

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS**

**Programų sistemų inžinerija (P175B015)
“Simpai”**

Atliko:

IFF-1/6 gr. studentai

Daugardas Lukšas

Domantas Bieliūnas

Lukas Kuzmickas

Kasparas Putrius

2023 m. Gegužės 10 d.

Priėmė:

lekt. BARISAS Dominykas

KAUNAS 2023

Turinys

1. Techninė užduotis	3
2. Burn-down grafikai ir retrospektyvos	4
Pirmojo sprinto retrospektyva (2 pav.)	4
Antrojo sprinto retrospektyva (3 pav.)	4
Trečiojo sprinto retrospektyva (4 pav.)	5
Ketvirtojo sprinto retrospektyva (5 pav.)	6
Penktojo sprinto retrospektyva (6 pav.)	7
3. Architektūra	8
4. Testavimas ir jo rezultatai	9
5. Trumpa naudotojo dokumentacija	21
5.1 Paskyros kūrimas	21
5.2 Prisijungimas	24
5.3 Pagrindinis puslapis	24
5.4 Navigacija	26
5.5 Pokalbis	26
5.6 Naudotojo profilis	28
6. Šaltiniai ir nuorodos	32

1. Techninė užduotis

Programų sistemos idėja – internetinė programėlė, kuri leidžia vartotojams susikurti profilį ir ieškoti panašių interesų partnerių. Programėlė veikia panašiu principu, kaip „Tinder“ programėlė. Tik vienas svarbesnių aspektų yra nuotraukos paslėpimas pačioj pradžioje, vartotojas vertina asmens aprašymą, o ne jo išvaizdą. Negalima pamatyti kito žmogaus, kol nėra bendro sutikimo (paspausto mygtuko).

Tik sumatchinus ir abiems paspaudus mygtuką „Show yourself“ jie gali parodyti save. Taip gal ir viena iš svarbesnių socialinių normų, žmogų vertiname pagal, koks jis yra, o ne pagal jo išvaizdos atributus. Vartotojas gali atlikinėti pokalbius su kitais (matched) vartotojais (tekstiniai pokalbiai), yra asmeninės paskyros sukūrimo procesas, prisijungimo sistema, pirmenybės nustatymai ir kita.



2. Burn-down grafikai ir retrospektyvos

Pirmojo sprinto retrospektyva (2 pav.)

- 1) Sprinto metu buvo pridedamos papildomos užduotys – keitėsi „story points“ kiekis – kito sprinto metu geriau susiplanuoti darbus ir nedėti papildomų užduočių.
- 2) Detalesni „user story“ aprašymai – padidina darbo našumą, nes viskas jau suplanuota.
- 3) Kasdieniniai susitikimai arba pranešimai – kasdieninės žinutės „reminderiai“ ką kiekvienos komandos narys daro, kaip sekasi, leidžia produktyviau ir greičiau atlikti užduotis.
- 4) Naujų technologijų integracija – „React“ ir „Typescript“ mokymasis padeda greičiau atlikti užsibrėžtus tikslus.



2 pav. Pirmojo sprinto „Burn-down“ grafikas.

Antrojo sprinto retrospektyva (3 pav.)

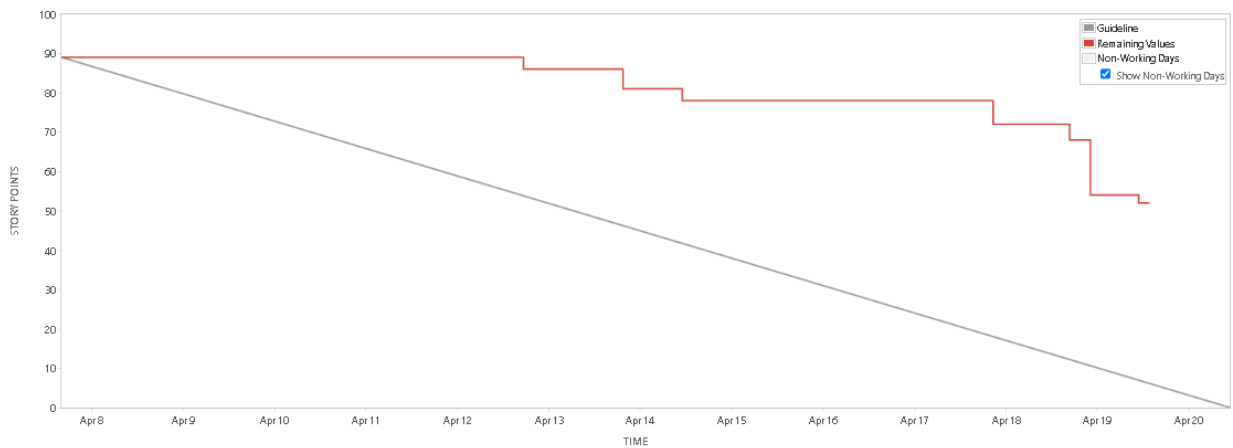
- 1) Nuolatinis darbas – kiekvieną dieną atliekamas bent vienas „commit“ arba „pull request“, dėl tinkamos užduoties, smulkesnių užduočių pasiskirstymas.
- 2) Kasdieniai susitikimai, kodo aptarimas, kodo apžvalga, optimizavimas – vienas komandos narys prieš pakeitimus peržiūri kiti komandos nario kodą, komentuoja, duoda patarimų.
- 3) Darbų susiskirstymas – prieš kiekvieną sprintą kiekvienas komandos narys, bent minimaliai privalo sau priskirti 9 „story points“ (t.y. bent 3 dienas darbo).



3 pav. Antrojo sprinto „Burn-down“ grafikas.

Trečiojo sprinto retrospektyva (4 pav.)

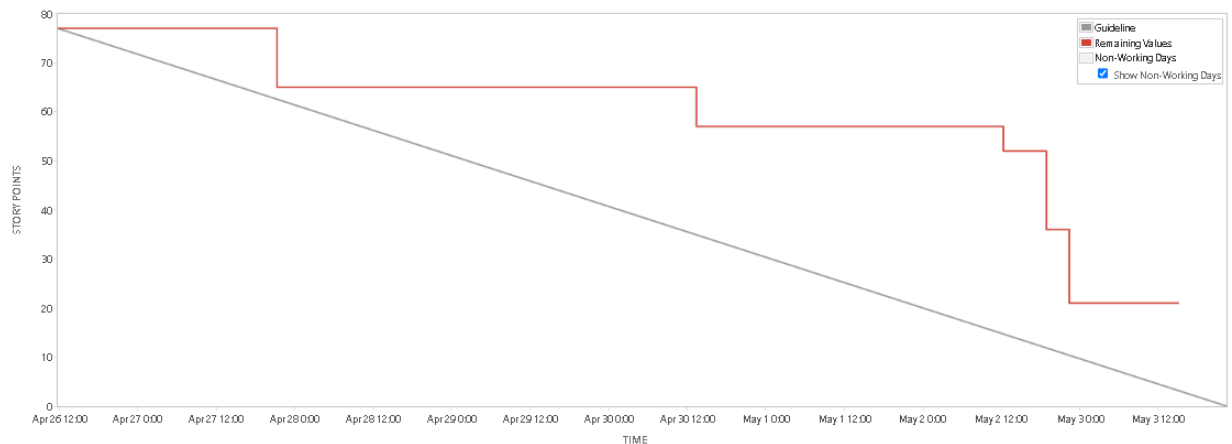
- 1) Kiekvieno sprinto pabaigos retrospektyva – sprinto pabaigoje komandos nariai susijungia trumpam pokalbiui apie sprintą, kaip sekėsi, kas buvo daroma blogai, kaip galima buvo daryti geriau – iš to daromos išvados ir ruošiamasi integruoti kitam sprintui.
- 2) Laiko planavimas – prieš atliekant sprintą, reikia apsvarstyti tokius atvejus, kaip – atostogos, nedarbo dienos, nelaimės ir t.t.



4 pav. Trečiojo sprinto „Burn-down“ grafikas

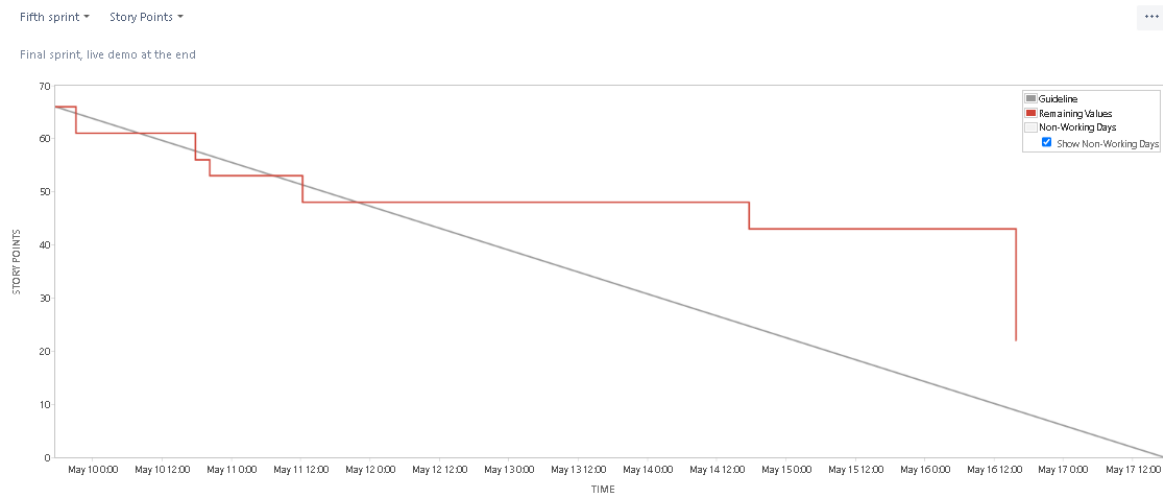
Ketvirtojo sprinto retrospektyva (5 pav.)

- 1) Trukdo kiti moduliai, kitais metais atsižvelgti, dideli projektai.
- 2) Per daug „user story“ užduočių sprintui – komandos nariai nespėja atlikti visų užsibrėžtų užduočių (mūsų grupei tinka tarp 60-70 „story points“ intervalo).
- 3) Daily standup – geriausia alternatyva „Slack“ arba Messenger.



5 pav. Ketvirtojo sprinto „Burn-down“ grafikas

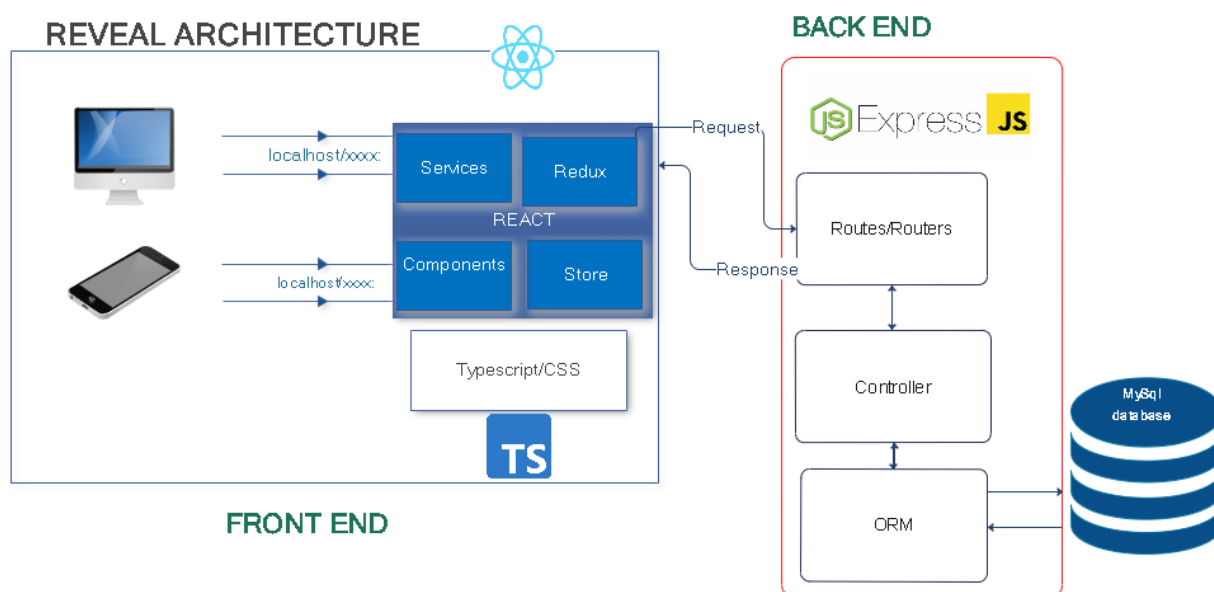
Penktojo sprinto retrospektyva (6 pav.)



6 pav. Penktojo sprinto „Burn-down“ grafikas

- 1) Pervėlus darbas – planuojamas darbas.
- 2) Daily standup – komunikacija.
- 3) Laiko planavimas – geriau susiplanuoti darbai.

3. Architektūra



7 pav. „Reveal“ architektūra.

4. Testavimas ir jo rezultatai

TC – „Front End“ komponentų testavimas.

BC – „Back End“ testavimas.

Šiems testavimo atvejams naudojame „Jest“, jis labai lengvai integruojamas į mūsų React-Typescript projektą.

Testam paleisti užtenka tik parašyti **npm run test (esant client folderyje)** ir **npm run test (esant reveal folderyje)**.

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC01	Button render	Render Button component without crashing	Store: Mock store data Render: component to document	Button component should render without crashing (display code in document).	Render is completed successfully without crashing	Pass
TC02	Button text render	Render Button component with button text = "sample button" and check if it renders in document.	Store: Mock store data Render: component to document with button text	Button component should render in document with text = "sample button"	Render is completed and document has "sample button" text	Pass
TC03	Button component call OnClick()	Render Button component with button text =	Store: Mock store data	Button component should register that function	Button component renders and when clicked	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
	function when clicked	"sample button" and execute an event when button is clicked and check if the function onClick() is called	Render: component Button is rendered with button "sample button", that has a function onClick()	onClick() is called when a click event is called	calls function onClick()	
TC04	Change password component renders without crashing	Render ChangePassword component without crashing	Store: Mock store data Render: component to document	Change password component should render without crashing (display code in document)	Render is completed successfully without crashing	Pass
TC05	Change password button calls the onSubmit() function when clicked	Render ChangePassword component with button text = "Change password" and execute an event when button is clicked and check if the function onClick() is called	Store: Mock store data Render: component ChangePassword is rendered with button "Change password", that has a function onClick()	When button "Change password" is clicked, function onClick() is called	Component ChangePassword renders with correct button and when button click event is executed it calls the onClick() function	Pass
TC06	ChangePassword component renders "old"	Render ChangePassword component with "old password"	Store: Mock store data Render: component ChangePassword is	The rendered document should have "old"	Rendered component ChangePassword has "old"	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
	password" input field	input field and check if it is in the document	rendered with input field "old password"	password" input field	password" input field	
TC07	ChangePassword component renders "new password" input field	Render ChangePassword component with "new password" input field and check if it is in the document	Store: Mock store data Render: component ChangePassword is rendered with input field "new password"	The rendered document should have "new password" input field	Rendered component ChangePassword has "new password" input field	Pass
TC08	Feedback component renders without crashing	Render Feedback component to document without crashing	Store: Mock store data Render: Feedback component	Render should create a Feedback component document without crashing	Render is completed successfully without crashing and creates a Feedback component document	Pass
TC09	Form component renders without crashing	Render Form component to document without crashing	Store: Mock store data Render: Form component	Render should create a Form component document without crashing	Render document is created without crashing	Pass
TC10	Input component renders without crashing	Render Input component to document	Store: Mock store data	Render should create an Input component document	Render document is created without crashing	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
		without crashing	Render: Input component	without crashing		
TC11	Login component renders without crashing	Render Login component to document without crashing	Store: Mock store data Render: Login component	Render should create a Login component document without crashing	Render document is created without crashing	Pass
TC12	Logo component renders without crashing	Render Logo component to document without crashing	Store: Mock store data Render: Logo component	Render should create a Logo component document without crashing	Render document is created without crashing	Pass
TC13	Match component renders without crashing	Render Match component to document without crashing	Store: Mock store data Render: Match component	Render should create a Match component document without crashing	Render document is created without crashing	Pass
TC14	Menu component renders without crashing	Render Menu component to document without crashing	Store: Mock store data Render: Menu component	Render should create a Menu component document without crashing	Render document is created without crashing	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC15	Navigation component renders without crashing	Render Navigation component to document without crashing	Store: Mock store data Render: Navigation component	Render should create a Navigation component document without crashing	Render document is created without crashing	Pass
TC16	PasswordReset component renders without crashing	Render PasswordReset component to document without crashing	Store: Mock store data Render: PasswordReset component	Render should create a PasswordReset component document without crashing	Render document is created without crashing	Pass
TC17	ProfileRegister component renders without crashing	Render ProfileRegister component to document without crashing	Store: Mock store data Render: ProfileRegister component	Render should create a ProfileRegister component document without crashing	Render document is created without crashing	Pass
TC18	Register component renders without crashing	Render Register component to document without crashing	Store: Mock store data Render: Register component	Render should create a Register component document without crashing	Render document is created without crashing	Pass
TC19	TermsAndConditions component	Render TermsAndConditions	Store: Mock store data	Render should create a	Render document is	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
	renders without crashing	component to document without crashing	Render: TermsAndConditions component	TermsAndConditions component document without crashing	created without crashing	
TC20	UserDelete component renders without crashing	Render UserDelete component to document without crashing	Store: Mock store data Render: UserDelete component	Render should create a UserDelete component document without crashing	Render document is created without crashing	Pass
TC21	Feedback component renders the form inputs	Feedback component form elements are correctly displayed	Store: Mock store data Render: Feedback component	Feedback component should render all the input forms	Feedback component renders all the input forms	Pass
TC20	Updates Feedback component state when input forms are changed	Feedback component form elements are updated correctly	Store: Mock store data Render: Feedback component Input: emailInput, categoryDropdown, issueInput	Feedback component should render all the state changes made in the input forms	Feedback component renders all the state changes made in the input forms	Pass
TC21	Renders Form component with correct CSS class name	Render Form component with "form" CSS class name	Store: Mock store data Render: Form component	Form component should render with a "form" CSS class name	Form component renders with a "form" CSS class name	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC22	Renders all the Form component children passed into it	Render Form component with correct children passed into it	Store: Mock store data Render: Form component Input: input child with data id	Form component should render the correct input child with data id	Form component renders the input child component with correct data id	Pass
TC24	Renders ChatBubble component without crashing	Render ChatBubble component without crashing	Store: Mock store data Render: ChatBubble component	ChatBubble component should be rendered to document without crashing	ChatBubble component renders without crashing	Pass
TC25	Renders Envelope component without crashing	Render Envelope component without crashing	Store: Mock store data Render: Envelope component	Envelope component should be rendered to document without crashing	Envelope component renders without crashing	Pass
TC26	Renders Eye component without crashing	Render Eye component without crashing	Store: Mock store data Render: Eye component	Eye component should be rendered to document without crashing	Eye component renders without crashing	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC27	Renders Heart component without crashing	Render Heart component to document without crashing	Store: Mock store data Render: Heart component	Heart component should render a document without crashing	Heart component renders a document without crashing	Pass
TC28	Renders Key component without crashing	Render Key component to document without crashing	Store: Mock store data Render: Key component	Key component should render a document without crashing	Key component renders a document without crashing	Pass
TC29	Renders User component without crashing	Render User component to document without crashing	Store: Mock store data Render: User component	User component should render a document without crashing	User component renders a document without crashing	Pass
TC30	Renders XMark component without crashing	Render XMark component to document without crashing	Store: Mock store data Render: XMark component	XMark component should render a document without crashing	XMark component renders a document without crashing	Pass
TC31	Renders InputErrors component without crashing	Render InputErrors component to document	Store: Mock store data Render: InputErrors component	InputErrors component should render a document	InputErrors component renders a document without crashing	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
		without crashing		without crashing		
TC32	Input component can render an Icon	Render Input component with an Icon	Store: Mock store data Render: Input component with an Icon	Input component should render with unique id icon in document	Input component renders with unique id icon in document	Pass
TC33	Input component can render errors	Render Input component with errors	Store: Mock store data Render: Input component with errors array	Input component should render with errors array	Input component renders with errors array	Pass
TC34	Login component email input field changes email state when text is entered	Render Login component with changed email input field	Store: Mock store data Render: Login component with changed email input state	Login component should render with changed email input state	Login component renders with changed email input state	Pass
TC35	Login component password input field changes password state when text is entered	Render Login component with changed password input field	Store: Mock store data Render: Login component with changed password input state	Login component should render with changed password input state	Login component renders with changed password input state	Pass
TC36	OnSubmit function is called when Login	Render Login component	Store: Mock store data	Login component should call	Login component calls the OnSubmit	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
	component button is clicked	with OnSubmit button and simulate a button click event to call OnSubmit function	Render: Login component	the OnSubmit function when the Login button is clicked	function when the Login button is clicked	
TC37	PasswordReset component email input field changes state correctly	Render PasswordReset component with changed email input field state	Store: Mock store data Render: PasswordReset component with changed email input field state	PasswordReset component should render with changed email input field state	PasswordReset component renders with changed email input field state	Pass
TC38	ProfileRegister component adds interests when "Add Interests" button is clicked	Render ProfileRegister component with added interests	Store: Mock store data Render: ProfileRegister component	ProfileRegister component should have added interests with simulated "Add Interests" button click	ProfileRegister component renders with added interests	Pass
TC39	Submits ProfileRegister component form successfully	Render ProfileRegister component form successfully	Store: Mock store data Render: ProfileRegister component	ProfileRegister component should have all filled out form data in document rendered out	ProfileRegister component renders with all form data	Pass
TC40	Should disable Register component	Render Register component	Store: Mock store data	Register component should have	Register component	Pass

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
	register button initially	with register button initially disabled	Render: Register component	register button disabled initially	register button is disabled	
TC41	TOS component displays correct list items	Render TOS component with correct list items	Store: Mock store data Render: TOS component	TOS component should render with correct list items	TOS component renders with correct list items	Pass
BC01						Pass
BC02						Pass
BC03						Pass
BC04						Pass

	Quarantena	Blauznomės	Funkcijos	Užrašai
src	0%	0/22	0%	0/10
src/components/BiggerInput	88.42%	13/19	87.89%	15/28
src/components/Button	100%	2/2	100%	2/2
src/components/ChangePassword	62.5%	10/16	50%	1/2
src/components/Chats	0%	0/30	0%	0/15
src/components/Feedback	92.3%	12/13	100%	2/2
src/components/Form	100%	2/2	100%	0/0
src/components/Icons	100%	14/14	50%	3/6
src/components/Input	84.21%	18/19	79.16%	19/24
src/components/Login	73.65%	14/19	50%	3/6
src/components/Logo	100%	1/1	100%	0/0
src/components/Match	82.6%	19/23	66.66%	6/9
src/components/Menu	100%	1/1	100%	0/0
src/components/Navigation	100%	5/5	100%	2/2
src/components>PasswordReset	75%	6/8	100%	0/0
src/components/ProfileRegister	95.15%	25/26	70%	7/10
src/components/Register	39.06%	25/64	32.81%	21/64
src/components/TermsAndConditions	100%	1/1	100%	0/0
src/components/UserDelete	82.5%	10/19	75%	3/4
src/models	0%	0/0	0%	0/0
src/services	29.82%	17/57	0%	0/4
src/store	66.66%	8/12	100%	0/0
src/store/auth	18.96%	11/58	0%	0/2
src/store/changePassword	33.33%	7/21	0%	0/4
src/store/chats	25%	7/28	100%	0/0
src/store/feedback	11.11%	5/45	0%	0/0

8 pav. Testų padengimas

```

Test Suites: 24 passed, 24 total
Tests:       47 passed, 47 total
Snapshots:   0 total
Time:        13.906 s
Ran all test suites.

```

9 pav. TC testų rezultatai

5. Trumpa naudotojo dokumentacija



Sveiki atvykę į Reveal - internetinę svetainę, kurioje galite susipažinti su kitais žmonėmis be nuotraukų! Naudojantis Reveal, galite užmegzti reikšmingus santykius su kitais pagal savo asmenybę ir interesus.

Čia pateikiame viską, ko jums reikia norint pradėti:

5.1 Paskyros kūrimas

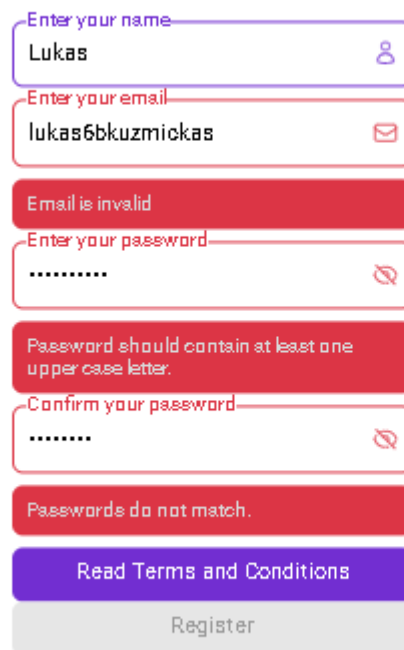
Norėdami naudoti Reveal, turite sukurti paskyrą. Tam yra skirtas Register puslapis. Čia registruojate savo paskyrą mūsų duombazėje.

The image shows a registration form with the following elements:

- Input field: "Enter your name" with a person icon.
- Input field: "Enter your email" with an envelope icon.
- Input field: "Enter your password" with an eye icon (password visibility toggle).
- Input field: "Confirm your password" with an eye icon (password visibility toggle).
- Button: "Read Terms and Conditions" (purple background).
- Button: "Register" (gray background).

9 pav. Paskyros kūrimo formą

Formą pildome, pagal tam tikrus reikalavimus, mūsų elektroninis paštas, turi būti teisingas, slaptažodis turi būti stiprus ir turi sutapti su patvirtintu.



Enter your name
Lukas

Enter your email
lukas6bkuzmickas

Email is invalid

Enter your password
.....

Password should contain at least one upper case letter.

Confirm your password
.....

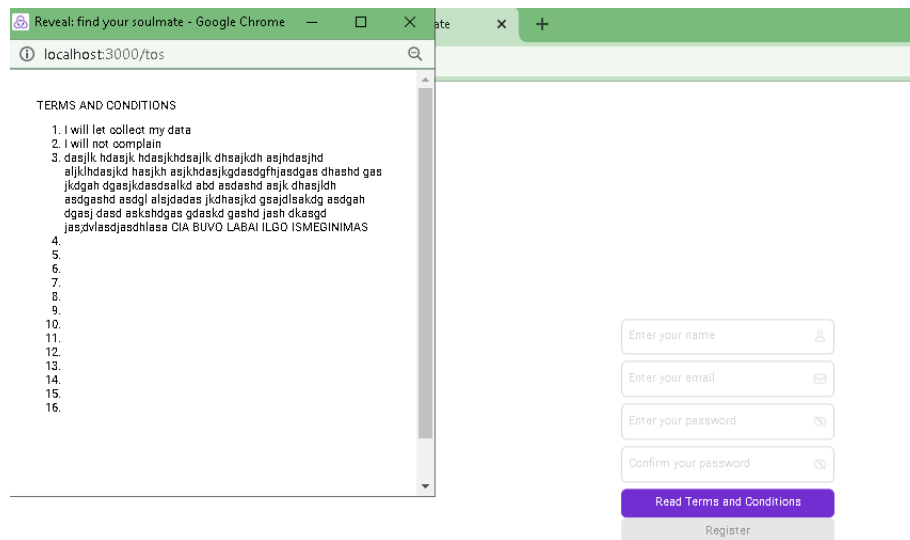
Passwords do not match.

Read Terms and Conditions

Register

10 pav. Įvedimo laukai

Užpildę visą reikalingą informaciją, privalome paskaityti TOS (Terms and Conditions), kitaip negalime užbaigti paskyros kūrimo.



Reveal: find your soulmate - Google Chrome

localhost:3000/tos

TERMS AND CONDITIONS

1. I will let collect my data
2. I will not complain
3. dasjlk hdsajk hdsajkhdsajlk dhsajkdh asjhdsajhd
aljkldhsajkd hasjkh asjkhdsajkgsdsgfhjasdghs dhasgd gas
jkdgh dgasjkdsdsalkd abd asdashd asjk dhasjldh
asdgashd asdgl alsjdadas jkdhasjkd gsajdsakdg asdgah
dgasj dased askshogas qdaskd gashd jash dksagd
jas,dviasdjasdhlasa CIA BUVO LABAI ILGO ISMEGINIMAS
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
- 12.
- 13.
- 14.
- 15.
- 16.

Enter your name

Enter your email

Enter your password

Confirm your password

Read Terms and Conditions

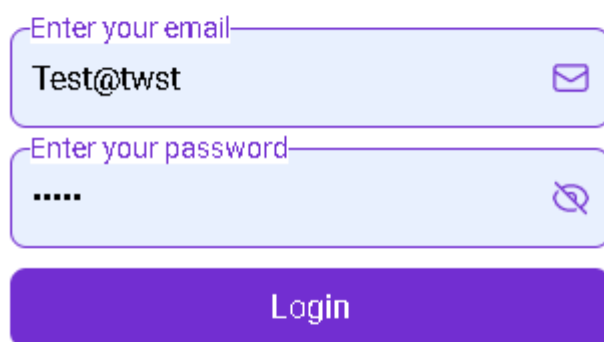
Register

11 pav. TOS laukas

Perskaite TOS (Terms and Conditions) galime užbaigti mūsų paskyros kūrimą.

5.2 Prisijungimas

Vartotojas, kuris jau yra susikūręs paskyrą, gali prisijungti per Login puslapį. Login puslapyje įvedame savo elektroninį paštą ir slaptažodį, su kuriuo kūrėmės paskyrą.

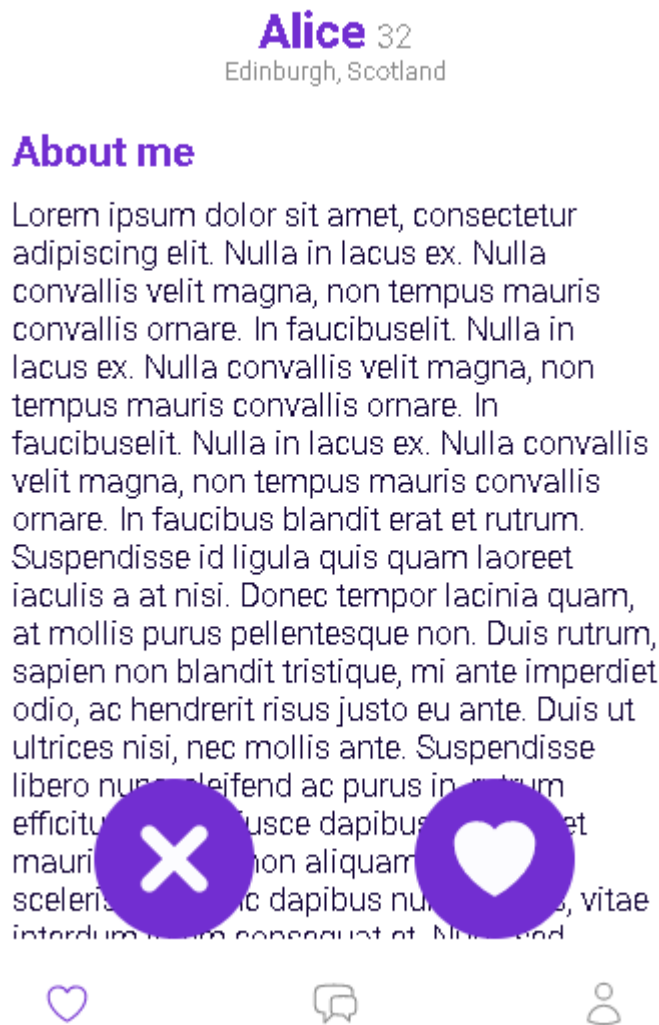


The image shows a login form with two input fields and a button. The first field is labeled 'Enter your email' and contains the text 'Test@twst'. The second field is labeled 'Enter your password' and contains five dots. Both fields have a small icon on the right: an envelope for email and a key with a slash for password. Below the fields is a blue button labeled 'Login'.

12 pav. Prisijungimas

5.3 Pagrindinis puslapis

Čia Reveal skiriasi, nuo kitų programėlių kaip Tinder, nedelsdami nematysite kito žmogaus nuotraukos. Vietoj to matysite šito žmogaus trumpą aprašymą ir du mygtukus „Like“ ir „Dislike“, kuriuos atitinkamai galite paspausti, jeigu norite su šiuo žmogumi „Matchinti“.



13 pav. Pagrindinis puslapis

Šiame puslapyje matome kito žmogaus trumpą aprašymą, pomėgius, amžių, vardą.



14 pav. Aprašymas

Turime du mygtukus (X) ir (♥).

(X) mygtukas – nenorime „Matchinti“ su pasirinktu žmogumi.

(♥) – mygtukas – norime „Matchinti“ su pasirinktu žmogumi.



15 pav. Mygtukai

Atitinkamai spausdami šiuos mygtukus, renkame mūsų „Matchus“, su kuriais vėliau galėsime bendrauti per mūsų „Chat“ puslapį.

5.4 Navigacija

Jau prisijungę prie Reveal. Galime naviguoti mūsų pasirinktais puslapiais.

Turime tris puslapius ir jų paskirtis –

- Main page komponentas – čia galime matyti savo „Matchus“ ir juos rinktis.
- Chat page komponentas – čia galime matyti savo pokalbius su kitais „Matchintais“ naudotojais, redaguoti savo pokalbio istorijas, siuntinėti žinutes tarp naudotojų.
- Naudotojo (user) profilio page komponentas – čia galime keisti savo profilio informaciją, slaptažodį, atsijungti nuo mūsų duotos paskyros.



16 pav. Navigacijos meniu

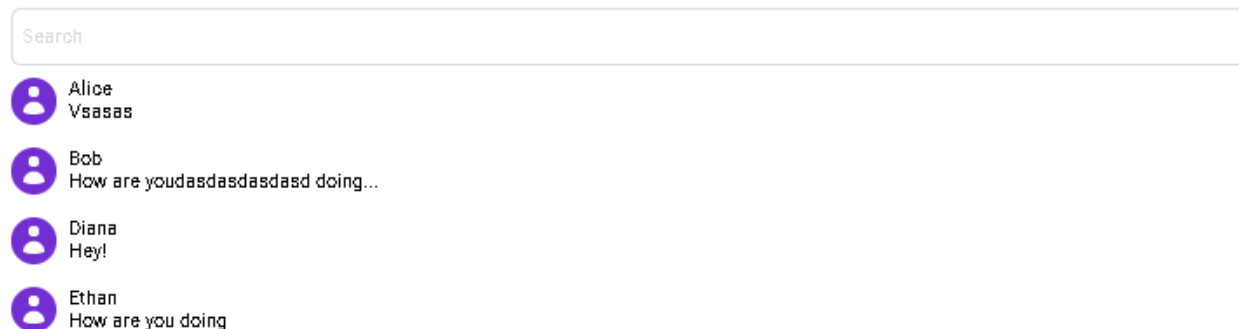
5.5 Pokalbis

Jei abu „Matchinate“ save, galėsite bendrauti vienas su kitu. Galite bendrauti, kiek norite, ir jei susitariate, netgi galite susitikti asmeniškai!

Matches

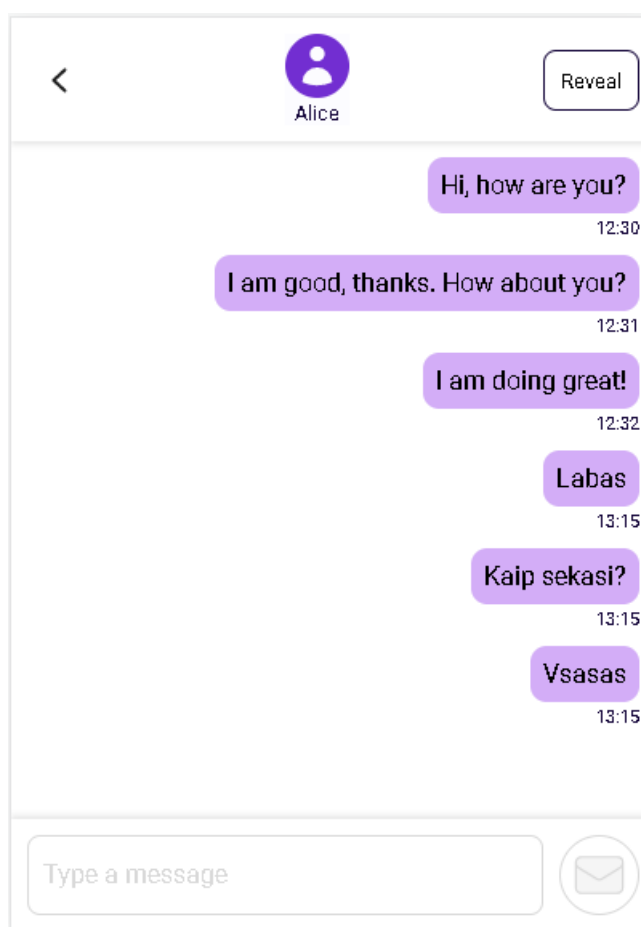


Chats



17 pav. Chat komponentas

Šiame puslapyje galime bendrauti su mūsų „Matchintais“ naudotojais.

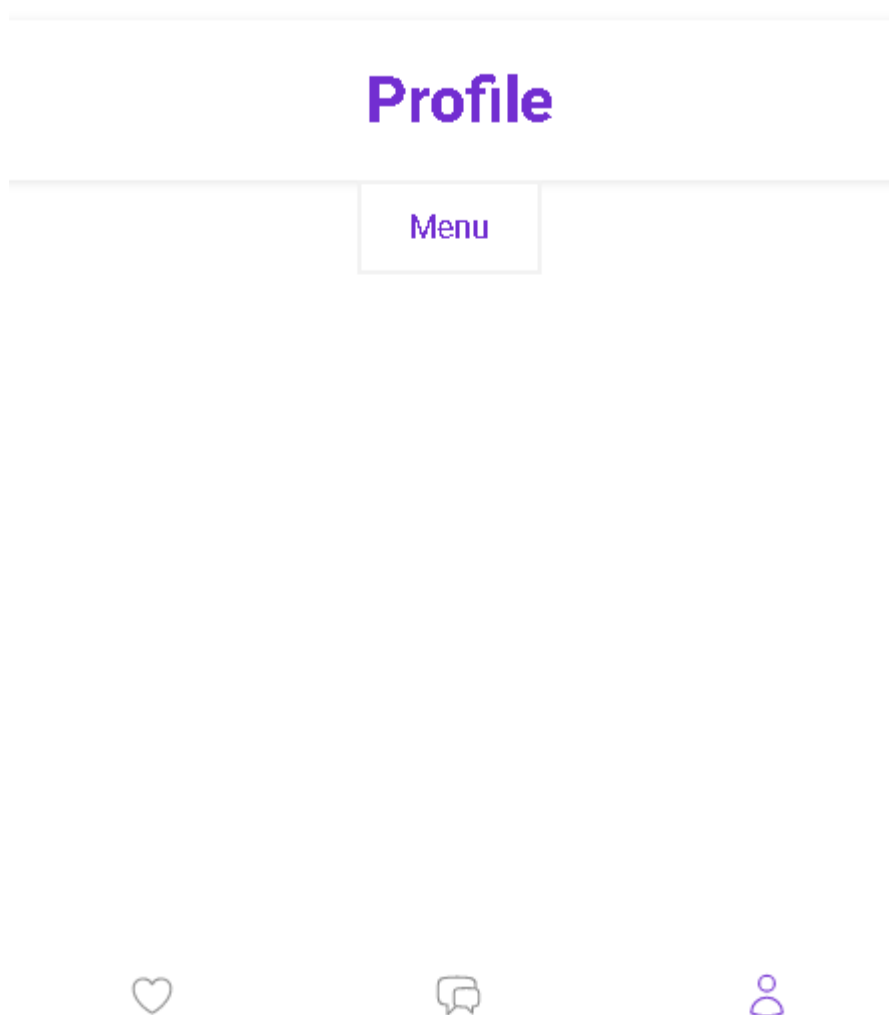


18 pav. Bendravimas

Abiems naudotojams paspaudus „Reveal” mygtuką jie gali vienas kitą pamatyti.

5.6 Naudotojo profilis

Čia galime pamatyti mūsų profilį. Galime pasikeisti mūsų naudojamą slaptažodį, ištrinti mūsų paskyrą, redaguoti profilio informaciją – interesai, amžius, pomėgiai ir t.t.

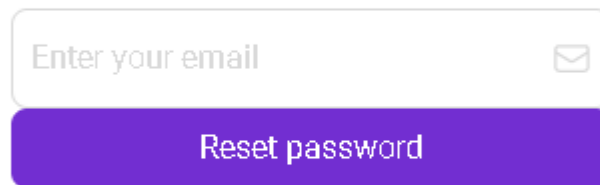


19 pav. Naudotojo profilis.

Čia galime pamatyti mūsų profilį. Turime meniu komponentą, kuriame turime 3 narius:

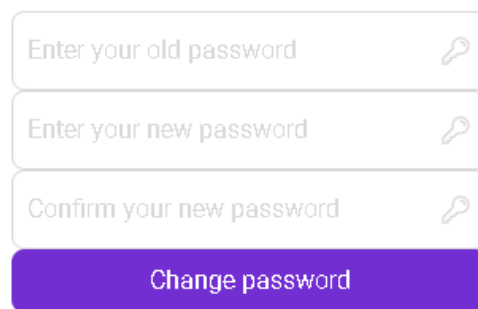
- Reset Password komponentas – čia galime pasikeisti savo slaptažodį – šitas puslapis yra kviečiamas, kai pamirštame savo slaptažodį „Login“ metu.

Čia suvedame savo registracijos metu naudota e. pašto adresą ir gauname tam tikras patvirtinimo žinutes, taip galime pasikeisti savo slaptažodį.

A screenshot of a web form for resetting a password. It features a light gray rounded rectangular input field with the placeholder text "Enter your email" and a small envelope icon on the right. Below the input field is a solid purple rectangular button with the text "Reset password" in white.

20 pav. Reset password

- Change Password komponentas – čia galime pasikeisti savo slaptažodį jau būnant prisijungus prie programėlės, užtenka tik žinoti savo buvusi slaptažodį ir susikurti naują.



A screenshot of a 'Change password' form. It consists of three stacked input fields, each with a key icon on the right. The first field is labeled 'Enter your old password', the second 'Enter your new password', and the third 'Confirm your new password'. Below these fields is a solid purple button with the text 'Change password' in white.



21 pav. Change password.

- Delete User komponentas – čia galime ištrinti savo registruotą paskyrą iš mūsų duomenų bazės. Po ištrinimo iškart nukreipiama į registracijos puslapį.

Delete User

Warning: This action is irreversible and will permanently delete the user's data.

Delete user data



22 pav. Delete User.

6. Šaltiniai ir nuorodos

Nuorodą į šio projekto repozitoriją: <https://bitbucket.org/psi-simpai-2023/reveal/src/dev/>

1. Documentation - React. In *TypeScript: Documentation - React* [interaktyvus]. Prieiga per internetą: <<https://www.typescriptlang.org/docs/handbook/react.html>>.
2. Getting Started with Redux | Redux. In *Getting Started with Redux / Redux* [interaktyvus]. 2022. Prieiga per internetą: <<https://redux.js.org/introduction/getting-started>>.