

Baigiamoji praktika

III-ojo tarpinio atsiskaitymo ataskaita

Organizacijos praktikos vadovas	<u>Projektų vadovas Raimundas Bučius</u> <i>(pareigos, vardas, pavardė, parašas)</i>
Praktikos vadovas universitete	<u>docentas Giedrius Ziberkas</u> <i>(pareigos, vardas, pavardė, parašas)</i>
Studentas:	<u>IFF-6/6 Ignas Jasonas</u> <i>(grupė, vardas, pavardė, parašas)</i>

1. Projektavimo eiga ir sistemos projektas

1.1. Projektavimo valdymas ir eiga

Projektas buvo kuriamas naudojantis iteraciniu „Scrum“ projektavimo modeliu. Iš pradžių buvo sukurtos visos užduotys „Redmine“ projektų valdymo įrankyje, skirtos funkcionalumui padengti. Buvo nubraižyta panaudos atvejų diagrama, veiklų diagramos. Visas darbas buvo suskirstytas į dviejų savaitių trukmės „sprint“us, kurių pradžioje vykdavo „sprint“o planavimai. Jų metu būdavo imamos užduotys iš viso užduočių sąrašo (kitai vadinamo „backlog“), įvertinamos taškais pagal sunkumą ir perkeliamos į „sprint“ą. Tų užduočių būdavo tiek, kad būtų įmanoma suspėti padaryti iki kito „sprint“o. Taip pat, pridėdant naują funkcionalumą būdavo apsitariama su backend'o programuotojais, kokius duomenų modelius jie nori gauti iš vartotojo sąsajos ir kokius jie grąžins atgal, kartais pakeičiamos veiklų diagramos (radus geresnį sprendimą ir pasitarus su projekto šeimininku). Tuomet kasdien vykdavo „stand-up“ susitikimai, kurių metu kiekvienas trumpai pasakydavo ką nuveikė vakar, ką veiks tą dieną ir su kokiomis problemomis susidūrė, visa tai užtrukdavo apie 10 minučių. Paskutinę „sprint“o dieną vykdavo jo peržiūra, kurios metu įvertindavome, ar viską spėjome padaryti, ką planavome, peržiūrėdavome „sprint“o užduotis ir uždarydavome tas užduotis, kurios būdavo visiškai pabaigtos.

1.2. Projektavimo technologija

Sistemos projektavimui buvo naudojama UML kalba ir MagicDraw įrankis. Pirmiausiai buvo nubraižyta panaudos atvejų diagrama, o funkcijų veikimui detalizuoti ir aiškiau pavaizduoti buvo nubraižytos veiklos diagramos kiekvienam panaudos atvejui.

1.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos

Sistemos „back-end“ dalis buvo kuriama naudojant „PHP“ programavimo kalbos „Symfony“ karkasą. Programavimas vyko „IntelliJ IDEA“ aplinkoje, naudota „Ubuntu“ 18.04 operacinė sistema. Naudota realiacinė duomenų bazė „MySQL“.

Sistemos „front-end“ daliai buvo naudojama „JavaScript“ programavimo kalbos „ReactJs“ biblioteka-karkasas bei „React Redux“ vidinės būsenos organizavimo ir valdymo biblioteka. Stiliai buvo aprašomi „CSS-in-JSS“ 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“ bei „Enzyme“. Unit testams realizuoti buvo naudojamas „Jest“ biblioteka.

1.4. Informacijos srautai

Kadangi sistema sukurta pasitelkiant REST technologiją, vartotojas, naudodamasis naršykle, siunčia HTTP užklausas į vidinę sistemos dalį, taip perduodamas duomenis ir gaudamas rezultatus. Vidinė sistemos dalis, gavusi užklausa, patikrina, ar vartotojas turi teisę šios užklauskos įvykdymui.

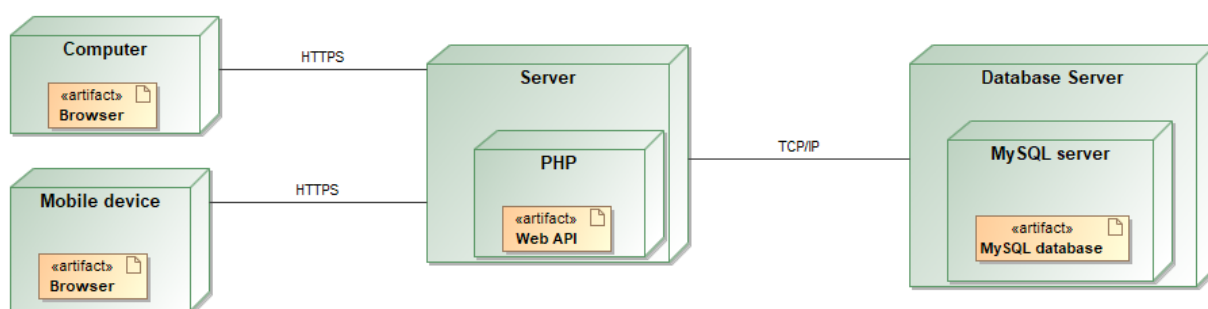
Jeigu taip – duomenys įrašomi / paimami iš duomenų bazės ir grąžinamas HTTP rezultatas vartotojui. Kitu atveju vartotojas gauna atsaką, reiškiantį, kad šis veiksmas jam nėra galimas.

1.5.Sistemos sudėtis

Sistema sudaryta iš trijų dalių:

- Vartotojo sąsajos – ją sudaro „JavaScript“ failai, „HTML“ pagrindinis failas („index.html“). 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.
- Valdymo dalies – ją sudaro „PHP“ klasių failai(plėtinys: „php“), konfigūraciniai failai (plėtiniai: „yaml“, „xml“, „env“), migraciniai failai (plėtinys: „sql“).
- Duomenų – duomenų bazėje saugoma informacija, kuri yra naudojama sistemos funkcionalumui įgyvendinti.

1.6.Komponentai



Pav. 1 Komponentų diegimo diagrama

1.7.Sąsajos įvertinimas pagal vartotojo patyrimą, profesinę terminologiją

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

1.8.Duomenų kontrolė

Vykdoma dviejų lygių validacija:

- Pirminė validacija – tai „front-end“ dalyje vykdoma validacija, kuri patikrina įvestus laukus pagal reikalavimus ir jeigu yra klaidų – nesiunčia užklauso ir grąžina klaidų pranešimus.
- Antrinė validacija – tai „back-end“ dalyje vykdoma validacija, kuri yra tokia pati, kaip ir pirminė, tačiau būtina, nes API yra atskirtas nuo „front-end“ dalies ir yra nepriklausoma nuo jos.

Jeigu nebūtų pirminės validacijos, tai vartotojui tektų laukti atsako iš pateiktos užklauso, o jeigu nebūtų antrinės validacijos, būtų lengva „nulaužti“ sistemos „back-end“ dalį, nes užklauso galima siųsti naudojantis tokiais įrankiais, kaip „Postman“, „Insomnia“, ir apeiti pirminę validaciją.

2. Testavimo eiga ir rezultatai

2.1.Testavimo planas

Sistema buvo testuojama pasitelkiant tokias testavimo priemones:

- Vienetų („Unit“) testai.
- Integraciniai testai
- Rankinis testavimas

Taip pat, parašytam kodui buvo pritaikyta statinė kodo analizė, naudojant „Eslint“ statinės kodo analizės įrankį.

Iš viso buvo parašyta 381 testai. Padengimas:

All files

73.35% Statements 1167/1591 48.39% Branches 301/622 68.99% Functions 307/445 72.24% Lines 1041/1441

2.2.Vienetų testai

Vienetų testams buvo pasitelkiama „Jest“ testavimo biblioteka. Buvo testuojama pagrindė pagalbinės funkcijos ir metodai.

2.3.Integraciniai testai

Integraciniai testai buvo rašomi pasitelkiant kelias bibliotekas:

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

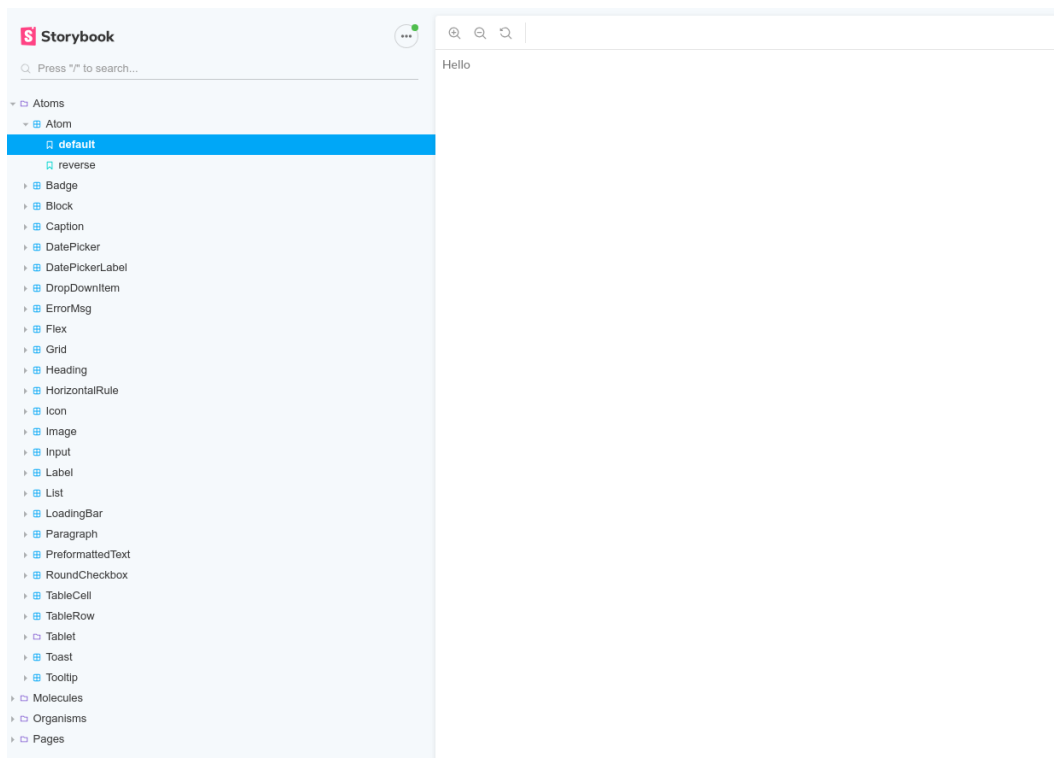
„Storybook“ bibliotekos pagalba parašyti testai pagrindė tikrina kaip komponentai atrodys vartotojų sąsajoje, su galimybe manipuluoti jiems paduodamus parametrus. Šiais testais padengta apie 80% naudojamų komponentų.

```
import React from 'react'
import styled from 'styled-components'
import { storiesOf } from '@storybook/react'
import Image from '.'
import bookyLogo from '../images/logo.svg'

const Background = styled.div`
  background-color: grey;
`

storiesOf('Atoms/Image', module)
  .add('default', () => (
    <Background>
      <Image src={bookyLogo} />
    </Background>
  ))
  .add('portrait', () => (
    <Background>
      <Image src="https://www.w3schools.com/w3css/img_lights.jpg" portrait />
    </Background>
  ))
```

Pav. 2 „Storybook“ testo aprašymo pavyzdys



Pav. 3 „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.

```
import snakeCase from 'lodash/snakeCase'

const action = prefix =>
  new Proxy(
    {},
    {
      get: (target, suffix) => () => ({
        type: snakeCase(`${prefix}_${suffix}`).toUpperCase(),
      }),
      apply: () => () => ({
        type: snakeCase(prefix).toUpperCase(),
      }),
    },
  )

const actions = new Proxy(
  {},
  {
    get: (target, property) => {
      Run Test | Debug Test
      if (/^[A-Z_]+$/.test(property)) {
        return property
      }
      return action(property)
    },
  },
)

module.exports = actions
```

Pav. 4 Netikras action objektas

```

const wrap = (props = {}) => shallow(<Button {...props} />).dive()
const mount = (props = {}) => EnzymeMount(<Button {...props} />, options)

Run Test | Debug Test
it('renders with different combination of props', () => {
  wrap({ disabled: true })
  wrap({ loading: true })
  wrap({ transparent: true })
  wrap({ disabled: true, transparent: true })
})

Run Test | Debug Test
it('renders children when passed in', () => {
  const wrapper = wrap({ children: 'test' })
  expect(wrapper.contains('test')).toBe(true)
})

Run Test | Debug Test
it('renders props when passed in', () => {
  const wrapper = wrap({ type: 'submit' })
  expect(wrapper.find({ type: 'submit' })).toHaveLength(1)
})

```

Pav. 5 „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.

```

const api = {
  post: () => {},
  get: () => {},
  put: () => {},
  delete: () => {},
}

const thunk = '123'
const meta = { thunk }
const payload = { fullData: true }

Run Test | Debug Test
describe('resetAfterSubmit', () => {
  const generator = sagas.resetAfterSubmit(api, undefined, { thunk })
  const emptySelection = []
  expect(generator.next().value).toEqual(put(setSelection(emptySelection)))
  expect(generator.next().value).toEqual(put(setFormTab(0)))
  expect(generator.next().value).toEqual(
    put(change('ReservationForm', 'description', '')),
  )
  expect(generator.next().value).toEqual(
    put(untouch('ReservationForm', 'description')),
  )
  expect(generator.next().value).toEqual(put(resetParticipants()))
  expect(generator.next().done).toBe(true)
})

```


Pav. 6 „Redux“ ir „Jest“ bibliotekso testo pavyzdys

2.4. 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.

Bug #46134

 Edit  Log time  Watch


**Toast message isn't shown when user tries to create meeting for past date**
Added by QA over 1 month ago. Updated over 1 month ago.

Status:	Closed	Start date:	2020-03-26
Priority:	Normal	Due date:	
Assignee:	-	% Done:	<div><div></div></div> 0%
Category:	-	Estimated time:	
Target version:	Sprint 14 (2020.03.14 - 2020.03.28)		
Platform:	Web	In the locked Branch?:	
Secondary Version (Scrum):		Story Points:	

DescriptionQuote

Steps to reproduce:

1. Select the meeting room
2. Selecting the the previous day
3. Adding time and name for the meeting
4. Save it

Bug reproduction:  [Image](#)

Actual results: meeting was not created because of the past date, but the toast message also isn't shown for the user that meeting can;t be created because of the selecting past time.

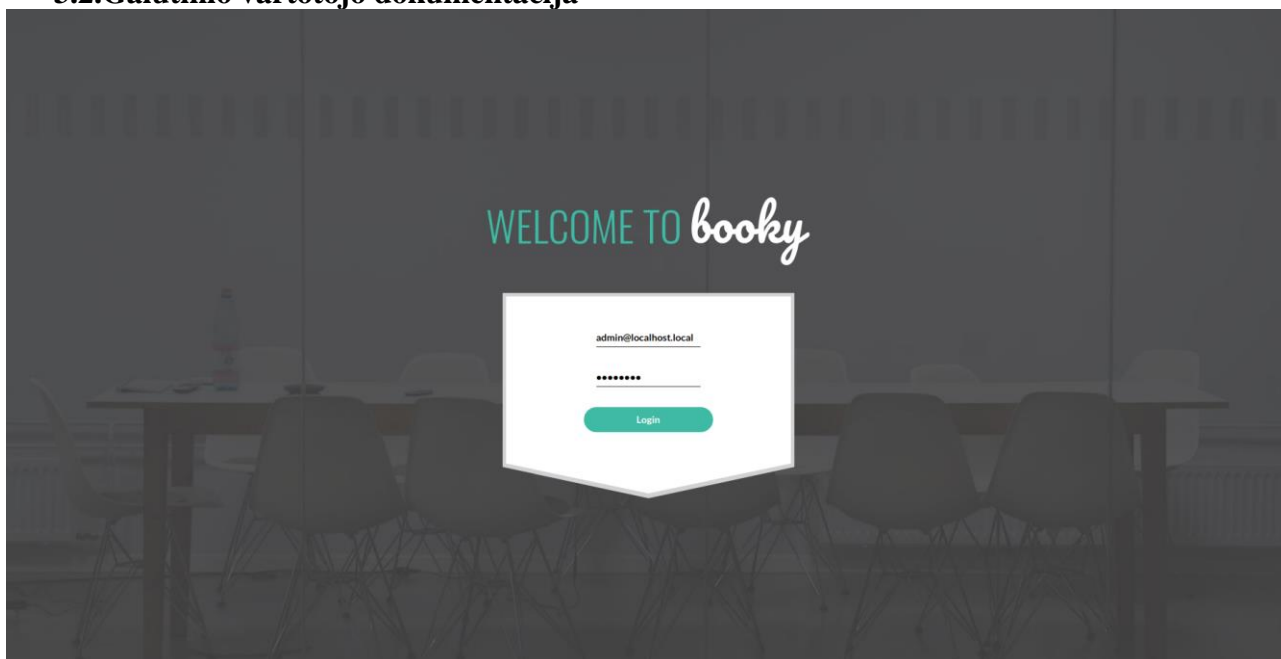
Pav. 7 Bug'o užpildymo pavyzdys

3. Dokumentacija

3.1. Apibendrintas sistemos galimybių aprašymas

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 visų 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.

3.2. Galutinio vartotojo dokumentacija



Pav. 8 Prisijungimo langas

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

STATUS ↑	NAME	BOOKED FOR	SIZE	OWNER
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	
Free	Test room Some test room		8	

Pav. 9 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, susitikimo pradžios laiką bei susitikimo narių skaičių. 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.

ROOMS LEISURE BOOKINGS ASSETS USERS

admin Administrator

Busy Free

< Test room

07:00
07:15
07:30
07:45
08:00
08:15
08:30
08:45
09:00
09:15
09:30
09:45
10:00
10:15
10:30
10:45
11:00
11:15

DATE / TIME Reset

Meeting date: 2020-05-07

Meeting starts at: Start time

Meeting ends at: End time

Meeting name:

Book

Pav. 10 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 sukūrimo atveju bus parodyta kiek iš viso buvo sukurta rezervacijų bei rezervacijos laikas atsiras laiko juostoje.

ROOMS LEISURE BOOKINGS ASSETS USERS

admin Administrator

13:30
13:45
14:00
14:15
14:30
14:45
15:00
15:15
15:30
15:45
16:00
16:15
16:30
16:45
17:00
17:15
17:30
17:45
18:00
18:15
18:30
18:45

booking Admin Admin

DATE / TIME Reset

Meeting date: 2020-05-07

Meeting starts at: 16:30

Meeting ends at: 17:15




Meeting name: booking

Update
Delete
Deselect

Pav. 11 Rezervacijos redagavimas

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

The image shows a mobile application interface for booking a meeting. At the top, there is a navigation bar with three icons: a calendar, a refresh/circular arrow, and a group of people. Below this, the main section is titled "DATE / TIME" in white, with a "Reset" link in teal to its right. The form contains four input fields, each with a teal underline and navigation arrows: "Meeting date:" with the value "2020-05-07", "Meeting starts at:" with the value "15:15", "Meeting ends at:" with the value "15:45", and "Meeting name:" which is currently empty. At the bottom of the screen, there are two large, rounded buttons: a teal "Book" button and a light gray "Deselect" button.



DATE / TIME [Reset](#)

Meeting date:
< 2020-05-07 >

Meeting starts at:
< 15:15 >

Meeting ends at:
< 15:45 >




Meeting name:

Book

Deselect

Pav. 12 Rezervacijos informacija

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



RECURRENCE

Reset

Repeat every

< 1 >

< Week >

Repeat on

M

T

W

T

F

S

S

Ends

☐ Never

☒ On < 2020-05-07 >

☐ Repeat < 1 >

Book

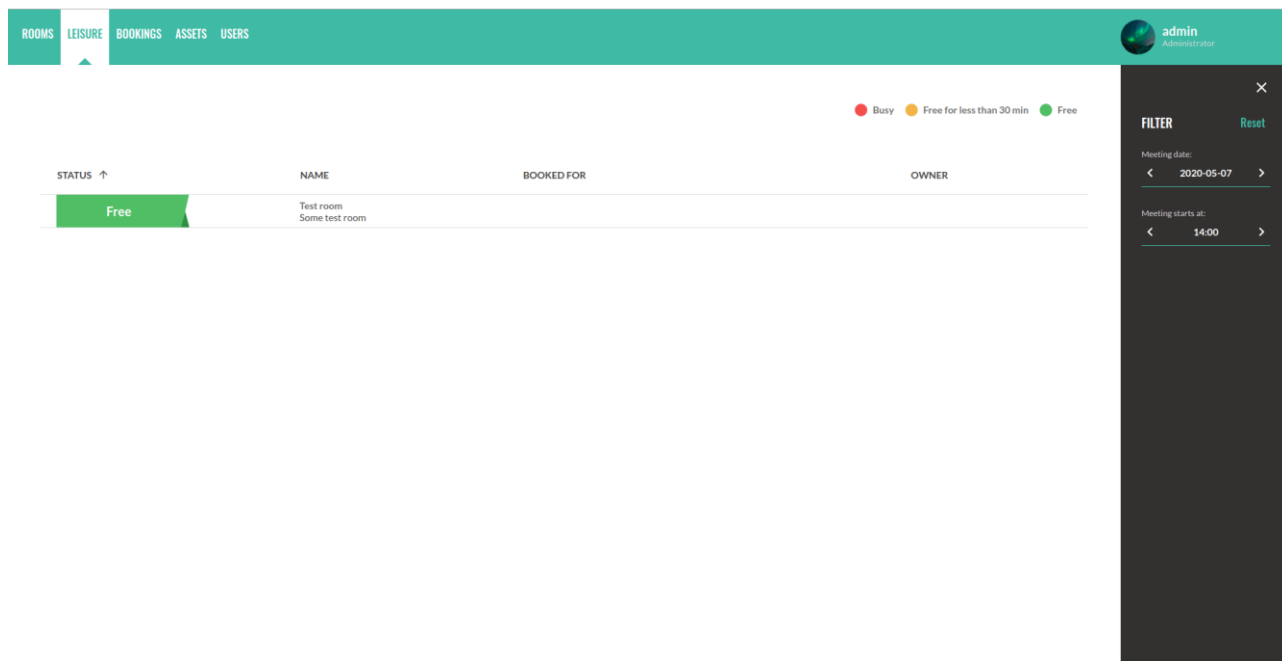
Pav. 13 Rezervacijos pasikartojimo opcijos

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. 14 Susitikimo narių pridėjimas

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



Pav. 15 Laisvalaikio įrangos langas

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

DATE	TIME	NAME	ROOM	PARTICIPANTS	OWNER
2019-12-05	12-45-13:30	test	Test room	-/8	Admin Admin
2019-12-19	11:15-12:15	sada	Test room	-/8	Admin Admin
2019-12-20	14-45-16:15	test	Test room	-/8	Admin Admin
2019-12-22	14-45-16:15	test	Test room	-/8	Admin Admin
2019-12-24	14-45-16:15	test	Test room	-/8	Admin Admin
2019-12-26	14-45-16:15	test	Test room	-/8	Admin Admin
2019-12-28	14-45-16:15	test	Test room	-/8	Admin Admin
2019-12-30	14-45-16:15	test	Test room	-/8	Admin Admin
2020-01-01	14-45-16:15	test	Test room	-/8	Admin Admin
2020-01-03	14-45-16:15	test	Test room	-/8	Admin Admin
2020-01-05	14-45-16:15	test	Test room	-/8	Admin Admin
2020-01-07	14-45-16:15	test	Test room	-/8	Admin Admin
2020-01-09	14-45-16:15	test	Test room	-/8	Admin Admin
2020-01-11	14-45-16:15	test	Test room	-/8	Admin Admin
2020-01-13	14-45-16:15	test	Test room	-/8	Admin Admin

Pav. 16 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.

ROOMSLEISUREBOOKINGSASSETSUSERS

adminAdministrator

ASSET	SIZE	DESCRIPTION
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room
Test room	8	Some test room

Pav. 17 Bendras susitikimo kabarių ir laisvalaikio įrangos sąrašas

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

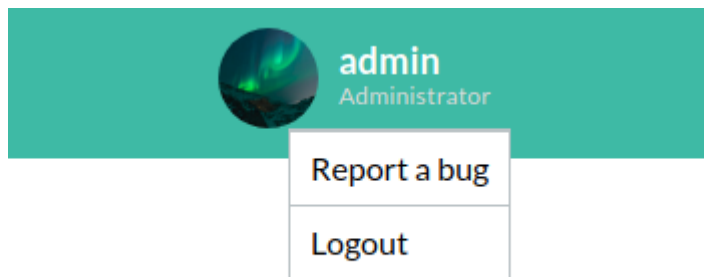
ROOMSLEISUREBOOKINGSASSETSUSERS

adminAdministrator

NAME	MAIL	ROLE
admin	admin@localhost	Administrator
qwe	qwe@localhost	User
devuser3	devuser3@devuser.local	Asset

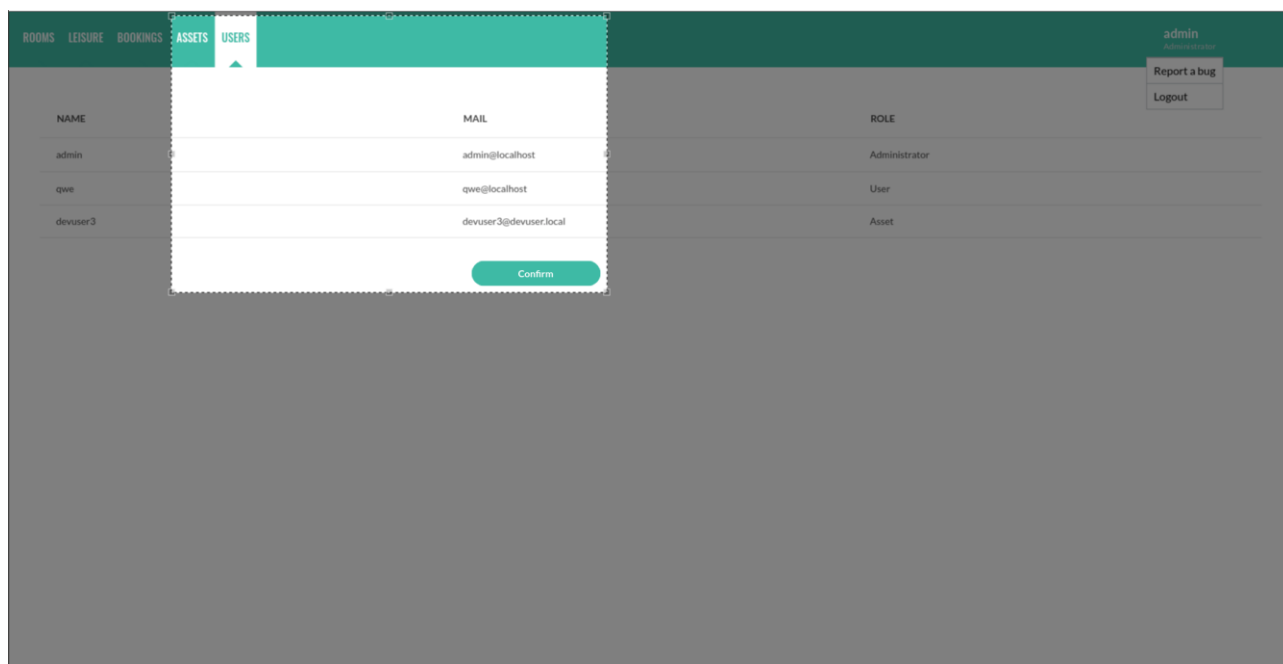
Pav. 18 Vartotojų sąrašas

Šiame lange administratorius gali matyti visų vartotojų sąrašą.



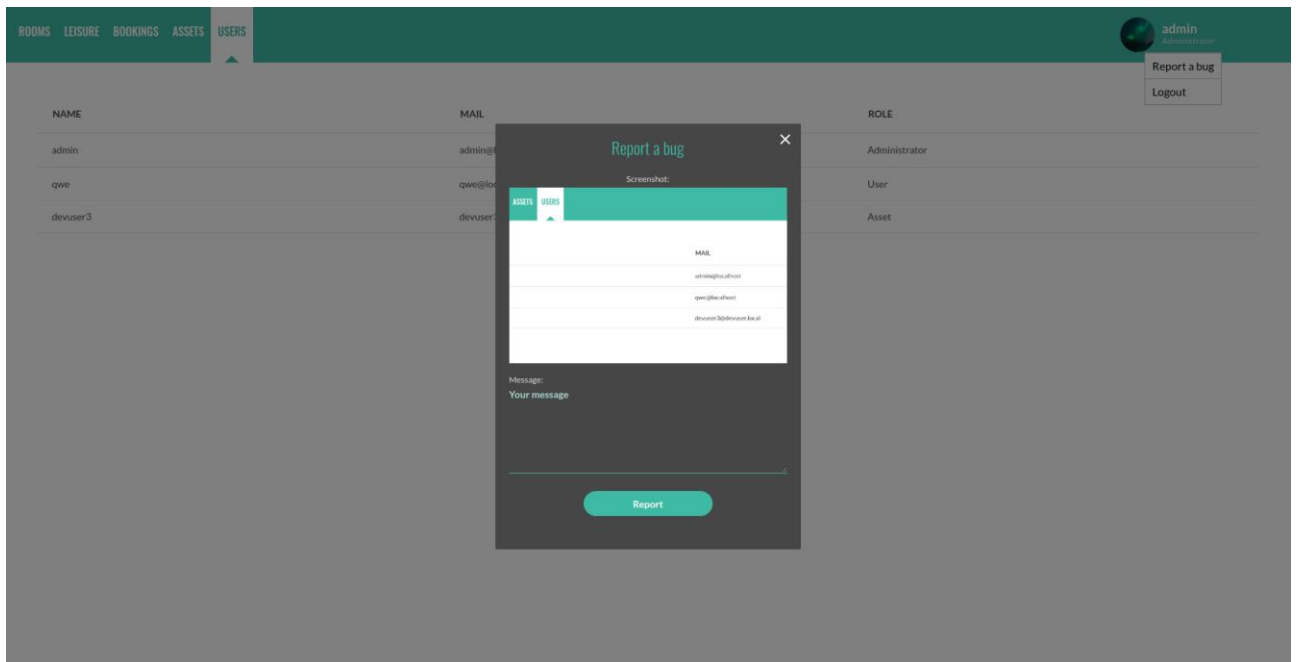
Pav. 19 Vartotojo meniu

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



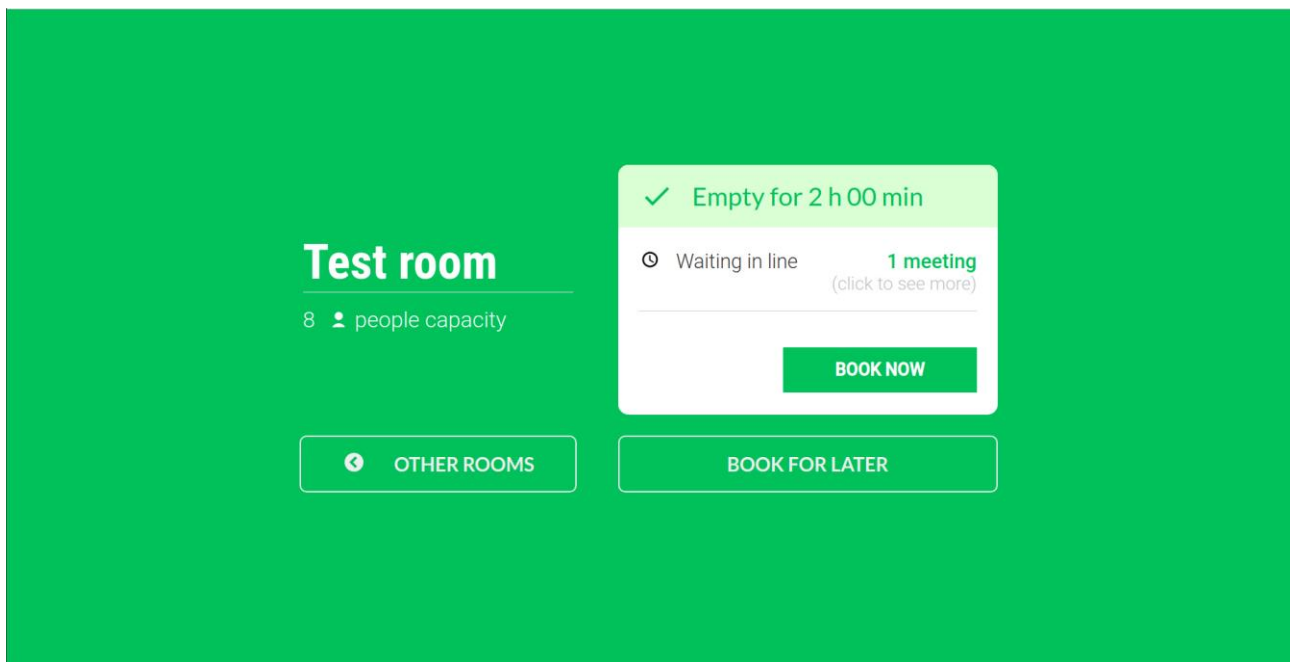
Pav. 20 Sistemos klaidos pranešimo lango fiksavimas

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



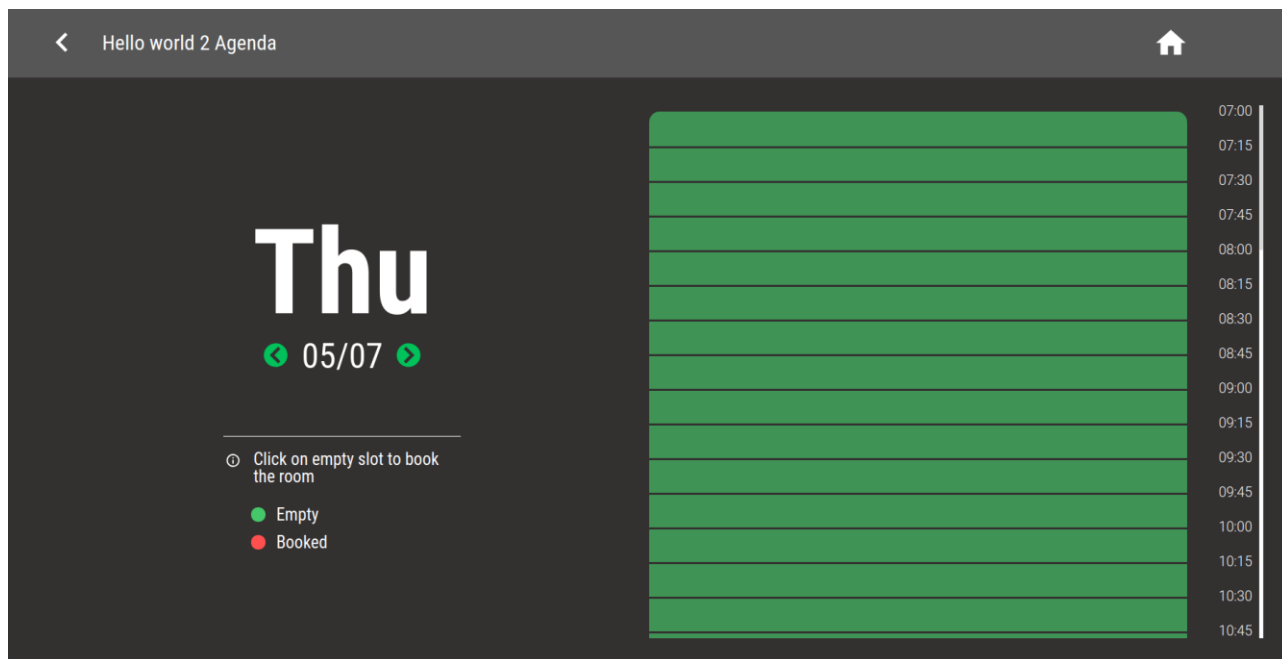
Pav. 21 Sistemos klaidos pranešimo sukūrimas

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



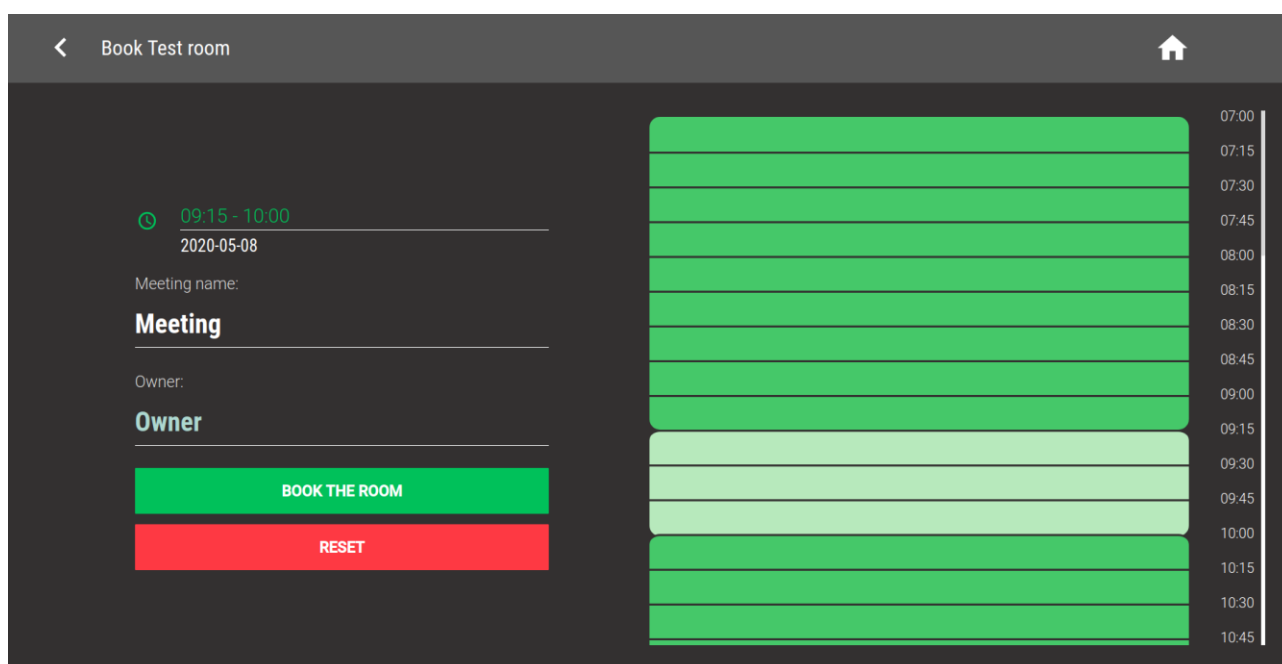
Pav. 22 Planšetės susitikimo kambario informacija

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.



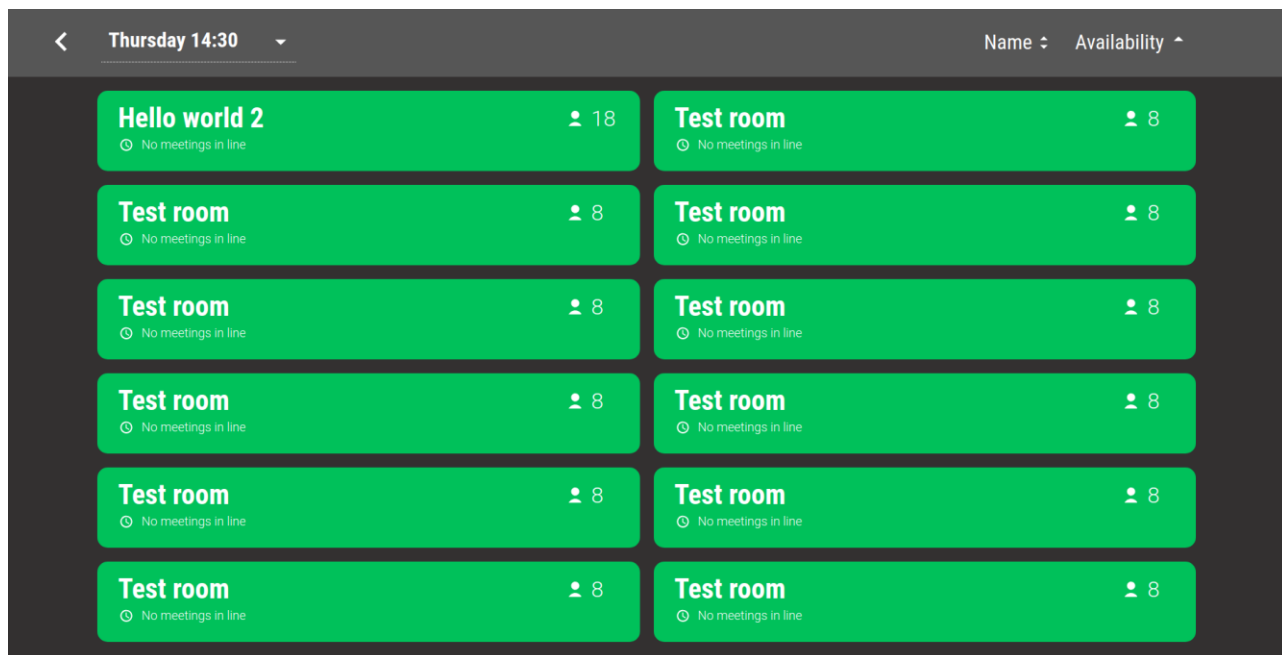
Pav. 23 Rezervacijos kūrimas planšetės lange

Šiame planšetės lange vartotojas gali kurti rezervacijas.



Pav. 24 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. 25 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.