

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1443

SUSTAV ZA PRAĆENJE STANJA MEDICINSKIH UREĐAJA

Filip Ljubotina

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1443

SUSTAV ZA PRAĆENJE STANJA MEDICINSKIH UREĐAJA

Filip Ljubotina

Zagreb, lipanj 2024.

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1443

Pristupnik: **Filip Ljubotina (0036541562)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: izv. prof. dr. sc. Alan Jović

Zadatak: **Sustav za praćenje stanja medicinskih uređaja**

Opis zadatka:

Cilj završnog rada je izrada web i mobilne aplikacije zaduženih za praćenje stanja medicinskih uređaja. Mobilna aplikacija treba poslužiti za slikanje i prepoznavanje barkoda medicinskih uređaja, a web aplikacija treba omogućiti učinkovitu administraciju medicinskih uređaja u smislu praćenja prisutnih medicinskih uređaja tijekom vremena, unos komentara o uređajima i pregleda dostupne dokumentacije za prisutne uređaje. Potrebno je isprobati nekoliko dostupnih biblioteka za prepoznavanje barkoda te ugraditi onu koja radi najbolje za navedenu primjenu. Prepoznati barkod je potrebno pohraniti u bazu podataka. Web aplikacija treba omogućiti registraciju novih korisnika, prijavu u sustav i administraciju uređaja, sve poštujući sigurnosne aspekte rada s podacima o medicinskim uređajima. Implementaciju web i mobilne aplikacije potrebno je napraviti u programskim jezicima po vlastitom izboru.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod	3
2. Funkcionalni zahtjevi	5
2.1. Web aplikacija	5
2.2. Mobilna aplikacija	12
3. Arhitektura i dizajn sustava	15
3.1. Baza podataka	16
3.1.1. Dijagram baze podataka	16
3.1.2. Opis tablica	16
3.2. Poslužiteljska strana	20
3.2.1. Dijagram razreda	20
3.2.2. Sigurnost	23
3.3. Klijentska strana web aplikacije	24
3.3.1. Dijagram komponenti	24
3.3.2. Sigurnost	26
3.4. Klijentska strana mobilne aplikacije	29
3.4.1. Dijagram komponenti	29
3.4.2. Sigurnost	30
4. Implementacija i korisničko sučelje	31
4.1. Korisničko sučelje web aplikacije	31
4.2. Korisničko sučelje mobilne aplikacije	35
4.3. Korištene tehnologije i alati	38
5. Zaključak	41

Literatura	42
Sažetak	43
Abstract	44

1. Uvod

U modernom zdravstvenom sustavu, praćenje i održavanje medicinskih uređaja od velike je važnosti za osiguranje visoke kvalitete skrbi pacijenata. Nažalost, mnoge bolnice i zdravstvene ustanove suočavaju se s nedostatkom kvalitetne evidencije medicinskih uređaja. Ovaj problem rezultira time da visokokvalificirani medicinski radnici troše više vremena nego što je potrebno na administrativne zadatke, umjesto da to vrijeme posvete pacijentima. U kontekstu sve starijeg stanovništva i povećanog opterećenja zdravstvenih djelatnika, efikasno upravljanje medicinskim uređajima postaje ključno.

Motivacija za ovaj završni rad leži u potrebi za razvojem sustava koji će omogućiti bolje praćenje stanja medicinskih uređaja, smanjujući administrativno opterećenje medicinskog osoblja i poboljšavajući ukupnu učinkovitost zdravstvene skrbi. Pri izradi sustava stavljen je poseban naglasak na jednostavnu integraciju u postojeću organizaciju bolnice te na jednostavno i upotrebljivo korisničko sučelje. Ovim pristupom osigurava se da sustav može biti lako usvojen od strane medicinskog osoblja, što dodatno doprinosi njegovoj učinkovitosti i praktičnosti u svakodnevnom radu.

Korisnici web aplikacije imaju mogućnost dodavanja mapa koji u ovom završnom radu predstavljaju različite lokacije unutar bolnice, kao što su krila, odjeli i sobe.

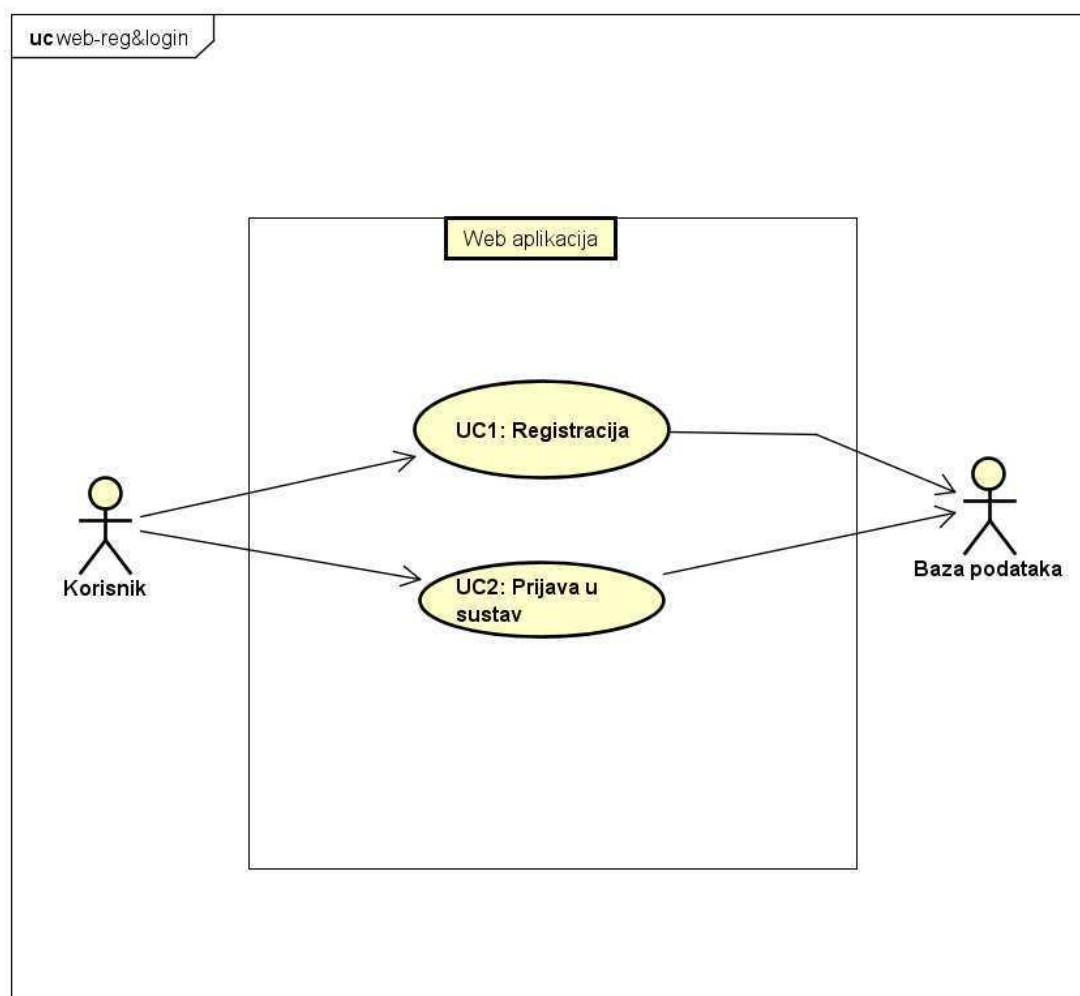
Svaka mapa može sadržavati više medicinskih uređaja, omogućavajući lako organiziranje i praćenje opreme prema lokacijama. Korisnici mogu dodijeliti medicinske uređaje određenim mapama, te pregledavati i ažurirati informacije za svaki uređaj pojedinačno. Podaci koje je moguće mijenjati uključuju status uređaja, inventarski broj, naziv uslužne tvrtke i druge relevantne informacije.

Osim web aplikacije, korisnicima je na raspolaganju i mobilna aplikacija koja omogućuje prepoznavanje barkoda na određenom uređaju. Sustav automatski registrira postoji li taj uređaj u bazi te, ako postoji, prikazuje osnovne podatke uređaja. Ako uređaj nije evidentiran u sustavu, mobilna aplikacija pruža mogućnost unosa novog uređaja u bazu.

2. Funkcionalni zahtjevi

Ovo poglavlje razrađuje sve funkcionalne zahtjeve sustava koji se sastoji od mobilne i web aplikacije te će se u tu svrhu koristiti UML dijagrami obrazaca uporabe [6].

2.1. Web aplikacija



Slika 2.1. Registracija i prijava u sustav

UC1: Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Registrirati novog korisnika u sustav, slika 2.1.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik nije prethodno registriran

Opis osnovnog tijeka:

1. Korisnik bira opciju "Sign Up".
2. Sustav prikazuje obrazac za unos podataka za registraciju.
3. Korisnik unosi potrebne podatke (ime, prezime, lozinka, e-mail).
4. Korisnik potvrđuje unos podataka.
5. Sustav provjerava unesene podatke i pohranjuje ih u bazu podataka.
6. Sustav prikazuje korisniku povratnu informaciju o uspješnosti registracije.

Opis mogućih odstupanja:

- Korisnik unosi već postojeće korisničko ime ili e-mail:
 1. Sustav obavještava korisnika da je korisničko ime ili e-mail već u upotrebi.

UC2: Prijava u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Prijaviti se u sustav kako bi se koristile ostale funkcionalnosti, slika 2.1.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prethodno registriran u sustav

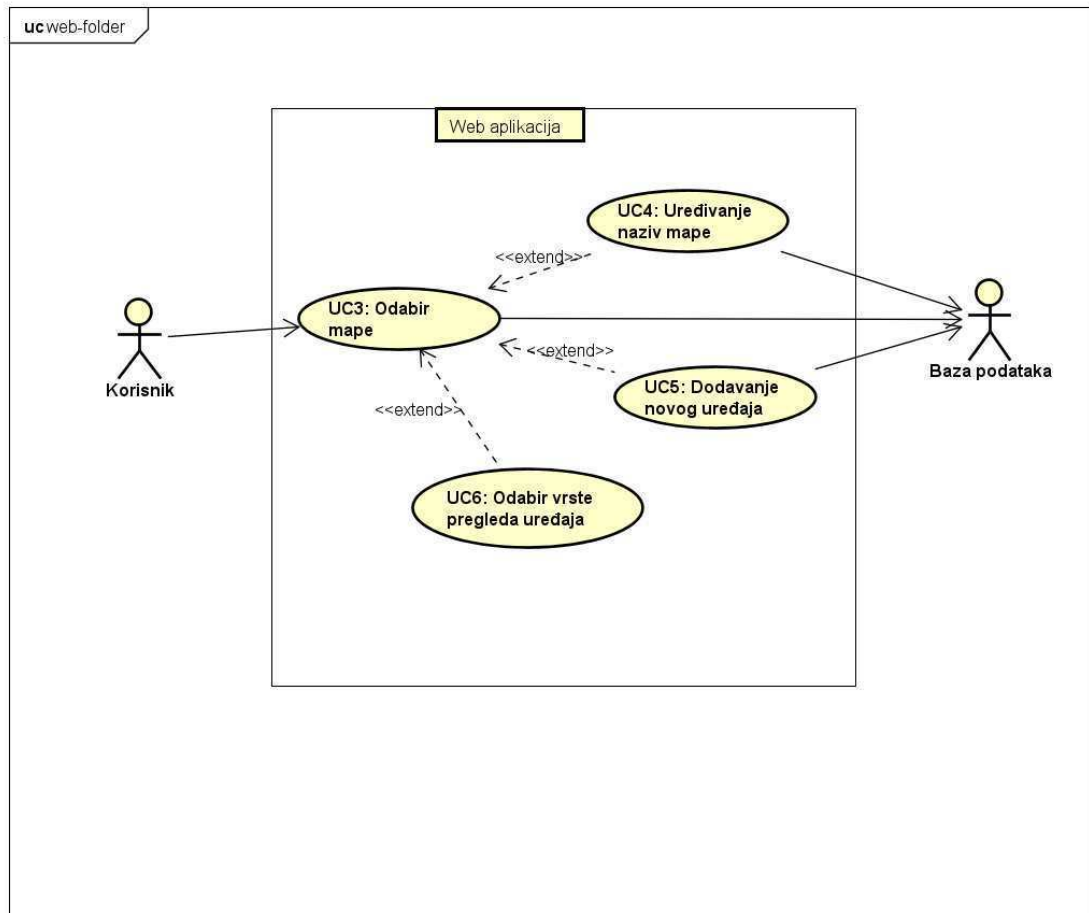
Opis osnovnog tijeka:

1. Korisnik bira opciju "Log In".
2. Sustav prikazuje obrazac za unos pristupnih podataka (email i lozinka).
3. Korisnik unosi pristupne podatke i potvrđuje unos.
4. Sustav provjerava unesene podatke s podacima u bazi podataka.
5. Sustav omogućuje korisniku pristup ostalim funkcionalnostima.

Opis mogućih odstupanja:

- Uneseni podaci nisu ispravni:

1. Sustav obavještava korisnika o neuspješnoj prijavi.



Slika 2.2. Upravljanje mapama

UC3: Odabir mape

- **Glavni sudionik:** Korisnik
- **Cilj:** Odabrati mapu kako bi se prikazali uređaji i podmape, slika 2.2.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav

Opis osnovnog tijeka:

1. Sustav prikazuje listu dostupnih mapa.
2. Korisnik bira željenu mapu.

3. Sustav prikazuje sve uređaje unutar odabrane mapa i njegovih podmapa te sve pod-mape unutar odabrane mape.

UC4: Uređivanje naziv mape

- **Glavni sudionik:** Korisnik
- **Cilj:** Urediti naziv mape, slika 2.2.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav

Opis osnovnog tijeka:

1. Korisnik odabire opciju za uređivanje mape.
2. Sustav prikazuje trenutni naziv mape i omogućava unos novog naziva.
3. Korisnik unosi novi naziv i potvrđuje promjenu.
4. Sustav sprema novi naziv mape u bazu podataka.

UC5: Dodavanje novog uređaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Dodati novi uređaj u sustav, slika 2.2.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i odabrana je određena mapa

Opis osnovnog tijeka:

1. Korisnik odabire opciju za dodavanje novog uređaja.
2. Sustav prikazuje obrazac za unos podataka o uređaju.
3. Korisnik unosi potrebne podatke o uređaju i potvrđuje unos.
4. Sustav sprema nove podatke o uređaju u bazu podataka.

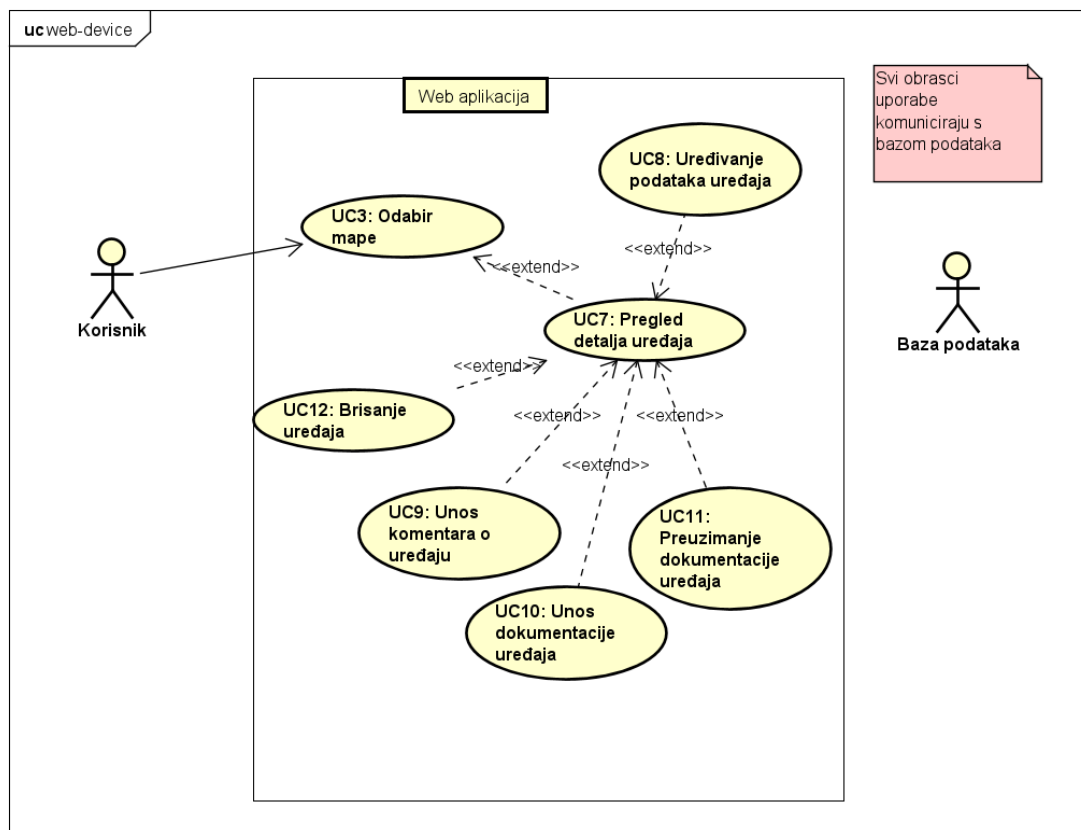
UC6: Odabir vrste pregleda uređaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Odabrati vrstu pregleda uređaja (kartični ili tablični), slika 2.2.
- **Sudionici:** /

- **Preduvjet:** Korisnik je prijavljen u sustav i odabrana je određena mapa

Opis osnovnog tijeka:

1. Sustav prikazuje dostupne opcije pregleda (kartični ili tablični).
2. Korisnik bira željenu opciju pregleda.
3. Sustav prikazuje uređaje unutar odabrane mape u odabranoj vrsti pregleda.



Slika 2.3. Pregled detalja uređaja

UC7: Pregled detalja uređaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati detaljne informacije o uređaju, slika 2.3.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i odabrao je mapu

Opis osnovnog tijeka:

1. Korisnik dvostrukim klikom odabire uređaj iz liste uređaja unutar mape.
2. Sustav prikazuje detaljne informacije o odabranom uređaju.
3. Korisnik može pregledati osnovne informacije, komentare i dostupne dokumente o uređaju.

UC8: Uređivanje podataka uređaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Urediti osnovne podatke o uređaju, slika 2.3.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i pregledava detalje uređaja

Opis osnovnog tijeka:

1. Korisnik odabire opciju za uređivanje podataka o uređaju.
2. Sustav omogućava unos novih vrijednosti za odabrane podatke.
3. Korisnik unosi nove vrijednosti i potvrđuje promjenu.
4. Sustav sprema nove podatke u bazu podataka.

UC9: Unos komentara o uređaju

- **Glavni sudionik:** Korisnik
- **Cilj:** Dodati novi komentar o uređaju, slika 2.3.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i pregledava detalje uređaja

Opis osnovnog tijeka:

1. Korisnik odabire opciju za unos novog komentara.
2. Sustav prikazuje obrazac za unos tipa komentara i samog komentara.
3. Korisnik unosi potrebne informacije i potvrđuje unos.
4. Sustav sprema novi komentar u bazu podataka.

UC10: Unos dokumentacije uređaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Dodati novu dokumentaciju za uređaj, slika 2.3.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i pregledava detalje uređaja

Opis osnovnog tijeka:

1. Korisnik odabire opciju za dodavanje nove dokumentacije.
2. Sustav prikazuje obrazac za unos datoteke, opisa i naziva dokumentacije.
3. Korisnik unosi potrebne informacije i prilaže datoteku.
4. Korisnik potvrđuje unos.
5. Sustav sprema novu dokumentaciju u bazu podataka.

UC11: Preuzimanje dokumentacije uređaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Preuzeti dokumentaciju za uređaj, slika 2.3.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i pregledava detalje uređaja

Opis osnovnog tijeka:

1. Korisnik pregledava dostupnu dokumentaciju u tablici.
2. Korisnik dvostrukim klikom odabire željenu datoteku.
3. Sustav omogućava preuzimanje odabrane datoteke.

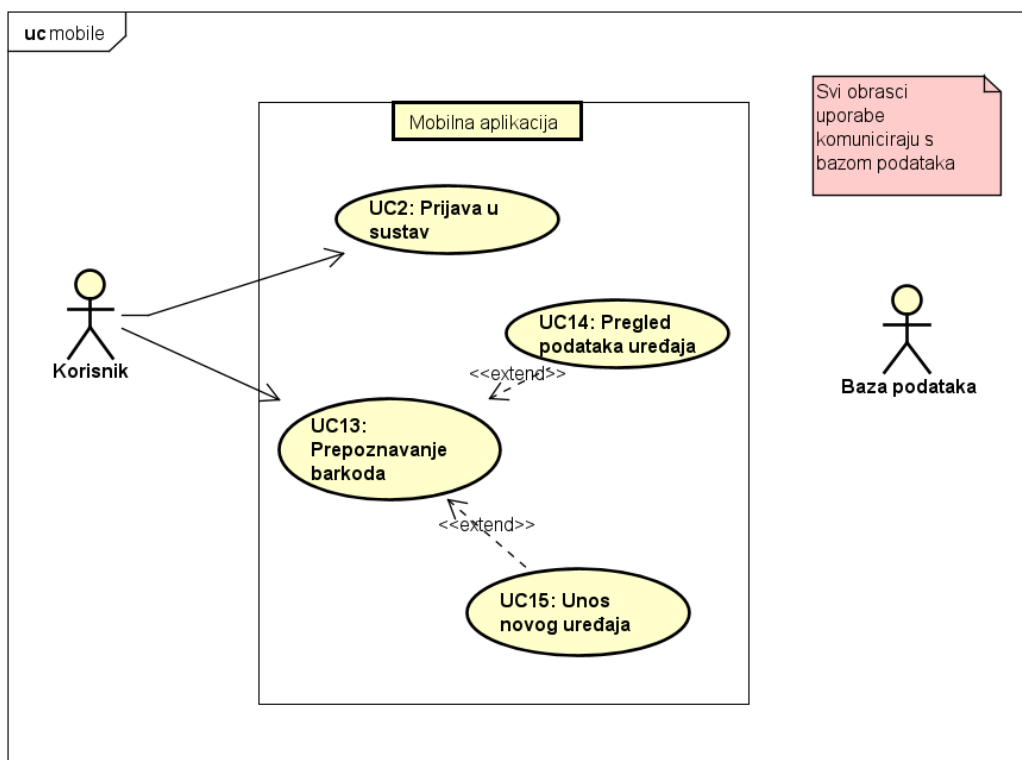
UC12: Brisanje uređaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Brisanje uređaja iz sustava, slika 2.3.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i pregledava detalje uređaja

Opis osnovnog tijeka:

1. Korisnik odabire opciju za brisanje uređaja.
2. Korisnik pregledava dostupnu dokumentaciju u tablici.
3. Korisnik dvostrukim klikom odabire željenu datoteku.
4. Sustav briše uređaj.

2.2. Mobilna aplikacija



Slika 2.4. Obrazac uporabe za mobilnu aplikaciju

UC13: Prepoznavanje barkoda

- **Glavni sudionik:** Korisnik
- **Cilj:** Prepoznati barkod uređaja, slika 2.4.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav

Opis osnovnog tijeka:

1. Korisnik odabire opciju za prepoznavanje barkoda.
2. Sustav aktivira kameru uređaja.

3. Korisnik usmjerava kameru prema barkodu uređaja.
4. Sustav prepoznaje barkod i šalje podatke na obradu.
5. Sustav provjerava postoji li uređaj s tim barkodom u bazi podataka.
6. Ako uređaj postoji, sustav prikazuje osnovne podatke o uređaju (UC14).
7. Ako uređaj ne postoji, sustav nudi mogućnost unosa novog uređaja (UC15).

UC14: Pregled podataka uređaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati osnovne podatke o uređaju, slika 2.4.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i uređaj s barkodom postoji u bazi podataka

Opis osnovnog tijeka:

1. Sustav automatski prikazuje osnovne podatke o uređaju (ime uređaja, ime mape u kojoj se nalazi uređaj, status uređaja, inventarski broj) nakon prepoznavanja barkoda.
2. Korisnik pregledava prikazane podatke.

UC15: Unos novog uređaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Unijeti novi uređaj u sustav, slika 2.4.
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i uređaj s barkodom ne postoji u bazi podataka

Opis osnovnog tijeka:

1. Sustav nudi opciju za unos novog uređaja nakon prepoznavanja barkoda koji ne postoji u bazi podataka.
2. Korisnik unosi potrebne podatke o uređaju.
3. Korisnik potvrđuje unos.

4. Sustav sprema nove podatke o uređaju u bazu podataka.

3. Arhitektura i dizajn sustava

Sustav je organiziran u četiri glavna dijela: baza podataka, poslužiteljska strana, klijentska strana web aplikacije i klijentska strana mobilne aplikacije.

Baza podataka koristi PostgreSQL za pohranu svih podataka potrebnih za rad sustava.

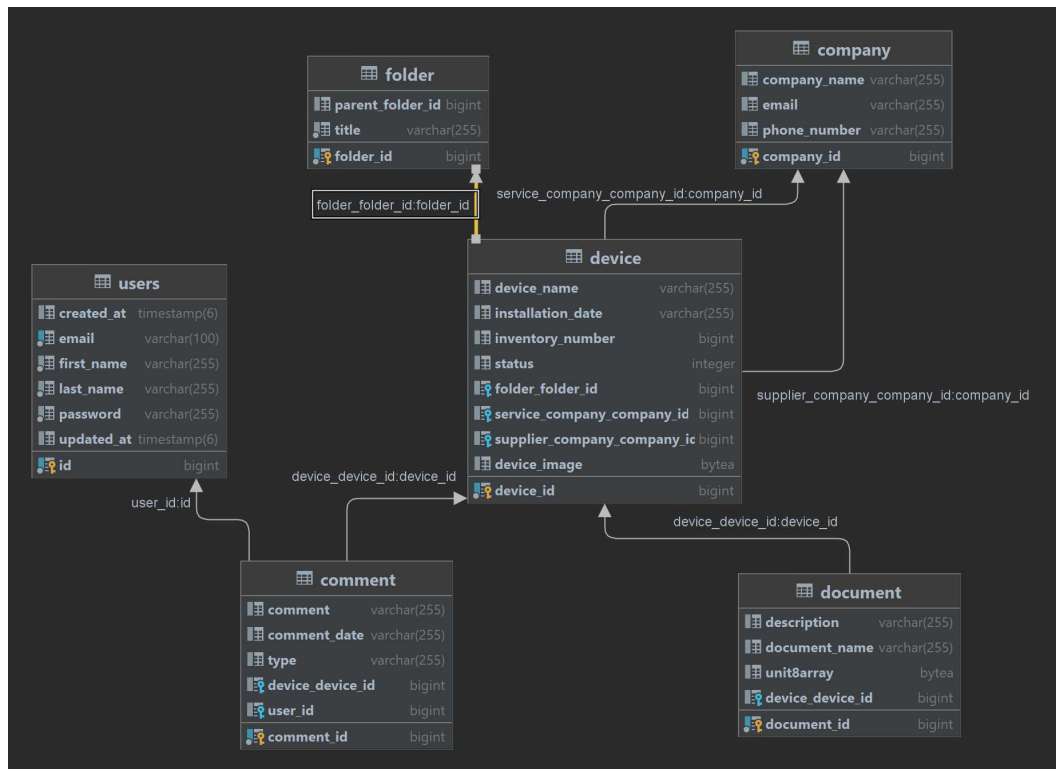
Poslužiteljska strana je implementirana korištenjem radnog okvira Spring Boot s programskim jezikom Java. Također, koristi se višeslojna arhitektura koja se sastoji od kontrolera, servisa i repozitorija.

Klijentska strana web aplikacije razvijena je pomoću radnog okvira Next.js s programskim jezikom TypeScript. Web aplikacija je izgrađena od komponenti, a za preusmjeravanje koristi se `Next.js routing` [2]. Komunikacija s poslužiteljem ostvaruje se putem biblioteke Axios [5].

Klijentska strana mobilne aplikacija razvijena je korištenjem radnog okvira Expo s programskim jezikom TypeScript. Organizirana je u komponentama, a za preusmjeravanje se koristi `Expo routing` [3]. Komunikacija s poslužiteljem ostvarena je također putem biblioteke Axios [5].

3.1. Baza podataka

3.1.1. Dijagram baze podataka



Slika 3.1. Dijagram baze podataka

Slika 3.1. prikazuje dijagram baze podataka sustava.

3.1.2. Opis tablica

Tablica `users`

Tablica `users` pohranjuje sve podatke o registriranim korisnicima. Podaci uključuju:

- `id`: Jedinstveni identifikator za svakog korisnika (primarni ključ).
- `created_at`: Datum i vrijeme kada je korisnik dodan u sustav.
- `email`: E-mail adresa korisnika.
- `first_name`: Ime korisnika.
- `last_name`: Prezime korisnika.
- `password`: Lozinka korisnika, koja je pohranjena u šifriranom obliku radi sigurnosti.

- `updated_at`: Datum i vrijeme kada su podaci o korisniku zadnji put ažurirani.

Odnosi

- `users -> comment`: Jedan korisnik može imati više komentara (One to Many). `id` iz tablice `users` je povezan s `user_id` u tablici `comment` (strani ključ).

Tablica `folder`

Tablica `folder` pohranjuje sve podatke o mapama koje se koriste za organizaciju uređaja. Podaci uključuju:

- `folder_id`: Jedinstveni identifikator za svaku mapu (primarni ključ).
- `parent_folder_id`: Identifikator nadređene mape, omogućuje hijerarhijsku strukturu mapa (strani ključ).
- `title`: Naslov ili ime mape.

Odnosi

- `folder->device`: Jedna mapa može sadržavati više uređaja (One to Many). `folder_id` iz tablice `folder` je povezan s `folder_folder_id` u tablici `device` (strani ključ).

Tablica `company`

Tablica `company` pohranjuje sve podatke o tvrtkama koje su uključene kao dobavljači ili servisne tvrtke za uređaje. Podaci uključuju:

- `company_id`: Jedinstveni identifikator za svaku tvrtku (primarni ključ).
- `company_name`: Naziv tvrtke.
- `email`: E-mail adresa tvrtke.
- `phone_number`: Broj telefona tvrtke.

Odnosi

- `company -> device`: Tvrtka može biti servisna tvrtka ili dobavljač uređaja (One to Many). `company_id` iz tablice `company` je povezan sa `service_company_`

`company_id` i `supplier_company_company_id` u tablici `device` (strani ključevi).

Tablica `device`

Tablica `device` pohranjuje sve podatke o uređajima koji su praćeni u sustavu. Podaci uključuju:

- `device_id`: Jedinstveni identifikator za svaki uređaj (primarni ključ).
- `device_name`: Naziv uređaja.
- `installation_date`: Datum instalacije uređaja.
- `inventory_number`: Inventarni broj uređaja, iščita se iz barkoda.
- `status`: Status uređaja.
- `folder_folder_id`: Identifikator mape u kojoj se uređaj nalazi (strani ključ).
- `service_company_company_id`: Identifikator servisne tvrtke odgovorne za uređaj (strani ključ).
- `supplier_company_company_id`: Identifikator tvrtke koja je dobavljač potrošnog materijala uređaja (strani ključ).
- `device_image`: Slika uređaja pohranjena u binarnom formatu.

Odnosi

- `device -> comment`: Jedan uređaj može imati više komentara (One to Many).
`device_id` iz tablice `device` je povezan s `device_device_id` u tablici `comment` (strani ključ).
- `device -> document`: Jedan uređaj može imati više dokumenata (One to Many).
`device_id` iz tablice `device` je povezan s `device_device_id` u tablici `document` (strani ključ).

Tablica `document`

Tablica `document` pohranjuje sve podatke o dokumentima koji se odnose na određene uređaje. Podaci uključuju:

- `document_id`: Jedinstveni identifikator za svaki dokument (primarni ključ).

- `description`: Opis dokumenta.
- `document_name`: Naziv dokumenta.
- `unit8array`: Sadržaj dokumenta u binarnom formatu.
- `device_device_id`: Identifikator uređaja na koji se dokument odnosi (strani ključ).

Odnosi

- `document -> device`: Jedan dokument je povezan s jednim uređajem (Many to One). `device_device_id` iz tablice `document` je povezan s `device_id` u tablici `device` (strani ključ).

Tablica `comment`

Tablica `comment` pohranjuje sve podatke o komentarima koji su dodani za određene uređaje. Podaci uključuju:

- `comment_id`: Jedinstveni identifikator za svaki komentar (primarni ključ).
- `comment`: Sadržaj komentara.
- `comment_date`: Datum kada je komentar dodan.
- `type`: Tip komentara.
- `device_device_id`: Identifikator uređaja na koji se komentar odnosi (strani ključ).
- `user_id`: Identifikator korisnika koji je dodao komentar (strani ključ).

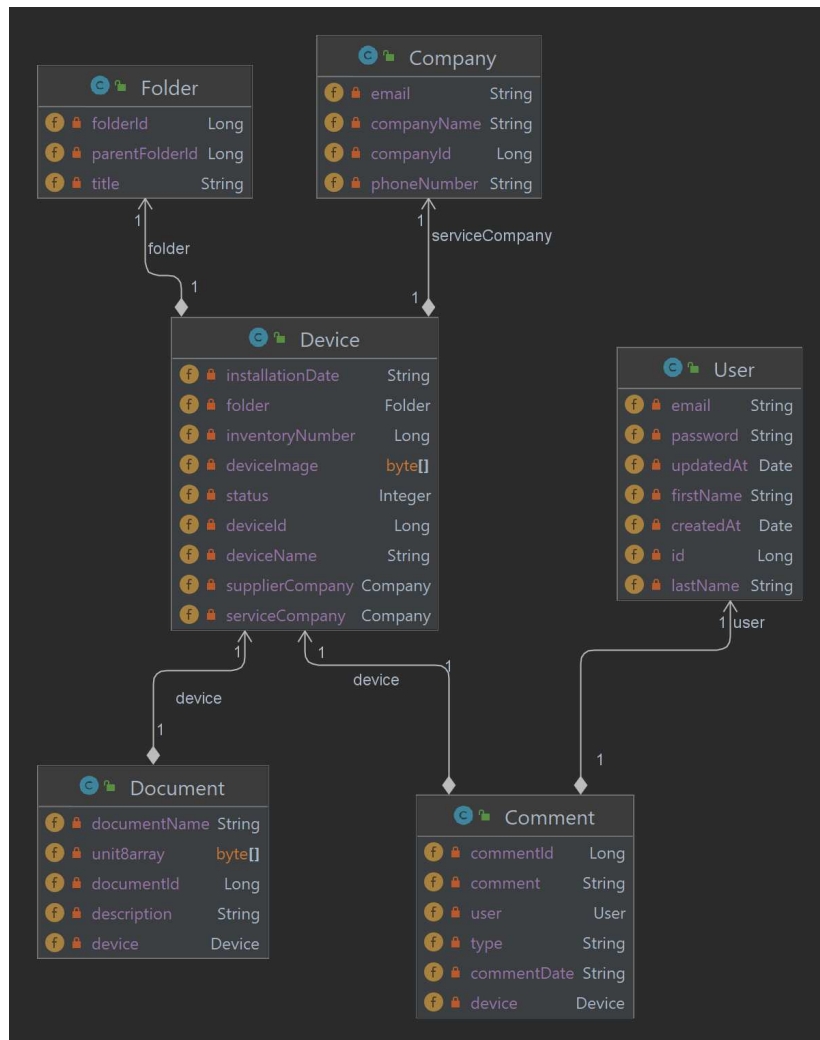
Odnosi

- `comment -> device`: Jedan komentar je povezan s jednim uređajem (Many to One). `device_device_id` iz tablice `comment` je povezan s `device_id` u tablici `device` (strani ključ).
- `comment -> users`: Jedan komentar je povezan s jednim korisnikom (Many to One). `user_id` iz tablice `comment` je povezan s `id` u tablici `users` (strani ključ).

3.2. Poslužiteljska strana

3.2.1. Dijagram razreda

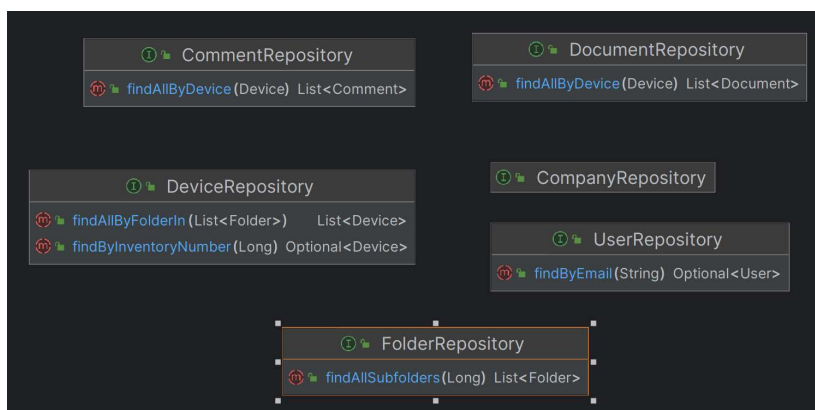
Entiteti



Slika 3.2. Dijagram razreda entiteta

Slika 3.2. prikazuje klase unutar aplikacije izrađene radnim okvirom Spring Boot koje su mapirane na entitete iz modela relacijske baze podataka. Svaka klasa na dijagramu odgovara specifičnoj tablici unutar baze, dok su veze među tablicama reprezentirane atributima koji referenciraju druge klase. Za mapiranje entiteta iz relacijske baze u objekte unutar aplikacije koristi se Jakarta Persistence API (JPA), što omogućava transformaciju relacijskog modela u objektno-orijentirani model.

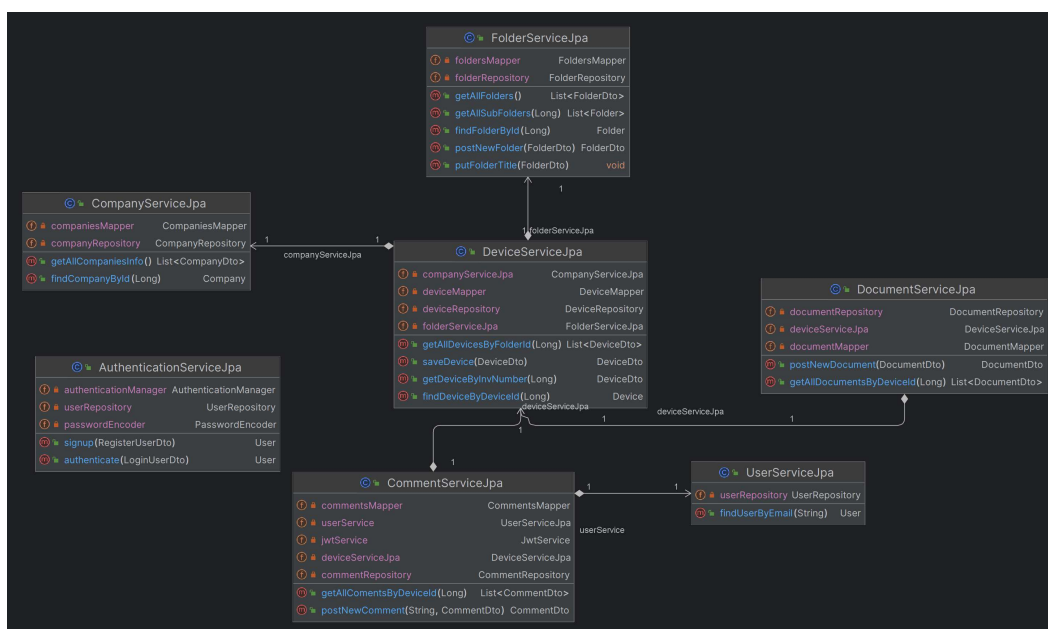
Repozitoriji



Slika 3.3. Dijagram razreda repozitorija

Slika 3.3. prikazuje repozitorije sučelja koji se koriste za rad na određenoj tablici za koju je to sučelje odgovorno. Svako sučelje prikazano na slici nasljeđuje generičko sučelje `JpaRepository`, koje zahtijeva dva parametra: tip entiteta (T) i tip primarnog ključa (ID). `JpaRepository` pruža osnovne operacije CRUD (*Create, Read, Update, Delete*).

Servisi

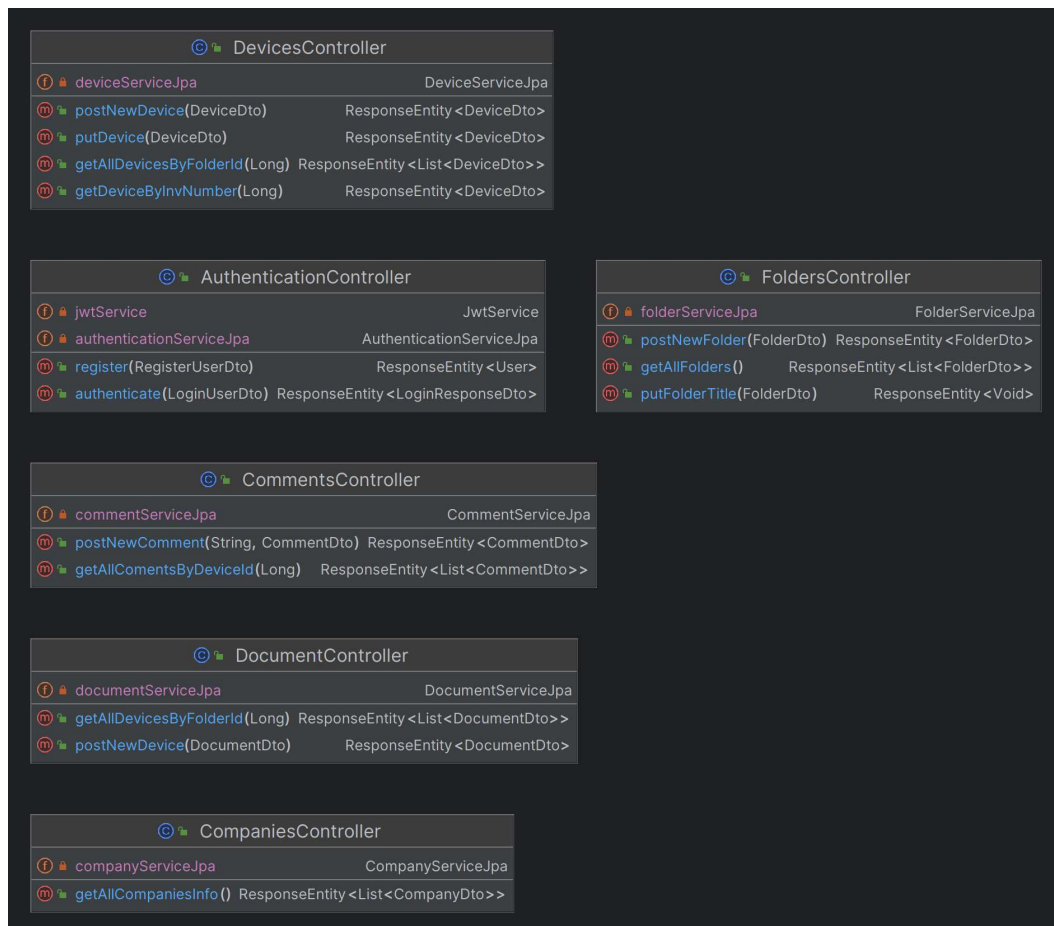


Slika 3.4. Dijagram razreda servisa

Klase servisa, prikazane na slici 3.4., služe kao posrednici između kontrolera i repozitorija, pružajući poslovnu logiku potrebnu za obradu podataka prije pohrane ili prikaza

korisniku. One koriste sučelja repozitorija za dohvaćanje, spremanje, ažuriranje i brisanje podataka. Klase servisa mogu se međusobno koristiti, čime se omogućuje ponovna upotreba koda i razdvajanje odgovornosti.

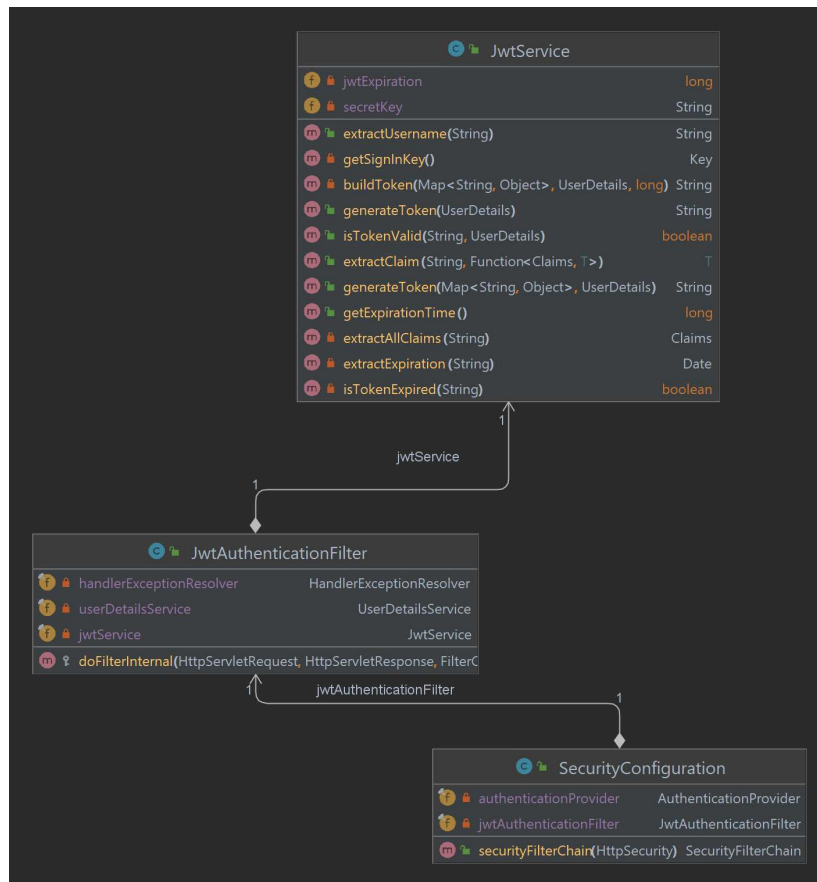
Kontroleri



Slika 3.5. Dijagram razreda kontrolera

Slika 3.5. prikazuje klase kontrolera koje služe kao središnja točka za obradu svih zahtjeva HTTP-om u aplikaciji. One definiraju krajnje točke (engl. *endpoint*) na koje korisnici mogu slati zahtjeve. Oko tih krajnjih točaka omata se radni okvir Spring Security (koji će se u nastavku detaljnije razraditi) kako bi se osiguralo da samo ovlašteni korisnici imaju pristup određenim resursima. Za komunikaciju između poslužitelja i klijenta koriste se objekti DTO (engl. *Data Transfer Object*), koje Spring Boot automatski mapira u objekte u JSON-u i obratno, olakšavajući razmjenu podataka u formatu razumljivom za obje strane.

3.2.2. Sigurnost



Slika 3.6. Dijagram razreda za sigurnost

JwtService

`JwtService`, slika 3.6., je glavna komponenta za rad s tokenima JWT (engl. *JSON Web Token*) [4]. Ova klasa upravlja generiranjem, validacijom i ekstrakcijom informacija iz tokena JWT. JWT se koristi za sigurno prenošenje informacija između klijenta i poslužitelja. Kada se korisnik prijavi, poslužitelj generira token JWT i šalje ga klijentu, koji zatim koristi taj token za autentifikaciju pri svakom zahtjevu prema zaštićenim resursima na poslužitelju.

Klasa `JwtService` koristi tajni ključ (`secretKey`) za potpisivanje tokena i određuje vrijeme valjanosti tokena (`jwtExpiration`). Klasa generira tokene JWT s dodatnim tvrdnjama (`claims`), koje uključuju korisničko ime i vrijeme izdavanja. Funkcija `generateToken` omogućava generiranje tokena s dodatnim tvrdnjama ili bez njih. Metode za validaciju tokena provjeravaju autentičnost i valjanost tokena, uključujući pro-

vjeru potpisa i vremena isteka. Funkcija `extractClaim` omogućava ekstrakciju specifičnih tvrdnji iz tokena.

JwtAuthenticationFilter

`JwtAuthenticationFilter`, slika 3.6., je filtar koji presreće zahtjeve HTTP-om prema aplikaciji. Ovaj filtar provjerava prisutnost tokena JWT u svakom dolaznom zahtjevu. Ako je token prisutan i valjan, filtar omogućava pristup resursu; ako nije, pristup je odbijen. Također, filtar se dodaje u sigurnosni lanac aplikacije i na taj način štiti od neovlaštenih pristupa.

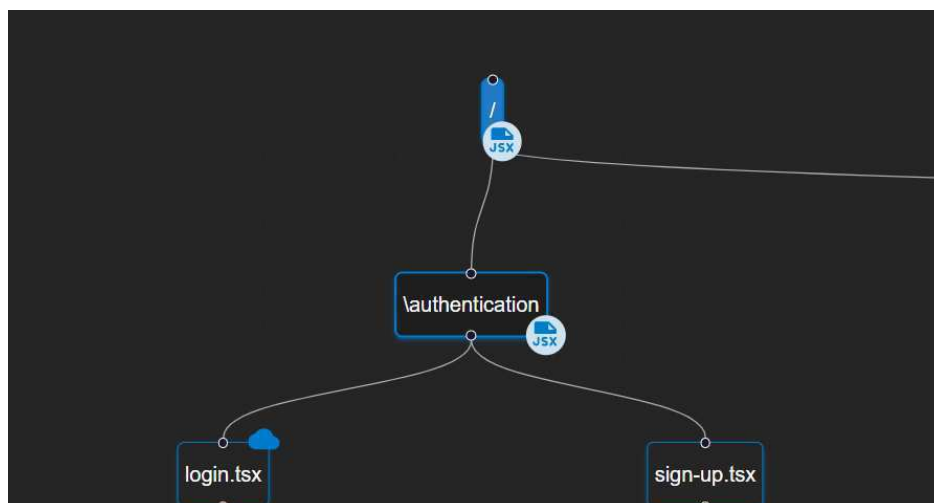
SecurityConfiguration

`SecurityConfiguration`, slika 3.6., je konfiguracijska klasa koja definira sigurnosne postavke aplikacije. Ova klasa integrira `JwtAuthenticationFilter` u sigurnosni lanac aplikacije i postavlja pravila autentifikacije i autorizacije. `SecurityConfiguration` osigurava da se svaki zahtjev prema aplikaciji provjeri i validira prije nego što dobije pristup zaštićenim resursima.

3.3. Klijentska strana web aplikacije

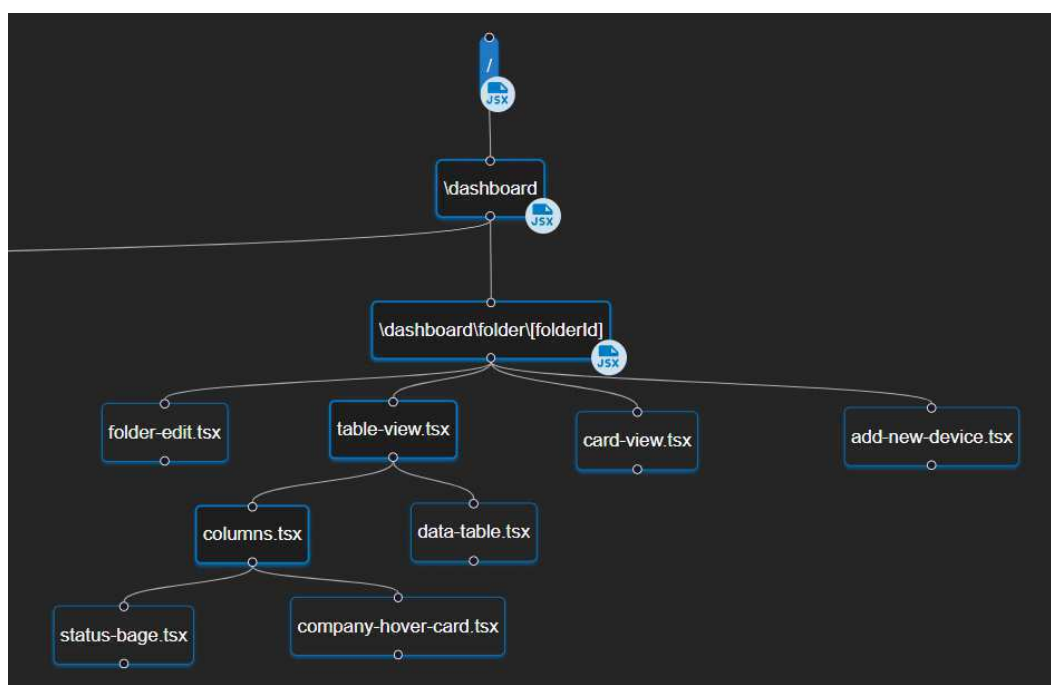
3.3.1. Dijagram komponenti

U ovom potpoglavlju razradit ćemo dijagrame komponenti klijentske strane web aplikacije. Polja koja započinju s prefiksom `/"` predstavljaju rute u pregledniku. Te rute definiraju navigacijske putanje koje korisnici mogu slijediti unutar aplikacije. S druge strane, polja koja nemaju prefiks `/"` odnose se na konkretne komponente, odnosno implementacije određenih funkcionalnosti unutar aplikacije.



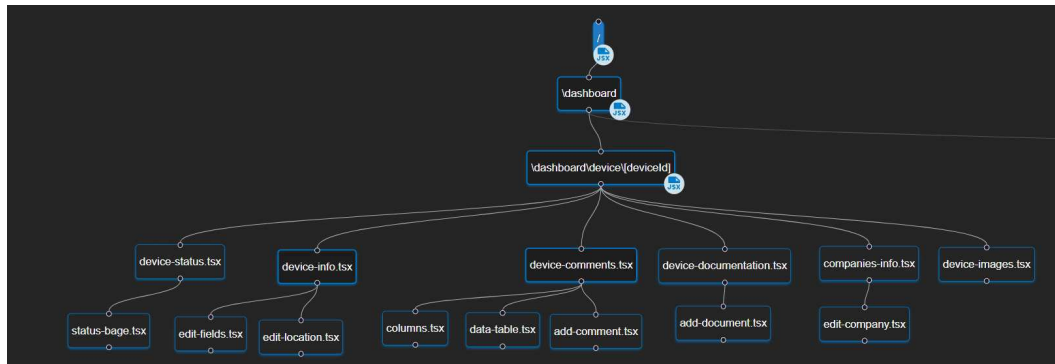
Slika 3.7. Dijagram komponenti za prijavu i registraciju

Slika 3.7. prikazuje strukturu komponenti vezanih za autentifikaciju korisnika u sustavu. Ruta `"/authentication"` predstavlja ulaznu točku za neprijavljene korisnike, gdje mogu pristupiti funkcionalnostima prijave, koja je predstavljena komponentom `login.tsx`, ili registracije u sustav, koja je predstavljena komponentom `sign-up.tsx`.



Slika 3.8. Dijagram komponenti za nadzornu ploču

Slika 3.8. prikazuje strukturu komponenti vezanih za nadzornu pluću (engl. *dashboard*). Nakon odabira određene mape na ruti `"/dashboard/folder/[folderId]"`, pored već spomenutih funkcionalnosti, korisnik može pregledavati sve uređaje u odabranoj mapi u kartičnom ili tabličnom prikazu.



Slika 3.9. Dijagram komponenti za prikaz detalja uređaja

Slika 3.9. prikazuje strukturu komponenti vezanih za prikaz detalja uređaja. Nakon što korisnik odabere detaljniji pregled podataka za određeni uređaj, odlazi na rutu `/dashboard/device/[deviceId]`. Na toj ruti prikazuju se sve informacije o uređaju, uključujući komentare uređaja i dostupnu dokumentaciju.

3.3.2. Sigurnost

Sigurnost na klijentskoj strani može se podijeliti u dva ključna dijela. Prvi dio se odnosi na osiguravanje ruta unutar aplikacije, gdje neprijavljeni korisnik smije pristupiti samo rutama za prijavu i registraciju u sustav. Ovo osigurava da su sve ostale rute zaštićene i dostupne samo ovlaštenim korisnicima. Drugi dio sigurnosti na klijentskoj strani uključuje autentifikaciju svakog zahtjeva prema poslužitelju, pri čemu klijentska strana mora u zaglavlju Authorization staviti odgovarajući JWT koji je generiran tijekom prijave u sustav.

Osiguravanje ruta u aplikaciji

```
export default async function middleware(req: NextRequest) {
  const cookie = cookies().get('session')?.value
  const session = await decrypt(cookie)

  if (!session?.userId && protectedRoutes.includes(req.nextUrl
    .pathname)) {
    const absoluteURL = new URL("/", req.nextUrl.origin);
    return NextResponse.redirect(absoluteURL.toString());
  }
}
```

Izvorni kod 3..1 Funkcija middleware

Za osiguranje ruta u aplikaciji koristi se `middleware`, izvorni kod 3..1, koji Next.js nudi zajedno s kolačićima (engl. *cookies*) [1]. Middleware omogućuje presretanje zah-
tjeva na razini poslužitelja prije nego što stignu do stvarnih ruta, a koristi se za pro-
vjeru je li korisnik prijavljen prije nego što mu se dozvoli pristup zaštićenim rutama.
Middleware provjerava postoji li valjan kolačić `session` koji sadrži podatke o koris-
niku. Ako kolačić postoji, dekriptira se kako bi se izvukli podaci o korisničkoj sjednici.
Ako korisnik nema valjan kolačić `session` ili nije prijavljen, bit će preusmjeren na po-
četnu stranicu.

Autentifikacija zahtjeva prema poslužitelju

```
const addTokenToHeaders = () => {
  const token = localStorage.getItem("token");

  if (token) {
    api.defaults.headers.common["Authorization"] = `Bearer ${token}`;
  } else {
    delete api.defaults.headers.common["Authorization"];
  }
};

export const ApiService = {
  get: async (url: string) => {
    addTokenToHeaders();
    return await api.get(url);
  },

  post: async (url: string, data: any) => {
    addTokenToHeaders();
    return await api.post(url, data);
  },

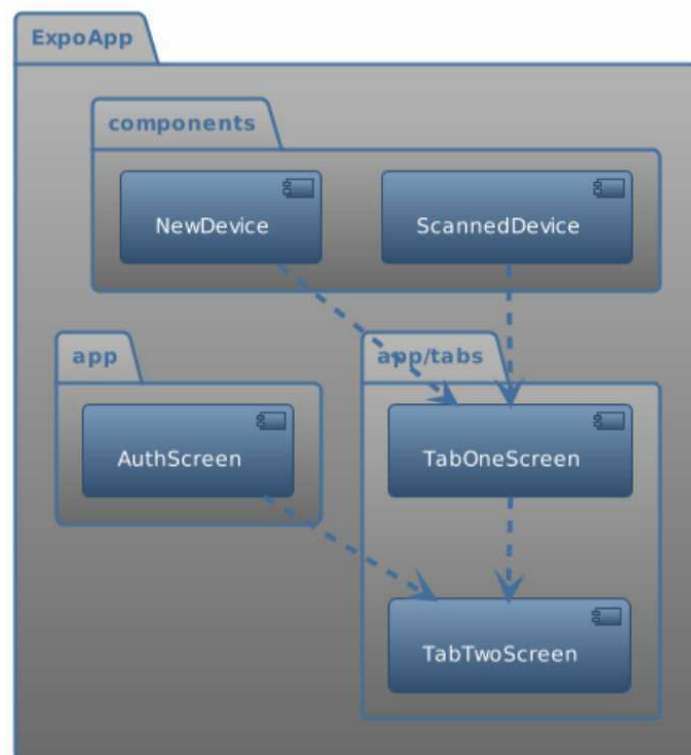
  put: async (url: string, data: any) => {
    addTokenToHeaders();
    return await api.put(url, data);
  },

  delete: async (url: string) => {
    addTokenToHeaders();
    return await api.delete(url);
  },
};
```

Za autentifikaciju zahtjeva prema poslužitelju koristi se Axios, koji je instanciran u varijabli `api`. Funkcija `addTokenToHeaders`, izvorni kod 3..2, provjerava postoji li token u lokalnoj pohrani i ako postoji, dodaje ga u zaglavlje `Authorization`. Unutar objekta `ApiService` definirane su funkcije CRUD (`get`, `post`, `put`, `delete`) koje pozivaju `addTokenToHeaders` prije samog slanja zahtjeva na poslužitelj. Ovaj pristup osigurava da svi zahtjevi automatski uključuju potrebni token `Authorization`.

3.4. Klijentska strana mobilne aplikacije

3.4.1. Dijagram komponenti



Slika 3.10. Dijagram komponenti mobilne aplikacije

Slika 3.10. prikazuje dijagram komponenti mobilne aplikacije izrađene u radnom okviru Expo. Aplikacija se sastoji od nekoliko glavnih komponenti grupiranih unutar dva direktorija: `components` i `app`.

Komponenta `AuthScreen` prikazuje početni ekran za prijavljivanje u sustav. Ako korisnik nije prijavljen, ne može pristupiti ostatku mobilne aplikacije. Komponenta `TabTwoScreen` automatski otvara kameru mobitela i čeka da se prepozna barkod. Nakon što je barkod prepoznat, komponenta `TabOneScreen` registrira prepoznavanje barkoda i, ovisno o tome je li prepoznati barkod već od prije u sustavu ili nije, prikazuje komponentu `NewDevice` ili komponentu `ScannedDevice`.

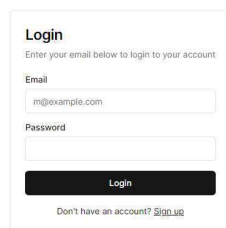
Komponenta `NewDevice` nudi unos podataka o novom uređaju u slučaju kada barkod ne postoji od prije u sustavu. S druge strane, komponenta `ScannedDevice` prikazuje osnovne podatke o uređaju s prepoznatim barkodom.

3.4.2. Sigurnost

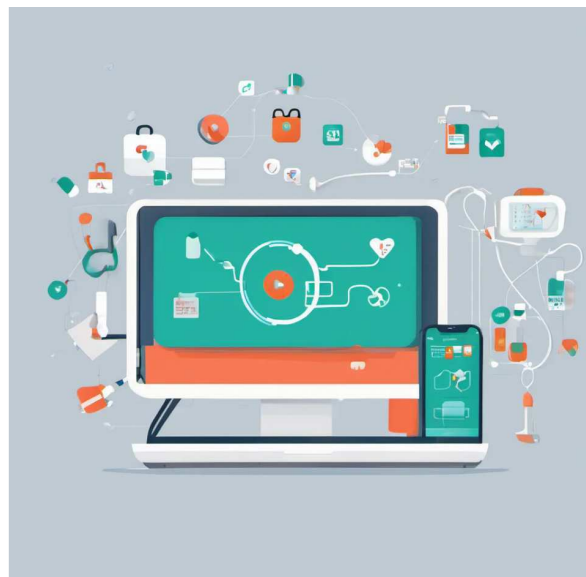
U mobilnoj aplikaciji, sigurnost je ostvarena na sličan način kao i na klijentskoj strani web aplikacije. Neprijavljenom korisniku prikazuje se samo prozor za prijavu. Svaki zahtjev prema poslužitelju automatski uključuje token `Authorization` koji je generiran tijekom prijave korisnika. Ova funkcionalnost implementirana je korištenjem biblioteke `Axios` isto kao i kod web aplikacije.

4. Implementacija i korisničko sučelje

4.1. Korisničko sučelje web aplikacije



A login form mockup with a white background and a thin grey border. It features the title "Login" in bold, followed by the instruction "Enter your email below to login to your account." Below this are two input fields: "Email" with the placeholder text "m@example.com" and "Password". A black "Login" button is positioned below the password field. At the bottom, there is a link that says "Don't have an account? [Sign up](#)".



Slika 4.1. Web aplikacija - Prijava u sustav

Sign Up

Enter your information to create an account

First name

Last name

Max

Robinson

Email

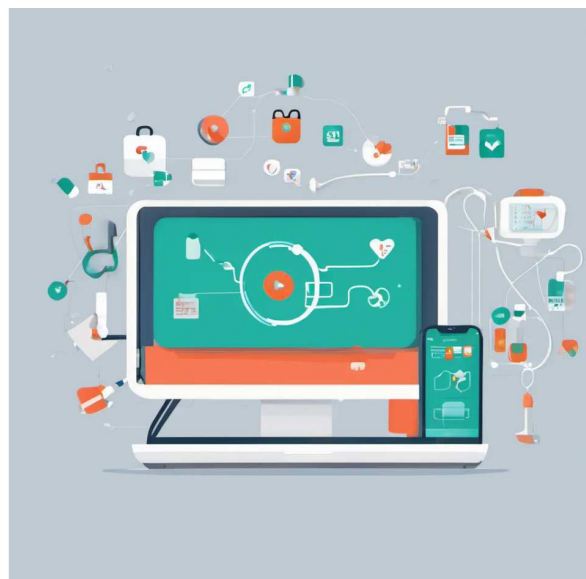
m@example.com

Password

Create an account

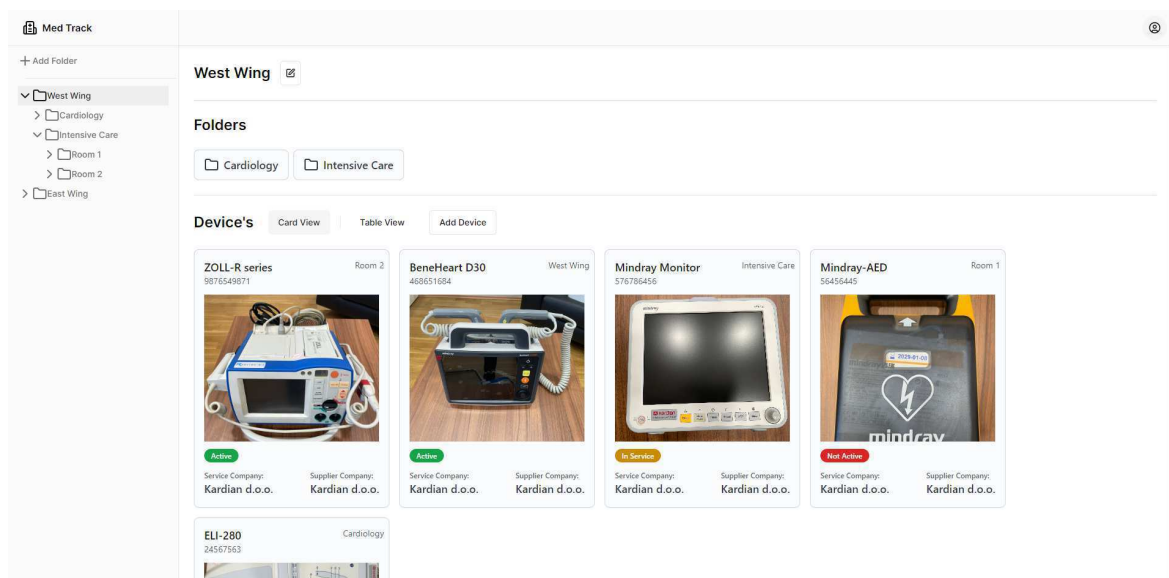
Already have an account?

Sign in



Slika 4.2. Registracija

Slike 4.1. i 4.2. prikazuju dva različita obrasca web aplikacije. Slika 4.1. prikazuje obrazac za prijavu u sustav, gdje korisnici unose svoju e-mail adresu i lozinku kako bi se prijavili. Slika 4.2. prikazuje obrazac za registraciju, gdje korisnici unose svoje ime, prezime, e-mail adresu i lozinku kako bi kreirali novi korisnički račun.



Slika 4.3. Kartični pregled

Med Track

+ Add Folder

- West Wing
 - Cardiology
 - Intensive Care
 - Room 1
 - Room 2
 - East Wing

West Wing

Folders

Cardiology Intensive Care

Device's Card View Table View Add Device

Q Search Device Name Search Device Name Filter by Status

Device Name	Inventory Number	Folder Name	Status	Service Company	Supplier Company
	9876549871	Room 2	Active	Kardian d.o.o.	Kardian d.o.o.
BeneHeart D30	468651684	West Wing	Active	Kardian d.o.o.	Kardian d.o.o.
Mindray Monitor	576786456	Intensive Care	In Service	Kardian d.o.o.	Kardian d.o.o.
Mindray-AED	56456445	Room 1	Not Active	Kardian d.o.o.	Kardian d.o.o.
ELI-280	24567563	Cardiology	Active	Kardian d.o.o.	Kardian d.o.o.

Slika 4.4. Tablični pregled

Slike 4.3. i 4.4. prikazuju dvije različite opcije pregleda uređaja u odabranoj mapi unutar sustava. Slika 4.3. prikazuje kartični pregled uređaja, dok slika 4.4. prikazuje tablični pregled uređaja.

Med Track

+ Add Folder

- West Wing
 - Cardiology
 - Intensive Care
 - Room 1
 - Room 2
 - East Wing

West Wing

Folders

Cardiology Intensive Care

Device's Card View Table View Add D

Q Search Device Name Search Device Name

Device Name Inventory Number

ZOLL-R series 9876549871

BeneHeart D30 468651684

Mindray Monitor 576786456

Mindray-AED 56456445

ELI-280 24567563

Add New Device

Enter information about the device.

Device Name

Inventory Number

Location West Wing

Installation Date 06/09/2024

Device Image Choose File No file chosen

Status Select status

Supplier Company Select Supplier Company

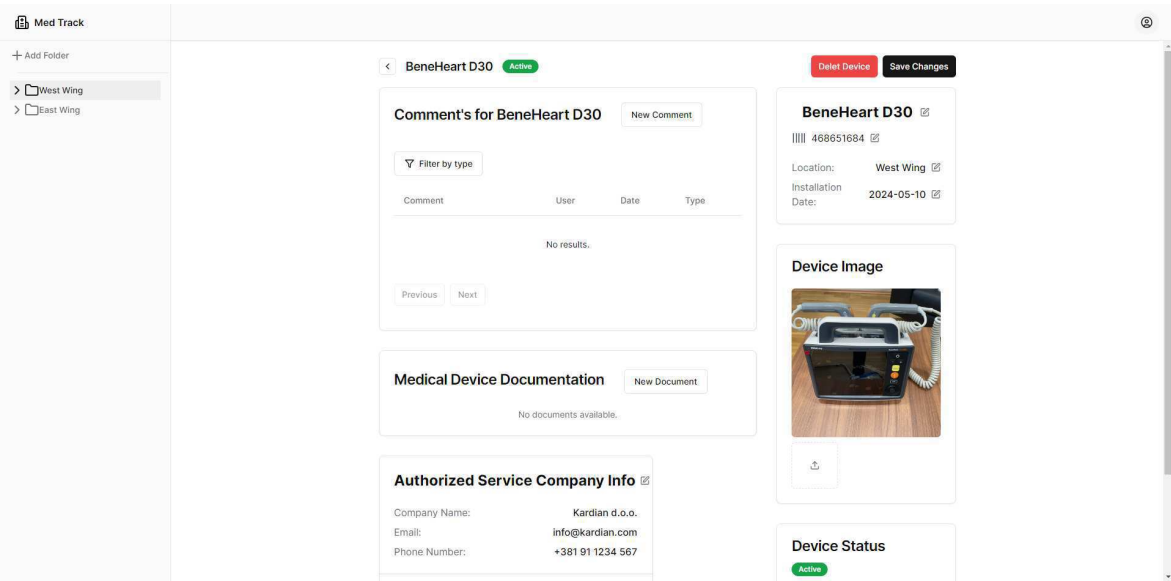
Service Company Select Service Company

Save changes

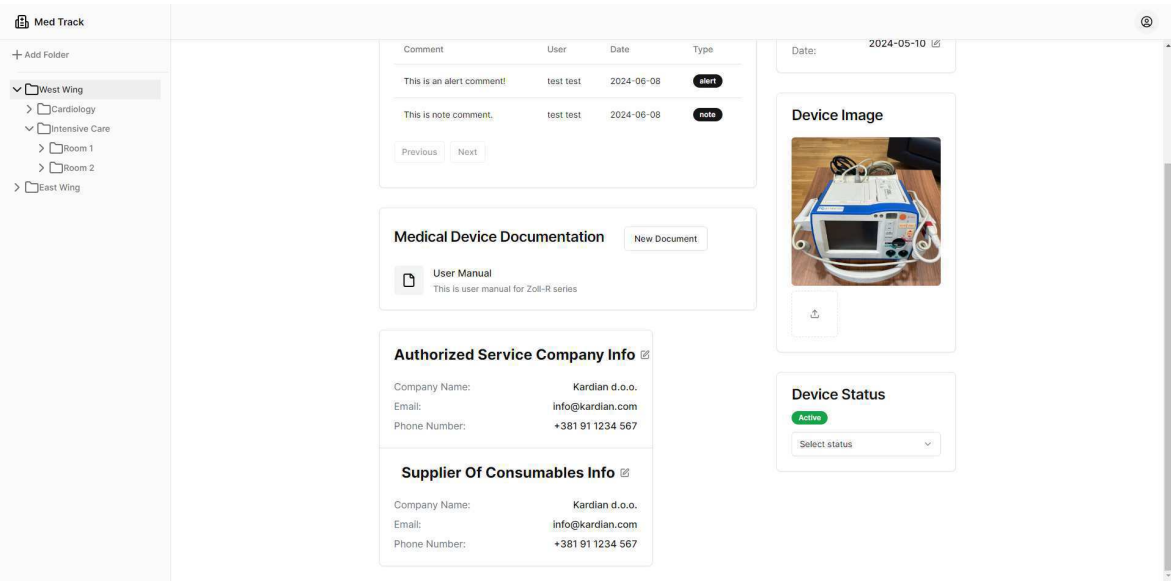
Slika 4.5. Dodavanje novog uređaja

Slika 4.5. prikazuje obrazac za unos novog uređaja u sustav. Korisnici mogu unijeti detaljne informacije o uređaju, uključujući naziv uređaja, inventarni broj, lokaciju, datum postavljanja uređaja, sliku uređaja, status uređaja, dobavljača potrošnog materijala i ovlašteni servis. Nakon popunjavanja svih potrebnih polja, korisnici mogu spremiti

promjene kako bi dodali novi uređaj u sustav.



Slika 4.6. Detaljan pregled uređaja 1.

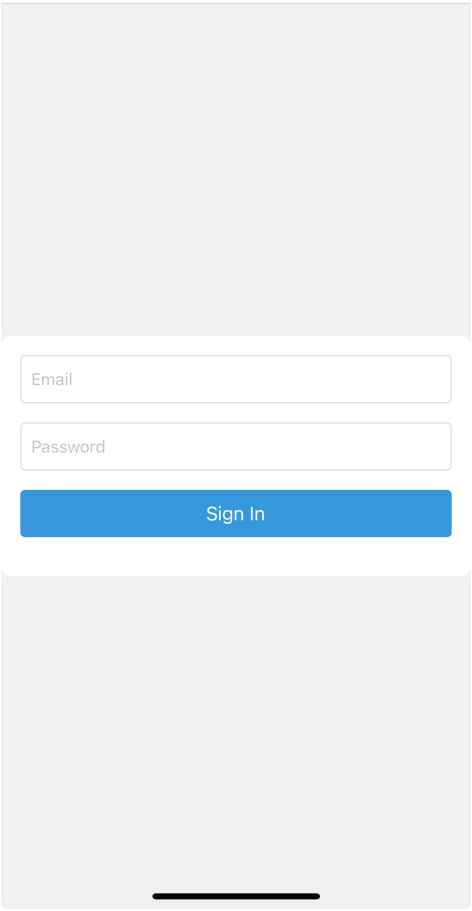


Slika 4.7. Detaljan pregled uređaja 2.

Slike 4.6. i 4.7. prikazuju detaljan pregled uređaja, pružajući korisnicima mogućnost izmjene pojedinačnih informacija uređaja. Korisnici mogu dodati komentare, učitati dokumentaciju te pregledavati informacije o ovlaštenim servisnim tvrtkama i dobavljačima potrošnog materijala. Također, postoji opcija brisanja uređaja iz sustava.

4.2. Korisničko sučelje mobilne aplikacije

index



The image shows a mobile application login screen. At the top, the word 'index' is centered. Below it is a large, light gray rectangular area. Underneath this area is a white rounded rectangle containing the login form. The form consists of two text input fields: the top one is labeled 'Email' and the bottom one is labeled 'Password'. Below these fields is a blue button with the text 'Sign In' in white. Below the button is another large, light gray rectangular area. At the very bottom of the screen is a thin black horizontal line, representing the mobile home indicator bar.

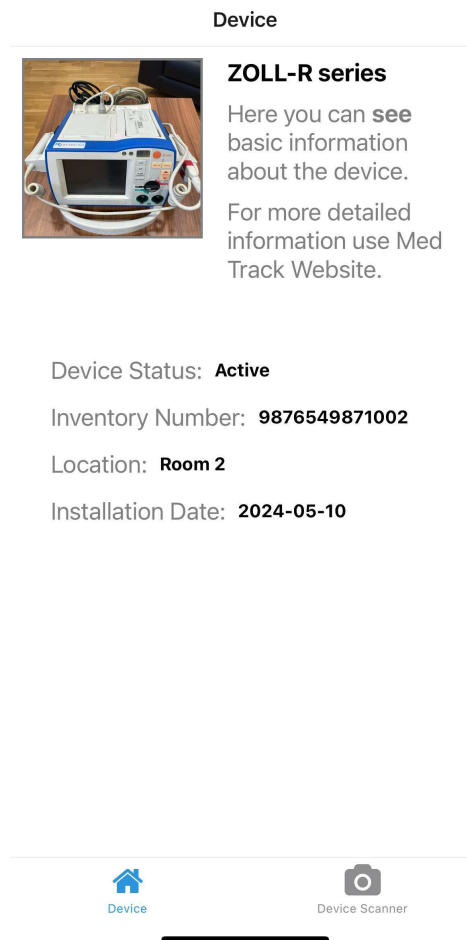
Slika 4.8. Mobilna aplikacija - Prijava u sustav

Slika 4.8. prikazuje obrazac za prijavu korisnika u sustav putem mobilne aplikacije. Korisnici unose svoju e-mail adresu i lozinku kako bi pristupili aplikaciji.



Slika 4.9. Prepoznavanje barkoda

Slika 4.9. prikazuje prepoznavanje barkoda pomoću kamere mobilnog uređaja.



Slika 4.10. Pregled podataka o uređaju

Slika 4.10. prikazuje pregled podataka o uređaju na mobilnoj aplikaciji u slučaju kada sustav detektira da je prepoznati barkod već u bazi podataka. Korisnici mogu vidjeti osnovne informacije o uređaju, uključujući status uređaja, inventarni broj, lokaciju i datum postavljanja uređaja .

Device



Here you can **enter** basic information about the device.

For more detailed information use Med Track Website.

Device Name: Device Name

Device Status: Select status

Inventory Number: 5467864500001

Location: Select location

Installation Date: 2024-06-09

Add Device

 Device
  Device Scanner

Slika 4.11. Dodavanje novog uređaja 1.

Device



Here you can **enter** basic information about the device.

For more detailed information use Med Track Website.

Device Name: Device Name

Device Status: Select status

Inventory Number: 5467864500001

Location: Select location

Installation

Search...

West Wing
 East Wing
 Cardiology
 Intensive Care
 Room 1

Add Device

 Device
  Scanner

Slika 4.12. Dodavanje novog uređaja 2.

Slike 4.11. i 4.12. prikazuju obrazac za unos podataka o novom uređaju u slučaju kada prepoznati barkod nije pohranjen u bazi podataka. Korisnici mogu unijeti osnovne informacije o uređaju, uključujući naziv uređaja, status, inventarni broj, lokaciju i datum postavljanja uređaja.

4.3. Korištene tehnologije i alati

U ovom radu upotrijebljene su sljedeće tehnologije i alati:

- **IntelliJ:** IntelliJ IDEA je razvojno okruženje koje podržava mnoge programske jezike, s naglaskom na Javu. Razvojna okolina IntelliJ služila je za inicijalizaciju projekta u Spring Bootu te izradu poslužiteljske strane sustava.

- **Spring Boot:** Spring Boot je Javin radni okvir koji omogućava brz i jednostavan razvoj aplikacija spremnih za produkciju na platformi Spring. Spring Boot se koristio za izradu poslužitelja.
- **Java:** Java je objektno orijentirani programski jezik koji je u širokoj primjeni na različitim uređajima. Jezik Java se koristio za pisanje poslužitelja.
- **PostgreSQL:** PostgreSQL je relacijski sustav za upravljanje bazama podataka koji podržava normu SQL. PostgreSQL je korišten za upravljanje bazom podataka sustava.
- **Visual Studio Code (VSC):** Visual Studio Code je uređivač koda koji podržava brojne jezike i u velikoj je primjeni u izradi web i mobilnih aplikacija te se koristio upravo u tu svrhu pri izradi završnog rada.
- **React:** React je biblioteka JavaScripta za izgradnju korisničkih sučelja, omogućavajući izradu komponenti koje se mogu ponovno koristiti. Biblioteka React je iskorištena za izradu klijentske strane web aplikacije.
- **Next.js:** Next.js je radni okvir za React koji omogućava generiranje sadržaja web stranica na poslužiteljskoj strani i statičko generiranje web stranica. Radni okvir Next.js se koristio zajedno s Reactom za izradu klijentske strane web aplikacije.
- **React Native:** React Native je okvir za izradu mobilnih aplikacija koristeći JavaScript i React, omogućavajući razvoj aplikacija koje rade na platformama iOS i Android. React Native je korišten za izradu mobilne aplikacije.
- **Expo:** Expo je radni okvir koji pruža alate i usluge za izradu, ispitivanje i distribuciju aplikacija rađenih u radnom okviru React Native. Expo je korišten za izradu klijentske strane mobilne aplikacije.
- **Expo Go:** Expo Go je mobilna aplikacija koja omogućava ispitivanje i pokretanje projekata u Expou na stvarnim uređajima bez potrebe za procesom kompiliranja. Aplikacija Expo Go koristi se u suradnji s radnim okvirom Expo za ispitivanje i pokretanje mobilne aplikacije sustava.
- **Astah:** Astah je programski alat za crtanje dijagrama, uključujući UML dijagrame

i dijagrame tokova podataka, koji pomažu u vizualizaciji i planiranju sustava. As-tah je korišten za izradu UML dijagrama, konkretno dijagrama obrazaca uporabe i dijagrama komponenti.

5. Zaključak

U ovom radu napravljen je osnovni sustav za evidenciju medicinskih uređaja unutar zdravstvenih ustanova. Sustav uključuje web i mobilnu aplikaciju koje zajedno omogućuju sveobuhvatno praćenje statusa uređaja, unos novih uređaja i pregled dostupne dokumentacije.

Iako je osnovna funkcionalnost sustava ostvarena, postoji još puno prostora za nadogradnju. Jedna od mogućih nadogradnji je implementacija sustava za više korisnika, gdje bi svakom korisniku mogle biti dodijeljene specifične ovlasti, poput pristupa samo određenim mapama. Dodatno, kao korisna funkcionalnost mogla bi se implementirati mogućnost podsjetnika za određene uređaje kako bi korisnici imali bolju evidenciju o održavanju i statusu svih uređaja. Nadalje, cijeli sustav se može bolje osigurati korištenjem gotovih rješenja poput OAuth2 za autentifikaciju, što bi dodatno povećalo sigurnost i zaštitu podataka. Implementacija ovih nadogradnji omogućila bi sustavu da postane još učinkovitiji i korisniji za krajnje korisnike, pružajući im sveobuhvatnu i sigurnu platformu za praćenje stanja medicinskih uređaja.

Tijekom izrade ovog rada stekao sam brojne vještine vezane za razvoj web i mobilnih aplikacija te se veselim budućem radu i nadogradnji ovog sustava.

Literatura

- [1] Next.js, *Authentication*, Next.js Documentation, (2024, lipnja). Poveznica: <https://nextjs.org/docs/app/building-your-application/authentication>; pristupljeno 7. svibnja 2024.
- [2] Next.js, *Routing*, Next.js Documentation, (2024, lipnja). Poveznica: <https://nextjs.org/docs/app/building-your-application/routing>; pristupljeno 7. svibnja 2024.
- [3] Expo, *File-based routing*, Expo Documentation, (2024, lipnja). Poveznica: <https://docs.expo.dev/develop/file-based-routing/>; pristupljeno 7. svibnja 2024.
- [4] Teric Cabrel T., *Implement JWT authentication in a Spring Boot 3 application*, Medium, (2024, lipnja). Poveznica: <https://medium.com/@tericcabrel/implement-jwt-authentication-in-a-spring-boot-3-application-5839e4fd8fac>; pristupljeno 7. svibnja 2024.
- [5] Axios, *Introduction*, Axios Documentation, (2024, lipnja). Poveznica: <https://axios-http.com/docs/intro>; pristupljeno 7. svibnja 2024.
- [6] Frid, N., Jović, A., *Modeliranje programske potpore UML-dijagramima*, E-skripta (radna inačica), Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2023.

Sažetak

SUSTAV ZA PRAĆENJE STANJA MEDICINSKIH UREĐAJA

Filip Ljubotina

U ovom radu detaljno je opisana izrada web i mobilne aplikacije namijenjene za efikasno upravljanje i praćenje stanja medicinskih uređaja. Rad istražuje različite funkcionalnosti koje sustav nudi. Također, detaljno su opisane ključne tehnologije i alati koji su korišteni u izgradnji sustava, s naglaskom na sigurnosne aspekte koji štite podatke o medicinskim uređajima.

Ključne riječi: medicinski uređaji; web aplikacija; mobilna aplikacija; sigurnost

Abstract

SYSTEM FOR MONITORING THE STATUS OF MEDICAL DEVICES

Filip Ljubotina

In this thesis, a detailed description of the development of web and mobile applications designed for the efficient management of medical devices is provided. The thesis explores various functionalities offered by the system. Additionally, key technologies and tools used in the system's construction are thoroughly described, with an emphasis on security aspects that protect data related to medical devices.

Keywords: medical devices; web application; mobile application; security