



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Karolis Strazdas

2D PROGRAMAVIMO ŽAIDIMAS

Projektas

Dėstytojas

Doc. dr. Sigitas Drąsutis

KAUNAS, 2018

Turinio lentelė

Terminų ir santraukų žodynas	3
Įvadas	4
1. Analizė	5
1.1. Esamų sprendimų analizė	5
1.1.1. Screeps.....	5
1.1.2. Robocode	6
1.1.3. Esamų sistemų palyginimas.....	7
1.1. Technologijų analizė.....	8
2. Projektas.....	10
2.1. Funkciniai reikalavimai sistemai	10
2.2. Nefunkciniai reikalavimai sistemai	10
2.3. Duomenų bazės projektas	11
2.4. Konteksto diagrama.....	12
2.6. UML diagramos.....	13
2.6.1. Panaudos atvejų diagrama.....	13
2.6.2. Veiklos diagramos	14
2.6.3. Sekų diagramos	25
3. Vartotojo vadovas	32
4. Išvados	41
5. Literatūros sąrašas.....	42
6. Priedas.....	43
6.1. Semestro darbų suvestinė	43

Terminų ir santraukų žodynas

HTML	Hiperteksto žymėjimo kalba – kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete (angl. <i>Hypertext Markup Language</i>).
Web Assembly	Instrukcijų formatas, į kurį sukompiliuotą C, C++ arba Rust kalbomis parašytą kodą galima vykdyti naršyklėje.
API	Aplikacijų programavimo sąsaja, apibrėžianti programuotojui viešai pasiekiamas funkcijas, klases ir metodus.
BSON	Binary JSON – tokiu formatu MongoDB duomenų bazėje saugomi JSON dokumentai.
CommonJS	Kodo skaidymo modulių formatas, naudojamas NodeJS. Leidžia organizuoti JavaScript kodą į atskirus blokus atskiruose failuose, o taip pat įtraukti šiuos atskirus modulių į vieną failą.
WebGL	JavaScript API, skirtas interaktyvios 2D ir 3D grafikos vaizdavimui interneto naršyklėje.

Įvadas

Kuriamo žaidimo idėja yra labai panaši į jau egzistuojantį projektą [Robocode](#), kurio tikslas taip pat yra suprogramuotų robotų kovos tinkle. Vis dėlto, Robocode neveikia naršyklėje – pastarąjį reikia įsidiegti į kompiuterį, o pati programa pasižymi gana nepatogia ir neišvaizdžia grafinė sąsaja. Taigi, šis darbas yra daromas norint realizuoti Robocode esančią idėją turintį žaidimą, kuris būtų prieinamas per naršyklę nepriklausomai nuo naudojamos operacinės sistemos ar įrenginio.

Darbo tikslas – sukurti robotų-tankų 2D programavimo žaidimą

Darbo uždaviniai:

1. Išanalizuoti esamus sprendimus
2. Išanalizuoti technologijas, kurios bus naudojamos realizuojant šį darbą
3. Suprojektuoti sistemą
4. Realizuoti ir ištestuoti sistemą
5. Dokumentuoti sistemą
6. Dokumentuoti žaidimo API, kuria naudosis žaidėjas

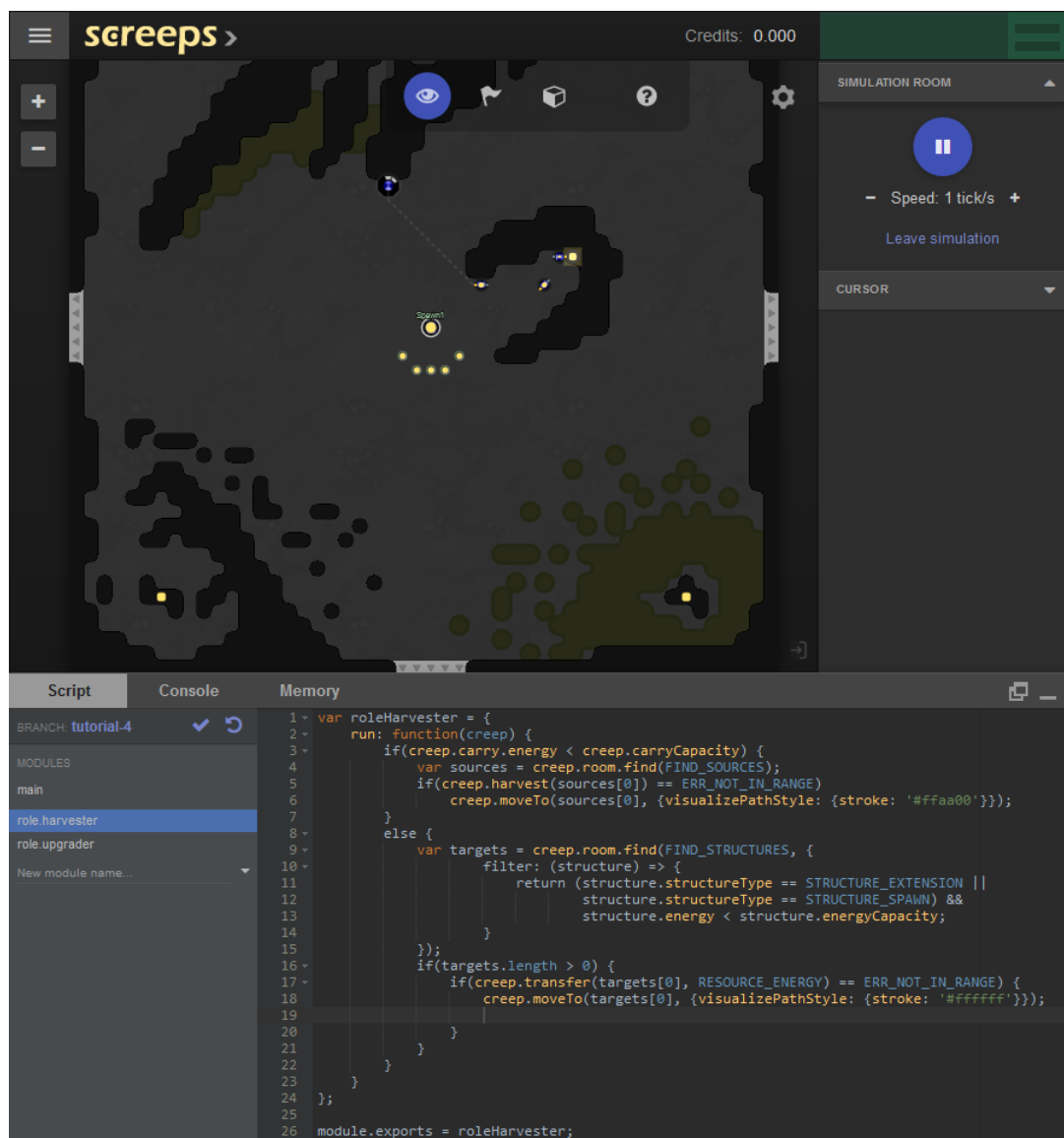
1. Analizė

1.1. Esamų sprendimų analizė

Šiame skyriuje bus analizuojami internete jau egzistuojantys programavimo žaidimai bei lyginamas jų funkcionalumas su realizuojamu darbu. (žr. Lentelė 1.1)

1.1.1. Screeps

Vienas populiariausių internete egzistuojančių programavimo žaidimų yra Screeps. Tai daugelio žaidėjų realaus laiko programavimo žaidimas, kuriame žaidėjas JavaScript programavimo kalba turi suprogramuoti robotų koloniją, kuri žemėlapyje rinktų resursus, užimtų naujas teritorijas, kovotų su kitais žaidėjais bei statytų gynybines struktūras [1]. Robotams parašytas kodas gali būti skaidomas **CommonJS** moduliais ir yra nepertraukiamai vykdomas realiu laiku, taigi pagrindinis žaidimo tikslas yra robotų kolonijos automatizavimas. (žr. Pav. 1.1)



Pav. 1.1 Daugelio žaidėjų programavimo žaidimas Screeps

Sistemos privalumai:

- Parašytas kodas vykdomas nepertraukiamai net ir neprisijungus žaidėjui
- Programavimas nelimituotas tik JavaScript kalba. Galima Naudoti C, C++ arba Rust programavimo kalbas naudojant **WebAssembly** modulius
- Programuoti galima ne tik kodo redaktoriuje, bet ir integruotoje konsolėje
- Išsami žaidimo **API** dokumentacija

Sistemos trūkumai:

- Žaidimas yra mokamas – kodo vykdymui reikia pirkti CPU laiką bei atmintį
- Gana lėtas žaidimo žemėlapių ir objektų jame vaizdavimas (angl. *rendering*)
- Kadangi visa logika yra vykdoma serveryje, žaidimo būsena ekrane atvaizduojama su tam tikru delsimu ir atnaujinimai yra pastebimai nenuoseklūs

1.1.2. Robocode

Robocode nėra naujas projektas - buvo pradėtas kurti dar 2000-aisiais metais. Žaidimo tikslas – programuoti robotus, kurie kovotų prieš kitų žaidėjų robotus [2]. Taip pat yra galimybė varžytis su kitais žaidėjais tinkle. Programuojama Java kalba integruotame Robocode redaktoriuje arba C#, naudojant integruotą kūrimo aplinką, kaip, pavyzdžiui, Visual Studio. Vis dėlto, Robocode reikia atsisiųsti į kompiuterį, kadangi pastarasis neveikia naršyklėje. (žr. Pav. 1.2)



Pav. 1.2 Robotų-tankų programavimo žaidimas Robocode

Sistemos privalumai:

- Nemokamas atviro kodo projektas
- Galima pačiam susikurti lokalų serverį norint varžytis su kitais žaidėjais
- Išsami žaidimo **API** dokumentacija

Sistemos trūkumai:

- Gana neišvaizdi grafinė sąsaja
- Neveikia naršyklėje – reikia atsisiųsti ir įsidiegti į kompiuterį

1.1.3. Esamų sistemų palyginimas

Kriterijus	Screeps	Robocode	Kuriama sistema
Robotų programavimas	Programuojama JavaScript skaidant kodą į CommonJS modulius	Programuojama Java ir C#	Bus galima programuoti tik JavaScript
Robotų kovų tipai	Kovojančių žaidėjų robotų skaičius neribojamas dėl iš esmės skirtingo žaidimo stiliaus	Vienas prieš vieną arba komandomis(2 prieš 2, 5 prieš 5)	Tik vienas prieš vieną
Žaidimo veikimas	Veikia naršyklėje ir kaip įdiegiamas klientas per Steam platformą	Veikia tik kaip atskira programa	Veiks tik naršyklėje
Žaidėjo profilio informacija	Rodoma informacija apie žaidėjo pasiekimus – sukurtų robotų skaičių, užimtos teritorijos kiekį ir pan.	Rodomi žaidėjo laimėtų kovų statistika lyderių lentoje	Bus renkami „patirties taškai“ pagal laimėtų kovų skaičių. Taip pat atrakinami laimėjimai(angl. <i>achievements</i>)
Žaidimas tinkle	Vyksta realiu laiku nepertraukiamai	Kuriamas lokalus serveris arba roboto kodas įkeliamas į oficialų Robocode serverį	Varžovų bus galima ieškoti tiesiogiai, o suradus – kova žaidėjui atvaizduojama ekrane realiu laiku

Lentelė 1.1 Egzistuojančių projektų bei realizuojamo darbo funkcionalumo palyginimas

1.1. Technologijų analizė

Handlebars

Šablonų variklis skirtas dinaminių HTML puslapių generavimui serveryje. Handlebars šablonai yra paprastas HTML kodas, tarp kurio yra įterpiamos specifinės Handlebars žymės [3]. Pustlapis sukompiuojamas funkcijai kaip parametą paduodant Handlebars šablono kodą bei JavaScript objektus, kuriuose esančius duomenis norima atvaizduoti puslapyje.

Privalumai

- Turi galimybę skaidyti HTML kodą į dalis (angl. *partials*) ir įterpti į kitus puslapius
- Sąlyginių sakinių bei ciklų palaikymas šablonuose

Trūkumai

- Šablonuose nepalaiko jokios logikos išskyrus tikrinimą, ar nurodytas JavaScript objektas egzistuoja
- Logikai šablonuose reikia rašyti atskiras Handlebars funkcijas „pagalbininkus“ (angl. *helpers*)

NodeJS

Tai – asinchroninė JavaScript vykdymo aplinka, skirta kurti tinklo aplikacijoms [4]. Taigi, naudojant NodeJS web serveriui aprašyti yra naudojama JavaScript programavimo kalba. Šiuo įrankiu kuriant tinklo projektus įprastai yra naudojami ir **NPM** (angl. *Node Package Manager*), kuris yra įdiegiamas kartu su NodeJS, paketai. Per NPM įdiegiami paketai – tam tikrą funkcionalumą suteikiantys JavaScript moduliai, kaip, pavyzdžiui, **express** HTTP maršrutų aprašymui, **bcrypt** slaptažodžių šifravimui ir pan.

PixiJS

Biblioteka, skirta interaktyvios grafikos vaizdavimui. Tai yra greitas 2D grafikos vaizdavimo variklis skirtas atvaizduoti, animuoti bei valdyti dvimačius grafinius objektus (angl. *sprites*) naršyklėje. PixiJS nėra konkrečiai naršyklinių žaidimų kūrimo įrankis ir yra skirtas interaktyviems internetiniams projektams. [5]

Privalumai

- **WebGL** 2D palaikymas
- **HTML 5 <canvas>** palaikymas jeigu naršyklė nepalaiko WebGL
- Nesunkus animacijų kūrimas

Trūkumai

- Nėra kolizijų nustatymo(angl. *collision detection*) tarp objektų
- Skirtas tik dvimatei grafikai

MongoDB

Nereliacinė duomenų bazė, sauganti duomenis **BSON** dokumentų formatu. Šios duomenų bazės dokumentai yra grupuojami į kolekcijas, kurių ekvivalentas reliacinėje duomenų bazėje būtų lentelė. Vis dėlto, kolekcijoms nėra apibrėžiama, kokius dokumentus su kokiais ir kiek duomenų laukų galima saugoti. Čia duomenų laukai yra saugomi kaip rakto-reikšmės poros. Be įvairių duomenų tipų palaikymo, dokumentų duomenų laukuose galima saugoti masyvus, kitus dokumentus arba dokumentų masyvus. Duomenų saugojimas MongoDB duomenų bazėje yra gana dinamiškas, kadangi toje pačioje kolekcijoje dokumentai gali skirtis duomenų tipais bei laukų skaičiumi. [6]

Privalumai

- Lankstus duomenų saugojimas
- Esant dideliame duomenų kiekiui galimas duomenų paskirstymas per keletą serverių (angl. *sharding*) [7]
- Dokumentuose galima saugoti nuorodas į kitus dokumentus, šitaip imituojant duomenų ryšius. Vis dėlto, tai nėra ekvivalentas ryšiams tarp lentelių reliacinėse duomenų bazėse

Trūkumai

- Duomenys, palyginti su reliacinėmis duomenų bazėmis, įprastai užima daugiau vietos, kadangi saugomi dokumentų duomenų laukų pavadinimai
- Užklausos duomenų paėmimui iš skirtingų dokumentų yra kiek sudėtingesnės, nei tradicinėse reliacinėse duomenų bazėse, kadangi nėra JOIN operatoriaus. Vis dėlto, MongoDB turi tam skirtas agregatines (angl. *aggregation*) funkcijas.

2. Projektas

2.1. Funkciniai reikalavimai sistemai

1. Sukurti žaidimo sesiją žaidimui tarp kelių žaidėjų
2. Sukurti naują JavaScript scenarijų
3. Redaguoti sukurta JavaScript scenarijų
4. Ištrinti sukurta JavaScript scenarijų
5. Suteikti galimybę naudotojui registruotis puslapyje
6. Suteikti galimybę užsiregistravusiam naudotojui prisijungti prie sistemos
7. Suteikti galimybę naudotojui kodą rašyti integruotame kodo redaktoriuje
8. Atkurti pamirštą paskyros slaptažodį
9. Keisti paskyros slaptažodį
10. Vykdyti žaidėjo parašytą kodą ir atvaizduoti kodo atliekamus veiksmus žaidimo scenoje realiu laiku
11. Suteikti galimybę žaidėjui vykdyti parašytą kodą simuliacijoje prieš save ar kitą parašytą scenarijų
12. Rodyti žaidėjo statistiką žaidėjo profilio puslapyje, kuriame pateikiama informacija apie žaidėjo laimėtas / pralaimėtas kovas bei atrakintus laimėjimus
13. Rodyti lyderių lentos puslapį, rodantį surikiuotą visų žaidėjų sąrašą pagal laimėtų kovų skaičių
14. Atskirame puslapyje rodyti žaidėjo atrakintus bei neatrakintus laimėjimus(angl. *achievements*)
15. Administratoriui rodyti užsiregistravusius sistemos naudotojus
16. Leisti administratoriui ištrinti užsiregistravusį naudotoją

2.2. Nefunkciniai reikalavimai sistemai

1. Žaidėjo aprašyto scenarijaus vykdymas serveryje turi būti izoliuotas ir vykdomas atskirame JavaScript kontekste
2. Žaidimo veiksmai, atvaizduojami ekrane, neturi priklausyti vaizdo atnaujinimo dažnio(kadrų per sekundę skaičiaus)
3. Žaidimo būsenos atnaujinimai žaidėjui iš serverio siunčiami naudojant **WebSocket** protokolą
4. Žaidimo vaizdas ekrane turi būti atnaujinamas ≥ 24 kartus per sekundę
5. Žaidėjas gali sukurti ne daugiau kaip 5 skirtingus scenarijus
6. Vieno aprašyto scenarijaus dydis diske neturi viršyti 32KB
7. Žaidėjo parašytas kodas yra saugomas **MongoDB** duomenų bazės kolekcijoje
8. Kiekviena nauja žaidimo sesija kovoms tinkle yra sukurama tik tarp dviejų žaidėjų(vienas prieš vieną)
9. Žaidėjo parašytas kodas yra vykdomas tik serveryje
10. Logika robotui turi būti rašoma JavaScript programavimo kalba

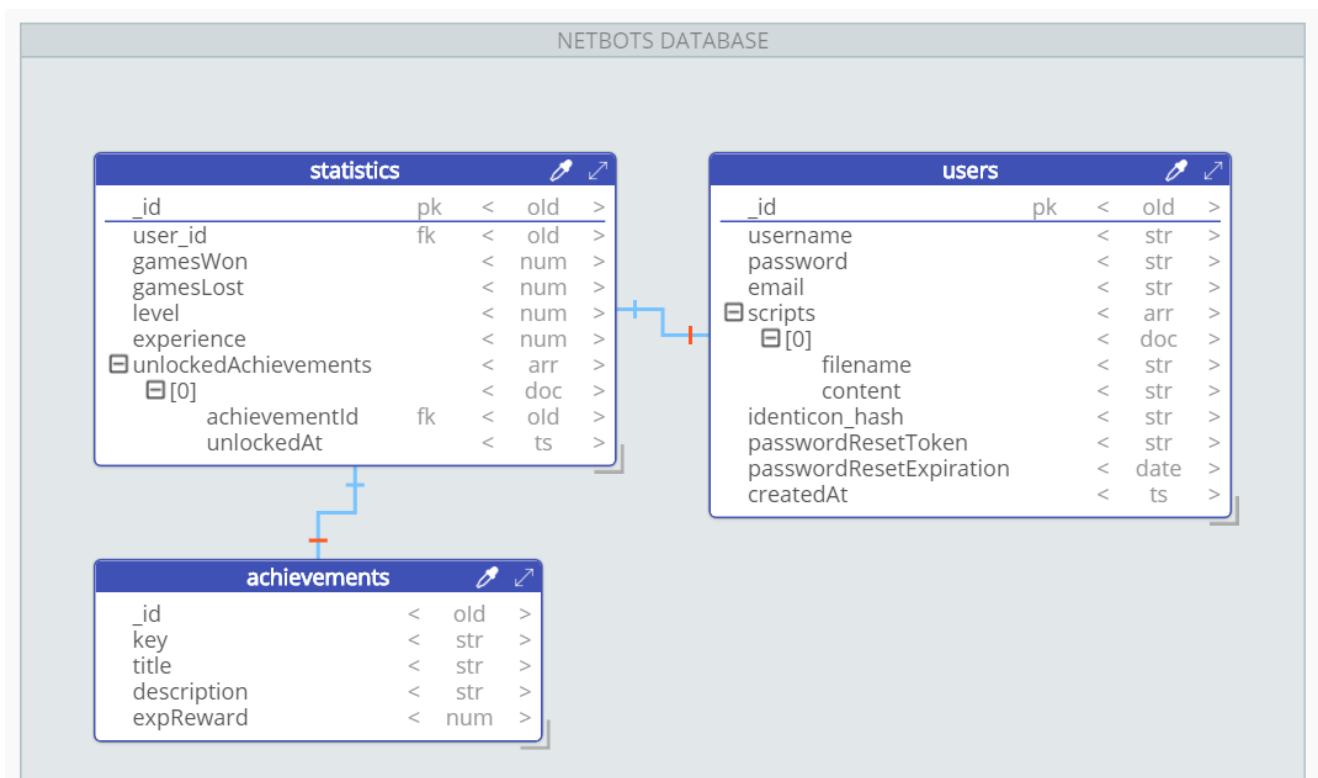
2.3. Duomenų bazės projektas

Kadangi buvo naudojama **MongoDB** duomenų bazė, pastarosios diagrama skiriasi nuo įprastinio reliacinės duomenų bazės modelio (žr. Pav. 2.1).

Paaiškinimai

Diagramoje matomas dokumentas **users**, be įprastų laukų, turi masyvo tipo **scripts** lauką, kuris yra skirtas saugoti įterptinių dokumentų sąrašą su laukais **filename** ir **content**. Šie yra skirti naudotojo parašytam kodui saugoti. Dokumente **statistics**, vietoje atskiros tarpinės lentelės reliacinės duomenų bazės atveju, matomas **unlockedAchievements** masyvas, saugantis įterptinių dokumentų sąrašą apie žaidėjo atrakintus laimėjimus.

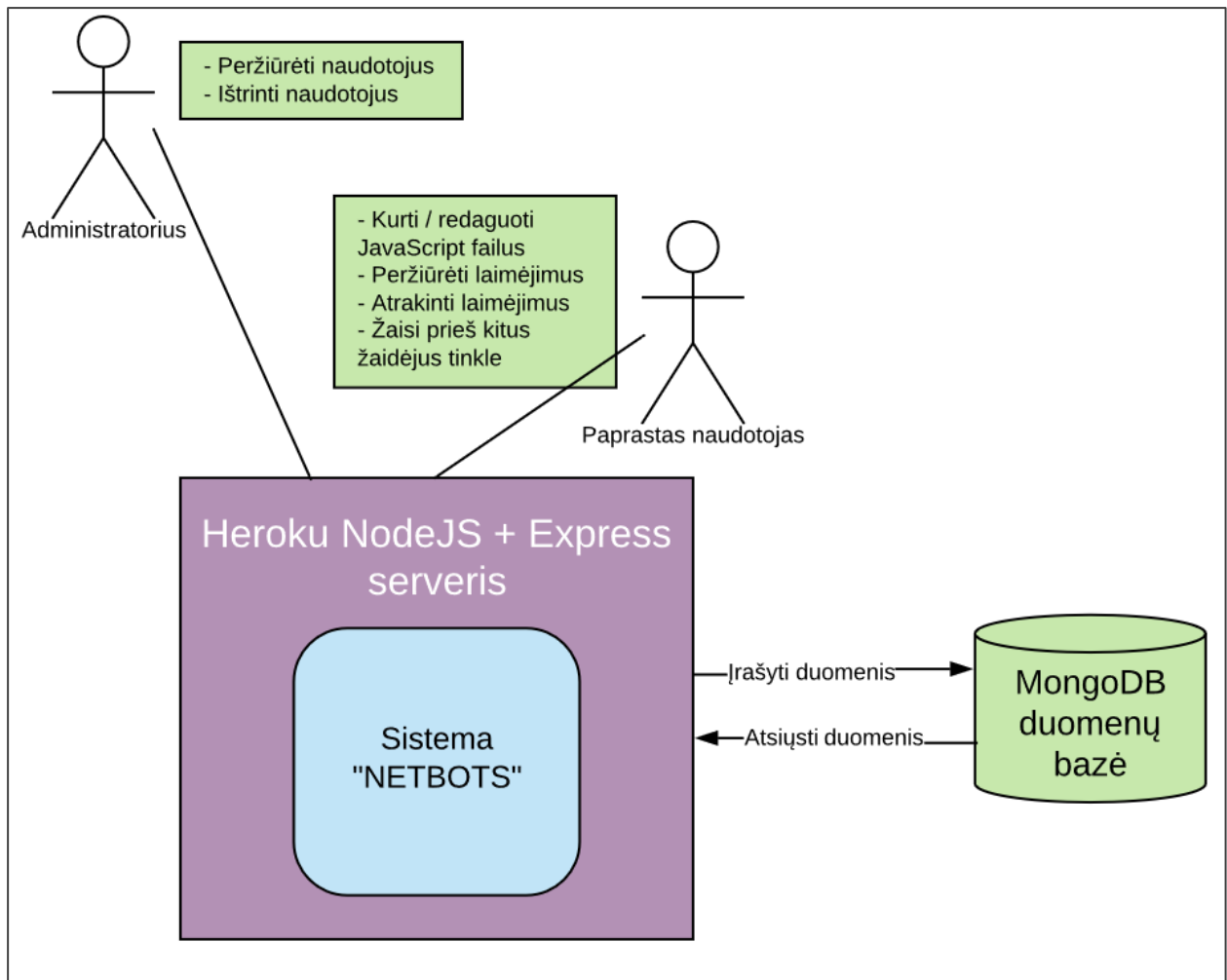
Kiekvienam naudotojo įrašui **users** dokumente yra sukuriamas vienas **statistics** įrašas, o **achievements** dokumentas visai duomenų basei yra tik vienas.



Pav. 2.1 Sistemos duomenų bazės diagrama

2.4. Konteksto diagrama

Sistemos konteksto diagrama, pavaizduota iliustracijoje Pav. 2.2 parodo sistemos sąveiką su ją supančia aplinka. Dviejų tipų naudotojai atlieka užklausas į web serverį, kuriame patalpinta sistema, o serveris komunikuoja su atskiru MongoDB duomenų bazės serveriu duomenų įrašymui, keitimui bei gavimui.

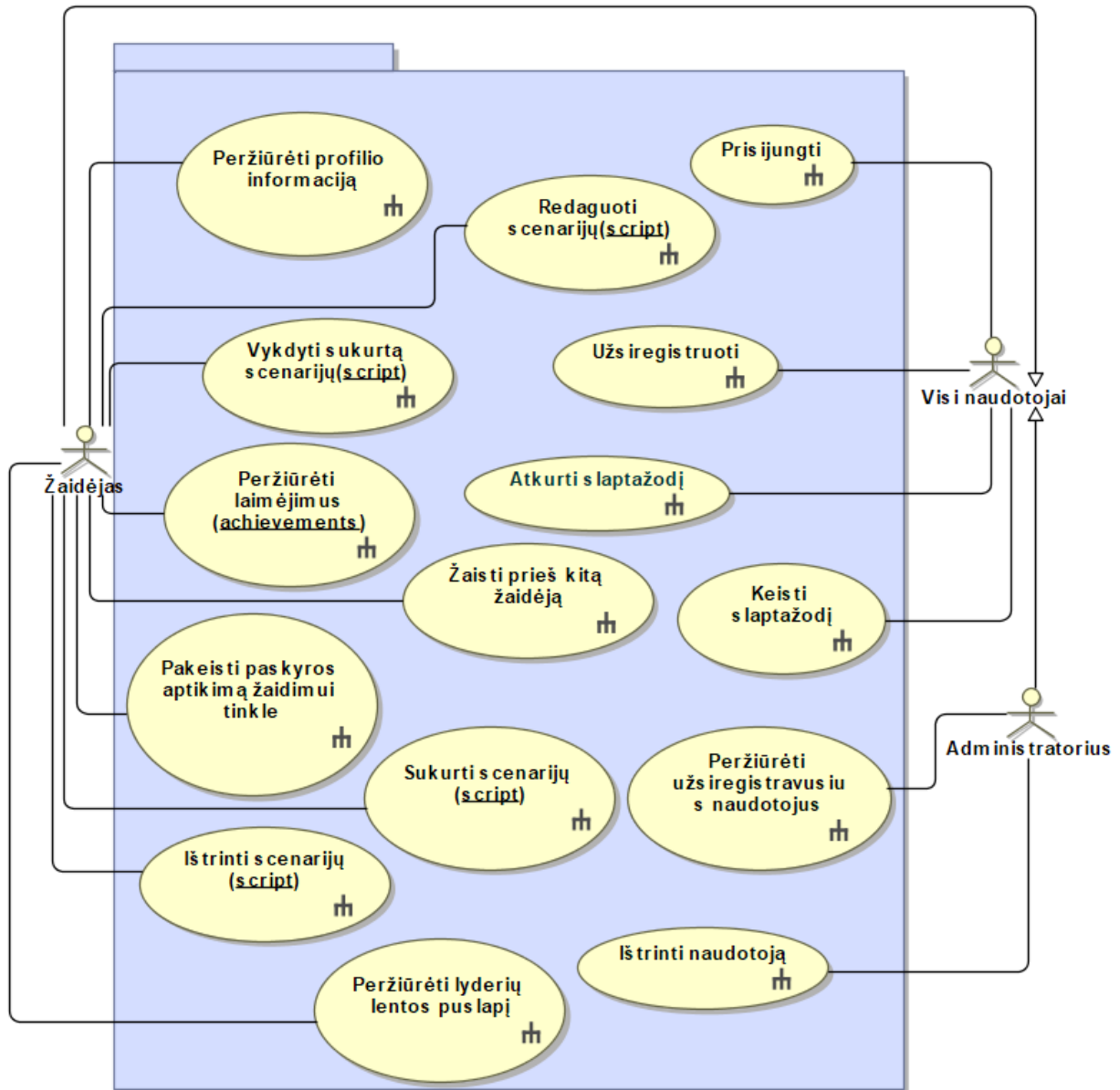


Pav. 2.2 Sistemos konteksto diagrama

2.6. UML diagramos

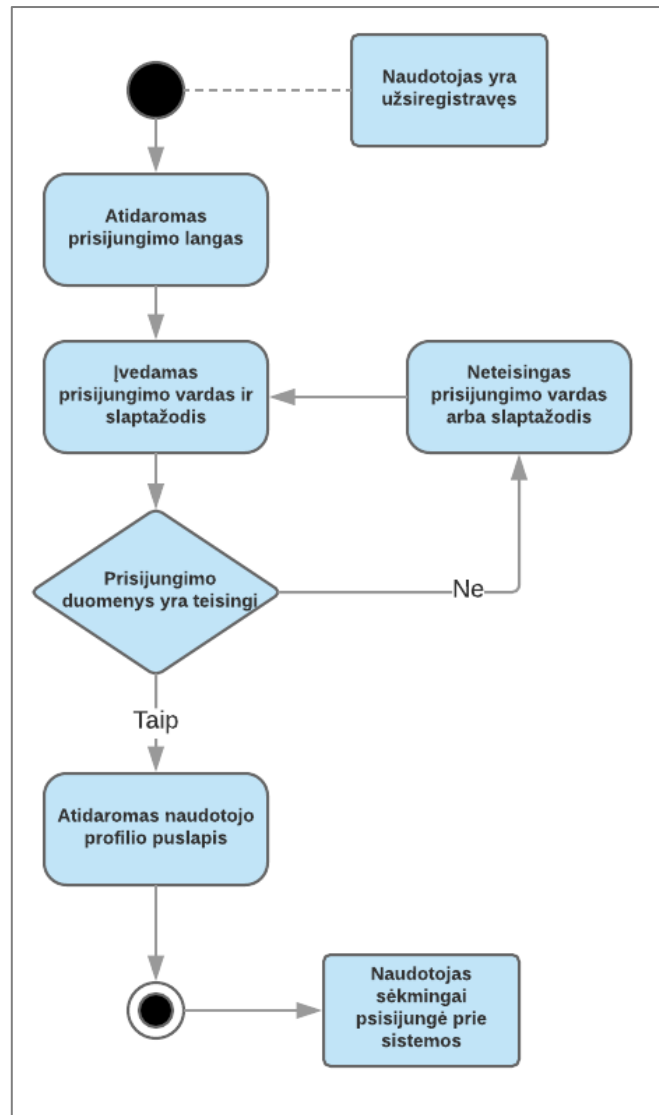
2.6.1. Panaudos atvejų diagrama

Žemiau pateikta kuriamos sistemos panaudos atvejų diagrama, atitinkanti anksčiau aprašytus funkcinis reikalavimus (žr. Pav. 2.3). Vis dėlto, funkcinuose reikalavimuose nurodytas kodo rašymas integruotame kodo redaktoriuje nėra įtrauktas į diagramą kaip atskiras panaudojimo atvejis, kadangi šis veiksmas apima scenarijų kūrimą be redagavimą.

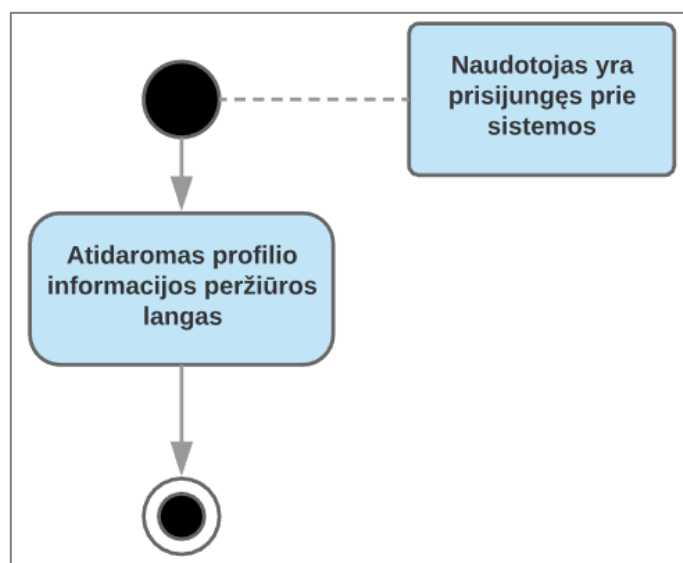


Pav. 2.3 Kuriamos sistemos panaudos atvejų diagrama

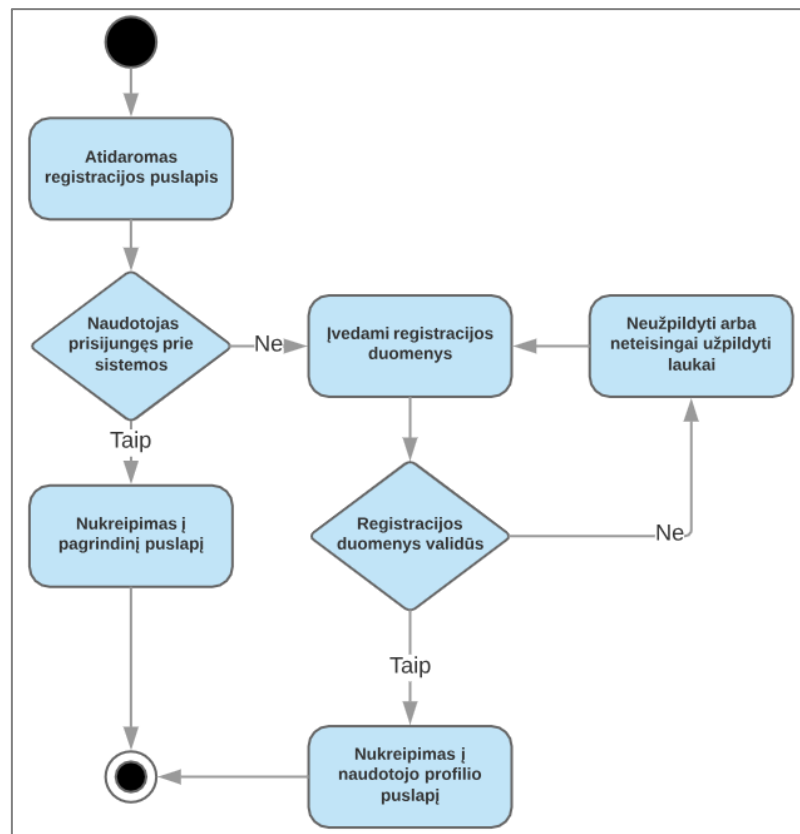
2.6.2. Veiklos diagramos



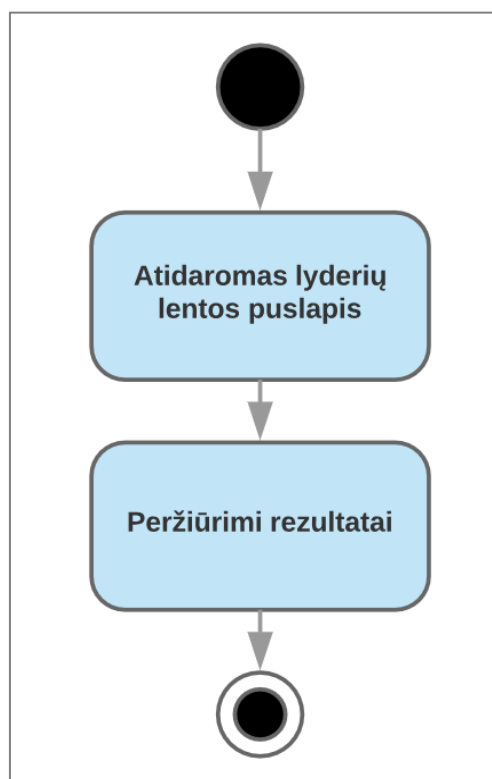
Pav. 2.4 Panaudojimo atvejo „Prisijungti“ veiklos diagrama



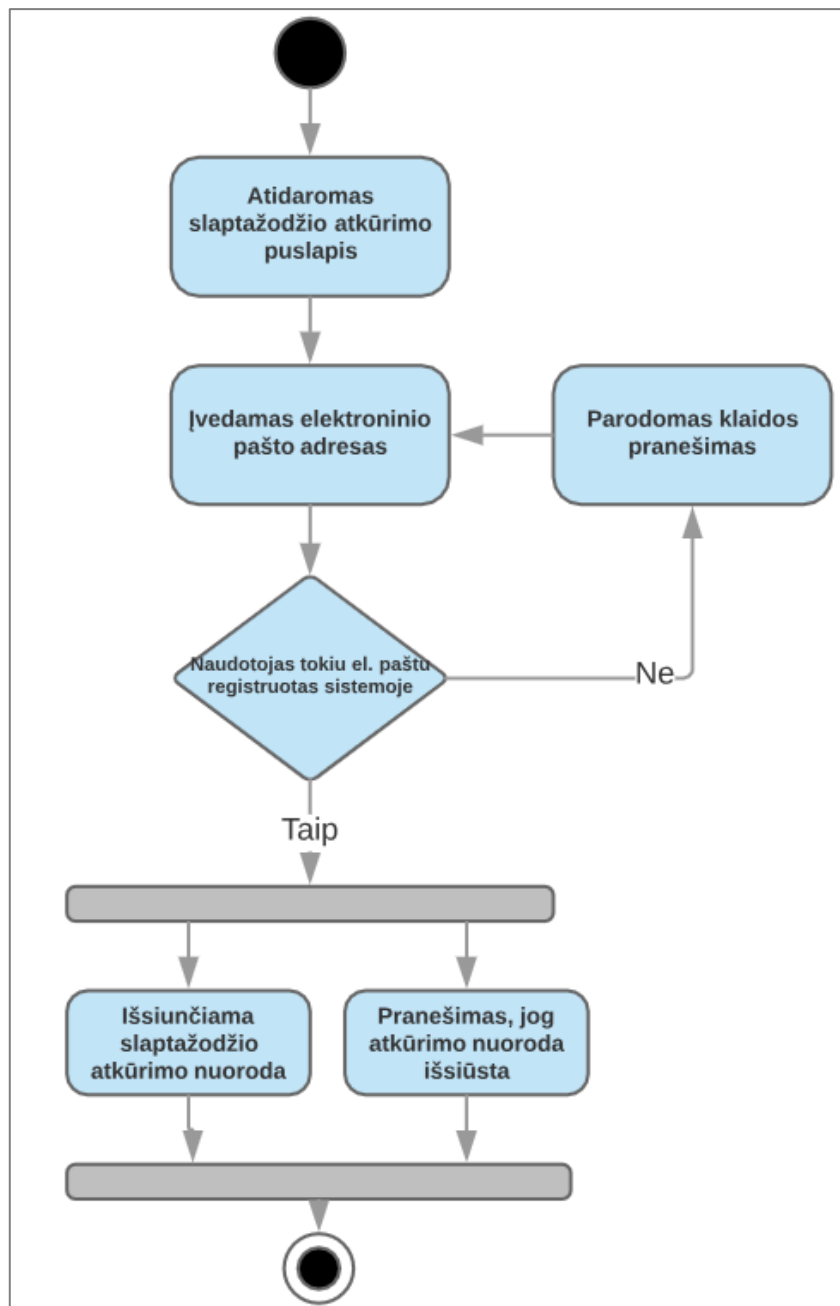
Pav. 2.5 Panaudojimo atvejo „Peržiūrėti profilio informaciją“ veiklos diagrama



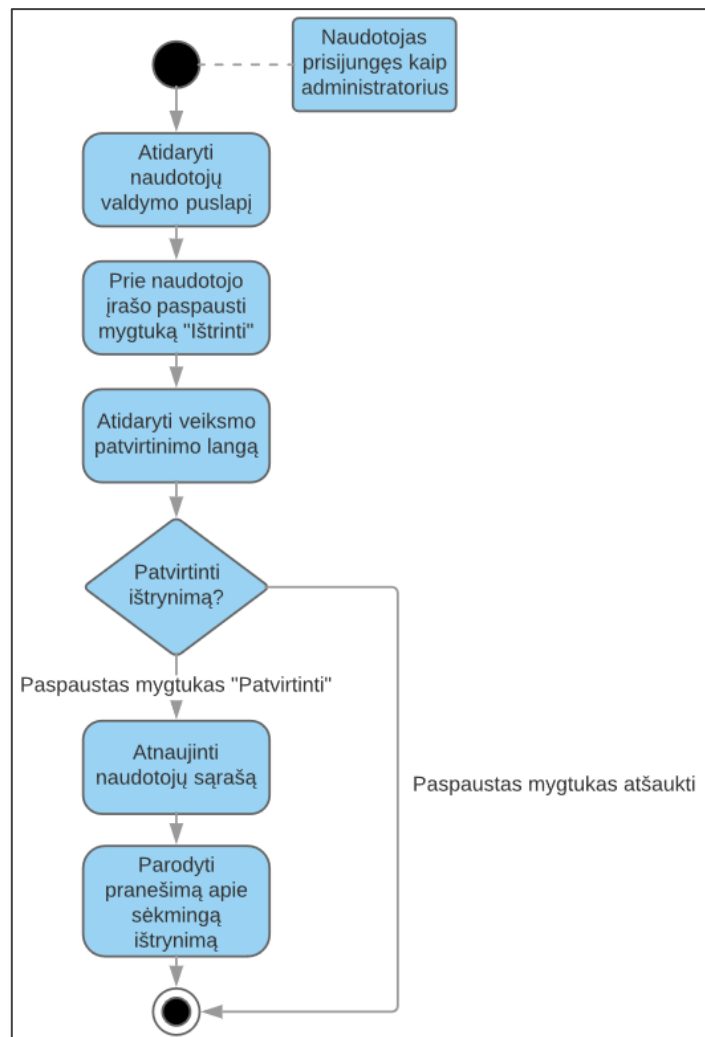
Pav. 2.6 Panaudojimo atvejo „Užsiregistruoti“ veiklos diagrama



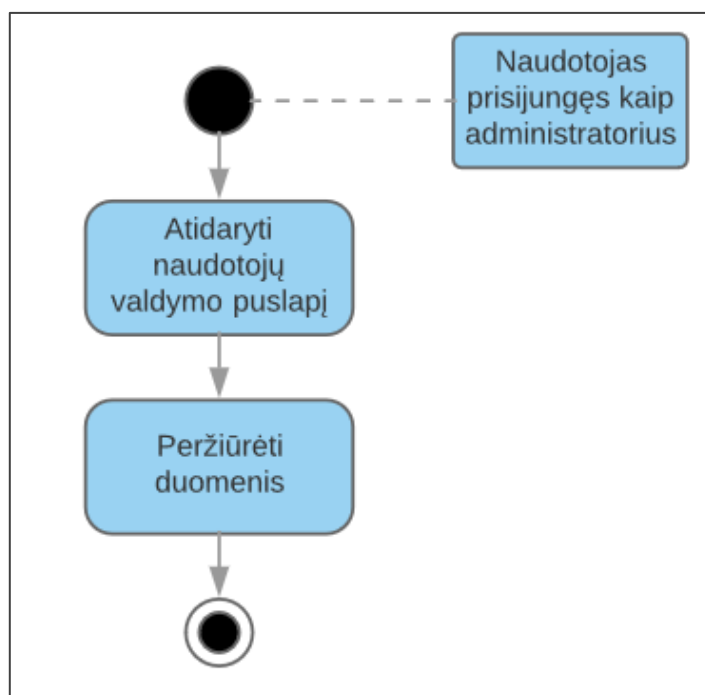
Pav. 2.7 Panaudos atvejo „Peržiūrėti lyderių lentos puslapį“ veiklos diagrama



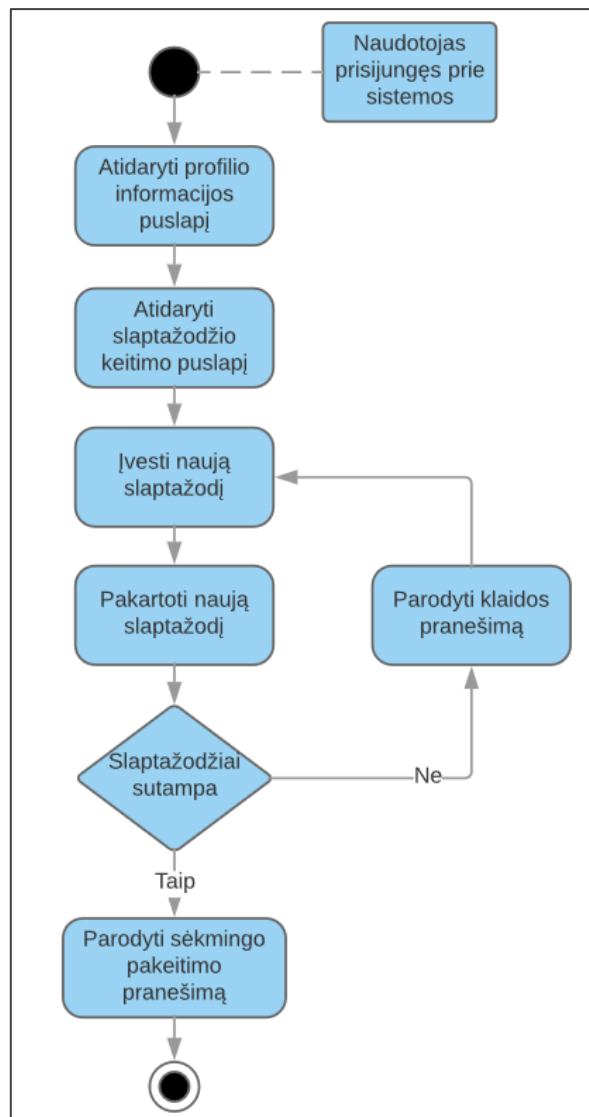
Pav. 2.8 Panaudojimo atvejo „Atkurti slaptažodį“ veiklos diagrama



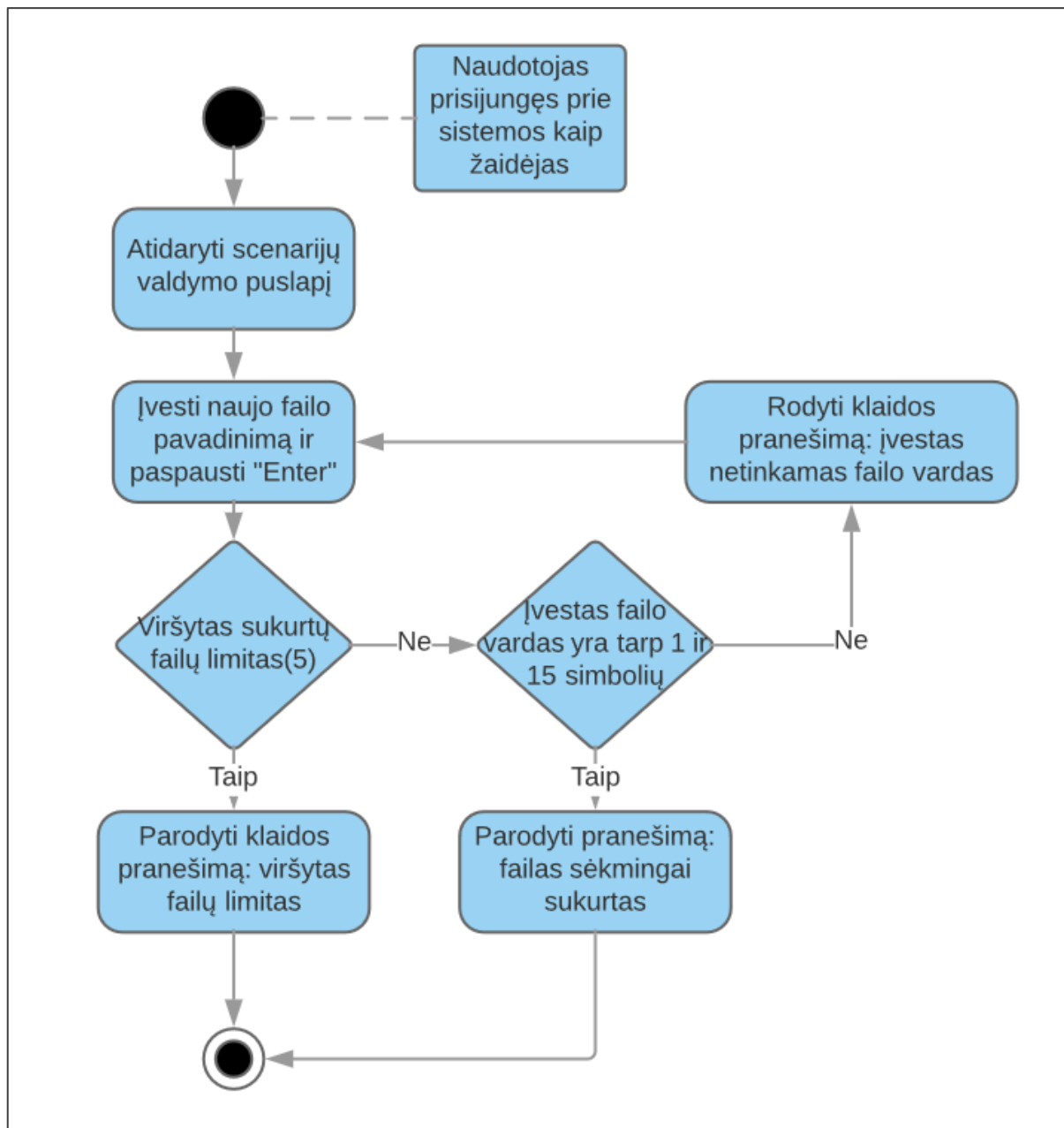
Pav. 2.9 Panaudojimo atvejo „Ištrinti naudotoją“ veiklos diagrama



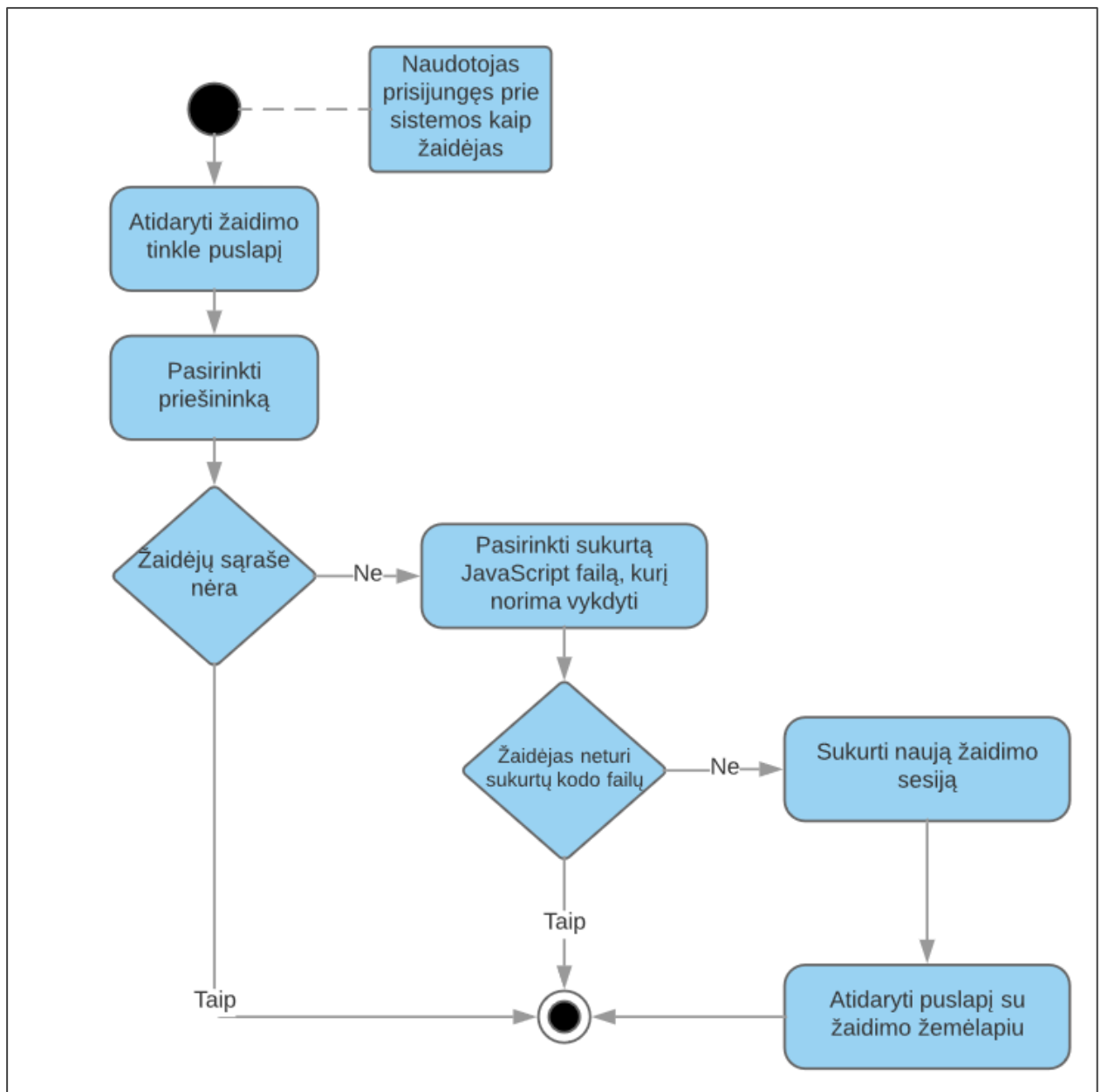
Pav. 2.10 Panaudojimo atvejo „Peržiūrėti užsiregistravusius naudotojus“ veiklos diagrama



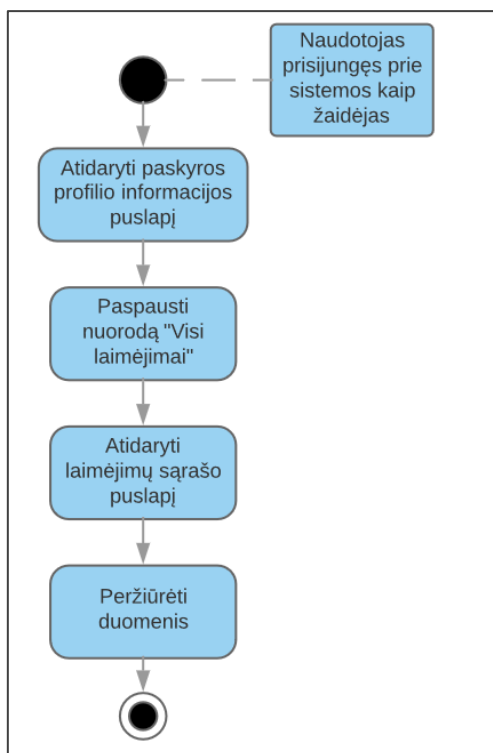
Pav. 2.11 Panaudojimo atvejo „Keisti slaptažodį“ veiklos diagrama



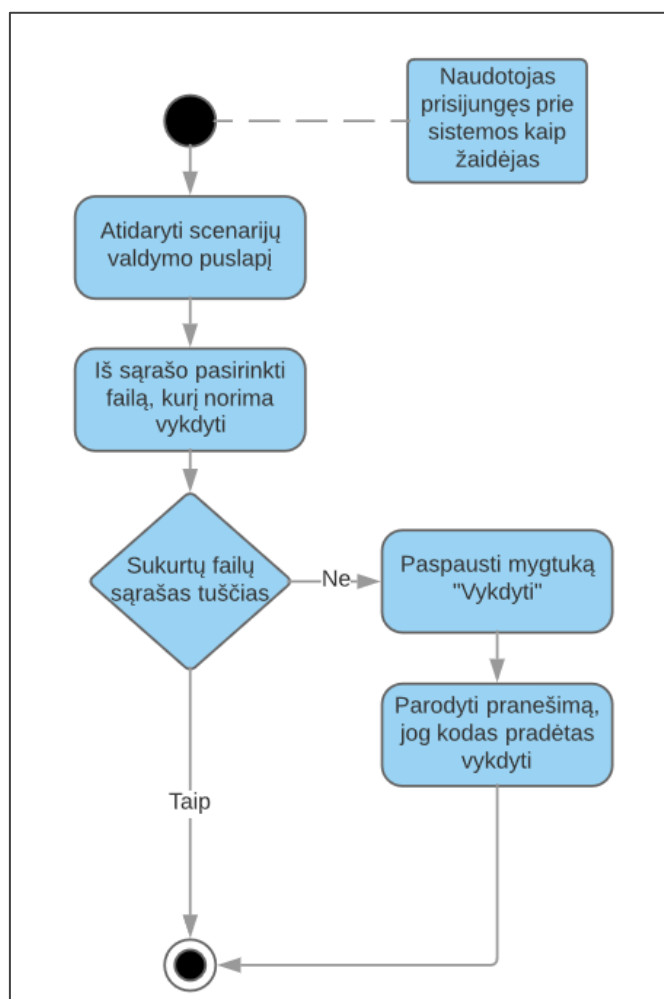
Pav. 2.12 Panaudojimo atvejo „Sukurti naują scenarijų“ veiklos diagrama



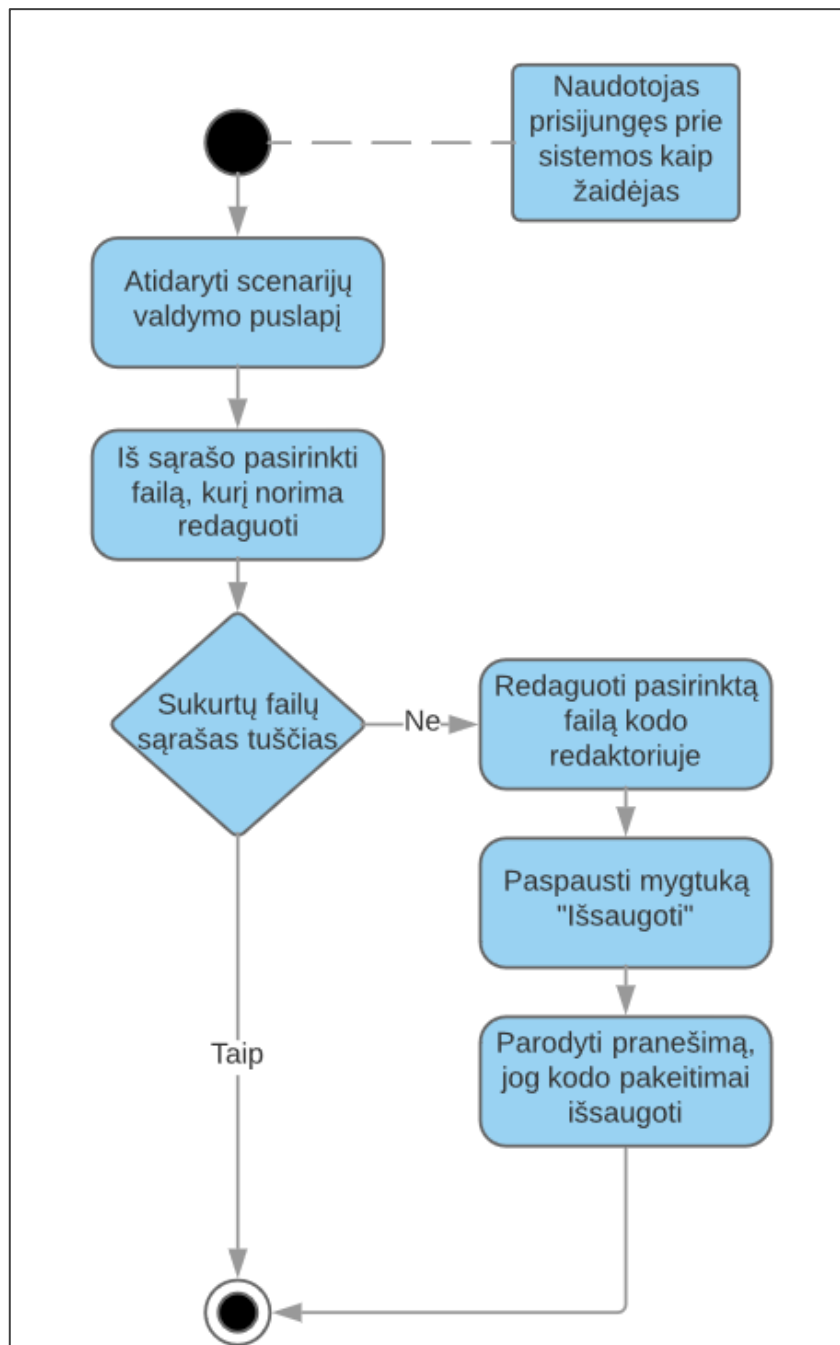
Pav. 2.13 Panaudojimo atvejo „Žaisti prieš kitą žaidėją“ veiklos diagrama



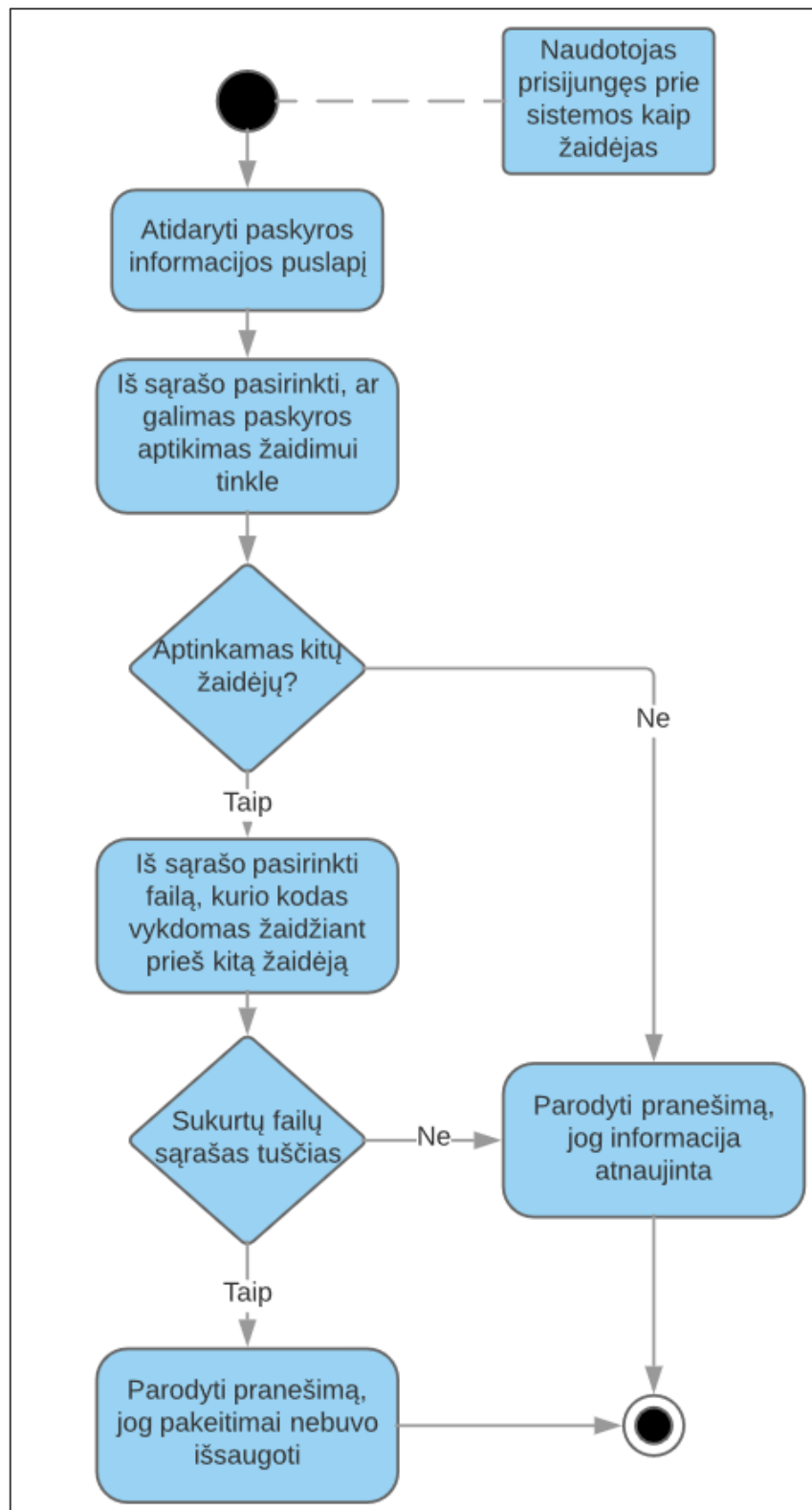
Pav. 2.14 Panaudojimo atvejo „Peržiūrėti laimėjimus“ veiklos diagrama



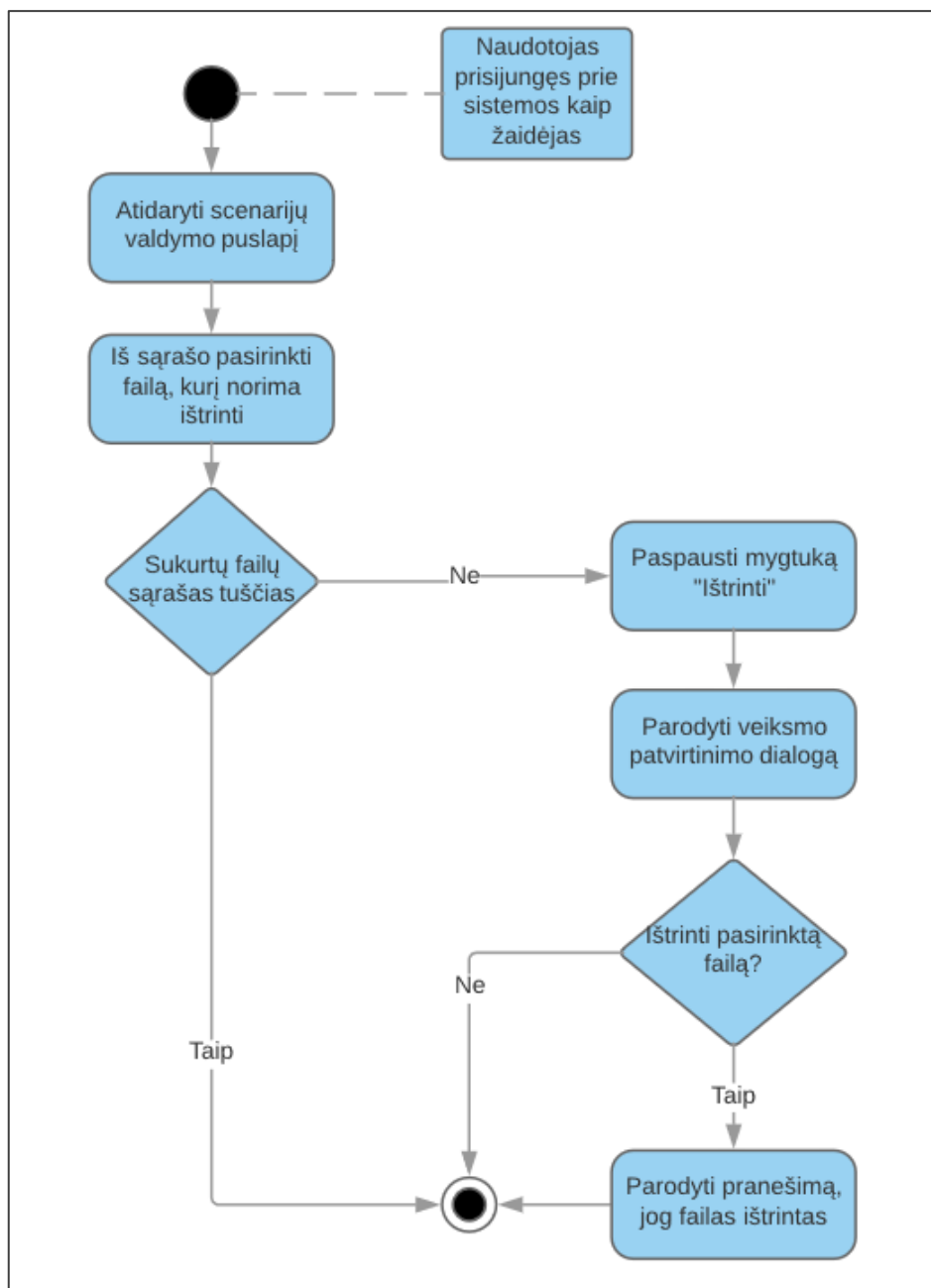
Pav. 2.15 Panaudojimo atvejo „Vykdėti sukurtą scenarijų“ veiklos diagrama



Pav. 2.16 Panaudojimo atvejo „Redaguoti scenarijų“ veiklos diagrama

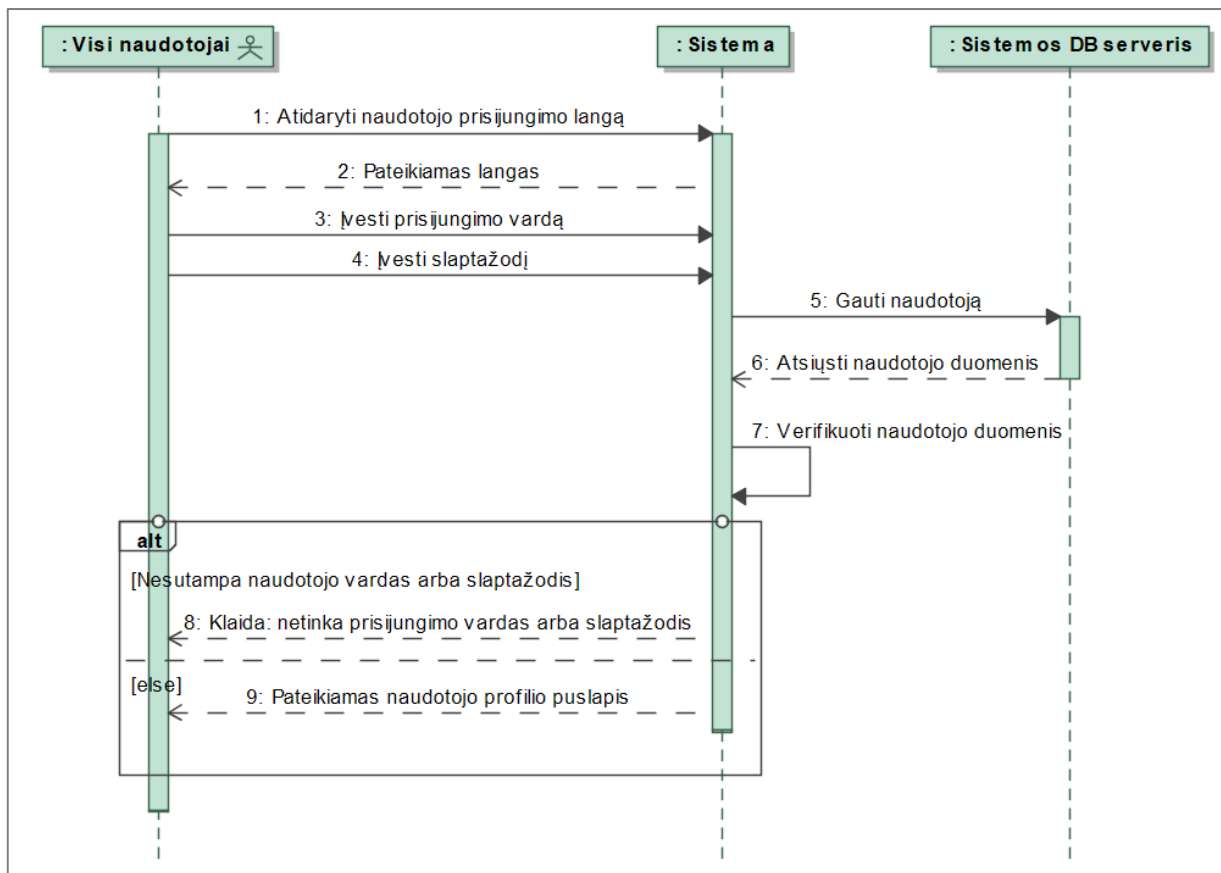


Pav. 2.17 Panaudojimo atvejo „Keisti paskyros aptikimą žaidimui tinkle“ veiklos diagrama

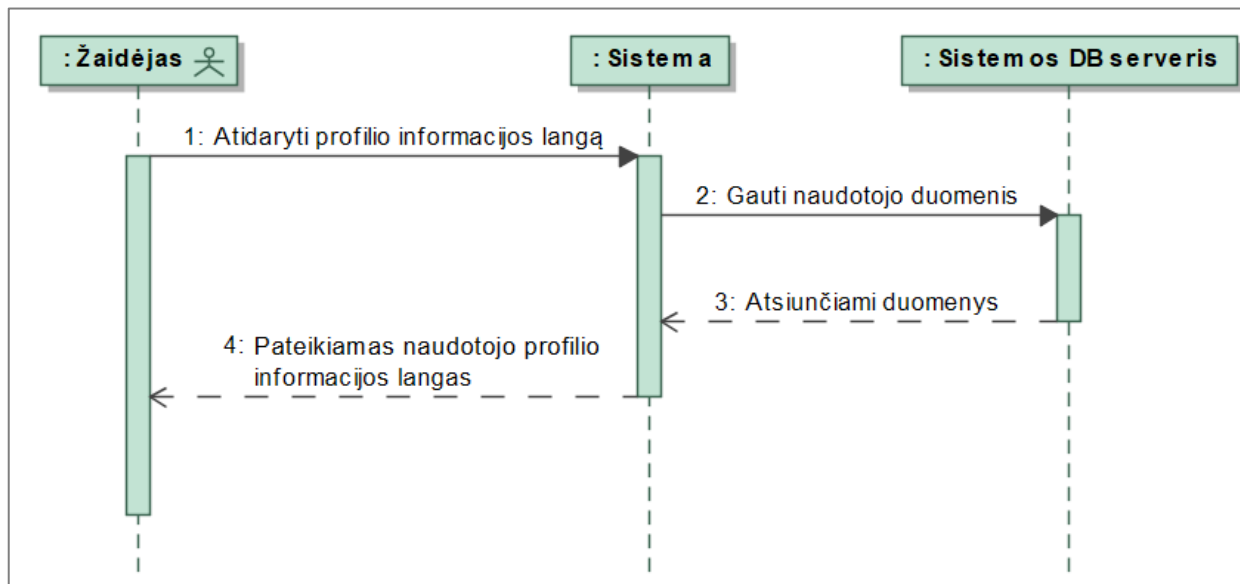


Pav. 2.18 Panaudojimo atvejo „Ištrinti scenarijų“ veiklos diagrama

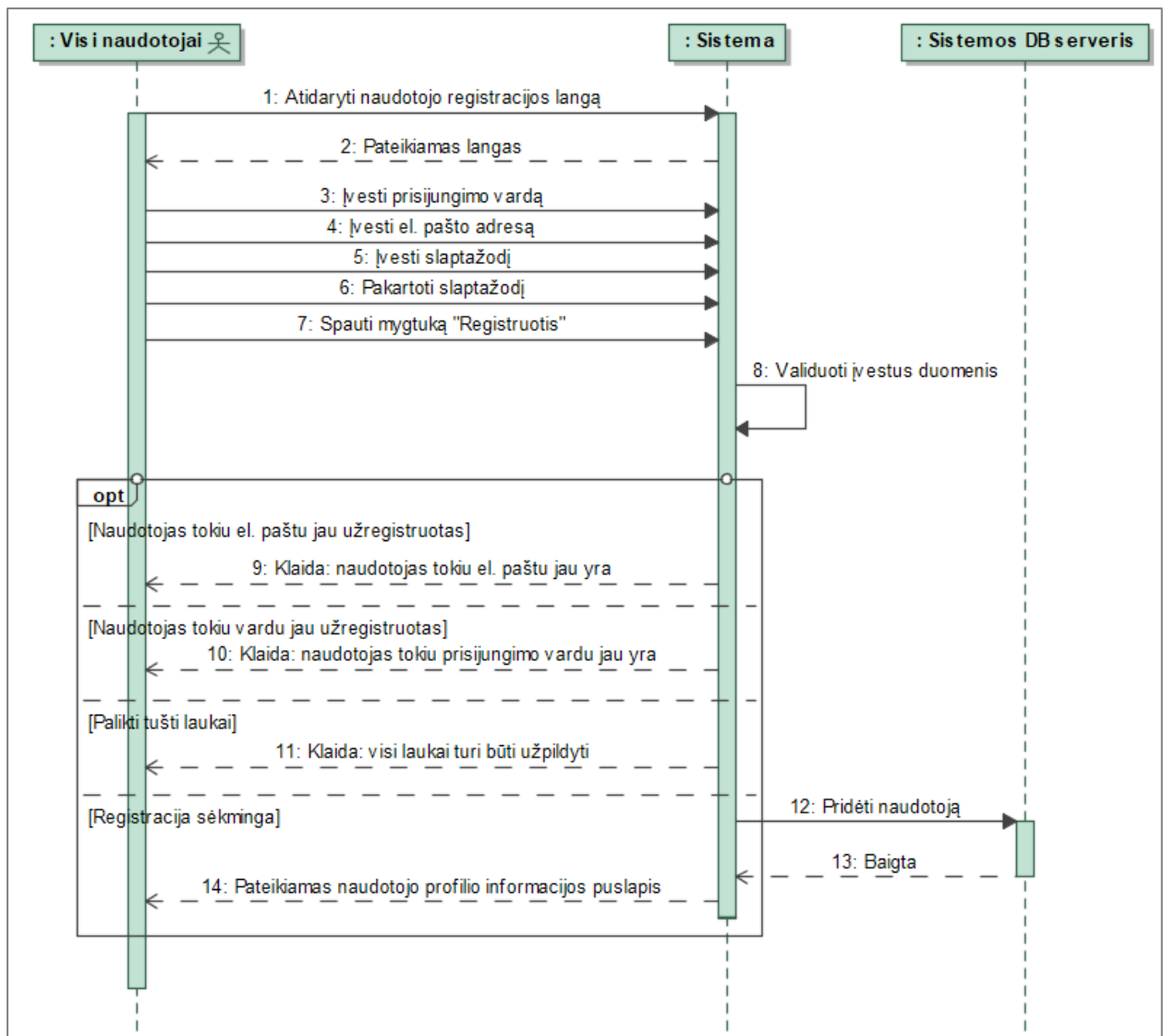
2.6.3. Sekų diagramos



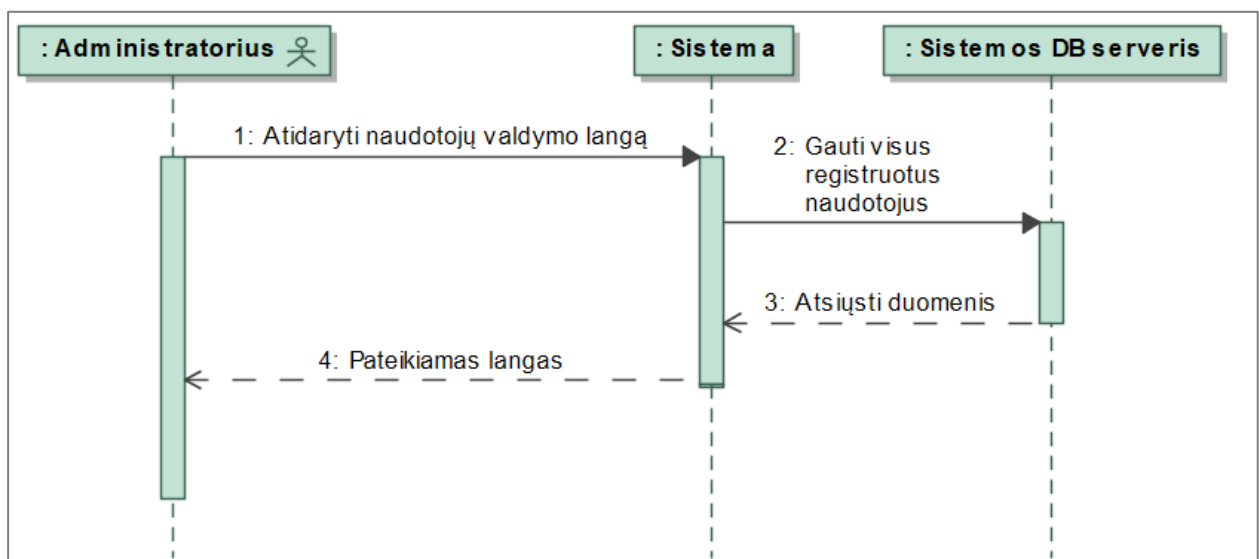
Pav. 2.19 Panaudojimo atvejo „Prisijungti“ sekų diagrama



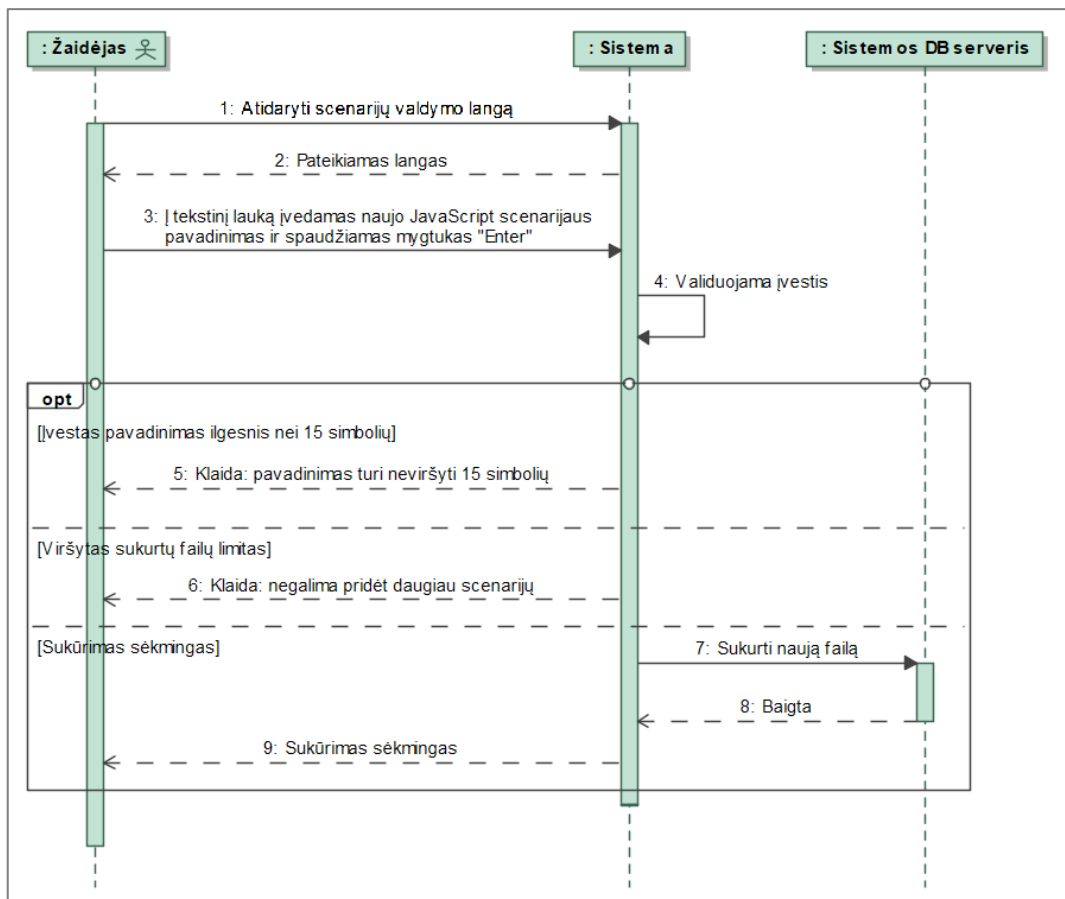
Pav. 2.20 Panaudojimo atvejo „Peržiūrėti profilio informaciją“ sekų diagrama



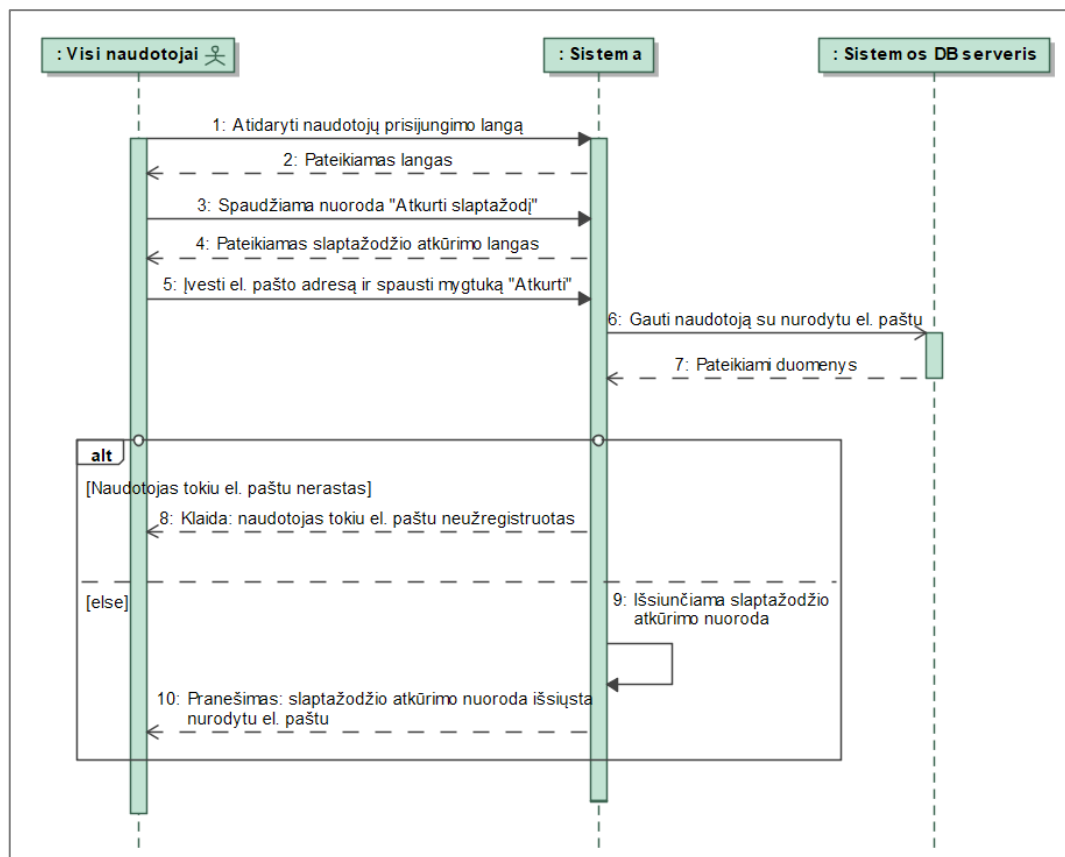
Pav. 2.21 Panaudojimo atvejo „Užsiregistruoti“ sekų diagrama



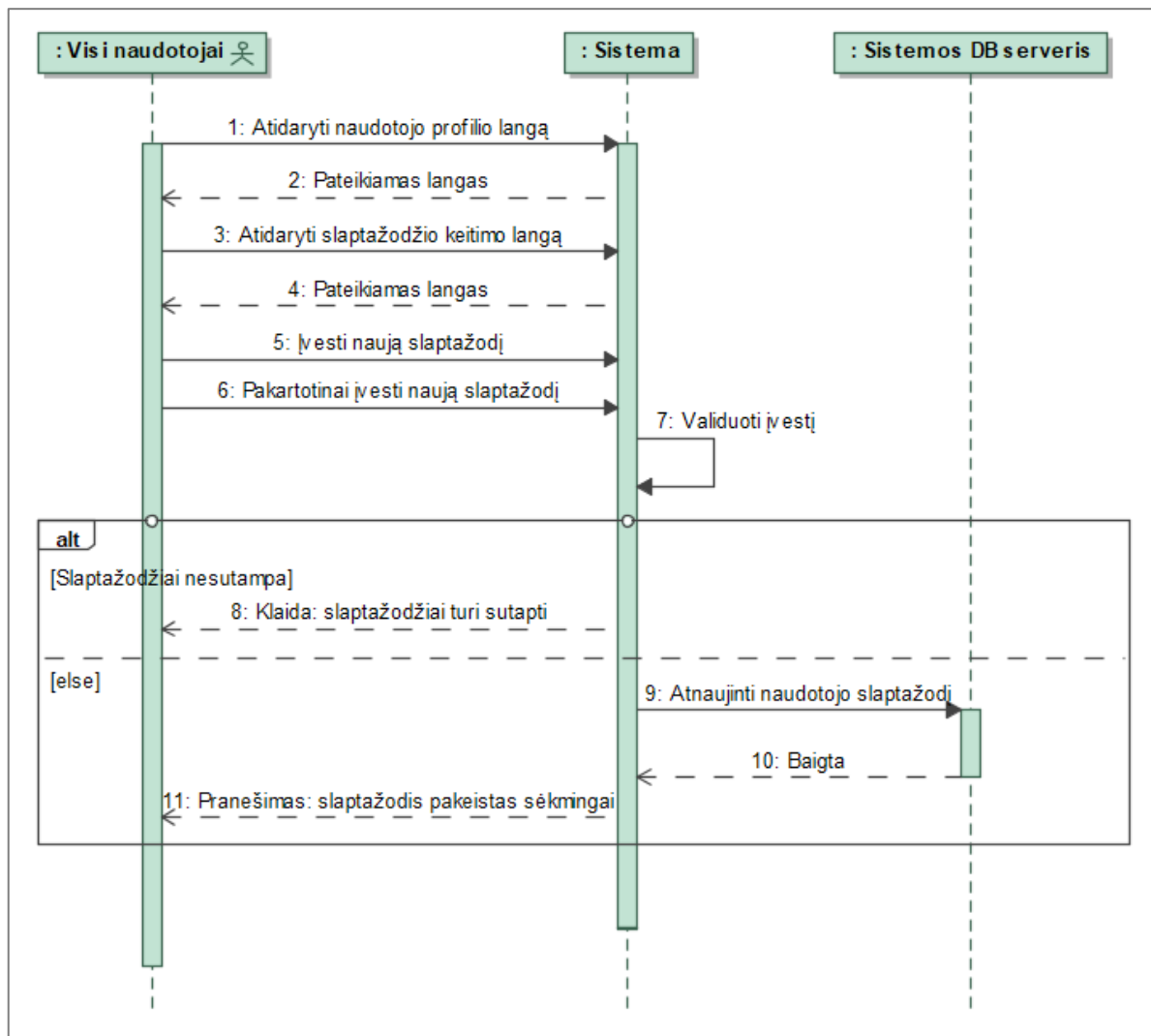
Pav. 2.22 Panaudojimo atvejo „Peržiūrėti užsiregistravusius naudotojus“ sekų diagrama



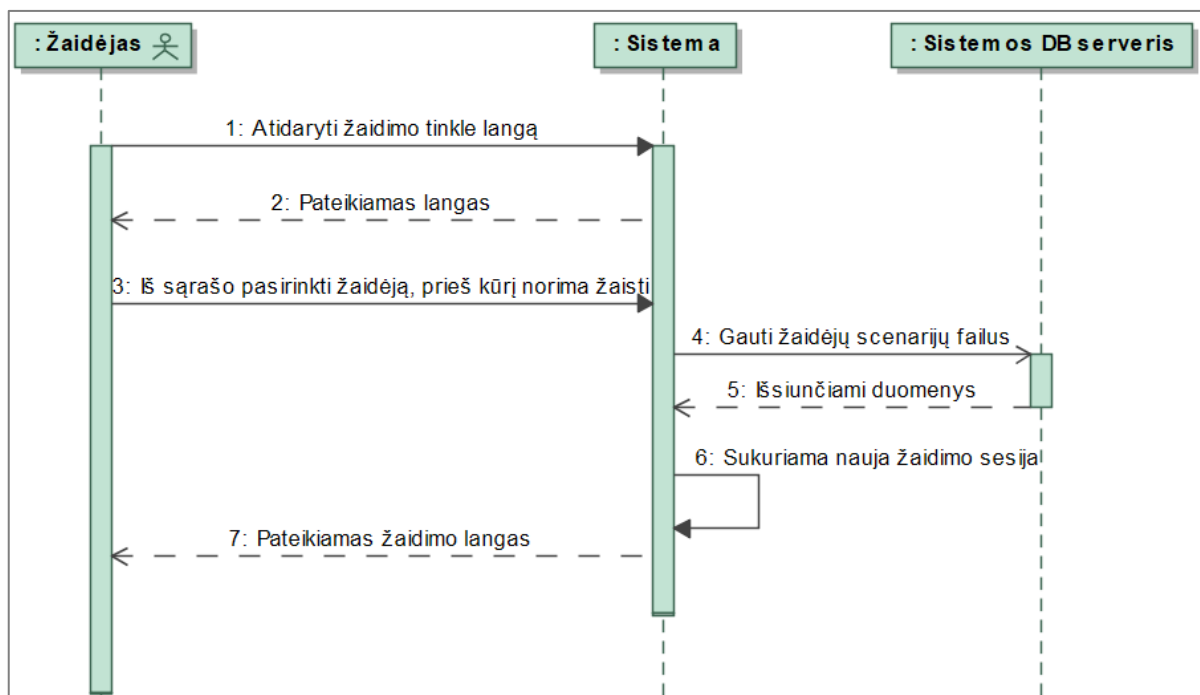
Pav. 2.23 Panaudojimo atvejo „Sukurti naują scenarijų“ sekų diagrama



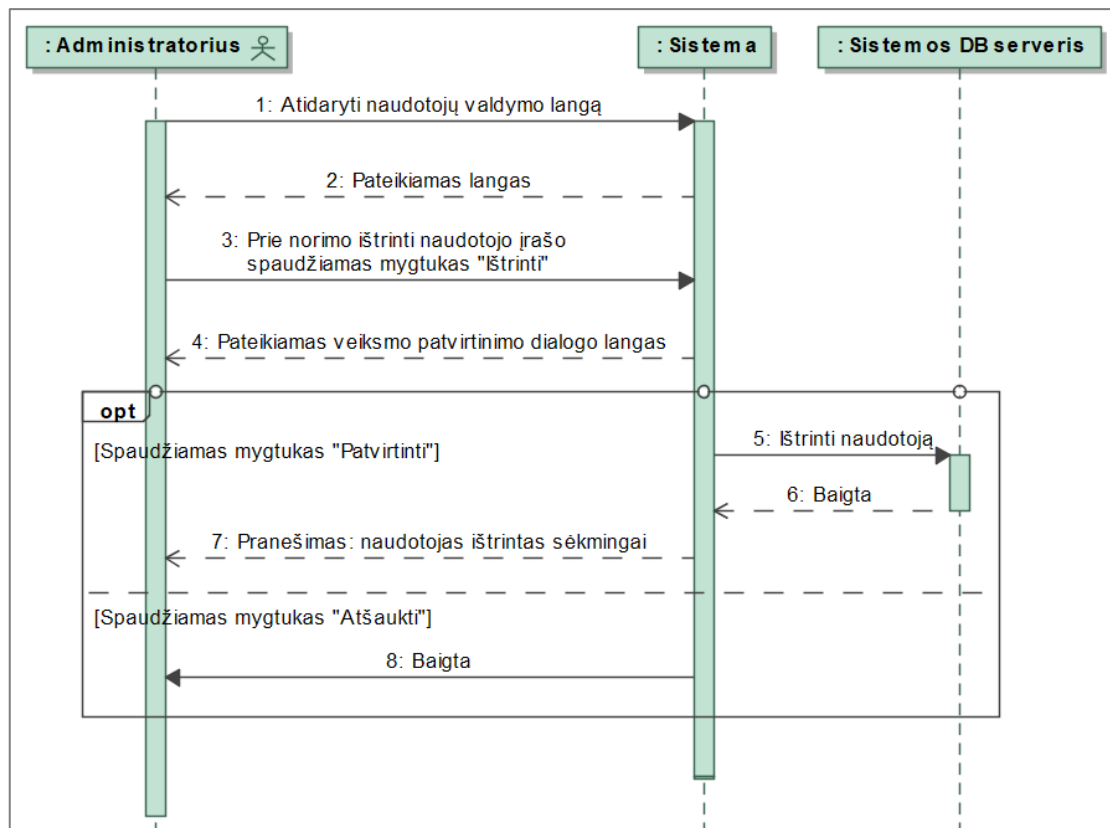
Pav. 2.24 Panaudojimo atvejo „Atkurti slaptažodį“ sekų diagrama



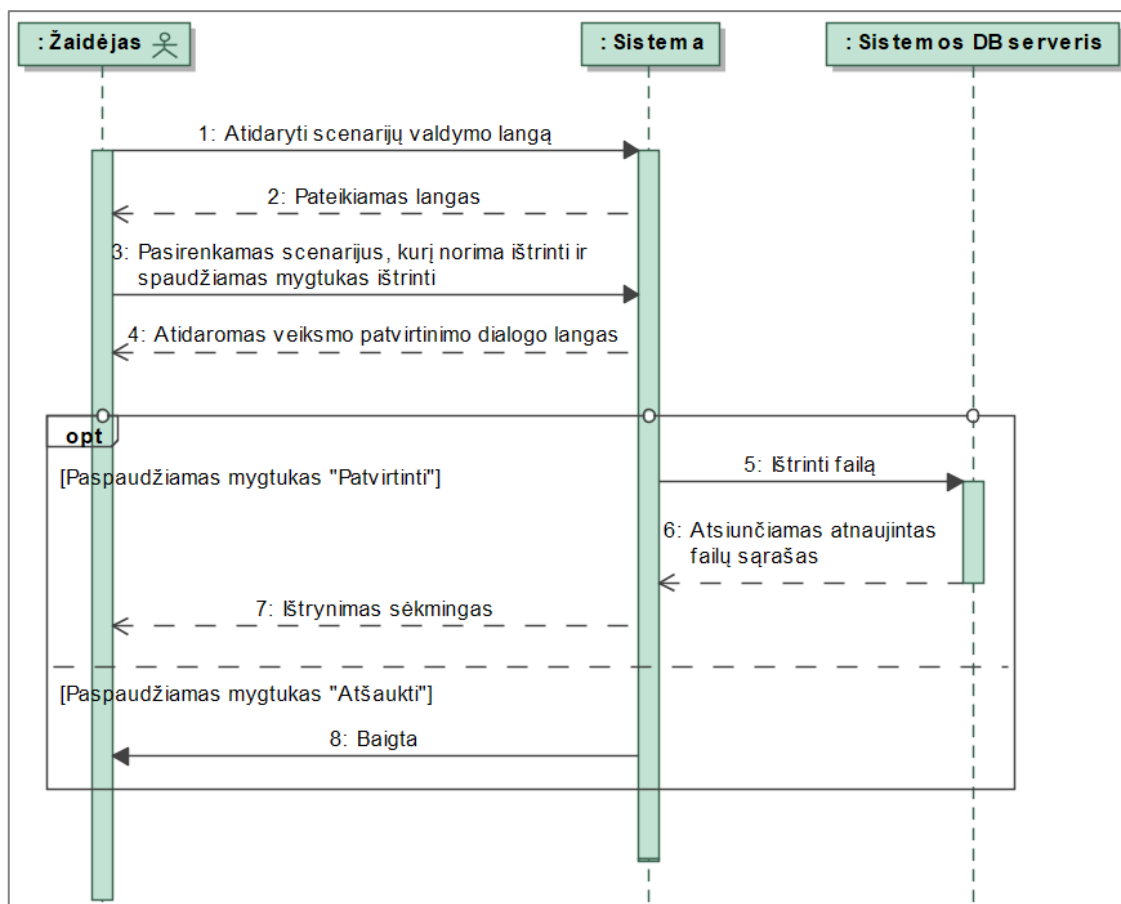
Pav. 2.25 Panaudojimo atvejo „Keisti slaptažodį“ sekų diagrama



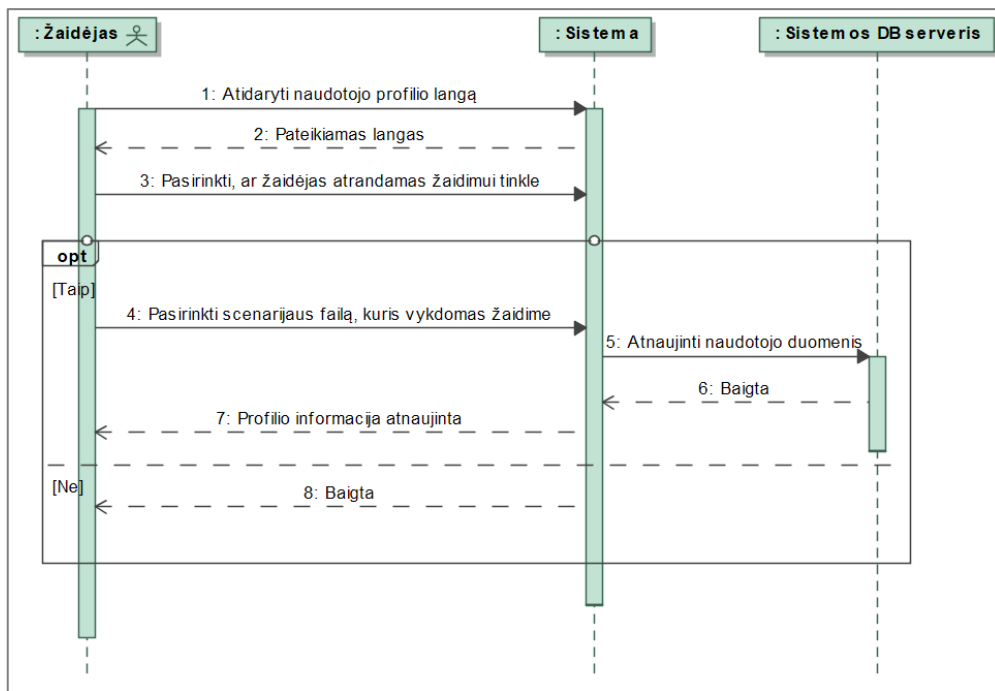
Pav. 2.26 Panaudojimo atvejo „Žaisti prieš kitą žaidėją“ sekų diagrama



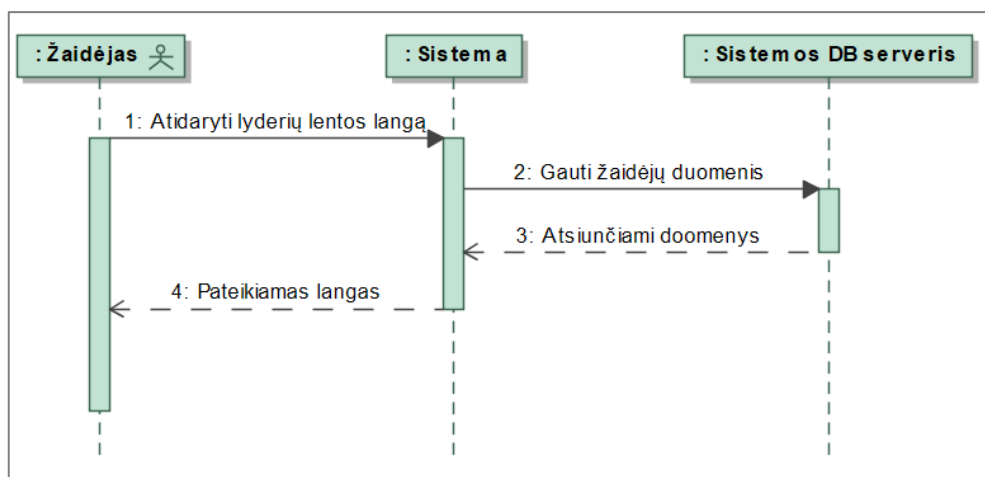
Pav. 2.27 Panaudojimo atvejo „Ištrinti naudotoją“ sekų diagrama



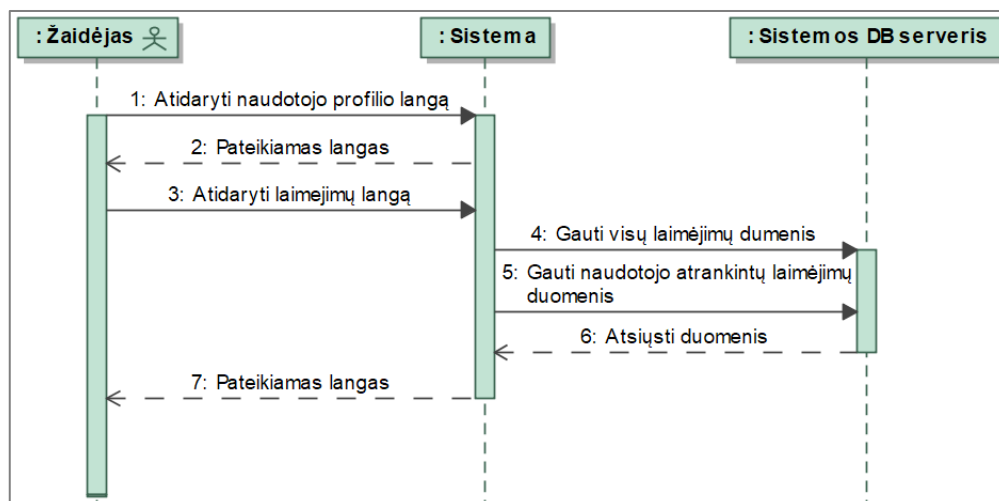
Pav. 2.28 Panaudojimo atvejo „Ištrinti scenarijų“ sekų diagrama



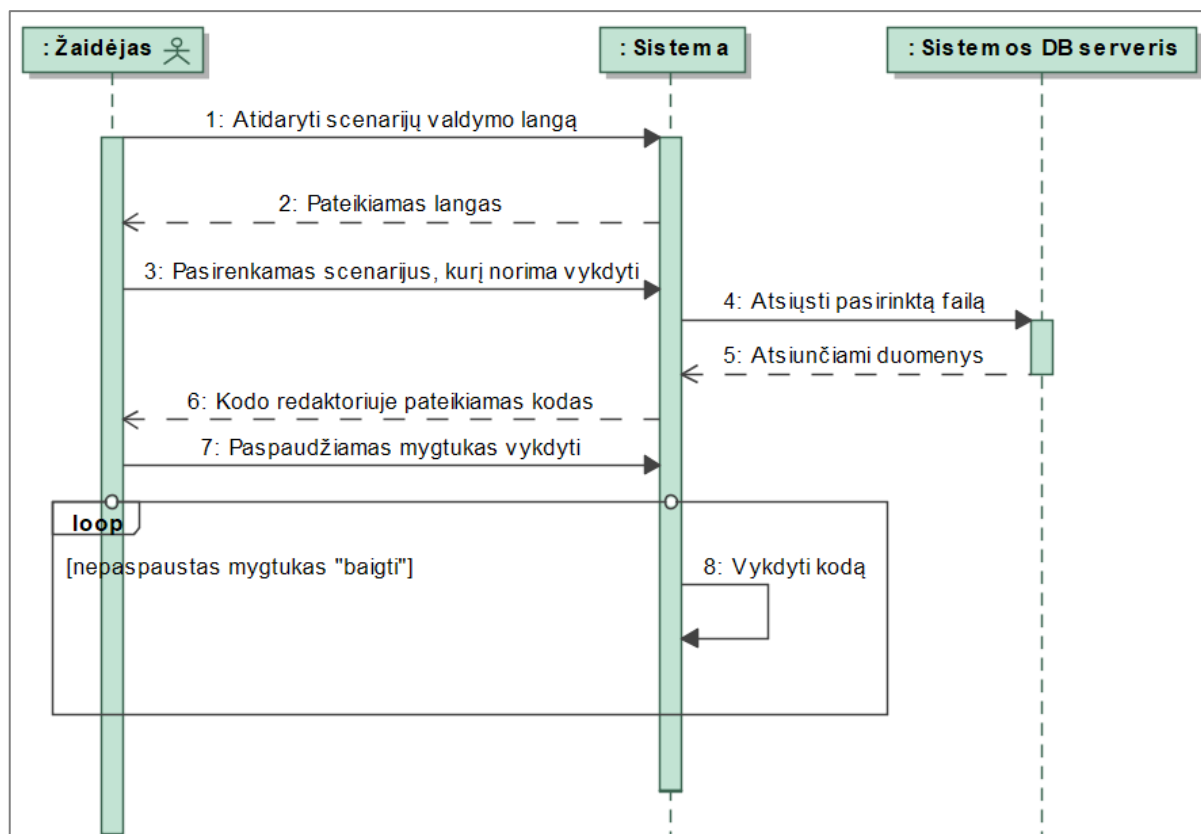
Pav. 2.29 Panaudojimo atvejo „Pakeisti paskyros aptikimą žaidimui tinkle“ sekų diagrama



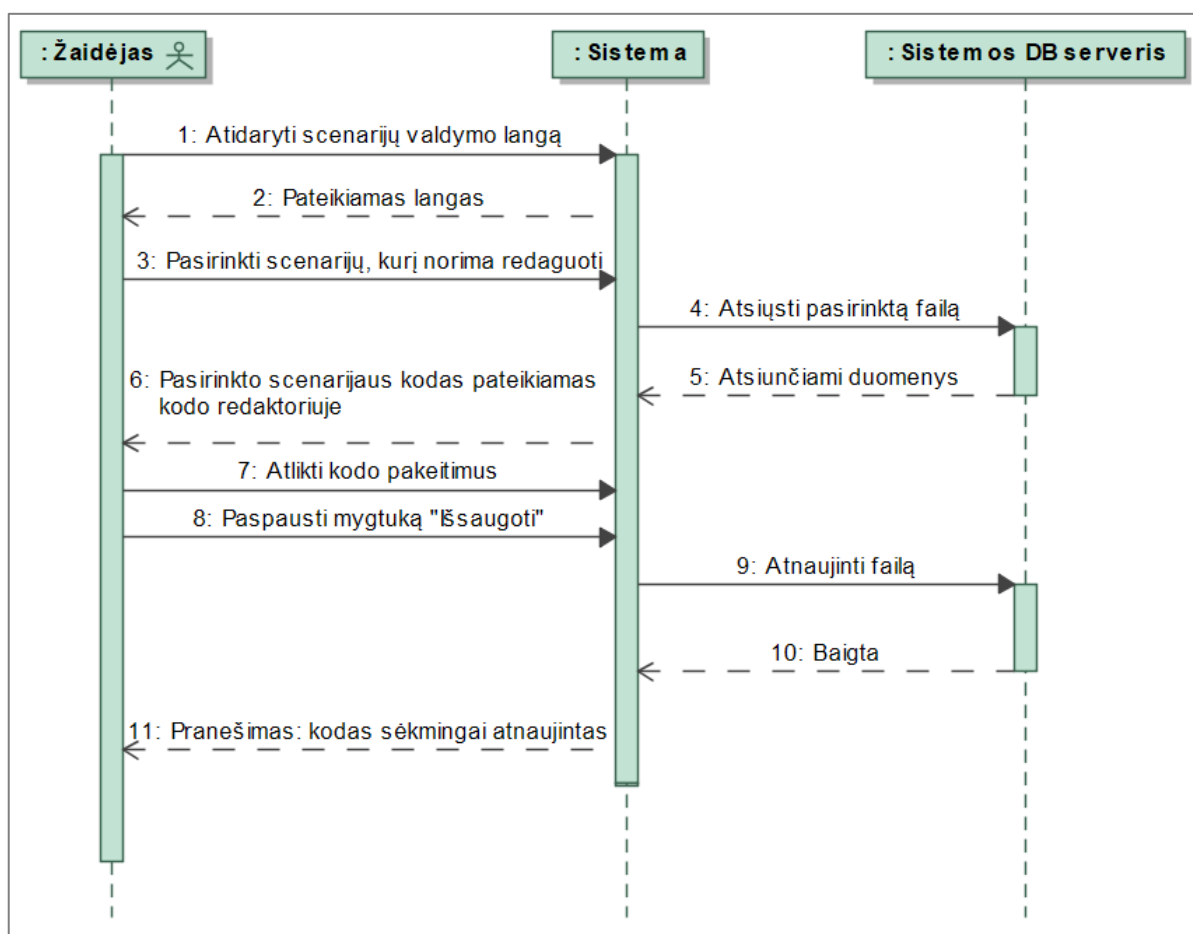
Pav. 2.30 Panaudojimo atvejo „Peržiūrėti lyderių lentos puslapį“ sekų diagrama



Pav. 2.31 Panaudojimo atvejo „Peržiūrėti laimėjimus“ sekų diagrama



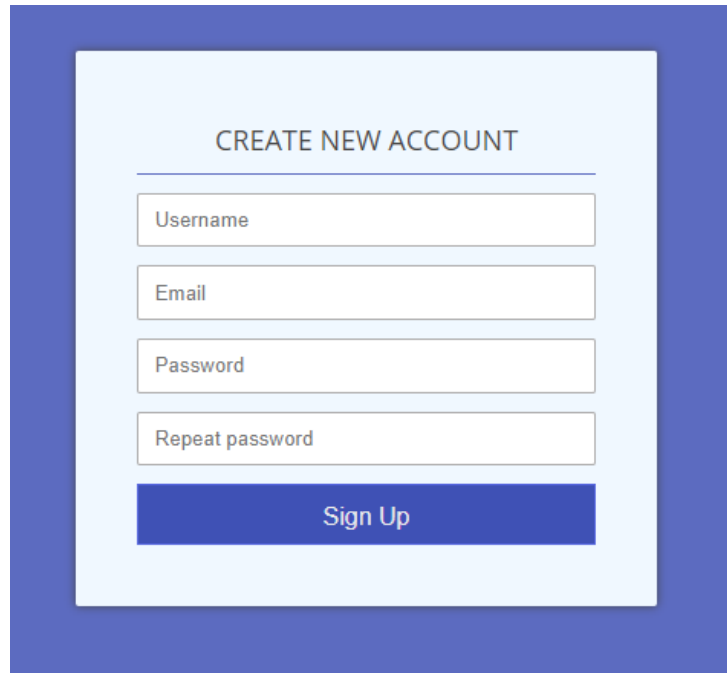
Pav. 2.32 Panaudojimo atvejo „Vykdyti sukurta scenarijų“ sekų diagrama



Pav. 2.33 Panaudojimo atvejo „Redaguoti scenarijų“ sekų diagrama

3. Vartotojo vadovas

Nuėjus nurodytu adresu atsidaro naudotojo registracijos langas (žr. Pav. 3.1)

The image shows a web registration form titled "CREATE NEW ACCOUNT". The form is centered on a light blue background with a dark blue border. It contains four input fields: "Username", "Email", "Password", and "Repeat password". Below these fields is a dark blue button labeled "Sign Up".

CREATE NEW ACCOUNT

Username

Email

Password

Repeat password

Sign Up

Pav. 3.1 Naudotojo registracijos langas (<http://web-bots.herokuapp.com/register>)

Įvedimo laukų ir mygtukų paaiškinimai:

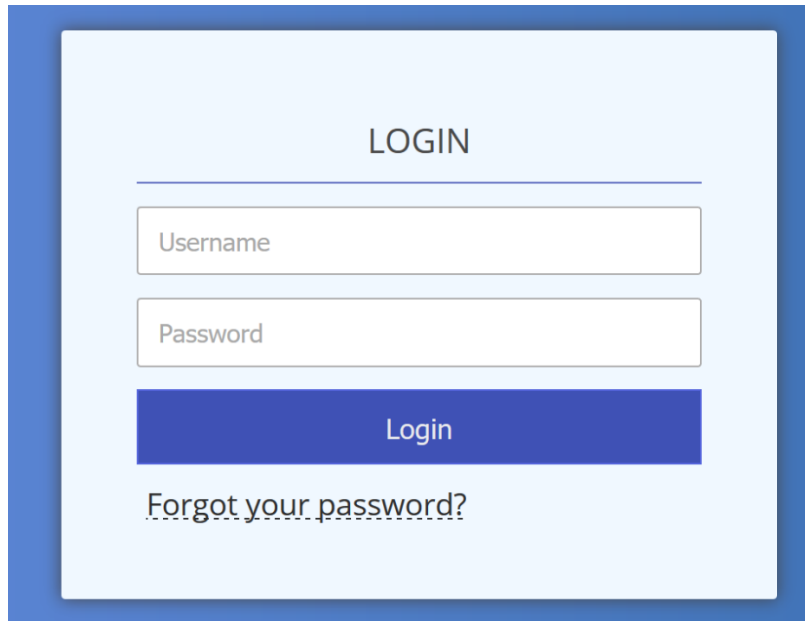
Username – naudotojo prisijungimo vardas

Password – naudotojo prisijungimo slaptažodis

Repeat password – pakartojamas anksčiau įvestas slaptažodis

Email – naudotojo elektroninis paštas

Žemiau(Pav. 3.2) parodytas naudotojo prisijungimo lango forma. Per tą patį puslapį jungiasi tiek papras tau naudotojas, tiek administratorius.

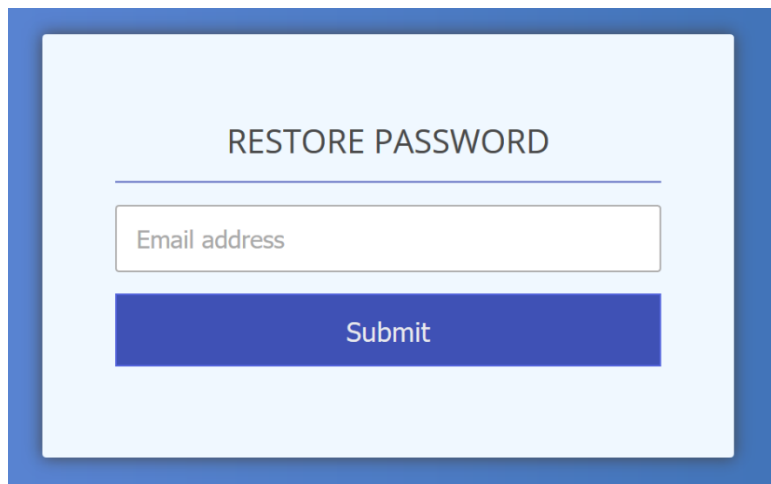


Pav. 3.2 Naudotojo prisijungimo forma(<http://web-bots.herokuapp.com/login>)

Username – naudotojo prisijungimo vardas

Password – naudotojo slaptažodis

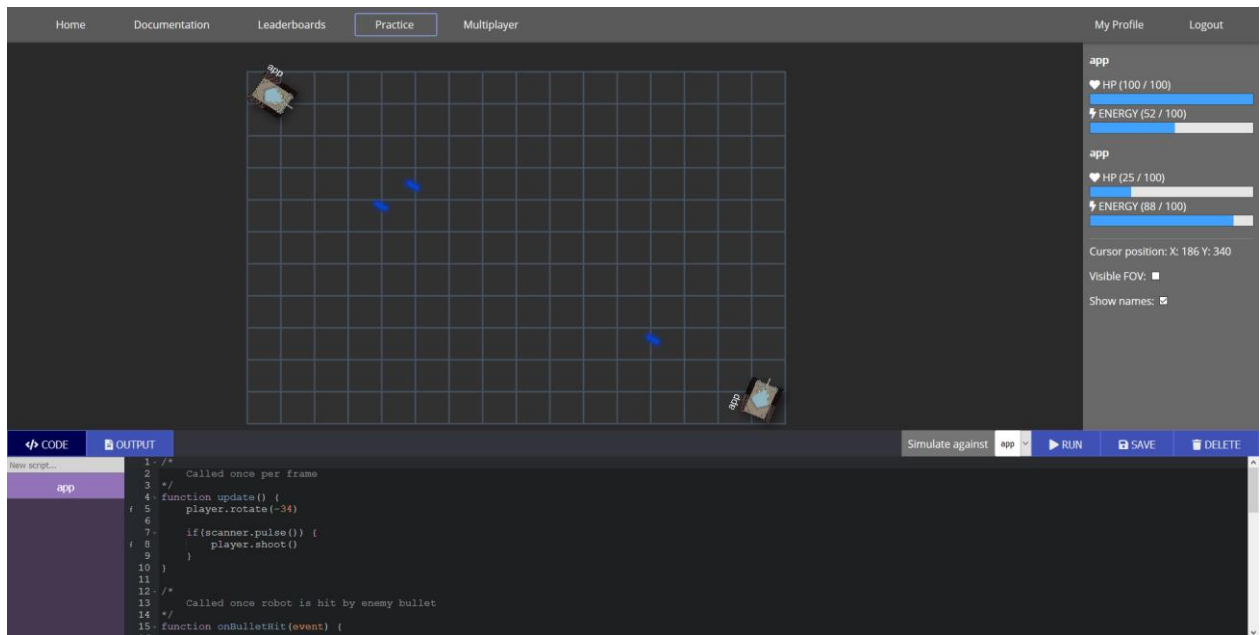
Forgot your password – paspaudus šia nuorodą naudotojas yra nukreipiamas į slaptažodžio atkūrimo puslapį.



Pav. 3.3 Slaptažodžio atkūrimo forma (<http://web-bots.herokuapp.com/users/restore-password>)

Iliustracijoje Pav. 3.3 pavaizduota slaptažodžio atkūrimo forma. Į lauką „*Email address*“ įvedamas naudotojo slaptažodis ir spaudžiamas mygtukas „*Submit*“. Jeigu naudotojas tokiu el. paštu yra užregistruotas, į el. paštą yra išsiunčiama slaptažodžio atkūrimo nuoroda ir parodomas pranešimas, jog veiksmas buvo atliktas sėkmingai.

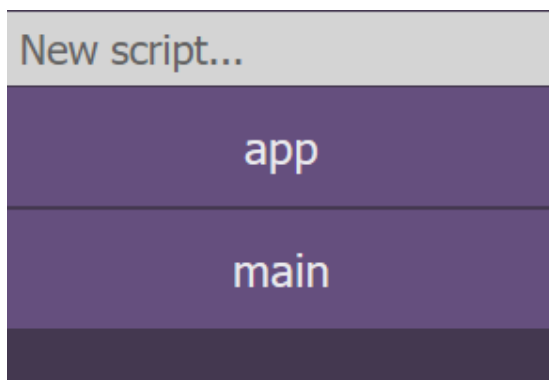
Žemiau (žr. Pav. 3.4) matomas prisijungusio naudotojo žaidimo simuliacijos langas, kuriame kuriami, trinami bei redaguojami robotui parašyti JavaScript scenarijai bei vykdomas parašytas kodas. Kodo vykdymui iš sąrašo kairėje pasirenkamas sukurtas failas, kurį vykdys robotas žemėlapiu viršuje kairėje. Taip pat iš iškrentančio sąrašo „Simulate against“ galima pasirinkti failą, kurio kodą vykdys robotas žemėlapiu apačioje dešinėje. Nepasirinkus abu robotai vykdys tą patį iš sąrašo pasirinkto failo kodą. Paspaudus mygtuką „Run“ atsiranda pranešimas, jog kodas pradėtas vykdyti. Kodo redaktoriaus aukštį galima didinti arba mažinti laikant kairį pelės mygtuką ir aukštyn arba žemyn velkant pilką dalį, kurioje išdėstyti mygtukai „Run“, „Save“ ir „Delete“.



Pav. 3.4 Žaidimo simuliacijos langas (1/2) (<http://web-bots.herokuapp.com/practice>)

Paveikslėlyje Pav. 3.5 matomi sukurti failai su pavadinimais „app“ ir „main“.

New script – naujo failo sukūrimo laukas, kuriame įvedamas failo pavadinimas ir spaudžiamas mygtukas „Enter“.



Pav. 3.5 Scenarijaus kūrimo forma

Paveikslėlyje Pav. 3.6 pavaizduoti sukurtų failų valdymo mygtukai.



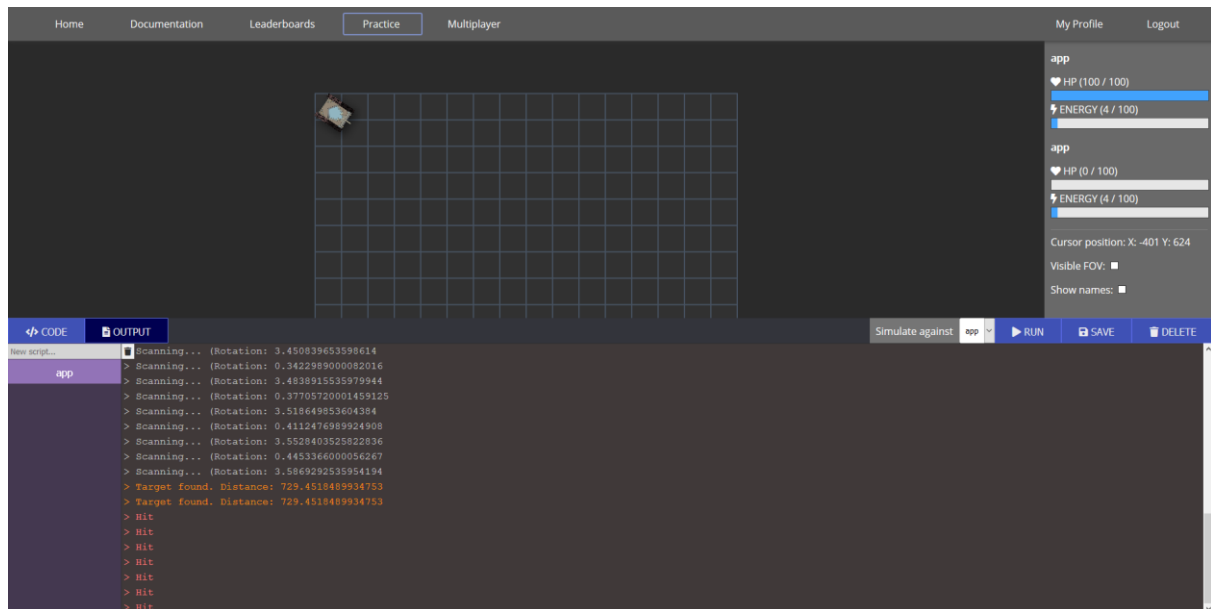
Pav. 3.6 Failų vykdymo, saugojimo ir trynimo mygtukai

Run – vykdomas pasirinkto failo kodas

Save – išsaugomi pasirinkto failo pakeitimai

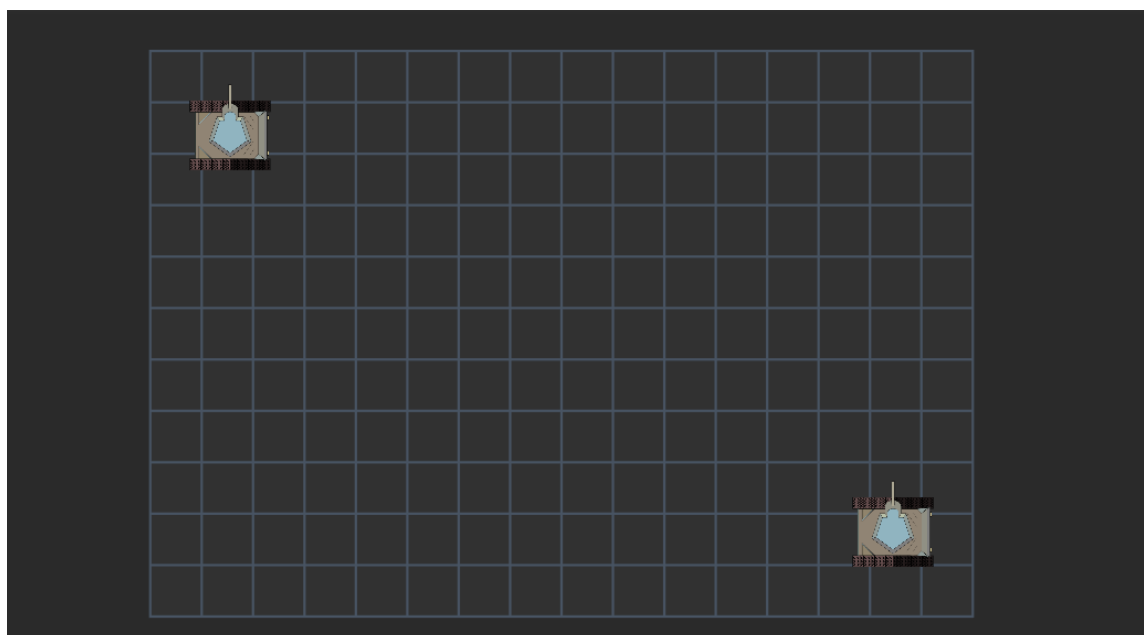
Delete – ištrinamas pasirinktas failas

Prie „CODE“ šalia esantis „OUTPUT“ mygtukas atidaro atskirą sekciją, kurioje rodomi visi išvesti pranešimai, naudojantis žaidimo API objekto *logger* funkcijomis. Galima išvesti trijų lygių pranešimus, kurie išvedimo lange atvaizduojami skirtingomis spalvomis. Langą galima išvalyti pastarojo kairės pusės viršutiniame kampe paspaudus mygtuką su šiukšliadėžės simboliu. Jeigu išvestų pranešimų yra daugiau nei 500, langas išvalomas automatiškai ir apie atliktą išvalymą parodomas pranešimas.



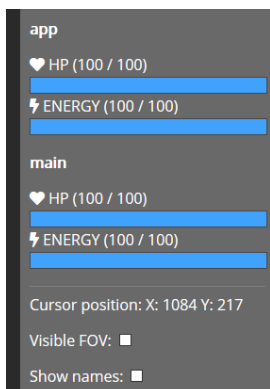
Pav. 3.7 Žaidimo simuliacijos langas (2/2) (<http://web-bots.herokuapp.com/practice>)

Žemiau (Pav. 3.8) pavaizduotas žaidimo žemėlapis, kuriame vyksta kova. Žaidimo žemėlapy galima didinti ir mažinti pelės ratuku, o taip pat vilkti ant žemėlapiu laikant kairį pelės mygtuką. Toks funkcionalumas įgyvendintas todėl, kad šitokiu žaidimo scenos valdymu galima žaidimo vaizdą pritaikyti įvairiems ekrano išmatavimams. Žaidimo arena tiek simuliacijos tiek žaidimo tinkle puslapyje atvaizduojama taip pat.



Pav. 3.8 Žaidimo arena

Tame pačiame žaidimo simuliacijos puslapio dešinėje pusėje matoma dalis (**Klaida! Nerastas nuorodos šaltinis.**) rodo robotų turimus gyvybės taškus bei energiją. Gyvybės taškai mažėja, kai šūviu į robotą pataiko priešininkas. Energija (*ENERGY*), kuri žaidimo metu vis regeneruoja, yra reikalinga šūviams – vieno šūvio vertė yra 25 taškai. Kai roboto gyvybės taškai (*HP*) pasiekia 0, žaidimo su kitu žaidėju metu pastarajam užskaitomas pralaimėjimas. Praktikos puslapyje paminėtu atveju nieko nenumatoma – simuliaciją galima vykdyti iš naujo paspaudus mygtuką „Run“



Pav. 3.9 Robotų būsenas rodantis langas

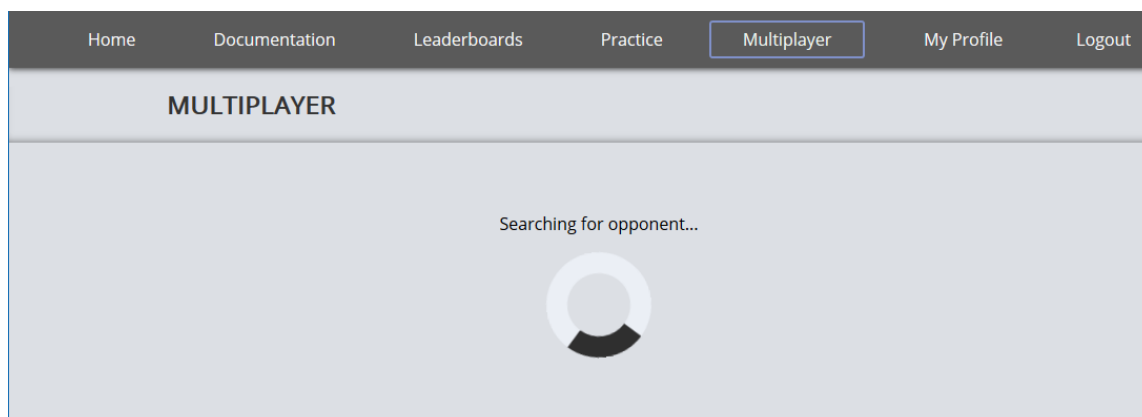
Be robotų gyvybės bei energijos taškų, apačioje, matomi:

Cursor position – pelės žymeklio koordinatės žaidimo scenoje

Visible FOV – pažymėjus šį lauką yra įjungiamas robotų matymo laukas žaidimo žemėlapyje

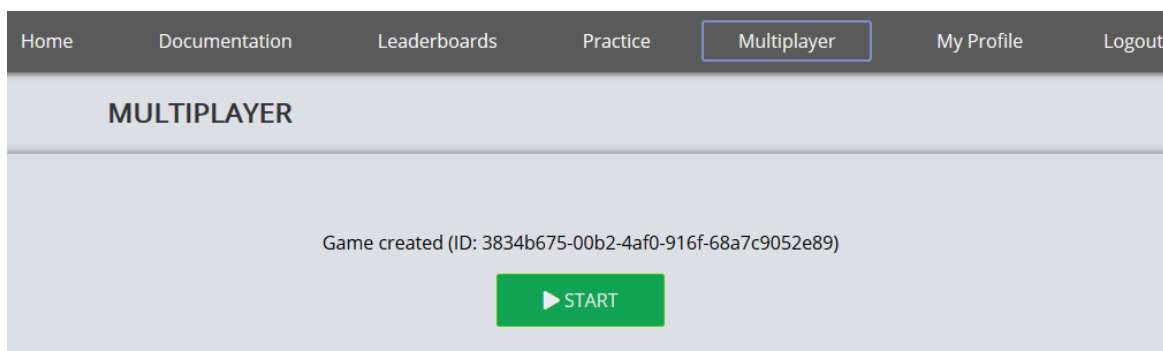
Show names – pažymėjus šį lauką prie robotų modelių rodomi žaidėjų vardai (žaidimo tinkle atveju) arba failų pavadinimai (kodo testavimo puslapyje)

Žaidėjui atsidarius „Multiplayer“ puslapį, matomas krovimo indikatorius (Žr. Pav. 3.10), o po kelių sekundžių parodomas mygtukas ir pranešimas, jog galima pradėti žaidimą.



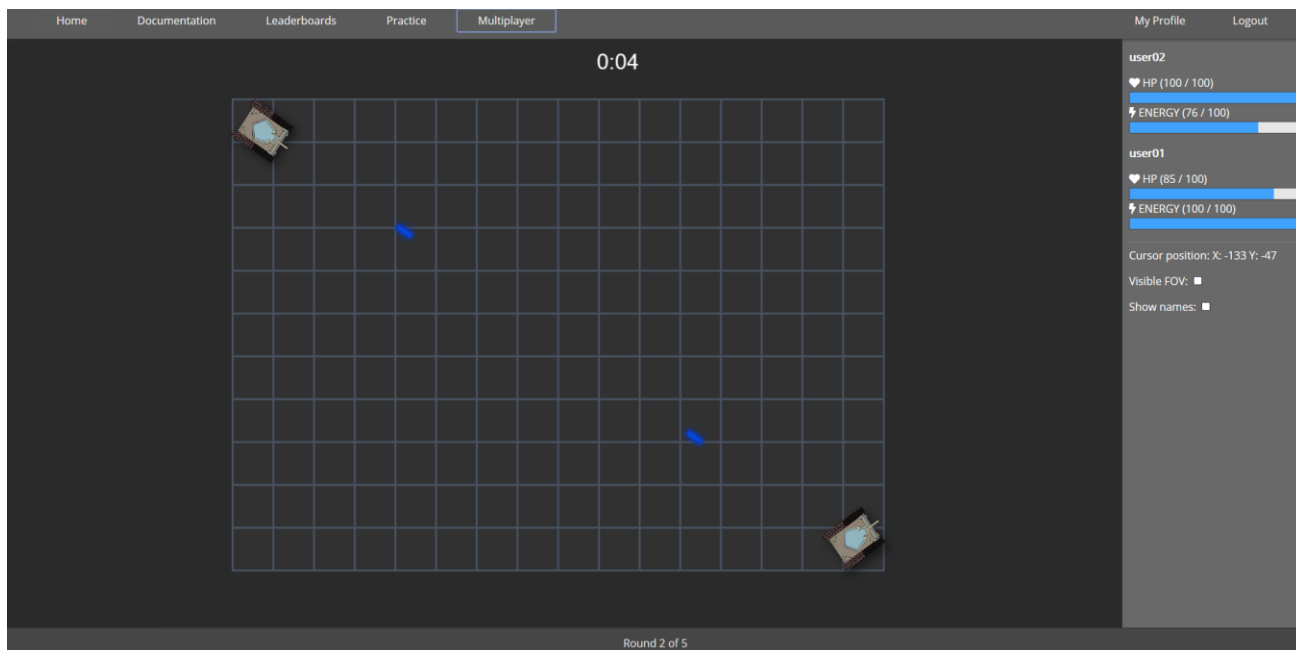
Pav. 3.10 Žaidimo tinkle ieškojimo puslapis (1/2) (<http://web-bots.herokuapp.com/lobby>)

Priešininkas žaidimui parenkamas atsitiktiniu būdu iš visų užsiregistravusių žaidėjų. Paspaudus mygtuką „Start“, naudotojas perkeliamas į kitą puslapį, kuriame vaizduojama žaidimo scena (Žr. Pav. 3.11).



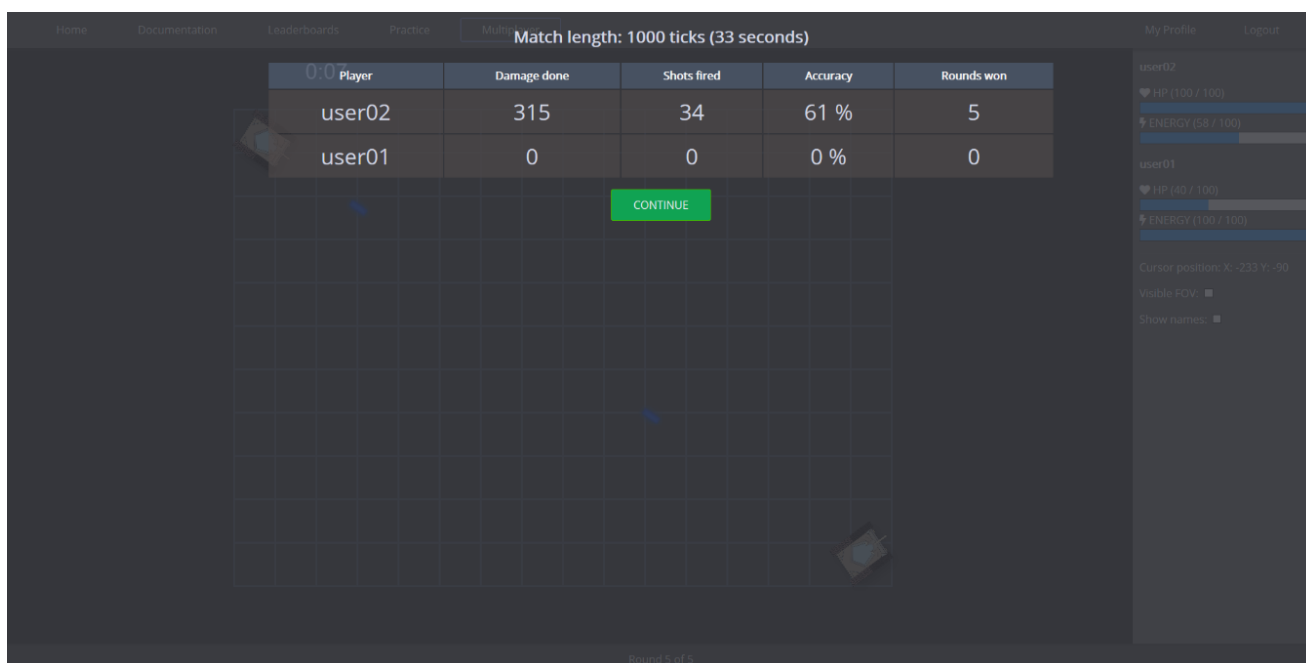
Pav. 3.11 Žaidimo tinkle ieškojimo puslapis (2/2) (<http://web-bots.herokuapp.com/lobby>)

Žaidimas tinkle susideda iš 5 raundų, kurių kiekvieno maksimali trukmė yra 45 sekundės. Praėjusių raundų skaičius matomas puslapio apačioje. Jeigu vienas iš robotų priešininkui padaro tiek žalos, jog pastarojo gyvybės taškai tampa lygūs 0 – pradedamas naujas raundas. Kiekvieno raundo pradžioje robotai atsiranda tose pačiose pozicijose ir turi po 100 gyvybės bei energijos taškų. Raundo laimėjimas yra užskaitomas tam žaidėjui, kurio robotas iki raundo pabaigos padarė priešininkui daugiausiai žalos.



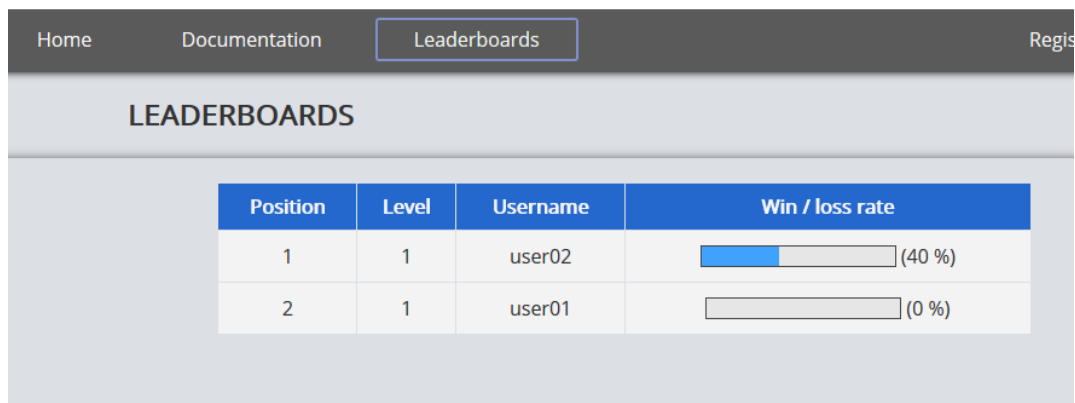
Pav. 3.12 Žaidimo tinkle puslapis (1/2)

Žaidimo pabaigoje parodoma žaidimo suvestinė. Paspaudus mygtuką „Continue“, žaidėjas nukreipiamas į savo paskyros profilio puslapį. Pasibaigus žaidimui yra atnaujinama žaidimo statistika – žaidime praleistas laikas, žaistų bei laimėtų žaidimų skaičius. Atnaujinus statistiką apskaičiuojami žaidėjo galimai atrakinti laimėjimai.



Pav. 3.13 Žaidimo tinkle puslapis (2/2)

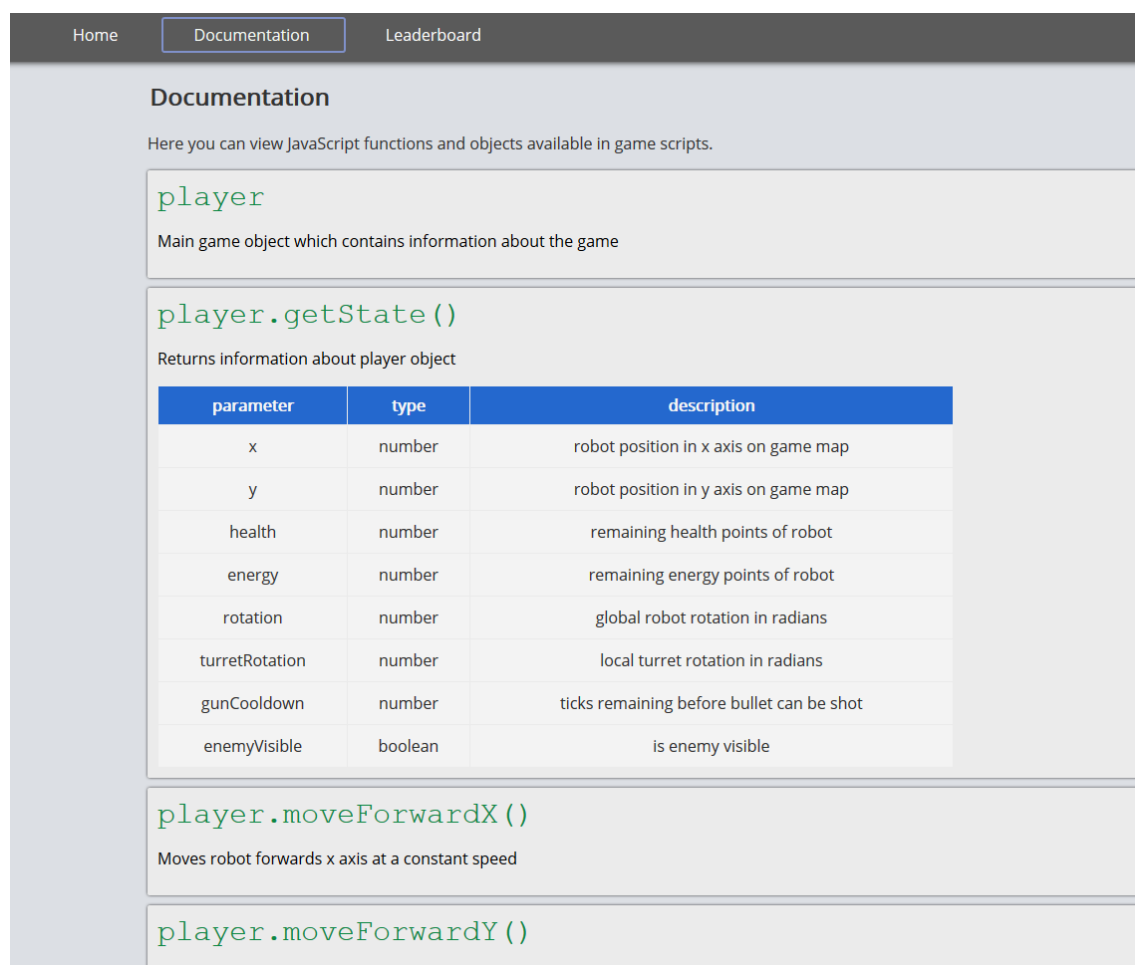
Lyderių lentos puslapyje (žr. Pav. 3.14) matoma visų žaidėjų sąrašas, surikiuotas pagal laimėjimų / pralaimėjimų procentą mažėjimo tvarka.



Position	Level	Username	Win / loss rate
1	1	user02	<div><div></div></div> (40 %)
2	1	user01	<div><div></div></div> (0 %)

Pav. 3.14 Lyderių lentos puslapis(<http://web-bots.herokuapp.com/leaderboard>)

Paveikslėlyje Pav. 3.15 pateikiama žaidimo funkcijų dokumentacijos lango iškarpa. Šiame puslapyje yra pateikiami aprašymai apie visus žaidimo kode galimus naudoti objektus bei funkcijas. Tekstas žalia spalva nurodo objektą arba jo funkciją, o po juo pateiktas pastarojo aprašymas. Lentelėse po funkcijos aprašymo yra pateikiama informacija apie iškviečiamos funkcijos grąžinamo objekto laukus.



Documentation

Here you can view JavaScript functions and objects available in game scripts.

`player`

Main game object which contains information about the game

`player.getState()`

Returns information about player object

parameter	type	description
x	number	robot position in x axis on game map
y	number	robot position in y axis on game map
health	number	remaining health points of robot
energy	number	remaining energy points of robot
rotation	number	global robot rotation in radians
turretRotation	number	local turret rotation in radians
gunCooldown	number	ticks remaining before bullet can be shot
enemyVisible	boolean	is enemy visible

`player.moveForwardX()`

Moves robot forwards x axis at a constant speed

`player.moveForwardY()`

Pav. 3.15 Žaidimo funkcijų dokumentacijos puslapis(<http://web-bots.herokuapp.com/documentation>)

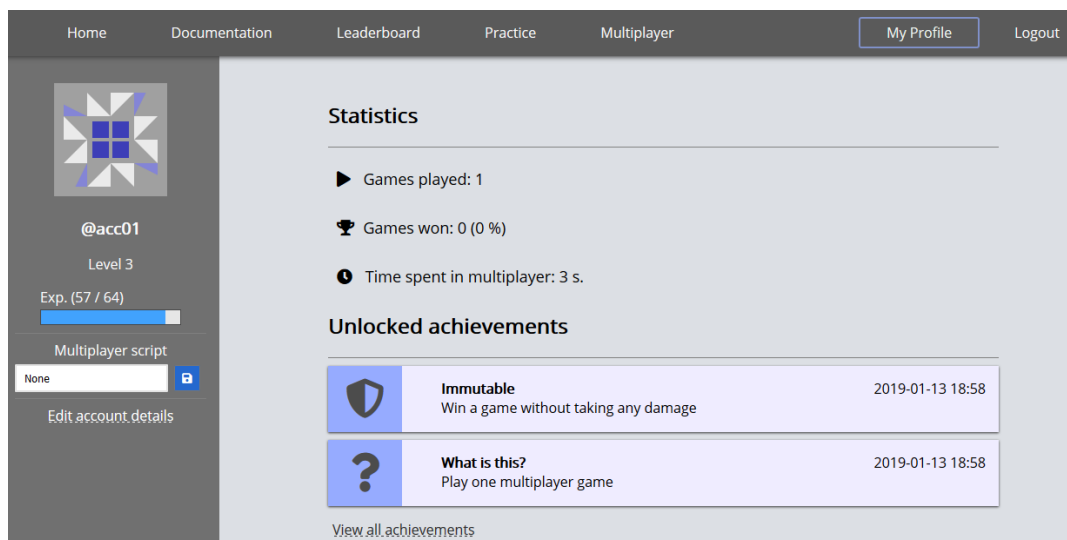
Žaidėjo profilio informacijos puslapyje yra pateikiama informacija apie laimėtų / pralaimėtų žaidimų skaičių, surinktų patirties taškų skaičių bei atrakintus laimėjimus (žr. Pav. 3.16). Puslapio kairėje, pilkesnėje sekcijoje, pateikiamas su **identicon.js** sugeneruotas naudotojo profilio paveikslėlis. Žemiau – naudotojo vardas, pasiektas lygis bei už laimėtus žaidimus bei atrakintus laimėjimus surinkti patirties taškai.

Statistics – šioje dalyje pateikiama informacija apie žaistų žaidimų skaičių

Achievements – rodomi atrakinti laimėjimai, o kiekviename iš jų nurodytas laimėjimo pavadinimas, aprašymas bei atrakinimo data.

View all achievements – paspaudus šią nuorodą naudotojas nukreipiamas į puslapį, kuriame rodomi laimėjimai, kuriuos galima atrakinti.

Multiplayer script – iškrentančiame sąraše galima pasirinkti, kuris JavaScript failas bus vykdomas žaidžiant prieš kitus žaidėjus. Pasirinkimas išsaugomas paspaudus mygtuką sąrašo kairėje. Jeigu pasirenkamas yra „None“, žaidėjas bus nerandamas kitiems žaidėjams.



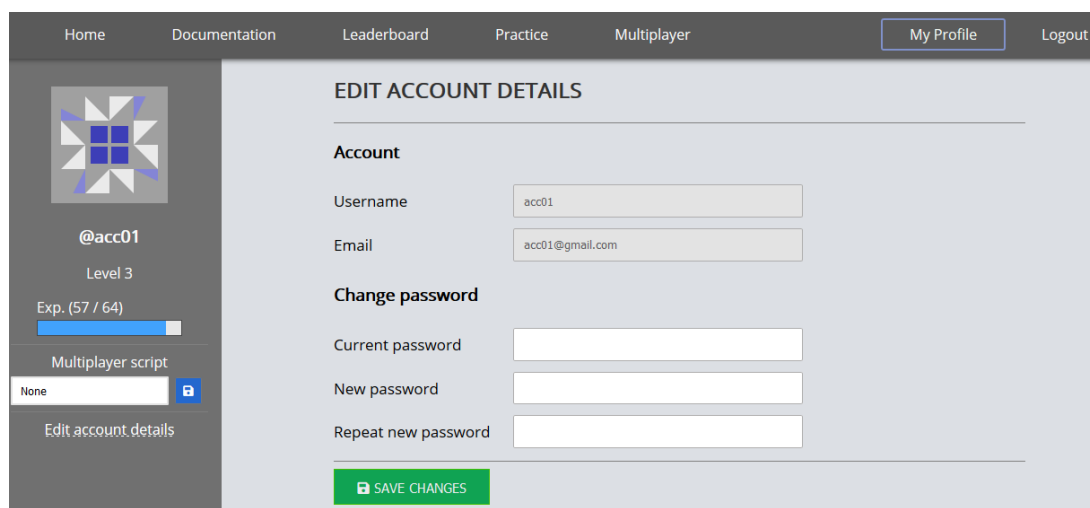
Pav. 3.16 Žaidėjo profilio informacijos puslapis (<http://web-bots.herokuapp.com/profile>)

Žaidėjo profilio puslapio kairėje pusėje paspaudus nuorodą „Edit account details“ atidaromas profilio informacijos keitimo puslapis. Pirmose dviejuose laukuose rodomas naudotojo prisijungimo vardas bei elektroninio pašto adresas. Žemiau – slaptažodžio keitimo forma.

Current password – dabartinis slaptažodis

New password – sugalvotas naujas slaptažodis

Repeat new password – pakartojamas naujas slaptažodis



Pav. 3.17 Profilio informacijos redagavimo puslapis (<http://web-bots.herokuapp.com/profile/edit-profile>)

Pagrindiniame puslapyje pateikiamas nedidelis sistemos aprašymas(Pav. 3.18). Viršuje matoma meniu juosta, kuria galima naviguoti per puslapius.

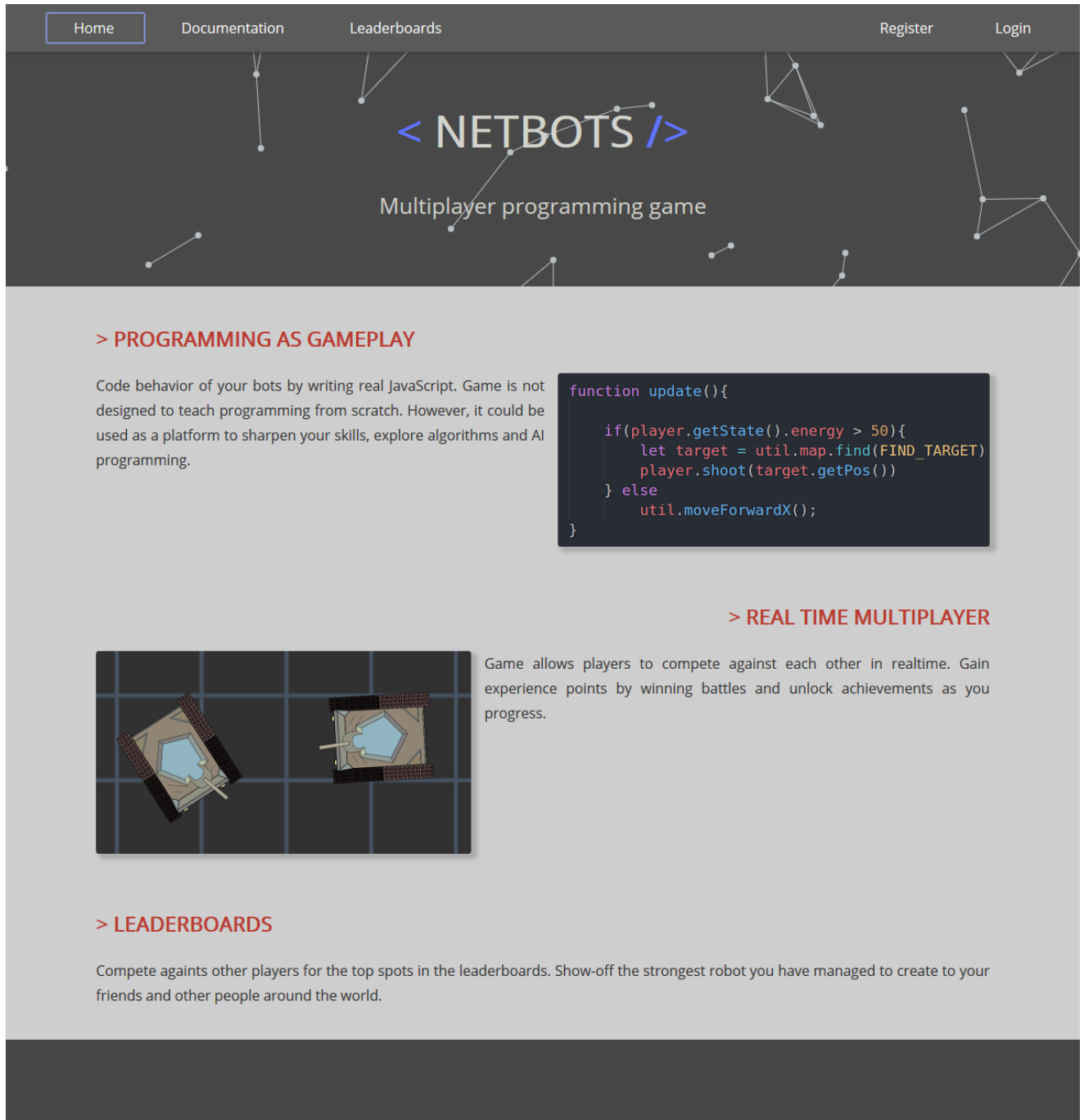
Home – į pagrindinį puslapį

Documentation – į žaidimo bei žaidėjo naudojamų JavaScript funkcijų aprašymo puslapį

Leaderboards – į žaidėjų lyderių lentos puslapį

Register – į naudotojo registracijos puslapį

Login – į naudotojo prisijungimo puslapį



Pav. 3.18 Pagrindinis puslapis(<http://web-bots.herokuapp.com/>)

4. Išvados

1. Atlikus esamų sprendimų analizę galima teigti, jog pastarieji turi savų privalumų ir trūkumų, tačiau yra gana populiarūs ir aktyviai palaikomi. Žaidime „Screeps“ realizuotas gana sudėtingas kodo vykdymas realiu laiku gerokai suprastina žaidimo greitaveiką. „Robocode“ tokios galimybės apskritai nesuteikia, išskyrus kodo įkėlimą žaidimui tinkle ir vėliau – rezultatų peržiūrą. Vis dėlto, šio semestro metu realizuotas projektas, palyginus su analizuotais sprendimais, turi gana mažai turinio ir funkcijų.
2. Atlikus technologijų analizę, pastebėta, jog *NodeJS* turi daugybę įvairių modulių, kurie gali palengvinti sistemos kūrimą. Vis dėlto, perteklinis įvairių NPM modulių naudojimas gali negatyviai paveikti sistemos greitaveiką ir kartais naudingiau tam tikras reikalingas funkcijas realizuoti pačiam. Taip pat, ne visi NPM moduliai yra visą laiką aktyviai palaikomi ir atnaujinami, gali turėti klaidų, o tai ilgainiui gali tapti realizuotos sistemos saugumo spraga.
3. Projektavimo metu buvo naudojamas MagicDraw įrankis, kurį naudojant buvo nesunkiai sukurtos panaudojimo atvejų bei sekų diagramos. Taip pat buvo naudojamas Hackolade vizualaus duomenų bazių modeliavimo programa, su kuria buvo suprojektuota *MongoDB* duomenų bazės schema bei iš jos sugeneruoti *Mongoose* modeliai.
4. Realizuojant sistemą buvo gana sudėtinga sukurti saugų bei greitai veikiantį naudotojo parašyto kodo vykdymą. Šiuo atveju, buvo pasirinktas NodeVM modulis, kuris yra skirtas sąlyginai saugiai vykdyti nepatikimą JavaScript kodą izoliuotoje aplinkoje bei riboti vykdymo laiką. Vis dėlto, pastarasis sprendimas neužtikrina visiško saugumo, o atliekamos operacijos yra pastebimai lėtos. Taip pat, žaidimo būsenos atnaujinimų siuntimui iš serverio buvo naudojamas *WebSocket* protokolas ir atnaujinimai siunčiami kiekvieną žaidimo ciklo iškvietimą. Kadangi tokiu būdu yra siunčiamas pakankamai didelis duomenų kiekis, naudotojui su lėtu interneto ryšiu žaidimo atnaujinimas bus nepastovus.
5. Atliekant sistemos testavimą buvo atrasta nemažai klaidų bei saugumo spragų, kurių didelė dalis buvo ištaisyta. Žaidimo API buvo testuojamas naudojant *Mocha* JavaScript testavimo karkasą.

5. Literatūros sąrašas

- [1] Screeps, „Introduction | Screeps Documentation,“ [Tinkle].
Available: <https://docs.screeps.com/introduction.html>.
- [2] F. Larsen, „Robocode - Robo Wiki,“ [Tinkle].
Available: <http://robowiki.net/wiki/Robocode>.
- [3] Y. Katz, „Handlebars.js: Minimal Templating on Steroids,“ Tilde Inc., [Tinkle].
Available: <https://handlebarsjs.com/expressions.html>.
- [4] R. v. d. Spuy, „Learning PIXI,“ [Tinkle].
Available: <https://github.com/kittykatattack/learningPixi#introduction>.
- [5] MongoDB, „Introduction to MongoDB - MongoDB Manual,“ [Tinkle].
Available: <https://docs.mongodb.com/manual/introduction/>.
- [6] MongoDB, „Sharding - MongoDB Manual,“ [Tinkle].
Available: <https://docs.mongodb.com/manual/sharding/>.
- [7] „About | Node.js,“ Node.js Foundation, [Tinkle].
Available: <https://nodejs.org/en/about/>.

6. Priedas

6.1. Semestro darbų suvestinė

Strazdas Karolis, IFF6-12

Veikla	Sugaištas laikas valandomis
Naudotojo kodo vykdymas serveryje	8
Žaidėjo kodo naudojamų JavaScript funkcijų ir objektų žaidime kūrimas (žaidimo API)	8
Žaidimo būsenos atnaujinimai per WebSocket ir žaidimo ciklas serveryje	8
Žaidimo sesijos kūrimas (žaidimas tinkle)	8
Stilių kūrimas bei koregavimas, puslapių pritaikymas mažiems ekranams	8
Testavimas ir atrastų klaidų taisymas	8
Žaidimo statistikos (žaidėjo patirties taškų, laimėjimų / pralaimėjimų registravimas)	6
Žaidimo scenos sukūrimas su PIXIJS	6
Žaidimo grafinių elementų kūrimas ir koregavimas	6
Laimėjimų atrakinimo sistemos kūrimas pagal žaidėjo statistiką	6
Kodo redaktoriaus integravimas puslapyje	5
Žaidimo API testavimas (<i>Mocha</i> testavimo karkasu)	5
Pavyzdinių žaidimo scenarijų kūrimas naudojant sukurtą žaidimo API	4
Žaidimo API pranešimų išvedimas ir pastaruosius atvaizduojantis langas	4
Kodo testavimo puslapis, failų kūrimo bei peržiūros forma	4
Slaptažodžio atkūrimas	3
MongoDB duomenų bazės projektavimas bei modelių aprašymas Mongoose moduliui	3
Naudotojo profilio puslapis	3
Vykstančio žaidimo tinkle statistikos rinkimas ir atvaizdavimas	3
Naudotojo autentifikacija	2
Naudotojo registracijos forma (su stiliumi)	2
Naudotojo autentifikacija	2
JavaScript scenarijų kūrimo funkcija	2
JavaScript scenarijų redagavimo funkcija	2
JavaScript scenarijų ištrynimo funkcija	2
Pagrindinis puslapis	2
Slaptažodžio keitimas	2
Naudotojo registracija su duomenų validavimu	1
Naudotojo prisijungimas su duomenų validavimu	1
Laimėjimų (visų) puslapis	1
Žaidimo API dokumentavimas	1
Projekto integravimas Heroku platformoje, automatizuotas projekto kūrimas (angl. building) iš Bitbucket repozitorijos	1
Naudotojo prisijungimo forma	0,5
Lyderių lentos puslapis	0,5
Viso:	128 val.