešamās informācijas iegūvi.
- sudoku-app-api/models/HistoryModel – vēstures modelis, kas nodrošina visu atrisināto sudoku skaita iegūšanu un visu atrisināto sudoku, kuros izmantota atbildes priekšā parādīšanas funkcionalitāte, skaita iegūšanu.
- sudoku-app-api/models/SudokuModel – sudoku modelis, kas nodrošina visu sistēmā esošo sudoku skaita iegūšanu.
- sudoku-app-api/models/UserModel – lietotāju modelis, kas nodrošina visu sistēmā esošo lietotāju informācijas un pēdējo 7 dienu laikā reģistrējušos lietotāju skaita iegūšanu.

2.2.2 Lietotāja saskarne

Lietotāja saskarne izstrādāta ar šādu struktūru: komponentēm, konteineriem, redux darbībām (actions) un redux reducētāji (reducers). Komponentēs tiek veidota informācijas parādīšana lietotājam un tā vizuālā noformēšana. Konteineros tiek veikta datu ielāde un atbilstoši tiem atgriez atbilstošo komponenti ar iegūtajiem datiem. Darbības un reducētāji ir daļa no redux, kurā tiek glabāti sistēmas stāvokļi, kas ir pieejami visā sistēmā. Pie darbībām ietilpst aizmugursistēmas pieprasījumu veidošana un datu ieguve, bet reducētājos tiek veiktas stāvokļu izmaiņas un datu pievienošana tiem.

2.2.2.1 Pieteikšanās modulis

Pieteikšanās modulī tiek izmantoti:

- sudoku-react-app/src/components/Login/Login – Pieteikšanās komponente, kurā tiek definēts pieteikšanās kartītes izskats.
- sudoku-react-app/src/containers/LoginContainer – Pieteikšanās konteineris, kurā tiek definēta lietotāja pieteikšanās funkcionalitāte un pieteikšanās komponentes parādīšana vai lietotāja novirzīšana uz pamata skatu, ja pieteikšanās veiksmīga.

- sudoku-react-app/src/redux/actions/userActions – lietotāja redux darbības, kur tiek definēta pieteikšanās pieprasījuma nosūtīšana uz aizmugursistēmu.
- sudoku-react-app/src/redux/reducers/userReducer – lietotāja redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.

2.2.2.2 Reģistrācijas modulis

Reģistrācijas modulī tiek izmantoti:

- sudoku-react-app/src/components/Register/Register – Reģistrācijas komponente, kurā tiek definēts reģistrācijas kartītes izskats.
- sudoku-react-app/src/containers/RegisterContainer – Reģistrācijas konteineris, kurā tiek definēta lietotāja reģistrācijas funkcionalitāte un reģistrācijas komponentes parādīšana vai lietotāja novirzīšana uz pieteikšanās skatu, ja reģistrācija veiksmīga.
- sudoku-react-app/src/redux/actions/userActions – lietotāja redux darbības, kur tiek definēta reģistrācijas pieprasījuma nosūtīšana uz aizmugursistēmu.
- sudoku-react-app/src/redux/reducers/userReducer – lietotāja redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.

2.2.2.3 Pamata modulis

Pamata modulī tiek izmantoti:

- sudoku-react-app/src/components/Header/Header – Galvenes komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteinerā.
- sudoku-react-app/src/components/DifficultyList/DifficultyList – Grūtības pakāpes izvēlnes komponente, kurā tiek definēts tās izskats.
- sudoku-react-app/src/containers/HeaderContainer – Galvenes konteineris, kurā tiek definēta lietotāja atteikšanās funkcionalitāte un galvenes komponentes parādīšana.
- sudoku-react-app/src/containers/DifficultyListContainer – Grūtības pakāpes izvēlnes konteineris, kurā tiek definēta sudoku iegūšana atkarībā no izvēlētajās grūtības pakāpes funkcionalitāte un grūtības pakāpes izvēlnes komponentes parādīšana.
- sudoku-react-app/src/redux/actions/userActions – lietotāja redux darbības, kur tiek definēta lietotāja informācijas iegūšanas pieprasījuma nosūtīšana uz aizmugursistēmu.

- sudoku-react-app/src/redux/actions/sudokuActions – sudoku redux darbības, kur tiek definēta sudoku iegūšana pēc nejaušības principa pieprasījuma nosūtīšana uz aizmugursistēmu
- sudoku-react-app/src/redux/reducers/userReducer – lietotāja redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.
- sudoku-react-app/src/redux/reducers/sudokuReducer – sudoku redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.

2.2.2.4 Sudoku modulis

Sudoku modulī tiek izmantoti:

- sudoku-react-app/src/components/SudokuBody/SudokuBody – Sudoku ietvara komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteīnera.
- sudoku-react-app/src/components/SudokuGrid/SudokuGrid – Sudoku laukuma komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteīnera.
- sudoku-react-app/src/containers/SudokuBodyContainer – Sudoku ietvara konteīneris, kurā tiek definēta sudoku ietvara komponentes parādīšana.
- sudoku-react-app/src/containers/SudokuGridContainer – Sudoku laukuma konteīneris, kurā tiek definēta sudoku atjaunināšanas funkcionalitāte un sudoku laukuma komponentes parādīšana.
- sudoku-react-app/src/redux/actions/sudokuActions – sudoku redux darbības, kur tiek definēta sudoku iegūšanas pieprasījuma nosūtīšana uz aizmugursistēmu.
- sudoku-react-app/src/redux/actions/historyActions – vēstures redux darbības, kur tiek definēta vēstures ieraksta atjaunošanas pieprasījuma nosūtīšana uz aizmugursistēmu
- sudoku-react-app/src/redux/reducers/sudokuReducer – sudoku redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.
- sudoku-react-app/src/redux/reducers/historyReducer – vēstures redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.

2.2.2.5 Vēstures modulis

Vēstures modulī tiek izmantoti:

- `sudoku-react-app/src/components/HistoryList/HistoryList` – Vēstures saraksta komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteinera.
- `sudoku-react-app/src/components/HistoryCard/HistoryCard` – Vēstures ieraksta komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteinera.
- `sudoku-react-app/src/containers/HistoryListContainer` – Vēstures saraksta konteineris, kurā tiek definēta lietotāja dalītās vēstures ierakstu iegūšanas funkcionalitāte un vēstures saraksta komponentes parādīšana.
- `sudoku-react-app/src/containers/HistoryCardContainer` – Vēstures ieraksta konteineris, kurā tiek definēta sudoku turpināšanas funkcionalitāte un vēstures ieraksta komponentes parādīšana.
- `sudoku-react-app/src/redux/actions/sudokuActions` – sudoku redux darbības, kur tiek definēta sudoku iegūšanas pieprasījuma nosūtīšana uz aizmugursistēmu.
- `sudoku-react-app/src/redux/actions/historyActions` – vēstures redux darbības, kur tiek definēta dalītās vēstures iegūšanas pieprasījuma nosūtīšana uz aizmugursistēmu.
- `sudoku-react-app/src/redux/reducers/sudokuReducer` – sudoku redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.
- `sudoku-react-app/src/redux/reducers/historyReducer` – vēstures redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.

2.2.2.6 Lietotāja profila modulis

Lietotāja profila modulī tiek izmantoti:

- `sudoku-react-app/src/components/ProfileCard/ProfileCard` – Profila skata komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteinera.
- `sudoku-react-app/src/components/UserEdit/UserEdit` – Lietotāja rediģēšanas komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteinera.

- sudoku-react-app/src/containers/ProfileCardContainer – Profila skata konteineris, kurā tiek definēta lietotāja informācijas rediģēšanas un statistikas iegūšanas funkcionalitāte, un profila skata komponentes parādīšana.
- sudoku-react-app/src/redux/actions/userActions – lietotāja redux darbības, kur tiek definēta lietotāja informācijas iegūšanas un lietotāja rediģēšanas pieprasījuma nosūtīšana uz aizmugursistēmu.
- sudoku-react-app/src/redux/actions/historyActions – vēstures redux darbības, kur tiek definēta lietotāja vēstures statistikas iegūšanas pieprasījuma nosūtīšana uz aizmugursistēmu
- sudoku-react-app/src/redux/reducers/userReducer – lietotāja redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.
- sudoku-react-app/src/redux/reducers/historyReducer – vēstures redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā

2.2.2.7 Administratora modulis

Administratora modulī tiek izmantoti:

- sudoku-react-app/src/components/AdminDashboard/AdminDashboard – Administratora paneļa komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteinerā.
- sudoku-react-app/src/components/UserCard/UserCard – Lietotāja informācijas komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteinerā.
- sudoku-react-app/src/containers/AdminDashboardContainer – Administratora paneļa konteineris, kurā tiek definēta sistēmas statistikas iegūšanas un lietotāja dzēšanas funkcionalitāte, un administratora paneļa komponentes parādīšana.
- sudoku-react-app/src/redux/actions/adminActions – administratora redux darbības, kur tiek definēta sistēmas statistikas iegūšanas pieprasījuma nosūtīšana uz aizmugursistēmu.
- sudoku-react-app/src/redux/actions/userActions – lietotāja redux darbības, kur tiek definēta lietotāja dzēšanas pieprasījuma nosūtīšana uz aizmugursistēmu
- sudoku-react-app/src/redux/reducers/adminReducer – administratora redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.
- sudoku-react-app/src/redux/reducers/userReducer – lietotāja redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā

2.2.2.8 Sudoku pievienošanas modulis

Sudoku pievienošanas modulī tiek izmantoti:

- `sudoku-react-app/src/components/SudokuAdd/SudokuAdd` – Sudoku pievienošanas komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteinerā.
- `sudoku-react-app/src/components/SudokuAddApproval/SudokuAddApproval` – sudoku pievienošanas apstiprinājuma komponente, kurā tiek definēts tās izskats un informācijas parādīšana, kas iegūta no konteinerā.
- `sudoku-react-app/src/containers/SudokuAddContainer` – Sudoku pievienošanas konteineris, kurā tiek definēta sudoku pievienošanas funkcionalitāte un sudoku pievienošanas komponentes parādīšana.
- `sudoku-react-app/src/redux/actions/sudokuActions` – sudoku redux darbības, kur tiek definēta sudoku pievienošanas pieprasījuma nosūtīšana uz aizmugursistēmu.
- `sudoku-react-app/src/redux/reducers/sudokuReducer` – sudoku redux reducētājs, kur tiek definētas stāvokļa izmaiņas sistēmā.

2.3 Datu bāzes projektējums

Sistēmas datu uzglabāšanai tiek izmantota bez-relāciju datu bāze MongoDB. Tā tika izvēlēta, jo dati tiek uzglabāti BSON (Binary JSON) formātā. Tā kā vietne tiek izstrādāta Javascript valodā, tad JSON formātu ir viegli īstenot programmas kodā. MongoDB izmanto nedaudz atšķirīgāku terminoloģiju kā relāciju veida datu bāzes, kā rezultātā relāciju datu bāžu tabulas MongoDB datu bāzē tiek interpretētas kā kolekcijas, rindas kā dokumenti un kolonnas kā lauki. [4.]

2.3.1 Datu bāzes kolekciju apraksts

2.3.1.1 Lietotāju kolekcija (users)

Lauks	Datu tips	Cita informācija	Apraksts
id	ObjectId	-	Automātiski izveidots lauks, kam tiek piešķirts unikāls objectId
username	String	trimmed, required, unique	Lietotājvārds, ko lietotājs izvēlas reģistrējoties vietnē. Tas noder kā lietotāja identifikators, un nepieciešams lietotāja pieteikšanās gadījumā.

email	String	trimmed, required, unique	E-pasts, ko lietotājs ievada reģistrējoties un var rediģēt savā profilā. Tas nepieciešams papildus lietotāja identifikācijai un veidam, kā sazināties ar lietotāju nepieciešamības gadījumā.
password	String	trimmed, required	Lietotāja izvēlēta parole, kura jāievada pieteikšanās skatā kopā ar lietotājvārdu.
level	String	trimmed, required, enumerated [“user” “moderator”, “admin”], default value: “user”	Lietotāja līmenis, kas nosaka lietotājam piešķirtās piekļuves tiesības konkrētām funkcionalitātēm.
timestamps	Date	-	Automātiski izveido divus laukus: created_at un updated_at, kas attiecīgi norāda, kad šis ieraksts izveidots un kad pēdējo reizi atjaunināts.

2.3.1.2 Sudoku kolekcija (sudokus)

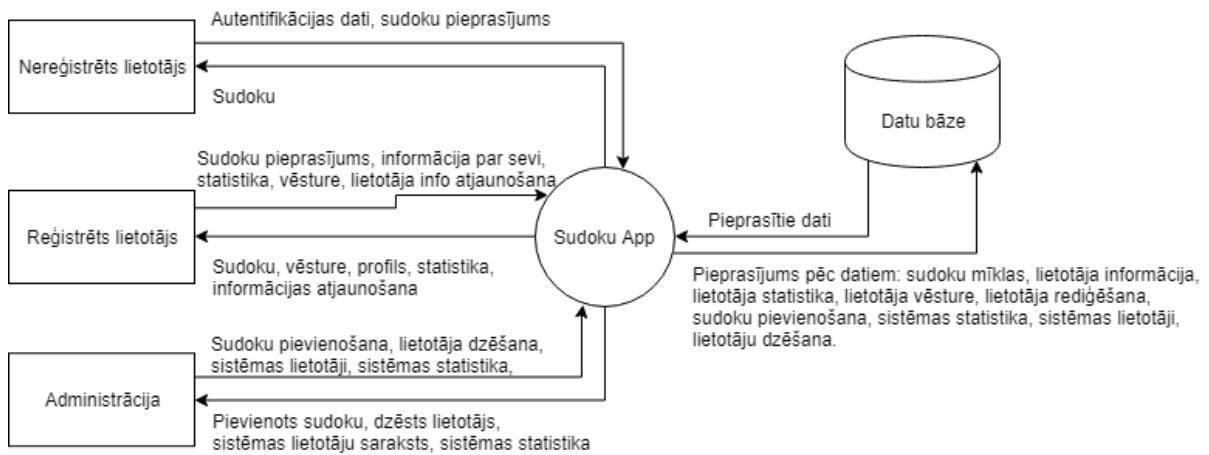
Lauks	Datu tips	Cita informācija	Apraksts
id	ObjectId	-	Automātiski izveidots lauks, kam tiek piešķirts unikāls objectId.
difficulty	String	trimmed, required, enumerated [“easy”, “medium”, “hard”, “very hard”]	Sudoku grūtības pakāpe, kura drīkst būt tikai kā easy, medium, hard vai very hard.
sudoku	String	trimmed, required, unique	Rakstzīmju virkne 81 simbola garumā, kura atbilst sudoku mīklai, kas savienota vienā simbolu virknē pa rindām, sākot no augšas, aizstājot tukšās vietas ar “0”.
timestamps	Date	-	Automātiski izveido divus laukus: created_at un updated_at, kas attiecīgi norāda, kad šis ieraksts izveidots un kad pēdējo reizi atjaunināts.

2.3.1.3 Vēstures kolekcija (histories)

Lauks	Datu tips	Cita informācija	Apraksts
id	ObjectId	-	Automātiski izveidots lauks, kam tiek piešķirts unikāls objectId.
userId	String	trimmed, required	Lietotāja identifikators.
sudokuId	String	trimmed, required	Sudoku identifikators.
answer	String	trimmed, required,	Rakstzīmju virkne 81 simbola garumā, kura atbilst sudoku mīklas risinājumam, kas savienots vienā simbolu virknē pa rindām, sākot no augšas, aizstājot tukšās vietas ar "0".
time	Number	required	Laiks sekundēs, cik pagājis, no sudoku mīklas risinājuma uzsākšanas līdz pēdējai saglabātajai atbildei.
completed	Boolean	required, default: false	Vienība, kas norāda vai sudoku ir pabeigts vai nē.
usedSolve	Boolean	required, default: false	Vienība, kas norāda vai sudoku atrisināšanā tika izmantota atbildes parādīšanas funkcionalitāte.
timestamps	Date	-	Automātiski izveido divus laukus: created_at un updated_at, kas attiecīgi norāda, kad šis ieraksts izveidots un kad pēdējo reizi atjaunināts.

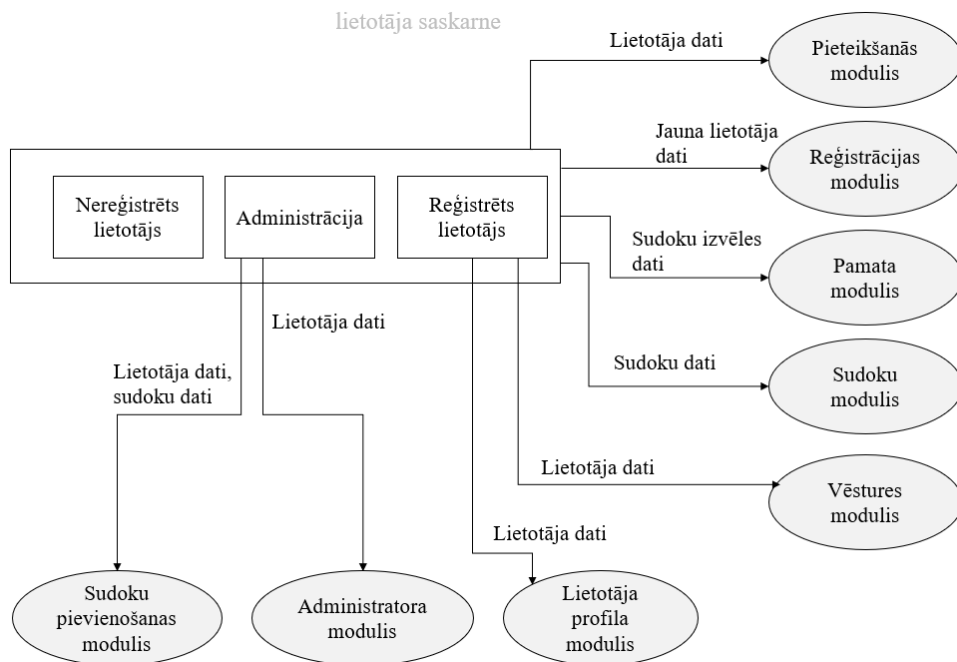
3. FUNKCIJU PROJEKTĒJUMS

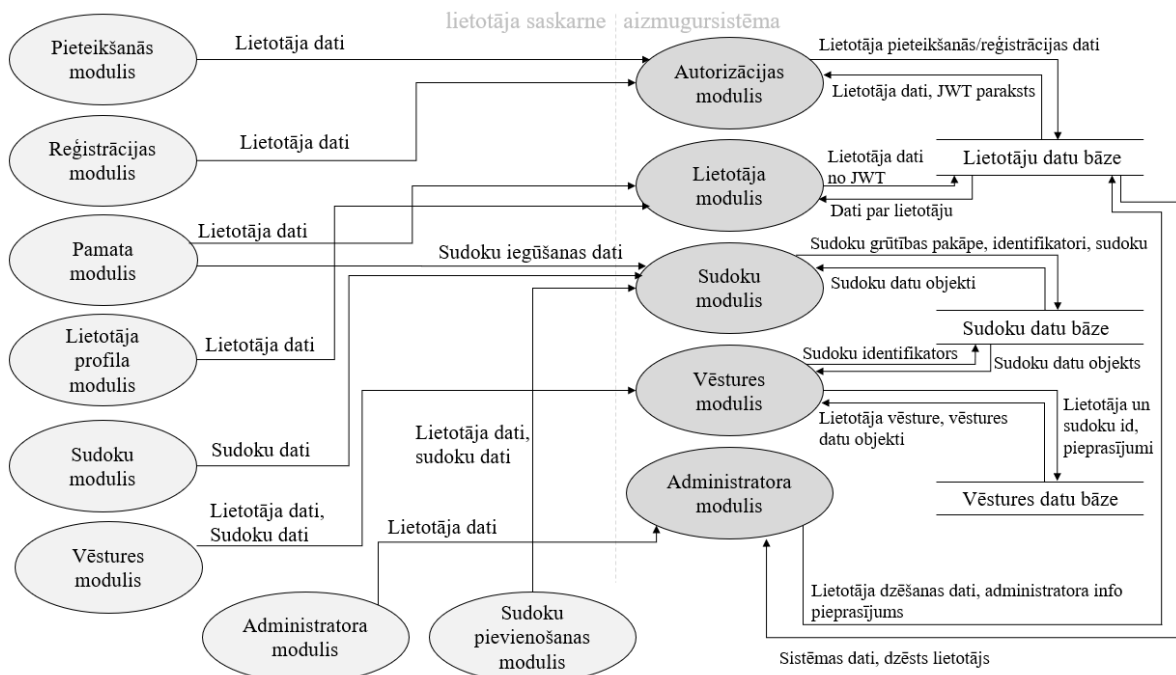
3.1 0. līmeņa datu plūsmas diagramma



3.2 1. līmeņa datu plūsmas diagramma

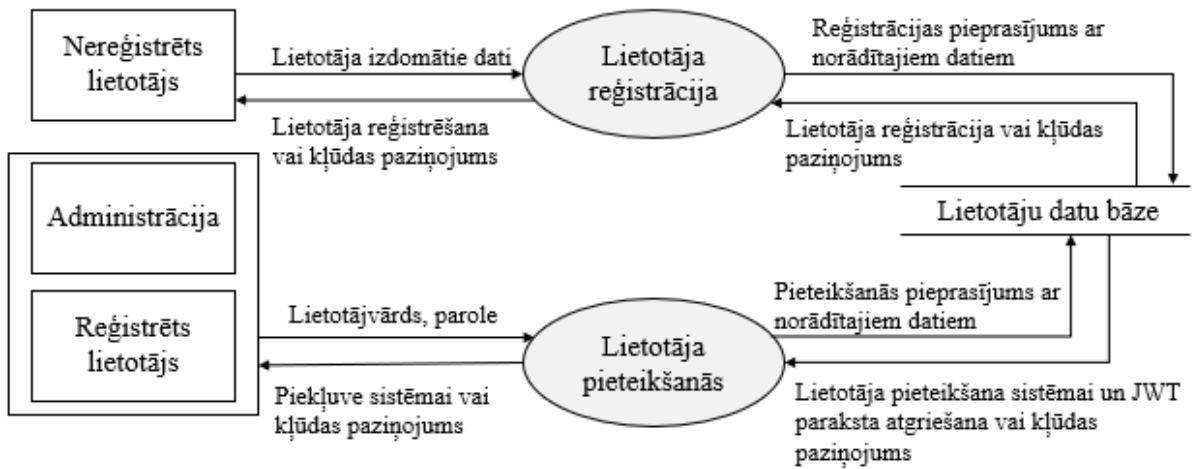
1. Līmeņa datu plūsmu diagramma tiek veidota divās daļās, lai tā būtu pārskatāmāka.



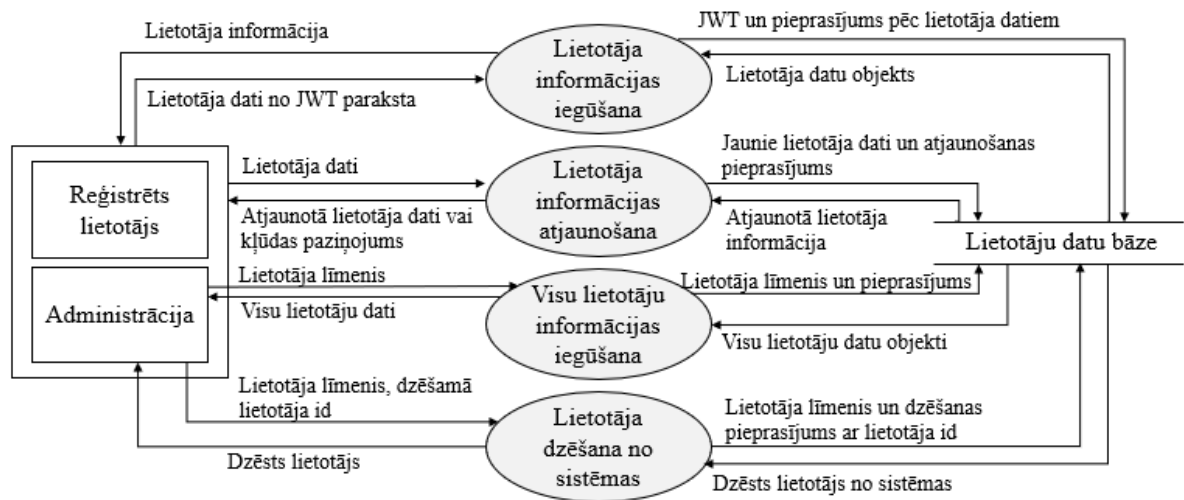


3.3 2. līmeņa datu plūsmas diagramma

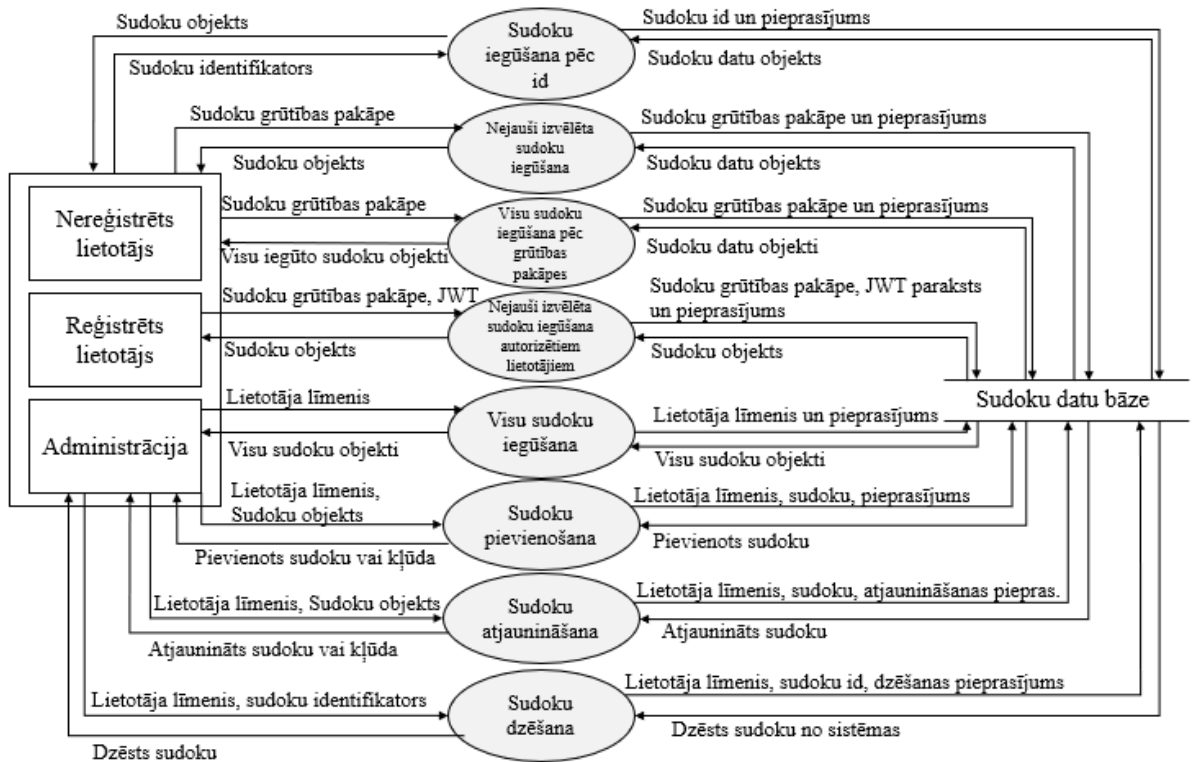
3.3.1 Autorizācijas modulis



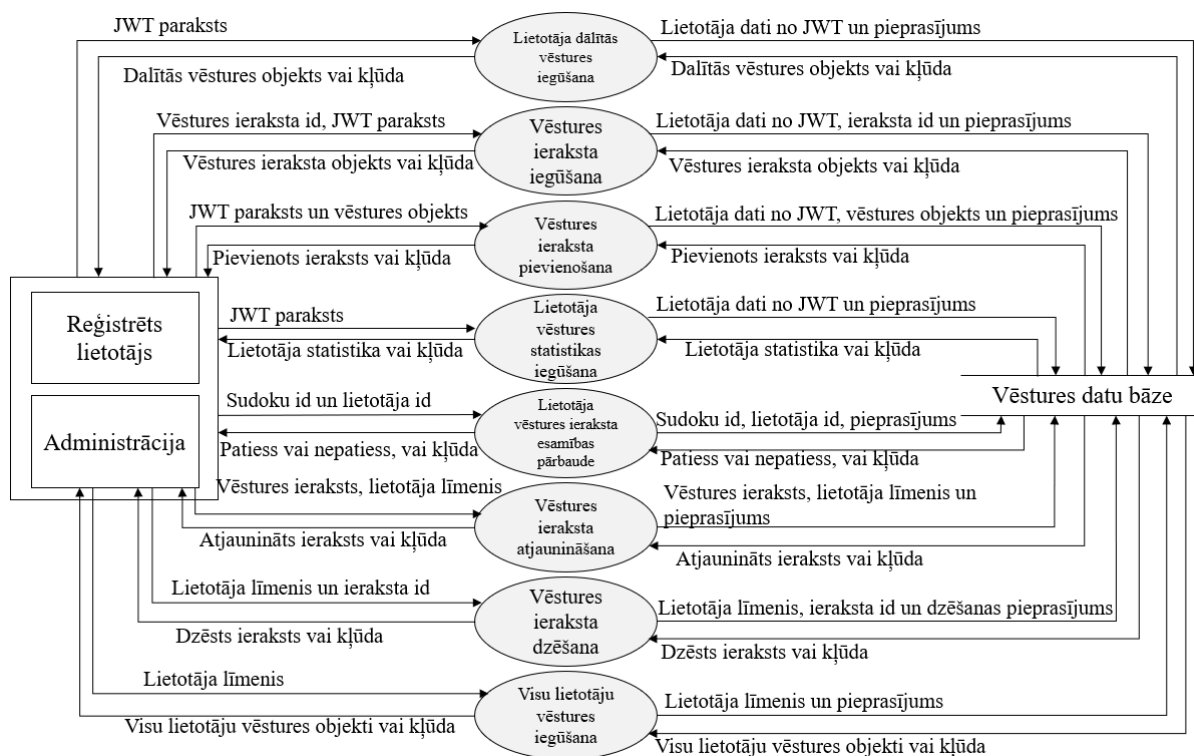
3.3.2 Lietotāja modulis



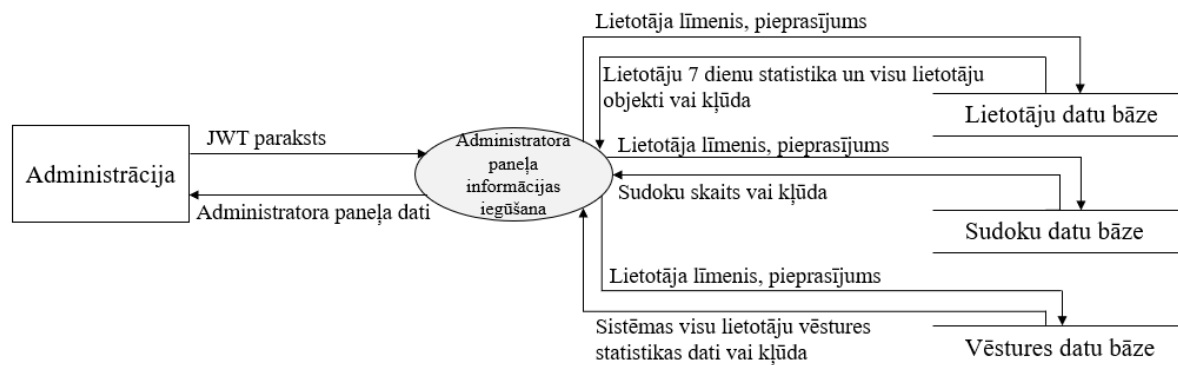
3.3.3 Sudoku modulis



3.3.4 Vēstures modulis



3.3.5 Administratora modulis



4. TESTĒŠANAS DOKUMENTĀCIJA

4.1 Ievads

4.1.1 Apraksts

Dokumentā tiek aprakstīti projekta vienībtestēšanas rezultāti. Testēšana tika veikta ar Jest testēšanas ietvaru, un Enzyme testēšanas rīku React daļu testēšanai, toties aizmugursistēmas API pierpasījumu testēšanai tika izmantots Postman izstrādes rīks, kurā izveidotie testi tika eksportēti kā json fails un sistēmā tiek izsaukti ar Newman bibliotēkas palīdzību.

4.1.2 Saistība ar citiem dokumentiem

Dokuments ir saistīts ar šī projekta programmatūras prasību specifikāciju un programmatūras projektējuma aprakstu.

4.2 Testēšanas žurnāls

4.2.1 Aizmugursistēma

4.2.1.1 Autorizācijas modulis

	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
Reģistrācija	1.	Padots lietotājvārds, e-pasts, parole	Reģistrācijas pieprasījumā tiek norādīts nederīgs e-pasts	API atgriež kļūdas paziņojumu un statusu 400	Izpildīts
	2.	Padots lietotājvārds, e-pasts, parole	Reģistrācijas pieprasījumā tiek norādīts lietotājvārds, kas jau eksistē sistēmā	API atgriež kļūdas paziņojumu un statusu 400	Izpildīts
	3.	Padots lietotājvārds, e-pasts, parole	Reģistrācijas pieprasījumā tiek norādīts e-pasts, kas jau eksistē sistēmā	API atgriež kļūdas paziņojumu un statusu 400	Izpildīts
Pieteikšanās	4.	Padots lietotājvārds, parole	Pieteikšanās pieprasījumā tiek norādīti derīgi lietotāja dati	API atgriež JWT parakstu un statusu 200	Izpildīts
	5.	Padots lietotājvārds, parole	Pieteikšanās pieprasījumā tiek norādīti nederīgi lietotāja dati	API atgriež kļūdas paziņojumu un statusu 400	Izpildīts

4.2.1.2 Lietotāja modulis

	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
Visu lietotāju info	1.	Lietotājs ir pieteicies sistēmā vismaz moderatora līmenī	Visu lietotāju info iegūšanas pieprasījums atgriež visu lietotāju objektus	API atgriež visu lietotāju objektus un statusu 200	Izpildīts
	2.	Lietotājs ir pieteicies sistēmā lietotāja līmenī	Visu lietotāju info iegūšanas pieprasījums neatgriež visu lietotāju objektu	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
Lietotāja info	3.	Lietotājs ir pieteicies sistēmā	Lietotāja info iegūšanas pieprasījums atgriež lietotāja objektu	API atgriež lietotāja objektu un statusu 200	Izpildīts
Lietotāja atjaunošana	4.	Padots lietotāja id, atjaunojamais objekts, lietotājs pieteicies sistēmā	Lietotāja info atjaunošanas pieprasījums atgriež jauno lietotāja objektu	API atgriež jauno lietotāja objektu un statusu 200	Izpildīts
	5.	Padots lietotāja id, atjaunojamais objekts ar e-pastu, lietotājs pieteicies sistēmā	Lietotāja info atjaunošanas pieprasījumā tiek norādīts jau eksistējošs e-pasts	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
	6.	Padots lietotāja id, atjaunojamais objekts ar lietotājvārdu, lietotājs pieteicies sistēmā	Lietotāja info atjaunošanas pieprasījumā tiek norādīts jau eksistējošs lietotājvārds	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
	7.	Padots lietotāja id, atjaunojamais objekts, lietotājs pieteicies sistēmā vismaz moderatora līmenī	Lietotāja info atjaunošanas pieprasījumā tiek norādīts neeksistējošs lietotāja id	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts

Lietotāja dzēšana	8.	Padots lietotāja id, lietotājs pieteicies sistēmā vismaz moderatora līmenī	Lietotāja info dzēšanas pieprasījumā tiek norādīts neeksistējošs lietotāja id	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
	9.	Padots lietotāja id, lietotājs pieteicies sistēmā lietotāja līmenī	Lietotāja info dzēšanas pieprasījums netiek izpildīts	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts

4.2.1.3 Sudoku modulis

	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
Sudoku pievienošana	1.	Padots sudoku objekts un lietotājs ir pieteicies sistēmā lietotāja līmenī	Sudoku netiek pievienots sistēmai, jo lietotājs nav moderatora vai administratora līmenī	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
	2.	Padots sudoku objekts un lietotājs ir pieteicies sistēmā vismaz moderatora līmenī	Sudoku netiek pievienots sistēmai, jo norādīta nederīga grūtības pakāpe	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
	3.	Padots sudoku objekts un lietotājs ir pieteicies sistēmā vismaz moderatora līmenī	Sudoku netiek pievienots sistēmai, jo sudoku garums neatbilst 81 simbolam	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
Visi sudoku	4.	-	Visu sudoku pieprasījums atgriež visus sudoku objektus	API atgriež lietotāja objektu un statusu 200	Izpildīts
Sudoku info	5.	Padots sudoku id	Sudoku info pieprasījums atgriež sudoku objektu	API atgriež jauno sudoku objektu un statusu 200	Izpildīts
	6.	Padots sudoku id	Sudoku info pieprasījums atgriež sudoku objektu, jo padots neeksistējošs sudoku id	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts

Sudoku pēc grūtības pakāpes	7.	Padota grūtības pakāpe	Sudoku pēc grūtības pakāpes pieprasījums atgriež sudoku objektus atbilstošajā grūtības pakāpē	API atgriež visu atbilstošo sudoku objektus un statusu 200	Izpildīts
Sudoku pēc nejausības principa	8.	Padota grūtības pakāpe	Sudoku pēc nejausības pieprasījums atgriež nejausi izvēlētu sudoku objektu atbilstošajā grūtības pakāpē	API atgriež sudoku objektu atbilstošajā grūtības pakāpē un statusu 200	Izpildīts
Sudoku pēc nejausības principa autorizētiem lietotājiem	9.	Padota grūtības pakāpe un lietotājs pieteicies sitēmā	Sudoku pēc nejausības pieprasījums atgriež nejausi izvēlētu sudoku objektu atbilstošajā grūtības pakāpē, kurš vēl nav sākts risināt	API atgriež sudoku objektu atbilstošajā grūtības pakāpē un statusu 200	Izpildīts
Sudoku atjaunināšana	10.	Padots sudoku id, atjaunojamais objekts, lietotājs pieteicies sitēmā vismaz moderatora līmenī	Sudoku info atjaunošanas pieprasījums atgriež jauno sudoku objektu	API atgriež jauno lietotāja objektu un statusu 200	Izpildīts
	11.	Padots sudoku id, atjaunojamais objekts, lietotājs pieteicies sitēmā vismaz moderatora līmenī	Sudoku info atjaunošanas pieprasījumā tiek norādīts neeksistējošs sudoku id	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
	12.	Padots sudoku id, atjaunojamais objekts, lietotājs pieteicies sitēmā lietotāja līmenī	Sudoku info atjaunošanas pieprasījums netiek izpildīts, jo lietotājs nav moderatora vai administratora līmenī	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts

Sudoku dzēšana	13.	Padots sudoku id, lietotājs pieteicies sistēmā vismaz moderatora līmenī	Sudoku dzēšanas pieprasījumā tiek norādīts neeksistējošs sudoku id	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
	14.	Padots sudoku id, lietotājs pieteicies sistēmā lietotāja līmenī	Sudoku dzēšanas pieprasījums netiek izpildīts, jo lietotājs nav moderatora vai administratora līmenī	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts

4.2.1.4 Vēstures modulis

	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
Visi vēstures ieraksti	1.	Lietotājs pieteicies sistēmā vismaz moderatora līmenī	Visu vēstures ierakstu pieprasījums atgriež visus vēstures ierakstu objektus	API atgriež visu vēstures ierakstu objektus un statusu 200	Izpildīts
	2.	Lietotājs pieteicies sistēmā lietotāja līmenī	Visu vēstures ierakstu pieprasījums neatgriež visus vēstures ierakstu objektus, jo lietotājs nav vismaz moderatora līmenī	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
Vēstures ieraksta info	3.	Padots vēstures ieraksta id un lietotājs ir pieteicies sistēmā vismaz moderatora līmenī	Vēstures ieraksta pieprasījums atgriež vēstures ieraksta objektu	API atgriež vēstures ieraksta objektu un statusu 200	Izpildīts
	4.	Padots vēstures ieraksta id un lietotājs ir pieteicies sistēmā lietotāja līmenī	Vēstures ieraksta pieprasījums neatgriež vēstures ieraksta objektu, jo lietotājs nav vismaz moderatora līmenī	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts

Vēstures ieraksta eksistence	5.	Padots sudoku id un lietotājs ir pieteicies sistēmā	Vēstures ieraksta eksistences pieprasījums atgriež apstiprinājumu, jo norādīts sudoku id, kuru lietotājs ir sācis risināt	API atgriež true un statusu 200	Izpildīts
	6.	Padots sudoku id un lietotājs ir pieteicies sistēmā	Vēstures ieraksta eksistences pieprasījums atgriež noraidījumu, jo norādīts sudoku id, kuru lietotājs nav sācis risināt	API atgriež false un statusu 200	Izpildīts
Sadalītā vēsture	7.	Lietotājs pieteicies sistēmā	Sadalītās vēstures pieprasījums atgriež vēstures ierakstu objektus sadalītus izpildītajos un iesāktajos	API atgriež sadalītās vēstures ierakstu objektu un statusu 200	Izpildīts
Vēstures statistika	8.	Lietotājs pieteicies sistēmā	Vēstures statistikas pieprasījums atgriež vēstures statistikas objektu	API atgriež vēstures statistikas objektu un statusu 200	Izpildīts
Vēstures ieraksta atjaunināšana	9.	Padots vēstures ieraksta id un lietotājs pieteicies sistēmā	Vēstures ieraksta atjaunošanas pieprasījums atgriež jauno vēstures ieraksta objektu	API atgriež jauno vēstures ieraksta objektu un statusu 200	Izpildīts
Vēstures ieraksta dzēšana	10.	Padots vēstures ieraksta id un lietotājs pieteicies sistēmā vismaz moderatora līmenī	Vēstures ieraksta dzēšanas pieprasījumā tiek norādīts neeksistējošs vēstures ieraksta id	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
	11.	Padots vēstures ieraksta id un lietotājs pieteicies sistēmā lietotāja līmenī	Vēstures ieraksta dzēšanas pieprasījums netiek izpildīts, jo lietotājs nav vismaz moderatora līmenī	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts

4.2.1.5 Administratora modulis

	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
Administratora paneļa dati	1.	Lietotājs pieteicies sistēmā lietotāja līmenī	Administratora paneļa datu pieprasījums netiek izpildīts, jo lietotājs nav vismaz moderatora līmenī	API atgriež kļūdas paziņojumu un statusu 500	Izpildīts
	2.	Lietotājs pieteicies sistēmā vismaz moderatora līmenī	Administratora paneļa datu pieprasījums atgriež datu objektu	API atgriež administratora paneļa datu objektu un statusu 200	Izpildīts

4.2.2 Lietotāja saskarne

4.2.2.1 Pieteikšanās modulis

Fails	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
Login	1.	-	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	2.	Padots mainīgais errorMessage ar kļūdas ziņojumu	Komponente sniedz sagaidāmo html struktūru ar iekļautu kļūdas paziņojumu	Kļūdas paziņojums tiek attēlots	Izpildīts
	3.	Padots mainīgais errorMessage, kurš ir null	Komponente sniedz sagaidāmo html struktūru bez kļūdas paziņojuma	Kļūdas paziņojums netiek attēlots	Izpildīts
	4.	-	Komponentē izmainās stāvoklis, kad tiek izsaukts onChange()	Stāvoklis izmainās	Izpildīts
	5.	Padota funkcija onLogin()	Komponentē izsauc onLogin() funkciju 1 reizi, ja tiek nospiests uz pieteikšanās pogas	Tiek izsaukta funkcija onLogin() 1 reizi	Izpildīts
	6.	Padota funkcija onRedirect()	Komponentē izsauc onRedirect() funkciju 1 reizi, ja tiek nospiests div elements, kas ved uz reģistrēšanās skatu	Tiek izsaukta funkcija onRedirect()	Izpildīts

4.2.2.2 Reģistrācijas modulis

Fails	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
Register	1.	-	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	2.	Padots mainīgais errorMessage ar kļūdas ziņojumu	Komponente sniedz sagaidāmo html struktūru ar iekļautu kļūdas paziņojumu	Kļūdas paziņojums tiek attēlots	Izpildīts
	3.	Padots mainīgais errorMessage, kurš ir null	Komponente sniedz sagaidāmo html struktūru bez kļūdas paziņojuma	Kļūdas paziņojums netiek attēlots	Izpildīts
	4.	-	Komponentē izmainās stāvoklis, kad tiek izsaukts onChange()	Stāvoklis izmainās	Izpildīts
	5.	Padota funkcija onRegister()	Komponentē izsauc onRegister() funkciju 1 reizi, ja tiek nospiests uz reģistrēšanās pogas	Tiek izsaukta funkcija onRegister() 1 reizi	Izpildīts
	6.	Padota funkcija onRedirect()	Komponentē izsauc onRedirect() funkciju 1 reizi, ja tiek nospiests div elements, kas ved uz pieteikšanās skatu	Tiek izsaukta funkcija onRedirect()	Izpildīts

4.2.2.3 Pamata modulis

Fails	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
Header	1.	-	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	2.	Lietotājs ir pieteicies sistēmā	Komponente sniedz sagaidāmo html struktūru, kad lietotājs ir pieteicies	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	3.	-	Komponentē izmainās stāvoklis, kad tiek izsaukts doRedirectToLogin()	Stāvoklis izmainās	Izpildīts

	4.	Padota funkcija onLogout()	Komponentē izmainās stāvoklis, kad tiek izsaukts handleLogout()	Stāvoklis izmainās	Izpildīts
	5.	-	Komponentē izmainās stāvoklis, kad tiek izsaukts toggleDropdown()	Stāvoklis izmainās	Izpildīts
	6.	Lietotājs, kurš ir lietotāja līmenī, ir pieteicies sistēmā	Komponente sniedz sagaidāmo html struktūru bez administratora opcijām izvēlnē	Administrators opcijas izvēlnē neparādās	Izpildīts
	7.	Lietotājs, kurš ir administratora līmenī, ir pieteicies sistēmā	Komponente sniedz sagaidāmo html struktūru ar administratora opcijām izvēlnē	Administrators opcijas izvēlnē parādās	Izpildīts
DifficultyList	8.	-	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	9.	Padota funkcija onDifficultySelect()	Komponentē izsauc onDifficultySelect() funkciju 4 reizes, ja tiek nospiests uz visām pogām	Tiek izsaukta funkcija onDifficultySelect() 4 reizes	Izpildīts
HeaderContainer	10.	Padota funkcija getUserSelf() un lietotājs ir pieteicies sistēmā.	Konteinerī tiek izsaukta funkcija getUserSelf() vienu reizi	Tiek izsaukta funkcija vienu reizi	Izpildīts
	11.	Padota funkcija logout()	Konteinerī tiek vienreiz izsaukta funkcija logout(), kad izsauc onLogout() funkciju	Logout() funkcija tiek izsaukta vienu reizi	Izpildīts
DifficultyListContainer	12.	-	Konteineris sniedz DifficultyList komponenti	Konteineris sniedz DifficultyList komponenti	Izpildīts
	13.	Padod sudoku objektu un funkcijas history.push() un getRandomized-SudokuByDifficulty()	Konteinerī tiek izsauktas funkcijas sudoku nejaušai atgriešanai un history.push() tiek izsaukts ar sudoku objektu	Tiek izsauktas funkcijas	Izpildīts
	14.	Padod sudoku objektu un funkcijas history.push() un getRandomized-SudokuByDifficulty()	Konteinerī tiek izsaukta funkcija sudoku nejaušai atgriešanai, bet history.push() netiek izsaukts	Tiek izsaukta nejauša sudoku atgriešanas funkcija, bet history.push() netiek izsaukts	Izpildīts

	15.	Padod sudokuAuth objektu un funkcijas history.push() un getAthorized-RandomizedSudoku-ByDifficulty(), un lietotājs ir pieteicies sistēmā	Konteinerī tiek izsauktas funkcijas sudoku nejaušai atgriešanai un history.push() tiek izsaukts ar sudoku objektu	Tiek izsauktas funkcijas	Izpildīts
	16.	Padod sudokuAuth objektu un funkcijas history.push() un getAthorized-RandomizedSudoku-ByDifficulty(), un lietotājs ir pieteicies sistēmā	Konteinerī tiek izsauktas funkcijas sudoku nejaušai atgriešanai un history.push() tiek izsaukts ar kļūdas ziņojumu	Tiek izsaukta nejauša sudoku atgriešanas funkcija un history.push() ar kļūdas ziņojumu	Izpildīts
	17.	Padod sudokuAuth objektu un funkcijas history.push() un getAthorized-RandomizedSudoku-ByDifficulty(), un lietotājs ir pieteicies sistēmā	Konteinerī tiek izsauktas funkcijas sudoku nejaušai atgriešanai, bet history.push() netiek izsaukts	Tiek izsaukta nejauša sudoku atgriešanas funkcija, bet history.push() netiek izsaukts	Izpildīts

4.2.2.4 Sudoku modulis

Faili	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
SudokuBody	1.	-	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	2.	Padots historyEntry un sudoku objekti	Komponente sniedz sagaidāmo html struktūru, kad tiek padots vēstures un sudoku objekts	Komponente sniedz sagaidāmo html struktūru	Izpildīts
SudokuGrid	3.	Padots sudoku objekts	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	4.	Padots sudoku objekts un funkcija onChangeSave()	Komponentē tiek atrisināta sudoku mīkla, ja tiek nospriests onSolve() un netiek saglabāta vēsture	Sudoku mīkla tiek atrisināta un netiek saglabāts vēstures ieraksts	Izpildīts

5.	Padots sudoku objekts un funkcija onChangeSave() un lietotājs ir pieteicies sistēmā	Komponentē tiek atrisināta sudoku mīkla, ja tiek nospiests onSolve() un tiek saglabāts vēstures ieraksts	Sudoku mīkla tiek atrisināta un tiek saglabāts vēstures ieraksts	Izpildīts
6.	Padots sudoku objekts un funkcija onChangeSave()	Komponentē tiek parādīts, ka sudoku mīkla ir pareiza, ja tiek nospiests onCheck() un netiek saglabāta vēsture	Tiek parādīts, ka sudoku mīkla ir pareiza un netiek saglabāts vēstures ieraksts	Izpildīts
7.	Padots sudoku objekts un funkcija onChangeSave() un lietotājs ir pieteicies sistēmā	Komponentē tiek parādīts, ka sudoku mīkla ir pareiza, ja tiek nospiests onCheck() un tiek saglabāta vēsture	Tiek parādīts, ka sudoku mīkla ir pareiza un tiek saglabāts vēstures ieraksts	Izpildīts
8.	Padots sudoku objekts un funkcija onChangeSave() un lietotājs ir pieteicies sistēmā	Komponentē tiek parādīts, ka sudoku mīkla ir nepareiza, ja tiek nospiests onCheck() un netiek saglabāta vēsture	Tiek parādīts, ka sudoku mīkla ir nepareiza un netiek saglabāts vēstures ieraksts	Izpildīts
9.	Padots sudoku objekts	Komponentē neizmainās stāvoklis, kad tiek izsaukts toggleDropdown() ar nederīgu simbolu	Stāvoklis nemainās	Izpildīts
10.	Padots sudoku objekts un funkcija onChangeSave()	Komponentē izmainās stāvoklis un netiek saglabāts vēstures ieraksts, kad tiek izsaukts onChange()	Stāvoklis izmainās un vēstures ieraksts nesaglabājas	Izpildīts
11.	Padots sudoku objekts un funkcija onChangeSave()	Komponentē izmainās stāvoklis un tiek saglabāts vēstures ieraksts, kad tiek izsaukts onChange()	Stāvoklis izmainās un vēstures ieraksts saglabājas	Izpildīts
12.	Padots sudoku un historyEntry objekts, un lietotājs ir pieteicies sistēmā	Komponentē tiek iestatīts sākotnējais stāvoklis ar vēstures ieraksta datiem	Stāvoklis tiek iestatīts	Izpildīts

SudokuBodyContainer	13.	Padod location objektu un funkciju history.push()	Konteinerī tiek izsaukta history.push() funkcija vienu reizi	Vienu reizi izsaukta funkcija	Izpildīts
	14.	Padod location objektu ar state.sudoku un state.historyEntry	Konteineris sniedz SudokuBody komponenti no vēstures	Konteineris sniedz SudokuBody komponenti no vēstures	Izpildīts
	15.	Padod location objektu ar state.historyEntry, bet bez state.sudoku	Konteineris nesniedz SudokuBody komponenti	Konteineris nesniedz SudokuBody komponenti	Izpildīts
	16.	Padod location objektu ar state.sudoku	Konteineris sniedz SudokuBody komponenti	Konteineris sniedz SudokuBody komponenti	Izpildīts
	17.	Padod location objektu ar state.errorMessage	Konteineris sniedz kļūdas ziņojumu	Konteineris sniedz kļūdas ziņojumu	Izpildīts
SudokuGridContainer	18.	-	Konteineris nesniedz SudokuGrid komponenti	Konteineris nesniedz SudokuGrid komponenti	Izpildīts
	19.	Padod sudoku un historyEntry objektus	Konteineris sniedz SudokuGrid komponenti no vēstures	Konteineris sniedz SudokuGrid komponenti no vēstures	Izpildīts
	20.	Padod sudoku objektu	Konteineris sniedz SudokuGrid komponenti	Konteineris sniedz SudokuGrid komponenti	Izpildīts
	21.	Padod funkciju updateHistoryEntry()	Konteinerī izsauc funkciju updateHistoryEntry() vienu reizi, kas tiek izsaukta onChangeSave() funkcija	Funkcija tiek izsaukta vienu reizi	Izpildīts

4.2.2.5 Vēstures modulis

Fails	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
HistoryList	1.	Padots dividedHistory objekts	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts

HistoryCard	2.	Padots historyEntry objekts, kurš ir atrisināts	Komponente sniedz sagaidāmo html struktūru ar atrisināto vēstures ierakstu	Tiek parādīts atrisinātais vēstures ieraksts	Izpildīts
	3.	Padots historyEntry objekts, kurš nav atrisināts	Komponente sniedz sagaidāmo html struktūru ar neatrisināto vēstures ierakstu	Tiek parādīts neatrisinātais vēstures ieraksts	Izpildīts
	4.	Padota funkcija onContinue()	Komponentē izsauc onContinue funkciju 1 reizi, ja tiek nospiests uz turpināt pogas	onContinue() funkcija tiek izsaukta vienu reizi	Izpildīts
	5.	Padots historyEntry objekts, kurš ir atrisināts un, kurā izmantoja atrisināšanas funkcionalitāti	Komponente sniedz sagaidāmo html struktūru ar atrisināto vēstures ierakstu un uzrakstu "with solve"	Tiek parādīts atrisinātais vēstures ieraksts ar uzrakstu "with solve"	Izpildīts
	6.	Padots historyEntry objekts, kurš ir atrisināts un, kurā neizmantoja atrisināšanas funkcionalitāti	Komponente sniedz sagaidāmo html struktūru ar atrisināto vēstures ierakstu un uzrakstu "without solve"	Tiek parādīts atrisinātais vēstures ieraksts ar uzrakstu "without solve"	Izpildīts
HistoryCardContainer	7.	-	Konteineris sniedz sagaidāmo komponenti	Konteineris sniedz sagaidāmo komponenti	Izpildīts
	8.	Padots historyEntry objekts	Konteineris sniedz komponenti HistoryCard	Konteineris sniedz komponenti HistoryCard	Izpildīts
	9.	Padots historyEntry objekts un funkcijas getSudoku() un redirectToSudoku()	Konteinerī izsauc getSudoku() un redirectToSudoku(), kad tiek izsaukta onContinue() funkcija	Abas funkcijas tiek izsauktas	Izpildīts
	10.	Padotas funkcijas getSudoku() un redirectToSudoku()	Konteinerī neizsauc getSudoku() un redirectToSudoku(), kad tiek izsaukta onContinue() funkcija, jo nav padots historyEntry	Abas funkcijas netiek izsauktas	Izpildīts

	11.	Padots historyEntry objekts un funkcijas getSudoku() un redirectToSudoku()	Konteinerī izsauc getSudoku(), bet neizsauc redirectToSudoku(), kad tiek izsaukta onContinue() funkcija, jo getSudoku() atgriež kļūdu	Tiek izsaukta getSudoku() funkcija, bet netiek izsaukta redirectToSudoku()	Izpildīts
HistoryListContainer	12.	Padota funkcija getDividedHistoryEntries()	Konteinerī izsauc getDividedHistoryEntries() vienu reizi	Funkcija tiek izsaukta vienu reizi	Izpildīts
	13.	Padots dividedHistory objekts un funkcija getDividedHistoryEntries()	Konteinerī sniedz HistoryList komponenti	Konteinerī sniedz HistoryList komponenti	Izpildīts
	14.	Padotas funkcijas getDividedHistoryEntries() un history.push()	Konteinerī izsauc getDividedHistoryEntries() un history.push() ar norādītajiem parametriem	Izsauktas abas funkcijas ar norādītajiem parametriem	Izpildīts

4.2.2.6 Lietotāja profila modulis

Fails	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
ProfileCard	1.	-	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	2.	-	Komponentē izmainās stāvoklis, kad tiek izsauktas funkcijas onEditOpen() un onEditClose()	Stāvoklis izmainās pēc katras funkcijas izsaukšanas	Izpildīts
	3.	Padots user objekts un funkcija onUpdate()	Komponentē izmainās stāvoklis un tiek izsaukta onUpdate() funkcija, un lietotājs netiek novirzīts	Stāvoklis izmainās, onUpdate funkcija tiek izsaukta un lietotājs netiek novirzīts	Izpildīts
	4.	Padots user objekts un funkcija onUpdate()	Komponentē izmainās stāvoklis un tiek izsaukta onUpdate() funkcija, un lietotājs tiek novirzīts	Stāvoklis izmainās, onUpdate funkcija tiek izsaukta un lietotājs tiek novirzīts	Izpildīts

	5.	-	Komponentē tiek pārveidots laika formāts uz lasāmāku formātu	Laika formāts tiek pārveidots	Izpildīts
	6.	Padots user un statistics objekts	Komponentē tiek izsaukts onEditOpen(), kad tiek nospiests uz pogas	Tiek izsaukta funkcija onEditOpen()	Izpildīts
	7.	Padots user un statistics objekts	Komponentē tiek atvērts lietotāja rediģēšanas logs	Tiek atvērta lietotāja rediģēšana	Izpildīts
UserEdit	8.	Padots user objekts	Komponente sniedz sagaidāmo html struktūru ar lietotāja informāciju	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	9.	Padots user objekts	Komponentē izmainās stāvoklis, kad tiek izsaukts onChange()	Stāvoklis izmainās	Izpildīts
	10.	Padots user objekts un funkcija onClose()	Komponentē izsauc onClose() funkciju 1 reizi, ja tiek nospiests uz atcelšanas pogas	onClose() funkcija tiek izsaukta vienu reizi	Izpildīts
	11.	Padots user objekts un funkcija onUserUpdate()	Komponentē izsauc onUserUpdate() funkciju 1 reizi, ja tiek nospiests uz saglabāšanas pogas	onUserUpdate() funkcija tiek izsaukta vienu reizi	Izpildīts
ProfileCardContainer	12.	Padotas funkcijas getUserSelf() un getHistoryStatistics() un lietotājs ir pieteicies sistēmā	Konteinerī funkcijas getUserSelf() un getHistoryStatistics() izsauc katru vienu reizi	Funkcijas izsauktas katra pa vienai reizei	Izpildīts
	13.	Padots user objekts	Konteineris sniedz ProfileCard komponenti	Konteineris sniedz ProfileCard komponenti	Izpildīts
	14.	Padotas funkcijas getUserSelf(), updateUser(), logout(), push()	Konteinerī izsauc updateUser(), logout() un history.push(), bet neizsauc getUserSelf()	Izsauktas nepieciešamās funkcijas un netiek izsaukta getUserSelf()	Izpildīts
	15.	Padota funkcija updateUser()	Konteinerī netiek izsaukta funkcija updateUser()	Netiek izsaukta funkcija	Izpildīts

	16.	Padotas funkcijas getUserSelf(), updateUser(), logout(), push()	Konteinerī izsauc updateUser() un getUserSelf(), bet neizsauc logout() un history.push()	Izsauktas updateUser() un getUserSelf(), bet netiek izsauktas logout() un history.push()	Izpildīts
--	-----	--	--	---	-----------

4.2.2.7 Administratora modulis

Fails	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
AdminDashboard	1.	-	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	2.	Padota funkcija onDeleteUser()	Komponentē izmainās stāvoklis un onDeleteUser() funkcija tiek izsaukta vienu reizi, kad tiek izsaukta funkcija handleDelete()	Stāvoklis izmainās un onDeleteUser() funkcija tiek izsaukta vienu reizi	Izpildīts
	3.	-	Komponentē izmainās stāvoklis, kad tiek izsaukta funkcija onOpenUserCard()	Stāvoklis izmainās	Izpildīts
	4.	-	Komponentē izmainās stāvoklis, kad tiek izsaukta funkcija onCloseUserCard()	Stāvoklis izmainās	Izpildīts
	5.	Padots adminDashboarddata objekts	Komponentē tiek sakārtoti lietotāji pēc to pievienošanas datuma	Lietotāji tiek sakārtoti	Izpildīts
	6.	Padots adminDashboarddata objekts	Komponentē tiek izsaukts onOpenUserCard(), kad tiek nospiests uz lietotājevārda	Tiek izsaukta funkcija onOpenUserCard()	Izpildīts
	7.	Padots adminDashboarddata objekts	Komponentē tiek atvērts UserCard, kad stāvoklis nomainās uz "atvērts"	Tiek atvērta lietotāja informācija	Izpildīts

UserCard	8.	Padots user objekts	Komponente sniedz sagaidāmo html struktūru ar lietotāja informāciju	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	9.	Padots user objekts un funkcija handleDelete()	Komponentē tiek izsaukta handleDelete() funkcija, kad tiek nospiesta poga "Dzēst"	handleDelete() funkcija tiek izsaukta	Izpildīts
AdminDashboardContainer	10.	-	Konteineris sniedz sagaidāmo komponenti	Konteineris sniedz sagaidāmo komponenti	Izpildīts
	11.	Padots user objekts ar lietotāja līmeni "lietotājs" un funkcijas getUserSelf(), history.push() un getAdminDashboardData()	Konteinerī netiek izsaukta getAdminDashboardData() funkcija un lietotājs tiek novirzīts	Netiek izsaukta funkcija un lietotājs tiek novirzīts	Izpildīts
	12.	Padots user objekts ar lietotāja līmeni "administrators" un funkcijas getUserSelf() un getAdminDashboardData()	Konteinerī tiek izsaukta getAdminDashboardData() funkcija	Tiek izsaukta funkcija	Izpildīts
	13.	Padotas deleteUser() un getAdminDashboardData() funkcijas	Konteinerī tiek izsaukta deleteUser() funkcija un getAdminDashboardData() funkcija	Tiek izsauktas abas funkcijas	Izpildīts

4.2.2.8 Sudoku pievienošanas modulis

Faili	Nr.	Priekšnosacījumi	Testpiemērs	Sagaidāmais rezultāts	Statuss
SudokuAdd	1.	-	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
	2.	-	Komponentē izmainās stāvoklis, kad tiek izsaukta onChange() ar derīgu simbolu	Stāvoklis izmainās	Izpildīts

	3.	-	Komponentē neizmainās stāvoklis, kad tiek izsaukta <code>onInputChange()</code> ar nederīgu simbolu	Stāvoklis neizmainās	Izpildīts	
	4.	-	Komponentē izmainās stāvoklis, kad tiek izsaukta <code>onDifficultyChange()</code> ar derīgu grūtības pakāpi	Stāvoklis izmainās	Izpildīts	
	5.	Padota funkcija <code>onSave()</code>	Komponentē tiek izsaukta <code>onSave()</code> funkcija un nomainīts stāvoklis, kad izsauc <code>onSudokuSave()</code> funkciju	Tiek izsaukta <code>onSave()</code> funkcija un nomainīts stāvoklis	Izpildīts	
	6.	Padota funkcija <code>onSave()</code>	Komponentē netiek izsaukta <code>onSave()</code> funkcija un tiek iestatīts <code>errorMessage</code> par dublikātiem	Netiek izsaukta funkcija <code>onSave()</code> un tiek iestatīts <code>errorMessage</code>	Izpildīts	
	7.	Padota funkcija <code>onSave()</code>	Komponentē netiek izsaukta <code>onSave()</code> funkcija un tiek iestatīts <code>errorMessage</code> par grūtības pakāpi	Netiek izsaukta funkcija <code>onSave()</code> un tiek iestatīts <code>errorMessage</code>	Izpildīts	
	8.	-	Komponentē tiek mainīts stāvoklis, kad tiek izsaukta <code>onOpenApproval()</code>	Stāvoklis izmainās.	Izpildīts	
	9.	-	Komponentē tiek mainīts stāvoklis, kad tiek izsaukta <code>onCloseApproval()</code>	Stāvoklis izmainās	Izpildīts	
	SudokuAddApproval	10.	-	Komponente sniedz sagaidāmo html struktūru	Komponente sniedz sagaidāmo html struktūru	Izpildīts
		11.	Padots user objekts un funkcija <code>handleSave()</code>	Komponentē tiek izsaukta <code>handleSave()</code> funkcija, kad tiek nospiesta poga "Apstiprināt"	<code>handleSave()</code> funkcija tiek izsaukta	Izpildīts

SudokuAddContainer	12.	-	Konteineris sniedz sagaidāmo komponenti	Konteineris sniedz sagaidāmo komponenti	Izpildīts
	13.	Padots user objekts ar lietotāja līmeni "user" un funkcijas getUserSelf() un history.push() un lietotājs ir pieteicies sistēmā	Konteinerī tiek izsauktas funkcijas getUSeRSelf() un history.push()	Tiek izsauktas abas funkcijas	Izpildīts
	14.	Padots user objekts ar lietotāja līmeni "admin" un funkcija getUserSelf()	Konteineris sniedz SudokuAdd komponenti	Konteineris sniedz SudokuAdd komponenti	Izpildīts
	15.	Padota funkcija addSudoku()	Konteinerī tiek izsaukta addSudoku() funkcija, kad izsauc onSave() funkciju	Tiek izsaukta funkcija	Izpildīts

5. PROJEKTA ORGANIZĀCIJA

Projekts tika izstrādāts pēc Kanban spējās izstrādes ietvara, kas atviegloja jaunas funkcionalitātes ieviešanu un izmaiņu veikšanu lietotnē. Projekts tika iedalīts divās daļās, aizmugursistēmā un lietotāja saskarnē, un tās abas izstrādāja viens cilvēks. Projektā piedalījās darba vadītājs, kurš sniedza dažādus ieteikumus programmatūras izstrādē un dokumentācijas veidošanā.

Projektā tika izmantots Git versiju kontroles rīks un publiski pieejamas repozitorijas abām sistēmas daļām GitHub vietnē. Projekta plānošanā tika izveidotas ‘problēmu biļetes’ atbilstošajās repozitorijās, ar unikāliem identifikācijas kodiem, kurus turpmāk izmantoja repozitorijas atzaru veidošanā (skat. 7. Konfigurāciju pārvaldība).

6. KVALITĀTES NODROŠINĀŠANA

Projekta izstrādes laikā tika veiktas vairākas aktivitātes kvalitātes nodrošināšanai:

- programmatūras prasību specifikācijas izstrāde,
- programmatūras projektējuma apraksta izstrāde, balstoties uz specifikāciju,
- programmatūras izstrāde,
- programmatūras testēšana un tās dokumentēšana,
- versiju kontroles rīku izmantošana projekta izstrādē,

Projekts tika izstrādāts Visual Studio Code izstrādes vidē. Aizmugursistēmas koda struktūra galvenokārt sastāv no modeļiem, kontrolieriem un maršrutiem, toties lietotāja saskarnes koda struktūra – no komponentēm, konteineriem, redux darbībām un reducētājiem. Kā sistēmas datu bāze tika izmantota bez-relāciju datu bāze MongoDB.

Programmatūras kods tika rakstīts, balstoties uz programmēšanas labo praksi, t.i., failus, klases, funkcijas un mainīgos nosaucot viegli saprotamos un nozīmei atbilstošos vārdos, kā rezultātā veidojas viegli lasāms kods. Nepieciešamības gadījumā kodā tika rakstīti komentāri.

7. KONFIGURĀCIJU PĀRVALDĪBA

Projektā tika izmantota Git versiju kontroles sistēma, kas nodrošināja programmatūras pirmkoda, projekta vadības un programmatūras versiju pārvaldību. Programmatūras kods tika augšupielādēts publiski pieejamās repozitorijās GitHub vietnē.

GitHub vietnē tika izveidotas divas repozitorijas, atbilstoši abām projekta daļām. Katrai funkcionalitātei tika izveidotas atbilstošas ‘problēmu biļetes’, ar unikāliem identifikatoriem. Aizmugursistēmā šie identifikatori atbilda SAA-<problēmas kārtas skaitlis>, kur SAA apzīmē Sudoku-App-API (repozitorijas nosaukums), toties lietotāja saskarnē identifikatori atbilda SRA-<problēmas kārtas skaitlis>, kur SRA apzīmē Sudoku-React-App (repozitorijas nosaukums). Pēc šiem identifikatoriem tika veidoti versijas atzari, kuros tika veikta norādītās funkcionalitātes izstrāde. Kad tā tika izstrādāta un pārbaudīta, tad kods tika augšupielādēts GitHub vietnē ar komentāru par padarīto. Toties, lai jaunizveidotā funkcionalitāte tiktu ieviesta jau esošajā programmatūras kodā, tad tika veidoti ‘vilksanas pieprasījumi’ (pull requests), kas norādīja uz koda apvienošanas problēmām, ja tādas bija, un piespieda vēlreiz pārskatīt kodu, pirms tas tiek apvienots.

8. DARBIETILPĪBAS NOVĒRTĒJUMS

Projekts tika izstrādāts pēc Kanban spējās izstrādes ietvara, tādējādi izstrādātā funkcionalitāte tika sadalīta moduļos, kuri tika vērtēti funkcijpunktos.

Lietotāja saskarnes funkcijpunkta vērtība tika pielīdzināta vienai cilvēkdienai, kas ir 8 stundas, toties aizmugursistēmā funkcijpunkta vērtību pielīdzināja vienai pusei cilvēkdienas, kas ir 4 stundas. Aizmugursistēmas funkcijpunkta vērtība tika izvēlēta zemāka, jo autoram jau bija neliela pieredze aizmugursistēmu izstrādē, tādējādi tās izstrādei izmantoto rīku un tehnoloģiju apguvei paredzot zemāku laika patēriņu.

Aizmugursistēmas izstrāde tika novērtēta 28 funkcijpunktos (skat. 1. Pielikumu), kas, attiecīgi iepriekš pieminētajam, atbilst 14 cilvēkdienām jeb aptuveni 0,7 personmēnešiem, pieņemot, ka viens personmēnesis ir aptuveni 20 cilvēkdienas.

Lietotāja saskarnes izstrāde tika novērtēta 44 funkcijpunktos (skat. 2. Pielikumu), kas, attiecīgi iepriekš minētajam, atbilst 44 cilvēkdienām jeb aptuveni 2,2 personmēnešiem, pieņemot, ka viens personmēnesis ir aptuveni 20 cilvēkdienas.

Kopā abu sistēmas daļu izstrādes process veido aptuveni 2,9 personmēnešus, kur netiek iekļauti projekta plānošanas darbi, kas norisinājās 1 cilvēknedēļā, kas attiecīgi ir aptuveni 0,25 personmēneši. Apvienojot sistēmas projektēšanas procesam patērēto laiku un sistēmas izstrādei patērēto laiku, tad kopā aptuvenā darbietilpība vērtējama 3,1 personmēnešu apjomā.

Toties darbietilpības vērtējums ir tikai aptuvens rādītājs. Reāli sistēmas izstrāde notika nedaudz vairāk kā 2 mēnešus, ko var redzēt autora uzņemtajos ekrānu uzņēmumos no abu sistēmas daļu GitHub repozitorijām, kur attēlots koda augšupielādes reižu skaits kopš repozitoriju izveidošanas (skat. 3. un 4. Pielikumu). Tas skaidrojams ar to, ka darbietilpības novērtējumā tika pieņemts, ka viena cilvēkdiena ir attiecināma 8 stundām, bet izstrādes laikā tā bieži vien pārsniedza 8 stundas.

9. SECINĀJUMI

Kvalifikācijas darba izstrādes rezultāts ir pilnībā funkcionējoša sudoku mīklu risināšanas lietotne, kas ir pieejama modernākajās tīmekļa pārlūkprogrammās, tās programmatūras prasību specifikācija, programmatūras prasību apraksts, testēšanas dokumentācija un tās programmatūras kods.

Darba projektēšanas laikā izvirzītie mērķi tika izpildīti, un attiecīgā funkcionalitāte ir ieviesta lietotnē. Projekts atbilst noteiktajām drošības un veiktspējas prasībām (skat. 5. Pielikumu). Darba izstrādes procesā tika secināts, ka izvēlētās tehnoloģijas bija atbilstošas projekta vajadzībām.

Kopumā kvalifikācijas darba izstrāde ir palīdzējusi autoram padziļinātāk apgūt jaunākās tīmekļa vieņu izstrādes tehnoloģijas, uzlabot savas zināšanas versiju kontroles sistēmās un to saprotamas struktūras nodrošināšanā, saprast, ka gandrīz vienmēr izstrādes procesā nāksies saskarties ar dažādām problēmām un kļūmēm, izprast vienībtestēšanas nozīmi un būtiskumu, izprast izstrādes tehnoloģiju dokumentāciju izmantošanas nepieciešamību projektu izstrādē, kā arī darba plānošanas nepieciešamību, lai nonāktu pie kvalitatīva rezultāta.

10. IZMANTOTĀ LITERATŪRA

1. NodeJS dokumentācija. [atsauce 23.05.2019] Pieejams: <https://nodejs.org/en/>
2. ReactJS dokumentācija. [atsauce 23.05.2019] Pieejams: <https://reactjs.org/>
3. Iepazīstināšana ar JSX sintaksi ReactJS dokumentācijā. [atsauce 23.05.2019] Pieejams: <https://reactjs.org/docs/introducing-jsx.html>
4. MongoDB dokumentācija. MongoDB un MySQL salīdzinājums. [atsauce 23.05.2019] Pieejams: <https://www.mongodb.com/compare/mongodb-mysql>
5. JSON Web Token dokumentācija. [atsauce 24.05.2019] Pieejams: <https://jwt.io/introduction/>
6. Redux stāvokļu ietvara dokumentācija. [atsauce 24.05.2019] Pieejams: <https://redux.js.org/>
7. Latvijas Universitātes E-studiju materiāls kursā Metakurss: Kvalifikācijas darbs, Microsoft Word fails “Kvalifikācijas darba izstrādes un aizstāvēšanas metodiskie norādījumi 2014”. [atsauce 25.05.2019] Pieejams E-studiju kursā: <https://estudijas.lu.lv/course/view.php?id=77>
8. Latvijas Universitātes E-studiju materiāls kursā DatZ2072: Programminženierija, Microsoft Word fails “LVS 96 1996 PPS - izkrāsots”. [atsauce 25.05.2019] Pieejams E-studiju kursā: <https://estudijas.lu.lv/course/view.php?id=3582>

11. PIELIKUMS

1. Pielikums. Aizmugursistēmas funkcijpunktu novērtējums pa moduļiem

Funkcionalitāte	Funkcijpunkti
Autorizācijas modulis	6
Lietotāja modulis	4
Sudoku modulis	8
Vēstures modulis	8
Administratora modulis	2
Kopā:	28

2. Pielikums. Lietotāja saskarnes funkcijpunktu novērtējums pa moduļiem

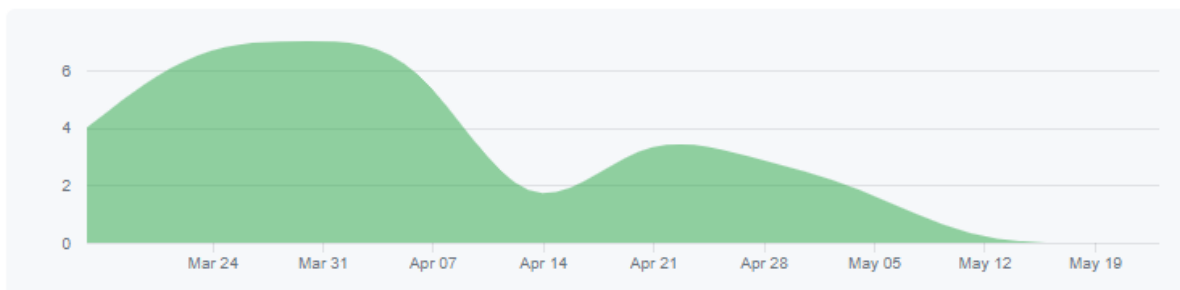
Funkcionalitāte	Funkcijpunkti
Pieteikšanās modulis	6
Reģistrācijas modulis	6
Pamata modulis	4
Sudoku modulis	10
Vēstures modulis	4
Lietotāja profila modulis	4
Administratora modulis	4
Sudoku pievienošanas modulis	6
Kopā:	44

3. Pielikums. Koda augšupielādes reizes aizmugursistēmas GitHub repozitorijā.

Mar 17, 2019 – May 24, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits

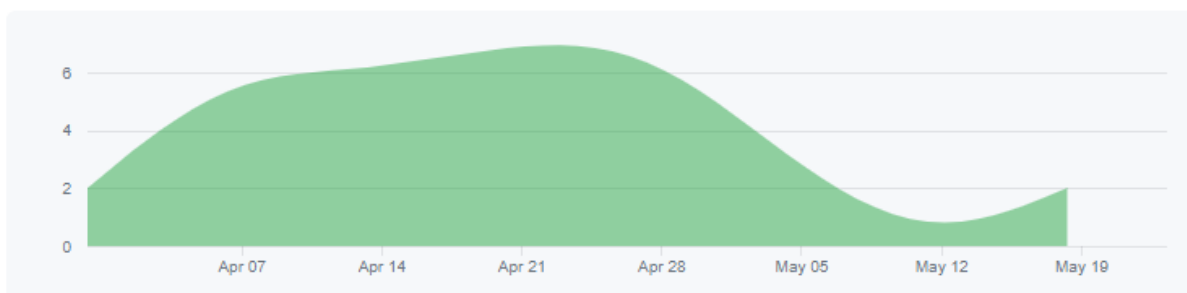


4. Pielikums. Koda augšupielādes reizes lietotāja saskarnes GitHub repozitorijā.

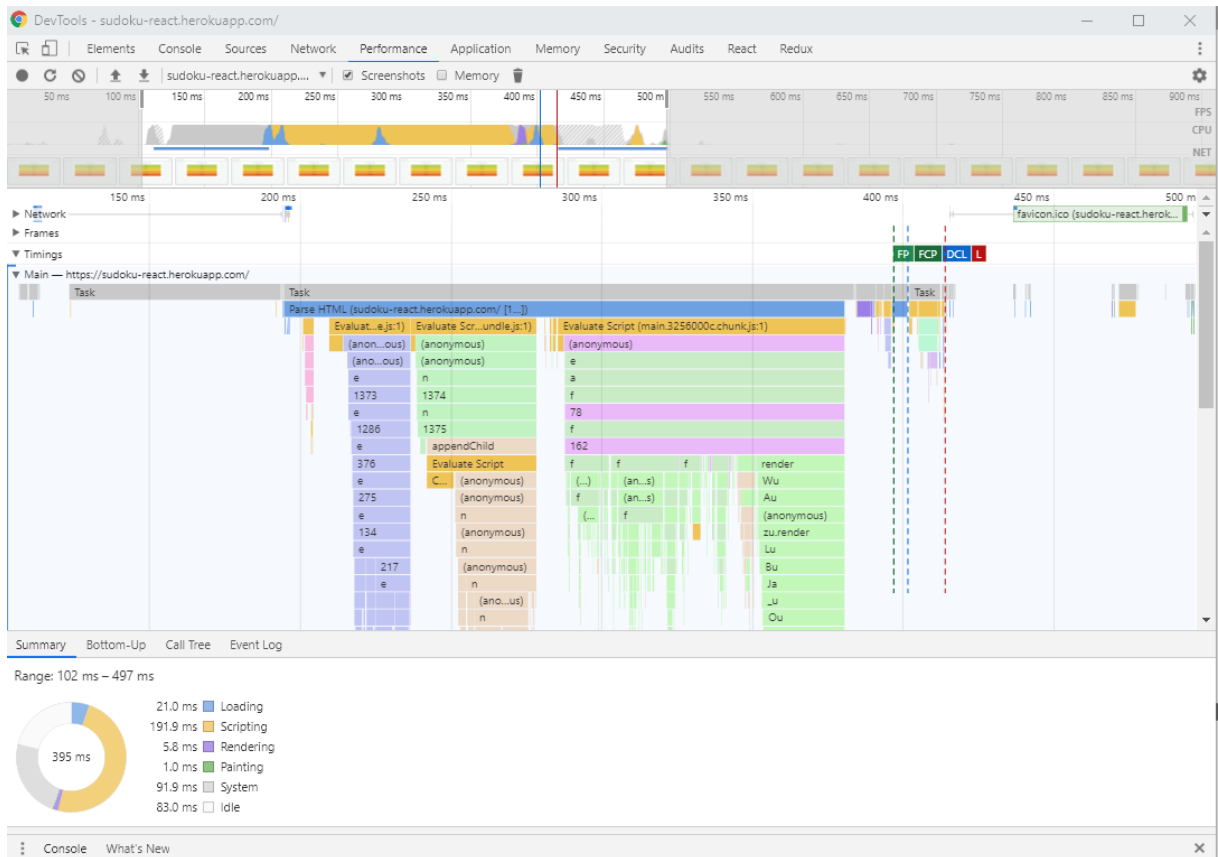
Mar 31, 2019 – May 24, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits



5. Pielikums. Sistēmas atbildes laiks uz vietnes ielādes pieprasījumu



Kvalifikācijas darbs „Tīmekļa lietotne sudoku mīklu risināšanai” 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: **Reinis Šadinovs** _____ .05.2019.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: **B.dat. Jānis Knets** _____ .05.2019.

Recenzents: **Miks Rozenbergs**

Darbs iesniegts 27.05.2019.

Kvalifikācijas darbu pārbaudījumu komisijas sekretāre: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2019. prot. Nr. _____

Komisijas sekretārs(-e): _____