

Kauno technologijos universitetas

Informatikos fakultetas

Susitikimo kambarių rezervavimo sistema

Baigiamosios praktikos ataskaita

|  |  |
| --- | --- |
|  |  |
| Ignas Jasonas  Studentas | 2020-05-21 |
|  |  |
| Raimundas Bučius  Vadovas | 2020-05-21 |
|  |  |
| Doc. Giedrius Ziberkas  Dėstytojas | (parašas)  (data) |
|  |  |

Kaunas 2020

Turinys

[Paveikslėlių sąrašas 4](#_Toc40959935)

[Lentelių sąrašas 5](#_Toc40959936)

[Įvadas 6](#_Toc40959937)

[1. Analizės dalis 7](#_Toc40959938)

[1.1. Informacija apie įmonę 7](#_Toc40959939)

[1.1.1. Įmonės veiklos sritis, specializacija 7](#_Toc40959940)

[1.1.2. Įmonės vizija 7](#_Toc40959941)

[1.1.3. Įmonės misija 7](#_Toc40959942)

[1.1.4. Kokiais būdais įmonė siekia įgyvendinti savo misiją ir viziją 7](#_Toc40959943)

[1.2. Informacija apie įmonėje naudojamas IT 7](#_Toc40959944)

[1.3. Įmonės veiklos procesai. 9](#_Toc40959945)

[1.3.1. Projekto inicijavimas. 9](#_Toc40959946)

[1.3.2. Projekto koncepcijos analizė. 9](#_Toc40959947)

[1.3.3. Kainos ir investicijų analizė. 9](#_Toc40959948)

[1.3.4. Projektų vykdymas, pristatymas ir dokumentacija. 9](#_Toc40959949)

[1.3.5. Komandinis darbas. 9](#_Toc40959950)

[1.3.6. Inovacijos. 10](#_Toc40959951)

[1.3.7. Produktų vystymo procesai. 10](#_Toc40959952)

[2. Praktikos dalis 11](#_Toc40959953)

[2.1. Techninė užduotis 11](#_Toc40959954)

[2.1.1. Sistemos apibrėžimas 11](#_Toc40959955)

[2.1.2. Bendras veiklos tikslas 11](#_Toc40959956)

[2.1.3. Sistemos pagrįstumas 11](#_Toc40959957)

[2.1.4. Konkurencija rinkoje 11](#_Toc40959958)

[2.1.5. Prototipai ir pagalbinė informacija 11](#_Toc40959959)

[2.1.6. Siūlomas sprendimas 12](#_Toc40959960)

[2.1.7. Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti 12](#_Toc40959961)

[2.2. Reikalavimų specifikacija 12](#_Toc40959962)

[2.2.1. Komercinė specifikacija 12](#_Toc40959963)

[2.2.2. Sistemos funkcijos 12](#_Toc40959964)

[2.2.3. Apribojimai 13](#_Toc40959965)

[2.2.4. Duomenų ir rezultatų specifikacija 14](#_Toc40959966)

[2.2.5. Vartotojo sąsajos specifikacija 17](#_Toc40959967)

[2.2.6. Realizacijai keliami reikalavimai. 24](#_Toc40959968)

[2.2.7. Techninė specifikacija. 24](#_Toc40959969)

[2.3. Projektavimo eiga ir sistemos projektas 25](#_Toc40959970)

[2.3.1. Projektavimo valdymas ir eiga 25](#_Toc40959971)

[2.3.2. Projektavimo technologija 25](#_Toc40959972)

[2.3.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos 25](#_Toc40959973)

[2.3.4. Informacijos srautai 25](#_Toc40959974)

[2.3.5. Sistemos sudėtis 25](#_Toc40959975)

[2.3.6. Komponentai 26](#_Toc40959976)

[2.3.7. Sąsajos įvertinimas pagal vartotojo patyrimą, profesinę terminologiją 26](#_Toc40959977)

[2.3.8. Duomenų kontrolė 26](#_Toc40959978)

[2.4. Testavimo eiga ir rezultatai 26](#_Toc40959979)

[2.4.1. Vienetų testai 27](#_Toc40959980)

[2.4.2. Integraciniai testai 27](#_Toc40959981)

[2.4.3. Rankiniai testai 30](#_Toc40959982)

[2.5. Dokumentacija 31](#_Toc40959983)

[2.5.1. Apibendrintas sistemos galimybių sąrašas 31](#_Toc40959984)

[2.5.2. Galutinio vartotojo dokumentacija 31](#_Toc40959985)

[Išvados 43](#_Toc40959986)

[Literatūros sąrašas 44](#_Toc40959987)

Paveikslėlių sąrašas

[**Pav. 2.1** Panaudos atvejų diagrama 13](#_Toc40958500)

[**Pav. 2.2.** Prisijungimo langas 18](#_Toc40958501)

[**Pav. 2.3**. Susirinkimo kambarių sąrašo langas 18](#_Toc40958502)

[**Pav. 2.4.** Susirinkimo kambarių sąrašo langas mobiliesiems įrenginiams 19](#_Toc40958503)

[**Pav. 2.5.** Susirinkimo kambario rezervavimo langas 19](#_Toc40958504)

[**Pav. 2.6**. Susirinkimo kambario rezervavimo langas mobiliesiems įrenginiams 20](#_Toc40958505)

[**Pav. 2.7**. Visų rezervacijų sąrašo peržiūros langas 21](#_Toc40958506)

[**Pav. 2.8**. Visų vartotojų sąrašo peržiūros langas 21](#_Toc40958507)

[**Pav. 2.9**. Susirinkimo kambario rezervavimo formos langas mobiliesiems įrenginiams 22](#_Toc40958508)

[**Pav. 2.10**. Susirinkimo kambario rezervavimo langas planšetėms 23](#_Toc40958509)

[**Pav. 2.11**. Susirinkimo kambario informacijos peržiūros langas planšetėms 23](#_Toc40958510)

[**Pav. 2.12**. Susirinkimo kambarių sąrašo peržiūros langas planšetėms 24](#_Toc40958511)

[**Pav. 2.13** Komponentų diegimo diagrama 26](#_Toc40958512)

[**Pav. 2.14** Kodo padengimas 27](#_Toc40958513)

[**Pav. 2.15** „Storybook“ testo aprašo pavyzdys 27](#_Toc40958514)

[**Pav. 2.16** „Storybook“ komponentų testavimo sąsajos langas 28](#_Toc40958515)

[**Pav. 2.17** Netikras action objektas 29](#_Toc40958516)

[**Pav. 2.18** „Enzyme“ ir „Jest“ bibliotekų testo pavyzdys 29](#_Toc40958517)

[**Pav. 2.19** „Redux“ ir „Jest“ bibliotekso testo pavyzdys 30](#_Toc40958518)

[**Pav. 2.20** Bug‘o užpildymo pavyzdys 30](#_Toc40958519)

[**Pav. 2.22** Prisijungimo langas 31](#_Toc40958520)

[**Pav. 2.23** Susitikimo kambarių sąrašo langas 32](#_Toc40958521)

[**Pav. 2.24** Rezervacijos kūrimo langas 32](#_Toc40958522)

[**Pav. 2.25** Rezervacijos redagavimas 33](#_Toc40958523)

[**Pav. 2.26** Rezervacijos informacija 34](#_Toc40958524)

[**Pav. 2.27** Rezervacijos pasikartojimo opcijos 35](#_Toc40958525)

[**Pav. 2.28** Susitikimo narių pridėjimas 36](#_Toc40958526)

[**Pav. 2.29** Laisvalaikio įrangos langas 37](#_Toc40958527)

[**Pav. 2.30** Visų rezervacijų langas 37](#_Toc40958528)

[**Pav. 2.31** Bendras susitikimo kabarių ir laisvalaikio įrangos sąrašas 38](#_Toc40958529)

[**Pav. 2.32** Vartotojų sąrašas 38](#_Toc40958530)

[**Pav. 2.33** Vartotojo meniu 39](#_Toc40958531)

[**Pav. 2.34** Sistemos klaidos pranešimo lango fiksavimas 39](#_Toc40958532)

[**Pav. 2.35** Sistemos klaidos pranešimo sukūrimas 40](#_Toc40958533)

[**Pav. 2.36** Planšetės susitikimo kambario informacija 40](#_Toc40958534)

[**Pav. 2.37** Rezervacijos kūrimas planšetės lange 41](#_Toc40958535)

[**Pav. 2.38** Rezervacijos kūrimas planšetės lange 41](#_Toc40958536)

[**Pav. 2.39** Planšetės lango susitikimo kambarių sąrašas 42](#_Toc40958537)

Lentelių sąrašas

[**Lentelė 2.1** Prisijungimo duomenys 14](#_Toc40958437)

[**Lentelė 2.2** Rezervacijos kūrimo duomenys 14](#_Toc40958438)

[**Lentelė 2.3** Klaidos pranešimo duomenys 14](#_Toc40958439)

[**Lentelė 2.4** Susirinkimo kambario elemento sąraše peržiūros rezultatai 15](#_Toc40958440)

[**Lentelė 2.5** Rezervacijos elemento sąraše rezultatai 15](#_Toc40958441)

[**Lentelė 2.6** Laisvalaikio įrangos elemto sąraše rezultatai 16](#_Toc40958442)

[**Lentelė 2.7** Vartotojo elemento sąraše rezultatai 16](#_Toc40958443)

[**Lentelė 2.8** Sėkmingos kambario rezervacijos/ų sukūrimo rezultatai 17](#_Toc40958444)

[**Lentelė 2.9** Susirinkimų kambarių rezervacijos elemento sąraše rezultatai 17](#_Toc40958445)

[**Lentelė 2.10** Techninė specifikacija 24](#_Toc40958446)

Įvadas

Darbe pristatoma sistema, kuri skirta įmonės susitikimo kambarių ir laisvalaikio įrangos rezervavimui siekiant pagerinti įmonės susitikimo kambarių valdymą ir organizavimą. Dabartinė naudojama sistema yra nelanksti, turinti klaidų ir sukurta naudojant senesnes technologijas, todėl nuspręsta kurti atnaujintą sistemą, kuri būtų lengvai plečiama. Susitikimo kambario arba laisvalaikio įrangos rezervacija sukuriama pasinaudojant sistemoje esančia laiko juosta, joje pažymint norimą laiką ir užpildant kitus nustatymus. Rezervacija matoma visiems darbuotojams, taip pat matoma prie susitikimo kambario esančioje planšetėje. Susitikimą galima atšaukti, peržiūrėti, redaguoti.

Ataskaitoje pateikiama išsami kuriamos sistemos, rinkos konkurentų analizė, įvardijami funkciniai, nefunkciniai reikalavimai, naudojamos technologijos ir įrankiai. Pateikiamas, sistemos testavimo procesas ir rezultatai, galutinio sistemos vartotojo ir diegimo vadovai. Darbo pabaigoje viskas apibendrinama ir pateikiamos išvados.

# Analizės dalis

## Informacija apie įmonę

### Įmonės veiklos sritis, specializacija

Įmonė „TeleSoftas“ yra programinės įrangos kūrimo bei konsultacijos įmonė, kurianti mobiliąją, „FinTech“ , „Big Data“, komunikavimo ir kitiems sektoriams priklausančią programinę įrangą. Jau 15 metų veikianti įmonė, turinti 6 ofisus Europoje ir Amerikoje, šiandien turi milijonus vartotojų ir teikia programų kūrimo paslaugas klientams iš viso pasaulio.

### Įmonės vizija

Suteikiant geriausius sprendimus bei inovacijas, augti ne tik patiems, bet ir klietams bei partneriams

### Įmonės misija

Suteikiant įrankius padėti verslams bei IT klientams tobulėti

### Kokiais būdais įmonė siekia įgyvendinti savo misiją ir viziją

Kuriant nuolat tobulėjačių programinės įrangos kūrimo specialistų bendruomenę, stengiantis pritraukti tarptautinius klientus, dalijantis savo patirtimi tarpusavyje siekiant įgyvendinti projektus.

„TeleSoftas“ laikosi „Step forward“ tasyklės – padėti klientams atrasti ir sukurti sprendimus išnaudojant naujas technologijas ir praktikas. Šios įmonės sukurtais sprendimais naudojasi įvairių industrijų sričių įmonės: logistikos, transporto, telekomunikacijų, sveikatos priežiūros, finansų ir kitos.

„Telesoftui“ yra be galo svarbi darbuotojų laimė, todėl stengiamasi atsižvelgti į visų poreikius ir problemas, taip padidinat iniciatyvos lygį. Plėtojamas patirties dalinimąsis, bendruomeniškumas. Darbuotojams sudaromos visos galimybės įsitraukti į veiklą bei priimti efektyviausius sprendimus – taip skatinant tobulėti.

## Informacija apie įmonėje naudojamas IT

Frontend programavimo technologijos/kalbos:

* JavaScript – objektiškai orientuota skriptų [programavimo kalba](https://lt.wikipedia.org/wiki/Programavimo_kalba), besiremianti prototipų principu.
* TypeScript – interpretuojama programavimo kalba kaip Javascript, tik .
* ReactJS – JavaScript kalba paremta biblioteka-karkasas.
* AngularJS – JavaScript kalba paremtas karkasas.

Backend programavimo technologijos/kalbos:

* C# – [objektiškai](https://lt.wikipedia.org/wiki/Objektinis_programavimas) orientuota [programavimo kalba](https://lt.wikipedia.org/wiki/Programavimo_kalba), sukurta [Microsoft](https://lt.wikipedia.org/wiki/Microsoft) kompanijoje kaip dalis [.NET](https://lt.wikipedia.org/wiki/.NET) iniciatyvos
* Java –  bendrosios paskirties, objektiškai orientuota [programavimo kalba](https://lt.wikipedia.org/wiki/Programavimo_kalba)
* PHP –  plačiai paplitusi [dinaminė](https://lt.wikipedia.org/w/index.php?title=Dinamin%C4%97_programavimo_kalba&action=edit&redlink=1) [interpretuojama](https://lt.wikipedia.org/w/index.php?title=Interpretuojama_kalba&action=edit&redlink=1) [programavimo kalba](https://lt.wikipedia.org/wiki/Programavimo_kalba)
* Go arba Golang – kompiliuojama objektinio programavimo kalba.
* Scala – 2003 m. sukurta [programavimo kalba](https://lt.wikipedia.org/wiki/Programavimo_kalba), sukurta glaustai išreikšti dažniausius [programavimo šablonus](https://lt.wikipedia.org/w/index.php?title=Programavimo_%C5%A1ablonai&action=edit&redlink=1), struktūras.

Mobilios aplikacijos:

* Android
  + Kotlin – „JetBrains“ sukurta funkcinė ir objektinė programavimo kalba.
  + Java – bendrosios paskirties, objektiškai orientuota [programavimo kalba](https://lt.wikipedia.org/wiki/Programavimo_kalba).
* iOS
  + Objective-C – objektinė programavimo kalba.
  + Swift – bendros paskirties programavimo kalba, skirta „Apple“ įrenginių produktams kurti.

Duomenų bazės:

* Oracle – reliacinė duomenų bazė.
* MsSQL – reliacinė duomenų bazė.
* MySQL – reliacinė duomenų bazė.
* MongoDB – nereliacinė duomenų bazė.

Versijavimo sistemos:

* BitBucket
* GitLab
* GitHub

Projektų valdymo sistemos:

* Redmine
* Jira

Kadangi įmonė bendradarbiauja su daug klientų ir partnerių, visems viena versijavimo sistema gali netikti, todėl yra naudojamos kelios.

## Įmonės veiklos procesai.

### Projekto inicijavimas.

Projektas inicijuojamas, kai klientas kreipiasi į įmonę su idėja. Tada procesas būna toks:

* Surandamas geriausias būdas kliento idėjai įgyvendinti.
* Identifikuojamos galimos problemos.
* Pasirenkama geriausiai tinkanti programinė įranga.
* Nustatomi kliento poreikiai ir surandamas galutinis sprendimas.

### Projekto koncepcijos analizė.

Projekto koncepcijos analizės procesas:

* Įvykdoma kliento verslo logikos arba modelio analizė
* Įvertinami turimi verslo valdymo įrankiai
* Išskiriami galimos programinės įrangos pliusai, minusai. Pagal ją sudaromas funkcionalumų sąrašas

### Kainos ir investicijų analizė.

Peržiūrint sudarytą funkcionalumų sąrašą, jį per kelias projekto vertininimo susitikimus analizuoja su projekto įgyvendinimu susiję asmenys: vyresnieji programuotojaim, architektai, dizaineriai, verslo klientų vadybininkai, analitikai. Įvertinamas sistemos sudėtingumas ir apimtis. Nustatomas laikas ir sąnaudos skirtos projekto sprendimui realizuoti. Pagal suformuojamas ir klientui pateikiamas kainos/investicijos pasiūlymas.

### Projektų vykdymas, pristatymas ir dokumentacija.

Projektų vykdymo, pristatymo ir dokumentavimo procesas:

* Paruošiamos komponentų, architektūros schemos
* Sudaromas projekto kūrimo eiga, žemėlapis
* Nustatomi funkcionalumų įgyvendinimo, jų pristatymo datos/terminai
* Sukuriamos atskiros užduotys, kurias išsiskirsto programuotojai ir dizaineriai
* Įgyvendinus numatytą funckionalumą, jis dokumentuojamas pagal iš anksto su klientu sutartą struktūrą.

### Komandinis darbas.

Komandinis darbas vyksta naudojant „Agile“ metodiką ir „Scrum“ (2)karkasą. Visos komandos dirba „Sprint‘ais“ (paprastai dviejų savaičių trukmės, bet tai gali keistis priklausomai nuo užduočių kiekio) ir naudoja „Scrum“ procesus (kiekvieną dieną vykstantys „Stand-up“ susirinkimai, po pasibaigusio „Sprin‘to“, jo peržiūra ir kito „Sprint‘o“ planavimas, pasiekto progreso demonstravimas, retrospektyvos). Užduotims kontroliuoti naudojamos projektų valdymo sistemos „Jira“ arba „Redmine“.

### Inovacijos.

Papildytos realybės inovatyvūs projektai:

* „VR Inner Child“ - akrofobijai (aukščio baimei) gydyti skirtas virtualios realybės sprendimas, sukurtas su psichologų pagalba, kurie padėjo nuspręsti, kaip perkelti psichoterapines technikas į virtualią realybę.
* „VR Medusa“ – virtualios realybės verslo komunikacijos platforma, kuri leidžia efektyviai komunikuoti virtualioje realybėje.

### Produktų vystymo procesai.

Produktų vystymo procesų seka:

* Idėjos sugalvojimas.
* Idėjos realizavimo planavimas.
* Programinės įrangos įgyvendinimas ir testavimas.
* Marketingo strategijos parengimas ir vystymas.
* Verslo analizė.
* Sukurtos įrangos testavimas rinkoje.
* Paleidimas į rinką.

# Praktikos dalis

## Techninė užduotis

### Sistemos apibrėžimas

Sistema „Booky“, skirta „Telesoftas“ įmonės, Vilniaus ir Kauno ofisų susirininkimo kambarių bei laisvalaikio zonų rezervavimams atlikti, stebėti jų užimtumą. Priėjimą prie sistemos gauna kiekvienas „Telesoftas“ įmonės darbuotojas. Sistema turi du pagrindinius naudojimo režimus. Pirmasis yra darbuotojo/administratoriaus naudojimo režimai, kai prie sistemos prisijungęs vartotojas mato visų kambarių užimtumą, gali juos rezervuoti, stebėti ir atšaukti savo sukurtas rezervacijas. Kitas režimas yra susirinkimo kambario naudojimo režimas. Šiuo atveju prie sistemos prisijungia vartotojas su specialia role ir kitokia nei pirmojo režimo vartotojo sąsaja yra rodoma konkretaus kambario informacija, su galimybe jį rezervuoti. Pastarąjį režimą naudoja specialios planšetės, prikabintos prie kiekvieno susirinkimo kambario įėjimo.

Mano darbo dalis yra sukurti šios sistemos vartotojo sąsają.

### Bendras veiklos tikslas

Projekto užsakovas yra įmonė „Telesoftas“ daranti šią sistemą savo darbuotojų apmokymo, darbo efektyvumo, organizavimo gerinimo tikslais.

### Sistemos pagrįstumas

Ši sistema pagrinde yra kuriama kaip šiuo metu naudojamos, senesnės sistemos pagerinimas. Ankstesnė sistema yra parašyta senesnėmis tecnhologijomis, todėl yra sunkiau išplečiama ir palaikoma. Norint pridėti naujo funkcionalumo, buvo nuspręsta kurti naują sistemą iš pagrindų, pritaikant naujausias technologijas.

### Konkurencija rinkoje

Konkurencijos galime sakyti nėra, nes ši sistema yra kuriama tik „Telesoftas“ įmonėje, todėl į rinką ji net nepakliūna. Bet tokių sistemų rinkoje yra tikrai nemažai. Populiariausi pavyzdžiai būtų „SyncSign“, „Meetio“, „YArooms“. Šios sistemos suteikia beveik tokias pačias funkcijas, su šiek tiek geresnia vartotojų sąsaja mobiliems įrenginiams.

### Prototipai ir pagalbinė informacija

Naują sistemą kurti buvo nuspręsta dėl šių priežasčių:

* Senoji sistema yra parašyta Angular technologija, kuomet visi frontend programuotojai specializuojasi į React (3) technologiją.
* Į seną sistemą pridėti naujų funkcijų yra ypač sunku, dėl menkos kokybės kodo.
* Senoje sistemoje yra palikta klaidų, kurioms taisyti prireiktų skirti nemažai laiko ir resursų.
* Senoji sistema nėra pritaikyta mobiliems įrenginiams naudoti

### Siūlomas sprendimas

Sistema rengiama pritaikant naujausias technologijas ir geras praktikas, kad būtų lengva ją palaikyti ir atnaujinti. Vartotojams stengiamasi sukurti efektyvi, patogi naudotis ir mobilius įrenginius palaikanti grafinė sąsaja. Visa grafinė sąsają kurti iš naujo, panaudojant kai kuriuos dizainus iš senesnės sistemos.

### Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti

Sistema sudaryta iš trijų posistemių: vartotojo posistemės, administratoriaus posistemės ir kambario posistemės.

Kuriant sistemą dirbo viena mišri komanda, kurią sudarė: 2 „front-end“ programuotojai, „back-end“ programuotojas, testuotojas ir projektų vadovas. Trukmė atlikti projektą nėra ribojama, dėl šiam projektui skiriamo mažesnio prioriteto.

## Reikalavimų specifikacija

### Komercinė specifikacija

Projekto užsakovas – UAB „Telesoftas“

Projekto vykdytojai:

* Projektų vadovas
* 2 Front-end programuotojai (Aš – realizuoju pagrindinę vartotojo sąsajos logiką, kitas programuotojas – realizuoja vartotojo sąsajos logiką planšetiniams įrenginiams)
* 1 Back-end programuotojas (Realizuoja serverinės dalies sprendimą)
* 1 Testuotojas (Testuoja visos sistemos sprendimą)

Produkto vartotojai – įmonės „Telesoftas“ darbuotojai

Projekto realizacijos laiko ir kainos apribojimai – projekto įvykdymui laikas neribojamas

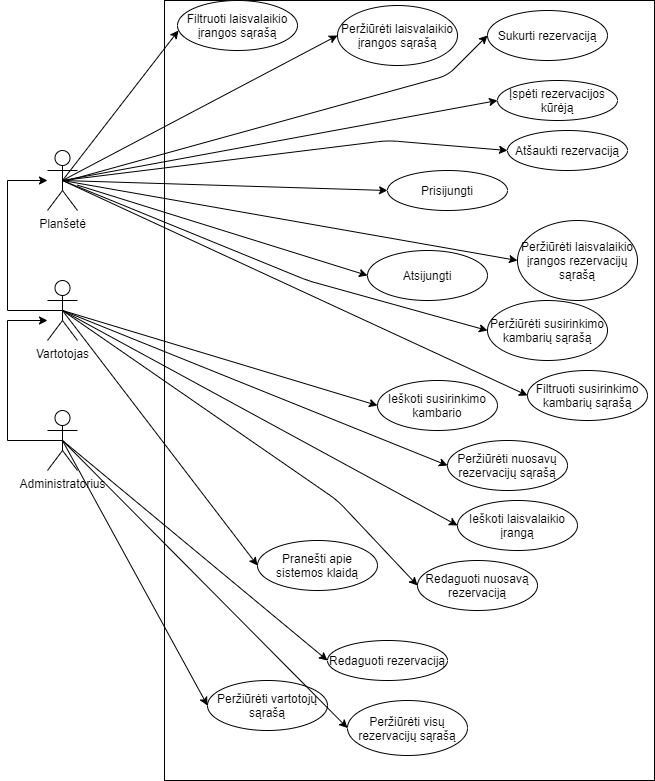
Pagrindinės priemonės:

* Backend - PHP programavimo kalba, Symphony karkasas
* Frontend - HTML, CSS, JavaScript programavimo kalbos ir ReactJs karkasas.

### Sistemos funkcijos

Funkciniai reikalavimai:

* Prisijungimas
* Atsijungimas
* Rezervacijos kūrimas
* Rezervacijos atšaukimas
* Rezervacijos redagavimas
* Susirinkimo kambarių sąrašo peržiūra
* Susirinkimo kambario rezervacijų sąrašo peržiūra
* Susirinkimo kambarių paieška
* Susirinkimo kambarių sąrašo filtravimas
* Laisvalaikio įrangos sąrašo peržiūra
* Laisvalaikio įrangos rezervacijų sąrašo peržiūra
* Laisvalaikio įrangos paieška
* Laisvalaikio įrangos sąrašo filtravimas
* Funkcija įspėti su rezervacijos kūrėją
* Vartotojų sąrašo peržiūra
* Nuosavų rezervacijų sąrašo peržiūra
* Visų rezervacijų sąrašo peržiūra
* Sistemos klaidos pranešimo funkcionalumas



**Pav. 2.1** Panaudos atvejų diagrama

### Apribojimai

Sistemos apribojimai:

* Sistema turi būti apsaugota SSL sertifikatu
* Sistema turi būti talpinama tik iš įmonės tinklo prieinamame serveryje
* Prie sistemos gali prisijungti tik įmonės „Telesoftas“ darbuotojams
* Rezervacijos negali duplikuotis laiko atžvilgiu
* Sistemos prisijungimo funkcija turi naudoti LDAP protokolą.

### Duomenų ir rezultatų specifikacija

|  |  |  |
| --- | --- | --- |
| Laukas | Tipas | Pavyzdys |
| email | String | vardenis@gmail.com |
| password | String | slaptažodis |

**Lentelė 2.1** Prisijungimo duomenys

|  |  |  |
| --- | --- | --- |
| Laukas | Tipas | Pavyzdys |
| selectedDate | Date | 2020-01-01 |
| frequency | String | Day |
| interval | Int | 1 |
| end | String | endsOn |
| repeatTimes | Int | 1 |

**Lentelė 2.2** Rezervacijos kūrimo duomenys

|  |  |  |
| --- | --- | --- |
| Laukas | Tipas | Pavyzdys |
| image | String | dataUrl |
| message | String | This is bad |

**Lentelė 2.3** Klaidos pranešimo duomenys

|  |  |  |
| --- | --- | --- |
| Laukas | Tipas | Pavyzdys |
| id | Int | 20 |
| name | String | Room name |
| size | Int | 8 |
| updated\_at | Date | 2019-12-05T08:40:37+00:00 |
| reservations | Reservation array |  |
| type | String | room |

**Lentelė 2.4** Susirinkimo kambario elemento sąraše peržiūros rezultatai

|  |  |  |
| --- | --- | --- |
| Laukas | Tipas | Pavyzdys |
| asset | Room |  |
| description | String | Reservation description |
| end\_date | Date | 2019-12-05T11:30:00+00:00 |
| id | Integer | 3 |
| owner | User |  |
| settings | Settings |  |
| start\_date | Date | 2019-12-05T10:45:00+00:00 |
| state | String | Active |
| type | String | recurring |

**Lentelė 2.5** Rezervacijos elemento sąraše rezultatai

|  |  |  |
| --- | --- | --- |
| Laukas | Tipas | Pavyzdys |
| id | Int | 1 |
| name | String | PS4 |
| description | String | PS4 laisvalaikio zona |
| type | String | item |
| created\_date | Date | 2020-01-21T08:18:22+00:00 |
| updated\_at | Date | 2021-01-21T08:18:22+00:00 |
| reservations | Reservation array |  |

**Lentelė 2.6** Laisvalaikio įrangos elemto sąraše rezultatai

|  |  |  |
| --- | --- | --- |
| Laukas | Tipas | Pavyzdys |
| id | Int | 1 |
| email | String | [Admin@localhost.com](mailto:Admin@localhost.com) |
| roles | String array | [„ROLE\_ADMIN“, „ROLE\_USER“] |
| username | String | Admin |
| profile | Profile |  |

**Lentelė 2.7** Vartotojo elemento sąraše rezultatai

|  |  |  |
| --- | --- | --- |
| Laukas | Tipas | Pavyzdys |
| created | Date array | [„2020-04-12“, „2020-05-12“] |
| excluded | Date array | [„2020-06-01“] |
| total | Total |  |

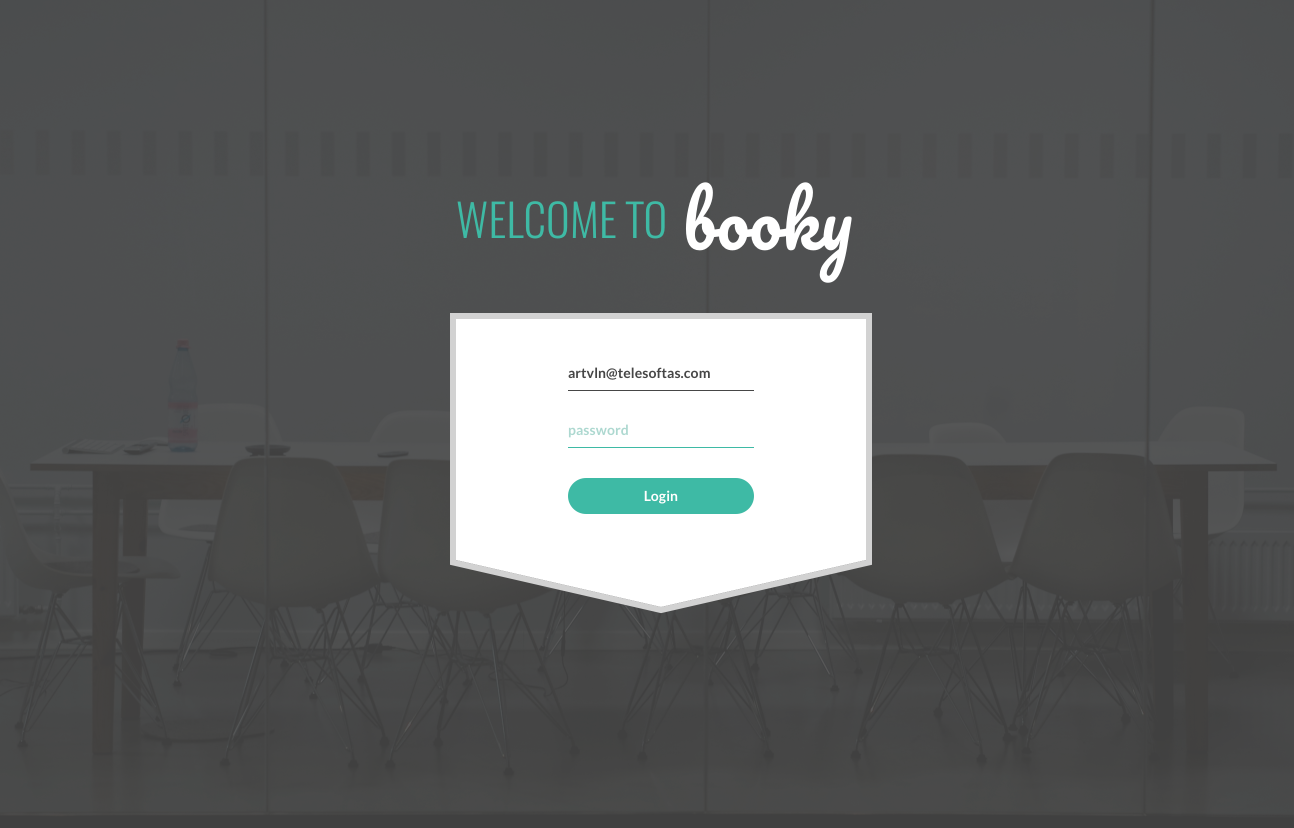
**Lentelė 2.8** Sėkmingos kambario rezervacijos/ų sukūrimo rezultatai

|  |  |  |
| --- | --- | --- |
| Laukas | Tipas | Pavyzdys |
| id | Int | 1587 |
| description | String | Meeting description |
| end\_date | Date | 2020-04-11T14:30:00+00:00 |
| start\_date | Date | 2020-04-11T14:00:00+00:00 |
| State | String | active |
| type | String | recurring |
| owner | User |  |
| settings | Settings |  |

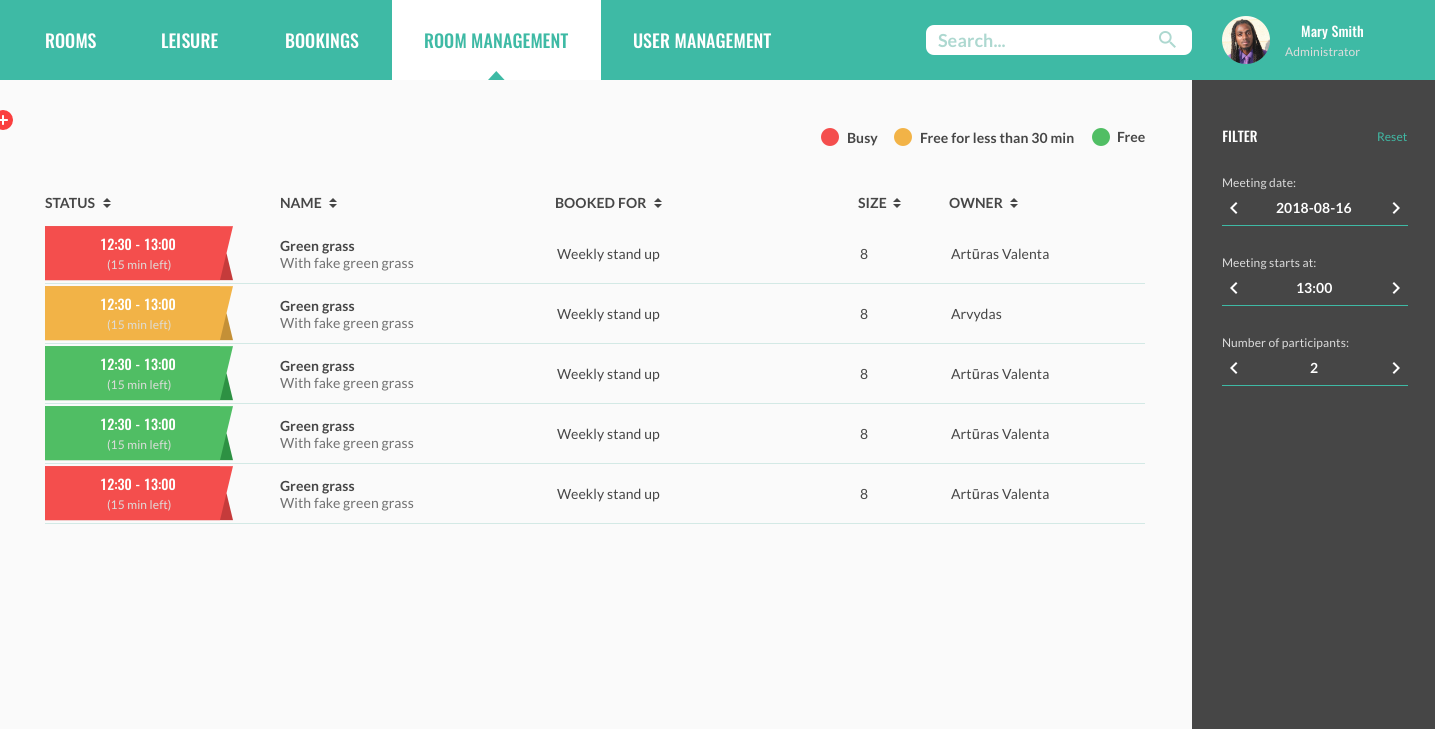
**Lentelė 2.9** Susirinkimų kambarių rezervacijos elemento sąraše rezultatai

### Vartotojo sąsajos specifikacija

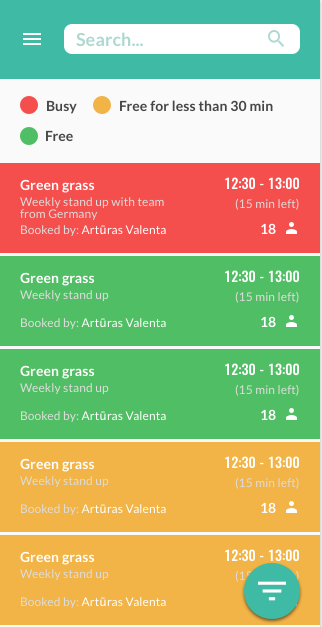
Kadangi sistemos pagrindinė paskirtis yra susirinkimo kambario / laisvaikio zonos rezervavimas, didelė dalis darbo orientuota į patogumą ir paprastumą rezervuojant kambarį tam tikram laikui.



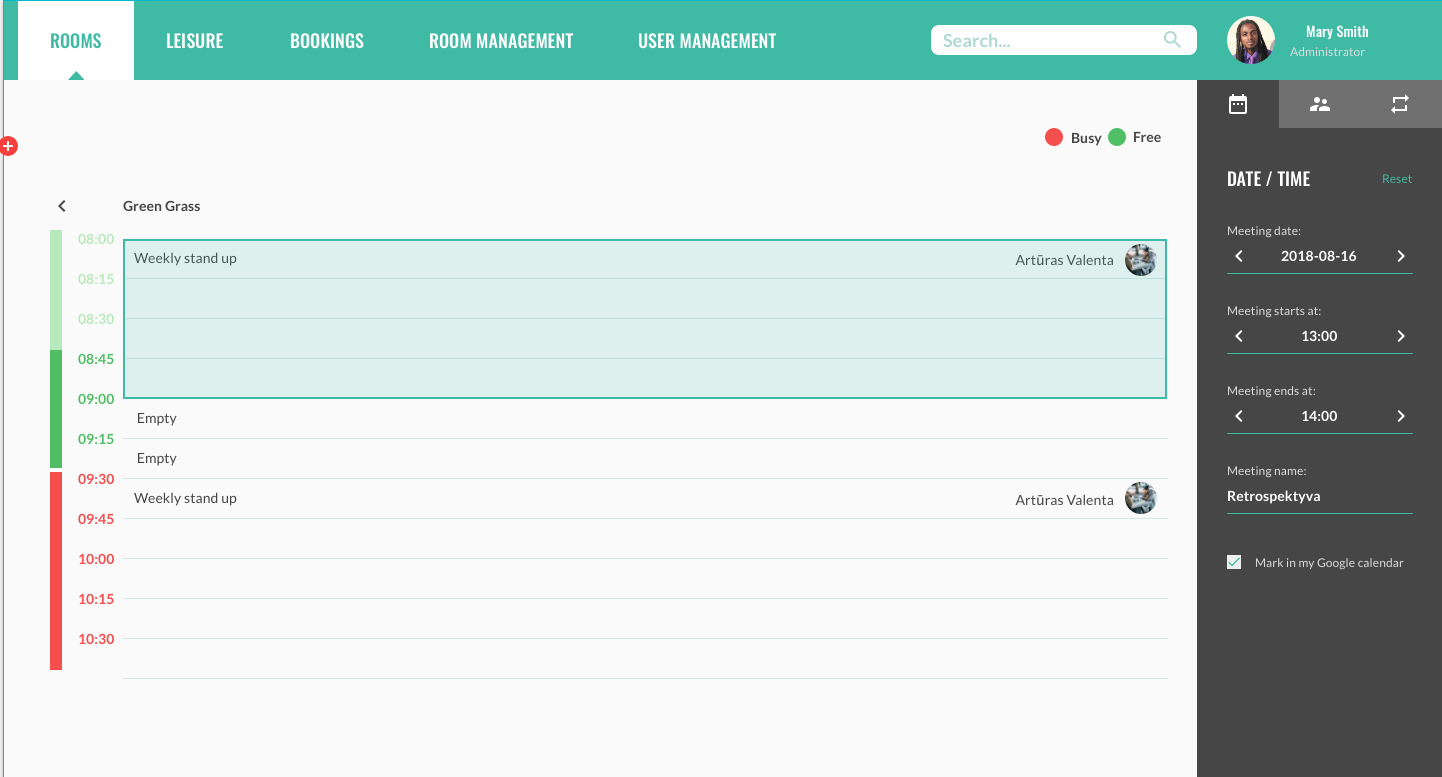
**Pav. 2.2.** Prisijungimo langas



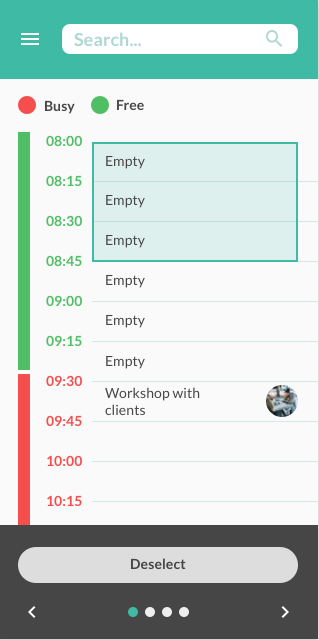
**Pav. 2.3**. Susirinkimo kambarių sąrašo langas



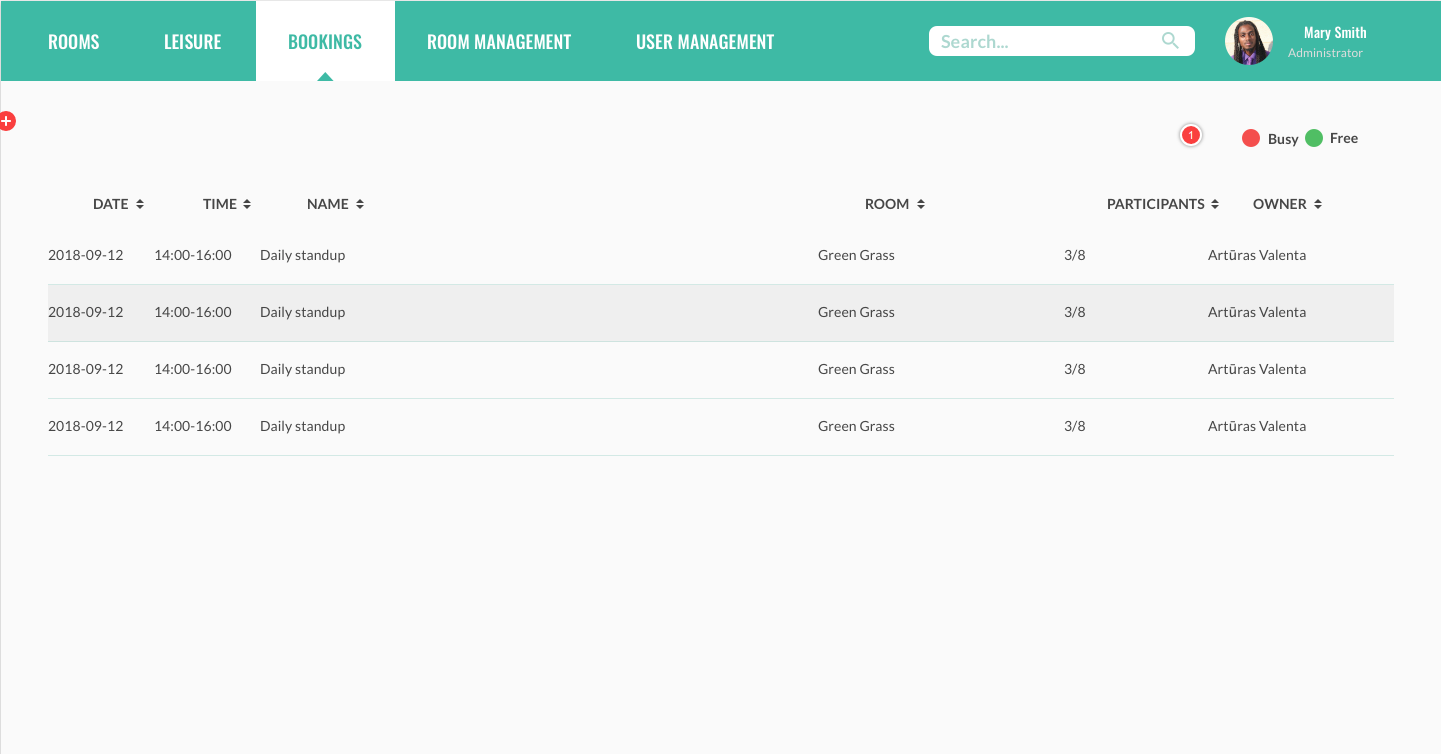
**Pav. 2.4.** Susirinkimo kambarių sąrašo langas mobiliesiems įrenginiams



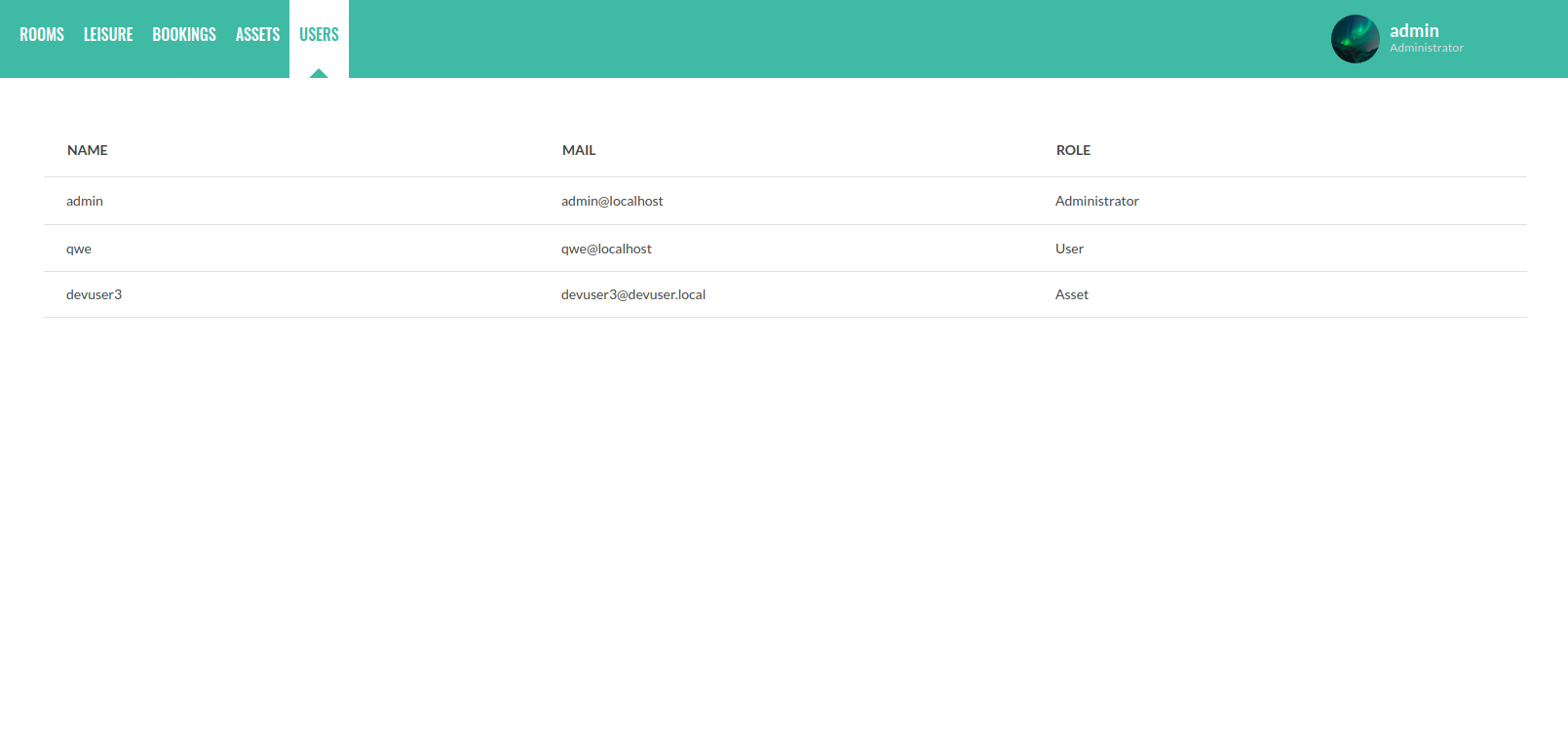
**Pav. 2.5.** Susirinkimo kambario rezervavimo langas



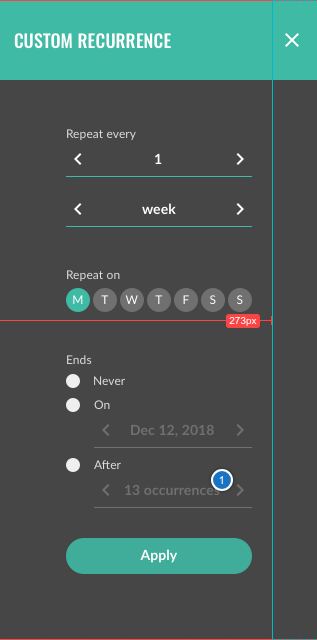
**Pav. 2.6**. Susirinkimo kambario rezervavimo langas mobiliesiems įrenginiams



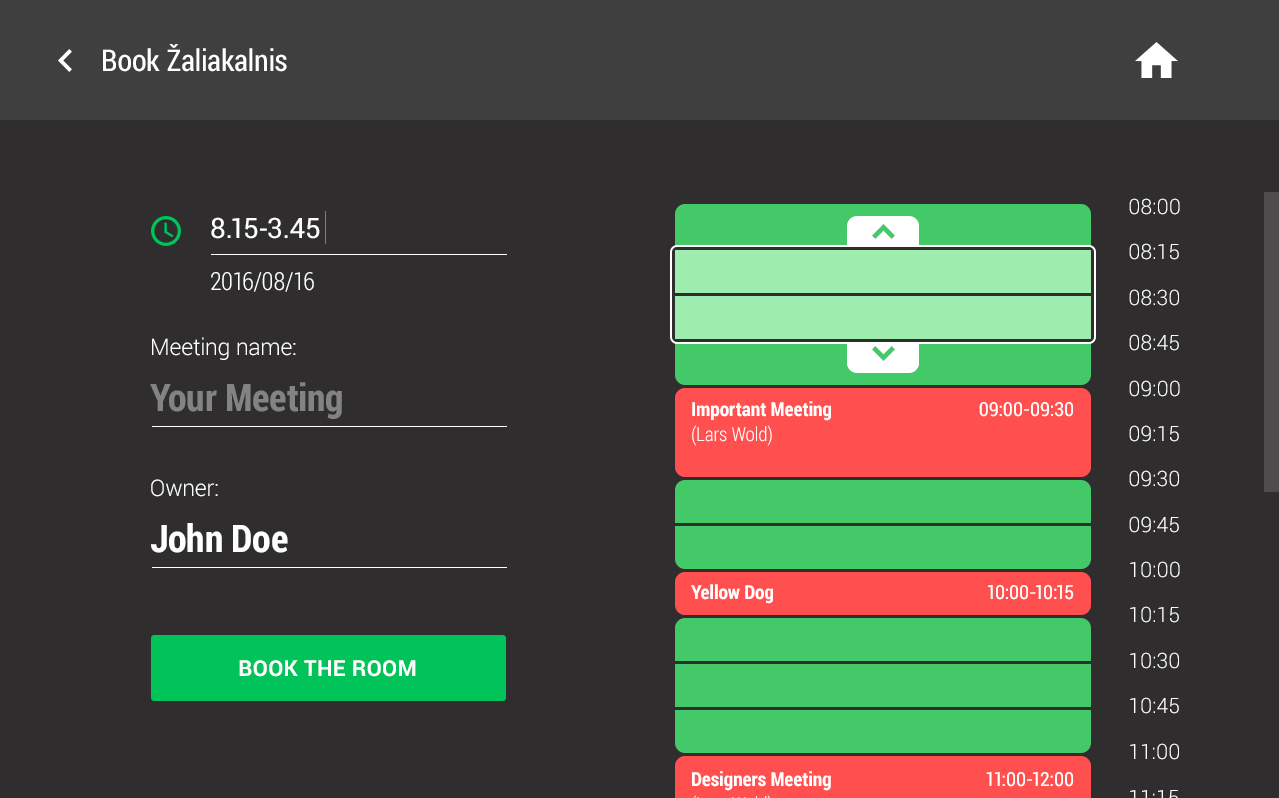
**Pav. 2.7**. Visų rezervacijų sąrašo peržiūros langas



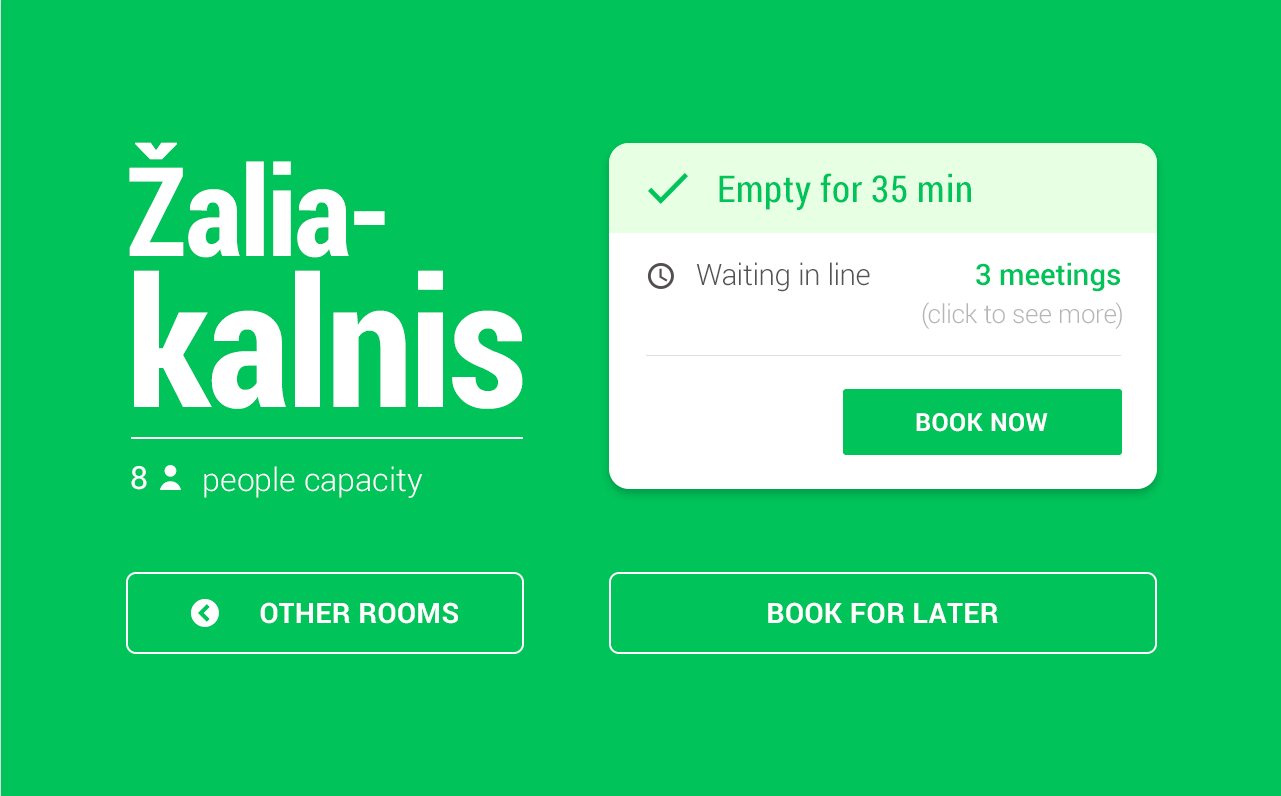
**Pav. 2.8**. Visų vartotojų sąrašo peržiūros langas



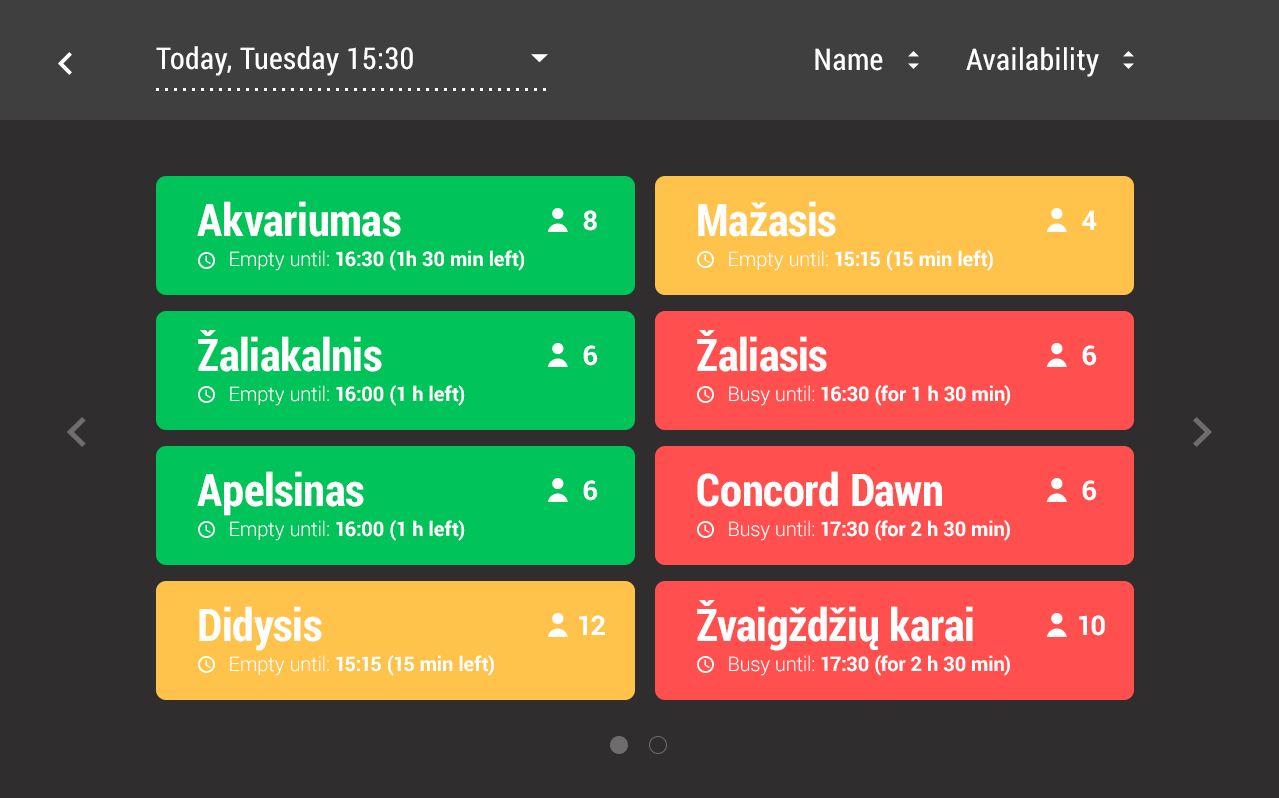
**Pav. 2.9**. Susirinkimo kambario rezervavimo formos langas mobiliesiems įrenginiams



**Pav. 2.10**. Susirinkimo kambario rezervavimo langas planšetėms



**Pav. 2.11**. Susirinkimo kambario informacijos peržiūros langas planšetėms



**Pav. 2.12**. Susirinkimo kambarių sąrašo peržiūros langas planšetėms

### Realizacijai keliami reikalavimai.

Nefunkciniai reikalavimai:

* Paprastas ir lengvai suprantamas dizainas
* Sistema turi prisitaikyti prie daugumos mobiliųjų įrenginių ir kompiuterių ekranų dydžių ir būti palaikoma pagrindinių naršyklių
* Aiškūs operacijos rezultato pranešimai
* Informacija apie sėkmingai atliktas operacijas turėtų būti vaizduojama žalios spalvos pranešimų languose
* Informacija turi būti atnaujinama kas 30 sekundžių planšetėse, dėl nuolatos rezervuojamų kambarių

### Techninė specifikacija.

|  |  |
| --- | --- |
| Reikalavimas | Parametrai |
| Serveris | Virtualus |
| Disko talpa | 16 GB |
| MySQL duombazės versija | 5.6 arba didesnė versija |
| RAM | 4 GB |
| CPU | 2x2.4 GHz |
| Operacinė sistema | 64-bitų Linux CentOS 7 |

**Lentelė 2.10** Techninė specifikacija

## Projektavimo eiga ir sistemos projektas

### Projektavimo valdymas ir eiga

Projektas buvo kuriamas pasinaudojant iteraciniu „Scrum“ projektavimo modeliu.

Vykdymo ir eigos procesas:

* Sukurtos visos funkcionalumams įgyvendinti skirtos užduotys „Redmine“ projektų valdymo įrankyje.
* Nubraižytos panaudos atvejų diagrama, veiklų diagramos.
* „Sprint‘ų“ suplanavimas, kuriuose numatomas tam tikro funckionalumo įgyvendinimas ir paskirstomos užduotys.
* „Sprint‘ų“ vykdymas, kiekvieną dieną darant „Stand-up“ susirinkimus, kuriuose aptariamos problemos ir galimi sprendimai.
* Įvykdžius „Sprint‘ą“, daromas apibendrinimas, padaryto progreso pristatymas bei kito „Sprint‘o“ koregavimas/planavimas.

### Projektavimo technologija

Sistemos buvo projektuojama panaudojant UML (4) kalba bei MagicDraw įrankį. Pradedant projektuoti sistemą, nubraižoma panaudos atvejų diagrama bei kiekvienam panaudos atvejui nubraižoma sekų diagrama.

### Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos

Sistemos „Front-end“ dalis buvo kuriama naudojant „JavaScript“ programavimo kalbos „ReactJs“ biblioteką-karkasą bei „React Redux“ (6) vidinės būsenos organizavimo ir valdymo biblioteka. Stiliai buvo aprašomi „CSS-in-JSS“ (7) kalba. Užklausoms siūsti buvo naudojama bazinė „fetch“ biblioteka. Viskas buvo taip pat programuojama naudojant „Ubuntu“ 18.04 operacinė sistema ir „Visual studio code“ programavimo aplinką. Testams realizuoti buvo pasitelkiama komponentų testavimo bibliotekos „Storybook“ (8) bei „Enzyme“ (9). Unit testam realizuoti buvo naudojamas „Jest“ (10) biblioteka.

Sistemos „Back-end“ dalis buvo kuriama panaudojant „PHP“ programavimo kalbos „Symfony“ (5) karkasą. Programuojama „IntelliJ IDEA“ aplinkoje, naudojant „Ubuntu“ 18.04 operacinę sistemą. Naudota „MySQL“ tipo realiacinė duomenų bazė.

### Informacijos srautai

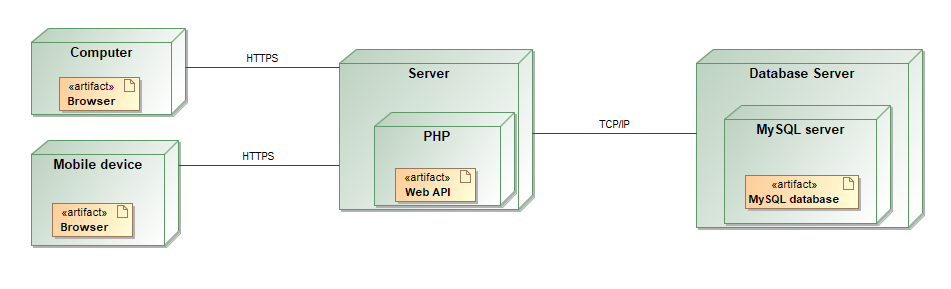
Sistema sukurta naudojant REST principus. Vartotojas, naudodamas naršyklę per vartotojo sąsają siunčia HTTP užklausas ir serverinę sistemos dalį, kurioje tos užklausos apdorojamos valdiklių, pakeičiant duomenų bazės įrašus ir grąžinant sutartą atsakymą atgal į vartotojo sąsają, kurioje parodomas atitinkamas atsakymas.

### Sistemos sudėtis

Sistema sudaryta iš trijų dalių:

* Vartotojo sąsajos – „React“ karkaso komponentų failai, „Javascript“ pagalbiniai, vidinės būsenos valdymo failai, „HTML“ failai. Kadangi „ReactJs“ bei „Styled-components“ bibliotekos suteikia galimybę „HTML“ ir „CSS“ kodą rašyti „JavaScript“ failuose, todėl juose aprašomas ne tik sistemos veikimas, bet ir turinys su stiliais.
* Serverinė dalis – „PHP“ klasių failai, konfigūraciniai failai, duombazės migracijų failai.
* Duomenų – duomenų bazė, kurioje saugoma visa sistemos informacija

### Komponentai



**Pav. 2.13** Komponentų diegimo diagrama

### Sąsajos įvertinimas pagal vartotojo patyrimą, profesinę terminologiją

Kuriant vartotojo sąsają buvo atsižvelgta į tai, kad sistemos vartotojai bus įmonės darbuotojai, kurių pagrindinis tikslas – pasižiūrėti užrezervuotą susitikimo kambarį, arba patiems jį rezervuoti. Todėl sąsaja buvo kuriama, greitai veikianti ir kuo paprastesnė naudoti, kad kambario rezervavimas neužimtų daug laiko.

### Duomenų kontrolė

Vykdyta dviejų lygių duomenų validacija:

* Vartotojo sąsajos validacija – „Front-end“ dalyje vykdoma duomenų validacija, ir priklausomai nuo jų teisingumo, siunčiama arba nesiunčiama užklausa į serverinę dalį.
* Serverinės dalies validacija – „Back-end“ dalyje vykdoma duomenų validacija, kuri yra tokia pati kaip vartotojo sąsajoje. Ji vykdoma todėl, nes vartotojas gali siųsti užklausas nenaudojant vartotojo sąsajos, todėl reikia antrą kartą tikrinti duomenų validumą.

Jeigu nebūtų vartotojo sąsajos validacijos, vartotojui reikėtų visuomet laukti atsakymo iš serverinės dalies, nes užklausa visada būtų siunčiama, nepriklausomai nuo duomenų validumo. Jeigu nebūtų serverinės dalies validacijos, neteisingais duomenų rinkiniais būtų galima sutrikdyti serverio veiklą.

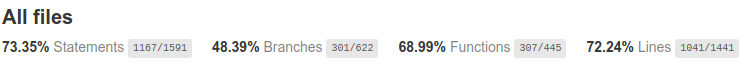
## Testavimo eiga ir rezultatai

Sistema buvo ištestuota šiais testavimo būdais:

* Vienetų testais.
* Integraciniai testais
* Rankiniu testavimu

Parašytui kodui taip pat buvo naudojama statinė kodo analizė. Buvo naudojamas „EsLint“ įrankis.

Iš viso buvo parašyti 381 testai. Padengimas:



**Pav. 2.14** Kodo padengimas

### Vienetų testai

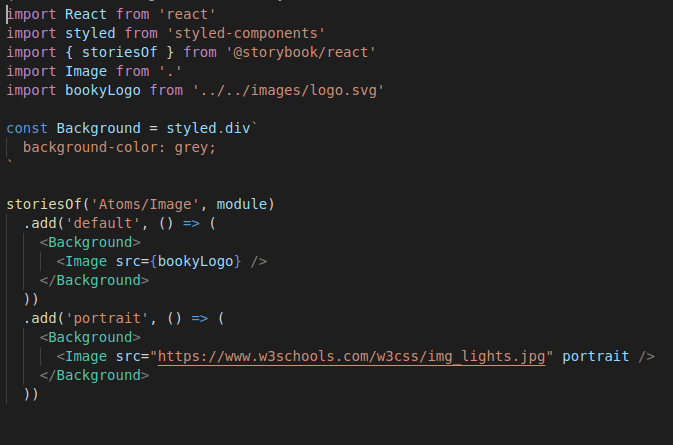
Vienetų testams buvo pasitlekiama „Jest“ testavimo biblioteka. Buvo testuojama pagrinde pagalbinės funkcijos ir metodai.

### Integraciniai testai

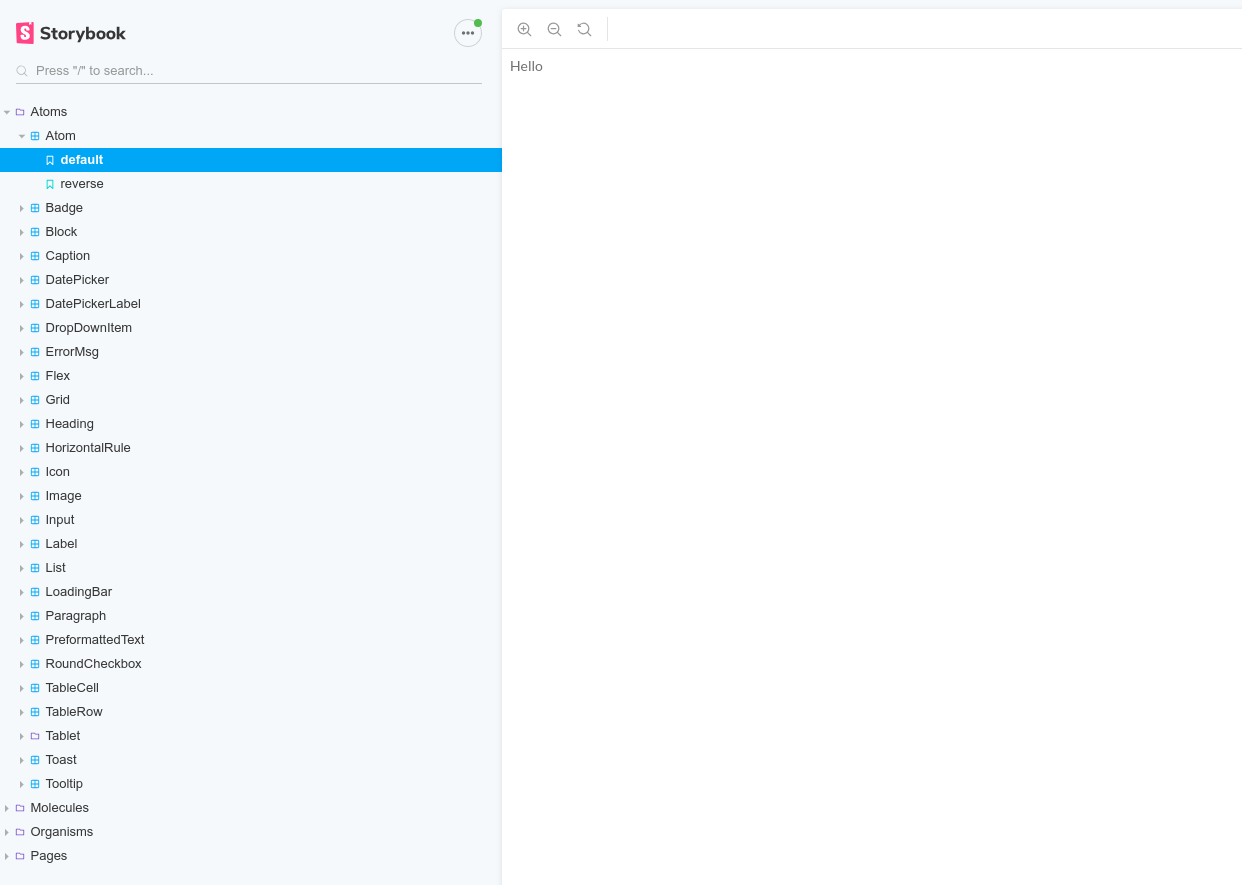
Integraciniai testai buvo rašomi pasitelkiant kelias bibliotekas:

* „Storybook“ bibliotekos testai
* „Enzyme“ ir „Jest“ bibliotekų teistai
* „Redux“ ir „Jest“ bibliotekų testai

„Storybook“ bibliotekos pagalba parašyti testai pagrinde tikrina kaip komponentai atrodys vartotojų sąsajoje, su galimybe manipuliuoti jiems paduodamus parametrus. Šiais testais padengta apie 80% naudojamų komponentų.



**Pav. 2.15** „Storybook“ testo aprašo pavyzdys

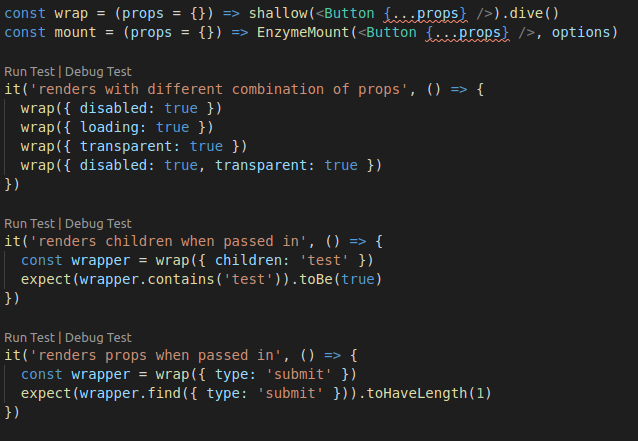


**Pav. 2.16** „Storybook“ komponentų testavimo sąsajos langas

„Enzyme“ ir „Jest“ bibliotekomis kuriami integraciniai testai, kurie patikrina kaip komponentai veikia tarpusavyje, ar jie išpildo savo funkcijas. „Jest“ bibliotekos pagalba buvo sukuriami netikri parametrai, kurie buvo paduodami komponentams ir „Enzyme“ bibliotekos pagalba komponentai buvo render‘inami netikroje aplinkoje.

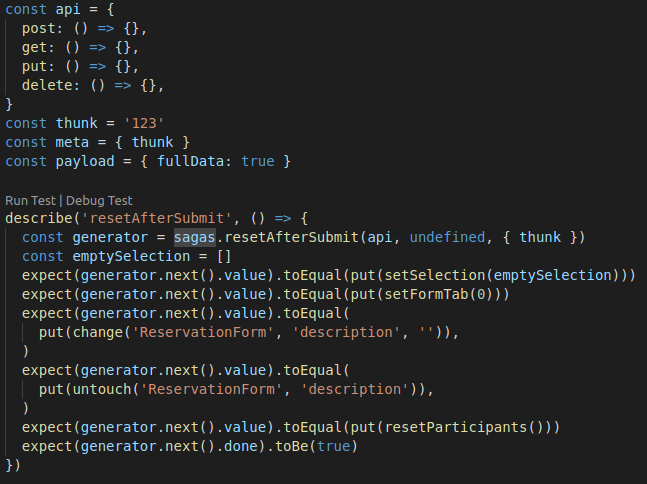


**Pav. 2.17** Netikras action objektas



**Pav. 2.18** „Enzyme“ ir „Jest“ bibliotekų testo pavyzdys

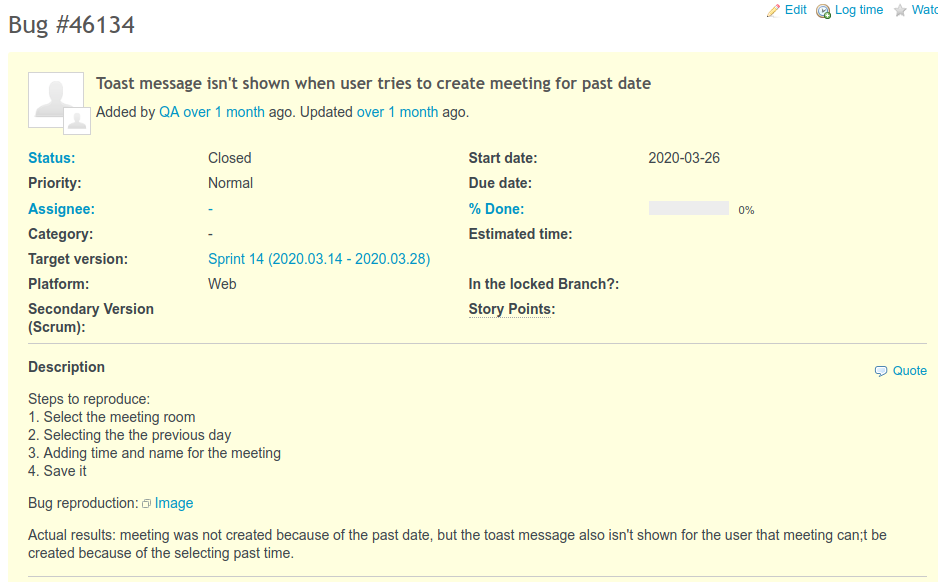
„Redux“ ir „Jest“ bibliotekos testai buvo rašomi patikrinti ar tinkamai veikia vidinės sąsajos būsenos valdymas ir organizavimas. Šiais testais taip pat buvo tikrinami, ar platforma gerai apdoroja asinchroninių užklausų į serverį atsakus.



**Pav. 2.19** „Redux“ ir „Jest“ bibliotekso testo pavyzdys

### Rankiniai testai

Rankiniai testai buvo atliekami pagrinde mūsų komandos testuotojų ir tai buvo pagrinde daroma tiesiogiai naudojantis sistema. Radus neatitikimą, „Redmine“ projektų valdymo įrankyje buvo užpildomas bug‘as.



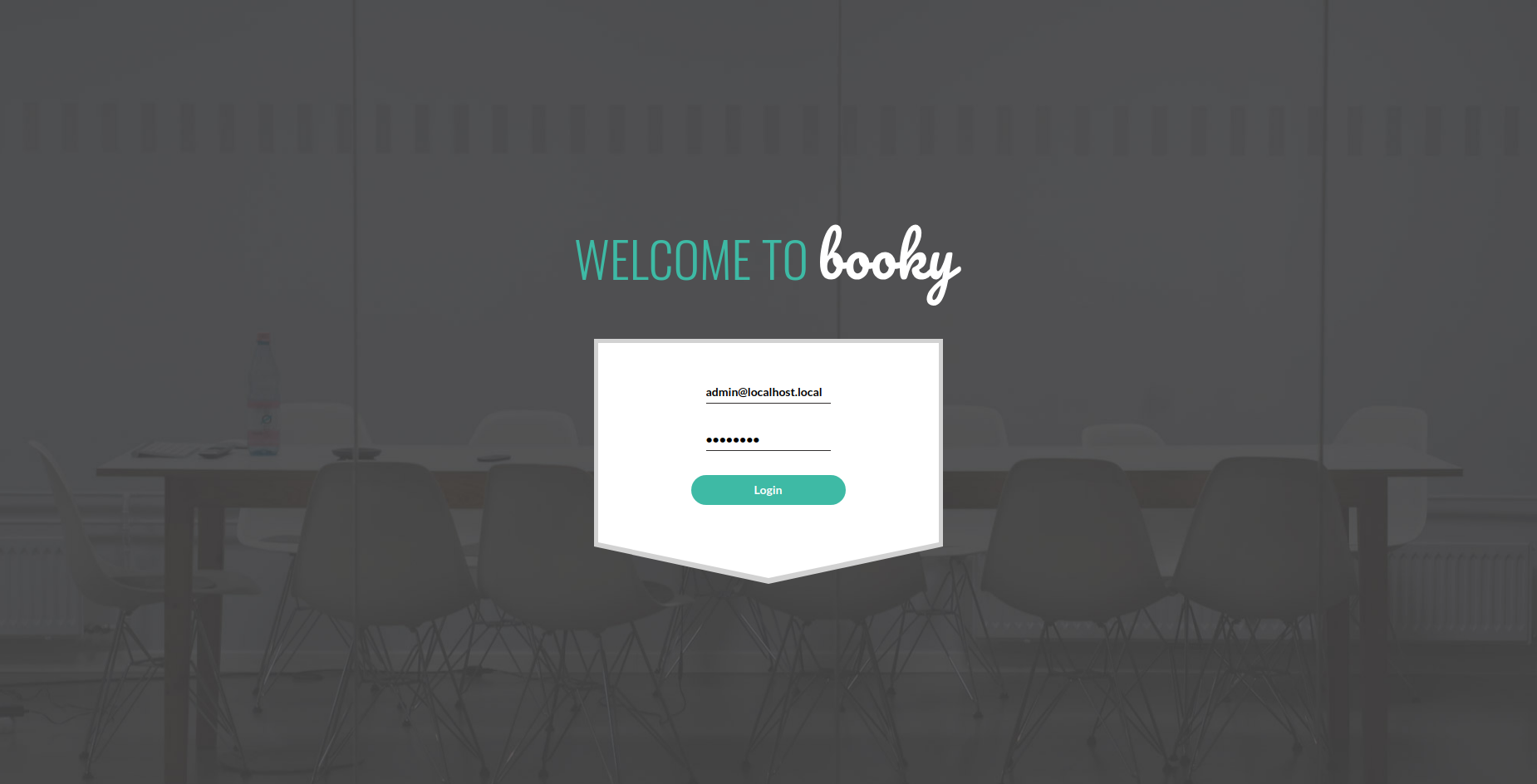
**Pav. 2.20** Bug‘o užpildymo pavyzdys

## Dokumentacija

### Apibendrintas sistemos galimybių sąrašas

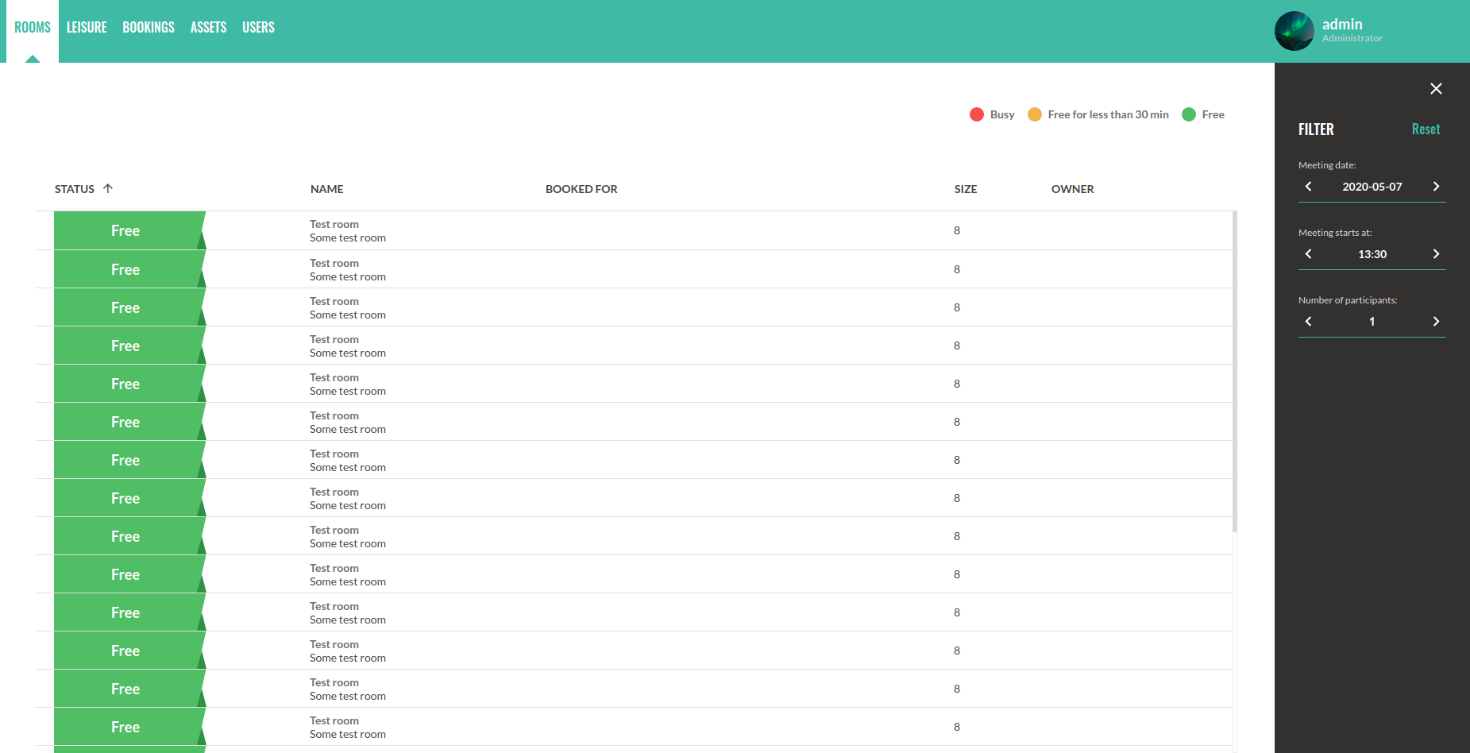
Sukurta sistema suteiks vartotojams patogiai matyti ir rezervuoti susitikimo kambarius bei laisvalaikio įrangą. Rezervuojant, vartotojas turės daug parinkčių kokią rezervaciją jis nori padaryti. Bus galima matyti susitikimo kambarių ir laisvalaikio įrangos sąrašą bei jį filtruoti. Prieinamas ir vartotojo sukurtų rezervacijų sąrašas, o administratoriui papildomai visųs rezervacijų ir vartotojų sąrašai. Jei susitikimų kambarys neteisingai rezervuotas, savininkas gali redaguoti arba pašalinti rezervaciją, o kiti vartotojai gali pranešti jam apie neteisingai rezervuotą kambarį. Pastebėjus klaidą sistemoje vartotojas galės apie ją pranešti sistemos kūrėjams.

### Galutinio vartotojo dokumentacija



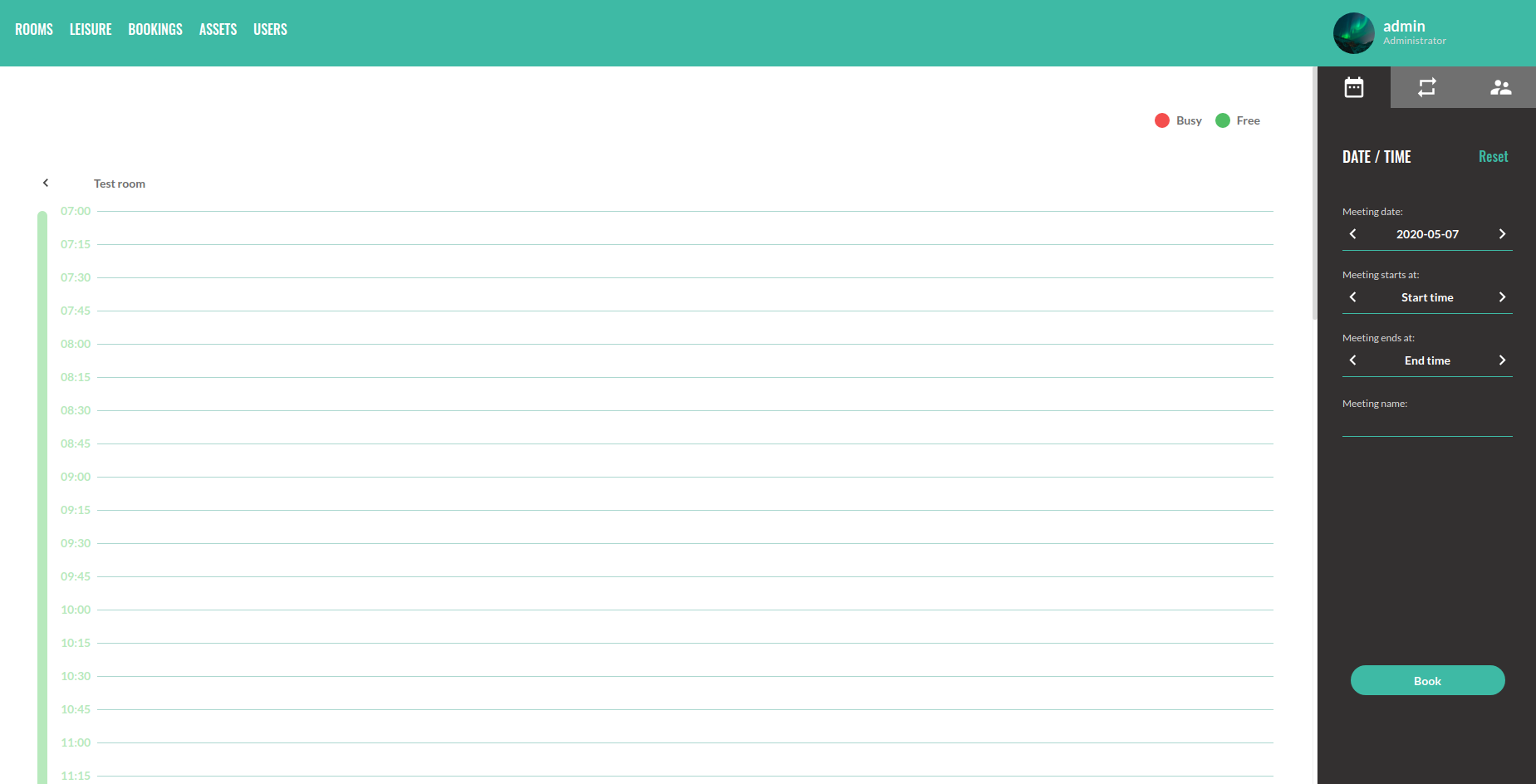
**Pav. 2.22** Prisijungimo langas

Šiame lange, kas prisijungtų, vartotojas turi įvesti savo darbovietės el. paštą ir slaptažodį. Tada spausti „Login“ mygtuką. Po sėkmingo prisijungimo vartotojas bus nukreiptas į susitikimo kambarių sąrašą.



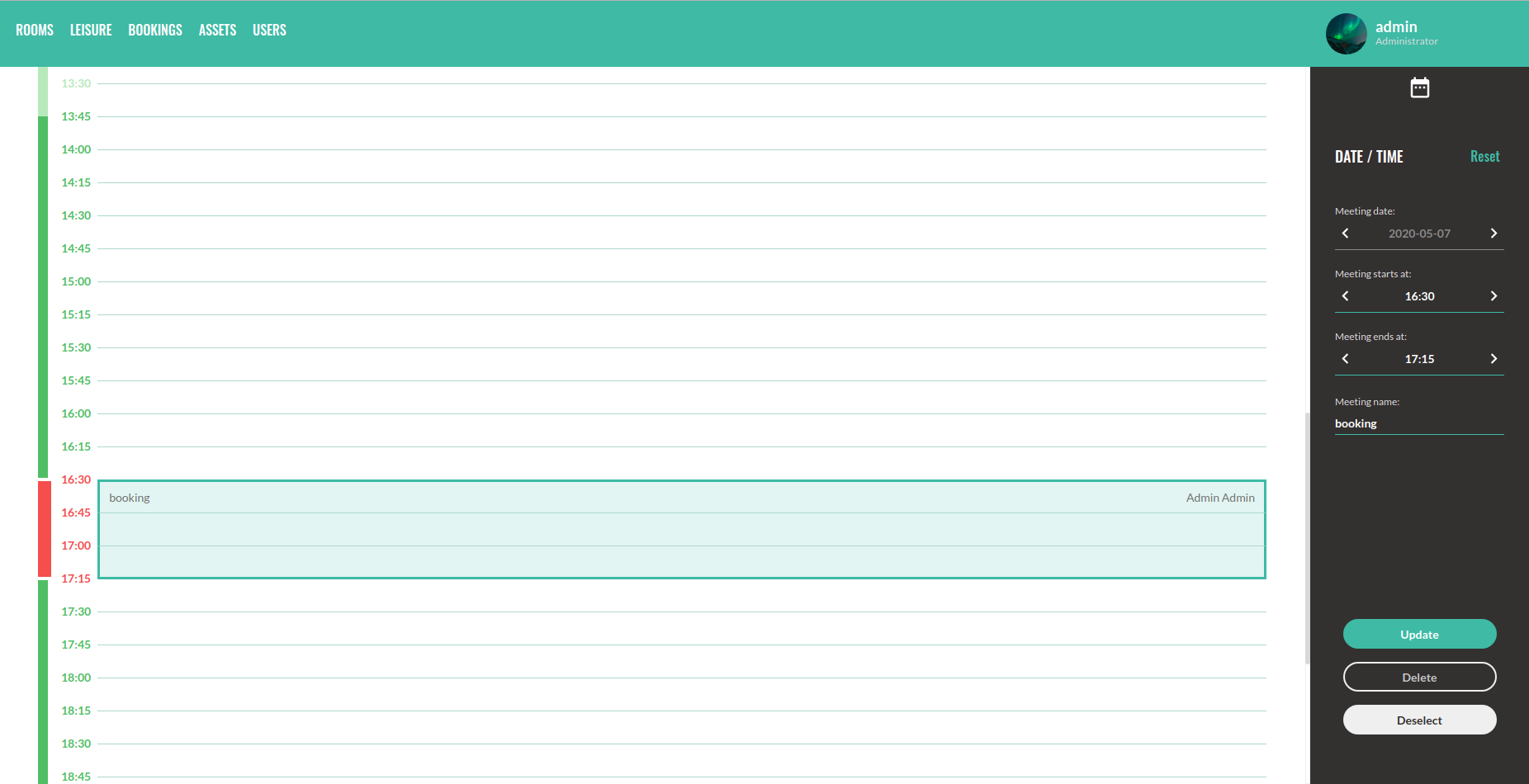
**Pav. 2.23** Susitikimo kambarių sąrašo langas

Šiame lange vartotojas gali matyti susitikimo kambarių sąrašą, jų būseną, aprašymą bei artimiausios rezervacijos kūrėją. Dešinėje yra sąrašo filtrai, pagal kuriuos bus filtruojamas susitikimo kambarių sąrašas. Vartotojas gali sąrašą filtruoti pagal susitikimo datą, sutikimo pradžios laiką bei susitikimo narių skačių. Viršuje galima matyti navigacijos juostą, paprasti vartotojai gali naviguoti tarp savo rezervacijų, susitikimo kambarių ir laisvalaikio įrangos sąrašų langų, administratorius gali prieti dar du papildomus langus: vartotojų ir bendro susitikimo kambarių bei laisvalaikio įrangos sąrašų. Vartotojui paspaudus ant susitikimo kambario, jis bus nukreiptas į jo rezervacijos kūrimo puslapį.



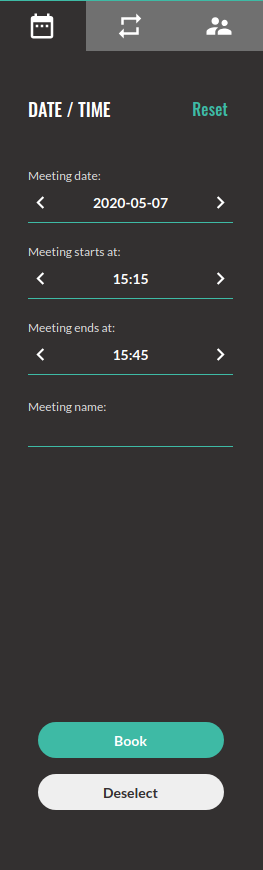
**Pav. 2.24** Rezervacijos kūrimo langas

Šiame lange vartotojas mato laiko juostą, bei joje rezervuotus laikus. Norint sukurti rezervaciją, vartotojas turi pažymėti laiko juostoje laisvą laiko tarpą, dešinėje įvesti susitikimo pavadinimą, pasirinkti norimas rezervacijos pasikartojimo opcijas bei pridėti susitikimo narius ir spausti mygtuką „Book“. Sėkmingo rezervacijos sūkurimo atveju bus parodyta kiek iš viso buvo sukurta rezervacijų bei rezervacijos laikas atsiras laiko juostoje.



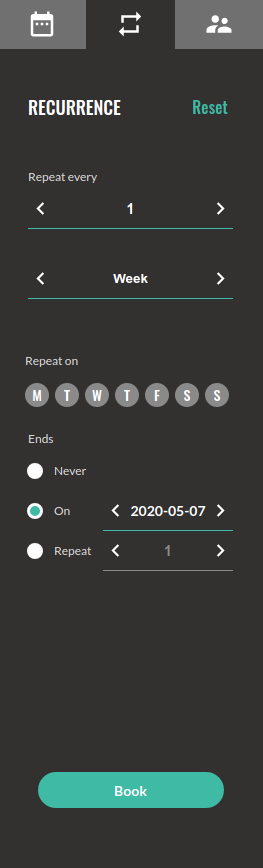
**Pav. 2.25** Rezervacijos redagavimas

Norint redaguoti rezervaciją, vartotojas turi pasirinkti rezervaciją laiko juostoje ir dešinėje pakeisti rezervacijos informaciją. Pakeitus informaciją spausti mygtuką „Update“.



**Pav. 2.26** Rezervacijos informacija

Kuriant/redaguojant rezervaciją vartotojas gali įvesti rezervacijos pradžios ir pabaigos laikus, jos datą bei susitikimo pavadinimą.



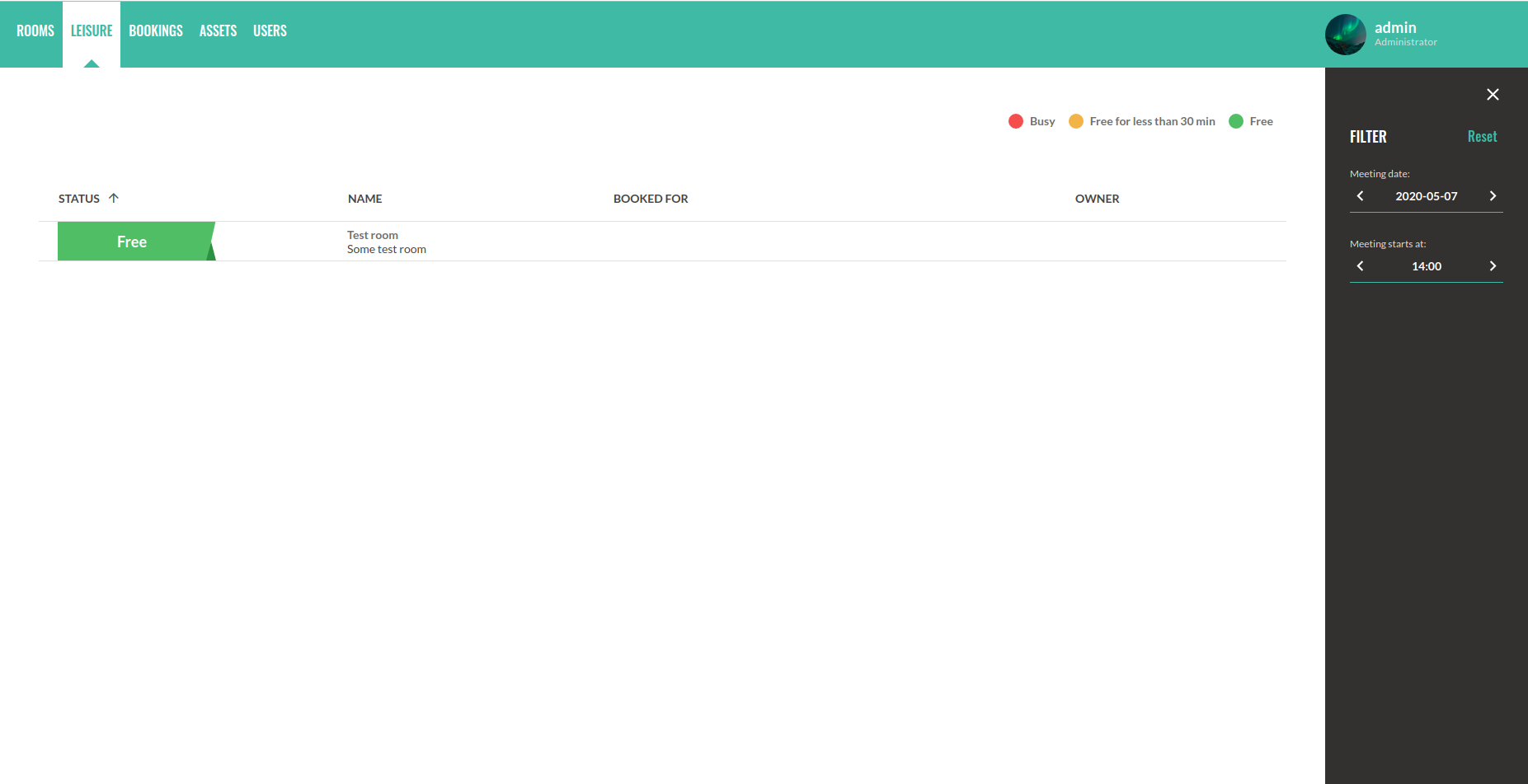
**Pav. 2.27** Rezervacijos pasikartojimo opcijos

Kuriant rezervaciją, vartotojas gali pasirinkti rezervacis kartojimosi opcijas. Jis gali pasirinkti kas kiek savaičių/dienų, kokiomis savaitės dienomis, kiek kartų arba iki kada rezervacija turėtų kartotis.



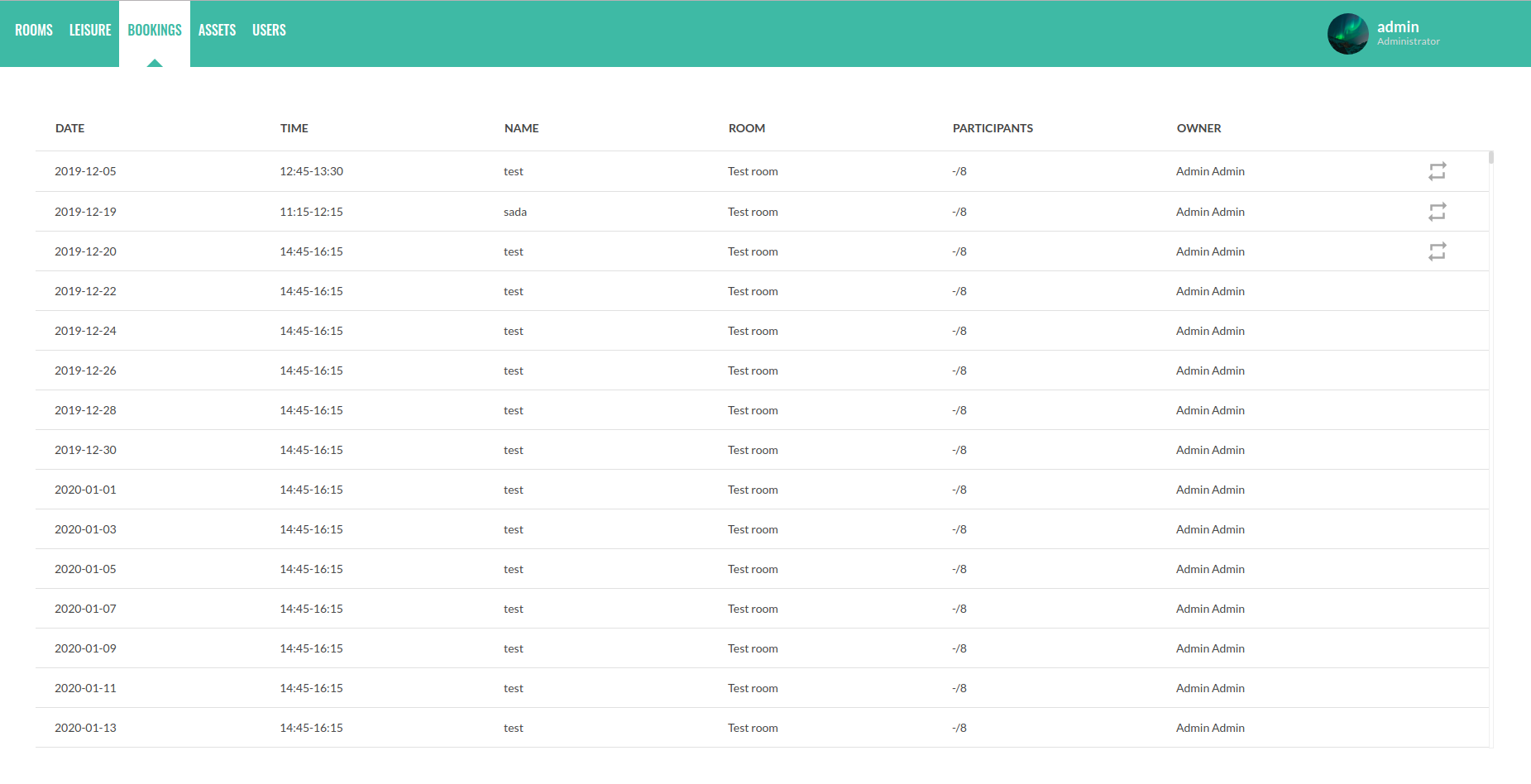
**Pav. 2.28** Susitikimo narių pridėjimas

Norint pridėti susitikimo narius reikia laukelyje „Enter name“ įvesti nario vardą ir spausti pliuso ikonėlę.



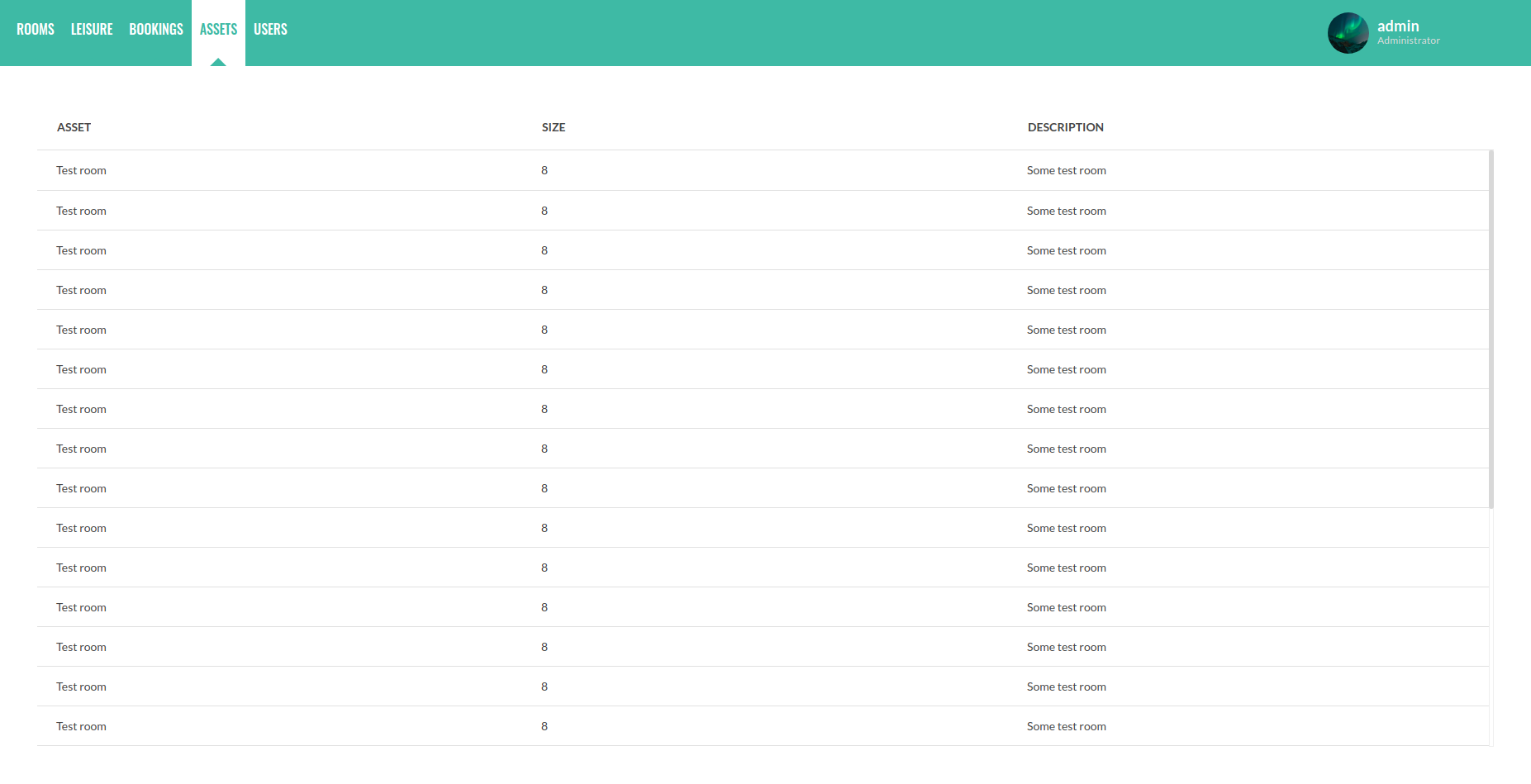
**Pav. 2.29** Laisvalaikio įrangos langas

Šiame lange varototjas mato laisvalaikio įrangos sąrašą, bei apibendrintą informaciją apie laisvalaikio įrangą kaip ir susitikimo kambarių sąraše. Šį sąrašą galima filtruoti pagal rezervacijos datą ir laiką. Paspaudus ant laisvalaikio įrangos, vartotojas bus nukreiptas į rezervacijos kūrimo puslapį, identišką kambario rezervavimo langui.



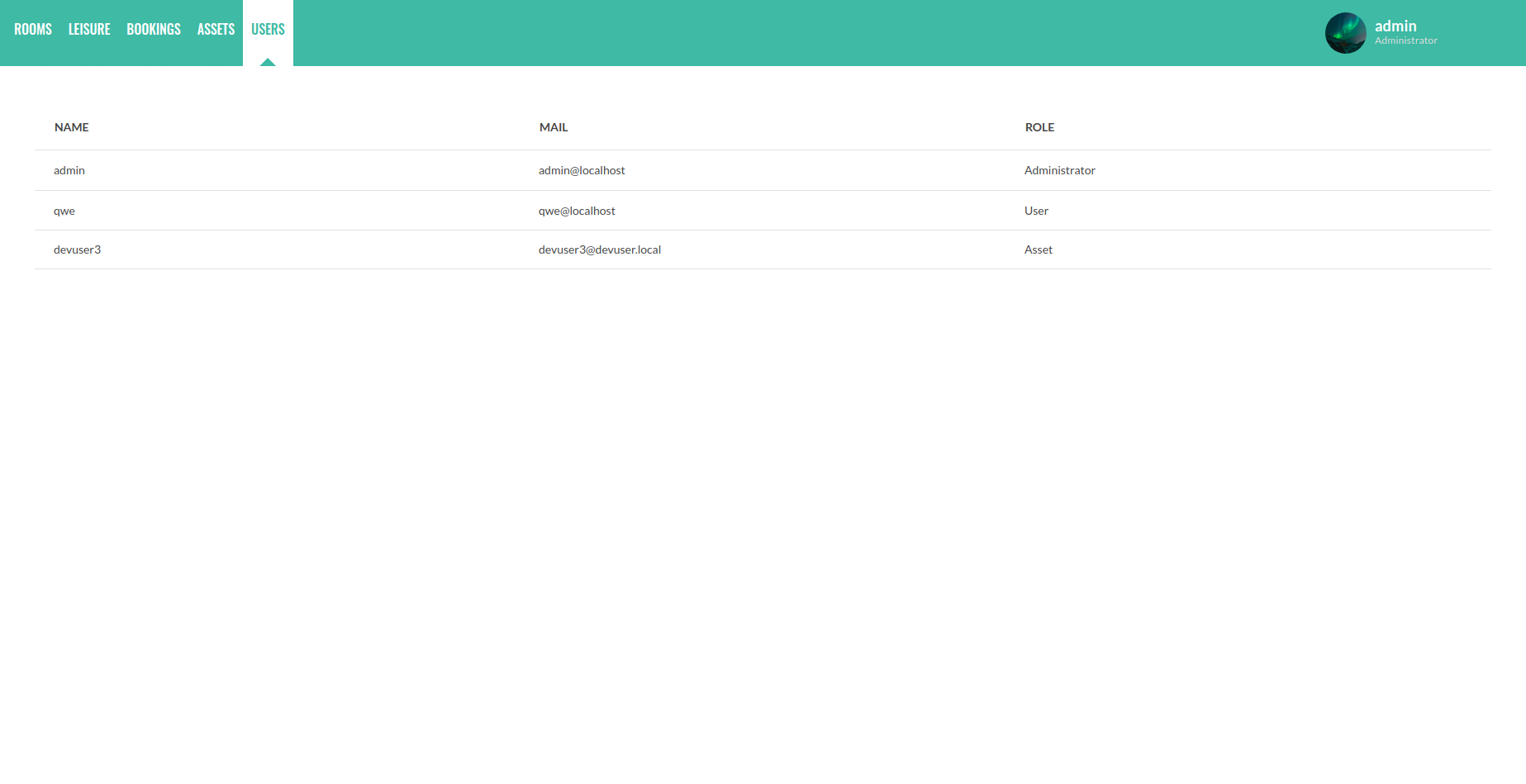
**Pav. 2.30** Visų rezervacijų langas

Šiame lange vartotojas gali matyti visas savo sukurtas rezervacijas ir informaciją apie jas. Paspaudus ant rezervacijos, vartotojas bus nukreiptas į rezervacijos redagavimo langą.



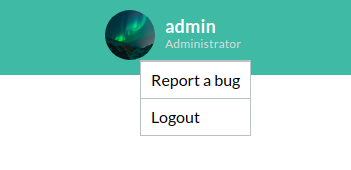
**Pav. 2.31** Bendras susitikimo kabarių ir laisvalaikio įrangos sąrašas

Šiame lange administratorius gali matyti bendrą susitikimo kambarių ir laisvalaikio įrangos sąrašą.



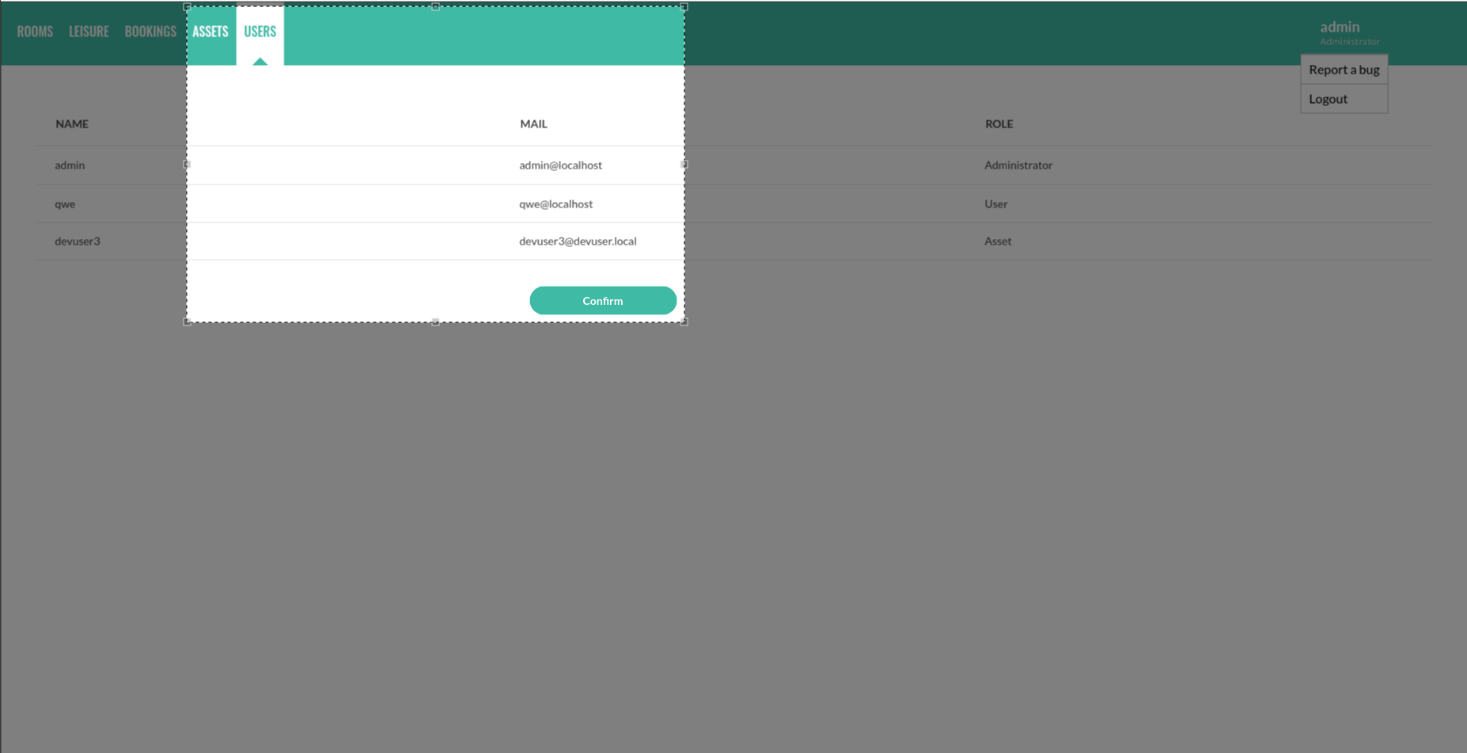
**Pav. 2.32** Vartotojų sąrašas

Šiame lange administratorius gali matyti visų vartotojų sąrašą.



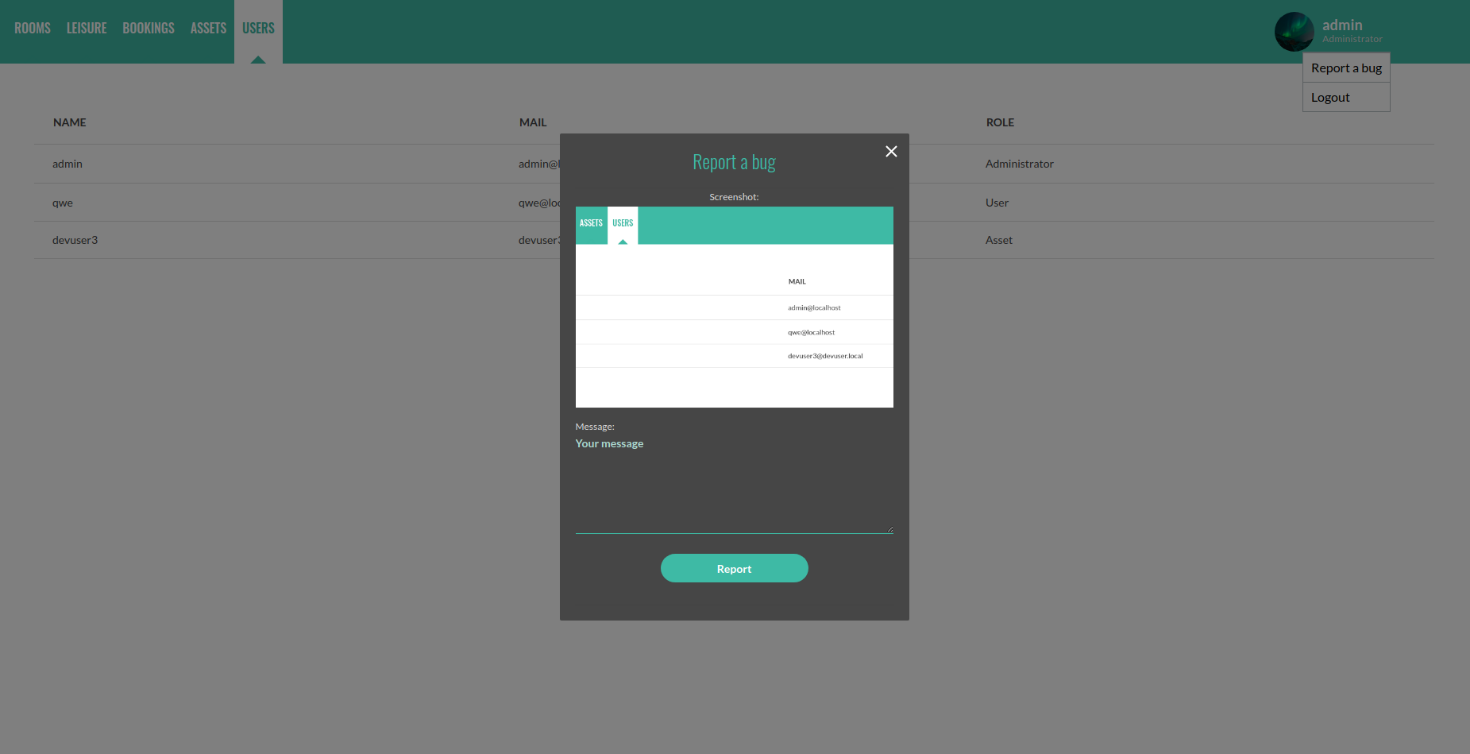
**Pav. 2.33** Vartotojo meniu

Vartotojui užvedus pele ant savo profilio, vartotojas turi atsijungimo ir sistemos klaidos pranešimo galimybes.



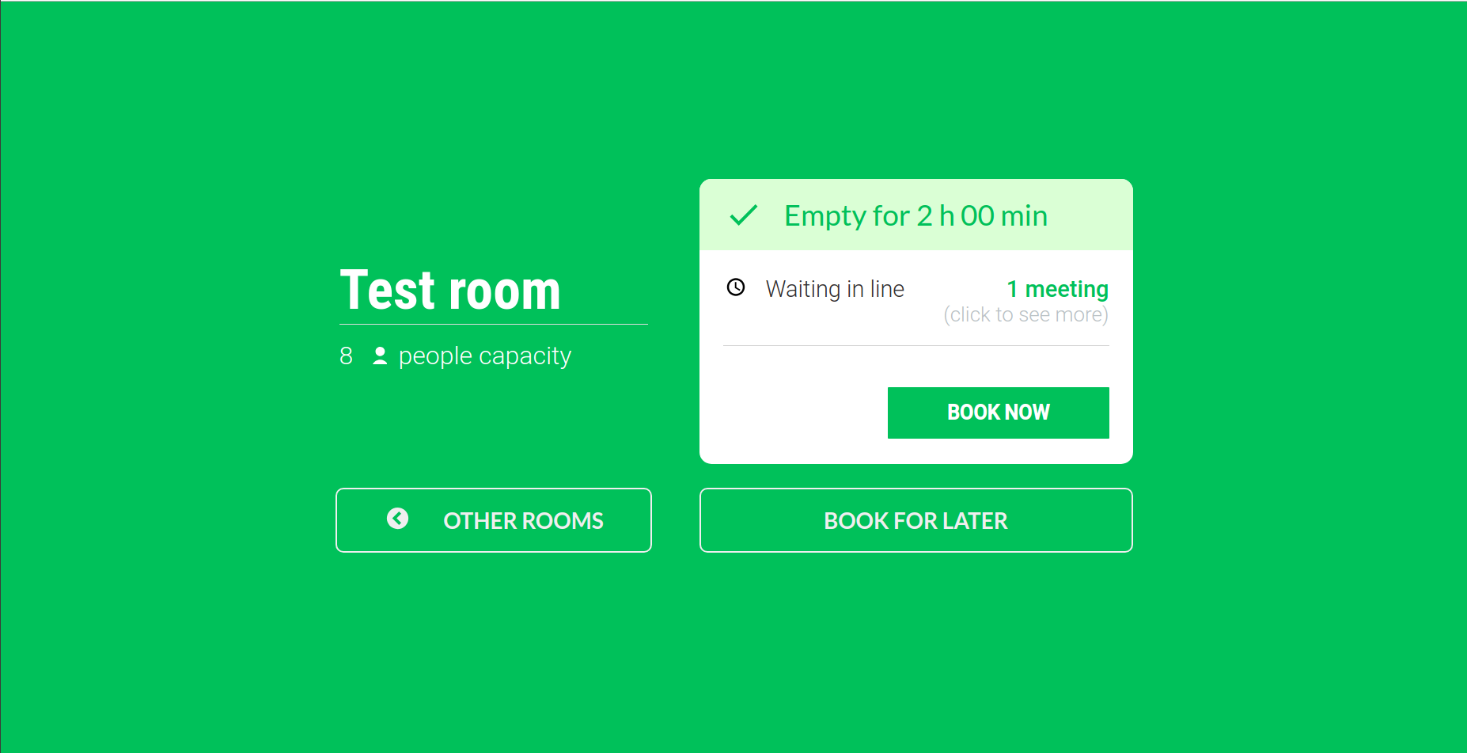
**Pav. 2.34** Sistemos klaidos pranešimo lango fiksavimas

Vartotojui pasirinkus sistemos klaidos pranešimo opciją, jis gauna galimybę fiksuoti klaidos vietą ekrane. Nurodžius ekrano vietą, paspausti mygtuką „Confirm“.



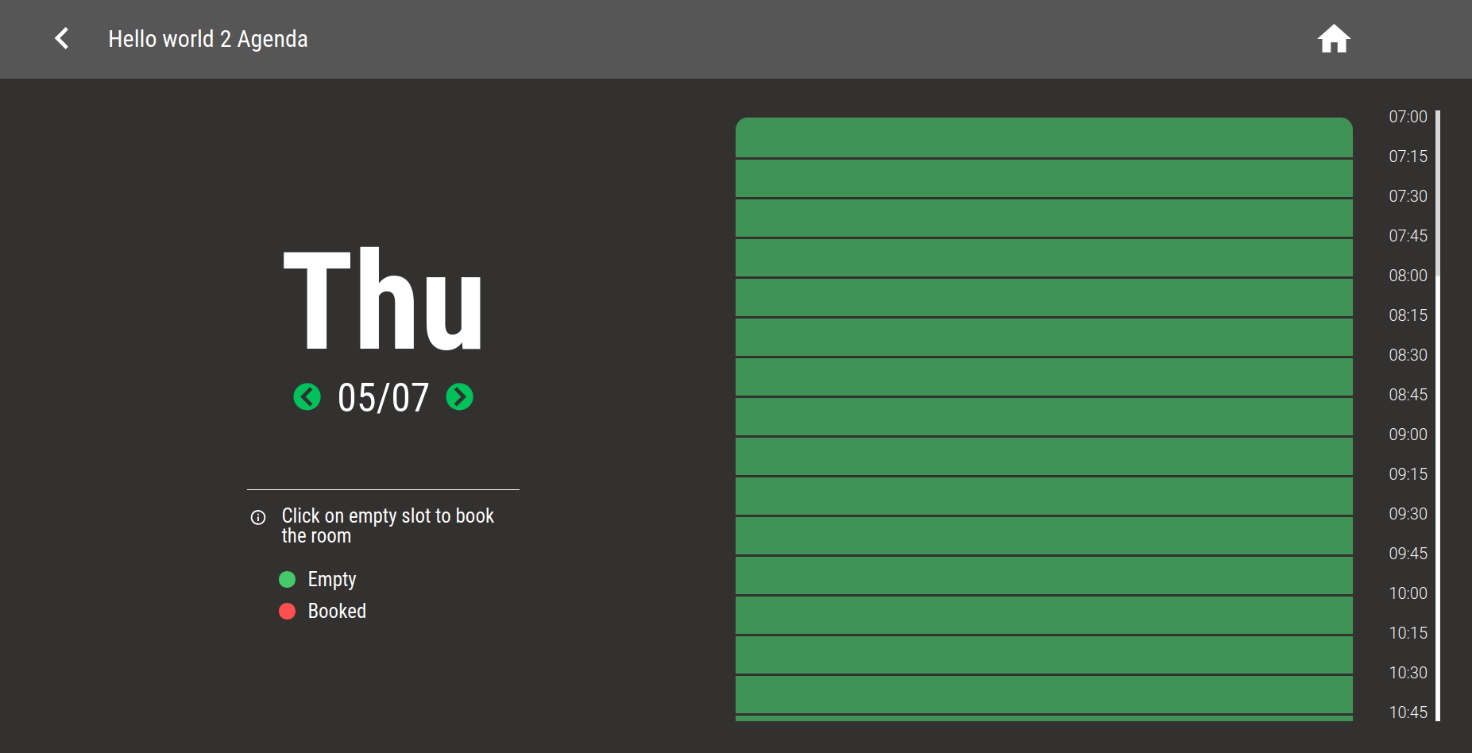
**Pav. 2.35** Sistemos klaidos pranešimo sukūrimas

Šiame lange vartotojas mato savo ekrano vietos užfiksavimą. Įvedus lauke „Your message“ klaidos aprašimą ir paspaudus mygtuką „Report“ klaidos pranešimus bus įrašytas.



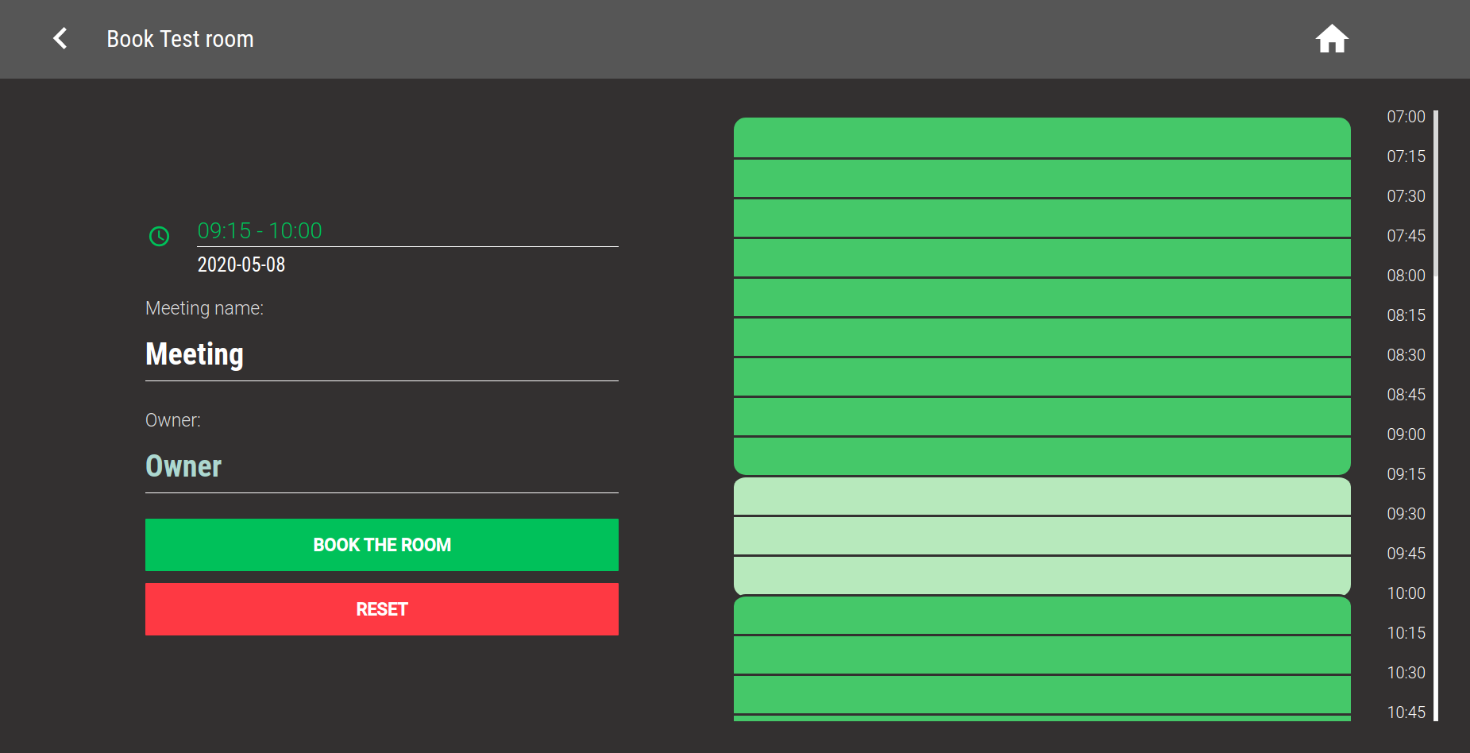
**Pav. 2.36** Planšetės susitikimo kambario informacija

Šiame planšetės lange vartotojas gali matyti informaciją apie susitikimo kambarį ir apie jo rezervacijas. Paspaudus mygtukas „OTHER ROOMS“ vartotojas bus nukreiptas į susitikimo kambarių sąrašą. Paspaudus mygtukus „BOOK NOW“ arba „BOOK FOR LATER“ vartotojas bus nukreiptas į rezervacijos kūrimo langą.



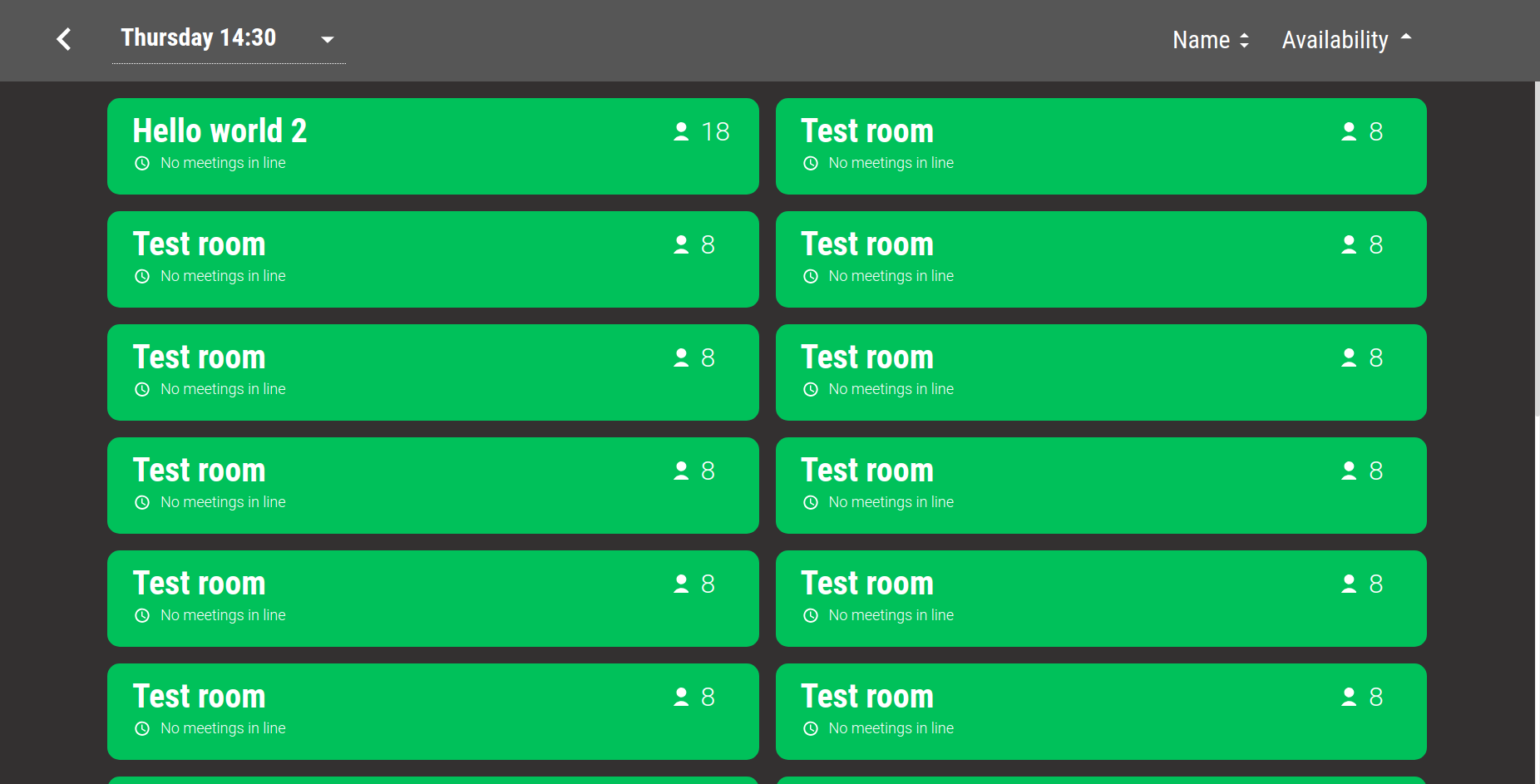
**Pav. 2.37** Rezervacijos kūrimas planšetės lange

Šiame planšetės lange vartotojas gali kurti rezervacijas.



**Pav. 2.38** Rezervacijos kūrimas planšetės lange

Vartotojui laiko juostoje pasirinkus laikotarpį, kairėje atsiras rezervacijos forma. Įvedus susitikimo pavadinimą ir paspaudus mygtuką „BOOK THE ROOM“ bus sukurta rezervacija. Paspaudus mygtuką „RESET“ formos duomenys ir pasirinktas laikotarpis bus pašalinti.



**Pav. 2.39** Planšetės lango susitikimo kambarių sąrašas

Šiame planšetės lange vartotojas gali matyti susitikimo kambarių sąrašą. Viršutinėje juostoje varotojas turi filtrus, pagal kuriuos bus filtruojamas sąrašas. Filtruoti galima pagal datą ir laiką. Rikiuoti galima pagal susitikimo kambario pavadinimą ir būseną.

Išvados

Atlikus darbą buvo prieitos išvados:

1. Žinant projektų vykdymo ir kūrimo procesus įmonėje, galima greičiau įsijungti į projekto veiklą, taip išvaistant mažiau resursų ir laiko.
2. Palaikant nuolatinį ryšį su projekto vadovu/užsakovu, išsiaiškinami funkcionalumų ypatumai, kaip jie turėtų būti įgyvendinti. Tai žinant, bus padaryta mažiau klaidų ir neatitikimų realizavus atitinkantį funkcionalumą/užduotį.
3. Projekto įgyvendinimo metu naudota „Agile“ metodika padėjo efektyviau suplanuoti sistemos kūrimo, testavimo procesus. Taip dėl galimo projekto užsakovo įsitraukimo į įgyvendinimo procesą, gaunamos naudingos pastabos bei patikslinimai. Tai padeda tiksliau įgyvendinti norimą galutinį produktą.
4. Kuriant produktą privaloma laikytis „Clean code“ ir kitų gerų praktikų/standartų, taip padidinant kodo skaitomumą ir efektyvumą įgyvendinant funkcionalumus dirbant komandoje.
5. Atliktas išsamus testavimas, kurio metu parašyti 381 vienetų ir integracinių testų užtikrino didelį sistemos patikimumą ir sumažino aptinkamų klaidų skaičių.
6. Kuriant sistemą susipažinta su komandinio darbo kuriant vieną sistemą procesais, praktiškai patirta „Agile“ metodika, buvo pritaikytos universitete gautos žinios taip pat susipažinta bei pramoktos naujos programinės įrangos kūrimo technologijos.

Literatūros sąrašas

1. Atlassian. *Atlassian.* [Tinkle] [Cituota: 2020 m. gegužės 21 d.] https://www.atlassian.com/.

2. Scrum. *Wikipedia.* [Tinkle] [Cituota: 2020 m. gegužės 21 d.] https://en.wikipedia.org/wiki/Scrum.

3. A JavaScript library for building user interfaces. *React.* [Tinkle] [Cituota: 2020 m. gegužės 21 d.] https://reactjs.org/.

4. Unified Modeling Language. *Wikipedia.* [Tinkle] [Cituota: 2020 m. gegužės 21 d.] https://lt.wikipedia.org/wiki/Unified\_Modeling\_Language.

5. Symfony. *Symfony.* [Tinkle] [Cituota: 2020 m. gegužės 21 d.] https://symfony.com/.

6. A Predictable State Container for JS Apps. *redux.* [Tinkle] [Cituota: 2020 m. gegužės 17 d.] https://redux.js.org/.

7. JSS is an authoring tool for CSS which uses JavaScript as a host language. *jss.* [Tinkle] [Cituota: 2020 m. gegužės 17 d.] https://github.com/cssinjs/jss.

8. Build bulletproof UI components faster. *Storybook.* [Tinkle] [Cituota: 2020 m. gegužės 21 d.] https://storybook.js.org/.

9. Enzyme is a JavaScript Testing utility for React that makes it easier to test your React Components' output. *EnzymeJS.* [Tinkle] [Cituota: 2020 m. gegužės 17 d.] https://enzymejs.github.io/enzyme/.

10. Getting Started. *jestjs.* [Tinkle] [Cituota: 2020 m. gegužės 17 d.] https://jestjs.io/docs/en/getting-started.