**VISOKA TEHNIČKA ŠKOLA**

**STRUKOVNIH STUDIJA**

**S** **U B O T I C A**

**PROJEKAT**

**iz predmeta Integrisani web sistemi**

**STUDENT MENTOR**

**Jevanđel Đurić, br.indexa:26121057 dr Zlatko Čović**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SUBOTICA 2023. god.**

**SADRŽAJ**

[OPIS ZADATKA...........................................................................................................3](#Opis_zadatka)

[REALIZACIJA ZADATKA..........................................................................................4](#Realizacija_zadatka)

[STRUKTURA BAZE PODATAKA..............................................................................6](#Baza)

[OPIS FUNKCIONALNOSTI.........................................................................................7](#Funkcionalnosti)

[KORIŠĆENA LITERATURA.....................................................................................13](#Literatura)

**OPIS ZADATKA**

Cilj projekta je bio kreirati nativnu aplikaciju koja se nadovezuje na projekat iz naprednog web programiranja. Aplikacija komunicira sa [web serverom](http://first.stud.vts.su.ac.rs/nwp/) putem API poziva. Svaki API poziv sadrži odgovarajući statusni kod, poruku i podatke(zavisno od zahteva). Dozvoljeno je slanje POST, GET, PUT, PATCH i DELETE zahteva da bi se zadovoljio CRUD projektni zahtev. Zahtevi su smešteni u različite direktorijum zavisno čije putanje moramo da ispravno navedemo. Ukoliko je putanja do nekog direktorijuma netačna dobićemo odgovarajuću poruku i moguće putanje do željenog direktorijuma u Allow zaglavnju. Svi pozivi su testirani u swagger alatu i može im se pristupiti putem [linka](https://first.stud.vts.su.ac.rs/nwp/documentation/swagger/). Pozivi se nalaze i u eksportovanoj postman kolekciji u kojoj je u svakom primeru ispod poziva sačuvan odgovor, poslani podaci i tip poziva.

**REALIZACIJA ZADATKA**

#RewriteRool root(api) direktorijuma

^/?([0-9a-z\_-]\*)/?([0-9a-z\_-]\*)?/?([0-9]\*)?$ router.php?resource=$1 [NC,L]

//obrada na php strani

$resource=$\_GET['resource']??'';

if (!in\_array($resource,$resources)){

http\_response\_code(404);

header("Valid-Resources: account, events, presentsInvites");

echo json\_encode([

"message" => "Resource '$resource' is not accessible!"

]);

exit();

}

//slanje podataka react native

const fetchData = async () => {

try {

const response = await fetch(

'https://first.stud.vts.su.ac.rs/nwp/api/events/',

{

method: 'POST',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({ email: email })

}

);

const result = await response.json();

if (result.status === 200) {

setEvents(result.data);

// await fetchData();

} else if (result.status === 404){

setEvents([]);

}

setLoading(false);

} catch (error) {

console.error('Error updating data:', error);

Alert.alert('Error', 'Failed to update data.');

}

};

//preuzimanje e-mail adrese iz “globalnog(asinhronog) skladišta”

const getData = async (key) => {

try {

const value = await AsyncStorage.getItem(key);

if (value !== null) {

// Data found

setEmail(JSON.parse(value));

} else {

// Data not found

console.log('No data found for the key: ', key);

}

} catch (error) {

console.error('Error retrieving data: ', error);

}

};

//ukoliko korisnik uđe u aplikaciju kad nije povezan sa wi-fi mrežom //moći će samo da pristupi početnom ekranu i ekranu “info”

useEffect(() => {

const unsubscribe = NetInfo.addEventListener((state) => {

setIsConnected(state.isConnected);

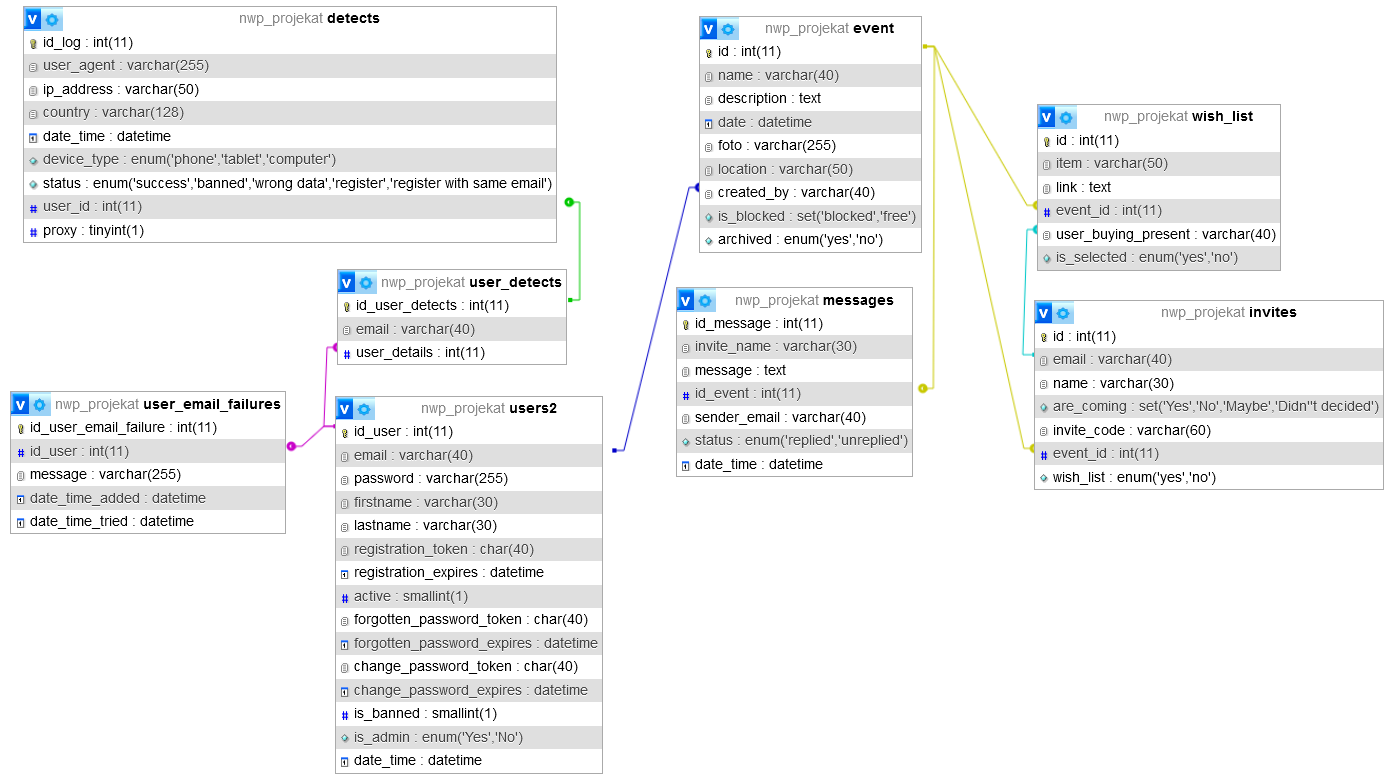
});

return () => unsubscribe();

}, []);

**STRUKTURA BAZE PODATAKA**

(slika baze sa localhosta)



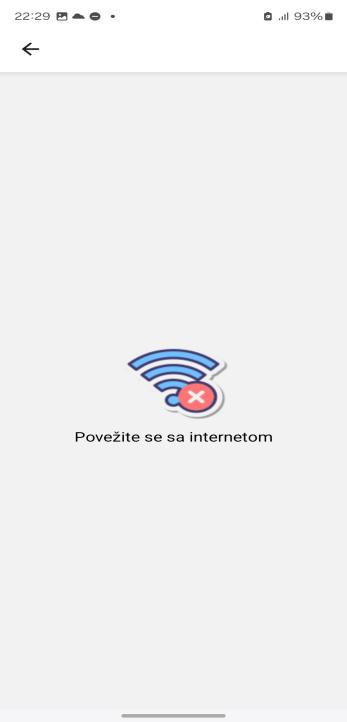
**O****PIS FUNKCIONALNOSTI**

Registracija

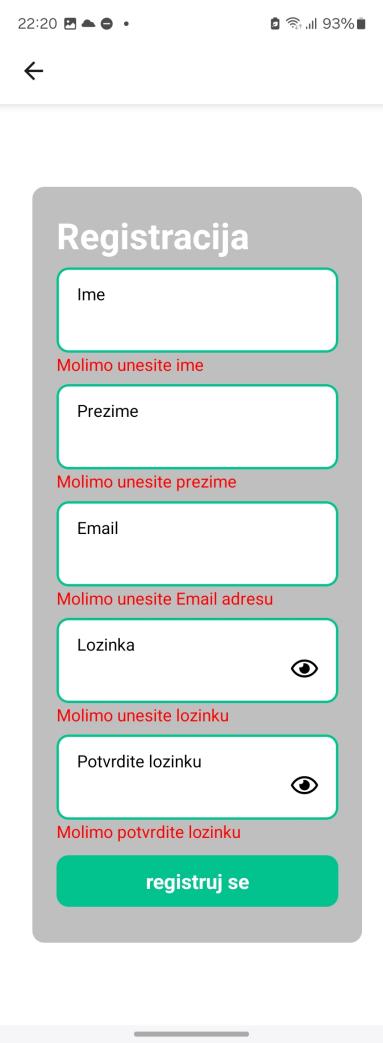
Potrebno je da smo povezani sa WiFi mrežom i da odaberemeo ikonicu u donjem levom uglu ekrana



Ukoliko nemamo interneta prikazaće nam se...

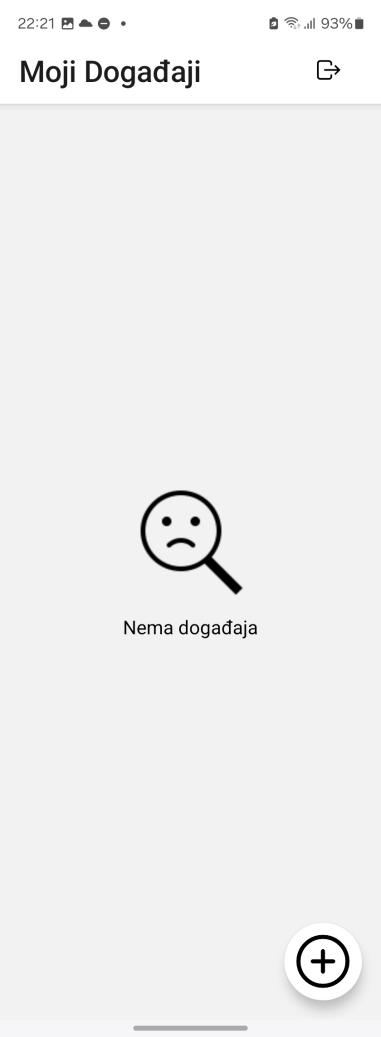


Ukoliko smo povezani sa internetom na sledećem ekranu prikazaće nam se login i ispod njega opcija registracija. Nakon što je pritisnemo odvešće nas na formu za registraciju koju je potrebno popuniti validnim podacima



Kreiranje događaja

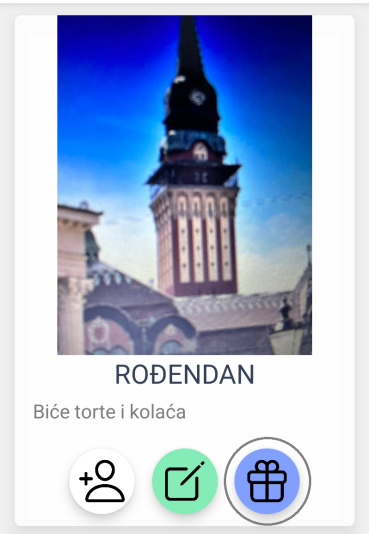
Nakon uspešnog logovanja potrebno je izabrati opciju u donjem desnom uglu ekrana u obliku plusa za dodavanje novog događaja i popuniti podatke na narednoj stranici.





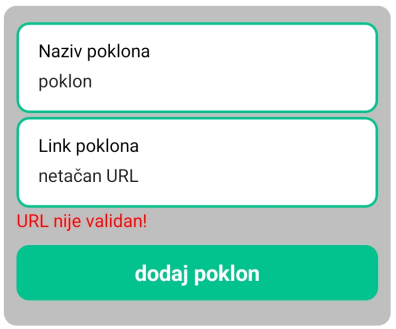
Dodavanje poklona

Nakon uspešnog kreiranja događaja poklon je moguće dodati pritiskom na opciju u donjem desnom uglu događaja i popunjavanjam obrasca na narednoj stranici.



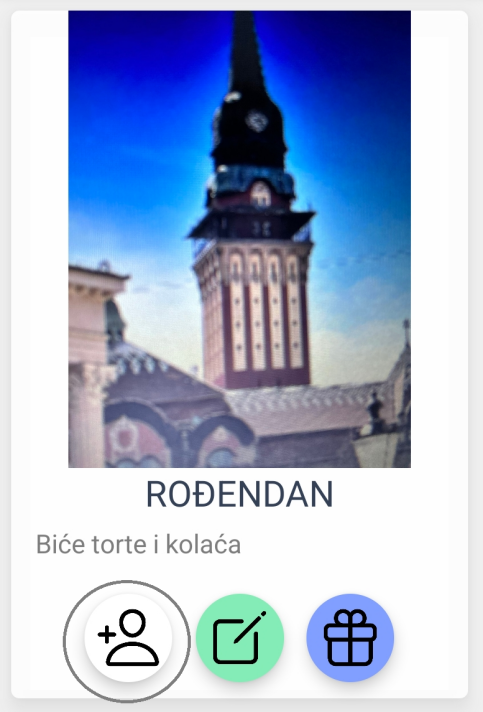


Ukoliko URL nije validan prikazuje se odgovarajuće obaveštenje ispod polja

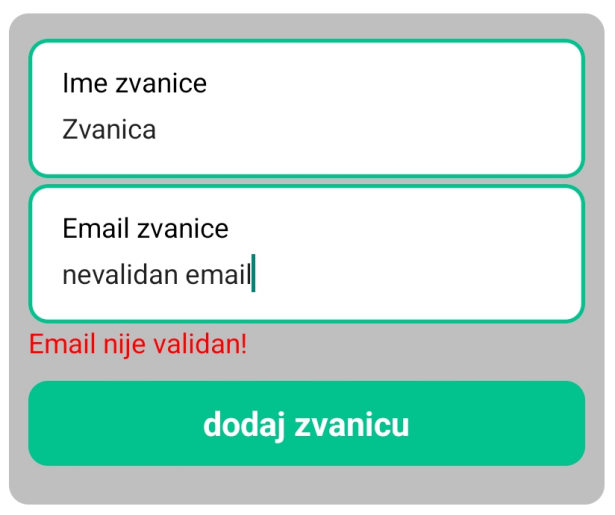


Dodavanje korisnika

Potrebno je da na događaju pritisnemo donju levu opciju

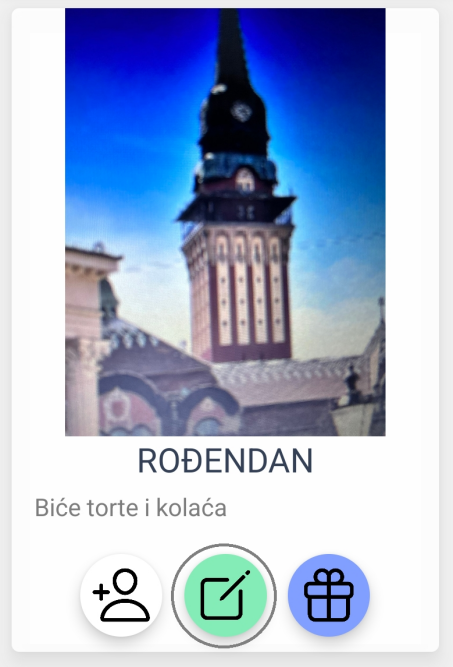


Kao i kod dodavanja poklona potrebno je da popunimo obrazac validnim podacima

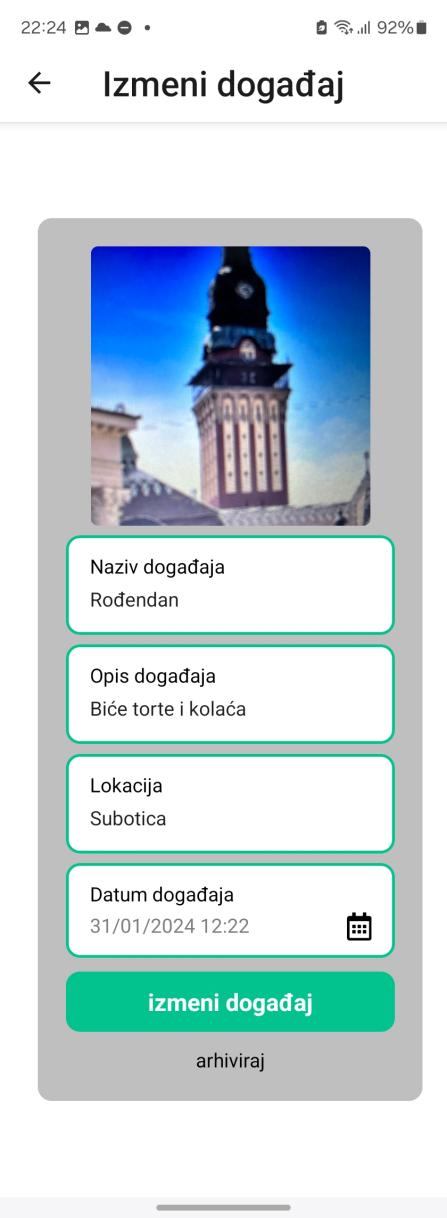


Arhiviranje događaja

Potrebno je da pritisnemo ikonicu u sredini događja i



na dnu narednog ekrana će se pojaviti opcija arhiviraj.



**KORIŠĆENA LITERATURA**

Uglavnom sam koristio web izvore sa navednih linkova datim redosledom:

W1. <https://people.vts.su.ac.rs/~chole>

W2. <https://stackoverflow.com/>

W3. <https://chat.openai.com>

W4.<https://github.com/kael-shipman/swagger-ui-plugins/tree/hierarchical-tags/packages/hierarchical-tags>

W5. <https://dev.to/chinmaymhatre/how-to-generate-apk-using-react-native-expo-kae>

W6. <https://icons8.com/icons>

W7. <https://ionic.io/ionicons>

W8. <https://reactnative.dev/>