



INFORMATIKOS FAKULTETAS

T120B165 Saityno taikomųjų programų projektavimas

Kursinio darbo ataskaita

Studentas: Vilius Vaičiulis, IFF-4/1
Dėstytojas: Tomas Blažauskas

KAUNAS 2017

1 Turinys

1. Projekto aprašymas	3
1.1 Sistemos paskirtis	3
1.2 Funkciniai reikalavimai	3
2 Sistemos architektūra	3
3 Vartotojo sąsajos projektas	4
4 API specifikacija	11
4.1 Darbalaukio objekto API specifikacija	11
4.2 Užduoties objekto API specifikacija	13
4.3 Vartotojo registracijos API specifikacija	15
5 API panaudojimo pavyzdžiai	15
6 Išvados	18

1. Projekto aprašymas

1.1 Sistemos paskirtis

Projekto tikslas – sukurti užduočių planavimo internetinę aplikaciją.

Veikimo principas – vartotojas mato savo užduočių korteles skirtinguose stulpeliuose („to do“, „doing“, „done“, „archived“), gali susikurti naujų užduočių, perkelti jas iš vieno stulpelio į kitą.

Vartotojas gali susikurti skirtingus darbalaukius, kuriuose gali susikurti skirtingų sričių užduotis. Galima susikurti naujų užduočių, peržiūrėti esamų sąrašą, detalai peržiūrėti vieną užduotį, pašalinti esamas užduotis. Tokiu būdu aplikacija padeda vartotojui lengviau struktūrizuoti ir planuoti visas savo turimas užduotis.

1.2 Funkciniai reikalavimai

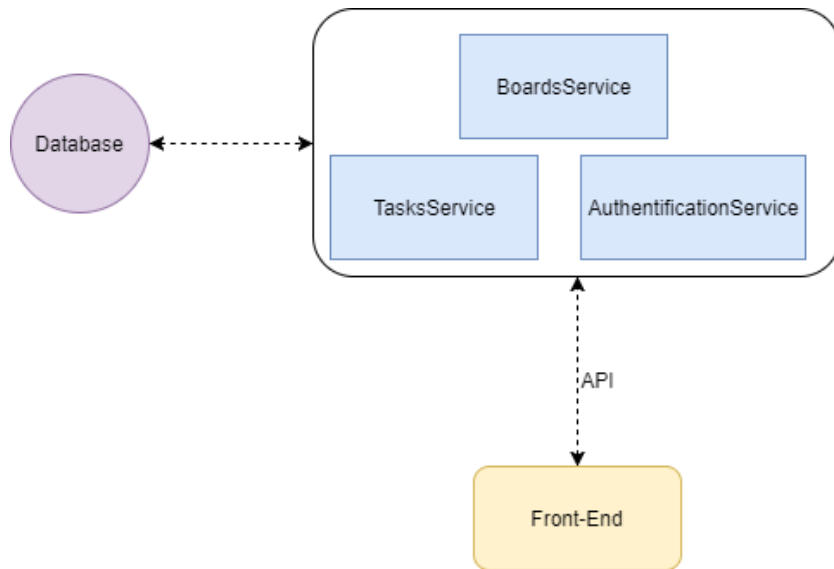
Planuojamos funkcijos:

- Naujo darbalaukio sukūrimas
- Darbalaukio pašalinimas
- Darbalaukio redagavimas
- Darbalaukių peržiūra
- Naujos užduoties sukūrimas
- Detali užduoties peržiūra
- Visų užduočių sąrašo peržiūra
- Esamos užduoties pakeitimas
- Užduočių perkėlimas tarp stulpelių
- Užduočių pašalinimas
- Prisijungimas su vartotojo vardu ir slaptažodžiu
- Registracija

2 Sistemos architektūra

Sudedamosios dalys:

- Front-end – naudojant React + redux + webpack
- Back-end – naudojant NodeJs. Pagrindiniai servisai – duomenų bazės valdymo ir vartotojų autentifikacija.

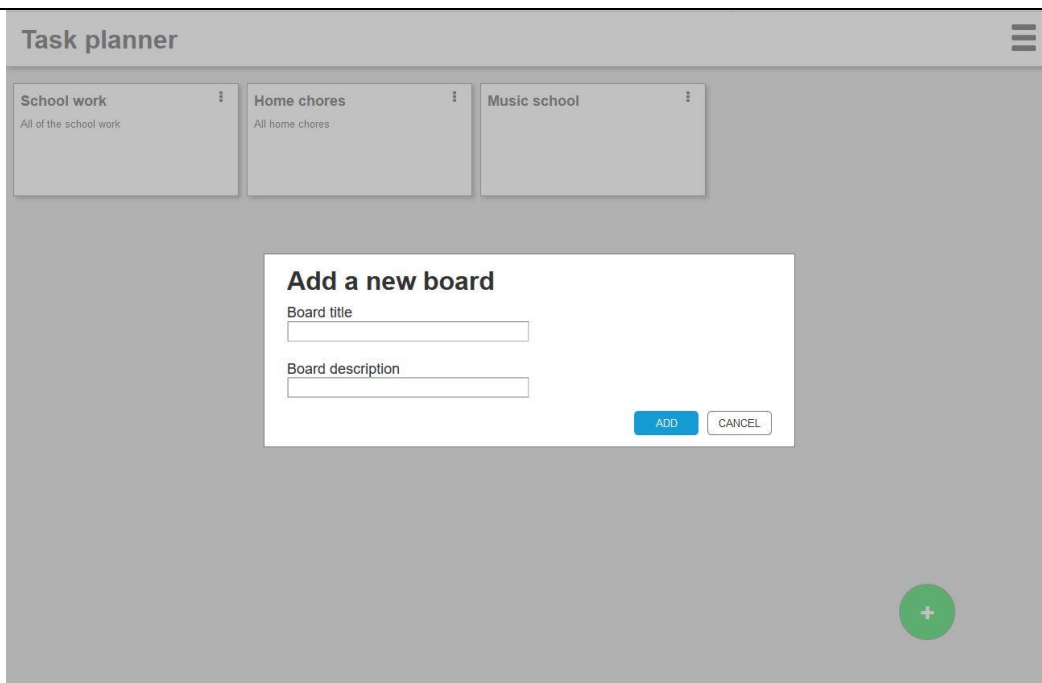


1 pav. Sistemos architektūra

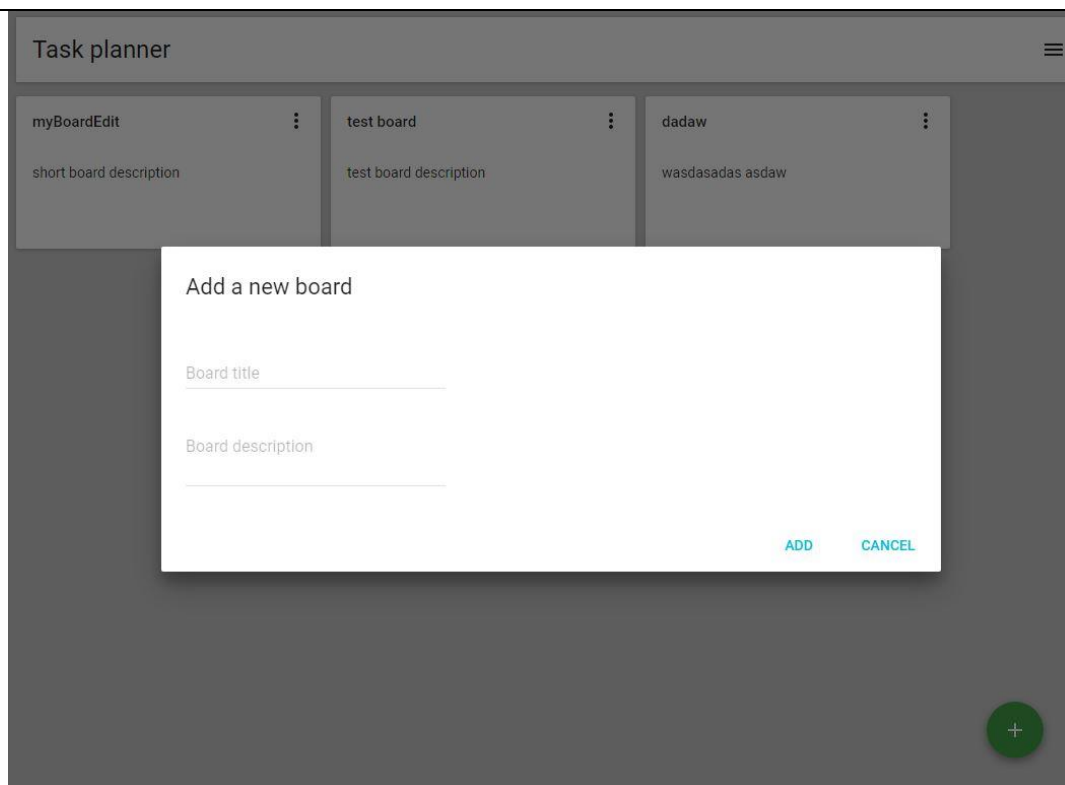
3 Vartotojo sąsajos projektas

Vartotojo sąsajos projektas buvo sukurtas naudojantis vartotojo sąsajos prototipų kūrimo programa Axure. Programuojant kliento pusę buvo atlikti keli patobulinimai, todėl prototipinė ir reali vartotojo sąsaja šiek tiek skiriasi.

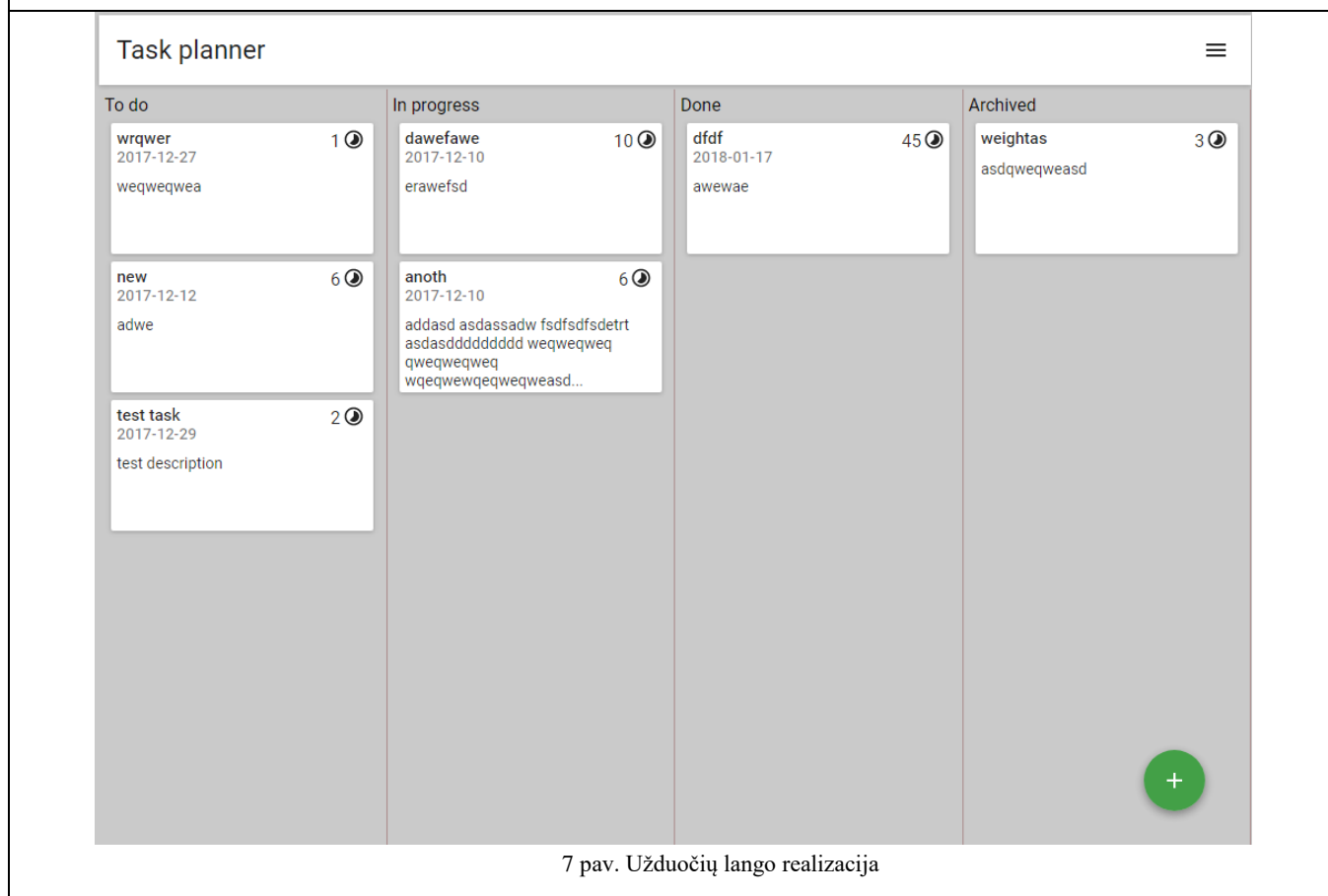
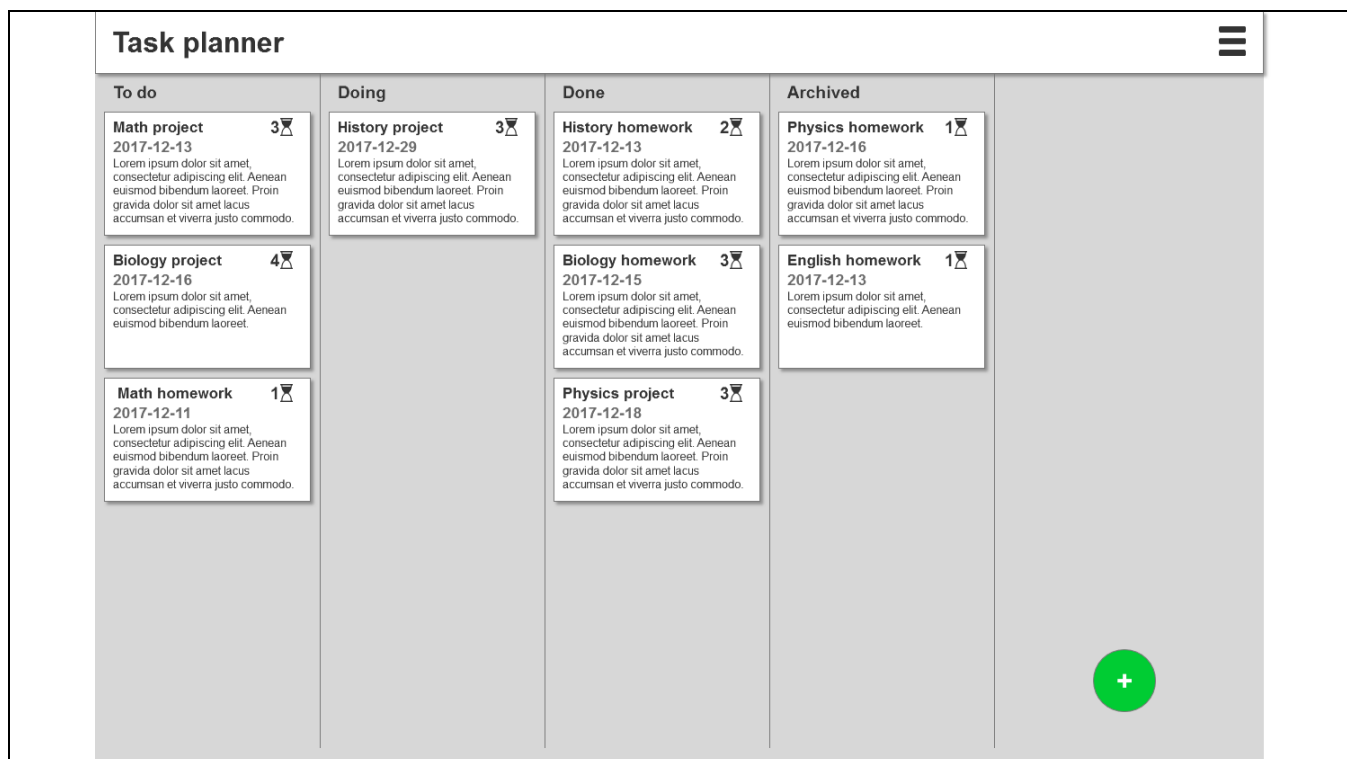


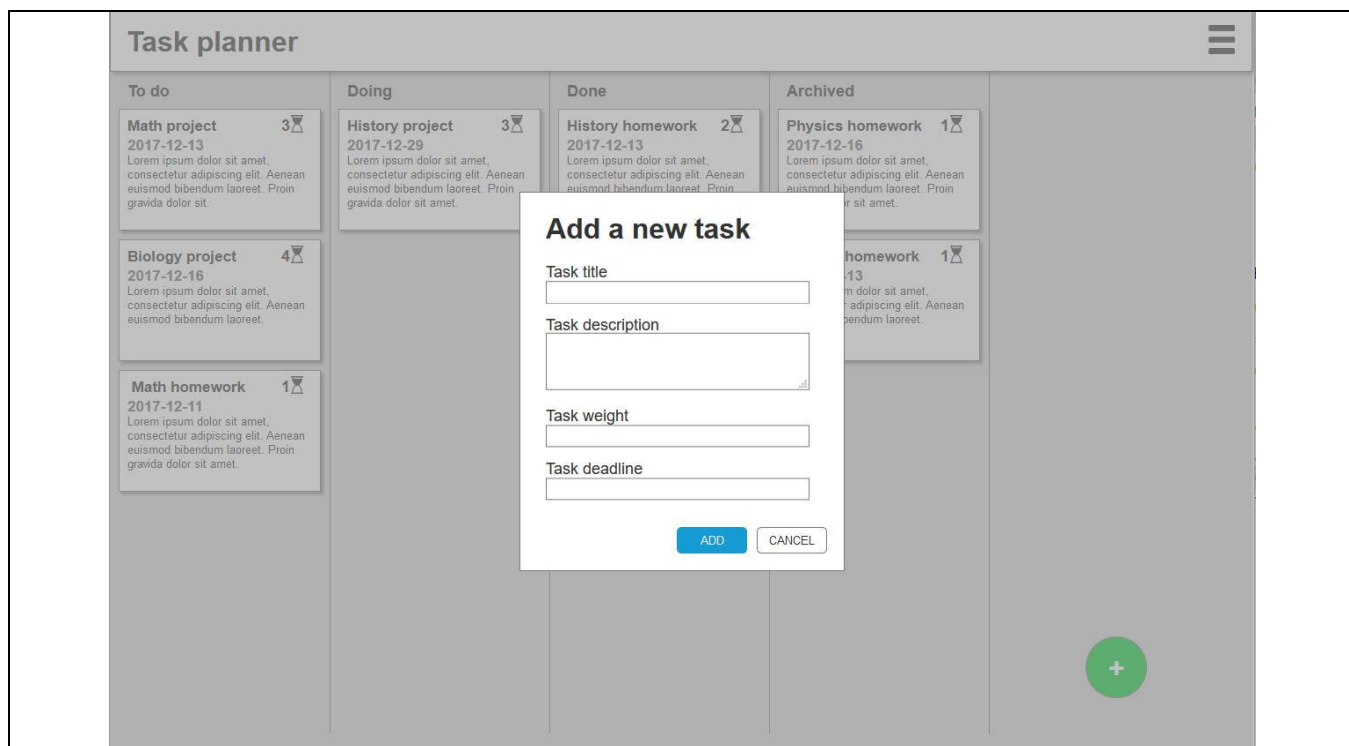


4 pav. Naujo darbalaukio kūrimo prototipas

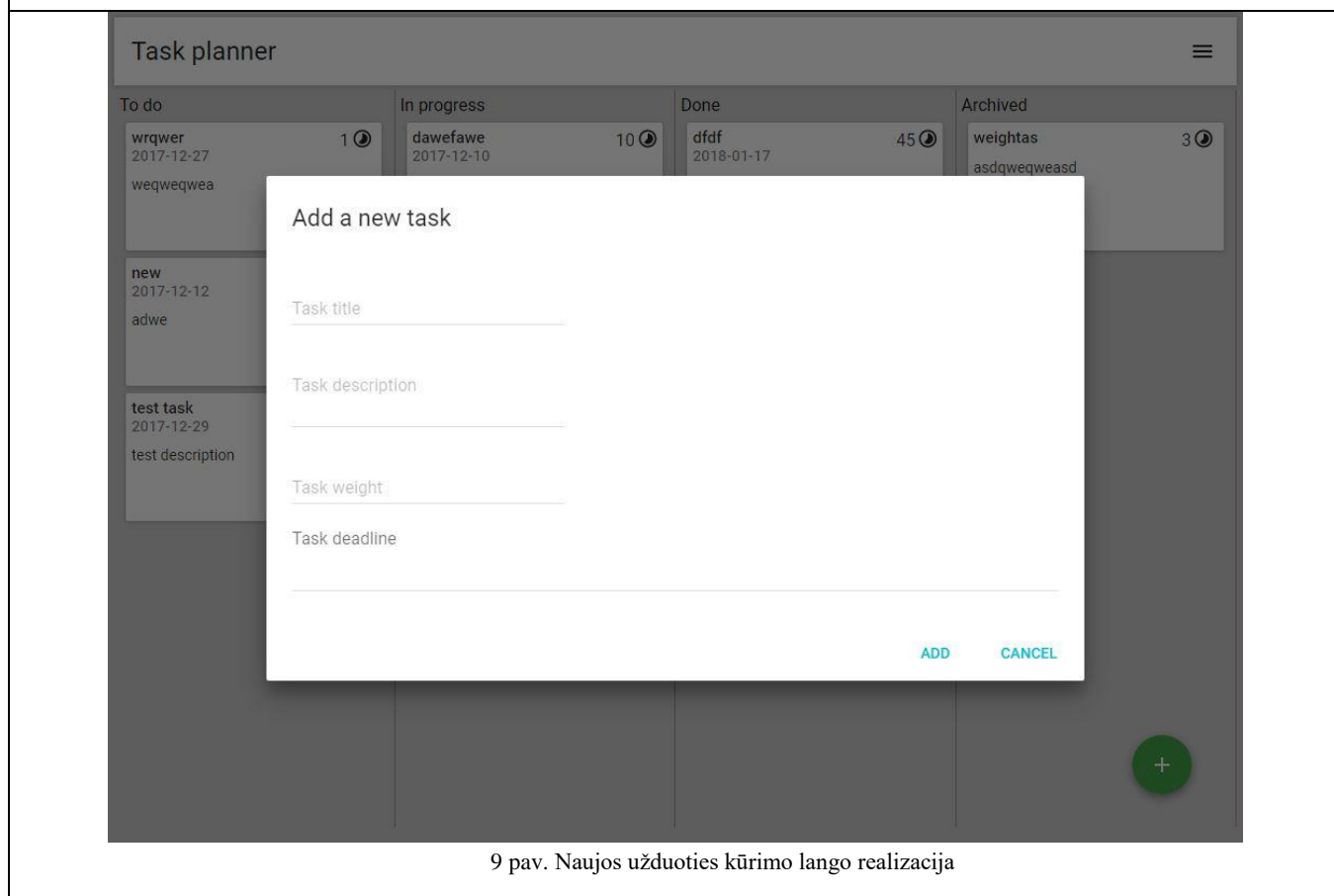


5 pav. Naujo darbalaukio kūrimo realizacija

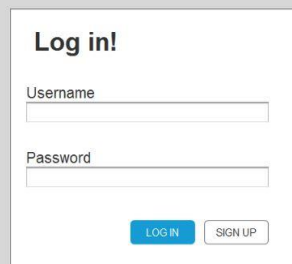




8 pav. Naujos užduoties kūrimo lango prototipas

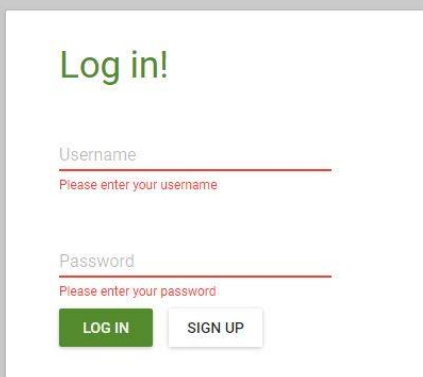


9 pav. Naujos užduoties kūrimo lango realizacija



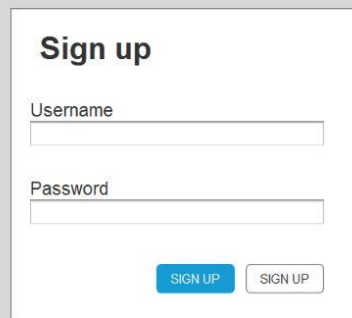
A login form prototype centered on a light gray background. The form is a white rectangle with a thin gray border. At the top, it says "Log in!" in bold black text. Below this are two input fields: "Username" and "Password", each with a light gray border. At the bottom, there are two buttons: a blue "LOG IN" button and a white "SIGN UP" button with a gray border.

10 pav. Prisijungimo lango prototipas



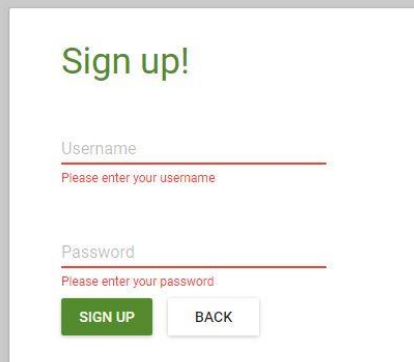
A login form realization centered on a light gray background. The form is a white rectangle with a thin gray border. At the top, it says "Log in!" in green text. Below this are two input fields: "Username" and "Password", each with a light gray border. Below the "Username" field, there is a red error message: "Please enter your username". Below the "Password" field, there is a red error message: "Please enter your password". At the bottom, there are two buttons: a green "LOG IN" button and a white "SIGN UP" button with a gray border.

11 pav. Prisijungimo lango realizacija



A wireframe prototype of a sign-up form. It features a white rectangular box with a thin black border. Inside the box, the title "Sign up" is at the top. Below it are two input fields: "Username" and "Password". At the bottom right of the box are two buttons: a blue one labeled "SIGN UP" and a white one with a grey border also labeled "SIGN UP".

12 pav. Registracijos lango prototipas



A final design realization of the sign-up form. The title "Sign up!" is in green. The input fields for "Username" and "Password" have red borders and red placeholder text: "Please enter your username" and "Please enter your password". The buttons are a green "SIGN UP" and a white "BACK" button with a grey border.

13 pav. Registracijos lango realizacija

4 API specifikacija

API specifikacija buvo struktūrizuota ir atvaizduota naudojant Swagger įrankį.

4.1 Darbalaukio objekto API specifikacija

GET /api/board

Description
Gets all user's boards.

Responses

Code	Description	Schema
200	success	<pre>▼ [▼ Board { name: string description: string }]</pre>
401	unauthorized	
404	not found	

Try this operation

14 pav. Visų vartotojo darbalaukių gavimas

POST /api/board

Description
Posts a new board.

Parameters

Name	Located in	Description	Required	Schema
body	body	New board object	Yes	<pre>▼ Board { name: string description: string }</pre>

Responses

Code	Description	Schema
201	success	<pre>▼ Board { name: string description: string }</pre>
401	unauthorized	

Try this operation

15 pav. Naujo darbalakio sukūrimas

GET /api/board/{id}

Description
Gets a board by id.

Parameters

Name	Located in	Description	Required	Schema
id	path	Id of the board to be retrieved.	Yes	≡ string

Responses

Code	Description	Schema
200	success	<pre> vBoard { name: string description: string } </pre>
401	unauthorized	
404	not found	

[Try this operation](#)

16 pav. Vieno darbalaukio gavimas

PUT /api/board/{id}

Description
Updates a board by id.

Parameters

Name	Located in	Description	Required	Schema
id	path	Id of the board to be updated.	Yes	≡ string
body	body	New board object	Yes	<pre> vBoard { name: string description: string } </pre>

Responses

Code	Description
204	success
401	unauthorized
404	not found

[Try this operation](#)

17 pav. Darbalaukio atnaujinimas

DELETE /api/board/{id}

Description
Deletes a board by id.

Parameters

Name	Located in	Description	Required	Schema
id	path	Id of the board to be deleted.	Yes	≡ string

Responses

Code	Description
204	success
401	unauthorized
404	not found

[Try this operation](#)

18 pav. Darbalaukio ištrynimasis

4.2 Užduoties objekto API specifikacija

GET /api/board/{id}/task

Description

Gets all board's tasks

Parameters

Name	Located in	Description	Required	Schema
id	path	Id of the board.	Yes	string

Responses

Code	Description	Schema
200	success	<div><div>[</div><div><div>Task {</div><div><div>name: string</div><div>description: string</div><div>status: integer</div><div>weight: integer</div><div>deadline: string</div></div></div><div>]</div></div>
401	unauthorized	
404	not found	

Try this operation

19 pav. Darbalaukio užduočių gavimas

POST /api/board/{id}/task

Description

Posts a new task to the board.

Parameters

Name	Located in	Description	Required	Schema
id	path	Id of the board.	Yes	string
body	body	New task object	Yes	<div><div>Task {</div><div><div>name: string</div><div>description: string</div><div>status: integer</div><div>weight: integer</div><div>deadline: string</div></div></div>

Responses

Code	Description	Schema
201	success	<div><div>Task {</div><div><div>name: string</div><div>description: string</div><div>status: integer</div><div>weight: integer</div><div>deadline: string</div></div></div>
401	unauthorized	

Try this operation

20 pav. Naujos užduoties sukūrimas

GET /api/board/{boardId}/task/{taskId}

Description

Gets a task by id.

Parameters

Name	Located in	Description	Required	Schema
boardId	path	Id of the board.	Yes	≡ string
taskId	path	Id of the task to be retrieved.	Yes	≡ string

Responses

Code	Description	Schema
200	success	<div>≡<pre>Task { name: string description: string status: integer weight: integer deadline: string }</pre></div>
401	unauthorized	
404	not found	

Try this operation

21 pav. Vienos užduoties gavimas

PUT /api/board/{boardId}/task/{taskId}

Description

Updates a task by id.

Parameters

Name	Located in	Description	Required	Schema
boardId	path	Id of the board.	Yes	≡ string
taskId	path	Id of the task to be updated.	Yes	≡ string
body	body	New task object	Yes	≡ <pre>Task { name: string description: string status: integer weight: integer deadline: string }</pre>

Responses

Code	Description
204	success
401	unauthorized
404	not found

Try this operation

22 pav. Užduoties atnaujinimas

DELETE /api/board/{boardId}/task/{taskId}

Description

Deletes a task by id.

Parameters

Name	Located in	Description	Required	Schema
boardId	path	Id of the board.	Yes	≡ string
taskId	path	Id of the task to be updated.	Yes	≡ string

Responses

Code	Description
204	success
401	unauthorized
404	not found

Try this operation

23 pav. Užduoties ištrynimasis

4.3 Vartotojo registracijos API specifikacija

POST /api/register

Description

Creates a new user

Parameters

Name	Located in	Description	Required	Schema
body	body	New board object	Yes	<pre>{ username: string password: string }</pre>

Responses

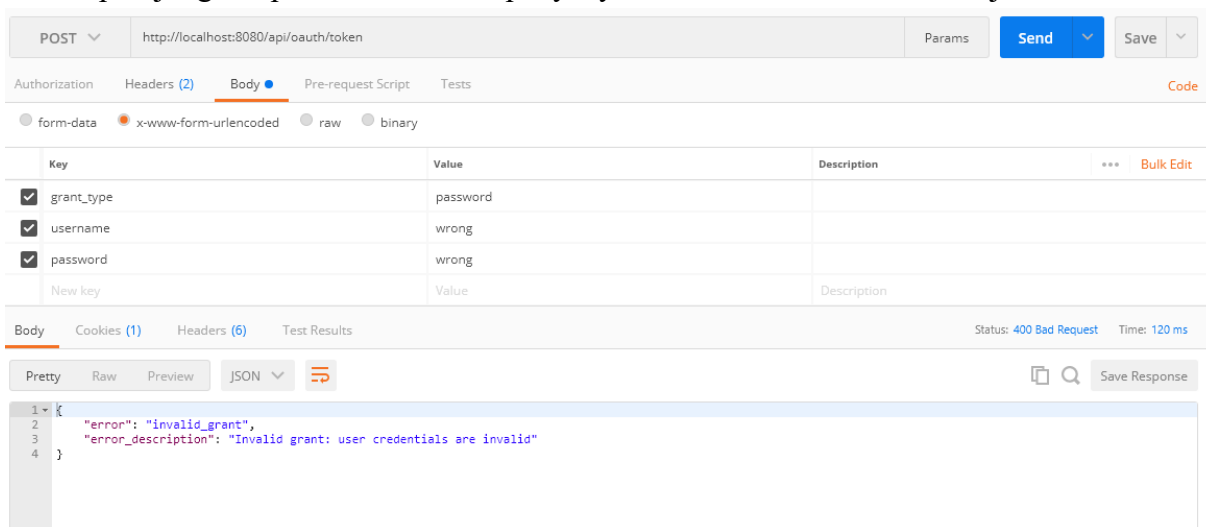
Code	Description	Schema
201	success	<pre>{ username: string password: string }</pre>
401	unauthorized	

Try this operation

24 pav. Vartotojo registracija

5 API panaudojimo pavyzdžiai

Pateikiamas prisijungimo prie sistemos API pavyzdys. API užklausi atlikti naudojamas Postman.



25 pav. Prisijungimas naudojant klaidingus prisijungimo duomenis

POST http://localhost:8080/api/oauth/token

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
grant_type	password	
username	tom	
password	goodpw	
New key	Value	Description

Body Cookies (1) Headers (8) Test Results

Status: 200 OK Time: 100 ms

Pretty Raw Preview JSON

```

1 {
2   "access_token": "9d1ed565e3f02a4e06eacd20dbcd72789c98da96",
3   "token_type": "Bearer",
4   "expires_in": 359999
5 }

```

26 pav. Prisijungimas naudojant teisingus duomenis

Dabar gautą tokeną įdedame į Authorization headerį su priedašu „Bearer“ ir galime kviesti API prašydami kitų sistemos resursų:

GET http://localhost:8080/api/board/

Authorization Headers (2) Body Pre-request Script Tests

Key	Value	Description
Authorization	Bearer 9d1ed565e3f02a4e06eacd20dbcd72789c98da96	
Content-Type	application/json	
New key	Value	Description

Body Cookies (1) Headers (6) Test Results

Status: 200 OK Time: 18 ms

Pretty Raw Preview JSON

```

1 {
2   "result": [
3     {
4       "_id": "5a004056e9983a38d4bab783",
5       "name": "myBoardEdit",
6       "description": "short board description"
7     },
8     {
9       "_id": "5a2e75c99aa68b3584ee180c",
10      "name": "test board",
11      "description": "test board description"
12    },
13    {
14      "_id": "5a2e8c6a5a59f74b5472721d",
15      "name": "dadaw",
16      "description": "wasdasadas\ndasdaw"
17    }
18  ]
19 }

```

27 pav. Gaunamas darbalaukių sąrašas

GET http://localhost:8080/api/board/5a004056e9983a38d4bab783/task Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Code

Key	Value	Description	...	Bulk Edit	Presets
Authorization	Bearer 9d1ed565e3f02a4e06eacd20dbcd72789c98da96				
Content-Type	application/json				
New key	Value	Description			

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 22 ms

Pretty Raw Preview JSON Save Response

```

1 {
2   "result": [
3     {
4       "_id": "5a2c54d26fa6da3954b4ebb2",
5       "name": "weights",
6       "description": "asdqeeweasd",
7       "status": 4,
8       "weight": "3",
9       "boardId": "5a004056e9983a38d4bab783",
10      "deadline": null
11    },
12    {
13      "_id": "5a2c5e184214ec2d14d93728",
14      "name": "urower",
15      "description": "weqeeweas",
16      "status": 1,
17      "weight": "1",
18      "deadline": "2017-12-27",
19      "boardId": "5a004056e9983a38d4bab783"
20    },
21    {
22      "_id": "5a2d5156b84ba03df8c43efa",
23      "name": "daweawe",
24      "description": "erawefsd",
25      "status": 2,
26      "weight": "10",
27      "deadline": "2017-12-10"
28    }
29  ]
30 }

```

28 pav. Gaunamas darbalaukio užduočių sąrašas

GET http://localhost:8080/api/board/5a004056e9983a38d4bab783/task/5a2c54d26fa6da3954b4ebb2 Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Code

Key	Value	Description	...	Bulk Edit	Presets
Authorization	Bearer 9d1ed565e3f02a4e06eacd20dbcd72789c98da96				
Content-Type	application/json				
New key	Value	Description			

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 210 ms

Pretty Raw Preview JSON Save Response

```

1 {
2   "result": {
3     "_id": "5a2c54d26fa6da3954b4ebb2",
4     "name": "weights",
5     "description": "asdqeeweasd",
6     "status": 4,
7     "weight": "3",
8     "boardId": "5a004056e9983a38d4bab783",
9     "deadline": null
10  }
11 }

```

29 pav. Gaunami konkrečios užduoties duomenys

6 Išvados

Laboratorinio darbo metu buvo sukurta REST principais paremtas API, priėjimui prie sistemos duomenų realizuotas OAuth2, taip pat sukurta API panaudojanti vartotojo sąsaja.

NodeJs yra patogus įrankis serverio pusės programavimui, leidžiantis greitai ir paprastai panaudoti kitas bibliotekas bei suteikiantis galimybę ir serverio pusę, ir kliento pusę programuoti JavaScript kalba.

Babel ir Webpack įrankiai leido patogiai transpiliuoti JavaScript kodą pritaikant ją naršyklėms, taip suteikdami galimybę programuoti ES6 versija ir panaudoti naujas JavaScript galimybes.

React ir Redux karkasais buvo suprogramuotas greitai ir sklandžiai veikiantis Single page app, kuris sukuria vartotojui geresnę patirtį naudojant vartotojo sąsają. Buvo panaudoti material-ui React komponentai tam, kad vartotojo sąsaja atitiktų Google Material dizaino gaires.