

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 59

Sustav za praćenje uredske inventurne liste

Niko Rudić

Zagreb, lipanj 2021.

ZAVRŠNI ZADATAK br. 59

Pristupnik: **Niko Rudić (0036508812)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: prof. dr. sc. Vedran Mornar

Zadatak: **Sustav za praćenje uredske inventurne liste**

Opis zadatka:

Ostvariti aplikaciju koja na jednostavan način omogućuje praćenje, prenošenje i otpisivanje predmeta za koje je pojedina osoba zadužena, skeniranjem QR-kodova zalijepljenih na predmet. Administrator sustava zadužen je za uređivanje podataka o inventuri, stvaranje novih QR kodova i zaduživanje osoba, odobravanje prijenosa zaduženja ili odobravanje otpisa zaduženja. Korisničko sučelje ostvariti mobilnom aplikacijom, a administratorsko sučelje ostvariti u web pregledniku. Odabrati odgovarajuće tehnologije i prikladnu bazu podataka.

Rok za predaju rada: 11. lipnja 2021.

SADRŽAJ

| | |
|--|-----------|
| 1. Uvod | 1 |
| 2. Zahtjevi | 2 |
| 3. Arhitektura i tehnologije | 3 |
| 3.1. Troslojna arhitektura | 3 |
| 3.1.1. Prezentacijski sloj | 3 |
| 3.1.2. Aplikacijski sloj | 4 |
| 3.1.3. Podatkovni sloj | 4 |
| 3.2. Korištene tehnologije | 4 |
| 3.2.1. REST API | 4 |
| 3.2.2. QR kod | 7 |
| 3.2.3. Vue.js | 8 |
| 3.2.4. React native | 10 |
| 3.2.5. Node.js | 12 |
| 4. Detalji sustava | 14 |
| 4.1. Baza podataka | 14 |
| 4.2. Serverska aplikacija (eng. <i>Backend</i>) | 15 |
| 4.2.1. Sloj za pristup podacima | 16 |
| 4.2.2. Poslovni / upravljački sloj | 16 |
| 4.2.3. Sloj usmjeravanja | 17 |
| 4.3. Web sučelje | 17 |
| 4.4. Mobilna aplikacija | 23 |
| 5. Zaključak | 27 |
| Literatura | 28 |

1. Uvod

U Republici Hrvatskoj svi poduzetnici su obavezni na početku poslovanja popisati svu imovinu i njihovu pojedinačnu vrijednost te barem jednom godišnje osvježavati taj popis sukladno članku 11. Zakona o računovodstvu (Narodne novine br. 109/2007) [12]. Rapidni razvoj tvrtke, a osobito onih u ICT sektoru, otežava praćenje imovinskog stanja. Osim zakonskih problema to može dovesti i do zadržavanja nekorištene i zastarjele imovine poput računala, laptopa, raznih uređaja i strojeva.

Često korišteno rješenje je zaduživanje svakog zaposlenika za imovinu koju taj zaposlenik koristi. Svake godine zaposlenik je dužan potvrditi da i dalje posjeduje sve predmete za koje je zadužen. U bilo kojem trenutku zaposlenik može otpisati predmet za koji je zadužen (npr. laptop u slučaju kvara ili dobivanja novog) tako da se javi osobi koja je za to zadužena te potpiše da taj predmet otpisuje. Kad osoba napusti firmu mora sva svoja zaduženja prenijeti vlasništvo na druge osobe te one predmete koje nije u mogućnosti pronaći treba otplatiti. S godinama, liste koje se prate pomoću papira i excel dokumenata postanu neorganizirane i zastarjele te se puno nepostojećih predmeta prenose na druge osobe. Iako takvo rješenje ubrzava i olakšava proces inventure imovine tvrtke, vidljivo je da s vremenom takve liste nisu rješenje. Također postaje puno teže pratiti starost imovine koju je potrebno redovito mijenjati (primjerice računalna oprema).

Rezultat ovog završnog rada je cjeloviti sustav za praćenje uredske inventurne liste koje će omogućiti tvrtkama, a i njihovim zaposlenicima, jednostavno praćenje imovine tvrtke, zaduženja zaposlenika, starosti pojedinog inventara te lagano prenošenje i otpisivanje zaduženja.

2. Zahtjevi

Samo registrirani korisnici imaju pristup sustavu, a s obzirom na to da se radi o internom sustavu neke tvrtke, administratori su zaduženi za registraciju korisnika. Sustav se sastoji od mobilnog i web sučelja.

Mobilnom sustavu mogu pristupiti svi registrirani korisnici i administratori koji u tom slučaju imaju iste funkcionalnosti kao i korisnici, a to su: skeniranje QR kodova, pregled vlastite liste zaduženog inventara, pregled i podnošenje zahtjeva za prijenos i otpis inventara.

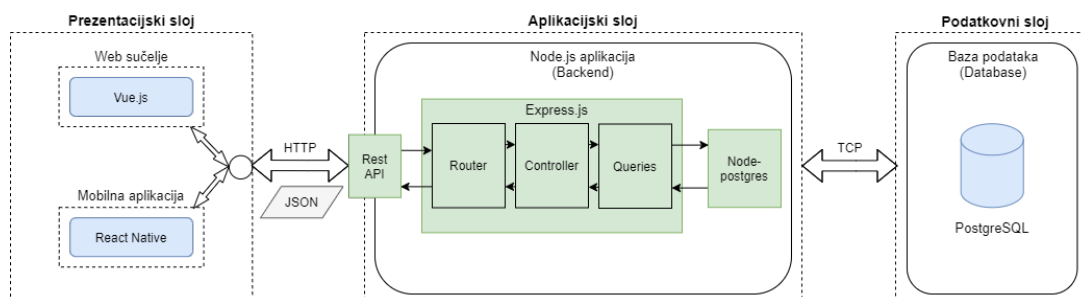
Pristup web sučelju imaju samo administratori te u njemu imaju pregled potpunog inventara tvrtke s mogućnošću naprednog pretraživanja po raznim kriterijima poput: vlasnika, datuma kupovine, grupe artikala, kupovnoj cijeni i sl.). Tako filtriran inventar je moguće izvesti u obliku CSV datoteke. Također je moguće dodavati nove artikle, a svaki pojedini artikl je moguće urediti, prenijeti na drugog korisnika i otpisati. Slične mogućnosti postoje i za pregled svih korisnika, zahtjeva za prijenos i zahtjeva za otpis. Svaki zahtjev je moguće prihvatiti ili odbiti, a već prihvaćeni ili odbijeni zahtjevi se ne brišu kako bi postojao zapis i kako bi se omogućila neporecivost podataka.

U web sučelju je omogućeno administratorima da za odabrane artikle stvore jedinstvene QR kodove koji se zatim ispisuju na naljepnice te zalijepe na inventar. Navedene QR kodove korisnici pomoću aplikacije mogu skenirati i jedino tako potvrditi da su i dalje u posjedu tog predmeta. Ostale predmete na koje nije moguće zalijepiti QR kodove je dovoljno samo potvrditi u aplikaciji.

3. Arhitektura i tehnologije

3.1. Troslojna arhitektura

Kao i kod svakog kompleksnijeg sustava, pri izradi ovog sustava korišteno je više programskih jezika, biblioteka, tehnologija i alata. Odabir raznih programskih jezika i tehnologija s vremenom predstavlja problem prilikom bilo kakvog ažuriranja pojedinog djela sustava pa se u tu svrhu preporučuje dijelove razdvajati na neovisne cjeline. Upravo zbog toga, odabrana je troslojna arhitektura (eng. *three-tier architecture*) koja razdvaja programsko rješenje na tri nezavisna sloja: prezentacijski sloj (eng. *presentation tier*), aplikacijski sloj (eng. *application tier*) i podatkovni sloj (eng. *data tier*). Takva struktura omogućuje neovisan razvoj svakog sloja zasebno kao i njihovo buduće ažuriranje, skaliranje, premještanje ili zamjenu. [8]



Slika 3.1: Dijagram troslojne arhitekture sustava

3.1.1. Prezentacijski sloj

Prezentacijski sloj ovog sustava se sastoji od dva neovisna aplikacijska rješenja: mobilna aplikacija i web sučelje koji komuniciraju s aplikacijskim slojem poštujući REST arhitekturu, odnosno koristeći HTTP zahtjeve kojima primaju i šalju strukturane JSON objekte. Web sučelje napisano koristeći *Vue.js* s (*javascriptframework*),

Axios, *SCSS*, *Javascript ES6* te *PrimeVue* kao radni okvir za dizajn korisničkog sučelja (*UI library*). Mobilna aplikacija je napisana koristeći *React Native* s *Expo* radnim okvirom za lakši i brži razvoj.

3.1.2. Aplikacijski sloj

Aplikacijski sloj je ostvaren koristeći *Node.js* i *Express.js* kao radni okvir za *Node.js*. Za dohvaćanje podataka s podatkovnog sloja, odnosno iz baze podataka se koristi biblioteka *node-postgres* koja olakšava implementaciju asinkronih zahtjeva i obećanja (eng. *promises*). Sučelje prema prezentacijskom sloju je sukladno REST arhitekturi, odnosno, koriste se GET, POST, PUT i DELETE HTTP zahtjevi koji su poslani s korisničkog sučelja na odgovarajući URI kako bi se dohvatili ili izmijenili podaci na serveru. U aplikacijskom sloju usmjeritelj (eng. *Router*) dobivene zahtjeve preusmjerava na odgovarajući kontroler (eng. *Controller*), koji zatim odgovarajuće podatke ubacuje u već definirane SQL upite (eng. *Queries*) te upit prosljeđuje na podatkovni sloj, koji upit izvrši nad bazom podataka i vraća rezultat izvršavanja upita.

3.1.3. Podatkovni sloj

Podatkovni sloj se sastoji od relacijske baze podataka stvorene pomoću jezika SQL i PostgreSQL sustava za upravljanjem bazom podataka. PostgreSQL je besplatan softver otvorenog koda s objektno-relacijskim sustavom za upravljanje bazama podataka (eng. *relational database management system* - RDBMS) koji koristi i proširuje standardni SQL jezik [1]. Iako se radi o naprednom sustavu s pregršt naprednih funkcionalnosti, za potrebe ovog projekta koristit će se samo one osnovne.

3.2. Korištene tehnologije

Iako je broj korištenih tehnologija i programskih jezika puno veći te ih je većina bila prije spomenuta, neke od bitnijih su detaljnije opisane u nastavku.

3.2.1. REST API

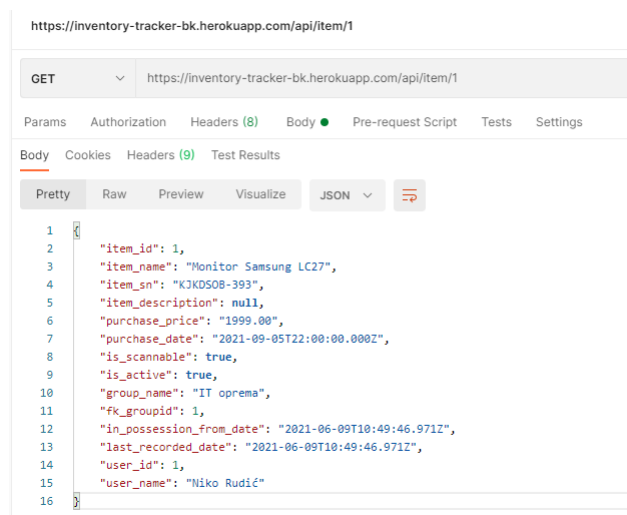
Iako REST (*Representational state transfer*) nije nužno tehnologija već arhitekturni stil izgradnje programskog rješenja, on je jedan od najvažnijih i najpopularnijih stilova današnjice te ga ovaj sustav implementira i koristi za laganu i strukturiranu razmjenu podataka između aplikacijskog i prezentacijskog sloja (klijenta i poslužitelja).

REST arhitektura naglašava važnost skalabilnosti, uniformnosti sučelja i neovisnosti komponenata definirajući set ograničenja, a ona su:

– **Klijent - Poslužitelj (eng. *Client - Server*)**

Ograničenje klijent-poslužitelj se odnosi na razdvajanje klijenta od poslužitelja. Poslužitelj je odgovoran za obradu i pružanje podataka, a klijent je odgovoran za prikaz tih podataka na korisničkom sučelju. Poslužitelj i klijent se mogu razvijati ili čak u potpunosti zamijeniti, neovisno jedan o drugome. Oni međusobno komuniciraju koristeći HTTP protokol i GET, POST, PUT i DELETE metode na odgovarajući URI. Zahtjevi se naknadno mogu proširiti koristeći *query stringove* (npr. `?id=3`), a kao rezultat takvih zahtjeva dobivamo HTML, JSON ili XML objekte (za ovaj sustav koriste se JSON objekti).

Primjer jednog takvog upita koristeći alat Postman može se vidjeti u nastavku.



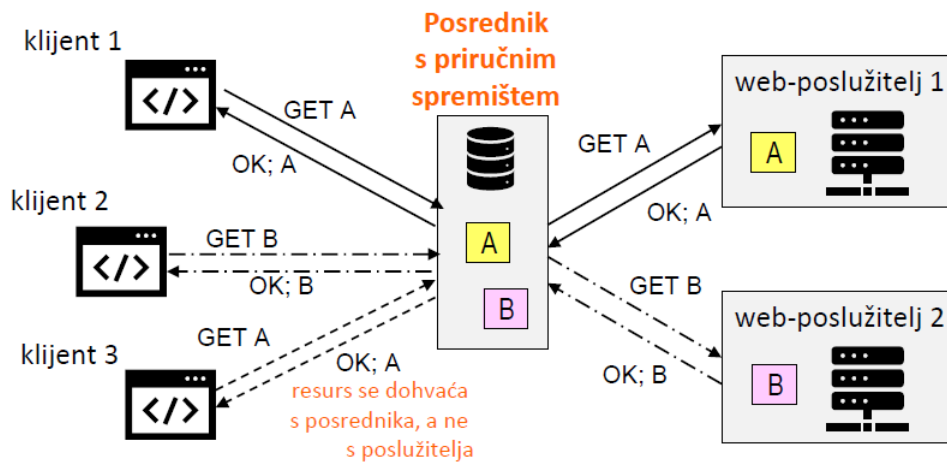
Slika 3.2: Primjer HTTP GET zahtjeva i odgovora [2]

– **Neovisnost zahtjeva (eng. *Stateless*)**

Neovisnost zahtjeva odnosi se na ograničenje u komunikaciji između klijenta i poslužitelja u kojoj svaki zahtjev i njegov odgovor trebaju biti neovisni o ostalim zahtjevima. To ograničenje također znači da svaki zahtjev treba sadržavati sve potrebne podatke za njegovu obradu i eventualnu autentifikaciju korisnika. Ovakva arhitektura je idealna za velike i prometne sustave jer omogućuje skalabilnost, pouzdanost i lakše oporavljanje prilikom neuspjelih zahtjeva, no nedostatak ovog pristupa je slanje svih podataka prilikom svakog zahtjeva, što može dovesti do smanjenja performansi.

– **Priručna memorija (eng. *Cache*)**

Ovo ograničenje odnosi se na poslužitelj koji treba implicitno ili eksplicitno označiti želi li da se njegovi odgovori spremaju u priručnu memoriju. Prednost spremanja podataka u priručnu memorije je rasterećenje poslužitelja jer se svaki sljedeći zahtjev za taj resurs može dohvatiti iz priručne memorije. Primjer takve interakcije je prikazan na prikazu ispod. Iako ovo ograničenje nema nedostataka, ne mogu se svi podaci spremati u priručnu memoriju, već ima smisla spremati samo one čija se vrijednost ne mijenja često.



Slika 3.3: Primjer posrednika s priručnom memorijom [16]

– **Slojeviti sustav (eng. *Layered system*)**

Implementacija slojevitog sustava znači da klijent ne može odrediti povezuje li se direktno na server ili na nekakav proxy poslužitelj koji se nalazi na putu između njega i poslužitelja. Osim što takva implementacija sustava može dodatno rasteretiti glavni sustav i poboljšati skalabilnost poslužujući dio zahtjeva iz vlastite priručne memorije, ona također omogućuje implementiranje dodatnog sigurnosnog sloja prije samog aplikacijskog sloja, koji štiti od neželjenih napada, kao što je napad uskraćivanjem resursa (eng. *DDoS - distributed denial-of-service*).

– **Kod na zahtjev (eng. *Code on demand*)**

REST također omogućuje proširenje i prilagođavanje klijentove funkcionalnosti prebacivanjem izvršnog koda, kao što su *Java apleti* ili klijentske skripte napisane u *JavaScriptu*.

– **Uniformno sučelje (eng. *Uniform interface*)**

Uniformno sučelje je ključno ograničenje i osnova svakog RESTful sustava. Ovakav dizajn sustava odvaja arhitekturu i omogućuje da se svaki dio razvija samostalno. [7]

3.2.2. QR kod

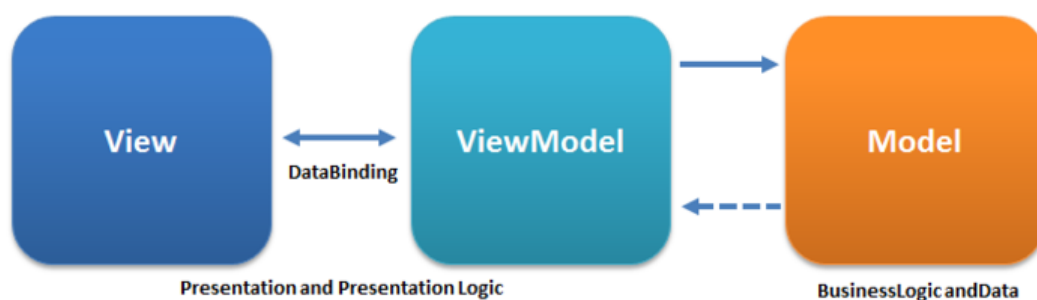
QR kod (eng. *quick response code*) je dvodimenzionalan simbol standardiziran po ISO standardu ISO/IEC18004. QR kod karakterizira izuzetno brza čitljivost iz svih smjerova te visoka gustoća podataka - do 100 puta veća od tadašnjih linearnih simbola poput barkodova, a maksimalno može prikazati 7089 numeričkih simbola. Osim brze čitljivosti i visoke gustoće, QR kod implementira i neke druge napredne značajke poput otpornosti na iskrivljene simbole, ispravljanje pogreške (četiri razine: 7%, 15%, 25% i 30% površine simbola), ulančavanje simbola, enkripcija i dr. QR kodove može očitati svaki mobitel s fotoaparatom i instaliranom aplikacijom za čitanje QR kodova ili već ugrađenom takvom funkcionalnošću. Upravo zbog navedenih razloga QR kod je jedan od najpopularnijih tipova matičnog barkoda današnjice te ga ovaj sustav implementira za očitavanje inventara. [6, 13]



Slika 3.4: Primjer QR koda [6]

3.2.3. Vue.js

Vue.js je besplatan JavaScript *framework* otvorenog koda (eng. *open source*) namijenjen za izgradnju korisničkog sučelja za HTML korištenjem JavaScript jezika. Početni radni okvir je vrlo malen (samo 33KB u gzip formatu), ali se i dalje radi o izuzetno fleksibilnom i naprednom alatu s kojim je pomoću samo nekoliko biblioteka moguće izraditi velike i napredne klijentske aplikacije, zbog čega poznat kao jedan od najpopularnijih JavaScript radnih okvira današnjice. Osim izuzetno male veličine, odlikuje ga jednostavnost upotrebe, fleksibilnost i ponovna iskoristljivost izrađenih komponenti koja je moguća zbog Model-View-ViewModel arhitekture. [9]



Slika 3.5: Prikaz arhitekture model-view-viewmodel [3]

Model-View-ViewModel je arhitekturni stil koji odvaja poslovnu logiku, odnosno rad nad podacima od grafičkog korisničkog sučelja, a povezuje ih međusloj ViewModel koji je odgovoran za pravilno vezivanje podataka na određene komponente grafičkog sučelja. [3]. Važno je naglasiti kako se tu radi o operacijama nad podacima koji su već stigli s poslužitelja te se čitava obrada događa na klijentu.

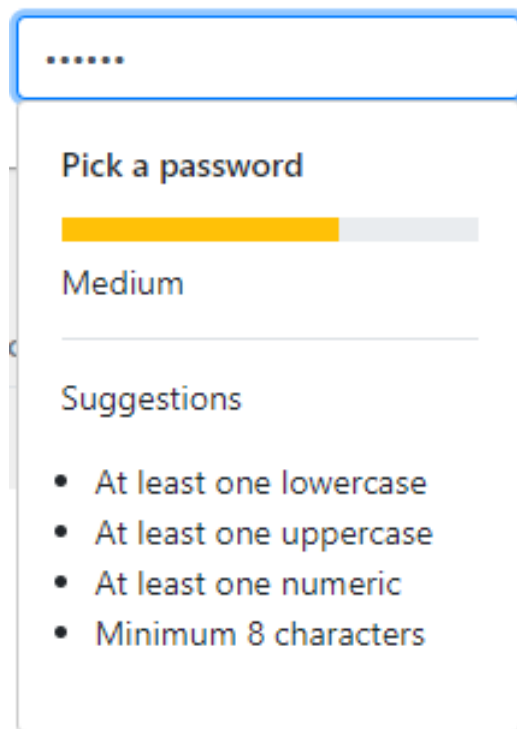
Njegova struktura sastoji se od komponenti (eng. *components*) koje proširuju osnovne HTML elemente i njih enkapsuliraju kako bi postigli višekratnu upotrebljivost koda. Takve komponente nalaze se unutar predložaka (eng. *templates*) koje Vue kompajlira u virtualne funkcije za prikaz DOM-a (virtualni objektni model dokumenta) te u kombinaciji sa sustavom reaktivnosti, koji veže JavaScript objekte na navedene komponente, omogućuje optimizirano ponovno prikazivanje prilikom svake promjene bez potrebe za osvježavanjem stranice.

Vue je jednostrana aplikacija (eng. *SPA - single page application*) te se izmjena njegovih komponenti događa unutar iste početne stranice (obično `index.html`) što omogućuje standardnu navigaciju pomoću URL-a, no kao rješenje se nudi službeno

podržana biblioteka *vue-router* koji služi kao usmjerivač. Njegov zadatak je da na temelju trenutne staze URL-a prikazuje zadanu komponentu unutar početne stranice koje su unaprijed određene, najčešće unutar *router.js* datoteke. [9].

Prime vue

Prime vue je biblioteka korisničkog sučelja napravljena za Vue.js. Sastoji se od preko 80 već gotovih komponenti koji omogućuju programerima brz i jednostavan razvoj. Omogućuje nekoliko alternativa za dizajn kao što su: Material design, Twitter bootstrap, Fluent UI i dr. Prime Vue nudi mogućnost kreiranja vlastitog dizajna, a teme se mijenjaju izmjenom samo jednog retka koda bez potrebe za naknadnim podešavanjem ili refaktoriranjem koda već gotove aplikacije. [5]



Slika 3.6: Prikaz Prime vue komponente za unos lozinke [5]

```
<template>
  <div>
    <Password v-model="passwordValue">
      <template #header>
        <h6>Pick a password</h6>
```

```

    </template>
    <template #footer="sp">
      {{sp.level}}
      <Divider />
      <p class="p-mt-2">Suggestions</p>
      <ul class="p-pl-2 p-mt-0" style="line-height: 1.5">
        <li>At least one lowercase</li>
        <li>At least one uppercase</li>
        <li>At least one numeric</li>
        <li>Minimum 8 characters</li>
      </ul>
    </template>
  </Password>
</div>
</template>

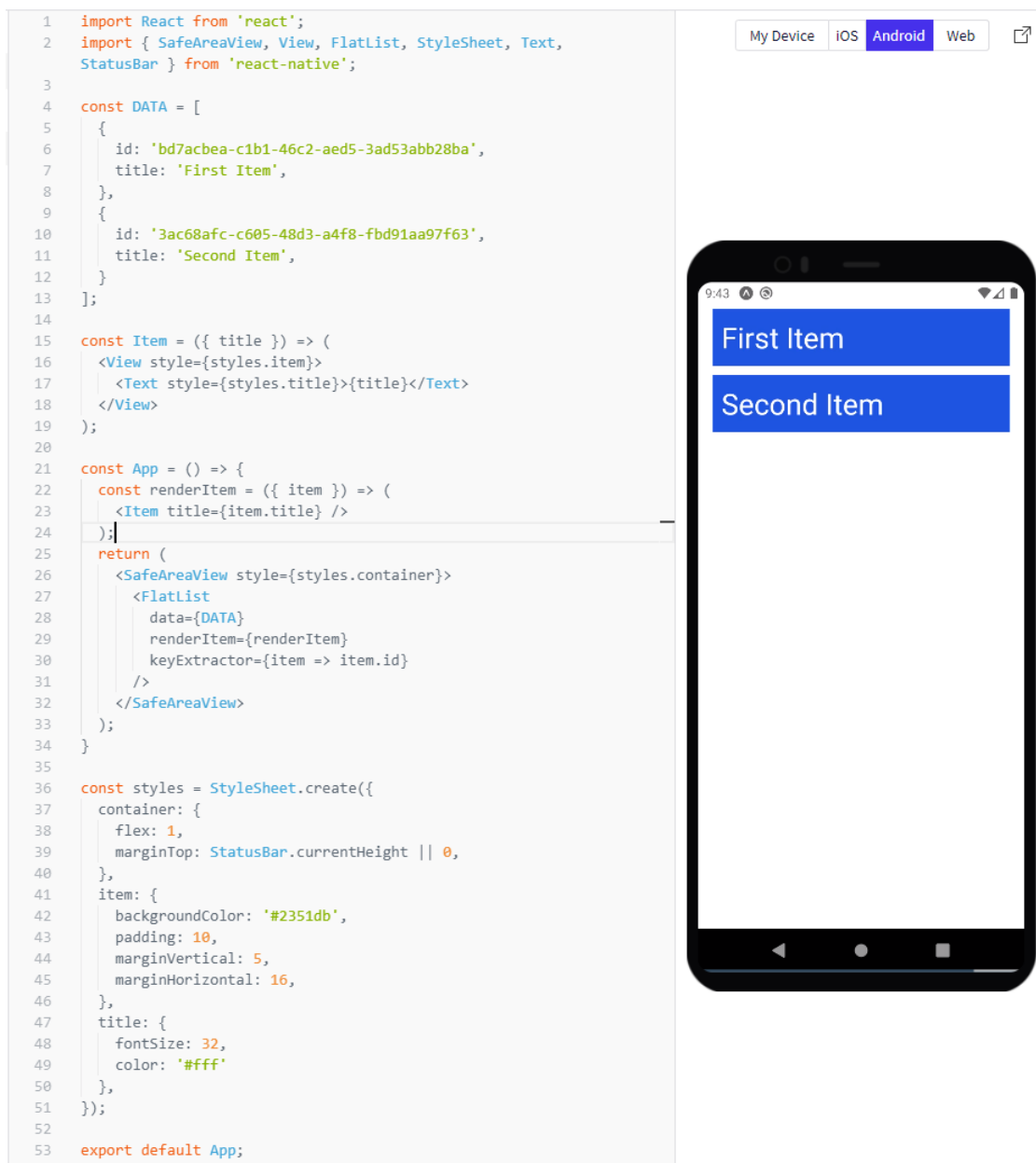
<script>
import Password from 'primevue/password';
export default {
  data() {
    return {
      passwordValue: null,
    }
  }
}
</script>

```

Slika 3.6 i isječak koda prikazuju primjer komponente za unos lozinke. Komponenta dolazi s već definiranim izgledom i funkcionalnostima koje se mogu uključivati dodavanjem unaprijed definiranih ključnih riječi u komponentu. Također postoje unaprijed definirani predlošci poput prikazanog predloška za prikaz razine sigurnosti lozinke. Navedene karakteristike omogućuju brz razvoj korisničkog sučelja koje zadržavaju isti dizajn kroz čitavu aplikaciju i na taj način poboljšavaju korisničko iskustvo.

3.2.4. React native

React native je JavaScript framework otvorenog koda kojeg je stvorio Facebook 2015. godine, a namijenjen je za razvoj korisničkih sučelja mobilnih aplikacija. Iako je pisan u JavaScriptu, njegovi sastavni dijelovi se prevode u komponente specifične za matičnu platformu. Takav pristup omogućuje da se održava samo jedan izvorni kod napisan u JavaScriptu, a posluhuje više platformi – najčešće su to iOS i Android, no aplikaciju je moguće prevesti i za Windows, macOS, AndroidTV, tvOS i Web. [10]



Slika 3.7: Prikaz React Native liste na Androidu [10]

Expo

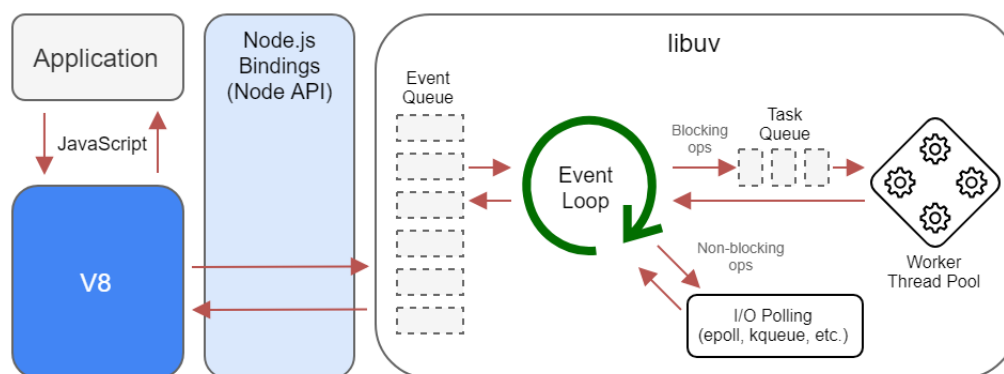
Expo je set alata i usluga stvoren oko React Native platforme kako bi pomogli u brzom razvoju, izgradnji i puštanju u rad aplikacija za iOS, Android i Web platforme. Uz opciju pokretanja koda u simulatoru, expo nudi mogućnost instaliranja službene aplikacije Expo Go te pokretanje vlastite aplikacije jednostavnim skeniranjem QR koda.

3.2.5. Node.js

Node.js je višeplatformsko Javascript okruženje koje se izvodi na V8 stroju i omogućuje izvođenje operacija na strani poslužitelja. To nam omogućuje korištenje istog programskog jezika (JavaScript) za izgradnju klijentske i poslužiteljske aplikacije što ubrzava i olakšava razvoj. Node.js sadrži module koji omogućavaju rad s datotekama, mrežama (DNS,HTTP,TCP,TLS/SSL i UDP protokoli), binarnim podacima, kriptografijom i sl. Za razliku od primjerice PHP-a, Node.js koristi neblokirajuće asinkrone funkcije koje ne čekaju na kraj izvršenja, već se nastavlja rad nad drugim operacijama i osluškuje se signal završetka. Ova funkcionalnost je moguća zbog implementirane arhitekture zasnovane na događajima[4].

Arhitektura zasnovana na događajima

Arhitektura zasnovana na događajima je softverska arhitektura koja počiva na generiranju, otkrivanju i reakciji na događaje. Komponente se međusobno ne pozivaju eksplicitno već jedne generiraju signale (događaje) dok druge osluškuju pojavu određenih događaja. Komponente koje čekaju na događaj, kao ni one koje ga generiraju, nemaju informaciju jedna o drugoj, niti znaju hoće li i koja komponenta na taj događaj reagirati. Svi događaji se javno objavljuju te se pozivaju registrirane procedure [15].



Slika 3.8: Node.js implementacija arhitekture zasnovane na događajima [11]

Ovakva implementacija arhitekture omogućuje Node.js-u da izvodi neblokirajuće, asinkrone naredbe I/O operacije nad operacijskim sustavom unatoč tome što JavaScript ima mogućnost izvođenja u samo jednoj niti. Ovo se ostvaruje na način da se maksimalan broj operacija dodjeljuje operacijskom sustavu jer on ima mogućnost izvođenja u

više niti, a Node.js nastavlja s izvođenjem drugih operacija i osluškuje signal operacijskog sustava o završenim operacijama te na njih reagira. Takva arhitektura omogućuje Node.js-u da bude memorijski efikasan, brz i skalabilan. [11].

Express.js

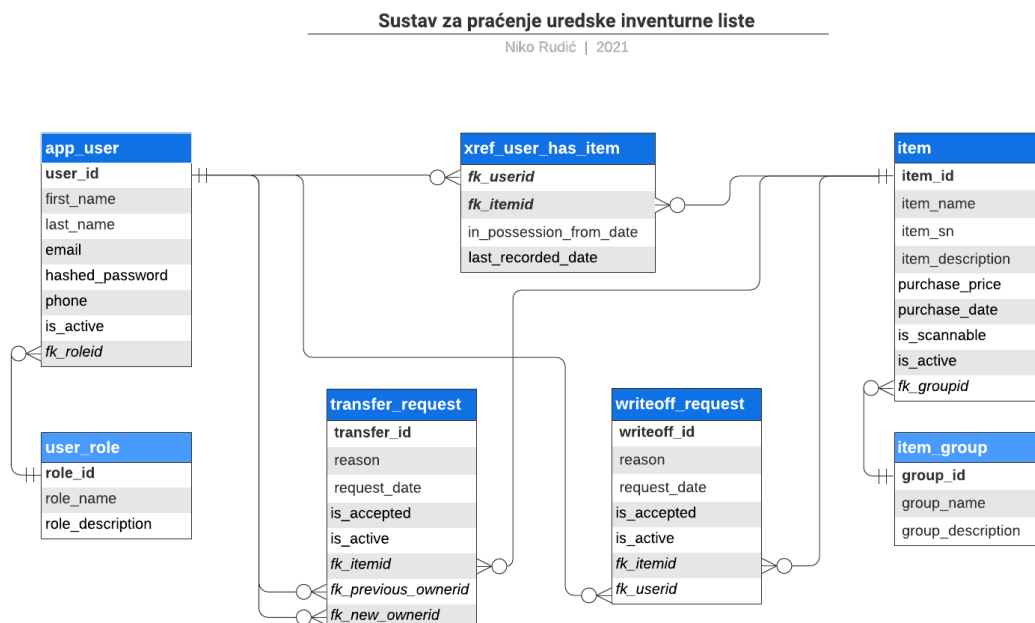
Express.js je besplatan radni okvir za Node.js. On olakšava i ubrzava izgradnju aplikacija te omogućuje lagano upravljanje nekim naprednijim značajkama, kao što su spajanje i upiti nad bazom podataka, usmjeravanje zahtjeva, održavanje sjednica, upravljanje greškama i sl. Express.js iskorištava pogodnosti neblokirajućih asinkronih operacija koje pruža Node.js i tzv. *Event Loop* koji omogućuje izuzetno brzo izvođenje i izbjegava neefikasnosti asociirane sa sinkronim izvođenjem.[14]

4. Detalji sustava

U ovom poglavlju opisani su svi detalji sustava počevši s dijagramom baze podataka, zatim serverske aplikacije te pristup i obrada podataka, nakon čega slijedi prikaz administratorskog web sučelja. Na kraju je prikazana mobilna aplikacije namijenjena za zaposlenike tvrtke.

4.1. Baza podataka

Relacijska baza podataka implementirana je koristeći PostgreSQL. Dijagram u nastavku prikazuje sve entitete implementirane baze podataka kao i njihove međusobne odnose.



Slika 4.1: Dijagram baze podataka

Svi korisnici se pohranjuju u entitetu ***app_user*** te kao primarni ključ se koristi polje *user_id*. Svaki korisnik ima točno jednu ulogu, a to je ostvareno pomoću stranog ključa (eng. *foreign key*) *fk_roleid* koji predstavlja referencu na primarni ključ *role_id* entiteta ***user_role***.

Svaki korisnik može biti vlasnik više artikala koji su pohranjeni u entitetu ***item***, a povezani su međutablicom ***xref_user_has_item***. Primarni ključ međutablice je kompozitni primarni ključ sastavljen od stranih ključeva na tablice ***app_user*** i ***item***. Međutablica dodatno sadrži podatke o datumu početka vlasništva *in_possession_from_date* i datumu zadnje potvrde *last_recorded_date*.

Svaki artikl pripada nekoj grupi artikala s kojom je povezan pomoću stranog ključa *fk_groupid* koji predstavlja referencu na primarni ključ *group_id* entiteta ***item_group***.

Svaki korisnik može napraviti više zahtjeva za prijenos koji se pohranjuju u entitetu ***transfer_request***. Svaki zahtjev za prijenos ima svoj primarni ključ *transfer_id* te tri strana ključa: *fk_itemid*, *fk_previous_ownerid* i *fk_next_ownerid* koji su reference na entitete ***item*** i ***app_user***.

Slično, svaki korisnik može napraviti više zahtjeva za otpis koji su pohranjeni u entitetu ***writeoff_request***, a imaju primarni ključ *writeoff_id* i strane ključeve *fk_itemid* i *fk_userid* koji su reference na entitete ***item*** i ***app_user***.

4.2. Serverska aplikacija (eng. *Backend*)

Serverska aplikacija napisana je koristeći *Node.js* radni okvir za *JavaScript*. Za olakšan razvoj dodatno je korišten *Express.js* kao dodatni, minimalistički radni okvir za *Node.js* koji olakšava upravljanje rutama i reagiranje na nadolazeće zahtjeve.

Troslojna arhitektura, osim u čitav sustav, je implementirana i u aplikacijski sloj koji je napisan u *Node.js*-u. Tri sloja su: aplikacijski sloj, poslovni (upravljački) sloj i sloj za pristup podacima. Struktura projekta aplikacije se može vidjeti u nastavku.

4.2.1. Sloj za pristup podacima

Sloj za pristup podacima ili podatkovni sloj je sloj najbliži bazi podataka i odgovoran je za dohvaćanje, ažuriranje i brisanje podataka. Za razgovor s bazom podataka umjesto objektno-relacijskog modela (ORM) implementiran je upravljački program za baze podataka *node-postgres*, koji omogućuje direktno pisanje SQL upita, koji se zatim prosljeđuju PostgreSQL bazi podataka, gdje se izvršavaju i vraćaju odgovor. Ovakav pristup omogućava maksimalnu fleksibilnost i kontrolu nad upitima, ali je podložan greškama i zahtijeva dobro poznavanje SQL jezika. Primjer upita za dohvaćanje svih artikala (inventara) je prikazan ispod.

```
const { Pool } = require('pg')
const { response } = require('express');

const.getItems = async () => {
  try {
    const sqlQuery =
      `SELECT item_id,item_name,item_sn,item_description,purchase_price,
        purchase_date,is_scannable,item.is_active,group_name,fk_groupid,
        in_possession_from_date, last_recorded_date, user_id,
        concat(first_name, ' ', last_name) as user_name
      FROM item
      LEFT JOIN item_group ON fk_groupid = group_id
      LEFT JOIN xref_user_has_item ON item_id = fk_itemid
      LEFT JOIN app_user ON fk_userid = user_id`

    const response = await pool.query(sqlQuery);
    console.log(response)
    return response.rows;

  } catch (error) {
    throw new Error(error.message);
  }
}
```

4.2.2. Poslovni / upravljački sloj

Poslovni sloj ili upravljački sloj (controller) je međusloj između podatkovnog sloja i sloja usmjeravanja. Usmjeravatelj proslijedi zahtjev odgovarajućem *kontroleru* koji zatim poziva metodu iz podatkovnog sloja, a ona vraća rezultat izvršavanja SQL upita nad bazom. Poslovni sloj zatim, ako je potrebno, obradi te podatke te rezultat vraća nazad usmjeritelju. Primjer kontrolera za dohvat podataka je prikazan ispod.

```
const itemQueries = require('../data/item');

const getItem = async (req, res, next) => {
  try {
    const items = await itemQueries.getItems();
    res.send(items)
  } catch (error) {
    res.status(400).send(error.message);
  }
}
```

4.2.3. Sloj usmjeravanja

Sloj usmjeravanja (eng. *routing layer*) implementiran je primjenom tzv. REST API servisa koji na temelju HTTP zahtjeva koji dolaze na unaprijed određene i strukturirane URL adrese, pronalazi odgovarajući kontroler i njemu ih prosljeđuje. Primjer usmjeravatelja za sve upite nad artiklima je dan u nastavku.

```
const express = require('express');
const itemController = require('../controllers/itemController');
const router = express.Router();

// .../api/item/...
router.post('/ownership', itemController.addItemOwnershipToUser);
router.put('/transfer', itemController.transferItemsToAnotherUser);
router.put('/ownership', itemController.updateUserItemLastRecordedDate);
router.delete('/ownership', itemController.deleteUserItemOwnership);
router.put('/disable/:id', itemController.disableItem);
router.get('/', itemController.getItems);
router.get('/:id', itemController.getItemById);
router.post('/', itemController.addItem);
router.put('/:id', itemController.editItem);
router.get('/user/:id', itemController.getItemsByUserId);

module.exports = router
```


Primjer strukturirane adrese je `/api/item/disable/5`. HTTP PUT zahtjev poslan na tu adresu prosljeđuje upit *itemControlleru* metodi *disableItem*.

4.3. Web sučelje

Web sučelje je napravljeno pomoću *Vue 3* radnog okvira za JavaScript te pomoću *Prime Vue* biblioteke za izradu korisničkog sučelja. S obzirom na to da je *Vue.js* napravljen za izradu jednostraničnih aplikacija (eng. *single page application*), navigacija je omogućena koristeći *Vue Router* koji za odgovarajući URL prikazuje odabranu

komponentu. Održavanje stanja varijabli kroz različite komponente postignuto je korištenjem *Vuex* biblioteke. Za sve upite na poslužitelj korišten je *Axios* koji olakšava asinkrono slanje HTTP zahtjeva.

Login



Prijava

Slika 4.2: Forma za prijavu na sučelje

Slika 4.2 prikazuje formu za prijavu na web sučelje, a ono je namijenjeno samo za administratore sustava te se u njega mogu prijaviti jedino administratori. Sučelje sadržava prikaz svih korisnika, artikala i zahtjeva za prijenos i otpis. Svi prikazi se mogu filtrirati po raznim kriterijima te ih se takve preuzeti u .csv formatu.

Novi korisnik

Izbriši filtre

Izvoz

Korisnici

Pretraživanje...

| Id | Ime | Prezime | Email | Kontakt br. | Status | Uloga | Akcije |
|----|---------|-----------|-----------------------|-------------|---------|---------------|--|
| 1 | Niko | Rudić | niko.rudic@fer.hr | 099123456 | Aktivan | Administrator | <div><div></div><div></div><div></div></div> |
| 2 | Ivan | Ivanković | ivan.ivankovic@fer.hr | 097345123 | Aktivan | Korisnik | <div><div></div><div></div><div></div></div> |
| 3 | Ana | Ivanković | ana.ivankovic@fer.hr | 091908765 | Aktivan | Korisnik | <div><div></div><div></div><div></div></div> |
| 4 | Dominik | Domec | dominik.domec@fer.hr | 099654321 | Aktivan | Korisnik | <div><div></div><div></div><div></div></div> |

<<

<

1

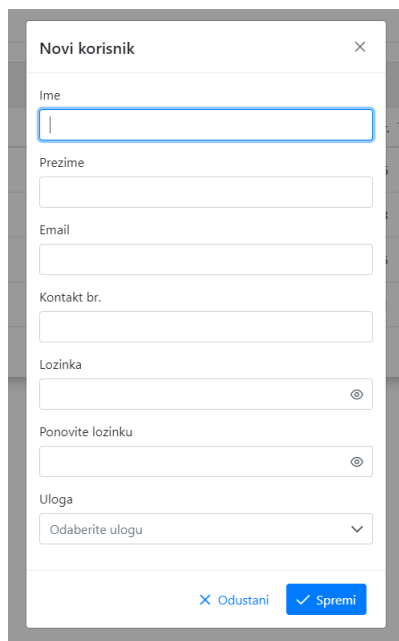
>

>>

10

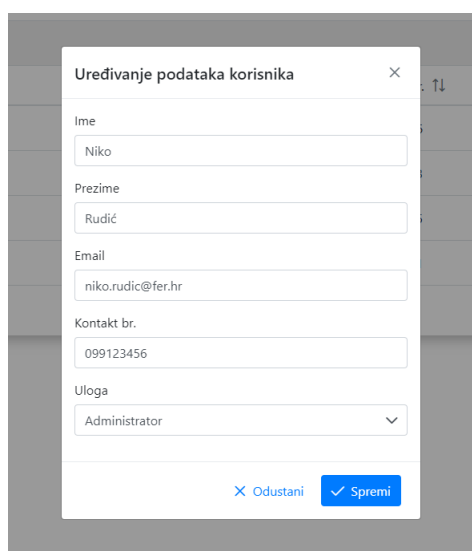
Slika 4.3: Prikaz tablice svih korisnika

Slika 4.3 prikazuje tablicu svih korisnika. Moguće je dodavati nove korisnike, uređivati postojeće, promijeniti njihovu lozinku te ih deaktivirati. Svi deaktivirani korisnici ostaju u sustavu s oznakom "Neaktivan" zbog održavanja zapisa.



Slika 4.4: Forma za registraciju novog korisnika

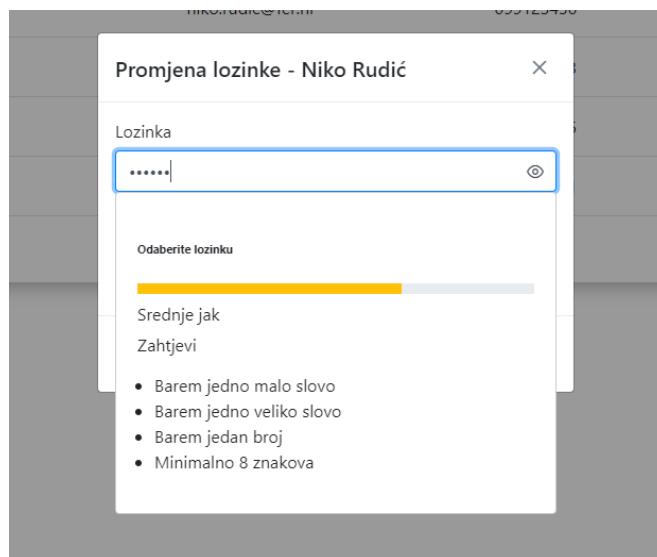
Pritiskom na gumb "Novi korisnik", prikazuje se forma za unos novog korisnika. Prilikom unosa potrebno je definirati korisnikovu lozinku koja treba sadržavati malo i veliko slovo, broj i barem 8 znakova. Prikaz zahtjeva za lozinku je vidljiv na slici 4.6. Također je potrebno odabrati njegovu ulogu iz izbornika koji dohvaća sve dostupne uloge na sustavu (u ovom slučaju samo administrator i korisnik).



Slika 4.5: Forma za uređivanje podataka korisnika

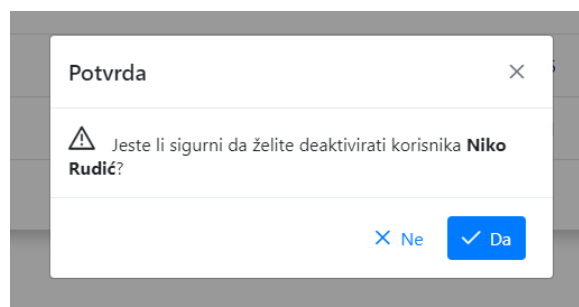
Pritiskom na gumb "Uređivanje korisnika", za odabranog korisnika prikazuje se

forma za uređivanje njegovih podataka. Prilikom izmjene podataka nije moguće izmijeniti i lozinku, to se mijenja u odvojenoj formi.

The image shows a web form titled "Promjena lozinke - Niko Rudić". It features a password input field labeled "Lozinka" with a blue border and a toggle icon. Below the input field is a strength indicator bar and the text "Odaberite lozinku". Underneath, it says "Srednje jak" and "Zahtjevi". A list of requirements is provided: "Barem jedno malo slovo", "Barem jedno veliko slovo", "Barem jedan broj", and "Minimalno 8 znakova".

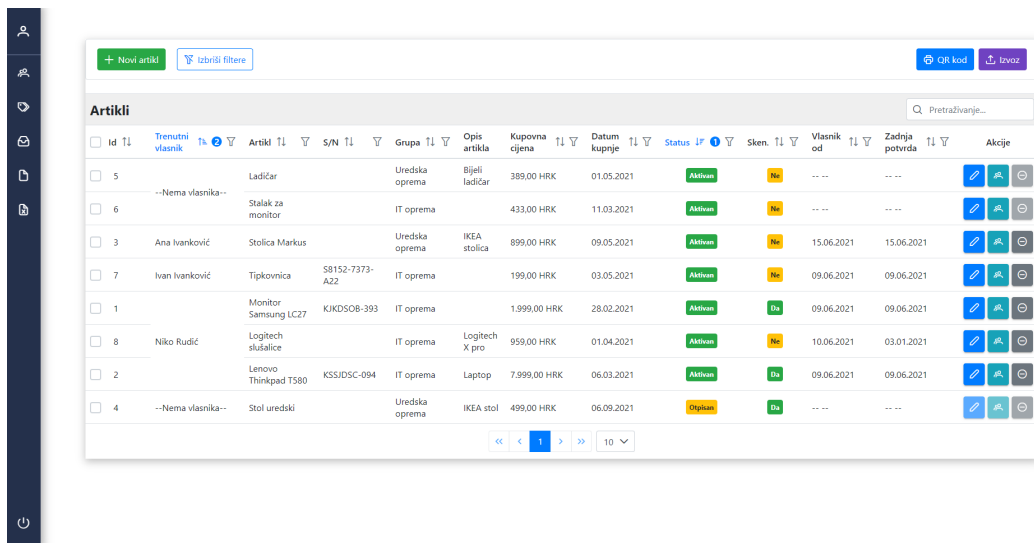
Slika 4.6: Forma za promjenu lozinke korisnika

Pritiskom na gumb "Promjena lozinke", otvara se forma za izmjenu lozinke koja zahtijeva unos nove lozinke, koja zadovoljava sve prikazane kriterije i ponavljanje te iste lozinke u polju ispod. Ako se obje lozinke podudaraju i zadovoljavaju navedene kriterije, lozinka je zamijenjena i njezin kriptografski sažetak je spremljen u bazu podataka.

The image shows a confirmation dialog box titled "Potvrda". It contains a warning icon and the text "Jeste li sigurni da želite deaktivirati korisnika **Niko Rudić**?". At the bottom, there are two buttons: "Ne" (No) with a blue 'X' icon and "Da" (Yes) with a blue checkmark icon.

Slika 4.7: Potvrda za deaktiviranje korisnika

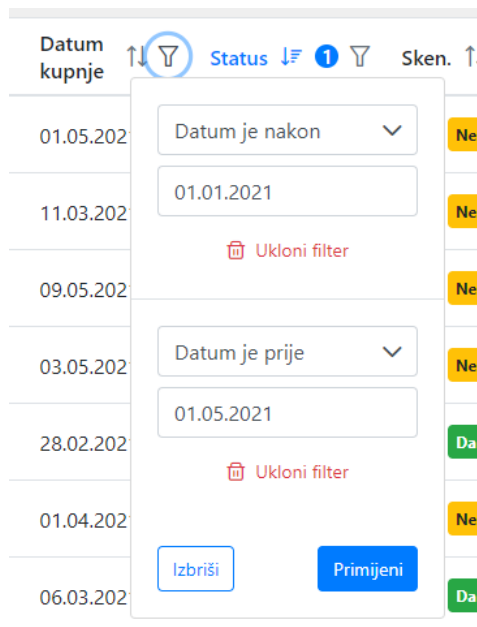
Deaktivacije korisnika se izvodi klikom na gumb "Deaktiviranje korisnika" prilikom čega se prikazuje poruka potvrde. Nakon potvrde provjerava se je li korisnik trenutni vlasnik artikala, ima li otvorenih zahtjeva za prijenos ili otpis, a ako je to slučaj, potrebno je prvo zahtjeve zatvoriti, artikle prenijeti ili otpisati, nakon čega je moguće deaktivirati korisnika.



| Id | Trenutni vlasnik | Artikl | S/N | Grupa | Opis artikla | Kupovna cijena | Datum kupnje | Status | Sken. | Vlasnik od | Zadnja potvrda | Akcije |
|----|-------------------|----------------------|----------------|----------------|----------------|----------------|--------------|---------|-------|------------|----------------|---------------------------|
| 5 | --Nema vlasnika-- | Ladičar | | Uredska oprema | Bijeli ladičar | 389,00 HRK | 01.05.2021 | Aktivan | Ne | -- | -- | [edit] [delete] [refresh] |
| 6 | | Stalac za monitor | | IT oprema | | 433,00 HRK | 11.03.2021 | Aktivan | Ne | -- | -- | [edit] [delete] [refresh] |
| 3 | Ana Ivanković | Stolica Markus | | Uredska oprema | IKEA stolica | 899,00 HRK | 09.05.2021 | Aktivan | Ne | 15.06.2021 | 15.06.2021 | [edit] [delete] [refresh] |
| 7 | Ivan Ivanković | Tipkovnica | S8152-7373-A22 | IT oprema | | 199,00 HRK | 03.05.2021 | Aktivan | Ne | 09.06.2021 | 09.06.2021 | [edit] [delete] [refresh] |
| 1 | | Monitor Samsung LC27 | KJKDSOB-393 | IT oprema | | 1.999,00 HRK | 28.02.2021 | Aktivan | Da | 09.06.2021 | 09.06.2021 | [edit] [delete] [refresh] |
| 8 | Niko Rudić | Logitech slušalice | | IT oprema | Logitech X pro | 959,00 HRK | 01.04.2021 | Aktivan | Ne | 10.06.2021 | 03.01.2021 | [edit] [delete] [refresh] |
| 2 | | Lenovo Thinkpad T580 | KSSJDS-094 | IT oprema | Laptop | 7.999,00 HRK | 06.03.2021 | Aktivan | Da | 09.06.2021 | 09.06.2021 | [edit] [delete] [refresh] |
| 4 | --Nema vlasnika-- | Stol uredski | | Uredska oprema | IKEA stol | 499,00 HRK | 06.09.2021 | Opisani | Da | -- | -- | [edit] [delete] [refresh] |

Slika 4.8: Prikaz tablice svih artikala

Slika 4.8 prikazuje tablicu svih artikala. Artikli su po zadanim postavkama poređani prema statusu (prvo aktivni, a onda neaktivni), pa zatim grupirani po vlasniku. Na taj način prvo se prikazuju artikli bez vlasnika (novi artikli). Isto kao i kod korisnika, klikom na gumb "Novi artikl", otvara se forma za dodavanje novog artikla. Također postoje akcije za uređivanje artikla, prijenos vlasništva i otpis vlasništva.



Datum kupnje ↑ Status ↓ 1 Sken. ↑

Datum je nakon ▼

01.01.2021

Ukloni filter

Datum je prije ▼

01.05.2021

Ukloni filter

Izbriši Primijeni

Slika 4.9: Primjer naprednog filtriranja tablice

Svaki stupac se može napredno filtrirati prema svojim vrijednostima, primjerice:

Grupa - naziv grupe je "IT oprema" + Datum kupnje je prije 01.05.2021. i nakon 01.01.2021. + Status je "Aktivan" + kupovna cijena je veća od 1000. Tako filtrirane artikle je moguće izvesti u .csv datoteku. Na taj način lakše dolazimo do raznih podataka o inventaru firme koji mogu poslužiti u naručivanju nove opreme ili zamijeni postojeće, stare opreme.

Gornja margina: 10.70 mm

Lijeva margina: 9.75 mm

Visina naljepnice: 38.00 mm

Širina naljepnice: 38.00 mm

Broj naljepnica po stranici: 65

Preskoči: 0

Ispis stranice

Ispod se nalazi pregled stranice za ispis.
Svaka stranica se može individualno prilagođavati.
Da biste ispisali, pritisnite Ctrl+P ili kliknite Ispis stranice.

Bitbox Markus

Tiplixton
18180-7171-422

Monitor Samsung LCD
P04C1008-010

Logitech slušalice

Lenovo Thinkpad T580
P 840010-004

Slika 4.10: Ispis QR kodova odabranih artikala

Artikle je moguće odabrati klikom na okvir lijevo od stupca Id, a za tako odabrane artikle je moguće ispisati QR kod klikom na gumb "QR kod" koji preusmjerava na prikaz ispisa. S obzirom na to da je ispis QR kodova zamišljen kao ispis na naljepnice koje mogu biti raznih veličina, omogućeno je podešavanje ispisa i veličine QR kodova s obzirom na veličinu i količinu naljepnica.

U web sučelju postoje tablice zahtjeva za prijenos i zahtjeva za otpis vlasništva. Obje tablice su gotovo identične pa je dovoljno prikazati samo jednu. Isto kao i prije, sve stupce je moguće sortirati i filtrirati kao i preuzeti čitavu filtriranu tablicu u .csv formatu.

| Id | Razlog | Datum zahtjeva | Status | Prihv. | Preth. vlasnik | Email preth. vl. | Novi vlasnik | Email novog vl. | Artikel | S/N | Akcije |
|----|-----------------------------------|-------------------|-----------|--------|-------------------|----------------------|---------------|----------------------|----------------------|-------------|---|
| 29 | --Pokrenuo administrator-- Razlog | 05.06.2021 | Neaktivan | Da | --Nema vlasnika-- | | Ana Ivanković | ana.ivankovic@fer.hr | Stolica Markus | -- | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |
| 7 | --Pokrenuo administrator-- Razlog | 09.06.2021 | Neaktivan | Da | --Nema vlasnika-- | | Ana Ivanković | ana.ivankovic@fer.hr | Monitor Samsung LC27 | KJKDSOB-393 | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |
| 8 | --Pokrenuo administrator-- Razlog | 09.06.2021 | Neaktivan | Da | --Nema vlasnika-- | | Ana Ivanković | ana.ivankovic@fer.hr | Monitor Samsung LC27 | KJKDSOB-393 | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |
| 9 | --Pokrenuo administrator-- Razlog | 09.06.2021 | Neaktivan | Da | --Nema vlasnika-- | | Ana Ivanković | ana.ivankovic@fer.hr | Monitor Samsung LC27 | KJKDSOB-393 | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |
| 10 | --Pokrenuo administrator-- Razlog | 09.06.2021 | Neaktivan | Da | --Nema vlasnika-- | | Niko Rudić | niko.rudic@fer.hr | Lenovo Thinkpad T580 | KSSJOSC-094 | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |
| 11 | --Pokrenuo administrator-- Razlog | 09.06.2021 | Neaktivan | Da | --Nema vlasnika-- | | Niko Rudić | niko.rudic@fer.hr | Stolica Markus | -- | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |
| 12 | --Pokrenuo administrator-- Razlog | 09.06.2021 | Neaktivan | Da | --Nema vlasnika-- | | Niko Rudić | niko.rudic@fer.hr | Stol uredski | -- | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |
| 13 | --Pokrenuo administrator-- Razlog | 09.06.2021 | Neaktivan | Da | Niko Rudić | niko.rudic@fer.hr | Ana Ivanković | ana.ivankovic@fer.hr | Stolica Markus | -- | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |
| 14 | --Pokrenuo administrator-- Razlog | 09.06.2021 | Neaktivan | Da | --Nema vlasnika-- | | Niko Rudić | niko.rudic@fer.hr | Ladičar | -- | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |
| 15 | --Pokrenuo administrator-- Razlog | 09.06.2021 | Neaktivan | Da | Ana Ivanković | ana.ivankovic@fer.hr | Niko Rudić | niko.rudic@fer.hr | Stolica Markus | -- | <input checked="" type="checkbox"/> Prihvati <input type="checkbox"/> Odbij |

Slika 4.11: Prikaz tablice svih zahtjeva za prijenos

U tablici se nalaze svi zahtjevi, pa i oni neaktivni (jer su već odobreni ili odbijeni) kako bi ostao zapis vlasništva.

Svi aktivni zahtjevi se mogu prihvatiti ili odbiti. Ako se odbiju, zahtjev dobiva status *neaktivan* i *prihvaćen: ne*. Zahtjeve koji se prihvate odlaze u stanje *aktivan* i *prihvaćen: da*. Prihvaćeni zahtjevi za prijenos ažuriraju zapis u tablici *xref_user_has_item* s kojim se ažurira vlasnik artikla navedenog u zahtjevu. Kod prihvaćanja zahtjeva za otpis uklanjanje se zapis u navedenoj tablici kako bi se uklonio vlasnik artikla, a zatim se ažurira artikal (tablica *item*) u status *neaktivan*.

4.4. Mobilna aplikacija

Mobilna aplikacija ostvarena je pomoću *React Native* radnog okvira za *JavaScript* i *Expo* radnog okvira za *React Native*.

Mobilna aplikacija je namijenjena zaposlenicima tvrtke, ali i administratorima sustava koji u mobilnoj aplikaciji imaju ulogu i sve mogućnosti običnog korisnika. Pri likom otvaranja aplikacije prikazuje se zaslon za prijavu. Registracija na mobilnoj aplikaciji nije moguća, već nju izvršava isključivo administrator u web sučelju. Za prijavu u sustav potrebno je upisati ispravnu email adresu i lozinku.

Sustav za praćenje
inventurne liste

Prijava

Email adresa

Unesite lozinku

Prijava

Slika 4.12: Forma za prijavu korisnika

Nakon prijave, pojavljuje se prikaz svih zaduženja korisnika zajedno s navigacijom na dnu. Sva zaduženja koja su u zadnjih mjesec dana skenirana i na taj način im je potvrđeno njihovo vlasništvo, su označeni s kvačicama, a oni koji nisu sa znakom X.

Zaduženja

| | |
|---|---|
| ✓ | Monitor Samsung LC27 S/N: : KJKDSOB-393 Zadnja potvrda: 2021-06-09 |
| ✗ | Lenovo Thinkpad T580 S/N: : KSSJDSC-094 Zadnja potvrda: 2021-01-03 |
| ✗ | Logitech slušalice S/N: : -- Nema S/N -- Zadnja potvrda: 2021-01-03 |

Home Prijenos Scan QR Otpis Izlaz

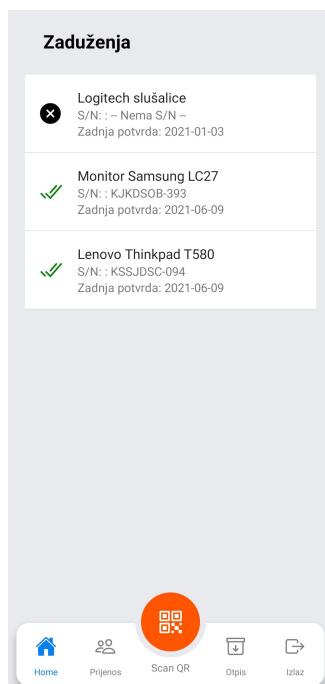
Slika 4.13: Prikaz liste zaduženja korisnika

Potvrda vlasništva artikla se izvodi skeniranjem QR koda zalijepljenim na navedeni artikl kako je prikazano na slici 4.14.



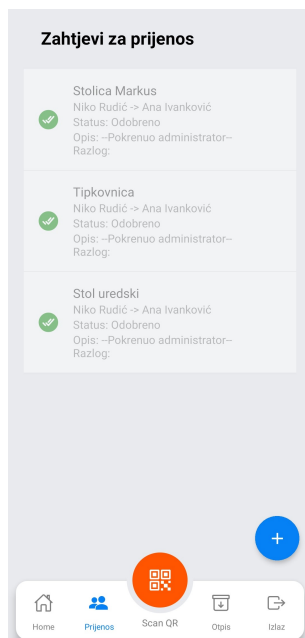
Slika 4.14: Prikaz skeniranja QR koda

Nakon skeniranja QR koda ažurira se status skeniranog artikla što je vidljivo na prikazu zaduženja.



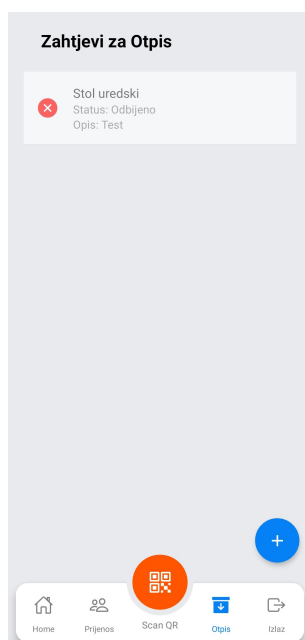
Slika 4.15: Prikaz ažurirane liste zaduženja

Osim liste zaduženja moguće je pregledati listu svih korisnikovih zahtjeva za prijenos zajedno s njihovim statusom i razlogom. Svi neaktivni zahtjevi su zasivljeni. Klikom na gumb + u donjem desnom kutu, otvara se forma za unos novog zahtjeva za prijenos. Slično kao zahtjevi za prijenos, postoje i zahtjevi za otpis koji omogućuju



Slika 4.16: Prikaz korisnikovih zahtjeva za prijenos

otpis vlasništva artikla i koji implementiraju iste funkcionalnosti.



Slika 4.17: Prikaz korisnikovih zahtjeva za otpis

5. Zaključak

Zadatak ovog rada bilo je osmisliti, oblikovati i proizvesti programsko rješenje koje olakšava praćenje, prenošenje i otpisivanje inventara firme koristeći QR kodove zalijepljene na predmet. Dio zadatka je bio i odabrati odgovarajuće tehnologije i prikladnu bazu podataka, što je zahtijevalo dosta truda i istraživanja. Kako bi odabrao odgovarajuću tehnologiju i platformu, bilo je potrebno i većinski osmisliti strukturu čitavog sustava te odabrati odgovarajuću strukturu izrade programskog rješenja i dobro definirati sve potrebne zahtjeve. Tek tada je bilo moguće napraviti kvalitetan i prikladan odabir tehnologije i programskih jezika. Prilikom odabira gore navedenog, potrebno je bilo obratiti pažnju i na mogućnost buduće nadogradnje i unaprjeđenja sustava što je ostvareno koristeći REST arhitekturu.

Rad na ovom sustavu bio je zanimljiv, poučan, ali i iznimno zahtjevan pothvat. S većinom korištenih tehnologija nisam bio upoznat što je zahtijevalo intenzivno istraživanje i proučavanje dokumentacija. No navedene tehnologije su namjerno odabrane, ne samo zato što odgovaraju zahtjevima, već zato što se radi o trenutno iznimno popularnim tehnologijama koje se koriste u širokoj primjeni te ih je bilo zadovoljstvo učiti i koristiti u praksi.

Uspješno su ostvareni svi zahtjevi i tražene funkcionalnosti ovog programskog rješenja te sustav doista olakšava praćenje inventara tvrtke. Osim lakšeg praćenja stanja sustav implementira napredno pretraživanje svih tablica što proširuje primjenu ovakvog sustava. Primjerice, u slučaju odluke o zamjeni svih računala i laptopa starijih od 3 godine, uz samo nekoliko klikova, može se dobiti lista čitavog inventara koji spada u tu grupu, zajedno s njihovim trenutnim vlasnicima. Nad sustavom je provedeno tek preliminarno testiranje te u slučaju njegove implementacije u stvarni sustav neke tvrtke, preporučuje se napraviti intenzivno testiranje.

LITERATURA

- [1] PostgreSQL. <https://www.postgresql.org/about/>, . Pristupljeno 29.svibnja.2021.
- [2] Postman. <https://www.postman.com/>, . Pristupljeno 03.travnja.2021.
- [3] Wikipedia: besplatna enciklopedija. Model-view-viewmodel. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>, . Pristupljeno 30.svibnja.2021.
- [4] Wikipedia: besplatna enciklopedija. Node.js. <https://en.wikipedia.org/wiki/Node.js>, . Pristupljeno 30.svibnja.2021.
- [5] Wikipedia: besplatna enciklopedija. Prime vue. <https://primefaces.org/primevue/>, . Pristupljeno 03.travnja.2021.
- [6] Wikipedia: besplatna enciklopedija. Qr code. https://en.wikipedia.org/wiki/QR_code, . Pristupljeno 30.svibnja.2021.
- [7] Wikipedia: besplatna enciklopedija. Representational state transfer. https://en.wikipedia.org/wiki/Representational_state_transfer, . Pristupljeno 29.svibnja.2021.
- [8] Wikipedia: besplatna enciklopedija. Multitier architecture. https://en.wikipedia.org/wiki/Multitier_architecture, . Pristupljeno 29.svibnja.2021.
- [9] Wikipedia: besplatna enciklopedija. Vue.js. <https://en.wikipedia.org/wiki/Vue.js>, . Pristupljeno 30.svibnja.2021.
- [10] Facebook. React native. <https://reactnative.dev/>. Pristupljeno 22.svibnja.2021.

- [11] Adranik Keshishyan. Node.js event loop architecture. <https://medium.com/preezma/node-js-event-loop-architecture-go-deeper-node-core-c96b4cec7aa4>.
Pristupljeno 1.lipnja.2021.
- [12] Hrvatski sabor. Zakon o računovodstvu, Narodne novine br. 109/2007 (2007, listopad). https://narodne-novine.nn.hr/clanci/sluzbeni/2007_10_109_3174.html. Pristupljeno 25.svibnja.2021.
- [13] Tan Jin Soon. *QR code*. Synthesis Journal, 2008.
- [14] Besant Technologies. What is express.js? <https://www.besanttechnologies.com/what-is-expressjs>. Pristupljeno 1.lipnja.2021.
- [15] računalne i inteligentne sustave FER Zavod za elektroniku, mikroelektroniku. Arhitektura zasnovana na događajima. Predavanje 5. Arhitektura programske potpore, Programsko inženjerstvo 2020./2021.
- [16] FER Zavod za telekomunikacije. Prikaz posrednika s priručnim spremištem. Slika preuzeta iz predavanja 5., Umrežene višemedijske usluge 2020./2021.

Sustav za praćenje uredske inventurne liste

Sažetak

Ovaj rad obradio je temu razvoja sustava za praćenje uredske inventurne liste. U početku su objašnjeni svi zahtjevi i ograničenja sustava, nakon čega je detaljno objašnjena korištena troslojna arhitektura za pristup, obradu i prikaz podataka. U sljedećem poglavlju su navedene i opisane sve bitne korištene tehnologije. Na kraju je opisan i prikazan čitav sustav rastavljen na njegove ključne dijelove te je prikazan konačan izgled web sučelja i mobilne aplikacije.

Ključne riječi: aplikacija, JavaScript, NodeJS, VueJS, React Native, PostgreSQL, QR kod

Office inventory tracking system

Abstract

This thesis addressed the topic of developing an office inventory tracking system. Initially, all system requirements and limitations were explained, after which the three-tier architecture used for data access, processing, and display was explained in detail. The next section lists and describes all the essential technologies used. Finally, the whole system is described and presented, separated into its key parts, and the final appearance of the web and mobile application is shown.

Keywords: application, JavaScript, NodeJS, VueJS, React Native, PostgreSQL, QR code