|  |  |  |
| --- | --- | --- |
| VILNIAUS KOLEGIJA  ELEKTRONIKOS IR INFORMATIKOS FAKULTETAS  PROGRAMINĖS ĮRANGOS KATEDRA | | |
|  | | |
| "SHIELD PROTECTION" PROJEKTO APRAŠAS | | |
| HIBRIDINĖS MOBILIOSIOS APLIKACIJOS  KD - 6531BX028 PI17B | | |
| STUDENTAI | (PARAŠAS)  2020-01-02 | MINDAUGAS BURVYS |
| DĖSTYTOJA | (PARAŠAS)  2020-01-02 | J. ZAILSKAS |
|  |  |  |
|  |  |  |
| 2018 | | |

# Turinys

[Turinys 2](#_Toc28924910)

[Paveikslėlių rodyklė 2](#_Toc28924911)

[Įvadas 3](#_Toc28924912)

[1. Užduoties formulavimas 4](#_Toc28924913)

[Funkciniai reikalavimai 4](#_Toc28924914)

[Nefunkciniai reikalavimai 4](#_Toc28924915)

[2. Užduoties analizė 5](#_Toc28924916)

[2.1. Duomenų bazė 5](#_Toc28924917)

[2.2. Duomenų bazės ER diagrama 5](#_Toc28924918)

[2.3. Sistemos panaudos atvejų diagrama 6](#_Toc28924919)

[2.4. Panaudos atvejai 7](#_Toc28924920)

[3. Programinės realizacijos aprašymas 11](#_Toc28924921)

[4. Naudotojo instrukcija 18](#_Toc28924922)

[Išvados ir pasiūlymai 21](#_Toc28924923)

[Informacijos šaltiniai 22](#_Toc28924924)

# Paveikslėlių rodyklė

[1 pav. duomenų bazės ER diagrama 5](#_Toc28923793)

[2 pav. panaudos atvejų diagrama 6](#_Toc28923794)

[3 pav. prisijungimo langas 18](#_Toc28923795)

[4 pav. registracijos langas 18](#_Toc28923796)

[5 pav. profilio langas 19](#_Toc28923797)

[6 pav. saugotinų objektų peržiūros langas 20](#_Toc28923798)

[7 pav. naujo saugotino objekto pridėjimo langas 20](#_Toc28923799)

# Įvadas

Šio projekto yra skirtas pademonstruoti su React Native framework‘u įgytas per semestrą žinias ir sukurti projektą, savo nuožiūra pasirinkta tema, kuris atliktų konkretų darbą, pasinaudojant mūsų įgytomis žiniomis.

**Užduoties tikslas**:

1. Sukurti sistemą, kuri valdytų atitinkamus duomenis saugotinoms ir saugomoms vietoms bei objektams
2. Sistema turi turėti vartotojo autentifikavimą ir vartotojų (sargų) registraciją.
3. Sistemoje yra panaudoti įvairaus tipo HTML komponentai ar API‘s
4. Sistema turi naudotis SQL ar NoSQL duomenų baze
5. Sistemos programinis kodas įkeltas į GitHub sistemą, su atitinkamai paruoštais šios sistemos failais
6. Sistemai parašyta dokumentacija.

**Projekto tematika:**

Projekto tema yra sistema, valdanti saugomus objektus, bei informuojanti apie atitinkamų objektų saugojimo statusą. Taip pat ši sistema atvaizduoja sąrašą saugomų objektų su jų informacija (adresais, pavadinimais ir pan.), vartotojų (sargų) sąrašą, leidžia registruoti naujus sargus, bei objektus, taip pat valdyti sargų informaciją, pamatyti prisijungimo ir registravimosi laikus ir pasimėgauti katino nuotrauka.

# Užduoties formulavimas

## Funkciniai reikalavimai

|  |  |  |
| --- | --- | --- |
| **Numeris** | **Funkcinis reikalavimas** | **Paaiškinimas** |
| 1 | Sistema turi turėti autentifikavimo sistemą | Sistema turi naudotojų prisijungimo ir registracijos sistemas. |
| 2 | Sistema atlieka duomenų perėmimo ir saugojimo funkcijas iš ir į pasirinktą duomenų bazėję | Sistema turi funkcijas gauti ir siųsti duomenis iš ir į duomenų bazę. Šioje sistemoje šiuos duomenis sudaro saugotini objektai, saugomi objektai, vartotojai, vartotojų prisijungimo duomenys, vartotojų prisijungimo laiko registras. |
| 3 | Sistema turi registruoti naudotojų prisijungimo laiką ir leisti šiuos laikus peržiūrėti | Sistema duomenų bazėje saugo kada ir kuris naudotojas buvo prisijungęs, bei registruotiems naudotojams leidžia peržiūrėti šia informaciją |
| 4 | Sistema turi saugoti, bei leisti peržiūrėti saugotinus objektus ir jau saugomus objektus | Sistema leidžia vartotojui pamatyti saugotinus (darbas dar nepradėtas) bei saugomus (saugos darbas atliekamas arba atliktas) objektus, taip pat pridėti naujus saugotinus objektus. |

## Nefunkciniai reikalavimai

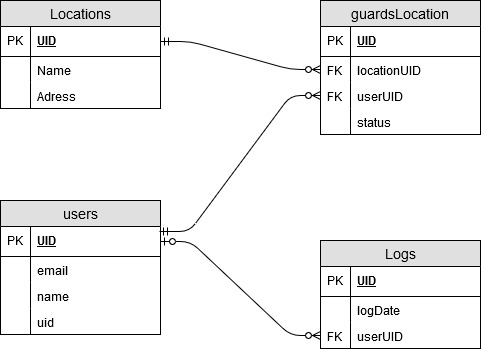
1. Sistema turi naudoti React Native framework’ą.
2. Sistema galėtų būti naudojama, kaip android, iphone ar web aplikacija,
3. Sistema atvaizduoja informaciją pasinaudojant framework’o komponentais ir jutikliais (angl. API’s).
4. Sistema naudoja Firebase duomenų bazę duomenų saugojimui ar vartotojų autentifikavimui.
5. Sistema turi būti funkciškai dali ir sistemos dalys (klasės ir metodai) turi atlikti tik jiems skirtą darbą (angl. single responsibility principle).
6. Sistema atvaizduoja informaciją aiškiais ir nuspėjamais vartotojui veiksmais.

# Užduoties analizė

## Duomenų bazė

Firebase yra mobilių ir web aplikacijų kūrimo aplinka. Šios aplinkos dėka yra lengva susikurti NoSQL duomenų bazę, autentifikavimo sistemą ar kitus dažnai sistemos darbui reikalingus objektus. Šiam projektui būtent buvo naudojama Firebase duomenų bazė. Nors jai nėra būtina kurti ryšių tarp lentelių ar naudoti specialių duomenų bazės iššaukimo sakinių (pavyzdžiui SQL duomenų bazės SELECT sakinys), taip pat kaip ir kurti raktinių laukų, šiam projektui buvo bandoma suprojektuoti duomenų bazę, kuri bent primintų SQL pagrindo duomenų bazes, lengvesniam duomenų bazės įsisavinimui ir valdymui.

## Duomenų bazės ER diagrama



1 pav. duomenų bazės ER diagrama

Šią duomenų bazę sudaro 4 lentelės (žr. 1 pav.). Kiekviena turi atitinkamus raktinius (PK, FK) laukus, kurie padeda palaikyti ryšį tarp šių duomenų.

* **Lentelė Locations**

Ši lentelė skirta saugoti atitinkamų saugotinų objektų ar vietų informaciją. Joje saugomi atitinkamų objektų pavadinimai ir adresai, kurie priskirti unikaliai ID (UID) reikšmei.

Ši lentelė yra naudojama nei vienos ar daugiau saugomų vietų (guardsLocation) lentelės duomenų.

* **Lentelė guardsLocation**

Ši lentelė skirta sieti saugomų objektų ar vietų, taip pat ir tos vietos sargų duomenis. Joje saugomas saugomos vietos UID (locationUID), atitinkamo sargo UID (userUID), bei saugomos vietos darbo statusas (status), kurie yra priskirti unikaliai ID (UID) reikšmei.

Ši lentelė įrašams naudoja vieną sargą ir vieną lokaciją. Dėl to norint sužinoti visus atitinkamos vietos ar objekto sargus reikia nuskaityti keletą šios lentelės įrašų.

* **Lentelė users**

Ši lentelė skirta saugoti visų sargų (sistemos registruotų vartotojų) profilio duomenis. Joje saugomas vartotojo el. paštas (email), vartotojo vardas (name) ir vartotojo ID (uid).

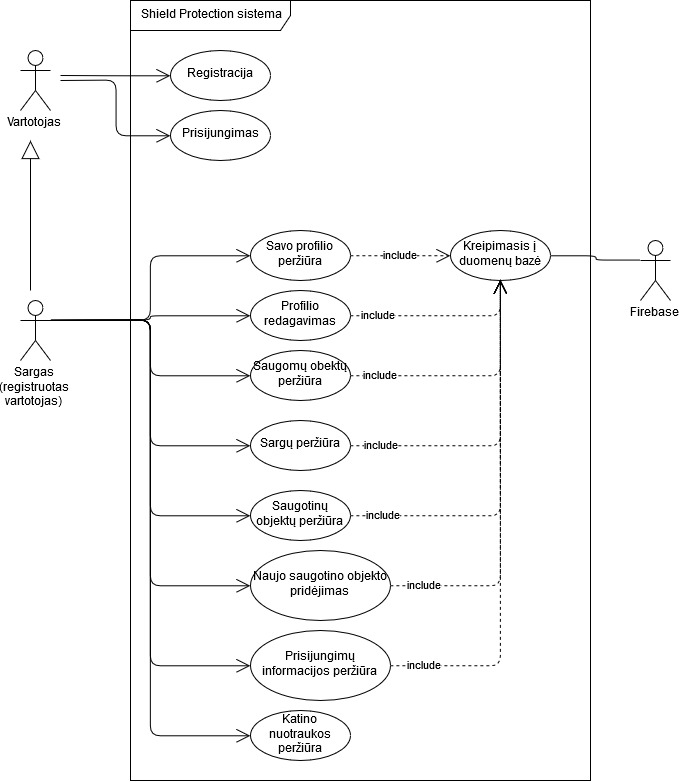
Šios lentelės įrašai yra naudojami guardsLocations lentelės įrašams, taip pat ir prisijungimo laiko įrašams (Logs).

* **Lentelė Logs**

Ši lentelė skirta saugoti prisijungimo laiko įrašus (kada atitinkamas vartotojas buvo prisijungęs). Joje saugomas atitinkamo vartotojo UID (userUID) ir prisijungimo laiko žymės (logDate), kurie priskirti unikaliai ID (UID) reikšmei.

Šioje lentelė yra naudojami sargų UID (userUID), kad priskirti atitinkamą tuo metu prisijungusį sargą.

## Sistemos panaudos atvejų diagrama



2 pav. panaudos atvejų diagrama

## Panaudos atvejai

|  |  |
| --- | --- |
| **Pavadinimas** | Registracija |
| **ID** | UC01 |
| **Trumpas aprašymas** | Sistemos naudotojas gali užsiregistruoti sistemoje |
| **Naudotojas (-ai)** | Neregistruotas naudotojas |
| **Pagrindinis scenarijus** | Naudotojas suveda atitinkamą el. pašto adresą ir slaptažodį, ir paspaudžia registracijos mygtuką. |
| **Alternatyvūs scenarijai** | * Suvedamas netinkamos formos el. pašto adresas, vartotojas informuojamas apie netinkamos formos el. pašto adresą. * Suvedamas jau egzistuojantis el. pašto adresas; vartotojas informuojamas apie jau egzistuojantį vartotoją šiuo el. pašto adresu. |

|  |  |
| --- | --- |
| **Pavadinimas** | Prisijungimas |
| **ID** | UC02 |
| **Trumpas aprašymas** | Sistemos naudotojas gali prisijungti prie sistemos, naudodamasis savo el. paštu ir slaptažodžiu |
| **Naudotojas (-ai)** | Registruotas naudotojas (sargas) |
| **Pagrindinis scenarijus** | Naudotojas suveda atitinkamą el. pašto adresą ir slaptažodį, ir paspaudžia prisijungimo mygtuką. |
| **Alternatyvūs scenarijai** | * Suvedamas netinkamos formos el. pašto adresas, vartotojas informuojamas apie netinkamos formos el. pašto adresą. * Suvedamas neegzistuojančio naudotojo el. paštas ir slaptažodis; vartotojas informuojamas apie tokio vartotojo neegzistavimą. |

|  |  |
| --- | --- |
| **Pavadinimas** | Savo profilio peržiūra |
| **ID** | UC03 |
| **Trumpas aprašymas** | Sistemos naudotojas mato savo profilio informaciją. |
| **Naudotojas (-ai)** | Registruotas naudotojas (sargas) |
| **Pagrindinis scenarijus** | Naudotojui, tik prisijungus prie sistemos atvaizduojami pagrindiniai jo duomenys. |

|  |  |
| --- | --- |
| **Pavadinimas** | Profilio redagavimas |
| **ID** | UC04 |
| **Trumpas aprašymas** | Naudotojas gali redaguoti ir keisti atitinkamus duomenis apie save. |
| **Naudotojas (-ai)** | Registruotas naudotojas (sargas) |
| **Pagrindinis scenarijus** | Sistemos naudotojas pasirenka profilio redagavimo funkciją (paspaudžia redaguoti profilį mygtuką); suveda naują vardą ir paspaudžia išsaugoti mygtuką. |

|  |  |
| --- | --- |
| **Pavadinimas** | Saugomų objektų peržiūra |
| **ID** | UC05 |
| **Trumpas aprašymas** | Naudotojas gali peržiūrėti saugomų objektų duomenis ir sargus dirbančius su šiais objektais. |
| **Naudotojas (-ai)** | Registruotas naudotojas (sargas) |
| **Pagrindinis scenarijus** | Naudotojas paspaudžia objektų peržiūrėjimo mygtuką (protected locations); naudotojo ekrane parodomas visų saugomų objektų sąrašas. |

|  |  |
| --- | --- |
| **Pavadinimas** | Sargų (registruotų vartotojų) peržiūra |
| **ID** | UC06 |
| **Trumpas aprašymas** | Naudotojas gali peržiūrėti visų dirbančių sargų (registruotų vartotojų) pagrindinius duomenis |
| **Naudotojas (-ai)** | Registruotas naudotojas (sargas) |
| **Pagrindinis scenarijus** | Naudotojas paspaudžia ant sargų duomenų peržiūrėjimo mygtuko (guards); naudotojo ekrane atvaizduojama sargų informacija |

|  |  |
| --- | --- |
| **Pavadinimas** | Saugotinų objektų peržiūra |
| **ID** | UC07 |
| **Trumpas aprašymas** | Naudotojas gali peržiūrėti visų saugotinų objektų pavadinimus ir informaciją apie juos. |
| **Naudotojas (-ai)** | Registruotas naudotojas (sargas) |
| **Pagrindinis scenarijus** | Naudotojas paspaudžia ant saugotinų objektų mygtuko (Locations); naudotojo ekrane atvaizduojama sargų informacija |

|  |  |
| --- | --- |
| **Pavadinimas** | Naujo saugotino objekto pridėjimas |
| **ID** | UC08 |
| **Trumpas aprašymas** | Naudotojas gali pridėti naujo saugotino objekto informaciją, prie saugotinų objektų |
| **Naudotojas (-ai)** | Registruotas naudotojas (sargas) |
| **Pagrindinis scenarijus** | Naudotojas paspaudžia ant pridėti naują lokaciją mygtuko, atidarytame lange suveda naujo objekto pavadinimą ir adresą ir paspaudžia ant pridėti objektą mygtuko. |

|  |  |
| --- | --- |
| **Pavadinimas** | Prisijungimų informacijos peržiūra |
| **ID** | UC09 |
| **Trumpas aprašymas** | Naudotojas gali peržiūrėti sargų prisijungimo laikus |
| **Naudotojas (-ai)** | Registruotas naudotojas (sargas) |
| **Pagrindinis scenarijus** | Naudotojas paspaudžia ant naudotojų prisijungimų informacijos mygtuko (Logs); naudotojo ekrane atvaizduojami prisijungusių naudotojų vardai ir laikai, kada jie prisijungė |

|  |  |
| --- | --- |
| **Pavadinimas** | Katino nuotraukos peržiūra |
| **ID** | UC10 |
| **Trumpas aprašymas** | Naudotojas gali pamatyti ypatingai svarbią katino nuotrauką. |
| **Naudotojas (-ai)** | Registruotas naudotojas (sargas) |
| **Pagrindinis scenarijus** | Naudotojas paspaudžia ant peržiūrėti katiną mygtuko (Cat...); naudotojo ekrane atvaizduojamas katinas |

|  |  |
| --- | --- |
| **Pavadinimas** | Kreipimasis į duomenų bazę |
| **ID** | UC11 |
| **Trumpas aprašymas** | Sistema, kreipiasi į duomenų bazę (šiame projekte Firebase), kad atliktų darbus su atitinkamais duomenis iš jos |
| **Naudotojas (-ai)** | Sistema |
| **Pagrindinis scenarijus** | Vartotojas atlieka veiksmą, reikalaujanti duomenų paėmimo ar įrašymo į duomenų bazę; sistema su atitinkama užklausa gauna duomenis ar atlieka su jais atitinkamą užduotį |

# Programinės realizacijos aprašymas

**Sistemos programinės realizacijos failų sąrašas:**

* actions
  + user.js
* config
  + Firebase.js
* images
  + ImagesExample.js
  + logo.png
  + LogoView.js
  + Zuu\_the\_sleepy\_cat.jpg
* navigation
  + SwitchNavigator.js
* reducers
  + index.js
* screens
  + EditProfile.js
  + GuardInfo.js
  + Locations.js
  + LocationsAdd.js
  + Login.js
  + Logs.js
  + Profile.js
  + ProtectedLocations.js
  + Signup.js
  + SleepyCat.js
* .env
* App.js

**Failų sandara ir veikimas:**

* **user.js**

Šis failas turi funkcijas ir veiksmų konstantas, skirtas prisijungti ir užregistruoti vartotojus. Šiam darbui atlikti yra naudojamas importuoto Firebase.js failo konstantos skirtos susisiekti su duomenų baze.

* **Failo sandara:**
  + updateEmail(email) funkcija
  + updatePassword(password) funkcija
  + login() funkcija
  + getUser(uid) funkcija
  + signup() funkcija
  + UPDATE\_EMAIL konstanta
  + UPDATE\_PASSWORD konstanta
  + LOGIN konstanta
  + SIGNUP konstanta
* updateEmail(email) funkcija atnaujina vartotojo objekto el. pašto adresą, kuris yra gautas iš duomenų bazės.
* updatePassword(password) funkcija atnaujina vartotojo objekto slaptažodį, kuris yra gautas iš duomenų bazės.
* login() funkcija gauna lokalaus vartotojo objekto duomenis (el. paštą ir slaptažodį), jais bando prisijunti, kreipiantis į Firebase autentifikavimo sistemą, sukuria prisijungimo statuso pranešimą vartotojui (ar prisijungimas sėkmingas ar ne) ir registruoja prisijungimo laiką duomenų bazėje.
* getUser(uid) funkcija kreipiasi į Firebase duomenų bazę ir pagal turimą vartotojo UID bando gauti atitinkamo vartotojo duomenis iš Firebase duomenų bazės.
* signup() funkcija gauna lokalaus vartotojo objekto duomenis (el. paštą ir slaptažodį), jais bando registruotis, kreipiantis į Firebase autentifikavimo sistemą ir jei šis veiksmas buvo sėkmingas, išsaugo vartotojo profilio duomenis duomenų bazėje, ir registruoja prisijungimo laiką duomenų bazėje.
* **Firebase.js**

Šis failas skirtas susieti sistemą su Firebase duomenų baze, tam naudojant Firebase ir „firebase/firestore“ modulius, bei atitinkamus duomenis, išsaugotus projekto aplinkos faile (.env), būtinus susieti Firebase su atitinkama duomenų baze.

* **Failo sandara:**
  + firebaseConfig konstanta, išsauganti būtinus prisijungimui duomenis, kaip objektą.
  + db konstanta, kuri skirt sutrumpini kodo užklausų, kurios kreipsis į duomenų bazę rašymui.
* **logo.png ir Zuu\_the\_sleepy\_Cat.png**

Tai paveikslėlių failai, kurie yra naudojami sistemoje.

* **ImagesExample.js**

Tai failas turintis metodą, kuris atvaizduoja Zuu\_the\_sleepy\_Cat.png paveikslėlį, su atitinkamais parametrais.

* **Failo sandara:**
  + ImagesExample() funkcija, atvaizduojanti HTML View elementą, kuris turi HTML Image elementą, su atitinkamu paveikslėliu ir jo stiliumi (dydis ir pan.).
* **LogoView.js**

Tai failas turintis metodą, kuris atvaizduoja logo.png paveikslėlį, su atitinkamais parametrais.

* **Failo sandara:**
  + LogoView() funkcija, atvaizduojanti HTML View elementą, kuris turi HTML Image elementą, su atitinkamu paveikslėliu ir jo stiliumi (dydis ir pan.).
* **SwitchNavigator.js**

Tai failas skirtas sukurti SwitchNavigator konstantą, skirta greičiau ir paprasčiau valdyti perėjimą tarp atskirų sistemos langų. Tam yra naudojamas react-navigation modulis, kuris leidžia sukurti šį objektą, kuris turi sąrašą langų, į kuriuos galima pereiti, taip pat ir saugodamas pagrindinį (pirmą šio navigatoriaus atidaromą) langą.

* **Failo sandara:**
  + SwitchNavigator konstanta.
* **index.js**

Tai failas skirtas sukurti user ir rootReducer konstantas. Šios konstantos yra naudojamos, kaip tarpinis elementas, tarp duomenų bazės ir sistemos, apibrėžiant tik tam tikrus veiksmus, kuriuos sistema gali atlikti, pasinaudojant redux moduliu.

* **Failo sandara:**
  + user konstanta, apibrėžianti prisijungimo, registracijos, el. pašto keitimo ir slaptažodžio keitimo veiksmus.
  + rootReducer konstanta, susiejanti visus joje paminėtų veiksmų konstantų objektus.
* **EditProfile.js**

Failas skirtas profilio redagavimo klasei ir jos atvaizdavimui (render).

* **Failo sandara:**
  + EditProfile klasė
    - handleBack() metodas skirtas vykdyti, grįžimo atgal į ankstesnį langą, funkciją.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis vaizdą.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **GuardInfo.js**

Failas skirtas sargų (registruotų vartotojų) informacijos atvaizdavimui (render).

* **Failo sandara:**
  + GuardInfo klasė
    - users state’as, skirtas saugoti iš duomenų bazės gautų duomenų objektus (vartotojų vardus ir el. paštus)
    - handleBack() metodas skirtas vykdyti, grįžimo atgal į ankstesnį langą, funkciją.
    - renderItem(obj) metodas skirtas atvaizduoti vieno objekto (išsaugoto state’e vartotojo vieneto) duomenis.
    - componentDidMount() metodas pradedamas vykdyti prieš atliekant kitus klasės veiksmus, skirtas gauti informacijai, kuri bus atvaizduota render() metode, iš duomenų bazės.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis vaizdą.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **Locations.js**

Failas skirtas saugotinų objektų (lokacijų) informacijos atvaizdavimui (render).

* **Failo sandara:**
  + Locations klasė
    - locations state’as, skirtas saugoti iš duomenų bazės gautų duomenų objektus (saugotino objekto ID, pavadinimą ir adresą)
    - handleBack() metodas skirtas vykdyti, grįžimo atgal į ankstesnį langą, funkciją.
    - handleAdd() metodas skirtas vykdyti, perėjimo į lokacijos pridėjimo lango, funkciją.
    - renderItem(obj) metodas skirtas atvaizduoti vieno objekto (išsaugoto state’e lokacijos vieneto) duomenis.
    - componentDidMount() metodas pradedamas vykdyti prieš atliekant kitus klasės veiksmus, skirtas gauti informacijai, kuri bus atvaizduota render() metode, iš duomenų bazės.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis vaizdą.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **LocationsAdd.js**

Failas skirtas papildomų saugotinų objektų (lokacijų) informacijos pridėjimui į duomenų bazę.

* **Failo sandara:**
  + LocationsAdd klasė
    - locName state’as, skirtas saugoti saugotino objekto pavadinimą.
    - locAdress state‘as skirtas saugoti saugotino objekto adresą.
    - handleBack() metodas skirtas vykdyti, grįžimo atgal į ankstesnį langą, funkciją.
    - handleAdd() metodas skirtas vykdyti lokacijos pridėjimo funkciją ir grįžimą į lokacijų langą.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis vaizdą.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **Login.js**

Failas skirtas sargų (registruotų vartotojų) informacijos atvaizdavimui (render). Ši klasė naudojasi Redux tarpiniu elementu, kad perimtų vartotojo duomenis iš duomenų bazės.

* **Failo sandara:**
  + Login klasė
    - handleLogin() metodas skirtas vykdyti, vartotojo prisijungimo funkciją, kviečiant user.js faile aprašytą veiksmą.
    - componentDidMount() metodas pradedamas vykdyti prieš atliekant kitus klasės veiksmus, skirtas gauti vartotojo informacijai iš duomenų bazės, jeigu yra prisijungęs vartotojas ar gautas teisingas autentifikavimo pranešimo pasikeitimas.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **Logs.js**

Failas skirtas vartotojų prisijungimų laiko registravimui ir atvaizdavimui (render).

* **Failo sandara:**
  + Logs klasė
    - logs state’as, skirtas saugoti iš duomenų bazės gautų duomenų objektus (vartotojų prisijungimų laikus)
    - handleBack() metodas skirtas vykdyti, grįžimo atgal į ankstesnį langą, funkciją.
    - renderItem(obj) metodas skirtas atvaizduoti vieno objekto (išsaugoto state’e log‘o vieneto) duomenis.
    - componentDidMount() metodas pradedamas vykdyti prieš atliekant kitus klasės veiksmus, skirtas gauti informacijai, kuri bus atvaizduota render() metode, iš duomenų bazės.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **Profile.js**

Failas skirtas sargo (registruoto vartotojo) informacijos atvaizdavimui (render).

* **Failo sandara:**
  + Profile klasė
    - handleSignout() metodas skirtas vykdyti atsijungimo ir grįžimo į prisijungimo langą funkcijai.
    - handleEdit() metodas skirtas pereiti į profilio redagavimo puslapį.
    - handleLocations() metodas skirtas pereiti į lokacijų atvaizdavimo puslapį.
    - handleCat() ypatingai būtinas metodas skirtas pereiti į SleepyCat atvaizdavimo puslapį.
    - handleProtected() metodas skirtas pereiti į saugomų objektų informacijos pusalapį
    - handleGuards() metodas skirtas pereiti į sargų informacijos atvaizdavimo puslapį.
    - handleLogs() metodas skirtas pereiti į vartotojų prisijungimų laikų atvaizdavimo puslapį.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **ProtectedLocations.js**

Failas skirtas saugomų objektų informacijos atvaizdavimui (render).

* **Failo sandara:**
  + ProtectedLocations klasė
    - GuardsLocation state’as, skirtas saugoti iš duomenų bazės gautų duomenų objektus saugomų objektų vietos ID, vartotojo ID, objekto saugojimo statusą, lokacijos pavadinimą ir lokacijos adresą).
    - handleBack() metodas skirtas vykdyti, grįžimo atgal į ankstesnį langą, funkciją.
    - renderItem(obj) metodas skirtas atvaizduoti vieno objekto (išsaugoto state’e saugomo objketo vieneto) duomenis.
    - componentDidMount() metodas pradedamas vykdyti prieš atliekant kitus klasės veiksmus, skirtas gauti informacijai, kuri bus atvaizduota render() metode, iš duomenų bazės.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **Signup.js**

Failas skirtas vartotojo registracijai ir registracijos vaizdo atvaizdavimui.

* **Failo sandara:**
  + Signup klasė
    - handleSignUp() metodas skirtas vykdyti registracijos funkciją, naudojantis user.js faile aprašytais veiksmais.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **SleepyCat.js**

Ypatingai būtinas failas skirtas atvaizdavimui katino paveikslėlį.

* **Failo sandara:**
  + SleepyCat klasė
    - handleBack() metodas skirtas vykdyti, grįžimo atgal į ankstesnį langą, funkciją.
    - render() metodas atvaizduojantis jame aprašyto HTML žymėjimo kalbos su įterptomis Java script kodo detalėmis.
  + styles kosntanta, skirta išsaugoti stiliaus (StyleSheer) objekto duomenis, HTML žymėjimo kalbos kodui sutrumpinti.
* **.env**

Failas skirtas išsaugoti Firebase prisijungimui būtinus duomenis.

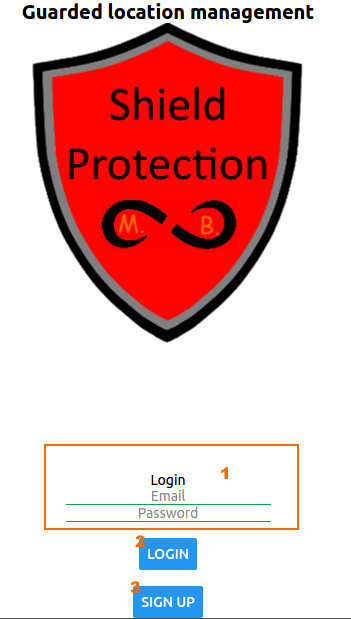
* **Failo sandara:**
  + Tekstinio tipo duomenys, nusakantys Firebase prisijungimo parametrus.
* **App.js**

Pradinis aplikacijos failas, kuriuo paleidžiama programa. Jame tik susiejamas redux tarpininkas su programa ir paleidžiamas SwitchNavigator navigatorius.

* **Failo sandara:**
  + App klasė, kuri yra pradinė (main) aplikacijos klasė
    - render() metodas skirtas atvaizduoti pradiniam vaizdui, kuris plėtojamas tik kitose klasėse.

# Naudotojo instrukcija

* **Prisijungimas**

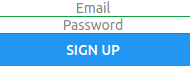


3 pav. prisijungimo langas

Sistemos darbas pradedamas nuo prisijungimo lango. Šiame lange turite prisijungimo informacijos lauką (žr. 3 pav. 1 punkto dėžutė), prisijungimo mygtukas ir registracijos mygtukas. Norint prisijungti reikia užpildyti prisijungimo duomenis (tai yra el. paštą ir slaptažodį) ir numeriu paspausti prisijungimo mygtuką. Įvykdžius teisingą prisijungimą bus matomas profilio langas (žr. 5 pav.).

Teisingam prisijungimui reikia naudoti atitinkamos formos el. pašto adresą ([tekstas@tekstas.tekstas](mailto:tekstas@tekstas.tekstas)), kitu atveju naudotojas bus informuojamas apie neteisingą el. pašto adreso formą.

* **Registracija**

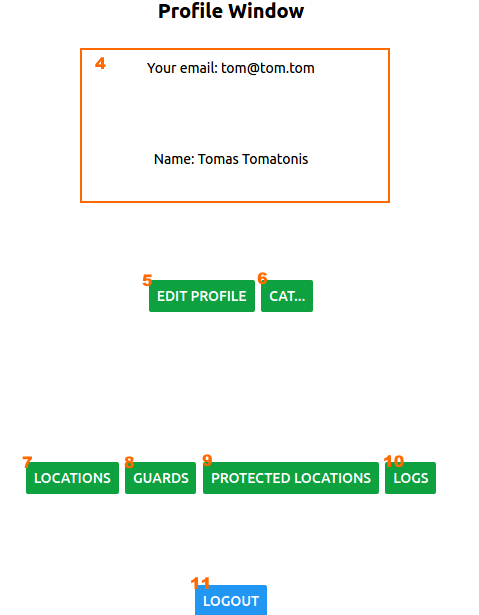
****

4 pav. registracijos langas

Norint užsiregistruoti reikia paspausti prisijungimo lange esantį registracijos mygtuką (žr. 3 pav.) ir naujai atsidariusiame registracijos lange suvesti tą pačią informaciją, kokios reikia prisijungimui (žr. 4 pav.). Tai atlikus paspausti registracijos mygtuką.

Teisingai registracijai reikia naudoti atitinkamos formos el. pašto adresą ([tekstas@tekstas.tekstas](mailto:tekstas@tekstas.tekstas)), kitu atveju naudotojas bus informuojamas apie neteisingą el. pašto adreso formą. Įvykdžius teisingą registraciją bus matomas profilio langas (žr. 5 pav.).

* **Profilio langas**

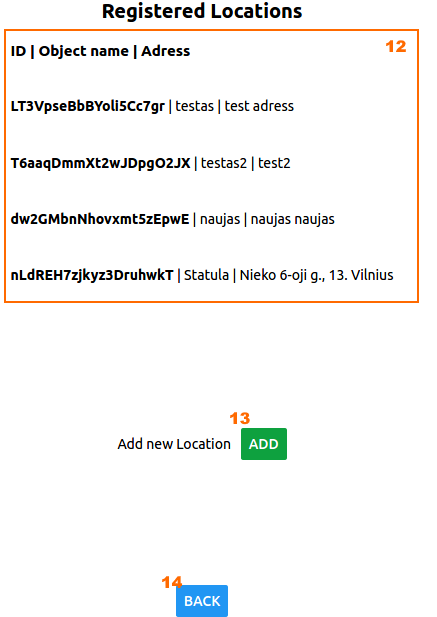
****

5 pav. profilio langas

Prisijungus ar užsiregistravus patenkame į profilio langą (žr. 5 pav.). Jame galime pamatyti savo profilio duomenis (dėžutė pažymėta numeriu 4), redaguoti profilį (5 mygtukas), pamatyti katino nuotrauką (6 mygtukas), taip pat peržiūrėti įvairius sistemos duomenis (7, 8, 9 ir 10 mygtukai) ir atsijungti nuo sistemos (11 mygtukas).

* **Duomenų peržiūra**

Norint peržiūrėti atitinkamus sistemos duomenis paspaudžiamas profilyje esantys mygtukai (6, 7, 8, 9, 10). Atsidariusiame lange atitinkama forma yra atvaizduoti atitinkami sistemos duomenys (žr. 6 pav. 12 dėžutė). Taip pat priklausomai nuo duomenų ir ką su jais sistema leidžia daryti yra atitinkamas funkcijas vykdantys mygtukai (13 mygtukas) ir yra grįžimo atgal į profilį mygtukas (14 mygtukas).

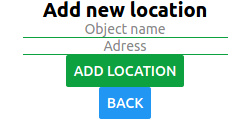


6 pav. saugotinų objektų peržiūros langas

* **Duomenų pridėjimas**

Norint pridėti atitinkamus duomenis sistemoje reikia naudotis atitinkamas funkcijas vykdančiais mygtukais (žr. 6 pav. 13 mygtukas). Atidarytame lange, sistema pateikia atitinkamus laukus, tam tikro objekto duomenims įrašyti, taip pat mygtukus pridėti objektą ir grįžti atgal į ankstesnį puslapį (žr. 7 pav.).

Priklausomai nuo duomenų tipo, atitinkamų duomenų pridėjimas gali skirtis nuo šioje instrukcijoje pavaizduoto pavyzdžio.



7 pav. naujo saugotino objekto pridėjimo langas

# Išvados ir pasiūlymai

Šio semestro metu pramokome rašyti React Native framework‘o internetines, bei mobiliąsias aplikacijas. Kuriant ši projektą buvo stengtasi pademonstruoti šiuos sugebėjimus, bei HTML ir Java script kalbų mąstymą, kuriant savo projektą. Šio projekto sistema turi sukūrtą vartotojų registraciją ir prisijungimą, norint atlikti didžiąją dalį sistemos veiksmų. Šioje sistemoje galima saugoti vietas ar objektus kuriuos norime ar galime saugoti. Taip pat joje galima peržiūrėti sargų atliekamus darbus (kuriuos objektus ar vietas jie saugo), bei pamatyti kada ir kuris sargas buvo prisijungęs prie sistemos. Šis projektas naudojasi daugybe HTML tipo komponentų, kaip View, Text, Button ir kiti. Duomenų valdymui sistema naudoja Firebase duomenų bazę, duomenys joje siekta kurti SQL pagrindu, norint patogiau juos valdyti ir sieti. Šis projektas, kartu su projekto dokumentacija yra patalpinti GitHub sistemoje adresu <https://github.com/LuciferBlade/2019> .

**Pasiūlymai:**

Norint toliau plėsti šią sistemą reiktų atsižvelgti į didesnį kiekį duomenų, kurie praverstų tokios sistemos darbui atitinkamose terpėse ir pridėti juos, bei jų valdymą, šioje sistemoje bei firebase duomenų bazėje. Taip pat norint konkrečiau stebėti ir valdyti tam tikrus duomenis, praverstų geresnis duomenų bazės dizainas ir valdymas perkeltas į atitinkamus reducer‘ius ir veiksmus. Taip pat norint patogesnės sistemos, reiktų sukurti geresnį aplikacijos dizainą, informacijos išdėstymą ir valdymą, jeigu aplikacija yra papildomai išplečiama, kam padėtų gilesnę panašių sistemų veikimo sąsajų analizė.

# Informacijos šaltiniai

1. UML diagramų standartų aprašymas. <https://www.uml-diagrams.org/>
2. Firebase konsolė ir dokumentacija. <https://console.firebase.google.com/>
3. React native dokumentacija. <https://facebook.github.io/react-native/docs>
4. React Redux. <https://react-redux.js.org/>
5. React native – React Redux Firebase. <https://react-redux-firebase.com/docs/integrations/react-native.html>