

**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)

---

## Turinys

<b>Paveikslėlių sąrašas .....</b>	<b>4</b>
<b>Lentelių sąrašas .....</b>	<b>5</b>
<b>Įvadas.....</b>	<b>6</b>
<b>1. Analizės dalis .....</b>	<b>7</b>
1.1. Informacija apie įmonę.....	7
1.1.1. Įmonės veiklos sritis, specializacija .....	7
1.1.2. Įmonės vizija .....	7
1.1.3. Įmonės misija .....	7
1.1.4. Kokiais būdais įmonė siekia įgyvendinti savo misiją ir viziją .....	7
1.2. Informacija apie įmonėje naudojamas IT .....	7
1.3. Įmonės veiklos procesai. ....	8
1.3.1. Projekto inicijavimas.....	8
1.3.2. Projekto koncepcijos analizė. ....	8
1.3.3. Kainos ir investicijų analizė. ....	9
1.3.4. Projektų vykdymas, pristatymas ir dokumentacija. ....	9
1.3.5. Komandinis darbas. ....	9
1.3.6. Inovacijos. ....	9
1.3.7. Produktų vystymo procesai. ....	9
<b>2. Praktikos dalis .....</b>	<b>11</b>
2.1. Techninė užduotis .....	11
2.1.1. Sistemos apibrėžimas .....	11
2.1.2. Bendras veiklos tikslas .....	11
2.1.3. Sistemos pagrįstumas .....	11
2.1.4. Konkurencija rinkoje.....	11
2.1.5. Prototipai ir pagalbinių informacija.....	11
2.1.6. Siūlomas sprendimas.....	12
2.1.7. Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti .....	12
2.2. Reikalavimų specifikacija .....	12
2.2.1. Komercinė specifikacija.....	12
2.2.2. Sistemos funkcijos .....	12
2.2.3. Apribojimai .....	13
2.2.4. Duomenų ir rezultatų specifikacija .....	14
2.2.5. Vartotojo sąsajos specifikacija .....	17
2.2.6. Realizacijai keliami reikalavimai. ....	24
2.2.7. Techninė specifikacija.....	24
2.3. Projektavimo eiga ir sistemos projektas.....	25
2.3.1. Projektavimo valdymas ir eiga .....	25
2.3.2. Projektavimo technologija .....	25
2.3.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinės sistemos .....	25
2.3.4. Informacijos šaltiniai.....	25
2.3.5. Sistemos sudėtis .....	25
2.3.6. Komponentai .....	26

2.3.7. Sąsajos įvertinimas pagal vartotojo patyrimą, profesinę terminologiją .....	26
2.3.8. Duomenų kontrolė.....	26
2.4. Testavimo eiga ir rezultatai .....	26
2.4.1. Vienetų testai.....	27
2.4.2. Integraciniai testai .....	27
2.4.3. Rankiniai testai.....	30
2.5. Dokumentacija .....	31
2.5.1. Apibendrintas sistemos galimybių sąrašas .....	31
2.5.2. Galutinio vartotojo dokumentacija.....	31
<b>Išvados.....</b>	<b>43</b>
<b>Literatūros sąrašas.....</b>	<b>44</b>

## **Paveikslėlių sąrašas**

<b>Pav. 2.1</b> Panaudos atvejų diagrama .....	13
<b>Pav. 2.2.</b> Prisijungimo langas.....	18
<b>Pav. 2.3.</b> Susirinkimo kambarių sąrašo langas.....	18
<b>Pav. 2.4.</b> Susirinkimo kambarių sąrašo langas mobiliesiems įrenginiams .....	19
<b>Pav. 2.5.</b> Susirinkimo kambario rezervavimo langas.....	19
<b>Pav. 2.6.</b> Susirinkimo kambario rezervavimo langas mobiliesiems įrenginiams .....	20
<b>Pav. 2.7.</b> Visų rezervacijų sąrašo peržiūros langas .....	21
<b>Pav. 2.8.</b> Visų vartotojų sąrašo peržiūros langas .....	21
<b>Pav. 2.9.</b> Susirinkimo kambario rezervavimo formos langas mobiliesiems įrenginiams .....	22
<b>Pav. 2.10.</b> Susirinkimo kambario rezervavimo langas planšetėms.....	23
<b>Pav. 2.11.</b> Susirinkimo kambario informacijos peržiūros langas planšetėms.....	23
<b>Pav. 2.12.</b> Susirinkimo kambarių sąrašo peržiūros langas planšetėms.....	24
<b>Pav. 2.13</b> Komponentų diegimo diagrama.....	26
<b>Pav. 2.14</b> Kodo padengimas .....	27
<b>Pav. 2.15</b> „Storybook“ testo aprašo pavyzdys .....	27
<b>Pav. 2.16</b> „Storybook“ komponentų testavimo sąsajos langas .....	28
<b>Pav. 2.17</b> Netikras action objektas .....	29
<b>Pav. 2.18</b> „Enzyme“ ir „Jest“ bibliotekų testo pavyzdys .....	29
<b>Pav. 2.19</b> „Redux“ ir „Jest“ bibliotekso testo pavyzdys .....	30
<b>Pav. 2.20</b> Bug'o užpildymo pavyzdys .....	30
<b>Pav. 2.22</b> Prisijungimo langas.....	31
<b>Pav. 2.23</b> Susitikimo kambarių sąrašo langas.....	32
<b>Pav. 2.24</b> Rezervacijos kūrimo langas .....	32
<b>Pav. 2.25</b> Rezervacijos redagavimas.....	33
<b>Pav. 2.26</b> Rezervacijos informacija .....	34
<b>Pav. 2.27</b> Rezervacijos pasikartojimo opcijos .....	35
<b>Pav. 2.28</b> Susitikimo narių pridėjimas .....	36
<b>Pav. 2.29</b> Laisvalaikio įrangos langas.....	37
<b>Pav. 2.30</b> Visų rezervacijų langas .....	37
<b>Pav. 2.31</b> Bendras susitikimo kabarių ir laisvalaikio įrangos sąrašas.....	38
<b>Pav. 2.32</b> Vartotojų sąrašas.....	38
<b>Pav. 2.33</b> Vartotojo meniu .....	39
<b>Pav. 2.34</b> Sistemos klaidos pranešimo lango fiksavimas.....	39
<b>Pav. 2.35</b> Sistemos klaidos pranešimo sukūrimas .....	40
<b>Pav. 2.36</b> Planšetės susitikimo kambario informacija .....	40
<b>Pav. 2.37</b> Rezervacijos kūrimas planšetės lange.....	41
<b>Pav. 2.38</b> Rezervacijos kūrimas planšetės lange.....	41
<b>Pav. 2.39</b> Planšetės lango susitikimo kambarių sąrašas.....	42

## **Lentelių sąrašas**

<b>Lentelė 2.1</b>	Prisijungimo duomenys .....	14
<b>Lentelė 2.2</b>	Rezervacijos kūrimo duomenys.....	14
<b>Lentelė 2.3</b>	Klaidos pranešimo duomenys.....	14
<b>Lentelė 2.4</b>	Susirinkimo kambario elemento sąraše peržiūros rezultatai .....	15
<b>Lentelė 2.5</b>	Rezervacijos elemento sąraše rezultatai .....	15
<b>Lentelė 2.6</b>	Laisvalaikio įrangos elemto sąraše rezultatai .....	16
<b>Lentelė 2.7</b>	Vartotojo elemento sąraše rezultatai.....	16
<b>Lentelė 2.8</b>	Sėkmingos kambario rezervacijos/ų sukūrimo rezultatai .....	17
<b>Lentelė 2.9</b>	Susirinkimų kambarių rezervacijos elemento sąraše rezultatai .....	17
<b>Lentelė 2.10</b>	Techninė specifikacija .....	24

## **Išvadas**

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 sukurama 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.

## **1. Analizės dalis**

### **1.1. Informacija apie įmonę**

#### **1.1.1. 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.

#### **1.1.2. Įmonės vizija**

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

#### **1.1.3. Įmonės misija**

Suteikiant įrankius padėti verslams bei IT klientams tobulėti

#### **1.1.4. Kokiais būdais įmonė siekia įgyvendinti savo misiją ir viziją**

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

„TeleSoftas“ laikosi „Step forward“ taisyklės – 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 padidinant iniciatyvos lygį. Plėtojamas patirties dalinimasis, bendruomeniškumas. Darbuotojams sudaromos visos galimybės įsitraukti į veiklą bei priimti efektyviausius sprendimus – taip skatinant tobulėti.

### **1.2. Informacija apie įmonėje naudojamą IT**

Frontend programavimo technologijos/kalbos:

- JavaScript – objektiškai orientuota skriptų 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 orientuota programavimo kalba, sukurta Microsoft kompanijoje kaip dalis .NET iniciatyvos
- Java – bendrosios paskirties, objektiškai orientuota programavimo kalba
- PHP – plačiai paplitusi dinaminė interpretuojama programavimo kalba
- Go arba Golang – kompiliuojama objektinio programavimo kalba.
- Scala – 2003 m. sukurta programavimo kalba, sukurta glaustai išreikšti dažniausius programavimo šablonus, struktūras.

Mobilios aplikacijos:

- Android
  - Kotlin – „JetBrains“ sukurta funkcinė ir objektinė programavimo kalba.
  - Java – bendrosios paskirties, objektiškai orientuota 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.

### **1.3. Įmonės veiklos procesai.**

#### **1.3.1. 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.

#### **1.3.2. 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

### **1.3.3. Kainos ir investicijų analizė.**

Peržiūrint sudarytą funkcionalumų sąrašą, jį per kelias projekto vertinimo susitikimus analizuoja su projekto įgyvendinimu susiję asmenys: vyresnieji programuotojai, 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.

### **1.3.4. Projektų vykdymas, pristatymas ir dokumentacija.**

Projektų vykdymo, pristatymo ir dokumentavimo procesas:

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

### **1.3.5. Komandinis darbas.**

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

### **1.3.6. 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.

### **1.3.7. 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.

## **2. Praktikos dalis**

### **2.1. Techninė užduotis**

#### **2.1.1. Sistemos apibrėžimas**

Sistema „Booky“, skirta „Telesoftas“ įmonės, Vilniaus ir Kauno biurų susirinkimo kambarių bei laisvalaikio zonų rezervavimams atlikti, stebėti jų užimtumą. Priėmimą 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 ją rezervuoti. Pastarąjį režimą naudoja specialios planšetės, prikabinotos prie kiekvieno susirinkimo kambario įėjimo.

Mano darbo dalis yra sukurti šios sistemos vartotojo sąsają.

#### **2.1.2. Bendras veiklos tikslas**

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

#### **2.1.3. Sistemos pagrindumas**

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

#### **2.1.4. 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.

#### **2.1.5. Prototipai ir pagalbinių 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

### **2.1.6. 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 sąsaja kurti iš naujo, panaudojant kai kuriuos dizainus iš senesnės sistemos.

### **2.1.7. Sistemos apimtis ir ištekliai, reikalingi sistemai sukurti**

Sistema sudaryta iš trijų posistemų: 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.

## **2.2. Reikalavimų specifikacija**

### **2.2.1. 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.

### **2.2.2. 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

### 2.2.3. 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.

#### 2.2.4. 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	<pre>{   description: "yes"   end_date: "2020-04-11T17:45:00+00:00"   id: 167   owner: {     email: "admin@localhost"     id: 1     profile: {       first_name: "Admin"       full_name: "Admin Admin"       id: 1       last_name: "Admin"     }     roles: ["ROLE_API_ADMIN", "ROLE_API_USER"]     username: "admin"   }   start_date: "2020-04-11T17:30:00+00:00"   type: "reservation" }</pre>
type	String	room

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

Laukas	Tipas	Pavyzdys
asset	Room	<pre>created_at: "2019-12-05T08:40:37+00:00" description: "Some test room" id: 1 name: "Test room" size: 8 type: "room" updated_at: "2019-12-05T08:40:37+00:00"</pre>
description	String	Reservation description
end_date	Date	2019-12-05T11:30:00+00:00
id	Integer	3
owner	User	<pre>email: "admin@localhost" id: 1 profile: {   first_name: "Admin"   full_name: "Admin Admin"   id: 1   last_name: "Admin" } roles: ["ROLE_API_ADMIN", "ROLE_API_USER"] username: "admin"</pre>
settings	Settings	<pre>ends_on: "2019-12-05T00:00:00+00:00" repeat_every_frequency: "DAILY" repeat_every_interval: 1 type: "custom"</pre>
start_date	Date	2019-12-05T10:45:00+00:00
state	String	Active
type	String	recurring

**Lentelė 2.5** Rezervacijos elemento sąrašo 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	<pre>{   description: "yes"   end_date: "2020-04-11T17:45:00+00:00"   id: 167   owner: {     email: "admin@localhost"     id: 1     profile: {       first_name: "Admin"       full_name: "Admin Admin"       id: 1       last_name: "Admin"     }     roles: ["ROLE_API_ADMIN", "ROLE_API_USER"]     username: "admin"   }   start_date: "2020-04-11T17:30:00+00:00"   type: "reservation" }</pre>

Lentelė 2.6 Laisvalaikio įrangos elemento sąrašo rezultatai

Laukas	Tipas	Pavyzdys
id	Int	1
email	String	<u>Admin@localhost.com</u>
roles	String array	[„ROLE_ADMIN“, „ROLE_USER“]
username	String	Admin
profile	Profile	<pre>first_name: "Admin" full_name: "Admin Admin" id: 1 last_name: "Admin"</pre>

Lentelė 2.7 Vartotojo elemento sąrašo rezultatai

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



total	Total	<pre>{   created: 1   excluded: 1 }</pre>
-------	-------	---

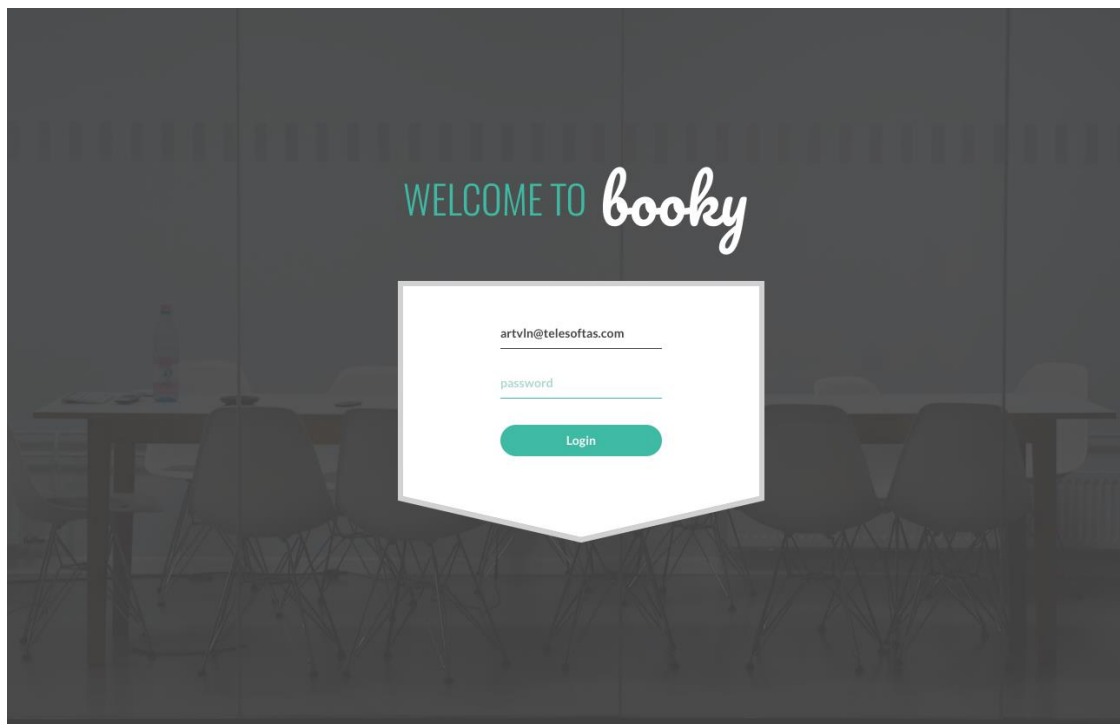
**Lentelė 2.8** Sėkmingos kambario rezervacijos/ų sukūrimo rezultatai

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	<pre>email: "admin@localhost" id: 1 profile: {   first_name: "Admin"   full_name: "Admin Admin"   id: 1   last_name: "Admin" } roles: ["ROLE_API_ADMIN", "ROLE_API_USER"] username: "admin"</pre>
settings	Settings	<pre>ends_after: 2 repeat_every_frequency: "DAILY" repeat_every_interval: 1 type: "custom"</pre>

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

### 2.2.5. 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

ROOMS

LEISURE

BOOKINGS

ROOM MANAGEMENT

USER MANAGEMENT

Search...

Mary Smith

Administrator

Busy

Free for less than 30 min

Free

STATUS	NAME	BOOKED FOR	SIZE	OWNER
<div>12:30 - 13:00</div> <div>(15 min left)</div>	<div>Green grass</div> <div>With fake green grass</div>	Weekly stand up	8	Artūras Valenta
<div>12:30 - 13:00</div> <div>(15 min left)</div>	<div>Green grass</div> <div>With fake green grass</div>	Weekly stand up	8	Arvydas
<div>12:30 - 13:00</div> <div>(15 min left)</div>	<div>Green grass</div> <div>With fake green grass</div>	Weekly stand up	8	Artūras Valenta
<div>12:30 - 13:00</div> <div>(15 min left)</div>	<div>Green grass</div> <div>With fake green grass</div>	Weekly stand up	8	Artūras Valenta
<div>12:30 - 13:00</div> <div>(15 min left)</div>	<div>Green grass</div> <div>With fake green grass</div>	Weekly stand up	8	Artūras Valenta

FILTER

Reset

Meeting date:

<

2018-08-16

>

Meeting starts at:

<

13:00

>

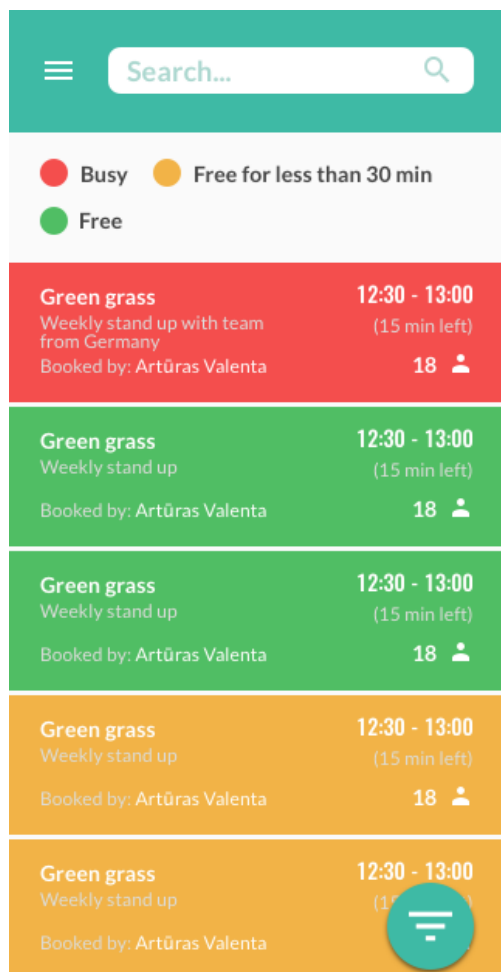
Number of participants:

<

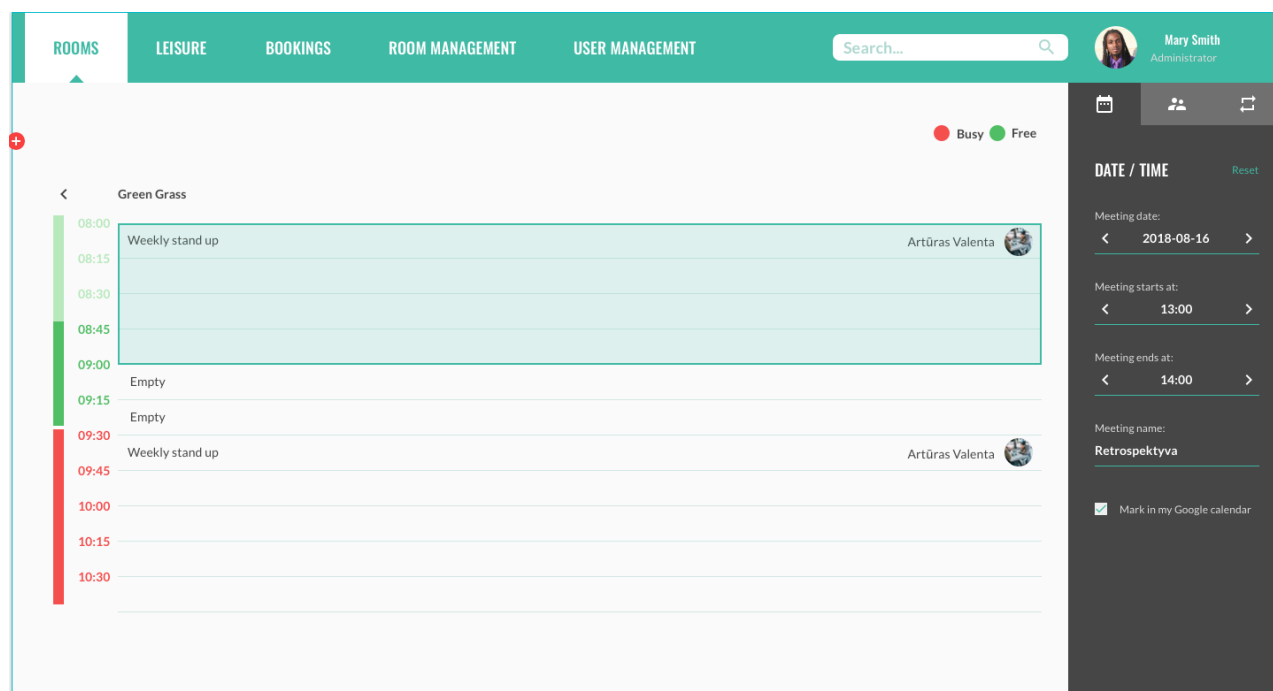
2

>

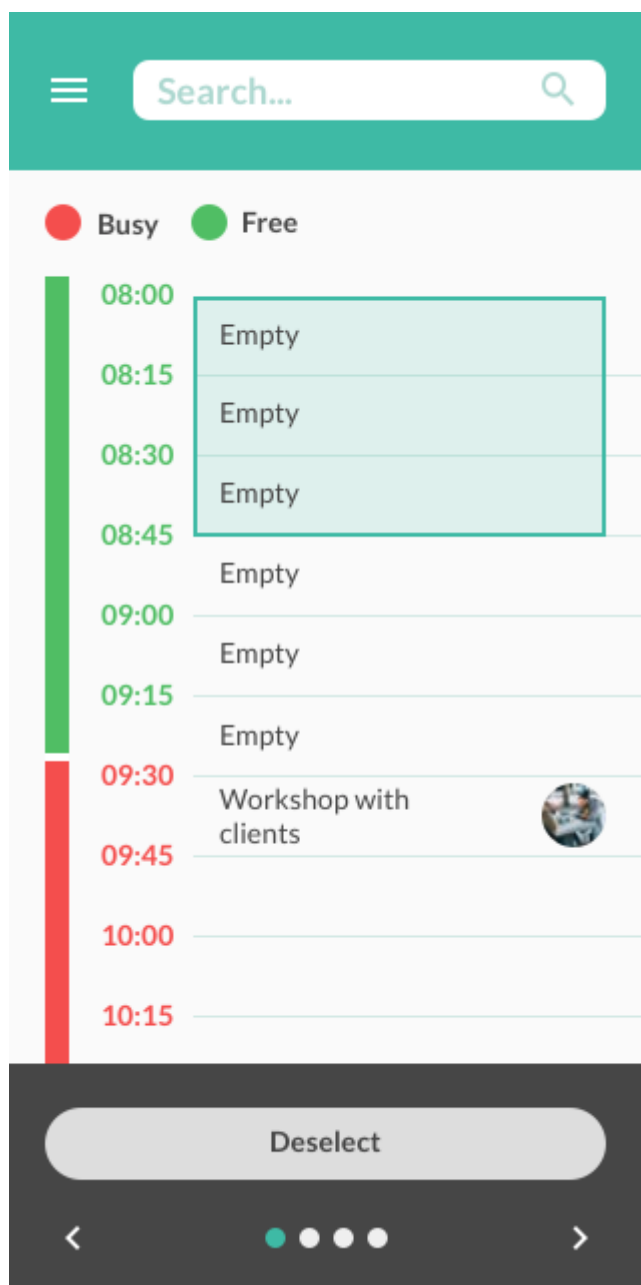
**Pav. 2.3.** Susirinkimo kambarių sąrašo langas



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



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



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

ROOMS	LEISURE	BOOKINGS	ROOM MANAGEMENT	USER MANAGEMENT	Search...	Mary Smith Administrator
						<div> <div></div> <div>Busy</div> <div>Free</div> </div>
DATE	TIME	NAME	ROOM	PARTICIPANTS	OWNER	
2018-09-12	14:00-16:00	Daily standup	Green Grass	3/8	Artūras Valenta	
2018-09-12	14:00-16:00	Daily standup	Green Grass	3/8	Artūras Valenta	
2018-09-12	14:00-16:00	Daily standup	Green Grass	3/8	Artūras Valenta	
2018-09-12	14:00-16:00	Daily standup	Green Grass	3/8	Artūras Valenta	

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

ROOMS	LEISURE	BOOKINGS	ASSETS	USERS	admin Administrator
NAME	MAIL	ROLE			
admin	admin@localhost	Administrator			
qwe	qwe@localhost	User			
devuser3	devuser3@devuser.local	Asset			

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

CUSTOM RECURRENCE

×

Repeat every

< 1 >

< week >

Repeat on

M

T

W

T

F

S

S

273px

Ends

☐ Never

☐ On

< Dec 12, 2018 >

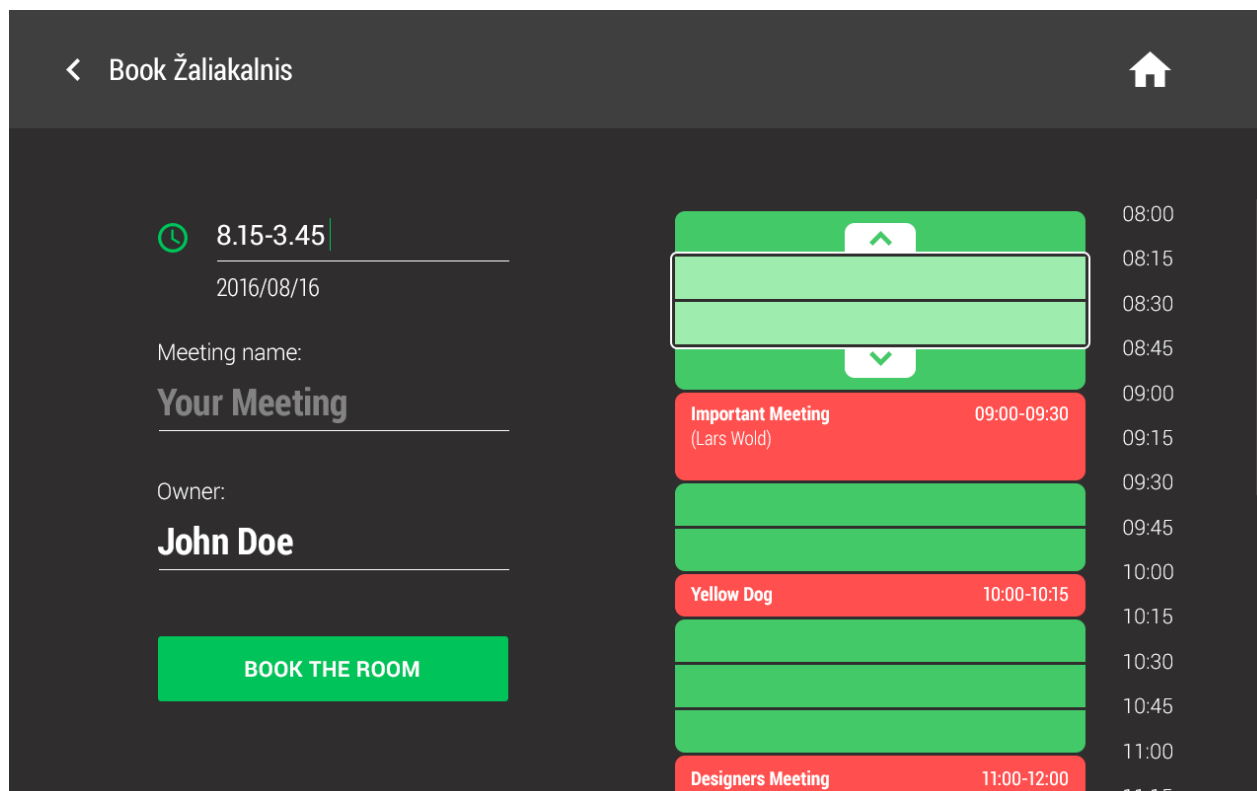
☐ After

< 13 occurrences >

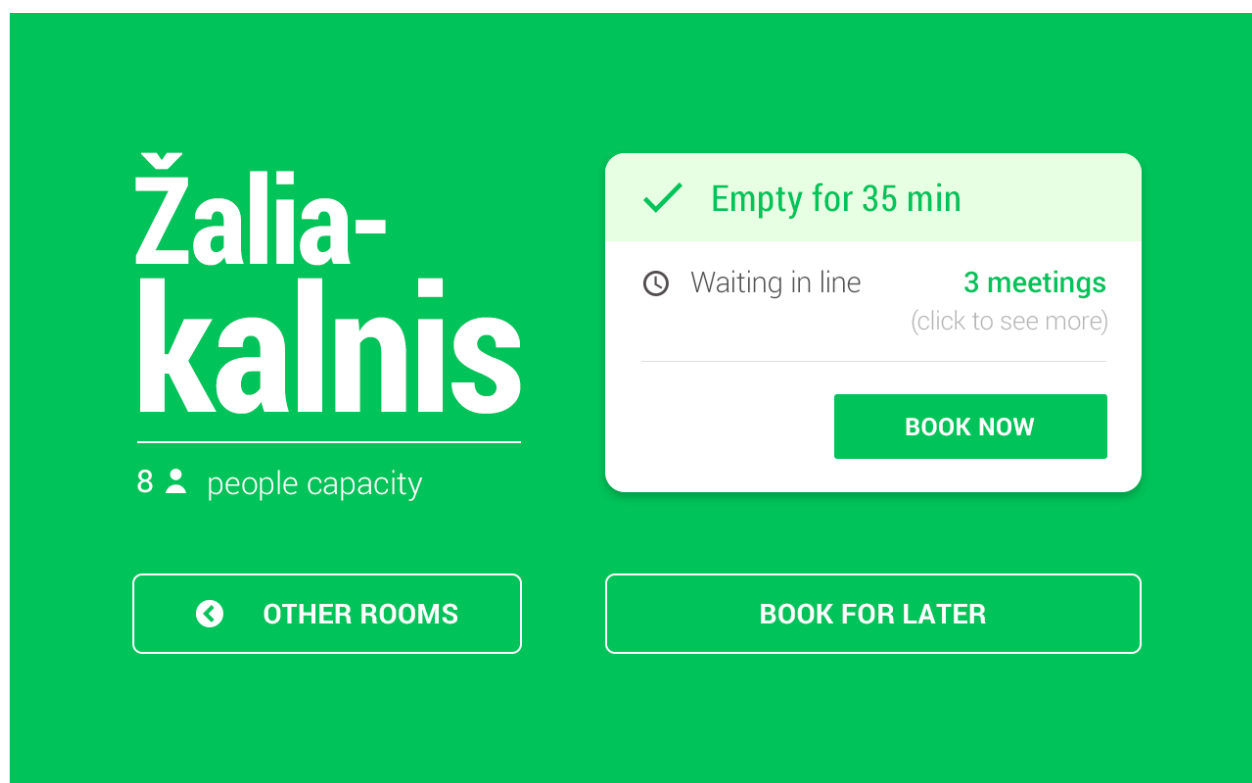
1

Apply

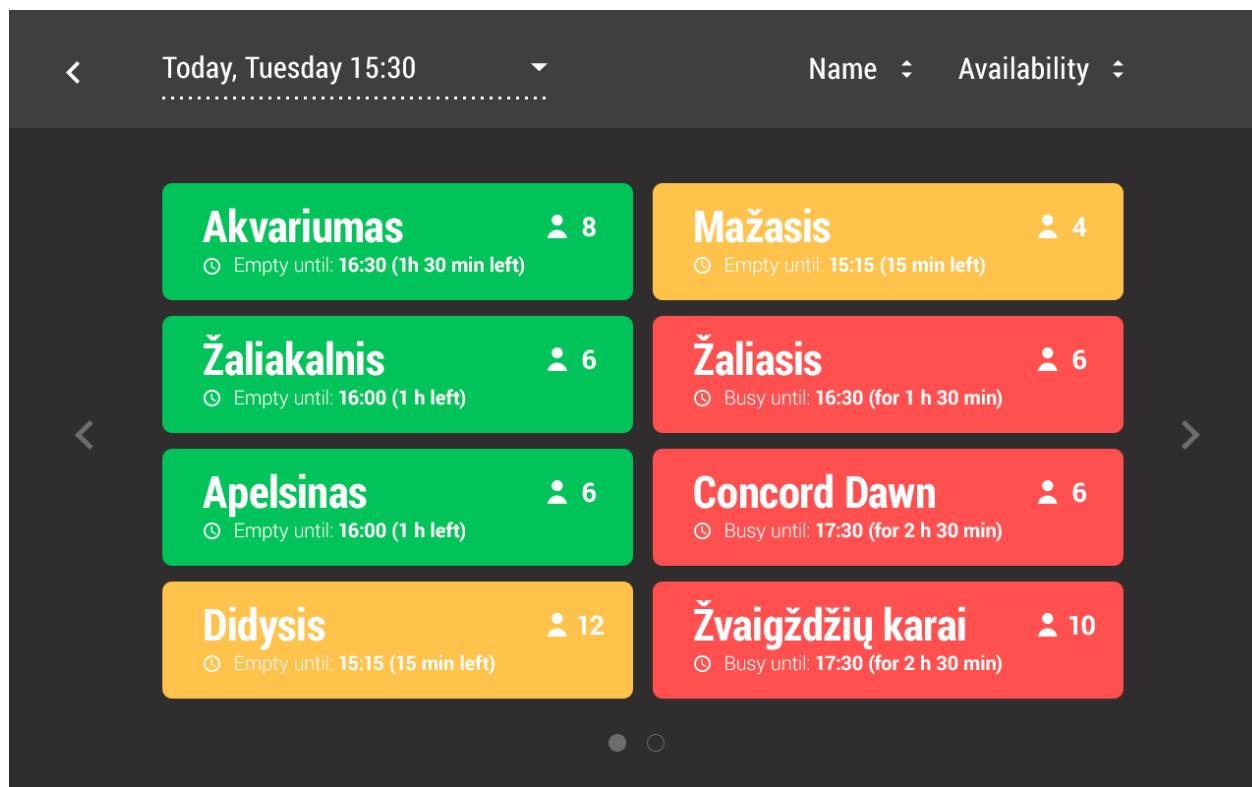
**Pav. 2.9.** Susirinkimo kambario rezervavimo formos langas mobiliems įrenginiams



**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

## 2.2.6. 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

## 2.2.7. 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



## **2.3. Projektavimo eiga ir sistemos projektas**

### **2.3.1. Projektavimo valdymas ir eiga**

Projektas buvo kuriamas pasinaudojant iteraciniu „Scrum“ projektavimo modeliu.

Vykdomo 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 funkcionalumo 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.

### **2.3.2. 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.

### **2.3.3. 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.

### **2.3.4. 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.

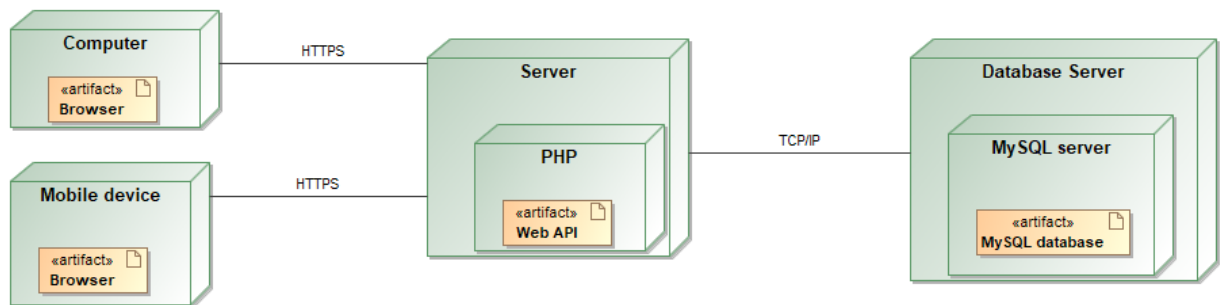
### **2.3.5. Sistemos sudėtis**

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

### 2.3.6. Komponentai



Pav. 2.13 Komponentų diegimo diagrama

### 2.3.7. 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.

### 2.3.8. 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.

## 2.4. Testavimo eiga ir rezultatai

Sistema buvo ištestuota šiais testavimo būdais:

- Vientų 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:

## All files

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

Pav. 2.14 Kodo padengimas

### 2.4.1. Vienetų testai

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

### 2.4.2. Integraciniai testai

Integraciniai testai buvo rašomi pasitelkiant kelias bibliotekas:

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

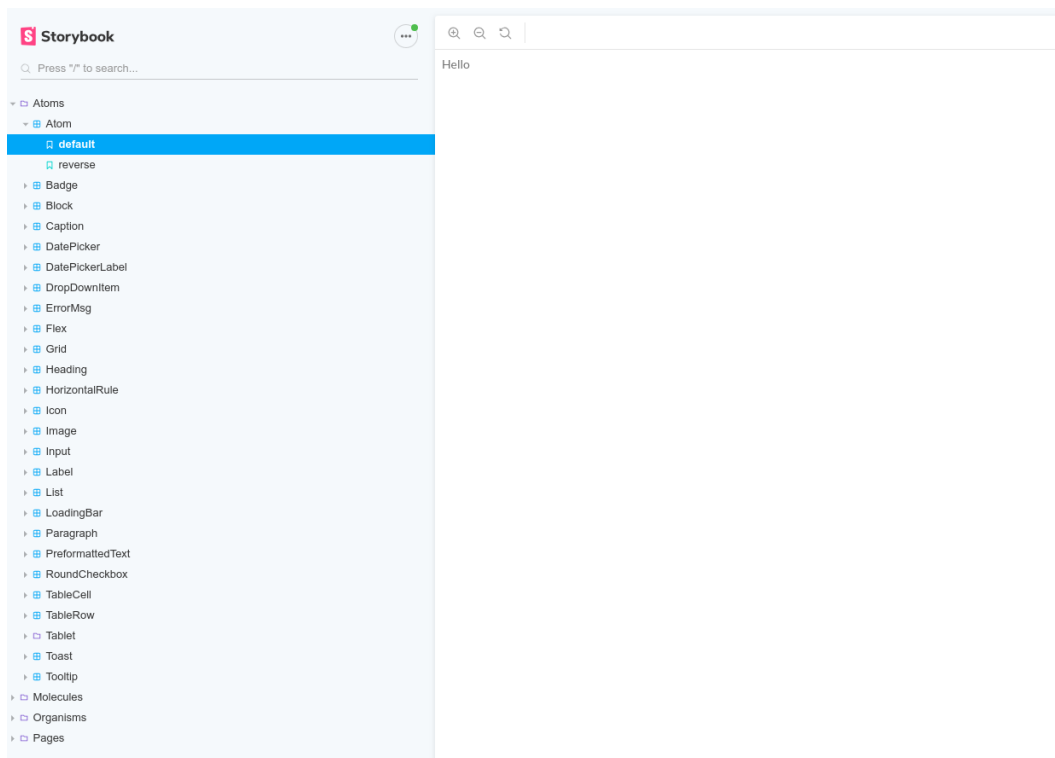
„Storybook“ bibliotekos pagalba parašyti testai pagrinde tikrina kaip komponentai atrodys vartotojų sąsajoje, su galimybe manipuluoti jiems paduodamus parametrus. Šiais testais padengta apie 80% naudojamų komponentų.

```
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. 2.15 „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.

```
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. 2.17 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. 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.

```

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

### 2.4.3. Rankiniai testai

Rankiniai testai buvo atliekami pagrindė mūsų komandos testuotojų ir tai buvo pagrindė 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.

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

**Description**
[Quote](#)

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.

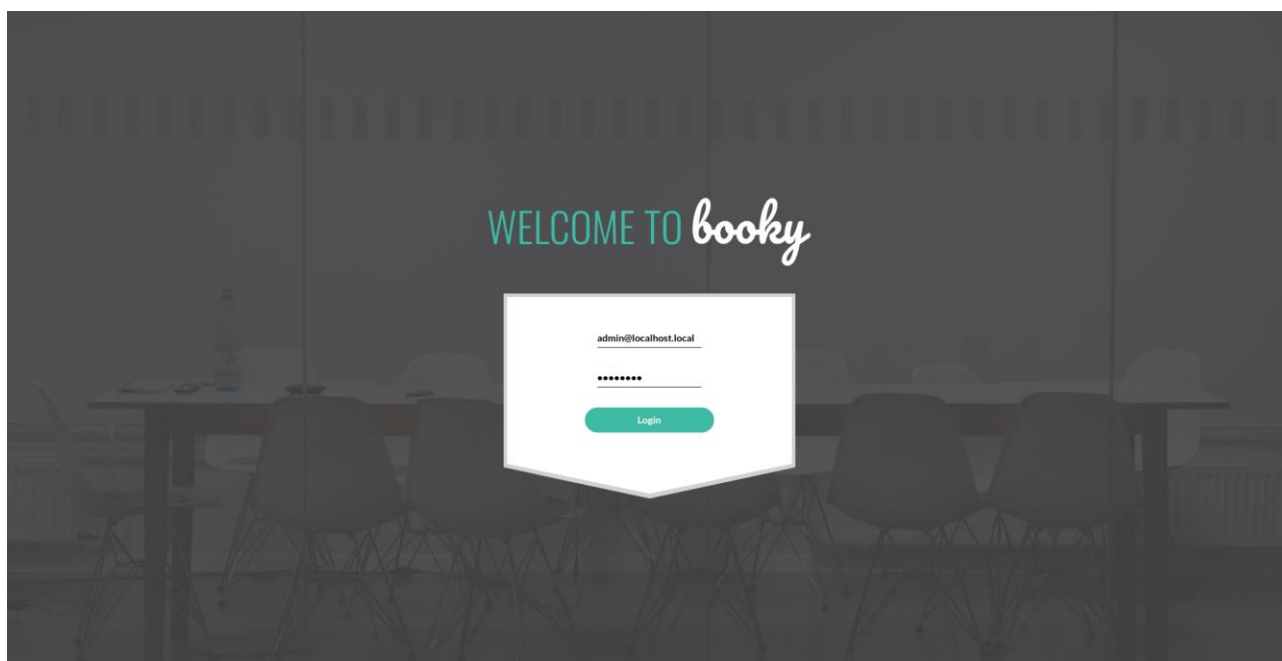
Pav. 2.20 Bug'o užpildymo pavyzdys

## 2.5. Dokumentacija

### 2.5.1. 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 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.

### 2.5.2. Galutinio vartotojo dokumentacija



**Pav. 2.21** 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.

ROOMS LEISURE BOOKINGS ASSETS USERS

admin Administrator

● Busy ● Free for less than 30 min ● Free

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	

FILTER Reset

Meeting date: < 2020-05-07 >

Meeting starts at: < 13:30 >

Number of participants: < 1 >

**Pav. 2.22** 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. 2.23** 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 kūrimo atveju bus parodyta kiek iš viso buvo sukurta rezervacijų bei rezervacijos laikas atsiras laiko juostoje.

The screenshot displays a web application for managing meetings. At the top, there is a navigation bar with links for ROOMS, LEISURE, BOOKINGS, ASSETS, and USERS. The main interface consists of a large time slot grid on the left and a sidebar on the right. The grid shows time slots from 13:30 to 18:45 in 15-minute increments. A booking is visible for the 16:30-17:00 slot, with the text 'booking' and 'Admin Admin' displayed. The sidebar on the right contains a 'DATE / TIME' section with a 'Reset' button and a 'Meeting name' field with the value 'booking'. Below this, there are buttons for 'Update', 'Delete', and 'Deselect'.

**Pav. 2.24** 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 dark header bar with three icons: a calendar, a refresh/circular arrow, and a group of people. Below the header, the main content area has a dark background. It features a section titled 'DATE / TIME' in white, with a 'Reset' link in teal to its right. The form includes three date and time pickers, each with teal arrows for navigation. The first picker is for the 'Meeting date' (2020-05-07), the second for 'Meeting starts at' (15:15), and the third for 'Meeting ends at' (15:45). Below these is a text input field for the 'Meeting name'. At the bottom of the form 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. 2.25** 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. 2.26** Rezervacijos pasikartojimo opcijos

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



**Pav. 2.27** Susitikimo narių pridėjimas

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

ROOMS

LEISURE

BOOKINGS

ASSETS

USERS

admin Administrator

● Busy

● Free for less than 30 min

● Free

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

FILTER

Reset

Meeting date: < 2020-05-07 >

Meeting starts at: < 14:00 >

**Pav. 2.28** Laisvalaikio įrangos langas

Šiame lange varototjas 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. 2.29** 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.

ROOMS LEISURE BOOKINGS ASSETS USERS			admin Administrator
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	

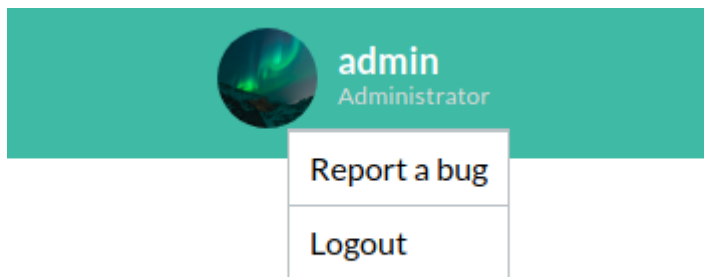
**Pav. 2.30** Bendras susitikimo kabarių ir laisvalaikio įrangos sąrašas

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

ROOMS LEISURE BOOKINGS ASSETS USERS			admin Administrator
NAME	MAIL	ROLE	
admin	admin@localhost	Administrator	
qwe	qwe@localhost	User	
devuser3	devuser3@devuser.local	Asset	

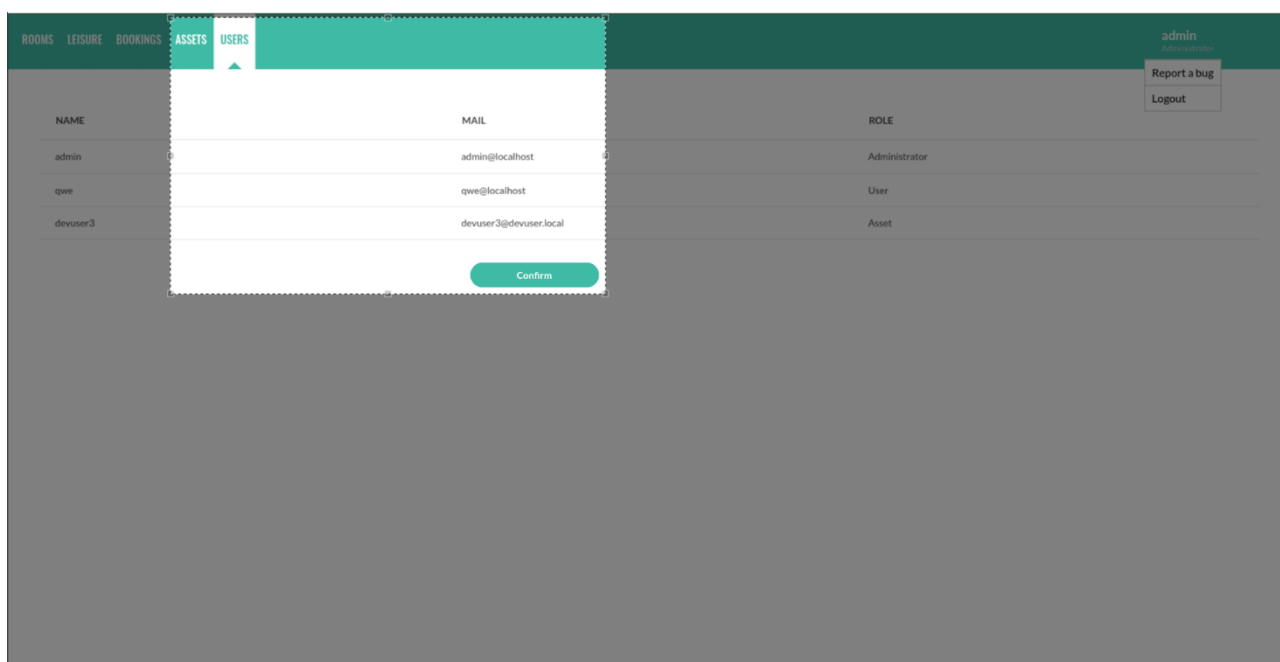
**Pav. 2.31** Vartotojų sąrašas

Šiame lange administratorius gali matyti visų vartotojų sąrašą.



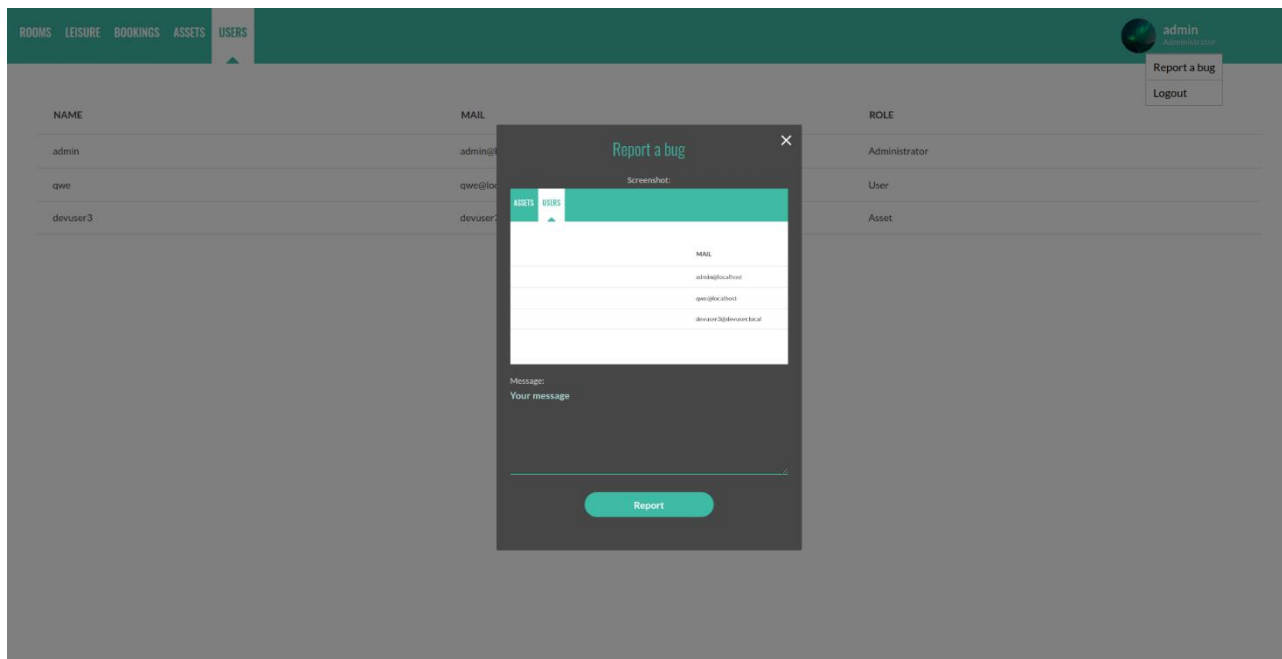
**Pav. 2.32** Vartotojo meniu

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



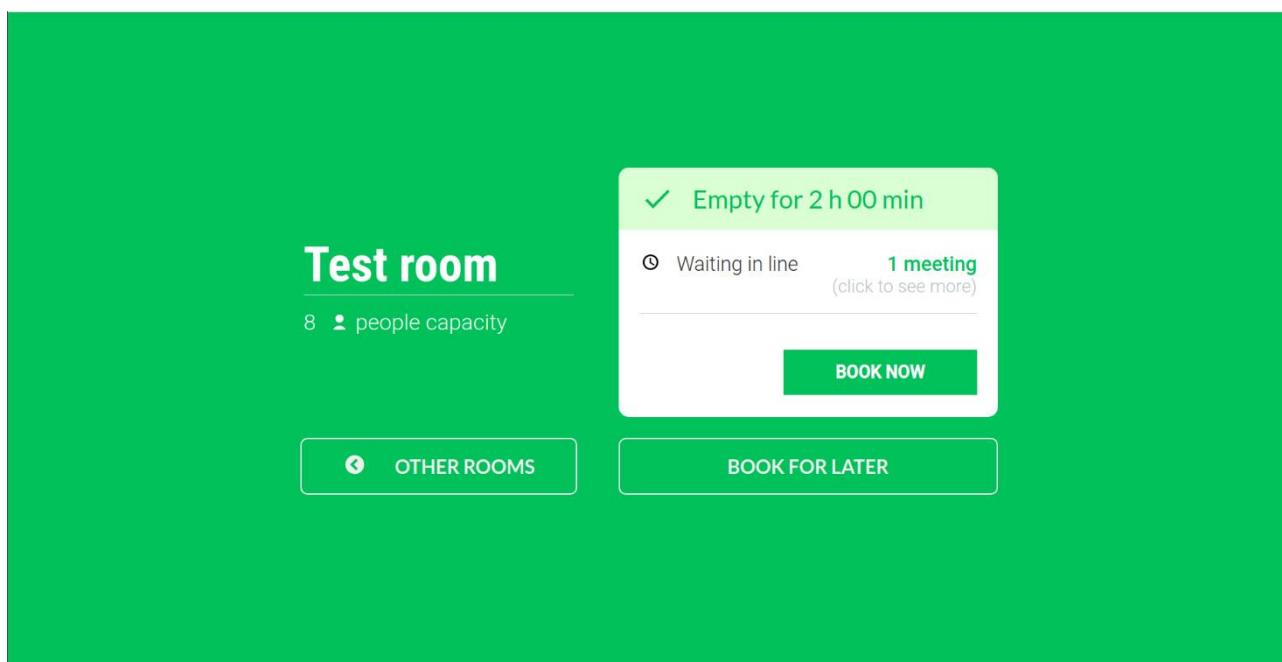
**Pav. 2.33** Sistemos klaidos pranešimo lango fiksavimas

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



**Pav. 2.34** Sistemos klaidos pranešimo sukūrimas

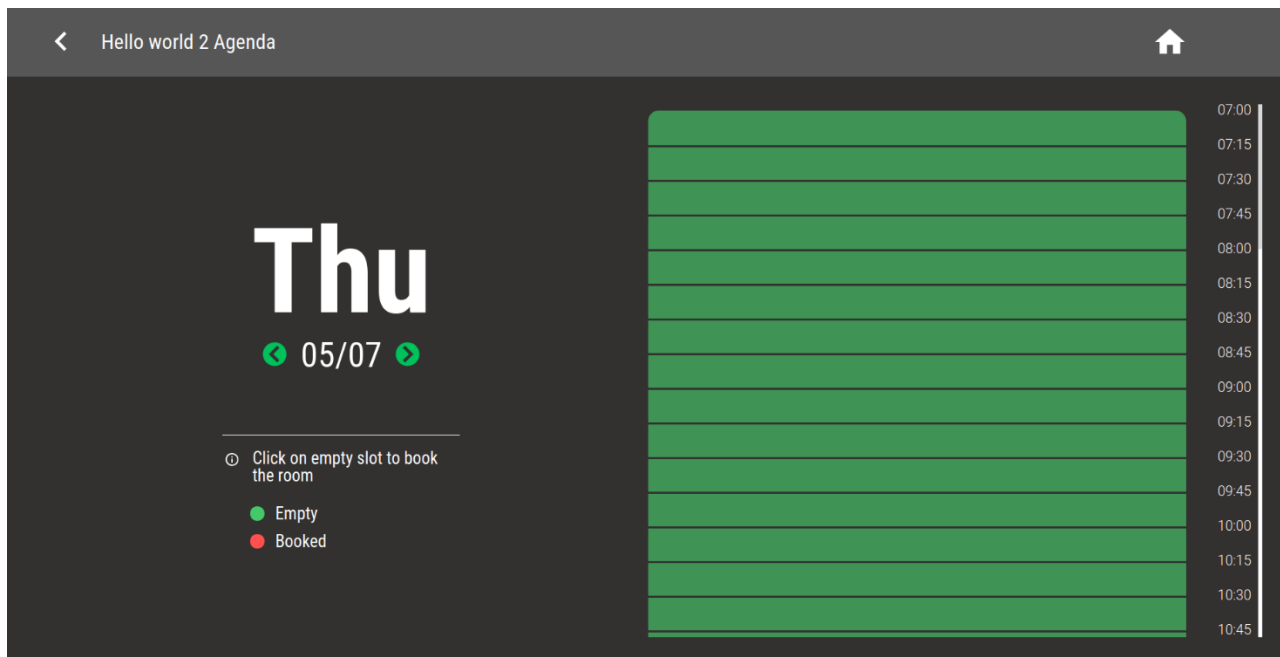
Šiame lange vartotojas mato savo ekrano vietos užfiksavimą. Įvedus lauke „Your message“ klaidos aprašimą ir paspaudus mygtuką „Report“ klaidos pranešimus bus įrašytas.



**Pav. 2.35** Planšetės susitikimo kambario informacija

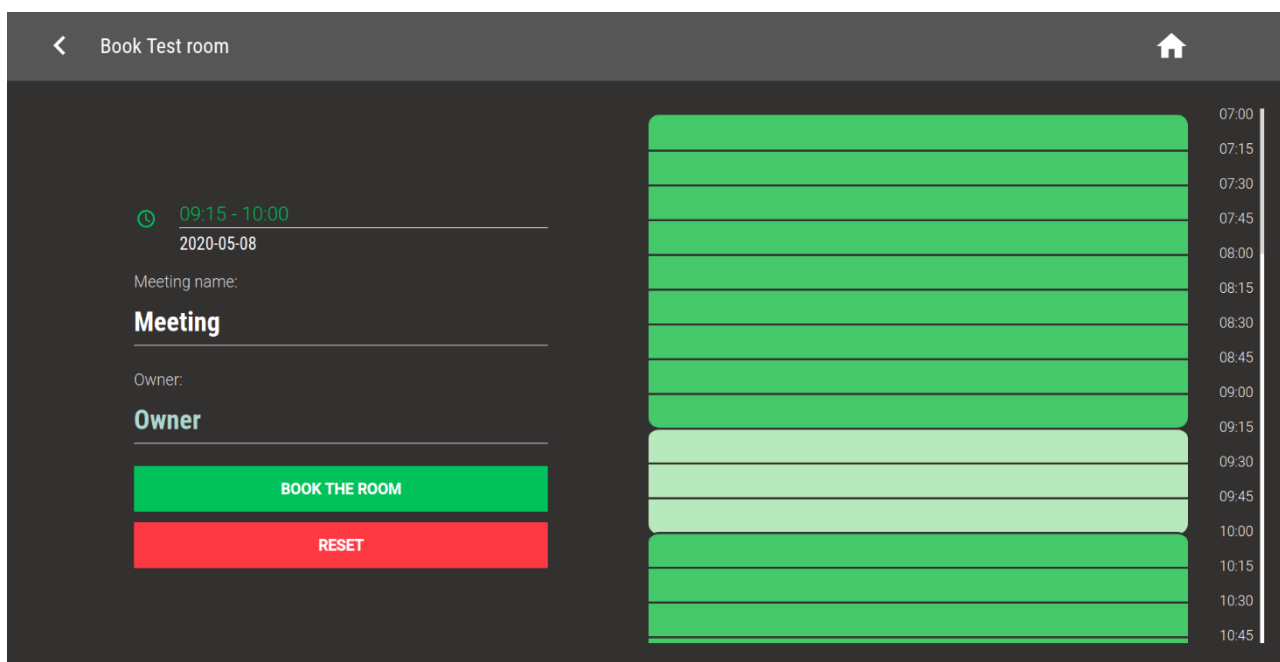
Šiame planšetės lange vartotojas gali matyti informaciją apie susitikimo kambarį ir apie jo rezervacijas. Paspaudus mygtukas „OTHER ROOMS“ vartotojas bus nukreiptas į susitikimo kambarių sąrašą. Paspaudus mygtukus „BOOK NOW“ arba „BOOK FOR LATER“ vartotojas bus nukreiptas į rezervacijos kūrimo langą.





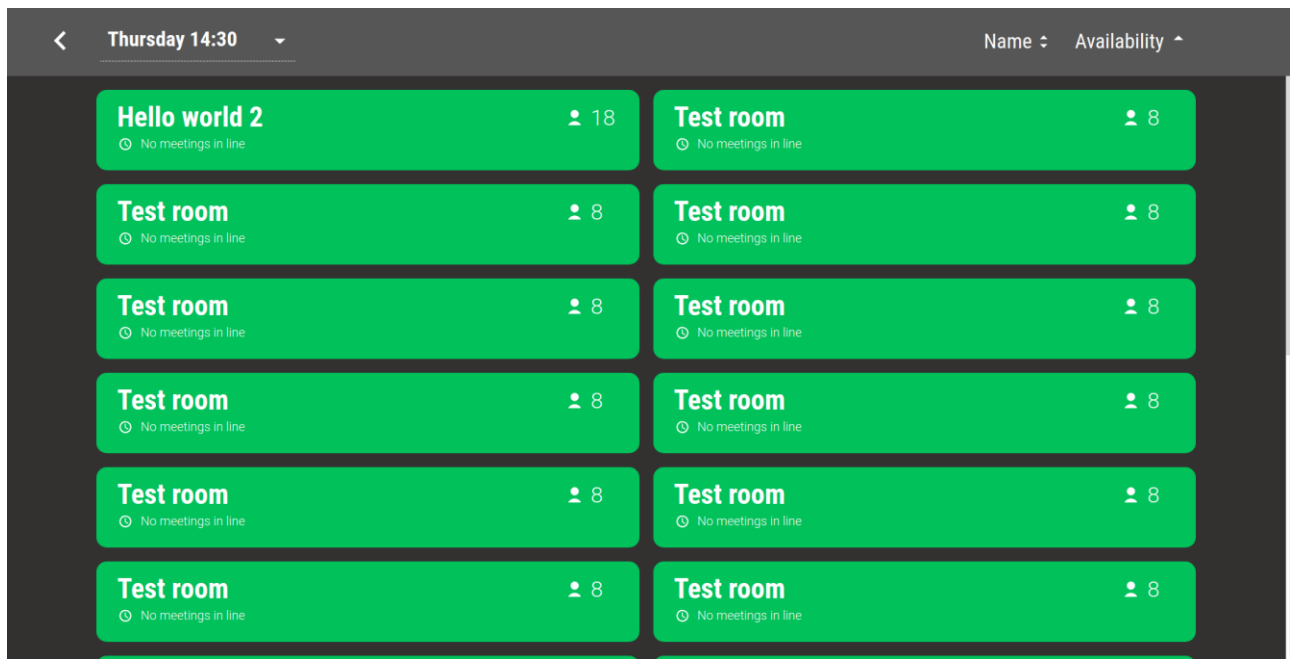
**Pav. 2.36** Rezervacijos kūrimas planšetės lange

Šiame planšetės lange vartotojas gali kurti rezervacijas.



**Pav. 2.37** 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. 2.38** 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](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>.