



Diplomski studij

**Informacijska i komunikacijska
tehnologija**

Računarstvo

**Telekomunikacije i informatika
Obradba informacija
Računalno inženjerstvo**

Ak.g. 2024./2025.

Internet stvari

BLEmindMe

Projekt

Luka Iveković

Leon Tišljarić

Oliver Ozvačić

Filip Antolić

Josip Mihelčić

Sadržaj

1. Uvod	3
1.1. Osnovno o usluzi	3
1.2. Korišteni senzori	3
1.3. Korišteni aktuatori	3
1.4. Slučaji uporabe	3
1.5. Slični projekti	4
2. Opis rješenja	5
2.1. Funkcionalnosti komponenti	5
1.1.1 Senzorski čvor	5
1.1.2 Aktuatori	5
1.1.3 Mrežni prilazi	5
1.1.4 IoT platforma	6
1.1.5 Korisničke aplikacije	7
2.2. Protokoli za komunikaciju između komponenti	7
2.3. Skica rješenja	7
3. IoT platforma	8
3.1. Osnovno o IoT platformi ThingsBoard	8
3.2. Funkcionalnosti korištene za realizaciju rješenja	8
4. Korisničke aplikacije	13
4.1. Razvijene korisničke aplikacije	13
4.2. Izgled korisničkih aplikacija	13
4.3. Korištene tehnologije	17
4.4. Najvažniji implementacijski detalji	17
1.1.6 Definirane rute	17
1.1.7 Komunikacija s ThingsBoard API	18
1.1.8 Registracija	19
5. Literatura	20

1. Uvod

1.1. Osnovno o usluzi

Usluga BLEmindMe razvijena je kao IoT rješenje koje korisnicima pomaže u praćenju i lociranju osobnih predmeta unutar prostora te ih upozorava u slučaju da pri izlasku iz prostorije zaborave neki od svojih predmeta. Sustav koristi ESP32 uređaj u kombinaciji s BLE iBeacon tehnologijom i ultrazvučnim senzorima za detekciju prisutnosti korisnika i njegovih stvari.

Kada sustav detektira da je korisnik napustio prostoriju, a neki od predmeta ostao unutar dometa ESP32 uređaja, aktivira se LED indikator kao vizualno upozorenje te se korisniku može poslati push notifikacija putem web aplikacije. Time se korisniku omogućuje pravovremeno postupanje kako bi izbjegao gubitak predmeta.

Web aplikacija omogućuje korisnicima registraciju i prijavu, pregled statusa uređaja i konfiguraciju sustava. Posebno su definirane administratorske i korisničke uloge za lakšu organizaciju i upravljanje.

1.2. Korišteni senzori

Ultrazvučni senzor udaljenost HC-SR04 mjeri udaljenost pomoću ultrazvučnih impulsa i mjerenja vremena koje je potrebno da zvučni valovi dođu do prepreke, odbiju se i vrate na izvor. Mjerenje se pokreće impulsom na trigger pin senzora, a na echo pin senzor šalje impuls koji traje onoliko koliko je vremena prošlo od slanja signala do detekcije.

1.3. Korišteni aktuatori

Kao aktuator koristimo LED diodu koju putem GPIO pina na uređaju ESP32C6 DevKitC-1 i otpornika palimo ili gasimo kako bi signalizirali korisniku kada zaboravi stvari u prostoriji.

1.4. Slučaji uporabe

- Praćenje okvirne lokacije stvari u odnosu na ESP32 uređaj kako bi ih po potrebi mogli locirati
- Obavješćavanje korisnika u slučaju da zaboravi stvari unutar prostorije kada ju napusti

1.5. Slični projekti

Jedan od sličnih projekata [3] koristio je uređaj ESP32 u kombinaciji s OLED zaslonom kako bi se izradio prenosivi uređaj za pronalaženje izgubljenih predmeta u blizini pomoću RSSI (Received Signal Strength Indicator) vrijednosti, ali bez uporabe IoT platforme. Sljedeći projekt [2] također koristi uređaj ESP32 koji ima ulogu nadziranja staklenika te mjerenja temperature, vlage zraka i vlažnosti tla, a podatke zatim putem MQTT protokola šalje na IoT platformu ThingsBoard s ciljem upravljanja navodnjavanjem. Treći projekt [1] implementira IoT sustav za praćenje paketa u stvarnom vremenu koristeći GPS tracker smješten unutar paketa čime se korisnicima omogućuje precizno praćenje lokacije pošiljke. Osim lokacije, sustav dodatno nadzire temperaturu i vlagu unutar paketa tijekom transporta.

2. Opis rješenja

2.1. Funkcionalnosti komponenti

1.1.1 Senzorski čvor

Senzorski čvor sastoji se od uređaja ESP32-C6 DevKitC-1 na koji je povezan ultrazvučni senzor udaljenosti HC-SR04. Uređaj se pri uključenju povezuje na zadanu Wi-Fi mrežu i podešava mrežne parametre putem protokola DHCP. Nakon uspostave Wi-Fi veze, uređaj počinje osluškivati BLE iBeacon komunikaciju. Kada su uspostavljene Wi-Fi i BLE komunikacije, uređaj se povezuje na zadani MQTT broker koristeći zadane proxy vjerodajnice za prijavu. Nakon uspostave MQTT proxy komunikacije, uređaj se pretplaćuje na topic „v1/devices/me/attributes“ i objavljuje na isti topic podatke o korisniku kojem pripada. Na taj publish, platforma odgovara na taj publish korisničkim imenom, lozinkom i identifikatorom za prijavu uređaja te popisom uređaja i MAC adresa koji su povezani s tim korisnikom. Dobiveni podatci se spremaju i prekida se proxy veza s brokerom. Nakon prekida proxy veze, uspostavlja se korisnička veza koja koristi podatke za prijavu koje je uređaj dobio od proxyja. Po uspješnoj prijavi, uređaj se pretplaćuje na topic „v1/devices/me/attributes“ na kojem osluškuje poruku kojoj je sadržaj JSON s ključem „lightStatus_<username>“, a vrijednost tog ključa postavlja se na LED. Unutar beskonačne petlje, uređaj prima oglašavanja od iBeacon uređaja koja sadrže MAC adrese uređaja koji se oglašavaju. Uređaj određuje koji od korisnikovih uređaja su prisutni u prostoriji usporedbom primljenih MAC adresa sa spremljenim MAC adresama. Vrijednost prisutnosti svih uređaja javlja platformi na topic „v1/devices/esp/<username>/stateUpdate“ s periodom 10 sekundi. Za svaki se uređaj određuje procjena udaljenosti od ESP32 uređaja koristeći jačinu BLE signala. Pomoću senzora HC-SR04 mjeri se udaljenost od senzora koji je postavljen na jednoj strani ulaznih vrata do druge strane vrata. Mjerenja se vrše 2 puta u sekundi. Ako mjerenje značajnije odstupa od prosjeka, uređaj zaključuje da je korisnik ušao/izašao iz prostora ovisno o prijašnjem stanju prisutnosti. O promjeni stanja prisutnosti obavještava platformu na topic „v1/devices/esp/<username>/userLeft“ i pokreće timeout za ponovna mjerenja od 10 sekundi unutar kojih ne bilježi ulaske/izlaske.

1.1.2 Aktuatori

Jedini aktuator u sustavu je LED koji je spojen na GPIO pin uređaja ESP32-C6 DevKitC-1. U seriju je spojen otpornik kako bi ograničio struju kroz diodu. Funkcionalnost aktuatora je da pruža korisniku vizualnu obavijest kada je opaženo da je zaboravljena neka od korisnikovih stvari unutar prostora kojemu je senzorski čvor pridijeljen.

1.1.3 Mrežni prilazi

Mrežni prilaz nije korišten zato što se ESP32 direktno povezuje na Wi-Fi i ima izvorno podržanu BLE komunikaciju.

1.1.4 IoT platforma

U projektu korištena je IoT platforma ThingsBoard (Community Edition) koja omogućuje jednostavno povezivanje, upravljanje i obradu podataka s uređaja. Kako je korištena inačica platforme otvorenog koda, projekt nema dodatne troškove vezane za njeno korištenje. U sklopu projekta na platformi su definirani uređaji (Devices), profili uređaja (Device profiles), lanci pravila (Rule chains) i korisnici (Customers).

Početno spajanje uređaja na platformu riješeno je korištenjem virtualnog "proxy" uređaja. Fizički ESP32 uređaj prvo se spaja na MQTT broker s javno poznatim vjerodajnicama. Nakon slanja korisničkih podataka, Proxy Rule Chain dohvaća stvarne MQTT vjerodajnice za taj ESP32 uređaj i popis povezanih BLE tagova pozivima na ThingsBoard REST API. Ti se podaci zatim vraćaju uređaju kao dijeljeni atribut (SHARED_SCOPE), a uređaj pri daljnjoj komunikaciji koristi dobivene jedinstvene vjerodajnice.

Komunikacija se odvija putem MQTT protokola. Uređaj šalje telemetrijske podatke na prilagođene teme poput v1/devices/esp/<username>/stateUpdate i v1/devices/esp/<username>/userLeft.

Definirana su tri profila uređaja:

- **ESP32:** Profil za kontrolne uređaje koji šalju telemetriju o prisutnosti tagova i statusu korisnika te primaju naredbe.
- **Tags:** Profil koji označava BLE iBeacon uređaje povezane sa stvarima koje korisnik želi pratiti (npr. ključevi, novčanik).
- **Proxy:** Profil za virtualni posrednik koji omogućuje inicijalnu autentifikaciju i dohvat konfiguracije za ESP32 uređaje.

Logika je implementirana korištenjem sljedećih lanaca pravila:

- **Proxy Rule Chain:** Prima zahtjev od ESP32 uređaja, autentificira korisnika putem REST API-ja, dohvaća popis povezanih tagova i stvarne MQTT vjerodajnice uređaja te ih vraća kao dijeljeni atribut.
- **ESP32 Rule Chain:** Obraduje telemetrijske poruke s ESP32 uređaja. Kada primi poruku da je korisnik napustio prostoriju (userLeft), aktivira odgodu od 10 sekundi i postavlja internu zastavicu hasLeft na true. Prilikom sljedeće poruke o stanju tagova (stateUpdate) provjerava tu zastavicu. Ako je korisnik otišao, a neki od tagova je i dalje prisutan, lanac postavlja lightState atribut i šalje ga kao naredbu uređaju da upali LED diodu. Lanac pravila također prosljeđuje obrađene telemetrijske podatke o tagovima u Tags Rule Chain.
- **Tags Rule Chain:** Služi kao završni korak za pohranu telemetrijskih podataka. Sadrži jedan čvor (Save telemetry) koji zapisuje podatke o stanju za svaki pojedini tag nakon što je ESP32 Rule Chain promijenio izvornog pošiljatelja poruke s ESP32 uređaja na sam tag.

Interakcija s korisnikom ostvarena je putem progresivne web aplikacije razvijene za potrebe projekta. Komunikacija između ThingsBoard platforme i aplikacije odvija se korištenjem REST API-ja.

1.1.5 Korisničke aplikacije

Korisnička aplikacija napravljena je kao progresivna web aplikacija. Aplikacija ima različite funkcionalnosti ovisno o ulozi korisniku definiranoj na platformi. Prva mogućnost je prijava i registracija. Nakon toga korisnici s ulogom „Tenant Admin“ imaju mogućnosti izrade novih uređaja, dodjelu uređaja korisnicima te povezivanja ESP32 uređaja s uređajima za praćenje (BLE tagovima). Korisnici koji nemaju ulogu administratora pomoću aplikacije mogu vidjeti svoje registrirane uređaje, njihove informacije te pregledati s kojim uređajima su njihovi uređaji povezani. Svi korisnici također mogu omogućiti slanje obavijesti što omogućuje primanje push notifikacija. Korisnik prima push notifikaciju kada zaboravi neku stvar.

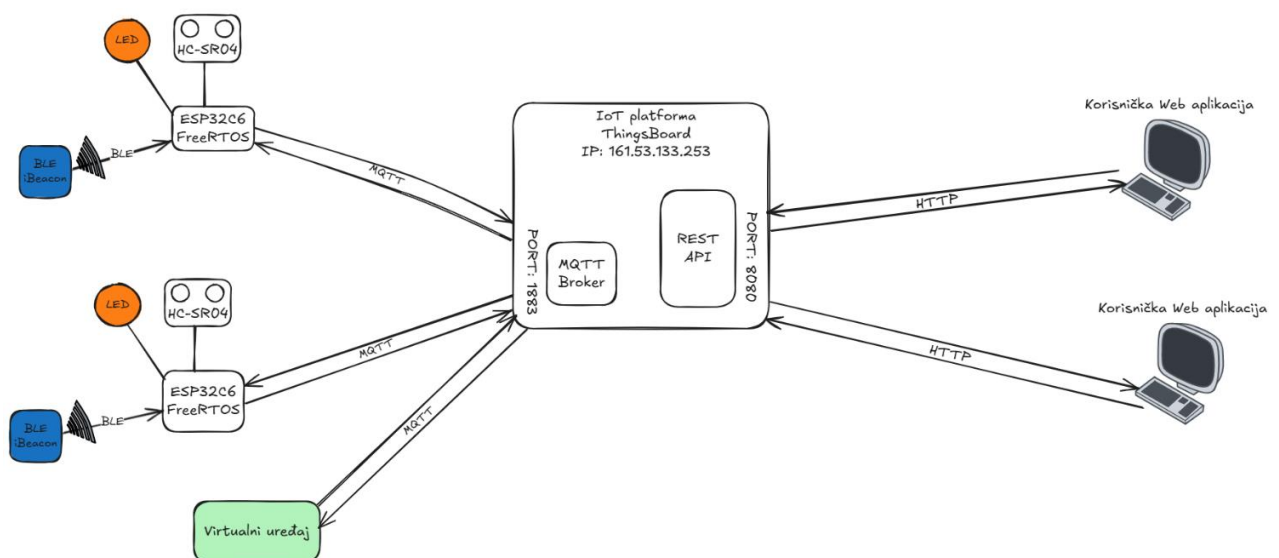
2.2. Protokoli za komunikaciju između komponenti

Između senzorskih čvorova i platforme koristi se protokol MQTT. MQTT je publish/subscribe protokol koji usmjeravanje poruka bazira na tzv. topicima na koje se korisnici pretplaćuju ili objavljuju podatke.

Između BLE iBeacon uređaja i senzorskog čvora koristi se iBeacon protokol kojeg je razvio Apple i za komunikaciju koristi BLE protokol, a omogućuje detekciju uređaja u blizini i procjenu njihove udaljenosti koristeći jačine signala.

Između platforme i korisničke web aplikacije koristi se protokol HTTP i sučelje REST API kojeg definira ThingsBoard za pristup i upravljanje svim korisnicima, uređajima i parametrima sustava.

2.3. Skica rješenja



3. IoT platforma

3.1. Osnovno o IoT platformi ThingsBoard

U ovom projektu korištena je IoT platforma ThingsBoard. IoT platforma ThingsBoard rješenje je otvorenog koda koje omogućuje povezivanje i upravljanje različitim IoT uređajima putem Interneta. S obzirom da se koristila verzija platforme otvorenog koda, za ovo rješenje nema nikakve naplate. Međutim, ThingsBoard također nudi verziju Professional Edition koja se naplaćuje mjesečno, a iznos ovisi o odabranom planu pretplate.

Platforma omogućuje prikupljanje, obradu, upravljanje i konačnu vizualizaciju podataka, a dodatno podržava koncepte kao što su: profili, sredstva, lanci pravila, definiranje veza između elemenata, pozivanje udaljene procedure, kreiranje događaja i kontrole alarma, autentifikacija uređaja i korisnika te upravljanje kupcima.

ThingsBoard uređaji (eng. devices) fizički su uređaji koji sudjeluju u mreži Interneta stvari te imaju mogućnost slanja podataka na platformu ThingsBoard. Svakom uređaju pridodaje se jedinstveni identifikator preko kojega se prepoznaje kao pošilatelj u poruci, a dodatno mu se može konfigurirati: naziv, profil uređaja, oznaka, atributi, vjerodostojnice, oznaka o načinu rada uređaja kao IoT mrežni prilaz, itd.

Profil (eng. profiles) omogućuju definiranje zajedničkih postavki uređaja te svaki uređaj mora imati samo jedan profil uređaja. Prilikom izrade novog profila zahtijeva se postavljanje lanca pravila kojemu će se prosljeđivati dolazne poruke te definiranje vrste protokola koji ih prijenosi do platforme.

Lanci pravila (eng. rule chains) definiraju tijekom prolaska dolazećih poruka kroz niz međusobno povezanih čvorova različitih funkcionalnosti. Dodatno se koriste za modifikaciju poruka na poslužiteljskoj strani prije njihove pohrane.

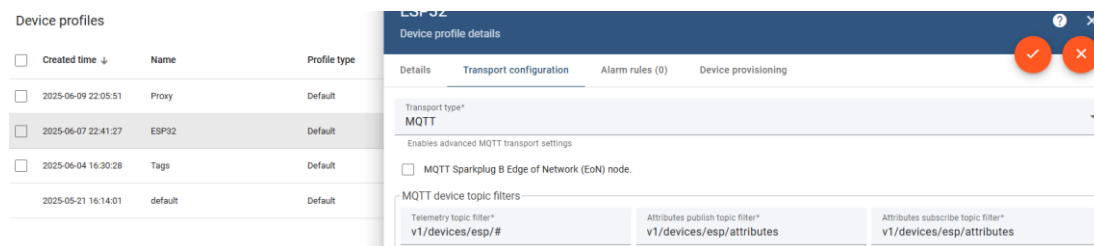
Korisnici (eng. customers) predstavljaju organizacije ili pojedince kojima je dodijeljen pristup određenim uređajima i resursima unutar platforme ThingsBoard. Svaki korisnik može imati vlastite uređaje, nadzorne ploče, lance pravila, podatke, itd.

3.2. Funkcionalnosti korištene za realizaciju rješenja

Kod kartice Customers izrađene su dvije grupe korisnika - Users i User2. Grupa Users ima dva korisnika, a druga grupa samo jednog. Ovdje nailazimo na ograničenje koje dopušta pridruživanje uređaja isključivo grupi korisnika, a ne pojedinačnom korisniku unutar grupe što zapravo zahtijeva da se u jednoj grupi nalazi samo jedan korisnik. Prema tome, pretpostavljamo da u grupi Users korisnici imaju zajedničke stvari.

Customers > Customer users			
Users: Customer users			
<input type="checkbox"/> Created time ↓	First name	Last name	Email
<input type="checkbox"/> 2025-06-04 16:01:04	Iva	Ivić	iva.ivic@gmail.com
<input type="checkbox"/> 2025-06-04 16:00:21	Marko	Markić	marko.markic@gmail.com

Kod kartice Device profiles definirana su tri različita profila: Tags, ESP32 i Proxy. Profil Tags dodjeljuje se svim stvarima korisnika (npr. novčaniku, mobilnom telefonu, ključevima, itd.). Profil ESP32 dodjeljuje se svim uređajima ESP32, tj. kontrolnim uređajima zaduženim za nadzor svih stvari u nekoj sobi. Profil Proxy postavlja se kod virtualnog posrednika koji je zadužen za vraćanje vjerodostojnica uređaja ESP32 nakon autentifikacije korisnika. Protokol MQTT koristio se za prijenos podataka od fizičkih uređaja do platforme. U slučaju uporabe protokola MQTT, ThingsBoard dozvoljava samo slanje podataka na teme telemetrije. Ovo ograničenje može se zaobići pretplatom na dijeljene attribute uređaja ESP32. Dijeljeni atributi dozvoljavaju da i fizički uređaji i mrežna aplikacija imaju njima pristup.



Uređaji se dodaju unutar kartice Devices. Za ovo rješenje definirana su dva uređaja – ESP1 i ESP2. Svaki od njih pridružen je različitoj grupi korisnika (Users i Users2), a njihova oznaka definira sobu za koju je zadužen. Također, za svaku grupu korisnika izrađene su različite stvari (Wallet, Keys, Phone i Headphones) koje su logičkom vezom pridružene jednom od ESP uređaja kako bi uređaj znao koje stvari treba pratiti. Svaka stvar ima oznaku koja definira njezinu stvarnu MAC adresu. Dodatno, definiran je virtualni posrednik Proxy s javno poznatim vjerodostojnicama kako bi se uređaji ESP32 mogli prvobitno spojiti na platformu i preuzeti svoje vjerodostojnice te podatke o stvarima za koje su zaduženi.

Devices Device Filter

<input type="checkbox"/>	Created time ↓	Name	Device profile	Label	State	Customer
<input type="checkbox"/>	2025-06-12 15:26:08	Headphones	Tags	00:11:22:33:44:00	Inactive	Users2
<input type="checkbox"/>	2025-06-12 15:25:21	Phone	Tags	00:11:22:33:44:99	Inactive	Users2
<input type="checkbox"/>	2025-06-12 15:24:02	ESP2	ESP32	Living Room	Inactive	Users2
<input type="checkbox"/>	2025-06-09 22:06:31	Proxy	Proxy		Inactive	
<input type="checkbox"/>	2025-06-09 21:35:52	Wallet	Tags	00:11:22:33:44:55	Inactive	Users
<input type="checkbox"/>	2025-06-07 22:43:20	ESP1	ESP32	Office	Inactive	Users
<input type="checkbox"/>	2025-06-05 17:49:07	Keys	Tags	00:11:22:33:44:77	Inactive	Users

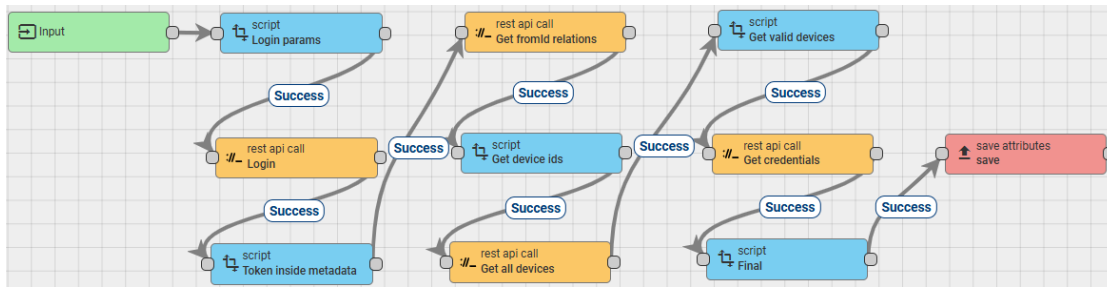
Nakon toga, izrađena su tri lanca pravila: Proxy Rule Chain, ESP32 Rule Chain i Tags Rule Chain. Lanac pravila Proxy Rule Chain kao ulaz prima poruku sljedećeg formata:

```
{
  username: "iva.ivic@gmail.com",
  password: "ivaivic",
  customerId: "ce4757a0-414b-11f0-a544-db21b46190ed",
  espId: "0c75afe0-43e0-11f0-a544-db21b46190ed"
}
```

Sljedeći čvor preuzima korisničko ime i lozinku te, putem krajnje točke za autentifikaciju REST API-ja platforme, dobiva token koji se zatim koristi za dohvat svih uređaja na temelju parametra customerId. Pretpostavlja se da svaki ESP uređaj ima svoj tvornički espId pomoću kojega se filtriraju i vraćaju samo oni uređaji za čiji je nadzor on zadužen. Predzadnji čvor formatira dohvaćene podatke u sljedeći format:

```
{
  "credentials": {
    "clientId": "ESP1",
    "userName": "ESP1",
    "password": "password1"
  },
  "devices": [{
    "name": "ESP1",
    "id": "0c75afe0-43e0-11f0-a544-db21b46190ed"
    "label": "Office"
  }, {
    "name": "Keys",
    "id": "9d7805f0-4224-11f0-a544-db21b46190ed"
    "label": "00:11:22:33:44:66"
  }, {
    "name": "Wallet",
    "id": "f4333f80-4568-11f0-a544-db21b46190ed"
    "label": "00:11:22:33:44:55"
  }],
  "result": "ok"
}
```

Zadnji čvor zatim pohranjuje te podatke u dijeljene atribute uređaja čime fizički uređaj pretplaćeni na temu vezanu uz atribute (`v1/devices/me/attributes`) prima te podatke.



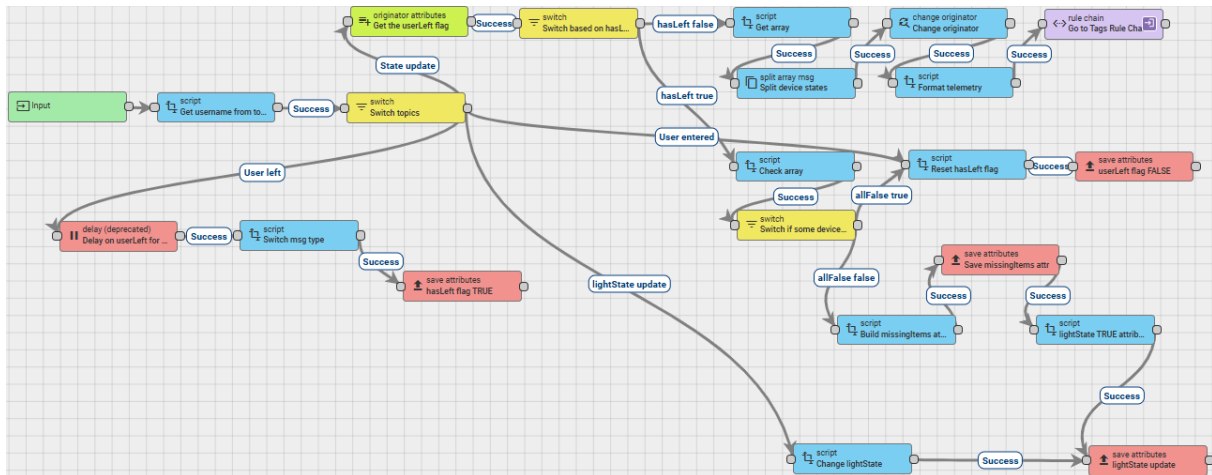
Lanac pravila ESP32 može primiti tri različite vrste poruka kao ulaz putem tri različite podteme s obzirom na to da je tema za slanje telemetrije definirana pomoću zamjenskog znaka „#“ – `v1/devices/esp/#`. Na temu `v1/devices/esp/stateUpdate` uređaj ESP šalje podatke o stanjima svake od korisnikovih stvari. Ti podaci uključuju: trenutnu udaljenost od uređaja, vidljivost, naziv uređaja te identifikator uređaja:

```
{
  "devices": [
    {
      "deviceId": "f4333f80-4568-11f0-a544-db21b46190ed",
      "deviceName": "Keys",
      "distance": 2.3,
      "present": true
    },
    {
      "deviceId": "9d7805f0-4224-11f0-a544-db21b46190ed",
      "deviceName": "Wallet",
      "distance": 1.1,
      "present": true
    }
  ]
}
```

Na temu `v1/devices/esp/userLeft` šalje se podatak tipa *boolean* u trenutku kada senzor očitava da je korisnik ušao ili izašao iz prostorije. Zbog ograničenja u korištenju samo jednog senzora, stanje se naizmjenično mijenja pri svakom novom očitavanju pokreta. Ova se poruka pohranjuje kao poslužiteljski atribut, a prilikom nove poruke o stanju svih stvari provjerava se je li korisnik napustio prostoriju te postoji li barem jedna stvar koja i dalje ima vidljivost postavljenu na *true*. U tom slučaju, zaključuje se da je korisnik ostavio neku od svojih stvari u sobi te se šalje poruka tipa *boolean* na pretplatničku temu `v1/devices/me/attributes` kako bi ESP uređaj uključio svoje svijetlo:

```
{
    "lightState_<username>": true
}
```

Kada korisnik ponovno uđe u prostoriju, atribut zadužen za praćenje njegove prisutnosti postavlja se na *false*. Atribut se također vraća se na početnu vrijednost *false* ako je korisnik napustio prostoriju, a poruke o stanju svih njegovih stvari imaju vidljivost postavljenu također na *false*. Time se zaključuje da je korisnik ponio sve stvari sa sobom.



4. Korisničke aplikacije

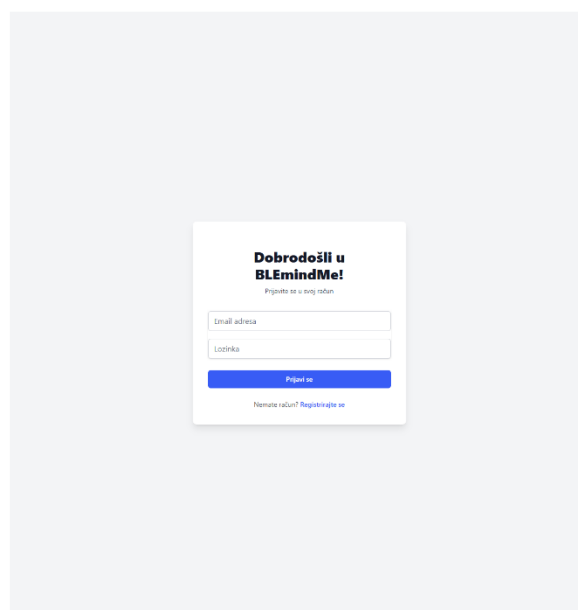
4.1. Razvijene korisničke aplikacije

Razvijena je progresivna web aplikacija za pregled i administraciju svih uređaja u sustavu. Aplikacija ima različite funkcionalnosti ovisno o ulozi korisniku definiranoj na platformi. Prva mogućnost je prijava i registracija. Korisnici koji se prijave sa ulogom „Tenant Admin“ imaju mogućnosti upravljanja uređajima u sustavu. Imaju mogućnosti dodavanja novih uređaja, dodjelu uređaja korisnicima te povezivanja ESP32 uređaja sa uređajima za praćenje (BLE tagovima). Korisnici koji nemaju ulogu administratora pomoću aplikacije mogu vidjeti svoje registrirane uređaje, njihove informacije te pregledati s kojim su uređajima njihovi uređaji povezani. Ključna funkcionalnost aplikacije je mogućnost primanja obavijesti u obliku push notifikacije kada ESP uređaj prepozna da je korisnik zaboravio neku stvar.

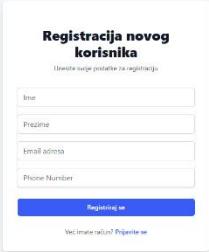
4.2. Izgled korisničkih aplikacija

Na slikama prikazani su najbitniji dijelovi aplikacije.

1. **Prijava** – već spomenuta prijava prijavljuje korisnika u račun i prikazuje mu akcije ovisno o njegovoj ulozi definiranoj na platformi



2. **Registracija** – osim prijave moguće je kreirati i novi račun za novog korisnika, nakon unosa glavnih informacija korisnik je preusmjeren na ThingsBoard platformu na kojoj mora postaviti password, nakon čega se može vratiti natrag u aplikaciju i prijaviti



Registracija novog korisnika
Unesite svoje podatke za registraciju

Ime

Prezime

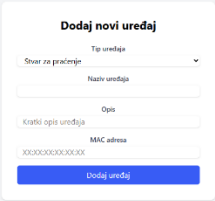
Email adresa

Phone Number

Registriraj se

Nec imate račun? Prijavite se

3. **Dodavanje novih uređaja** – ako korisnik ima ulogu administratora, može dodavati nove uređaje; moguće je odrediti vrstu uređaja (ESP ili Tag), dodijeliti im naziv i opis te upisati MAC adresu



BLEmindMe [Pocetna](#) [Upravljanje uređajima](#) [Dodaj uređaj](#) [Povezivanje uređaja](#) [Izika: hrvatski@ble.hr](#) [Odjava](#)

Dodaj novi uređaj

Tip uređaja
☐ Stvar za prenošenje ☒ Tip uređaja ▼

Naziv uređaja

Opis

Kratki opis uređaja

MAC adresa

Dodaj uređaj

4. **Dodjela uređaja korisnicima** – ako korisnik ima ulogu administratora, može dodjeljivati postojeće uređaje postojećim korisnicima

BLEmindMe Početna Upravljanje uređajima Dodjela uređaja Povezivanje uređaja Iuka.Irekovic@fer.hr Odjava

Dodjeli uređaj korisniku

Odaberi korisnika
test@test.com (Istari testni)

Odaberi uređaj
Laptop (Tag)

Dodjeli uređaj

5. **Povezivanje uređaja** – ako korisnik ima ulogu administratora, može povezivati ESP uređaje s uređajima za praćenje (BLE tagovima)

BLEmindMe Početna Upravljanje uređajima Dodjela uređaja Povezivanje uređaja Iuka.Irekovic@fer.hr Odjava

Povezivanje uređaja

Uređaji za provjeru (ESP32)

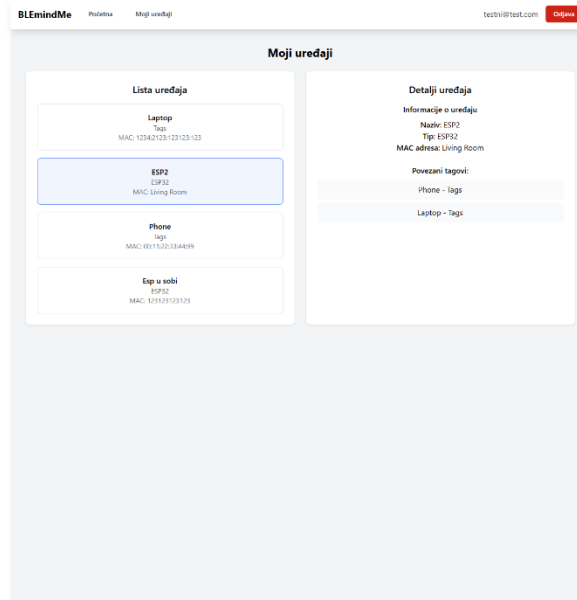
ODABERI	NAZIV	OPIS	MAC ADRESA
<input checked="" type="radio"/>	ESP1	ESP32	Office
<input type="radio"/>	ESP2	ESP32	Living Room
<input type="radio"/>	ESPARA 2	ESP32	123123123
<input type="radio"/>	Esp u sobi	ESP32	123123123123

Stvari za praćenje (BLE Tags)

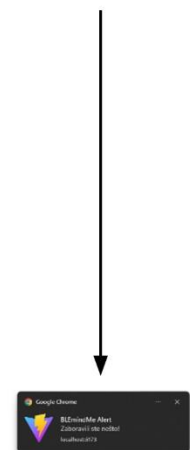
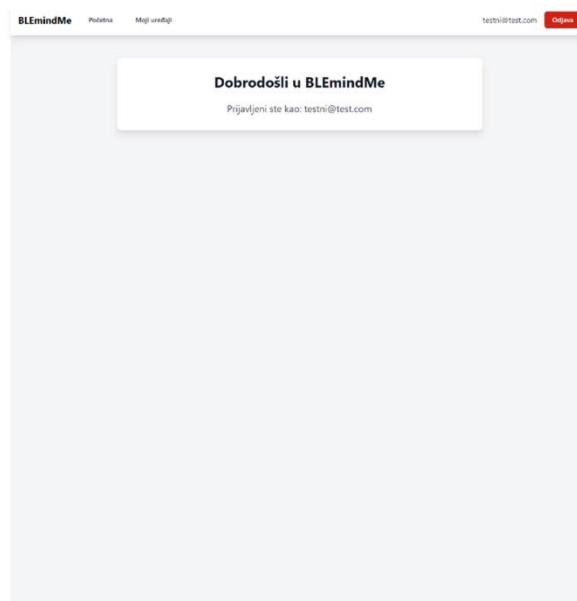
ODABERI	NAZIV	OPIS	MAC ADRESA
<input type="radio"/>	Laptop	Tags	123421231231
<input type="radio"/>	Keys	Tags	c7d1115c7e2
<input checked="" type="radio"/>	Phone	Tags	0011231231
<input type="radio"/>	A4872CCDF33ardesd	Tags	Room 234 5e
<input type="radio"/>	ESPARA	Tags	12312312312
<input type="radio"/>	Headphones	Tags	001122331
<input type="radio"/>	Wallet	Tags	e1175d6c7b1

Poveži uređaje

6. **Pregled uređaja** – korisnici koji nemaju ulogu administratora mogu pregledavati svoju listu uređaja, njihove osnovne informacije te uređaje s kojima su njihovi uređaji povezani



7. **Obavijesti** – svi korisnici mogu omogućiti primanje obavijesti, nakon čega će svaki put kada ESP zamijeti da je korisnik zaboravio neku stvar primiti obavijest u obliku push notifikacije



4.3. Korištene tehnologije

Za izradu progresivne web aplikacije korišten je JavaScript programski jezik, Vite build tool i React biblioteka.

4.4. Najvažniji implementacijski detalji

1.1.6 Definirane rute

S obzirom da aplikacija pruža različite mogućnosti ovisno o ulozi prijavljenog korisnika, napravljene su posebne rute koje definiraju mogućnost pristupa korisnika određenoj ruti.

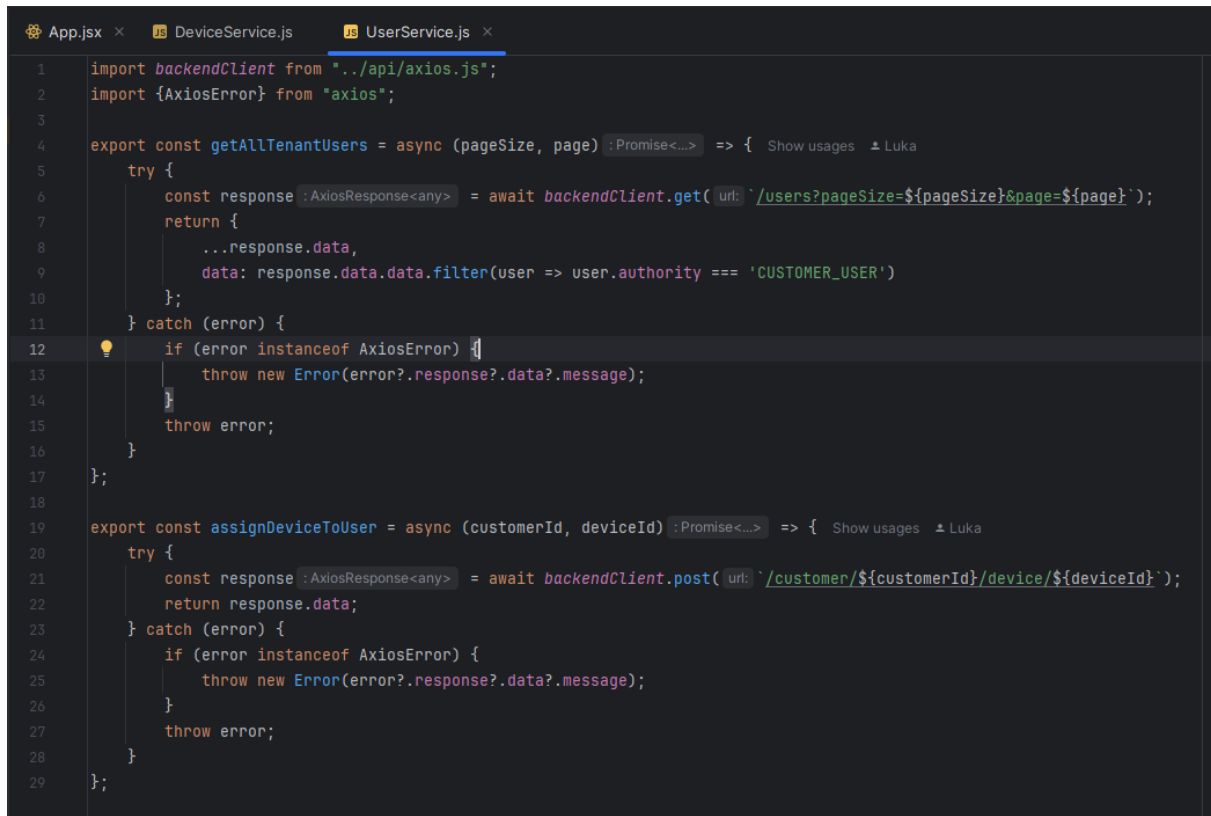
```

App.jsx x
16 function App() { Show usages 1 Luka
17   return (
18     <AuthProvider>
19       <NotificationProvider>
20         <Router>
21           <div className="App">
22             <Routes>
23               <Route path="/login" element={<Login />} />
24               <Route path="/register" element={<Registration />} />
25               <Route path="/home" element={<PrivateRoute><Home /></PrivateRoute>} />
26               <Route path="/" element={<Login />} />
27             <Route
28               path="/device-management"
29               element={
30                 <PrivateRoute>
31                   <TenantAdminRoute>
32                     <DeviceManagement />
33                   </TenantAdminRoute>
34                 </PrivateRoute>
35               }
36             />
37             <Route
38               path="/device-assignment"
39               element={
40                 <PrivateRoute>
41                   <TenantAdminRoute>
42                     <DeviceAssignment />
43                   </TenantAdminRoute>
44                 </PrivateRoute>
45               }
46             />
47             <Route
48               path="/device-mapping"
49               element={
50                 <PrivateRoute>
51                   <TenantAdminRoute>
52                     <DeviceMapping />
53                   </TenantAdminRoute>
54                 </PrivateRoute>
55               }
56             />
57             <Route
58               path="/my-devices"
59               element={
60                 <PrivateRoute>
61                   <CustomerUserRoute>
62                     <CustomerDevices />
63                   </CustomerUserRoute>
64                 </PrivateRoute>
65               }
66             />
67           </Routes>
68         </div>
69       </Router>
70     </NotificationProvider>
71   </AuthProvider>
72 )
73 }

```

1.1.7 Komunikacija s ThingsBoard API

Web aplikacija dohvaća podatke iz ThingsBoard platforme pomoću definiranog API-ja na platformi preko HTTP protokola. Za potrebe pozivanja API-ja koristio se axios. Primjer poziva API-ja prikazan je na slici. Stvorena je custom axios instanca koja ima definiran request interceptor pomoću kojega na svaki request prema ThingsBoard API-ju dodaje header „X-Authorization“ za autentifikaciju i autorizaciju.



```
1 import backendClient from "../api/axios.js";
2 import {AxiosError} from "axios";
3
4 export const getAllTenantUsers = async (pageSize, page) :Promise<...> => { Show usages  ▲ Luka
5   try {
6     const response :AxiosResponse<any> = await backendClient.get( url: `/users?pageSize=${pageSize}&page=${page}` );
7     return {
8       ...response.data,
9       data: response.data.data.filter(user => user.authority === 'CUSTOMER_USER')
10    };
11  } catch (error) {
12    if (error instanceof AxiosError) {
13      throw new Error(error?.response?.data?.message);
14    }
15    throw error;
16  }
17 };
18
19 export const assignDeviceToUser = async (customerId, deviceId) :Promise<...> => { Show usages  ▲ Luka
20   try {
21     const response :AxiosResponse<any> = await backendClient.post( url: `/customer/${customerId}/device/${deviceId}` );
22     return response.data;
23   } catch (error) {
24     if (error instanceof AxiosError) {
25       throw new Error(error?.response?.data?.message);
26     }
27     throw error;
28   }
29 };
```

1.1.8 Registracija

Bitno je spomenuti i implementaciju registracije koja nije bila vrlo jednostavna zbog ograničenosti dostupnih API-ja na ThingsBoard platformi za registraciju. S obzirom da su svi API-ji za izradu novih korisnika zaštićeni autentifikacijom prvo je bilo potrebno logirati korisnika s računom administratora koji ima pravo na dodavanje korisnika. Nakon toga moguće je kreirati novog korisnika, a prije kreiranja novog korisnika potrebno je kreirati i novog „consumera“. Naime, kreiranjem korisnika korisnik ne postavlja lozinku za svoj račun pa je potrebno dodatno dohvatiti link za aktivaciju te preusmjeriti korisnika na taj link za postavljanje lozinke unutar ThingsBoard platforme. Na kraju potrebno je odjaviti administratora.

```
53
54   const registerUser = async (userData) => { Show usages  ▲ Luka
55     // eslint-disable-next-line no-useless-catch
56     try {
57       await loginUser(TENANT_USERNAME, TENANT_PASSWORD);
58
59       const createdCustomer :any|undefined = await createCustomer(userData.firstName, userData.email, userData.phoneNumber);
60       const createdUser :any|undefined = await createUser(userData.email, userData.firstName, userData.lastName, userData.phoneNumber, createdCustomer.id.id);
61
62       const activationLink :any|undefined = await getUserActivationLink(createdUser.id.id);
63
64       await logoutUser();
65
66       return activationLink;
67     } catch (error) {
68       throw error;
69     }
70   };
71
72
```

Kako bi se osigurala sigurnost ovoga pristupa, potrebno je izraditi novoga korisnika s ulogom administratora za ovu aplikaciju. Uz to, vrlo je bitno sigurno pohraniti kredencijale tako što se te informacije pohranjuju na serveru, a ne na klijentskoj aplikaciji i dobro šifriraju.

5. Literatura

- [1] Chhorn, S., et al. "Development of ESP32-Based Smart Greenhouse Controller." 2022 *IEEE 8th World Forum on Internet of Things (WF-IoT)*, 2022, pp. 1-6. *IEEE Xplore*, <https://doi.org/10.1109/WF-IoT54382.2022.10152112>. Accessed June 2025.
- [2] Motade, S. N., et al. "Smart Data Tracking for Package Transportation." 2023 *International Conference on Network, Multimedia and Information Technology (NMITCON)*, 2023, pp. 1-3. *IEEE Xplore*, <https://doi.org/10.1109/NMITCON58196.2023.10275845>. Accessed June 2025.
- [3] Sachan, K. K., and A. Natarajan. "Proximity Detection Based Low-Cost and Handheld IoT Device for Tracking Lost Objects." 2024 *IEEE International Symposium on Smart Electronic Systems (iSES)*, 2024, pp. 40-43. *IEEE Xplore*, <https://doi.org/10.1109/iSES63344.2024.00019>. Accessed June 2025.