

Univerzitet u Beogradu  
Fakultet organizacionih nauka  
Laboratorija za elektronsko poslovanje



Web aplikacija studentske službe  
Projektni rad iz Internet tehnologija

Studenti:

Veljko Blagojević 0353/2019

Nikola Vujčić 0062/2019

Milan Stanković 1004/2020

U Beogradu, 2023.

# Korisnički zahtev

---

Potrebno je napraviti web aplikaciju studentske službe. Rad sa službom je moguće obaviti u ulozi studenta ili u ulozi administratora. Prilikom pokretanja web sajta, korisniku treba da se prikazuje web stranica i meni sa sledećim opcijama: Obaveštenja, Molba, Podaci o studentu, O fakultetu i Kraj Rada. Ukoliko je prijavljeni korisnik u funkciji student, njemu se prikazuju i dodatne opcije: Prijava ispita, Biranje predmeta, Uspešna polaganja, Neuspešna polaganja, Moji Predmeti i Analitika. Prijavljenom administratoru se prikazuju dodatne opcije: Upiši ocenu, Izmeni podatke o studentu i ADMIN sekcija.

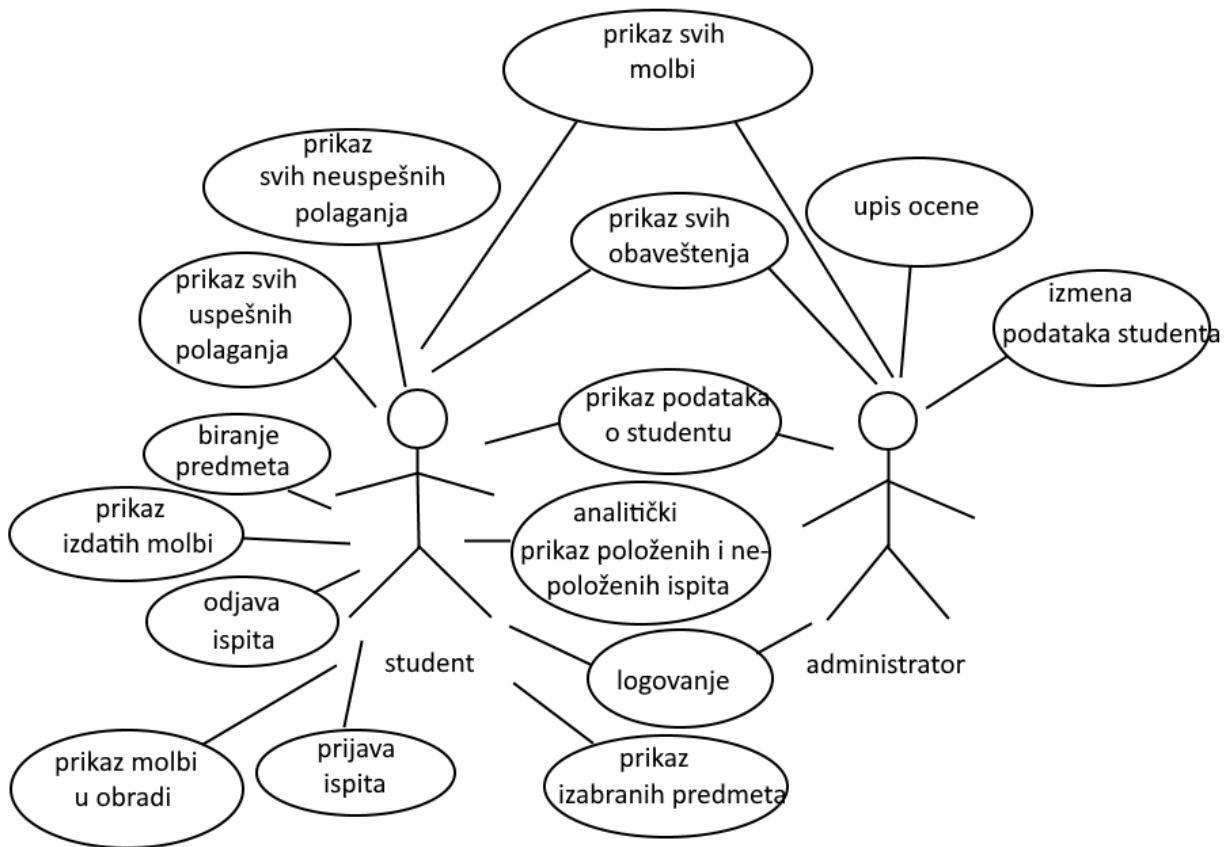
U sekciji obaveštenja su izlistana sva obaveštenja datog korisnika. Prijava ispita se obavlja klikom na odgovarajuće dugme, a odjava ispita se obavlja klikom na dugme x pored prijavljenog ispita, u istoimenoj tabeli. Izbor predmeta za slušanje treba biti implementiran na sledeći način: klikom na dugme "Potvrdi" se izabrani predmeti čuvaju. Potrebno je takođe ispisati i podatke o ukupnom broju ESP bodova, kao i prosečnoj oceni studenta u sekciji "Uspešna polaganja". Neuspešna polaganja studenta je neophodno evidentirati i prikazati tabelarno korisniku. U odeljku "Molbe" je potrebno omogućiti unos nove molbe studenta i prikazati Izdate molbe i molbe u obradi. Klikom na dugme "Potvrdi" se molba čuva u bazi podataka. Podaci o studentu koje treba prikazati su: broj indeksa, ime, prezime, username, ime roditelja, JMBG, mesto rođenja, lični email, studentski email, broj telefona, kao i slika studenta. Analitika treba da sadrži 2 dijagrama koji prikazuju odnos položenih ispita i neuspešnih polaganja, kao i raspodelu ocena po ispitima. O fakultetu treba da sadrži podatke o fakultetu kao i google map prikaz lokacije na kojoj se fakultet nalazi. Neophodno je omogućiti i upis ocene studenta administratoru. Administrator može i da izmeni podatke o studentu kojeg on izabere iz liste dostupnih studenata, i klikom na dugme "Potvrdi" se ažurirani podaci o studentu čuvaju u bazi podataka. Na kraju, neophodno je omogućiti administratoru pristup servisnim informacijama o mreži putem ADMIN sekcije.

# Model slučajeva korišćenja

---

U ovom sistemu prepoznajemo 15 slučajeva korišćenja:

1. Logovanje
2. Prikaz svih obaveštenja
3. Unos molbe
4. Prikaz molbi u obradi
5. Prikaz Izdatih molbi
6. Upis ocene
7. Prikaz podataka studenta
8. Izmena podataka studenta
9. Prijava ispita
10. Odjava ispita
11. Biranje predmeta
12. Prikaz svih uspešnih polaganja ispita
13. Prikaz svih neuspešnih polaganja ispita
14. Prikaz izabralih predmeta
15. Analitički prikaz položenih i nepoloženih ispita



Slika 1: Model slučajeva korišćenja

- **Prikaz svih obaveštenja** – korisniku su izlistana sva dostupna obaveštenja zasebno, na jednoj strani u celini
- **Unos molbe** – student bira iz combo-box-a tip molbe, unosi tekstualno obrazloženje za datu molbu, i klikom na dugme “Potvrdi” podnosi elektronsku molbu studentskoj službi
- **Prijava ispita** – studentu su izlistani dostupni ispiti u operativnoj memoriji. Ispiti su prikazani tabelarno. Pritisom na dugme “Prijava ispit”, student prebacuje dati ispit u sekciju “Prijavljeni ispiti”. Klikom na X dugme pored prijavljenog ispita, dati ispit se vraća u početnu tabelu. Baza podataka se ažurira na ispravan način nakon akcija studenta
- **Izmena podataka studenta** – Administratoru su tabelarno prikazani svi studenti učitani u operativnu memoriju. Izborom jednog studenta i klikom na dugme “Izmeni” otvara se stranica na kojoj su prikazani podaci o studentu koje je moguće izmeniti. Klikom na dugme “Potvrdi” se date izmene čuvaju u bazi podataka.
- **Prikaz svih uspešnih polaganja ispita** – Studentu su tabelarno prikazana sva uspešna polaganja ispita, prosečna ocena, kao i broj ostvarenih ESP bodova.  
itd.

# Ponašanje softverskog sistema

---

Radi detaljnijeg opisa ponašanja sistema pozvaćemo se na UML dijagrame sekvenci. S obzirom na obim rada, ovde su prikazani dijagrami sekvenci prvih 5 slučajeva korišćenja. Sve ovde pokazano analogno važi i za ostale slučajeve korišćenja.

NAZIV: Logovanje

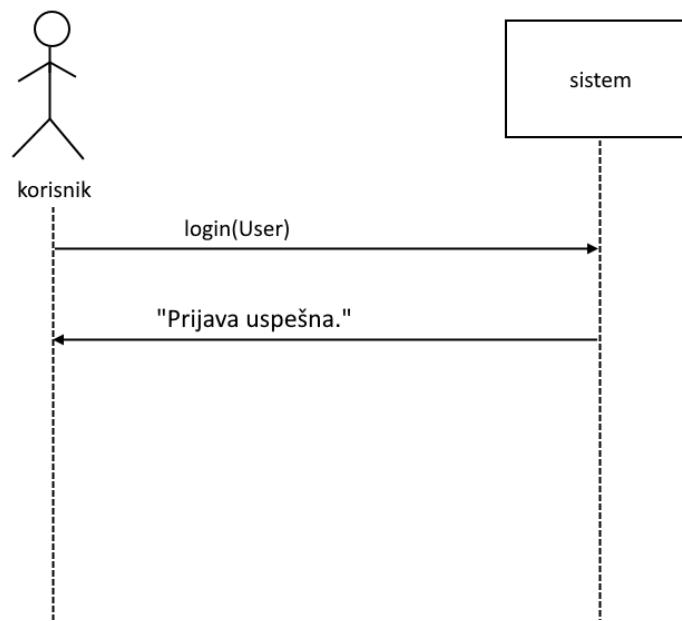
AKTERI: Korisnik (student ili admin)

UČESNICI: korisnik I sistem

PREDUSLOV: Sistem je uključen i korisniku je omogućeno prijavljivanje

SK1: Osnovni scenario:

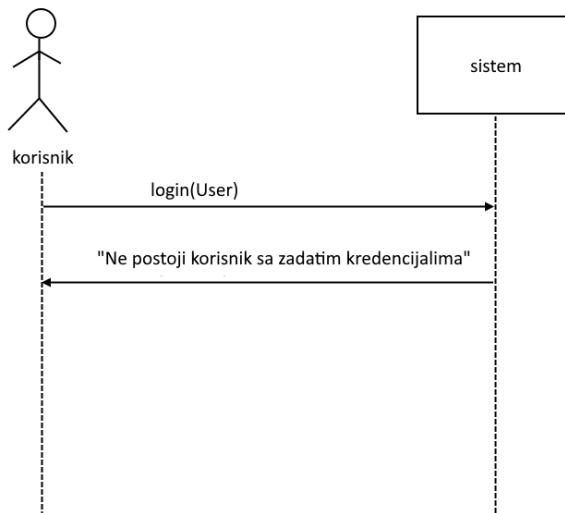
1. Korisnik pokreće formu za prijavu (APSO)
2. Sistem prikazuje korisniku formu (SO)
3. Korisnik poziva sistem da prijavi korisnika (APSO)
4. Sistem pronalazi korisnika i instancira objekat (SO)
5. Sistem prikazuje korisniku poruku "prijava uspešna"



Slika : Dijagram sekvenci glavnog scenarija SK1

Alternativni scenario:

- 5.1. Ukoliko sistem ne može da pronađe korisnika sa zadatim kredencijalima, ispisuje korisniku poruku "Ne postoji korisnik sa zadatim kredencijalima." (IA)



Slika : Dijagram sekvenci prvog alternativnog scenarija SK1

NAZIV: Prikaz svih obaveštenja

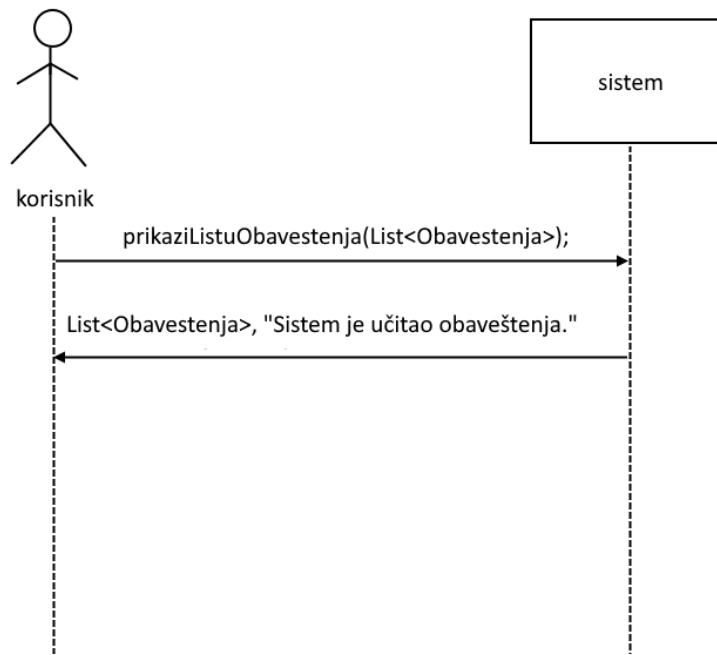
AKTERI: Korisnik (student ili admin)

UČESNICI: korisnik I sistem

PREDUSLOV: Sistem je uključen i prikazana je forma za rad sa klijentom.

SK2: Osnovni scenario:

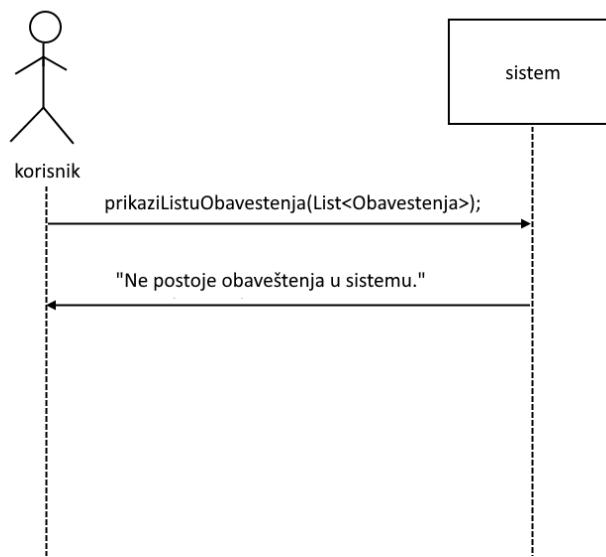
1. Korisnik poziva sistem da prikaže sva obaveštenja (APSO)
2. Sistem pronalazi sva obaveštenja (SO)
3. Sistem prikazuje korisniku obaveštenja i poruku "Sistem je učitao obaveštenja"



Slika : Dijagram sekvenci glavnog scenarija SK2

Alternativni scenario:

- 3.1 Ukoliko sistem ne može da pronađe obaveštenja, ispisuje korisniku poruku "Ne postoje obaveštenja u sistemu." (IA)



Slika : Dijagram sekvenci prvog alternativnog scenarija SK2

NAZIV: Unos Molbe

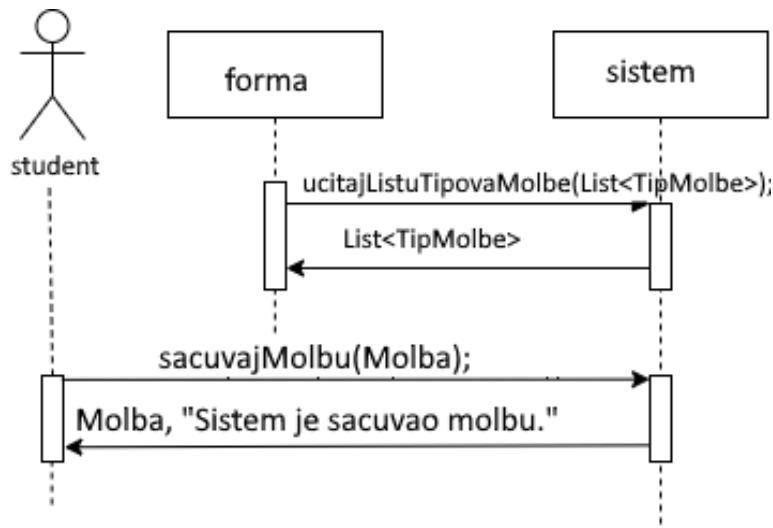
AKTERI: student

UČESNICI: student I sistem

PREDUSLOV: Sistem je uključen i prikazana je forma za rad sa studentom. Učitana je lista tipova molbe

SK3: Osnovni scenario:

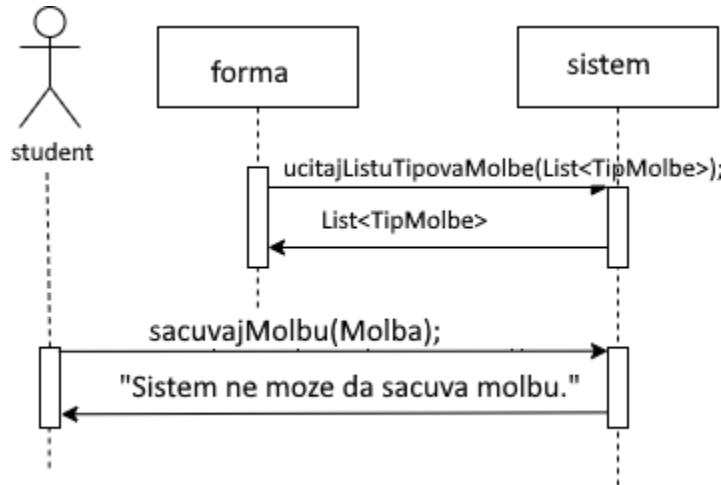
1. Korisnik poziva sistem da sačuva unete podatke o molbi (APSO)
2. Sistem čuva molbu i prikazuje korisniku poruku "Sistem je sačuвао molbu.".



Slika : Dijagram sekvenci glavnog scenarija SK3

Alternativni scenario:

2.1 Ukoliko sistem ne može da sačuva molbu, ispisuje korisniku poruku "Sistem ne može da sačuva molbu." (IA)



Slika : Dijagram sekvenci prvog alternativnog scenario SK3

NAZIV: Prikaz molbi u obradi

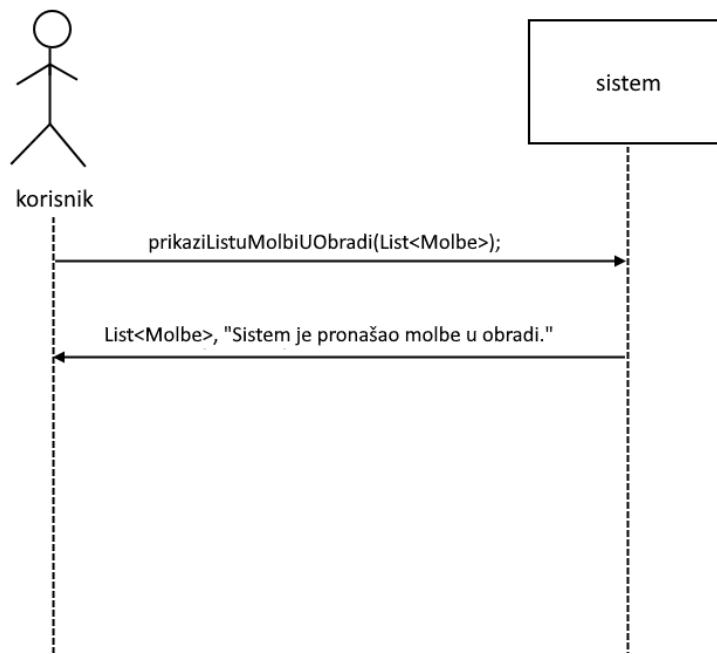
AKTERI: student

UČESNICI: student | sistem

PREDUSLOV: Sistem je uključen i prikazana je forma za rad sa studentom.

SK3: Osnovni scenario:

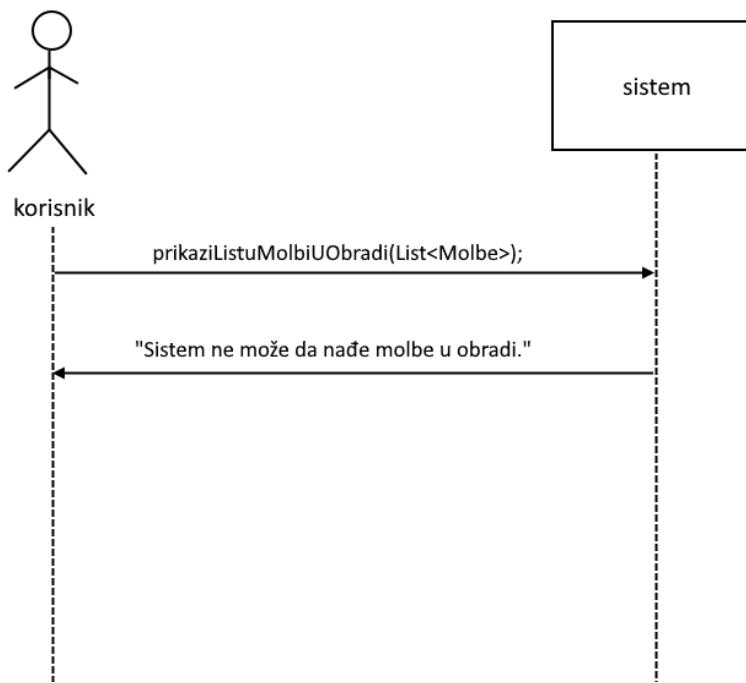
1. Korisnik poziva sistem da prikaže sve molbe u obradi (APSO)
2. Sistem nalazi sve molbe u obradi (SO)
3. Sistem prikazuje korisniku sve molbe u obradi i poruku "Sistem je pronašao molbe u obradi."



Slika : Dijagram sekvenci glavnog scenarija SK4

Alternativni scenario:

2.1 Ukoliko sistem ne može da nađe molbe u obradi, ispisuje korisniku poruku "Sistem ne može da nađe molbe u obradi." (IA)



Slika : Dijagram sekvenci prvog alternativnog scenarija SK4

NAZIV: Prikaz izdatih molbi

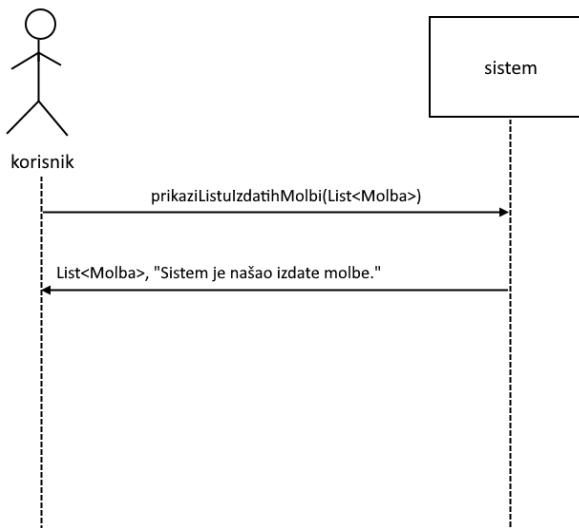
AKTERI: student

UČESNICI: student I sistem

PREDUSLOV: Sistem je uključen i prikazana je forma za rad sa studentom.

SK3: Osnovni scenario:

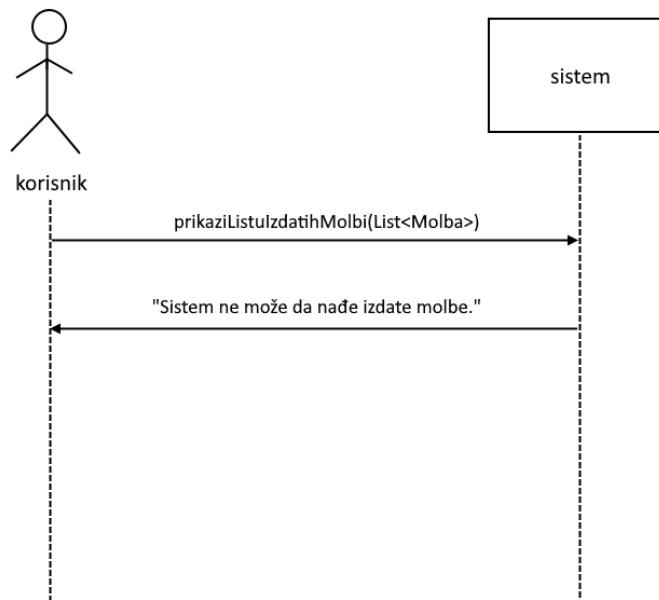
1. Korisnik poziva sistem da prikaže sve izdate molbe (APSO)
2. Sistem nalazi sve izdate molbe (SO)
3. Sistem prikazuje korisniku sve izdate molbe i poruku "Sistem je našao izdate molbe.".



Slika : Dijagram sekvenci glavnog scenarija SK5

Alternativni scenario:

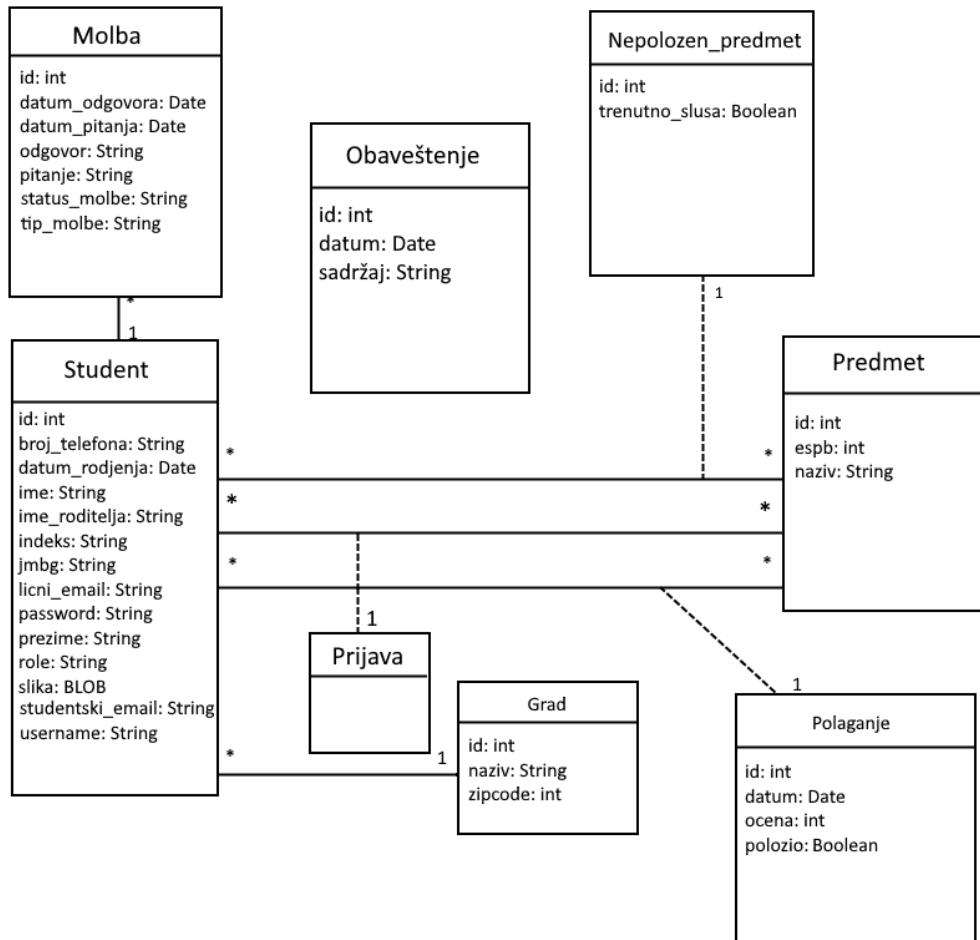
2.1 Ukoliko sistem ne može da nađe izdate molbe, ispisuje korisniku poruku "Sistem ne može da nađe izdate molbe." (IA)



Slika : Dijagram sekvenci prvog alternativnog scenarija SK5

# Struktura softverskog sistema

Da bismo detaljnije opisali strukturu softverskog sistema, koristimo sledeći dijagram klasa:



Slika 2: Dijagram klasa sistema

- Student tabela čuva relevantne podatke o studentima (i administratorima)
- Prijava je tabela koja označava ispit koji je prijavljen od strane studenta
- Predmet je tabela koja sadži relevantne podatke o predmetima
- Grad je tabela koja ima vezu sa studentom, koja sadži relevantne podatke o rodnom mestu studenta
- Polaganje je asocijativna klasa Studenta i Predmeta (isto kao i Prijava), koja sadži podatke o uspešnosti polaganja, oceni i datumu polaganja ispita

- Nepoložen predmet sadži podatke o tome da li student sluša predmet ili ne (asocijativna klasa studenta i predmeta)
- Obaveštenje je klasa koja čuva sve relevantne podatke koje je neophodno prikazati svim studentima

# API Specifikacija

---

Opis funkcije	Registracija
HTTP metoda	POST
URL	/register
URL parametri	(nema)
HTTP body parametri	JSON objekat sa atributima firstname[string], lastname[string], email[string], username[string], password[string] Primer: { "Ime": "Pera", "Prezime" : "Peric", "Email" : "example@gmail.com", "username" : "ppetar", "password": "123abc" }
Format HTTP body parametara	application/json
Izlazni parametri	
Format izlaznih parametara	application/json

Opis funkcije	Autentikacija
HTTP metoda	POST
URL	/authenticate
URL parametri	(nema)
HTTP body parametri	JSON objekat sa atributima username[string], password[string]. Primer: { "username": "ppetar", "password": "123abc" }
Format HTTP body parametara	application/json

Izlazni parametri	(nema)
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje grada iz baze po id-ju
HTTP metoda	GET
URL	/api/v1/gradovi
URL parametri	id[long]
HTTP body parametri	
Format HTTP body parametara	application/json
Izlazni parametri	<p>JSON objekat tipa grad</p> <p>Primer:</p> <pre>{   "id": 1,   "naziv": "Beograd",   "zipcode": 11000 }</pre>
Format izlaznih parametara	application/json

Opis funkcije	Brisanje grada sa datim id-jem
HTTP metoda	DELETE
URL	/api/v1/gradovi
URL parametri	id[long]
HTTP body parametri	(nema)
Format HTTP body parametara	(nema)
Izlazni parametri	
Format izlaznih parametara	Application/json

Opis funkcije	Preuzimanje svih molbi iz baze
HTTP metoda	GET
URL	/api/v1/molbe
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	(nema)
Izlazni parametri	<p>JSON niz objekata tipa molba</p> <p>Primer:</p> <pre>[   {     "id" : 1,     "datum_odgovora": "2023-02-06",</pre>

```
        "datum_pitanja": "2023-02-04",
        "odgovor": "Nije moguće ovo izvršiti",
        "pitanje": "Da li možete da mi prijavite
        sve ispite?",
        "status_molbe": "RAZRESENA",
        "tip_molbe":
        "PROMENA_PODATAKA_O_STUDENTU",
        "student_id": 1
    },
    {
        "id" : 2,
        "datum_odgovora": null,
        "datum_pitanja": "2023-02-01",
        "odgovor": null
        "pitanje": "Da li možete da mi promenite
        adresu stanovanja?",
        "status_molbe": "U_OBRADI",
        "tip_molbe":
        "PROMENA_PODATAKA_O_STUDENTU",
        "student_id": 2
    },
    {
        "id" : 3,
        "datum_odgovora": null,
        "datum_pitanja": "2023-01-31",
        "odgovor": null
        "pitanje": "Da li možete da promenite
        moju sliku?",
        "status_molbe": "U_OBRADI",
        "tip_molbe":
        "PROMENA_PODATAKA_O_STUDENTU",
        "student_id": 1
    },
    {
        "id" : 3,
        "datum_odgovora": 2023-02-07,
        "datum_pitanja": "2023-01-30",
        "odgovor": "Ispravljeno."
        "pitanje": "Da li možete da ispravite moj
        broj indeksa?",
        "status_molbe": "RAZRESENA",
        "tip_molbe":
        "PROMENA_PODATAKA_O_STUDENTU",
        "student_id": 2
    }
}
```

	<pre>     } ]</pre>
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje molbi u obradi
HTTP metoda	GET
URL	/api/v1/molbe/uobradi
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	<pre> [ {     "id" : 2,     "datum_odgovora": null,     "datum_pitanja": "2023-02-01",     "odgovor": null     "pitanje": "Da li možete da mi promenite adresu stanovanja?",     "status_molbe": "U_OBRADI",     "tip_molbe": "PROMENA_PODATAKA_O_STUDENTU",     "student_id": 2 }, {     "id" : 3,     "datum_odgovora": null,     "datum_pitanja": "2023-01-31",     "odgovor": null     "pitanje": "Da li možete da promenite moju sliku?",     "status_molbe": "U_OBRADI",     "tip_molbe": "PROMENA_PODATAKA_O_STUDENTU",     "student_id": 1 }, ]</pre>
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje molbi koje su razrešene
HTTP metoda	GET
URL	/api/v1/molbe/uobradi

URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	[ { “id” : 1, “datum_odgovora”: “2023-02-06”, “datum_pitanja”: “2023-02-04”, “odgovor”: “Nije moguće ovo izvršiti”, “pitanje”: “Da li možete da mi prijavite sve ispite?”, “status_molbe”: “RAZRESENA”, “tip_molbe”: “PROMENA_PODATAKA_O_STUDENTU”, “student_id”: 1 }, { “id” : 3, “datum_odgovora”: 2023-02-07, “datum_pitanja”: “2023-01-30”, “odgovor”: “Ispravljeno.” “pitanje”: “Da li možete da ispravite moj broj indeksa?”, “status_molbe”: “RAZRESENA”, “tip_molbe”: “PROMENA_PODATAKA_O_STUDENTU”, “student_id”: 2 } ]
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje molbi koje su razrešene
HTTP metoda	GET
URL	/api/v1/molbe/{id}
URL parametri	Id:long
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	{ “id” : 1, “datum_odgovora”: “2023-02-06”, “datum_pitanja”: “2023-02-04”, “odgovor”: “Nije moguće ovo izvršiti”,

	<pre>     "pitanje": "Da li možete da mi prijavite sve ispite?",     "status_molbe": "RAZRESENA",     "tip_molbe":     "PROMENA_PODATAKA_O_STUDENTU",     "student_id": 1   } </pre>
Format izlaznih parametara	application/json

Opis funkcije	Dodavanje molbe u obradi
HTTP metoda	POST
URL	/api/v1/molbe/uobradi
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	<pre> {   "id" : 5,   "datum_odgovora": null,   "datum_pitanja": "2023-02-10",   "odgovor": null,   "pitanje": "Da li možete da mi odjavite ispit \"Matematika\"?"   "status_molbe": "U_OBRADI",   "tip_molbe":   "PROMENA_PODATAKA_O_STUDENTU",   "student_id": 1 } </pre>
Format izlaznih parametara	application/json

Opis funkcije	Razresavanje molbe
HTTP metoda	PATCH
URL	/api/v1/molbe/razresi
URL parametri	(nema)
HTTP body parametri	<pre> {   "id" : 5,   "datum_odgovora": null,   "datum_pitanja": "2023-02-10",   "odgovor": null,   "pitanje": "Da li možete da mi odjavite ispit \"Matematika\"?"   "status_molbe": "U_OBRADI", } </pre>

	<pre> "tip_molbe":  "PROMENA_PODATAKA_O_STUDENTU",  "student_id": 1  } </pre>
Format HTTP body parametara	application/json
Izlazni parametri	<pre> {   "id" : 5,   "datum_odgovora": "2023-02-20",   "datum_pitanja": "2023-02-10",   "odgovor": null,   "pitanje": "Da li možete da mi odjavite ispit \"Matematika\"?",    "status_molbe": "RAZRESENA",   "tip_molbe":  "PROMENA_PODATAKA_O_STUDENTU",  "student_id": 1 } </pre>
Format izlaznih parametara	application/json

Opis funkcije	Razresavanje molbe sa zadatim id-jem
HTTP metoda	PATCH
URL	/api/v1/molbe/{id}/razresi
URL parametri	Id: long
HTTP body parametri	Id: long, odgovor: String
Format HTTP body parametara	application/json
Izlazni parametri	<pre> {   "id" : 5,   "datum_odgovora": "2023-02-20",   "datum_pitanja": "2023-02-10",   "odgovor": "Urađeno",   "pitanje": "Da li možete da mi odjavite ispit \"Matematika\"?",    "status_molbe": "RAZRESENA",   "tip_molbe":  "PROMENA_PODATAKA_O_STUDENTU",  "student_id": 1 } </pre>
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje obaveštenja sa zadatim ID-jem
HTTP metoda	GET

URL	/api/v1/obavestenje/{id}
URL parametri	Id: long
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	{           "id" : 1,           "datum": "2023-02-20",           "sadrzaj": "Obavestavamo studente da je subota, 04-03. radni dan i da se radi po rasporedu za sredu."         }
Format izlaznih parametara	application/json

Opis funkcije	Brisanje obaveštenja sa zadatim ID-jem
HTTP metoda	DELETE
URL	/api/v1/obavestenje/{id}
URL parametri	Id: long
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	
Format izlaznih parametara	application/json

Opis funkcije	Dodavanje obaveštenja
HTTP metoda	POST
URL	/api/v1/obavestenje
URL parametri	(nema)
HTTP body parametri	{           "id" : 1,           "datum": "2023-02-20",           "sadrzaj": "Konkurs za prakse firme X."         }
Format HTTP body parametara	application/json
Izlazni parametri	
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje svih polaganja
HTTP metoda	GET
URL	/api/v1/polaganja
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	<pre>[   {     "id": 1,     "datum": "2023-01-04",     "ocena": "10",     "polozio": "true",     "predmet_id": 1,     "student_id": 2   },   {     "id": 2,     "datum": "2023-01-17",     "ocena": "5",     "polozio": "false",     "predmet_id": 2,     "student_id": 2   },   {     "id": 3,     "datum": "2023-01-04",     "ocena": "8",     "polozio": "true",     "predmet_id": 2,     "student_id": 1   } ]</pre>
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje polaganja sa zadatim ID-jem
HTTP metoda	GET
URL	/api/v1/polaganja/{id}
URL parametri	Id: long
HTTP body parametri	(nema)
Format HTTP body parametara	application/json

Izlazni parametri	<pre>{     "id": 1,     "datum": "2023-01-04",     "ocena": "10",     "polozio": "true",     "predmet_id": 1,     "student_id": 2 }</pre>
Format izlaznih parametara	application/json

Opis funkcije	Dodavanje polaganja
HTTP metoda	POST
URL	/api/v1/polaganja
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	<pre>{     "id": 4,     "datum": "2023-02-10",     "ocena": "7",     "polozio": "true",     "predmet_id": 2,     "student_id": 2 }</pre>
Format izlaznih parametara	application/json

Opis funkcije	Brisanje polaganja
HTTP metoda	DELETE
URL	/api/v1/polaganja/{id}
URL parametri	Id:long
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje uspešnih polaganja
HTTP metoda	GET
URL	/api/v1/polaganja/uspesna
URL parametri	(nema)

HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	[ { "id": 1, "datum": "2023-01-04", "ocena": "10", "polozio": "true", "predmet_id": 1, "student_id": 2 }, { "id": 3, "datum": "2023-01-04", "ocena": "8", "polozio": "true", "predmet_id": 2, "student_id": 1 } ]
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje neuspjehih polaganja
HTTP metoda	GET
URL	/api/v1/polaganja/uspesna
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	[ { "id": 2, "datum": "2023-01-17", "ocena": "5", "polozio": "false", "predmet_id": 2, "student_id": 2 }]

	<pre>     } ]</pre>
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje nepoloženih predmeta
HTTP metoda	GET
URL	/api/v1/predmeti/nepolozeni
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	<pre> [ {   "id": 1,   "trenutno_slusa": false,   "predmet_id": 1,   "student_id": 1 }, {   "id": 2,   "trenutno_slusa": true,   "predmet_id": 2,   "student_id": 1 }, {   "id": 3,   "trenutno_slusa": true,   "predmet_id": 1,   "student_id": 2 } ]</pre>
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje nepoloženih predmeta koji se trenutno slušaju
HTTP metoda	GET
URL	/api/v1/predmeti/slusa
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json

Izlazni parametri	[ { "id": 2, "trenutno_slusa": true, "predmet_id": 2, "student_id": 1 }, { "id": 3, "trenutno_slusa": true, "predmet_id": 1, "student_id": 2 } ]
Format izlaznih parametara	application/json

Opis funkcije	Biranje predmeta za slušanje
HTTP metoda	PATCH
URL	/api/v1/predmeti/slusa
URL parametri	{ "id": 3, "espb: 6, "naziv": "Elektronsko poslovanje" }
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje svih prijava
HTTP metoda	GET
URL	/api/v1/prijave/
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	[ { "student_id": 1, "predmet_id": 2 }, { "student_id": 2, "predmet_id": 1 } ]

	<pre>       "predmet_id": 2     },     {       "student_id": 1,       "predmet_id": 4     }   ] </pre>
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje prijave sa zadatim id-jem
HTTP metoda	GET
URL	/api/v1/prijave/{id}
URL parametri	Id:Long
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	<pre> [    {     "student_id": 1,     "predmet_id": 2   },   {     "student_id": 2,     "predmet_id": 2   },   {     "student_id": 1,     "predmet_id": 4   } ] </pre>
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje prijave sa zadatim id-jem
HTTP metoda	GET
URL	/api/v1/prijave/{id}
URL parametri	Id:Long
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	[

	<pre>       "student_id": 1,       "predmet_id": 2     },     {       "student_id": 2,       "predmet_id": 2     },     {       "student_id": 1,       "predmet_id": 4     }   ] </pre>
Format izlaznih parametara	application/json

Opis funkcije	Dodavanje nove prijave
HTTP metoda	POST
URL	/api/v1/prijave
URL parametri	Id:Long
HTTP body parametri	<pre> {   "student_id": 1,   "predmet_id": 3 } </pre>
Format HTTP body parametara	application/json
Izlazni parametri	
Format izlaznih parametara	application/json

Opis funkcije	Brisanje prijave sa zadatim id-jem
HTTP metoda	DELETE
URL	/api/v1/prijave/{id}
URL parametri	Id:Long
HTTP body parametri	
Format HTTP body parametara	application/json
Izlazni parametri	
Format izlaznih parametara	application/json

Opis funkcije	Preuzimanje svih studenata
---------------	----------------------------

HTTP metoda	GET
URL	/api/v1/studenti
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	[ { "id": 0, "broj_telefona": "+38167777777", "datum_rodjenja": null, "ime": "Admin", "ime_roditelja": "Admin", "indeks": "0000/2019", "jmbg": "000000000000", "licni_email": " <a href="mailto:admin@gmail.com">admin@gmail.com</a> ", "password": "\$2a\$12\$Y8KKkfMj1/rxHj5Uzq.U9.Ntog8gGp8BoUla9Fjlo7r5Vr7UegfxW", "prezime": "Admin", "role": "ADMIN", "slika": " <a href="https://balkan-hub.com/prodavnica/483-Niara_thickbox/admin.jpg">https://balkan-hub.com/prodavnica/483-Niara_thickbox/admin.jpg</a> ", "studentski_email": "aa00000000@student.fon.bg.ac.rs", "username": "aa00000000", "mesto_rodjenja_id": 1 }, { "id": 1, "broj_telefona": "+381677116969", "datum_rodjenja": null, "ime": "Veljko", "ime_roditelja": "Borislav", "indeks": "0353/2019", "jmbg": "00208000860069", "licni_email": " <a href="mailto:veljko@gmail.com">veljko@gmail.com</a> ", }

	<pre> "password": "\$2a\$12\$yLCUIQXTK4LQM1jGzZ0ldeAOs7QPW7B1z62g6o8c.07KbDq5K7Xka",   "prezime": "Blagojevic",   "role": "USER",   "slika": "https://media.licdn.com/dms/image/D4D03AQGfDciO9qccSg/profile- displayphoto- shrink_800_800/0/1665691382240?e=1680739200&amp;v=beta&amp;t=OmDSbHTkRPAUeg G52SrowRDi8GiqjSDvE7c6OCQvL6k",   "studentski_email": "vb20190353@student.fon.bg.ac.rs",   "username": "vb20190353",   "mesto_rodjenja_id": 3 }, {   "id": 2,   "broj_telefona": "+381677114512",   "datum_rodjenja": null,   "ime": "Nikola",   "ime_roditelja": "Nesa",   "indeks": "0062/2019",   "jmbg": "1201001860051",   "licni_email": "<a href="mailto:nikola@gmail.com">nikola@gmail.com</a>",   "password": "\$2a\$12\$DSxUIJvdJcrlv6iBFxmpGup6.7Mf6MVwiFtCZEM3H7iqvqPDerq6W",   "prezime": "Vujcic",   "role": "USER",   "slika": "https://media.licdn.com/dms/image/D4E03AQERPaxNYTUzXg/profile- displayphoto- shrink_800_800/0/1667431634582?e=1680739200&amp;v=beta&amp;t=KXdvm7fvNZBwtoi PuZ1xT3gZKLlrRWMSn1dudYf-Ir0",   "studentski_email": "nv20190062@student.fon.bg.ac.rs",   "username": "nv20190062",   "mesto_rodjenja_id": 6 } ] </pre>
Format izlaznih parametara	application/json

Opis funkcije	/api/v1/studenti/trenutni
HTTP metoda	GET
URL	/api/v1/studenti

URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	application/json
Izlazni parametri	<pre>{     "id": 0,     "broj_telefona": "+38167777777",     "datum_rodjenja": null,     "ime": "Admin",     "ime_roditelja": "Admin",     "indeks": "0000/2019",     "jmbg": "000000000000",     "licni_email": "<a href="mailto:admin@gmail.com">admin@gmail.com</a>",     "password": "\$2a\$12\$Y8KKkfMj1/rxHj5Uzq.U9.Ntog8gGp8BoUla9Fjlo7r5Vr7UegfXW",     "prezime": "Admin",     "role": "ADMIN",     "slika": "<a href="https://balkan-hub.com/prodavnica/483-Niara_thickbox/admin.jpg">https://balkan-hub.com/prodavnica/483-Niara_thickbox/admin.jpg</a>",     "studentski_email": "aa00000000@student.fon.bg.ac.rs",     "username": "aa00000000",     "mesto_rodjenja_id": 1 }</pre>
Format izlaznih parametara	application/json

# Opis arhitekture aplikacije

---

## Java Spring

Java Spring je open source okvir (eng. Framework) programskog jezika Java koji se neretko koristi u web development-u. Iako se najčešće koristi u svrhe razvoja web aplikacija, jezgro samog okvira može koristiti bilo koja Java aplikacija. Snaga ovog okvira leži u tome što on nije glomazan (eng. Lightweight), te obična verzija ovog okvira zauzima samo oko 2MB. Još jedna prednost je i modularnost samog okvira, te je fajlsistem (eng. Filesystem) vrlo intuitivan i jasan programeru. Ovaj okvir je kompatibilan sa MVC paternom projektovanja i kao takav idealan za pravljenje web aplikacija. Mi smo Java Spring okvir koristili za izradu backend dela ovog sajta.

## React

React je besplatna i open source frontend biblioteka programskog jezika Javascript koja služi za izradu korisničkih interfejsa zasnovanih na komponentama. Jednostavan je i lak za upotrebu jer se oslanja na osnovne tehnologije HTML-a i CSS-a (eng. Hypertext markup language i cascading style sheets). Osnovan od strane kompanije Meta (facebook), 2013. godine, React je i dalje jedan od najpopularnijih i najpoznatijih biblioteka programskog jezika Javascript. Komponente u ovoj biblioteci mogu biti klasne ili funkcionalne, a mi smo koristili oba tipa komponenata u našem radu. Ovu biblioteku smo prevashodno (čak isključivo) koristili za dizajn korisničkog interfejsa i frontend deo naše aplikacije.

## Axios

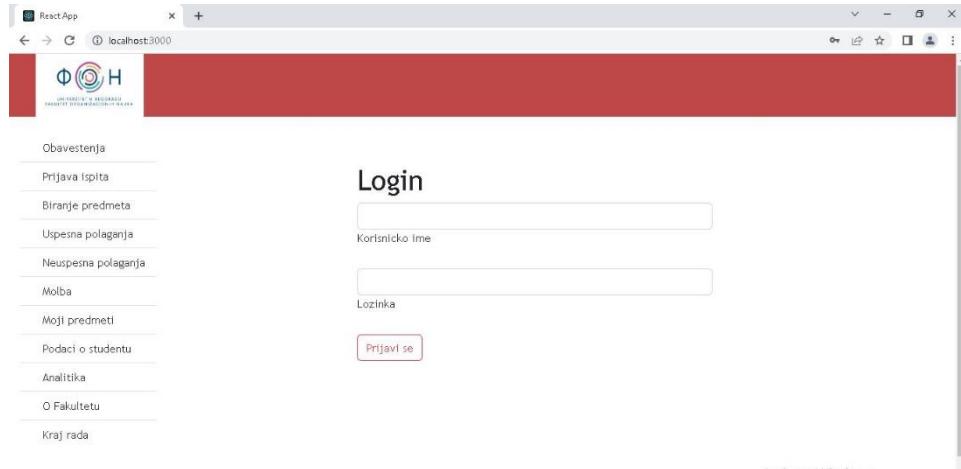
Axios je lightweight HTTP klijent zasnovan na Javascript obećanjima (eng. Promise) koji omogućava slanje HTTP zahteva sa i bez tela, kao i kreiranje zaglavlja. Ovaj moćan alat za slanje podataka smo koristili kako bismo povezali backend deo sa frontend-om, i vukli smislene podatke iz baze, kad god ih bilo neophodno prikazati korisniku. Podatke koje je korisnik uneo i izmenio smo slali nazad na backend. S obzirom da je “promise-based”, ovaj alat podržava korišćenje kako regularne sintakse (sa “then” nakon izvršavanja), tako i async-await sintakse karakteristične za Javascript programski jezik. U ovom radu smo koristili oba tipa sintakse za različite pozive.

## MySQL baza podataka

Za MySQL bazu podataka nije potreban uvod i pojašnjenje, jer ovaj tip relacione baze podataka se ustalio kao industrijski standard u sferi baza podataka. Alat koji je korišćen za rad sa tabelama u MySQL bazi podataka je besplatni program pod nazivom SQLYog.

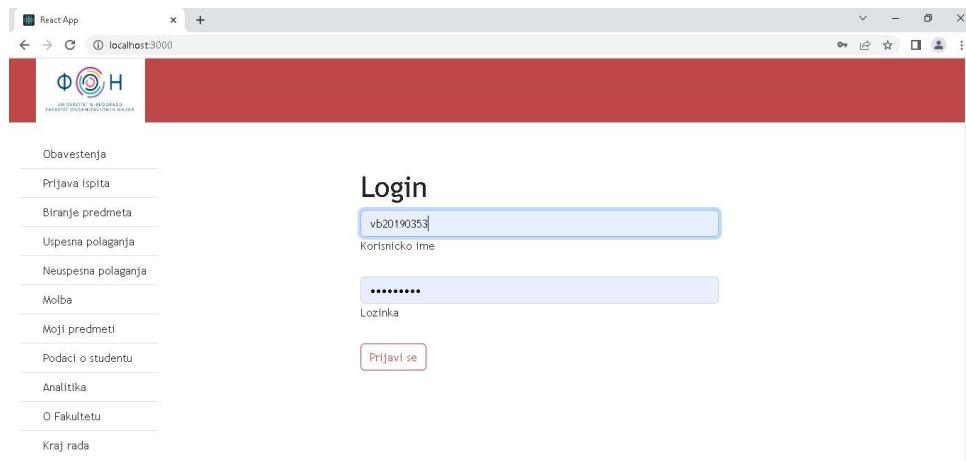
# Korisničko uputstvo

Kada korisnik pokrene sajt, njemu se otvara sledeća stranica:



Slika: Početna stranica

Sa leve strane sa nalazi navigacioni meni korisnika. Na samoj stranici je forma za prijavu gde korisnik unosi relevantne podatke i pritiska dugme "Prijavi se" da bi nastavio rad na sajtu.



Slika: Korisnik je uneo podatke. U pitanju je student

Nakon unosa podataka (vidimo da je korisnik običan student, te nema admin privilegije), on biva preusmeren na stranicu sa Obaveštenjima.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/Obavestenja". The header features the university logo and the text "UNIVERZITET U NOVOM SADU" and "FAKULTET INFORMATIČKIH NAUKA". On the left, a sidebar menu includes links for "Obavestenja", "Prijava Ispita", "Biranje predmeta", "Uspesna polaganja", "Neuspesna polaganja", "Molba", "Moji predmeti", "Podaci o studentu", "Analitika", "O Fakultetu", and "Kraj rada". The main content area is titled "Obavestenja" and contains two red-bordered boxes. The first box is dated "2022-11-11" and states: "Prva kolokvijumska nedelja u zimskom semestru će se održati u periodu od 21. do 26. novembra 2022. godine, prema rasporedu koji možete pogledati na sajtu fakulteta." The second box is dated "2022-09-09" and states: "Služba za osnovne studije neće raditi, u periodu od 26.-30.09.2022. godine, zbog poslova na upisu u više godine studija, ali će hoće za potrebe zaključivanja i na prijemu dokumentacije za prelaze sa drugih ustanova."

Slika : Stranica sa obaveštenjima

Nakon što je student pročitao obaveštenja, on može preko navigacionog menija da ode na drugu lokaciju, što mu otvara nove stranice. Prva od tih lokacija je sekcija za prijavu ispita. Onda se otvara sledeća stranica:

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/prijavaispita". The header and sidebar are identical to the previous screenshot. The main content area is titled "Prijava ispita" and displays a table of subjects with their credits and registration buttons. The table has columns for "#", "Predmet", "ESPB", and "Prijava". The subjects listed are: 1. Engleski jezik 3 (3 credits, button "Prijava ispit"), 2. Neuronske mreže (5 credits, button "Prijava ispit"), and 3. Inteligentni sistemi (5 credits, button "Prijava ispit"). Below this is a section titled "Prijavljeni ispiti" which shows a single entry: 1. Engleski jezik 3 (3 credits, button "X").

Slika : Prijava ispita

Klikom na dugme "Prijava ispit" se dati ispit briše iz sekcije "Prijava ispita" i pojavljuje u sekciji "Prijavljeni ispiti". Klikom na dugme X pored prijavljenog ispita, se isti odjavljuje.

# Prijavljeni ispiti

#	Predmet	ESPB	
1	Engleski jezik 3	3	<input checked="" type="checkbox"/>
2	Inteligentni sistemi	5	<input checked="" type="checkbox"/>
3	Neuronske mreze	5	<input checked="" type="checkbox"/>

Activate Win  
GIGA SUMMER

Slika : Prijavljeni ispiti

Sledeća adresa na koju može da se uputi student je sekција за biranje predmeta. Nakon otvaranja stranice, student vidi sledeće:

#	Predmet	ESPB	Dodaj
1	Ekonomija		<input type="checkbox"/>
2	Programiranje 2		<input type="checkbox"/>
3	Napredno programiranje		<input type="checkbox"/>
4	Engleski jezik 3		<input checked="" type="checkbox"/>
5	Neuronske mreze		<input checked="" type="checkbox"/>
6	Inteligentni sistemi		<input checked="" type="checkbox"/>

Slika : Opcija- izbor predmeta za slušanje

Čekiranjem checkbox-a pored odgovarajućeg predmeta se isti obeležava za izbor. Da bi se finalizovao izbor predmeta za slušanje, neophodno je pritisnuti dugme "Potvrđi" čime se izbor čuva.

U sekciji „Uspešna polaganja“ se studentu prikazuje spisak uspešno položenih ispita, broj ESP bodova i podaci o prosečnoj oceni.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/ispitiUspesno". The page features a red header bar with the Faculty of Organization logo. On the left, a sidebar menu includes links for "Obavestenja", "Prijava ispita", "Biranje predmeta", "Uspesna polaganja" (highlighted in blue), "Neuspesna polaganja", "Molba", "Moji predmeti", "Podaci o studentu", "Analitika", "O Fakultetu", and "Kraj rada". The main content area is titled "Ispiti" and displays a table of examination results:

#	Predmet	ESPB	Ocena	Datum
1	Programiranje 1	4	6	2022-06-15
2	Strukture podataka i algoritmi	6	8	2022-12-15
3	Elektronsko poslovanje	6	9	2023-01-04
4	Matematika 1	6	9	2022-07-13
5	Projektovanje informacionih sistema	6	10	2023-01-02

Below the table, two statistics are shown: "Prosecna ocena: 8.4" and "Ukupno osvojeno ESPB: 28".

Slika : Sekcija Uspešna polaganja

Neuspešna polaganja funkcionišu na sličan način, mada je prikaz malo drugačiji. Korišćena je paginacija, tako da se studentu na početku prikazuje svega par ispita, a on klikom na odgovarajuće dugme prelazi na adekvatnu stranicu. S obzirom da postoji samo jedna stranica u ovom slučaju, njemu će se samo pojavitи dugme sa brojem 1.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/ispitiNeuspesno". The page features a red header bar with the Faculty of Organization logo. On the left, a sidebar menu includes links for "Obavestenja", "Prijava ispita", "Biranje predmeta", "Uspesna polaganja", "Neuspesna polaganja" (highlighted in blue), "Molba", "Moji predmeti", "Podaci o studentu", "Analitika", "O Fakultetu", and "Kraj rada". The main content area is titled "Ispiti" and displays a table of examination results:

#	Predmet	ESPB	Ocena	Datum
1	Matematika 1	6	5	2021-09-01

A red rectangular button with the number "1" is positioned next to the table.

Slika : Neuspesna polaganja

Molbe se mogu podeliti u 2 kategorije. Molbe u obradi i razrešene molbe. Klikom na link “Molbe”, studentu je prikazan sledeći meni:

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/molbeUObradi". At the top, there is a navigation bar with icons for back, forward, search, and refresh, along with a user profile icon. The header features the logo of the Faculty of Organizational Sciences (FOH) and the text "UNIVERZITET U BEOGRADU FACULTET ORGANIZACIONIH NAUKA". On the left, a sidebar menu lists various options: Obavestenja, Prijava ispita, Biranje predmeta, Uspesna polaganja, Neuspesna polaganja, Molba, Moji predmeti, Podaci o studentu, Analitika, O Fakultetu, and Kraj rada. The "Molbe" section is highlighted. Below the sidebar, the main content area has a title "Molbe" and three tabs: "U obradi", "Izdate", and "Podnesi molbu". The "Podnesi molbu" tab is selected, showing a dark grey background. At the bottom of the page, there is a footer with links: "Tip molbe", "Pitanje", and "Odgovor".

Slika : Meni molbe

Klikom na dugmiće “U obradi” I “Izdate” se korisniku prikazuju molbe u obradi i izdate molbe. S obzirom da ovaj korisnik nema molbe, prelazimo na sekciju “Podnesi molbu”.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/unesiMolbu?". The layout is identical to the previous screenshot, with the FOH logo at the top and a sidebar menu on the left. The main content area has a title "Unos molbe" and a "Tip molbe:" dropdown menu set to "Promena podataka o studentu". Below the dropdown is a text input field labeled "Obrazlozenje" with a large empty text area. At the bottom right, there is a red button labeled "Potvrdi".

Slika : Stranica “Podnesi molbu”

Ovde student iz menija bira tip molbe, i ručno unosi obrazloženje. Primer unosa molbe:

The screenshot shows a web application window titled "React App" with the URL "localhost:3000/unesiMolbu?". The header features the logo of the Faculty of Organizational Sciences (FOH) and the text "UNIVERZITET U BEOGRADU FAKULTET DILJAVSKOG I NAJCKE". On the left, a vertical sidebar menu lists various options: Obavestenja, Prijava ispita, Biranje predmeta, Uspesna polaganja, Neuspesna polaganja, Molba, Moji predmeti, Podaci o studentu, Analitika, O Fakultetu, and Kraj rada. The "Molba" option is currently selected. The main content area is titled "Unos molbe". It contains a "Tip molbe:" dropdown set to "Promena podataka o studentu" and a "Obrazloženje" text input field containing the message: "Poštovani, da li biste mogli da promenite moju sliku? Zamolio bih Vas da iskoristite sliku iz mog indeksa. Hvala unapred!" A red "Potvrdi" button is at the bottom.

Slika : Unos molbe

Pritiskom na dugme "Potvrdi" se molba čuva u sistemu.

Kada student ode na stranicu "Moji predmeti", njemu je predstavljen tabelarni prikaz svih predmeta prijavljenih za slušanje. Korišćena je paginacija, kao što je objašnjeno u sekciji "Neuspesna polaganja".

The screenshot shows a web application window titled "React App" with the URL "localhost:3000/mojiPredmeti". The header features the logo of the Faculty of Organizational Sciences (FOH) and the text "UNIVERZITET U BEOGRADU FAKULTET DILJAVSKOG I NAJCKE". On the left, a vertical sidebar menu lists various options: Obavestenja, Prijava ispita, Biranje predmeta, Uspesna polaganja, Neuspesna polaganja, Molba, Moji predmeti, Podaci o studentu, Analitika, O Fakultetu, and Kraj rada. The "Moji predmeti" option is currently selected. The main content area is titled "Moji predmeti" and displays a table of registered subjects:

#	Predmet	ESPB
1	Engleski jezik 3	3
2	Neuronske mreže	5
3	Inteligentni sistemi	5

A red "1" button is located at the bottom left of the table.

Slika : Stranica "Moji predmeti"

Stranica "Podaci o studentu" na korisniku jasan način prikazuje sve relevantne podatke prijavljenog studenta.

The screenshot shows a web application window titled 'React App' at 'localhost:3000/student'. The header features the logo of the Faculty of Organizational Sciences (FOCH) and the text 'UNIVERZITET U BEOGRADU FACULTET ORGANIZACIONIH NAUKA'. On the left, a sidebar menu includes: Obavestenja, Prijava ispita, Biranje predmeta, Uspesna polaganja, Neuspesna polaganja, Molba, Moji predmeti, Podaci o studentu (which is highlighted), Analitika, O Fakultetu, and Kraj rada. The main content area is titled 'Podaci o studentu' and displays the following data in a table:

Broj indeksa	0353/2019
Ime i prezime	Veljko Blagojevic
Username	vb20190353
Ime roditelja	Borislav
JMBG	0208000860069
Mesto rođenja	Omoljica
Licni email	vejko@gmail.com
Studentski email	vb20190353@student.fon.bg.ac.rs
Broj telefona	+381677116969

To the right of the table is a photograph of a young man sitting in a lecture hall. A watermark for 'Activate Windows' is visible in the bottom right corner.

Slika : Podaci o student

Analitika je stranica na kojoj se korišćenjem kružnih dijagrama predstavlja odnos položenih i nepoloženih ispita, kao i procentualni udio svake od ocena položenih ispita prijavljenog studenta. To izgleda ovako:

The screenshot shows a web application window titled 'React App' at 'localhost:3000/analitika'. The header features the logo of the Faculty of Organizational Sciences (FOCH) and the text 'UNIVERZITET U BEOGRADU FACULTET ORGANIZACIONIH NAUKA'. On the left, a sidebar menu includes: Obavestenja, Prijava ispita, Biranje predmeta, Uspesna polaganja, Neuspesna polaganja, Molba, Moji predmeti, Podaci o studentu (which is highlighted), Analitika, O Fakultetu, and Kraj rada. The main content area displays the following information:

**Podaci o polaganju ispita studenta:**  
Veljko Blagojevic

**Sa brojem indeksa:**  
0353/2019

Below this, there are two pie charts. The first pie chart is mostly red and has two segments labeled '1' and '5'. The second pie chart is divided into five segments of different colors: purple, yellow, orange, green, and blue. Below the charts is a legend:

sestice	sedmice	osmice	devetke	desete
---------	---------	--------	---------	--------

A watermark for 'Activate Windows' is visible in the bottom right corner.

Slika : Analitika studenta

Stranica "O fakultetu" sadrži relevantne podatke o fakultetu Organizacionih nauka, kao i gugl mapu sa lokacijom Fakulteta organizacionih nauka.

The screenshot shows a web browser window titled 'React App' with the URL 'localhost:3000/oNama'. The page has a red header bar with the Faculty of Organizational Sciences logo. A sidebar on the left contains links: Obavestenja, Prijava ispita, Biranje predmeta, Uspesna polaganja, Neuspesna polaganja, Molba, Moji predmeti, Podaci o studentu, Analitika, O Fakultetu, and Kraj rada. The main content area displays the title 'Fakultet organizacionih nauka'. Below it is a text block stating that the faculty was founded in 1969 and offers three programs: management, operational, and a quality management program. It also mentions that the faculty has branches in Belgrade and Novi Sad. To the right of the text is a map showing the location of Jove Ilića 154, Beograd 11000, Serbia. A red pin marks the exact location. The map also shows surrounding landmarks like Beo Shopping Center, Kralj Petar I, and Konjarnik. At the bottom right of the page, there is a watermark: 'Activate Windows Go to Settings to activate Windows.'

Slika : Stranica "O fakultetu"

Ukoliko student klikne na link "Kraj Rada" biće preusmeren na početnu stranicu, ali i izlogovan. U nastavku sledi uputsvo za rad administratora. Prvo je neophodno da se administrator prijavi svojim kredencijalima:

The screenshot shows a web browser window titled 'React App' with the URL 'localhost:3000/krajRada'. The page has a red header bar with the Faculty of Organizational Sciences logo. A sidebar on the left contains links: Obavestenja, Prijava Ispita, Biranje predmeta, Uspesna polaganja, Neuspesna polaganja, Molba, Moji predmeti, Podaci o studentu, Analitika, O Fakultetu, and Kraj rada. The main content area displays a 'Login' form with two input fields: 'Korisnicko ime' containing 'aa00000000' and 'Lozinka' containing '\*\*\*\*\*'. A 'Prijava se' button is located below the inputs. At the bottom right of the page, there is a watermark: 'Activate Windows Go to Settings to activate Windows.'

Slika : Kraj rada. Prijava administrator

Nakon prijave, i administratoru se, kao i studentu, prikazuje strana sa Obaveštenjima.

Slika : Početna strana, obaveštenja

Podaci o studentu su jedinstveni jer prikazuju administratora, a ne studenta u pravom smislu te reči.

Podaci o studentu	
Broj indeksa	0000/2019
Ime i prezime	Admin Admin
Username	aa00000000
Ime roditelja	Admin
JMBG	00000000000000
Mesto rodjenja	Beograd
Licni email	admin@gmail.com
Studentski email	aa00000000@student.fon.bg.ac.rs

Slika : Podaci o administrator "studentu"

Ukoliko administrator želi da izmeni podatke nekog od studenata, on to vrši u sekciji „Izmeni podatke o studentu“.

The screenshot shows a web application interface titled "Molbe". On the left, there is a sidebar with navigation links: "Obavestenja", "Upisi ocenu", "Molba", "Podaci o studentu", "Izmeni podatke o studentu" (which is highlighted in grey), "O Fakultetu", "ADMIN", and "Kraj rada". The main content area has a title "Molbe" and a table with three rows. The table columns are "#", "Student", "Indeks", and "Izmeni". The data rows are: 0 Admin 0000/2019 (with a pink "Izmeni" button), 1 Veljko 0353/2019 (with a pink "Izmeni" button), and 2 Nikola 0062/2019 (with a pink "Izmeni" button).

Slika : Izmena podataka o studentu

Kada se otvori stranica za izmenu podataka o studentima, administrator može izabrati studenta za izmenu iz tabele. Pritisak na dugme „Izmeni“ ga onda preusmerava na odgovarajuću stranicu.

The screenshot shows a web application interface titled "Podaci o studentu". On the left, there is a sidebar with the same navigation links as the previous screenshot. The main content area displays a form with fields for "Broj indeksa" (0353/2019), "Ime i prezime" (Veljko Blagojević), "Username" (vb20190353), "Ime roditelja" (Borislav), "JMBG" (0208000860069), "Mesto rodjenja" (Omoljica), "Licni email" (vejko@gmail.com), and "Studentski email" (vb20190353@student.fon.). To the right of the form is a photo of a young man sitting at a desk in a lecture hall. Below the photo is a "Potvrdi" (Confirm) button. A watermark for "Activate Windows Go to Settings to activate Windows." is visible in the bottom right corner.

Slika : Promena podataka studenta

Klikom na dugme „Potvrdi“ se promene o studentu čuvaju u sistemu.

Jedan od najbitnijih delova sajta, iz perspektive administratora, je i stranica "ADMIN". Dostupna samo administratorima, ova stranica daje detaljne informacije o IP adresi mreže na kojoj je administrator prijavljen. Ovo se odvija u 2 faze.

The screenshot shows a web browser window titled 'React App' with the URL 'localhost:3000/admin'. The page has a red header bar with the logo 'ΦΟΗ' and the text 'UNIVERZITET U NOVOM SADU' and 'FAKULTET DIOGAZACIJE I NAJEGA'. On the left, there is a sidebar with a navigation menu:

- Obavestenja
- Upisi ocenu
- Molba
- Podaci o studentu
- Izmeni podatke o studentu
- O Fakultetu
- ADMIN** (highlighted)
- Kraj rada

The main content area is titled 'Podaci za administratora' and contains the following text: 'U ovoj sekciji možete dobiti dodatne informacije o javnoj IP adresi servisa i druge servisne informacije.' Below this text are two red buttons:

- Prikaži trenutnu javnu IP adresu
- Prikaži dodatne informacije o adresi

Slika : Podaci za administrator

Prvo dugme na ovoj stranici će samo prikazati javnu IP adresu računara, dok drugo dugme daje dalje relevantne informacije

## Podaci za administratora

U ovoj sekciji možete dobiti dodatne informacije o javnoj IP adresi servisa i druge servisne informacije.

Sakrij trenutnu javnu IP adresu

**109.245.192.77**

Prikaži dodatne informacije o adresi

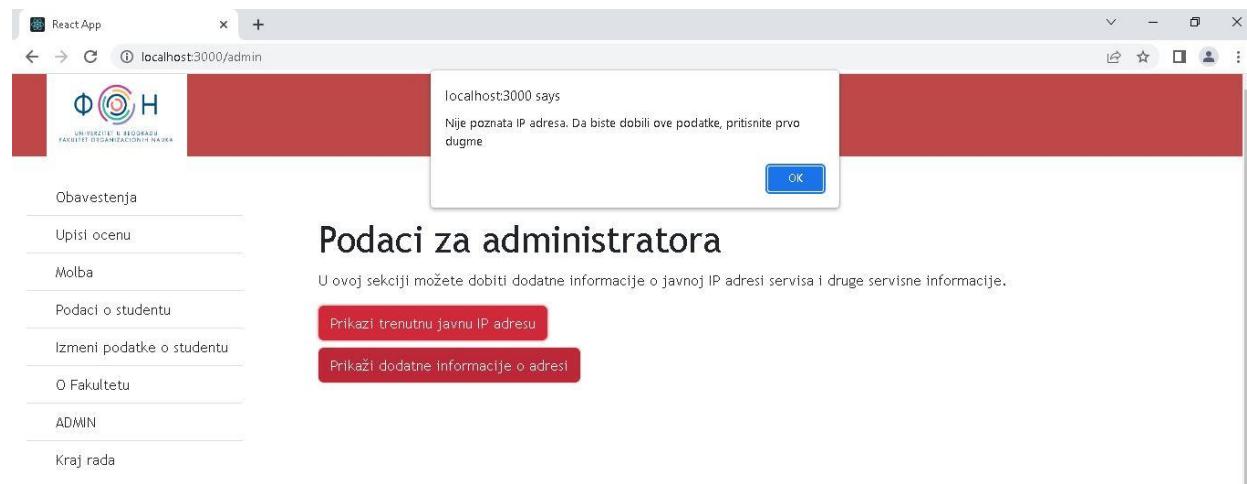
Slika : Prikazana IP adresa administratorskog sistema

Sakrij podatke

#	Ključ	Vrednost
1	asn	AS15958
2	grad	Novi Sad
3	kontinent	EU
4	država	RS
5	region ID	88361
6	pozivni kod	+381
7	prestonica	Belgrade
8	državni kod	RS
9	državni kod iso3	SRB
10	naziv države	Serbia
11	valuta	RSD
12	region	Vojvodina
13	vremenska zona	Europe/Belgrade

#### Slika : Detaljnije informacije o IP adresi.

Potrebno je da administrator prvo pritisne prvo dugme, a zatim drugo. Ukoliko on to ne učini, i pokuša da prikaže detaljne informacije nepoznate IP adrese, njemu se prikazuje sledeće obaveštenje:



Slika : Greška pri radu administratora

# Prikaz reprezentativnih delova koda

Ovaj softverski sistem je napravljen po uzoru na troslojnu arhitekturu, i kao takav sadrži 3 dela. To su frontend, backend i baza podataka. U ovom delu će detaljnije biti opisana prva 2 dela softverskog sistema.

## Backend

Backend deo je implementiran u programskom jeziku Java, u okviru Spring, korišćenjem Intellij razvojnog okruženja.



The screenshot shows the code editor in IntelliJ IDEA with the following Java code:

```
2 import ...
3
4     ^ gildrausNV +1
5
6     @SpringBootApplication
7
8     public class StudentskaSluzbaApplication {
9
10    no usages ^ VeljkoBlagojevic
11
12    public static void main(String[] args) { SpringApplication.run(StudentskaSluzbaApplication.class, args); }
13
14    no usages ^ gildrausNV
15
16    @Bean
17    public WebMvcConfigurer configure(){
18        ^ gildrausNV
19        return new WebMvcConfigurer() {
20            no usages ^ gildrausNV
21            @Override
22            public void addCorsMappings(CorsRegistry registry){
23                registry.addMapping( pathPattern: "/*").allowedOrigins("http://localhost:3000");
24            }
25        };
26    }
27
28
29 }
```

Slika : Main klasa backend projekta

Izvršenje kao i uvek počinje od main klase. Ovde vidimo da se prilikom mapiranja dozvolja dostavljanje podataka sa porta 3000 frontend aplikacije.

U nastavku slede screenshot-ovi domenskih klasa projekta.

```
11  @AllArgsConstructor  
12  @Builder  
13  @Getter  
14  @Setter  
15  @EqualsAndHashCode  
16  @ToString  
17  
18  @Entity(name = "grad")  
19  public class Grad {  
20      no usages  
21      @Id  
22      @GeneratedValue(strategy = GenerationType.AUTO)  
23      private Long id;  
24      no usages  
25      @NotBlank(message = "Mora postojati naziv grada")  
26      private String naziv;  
27      no usages  
28      @NaturalId  
29      @Min(value = 10_000, message = "Zipcode mora imati tacno 5 cifara")  
30      @Max(value = 99_999, message = "Zipcode mora imati tacno 5 cifara")  
31      private Integer zipcode;  
32  }
```

Slika : Klasa Grad

```
20 public class Molba {  
21     no usages  
22     @Id  
23     @GeneratedValue(strategy = GenerationType.AUTO)  
24     private Long id;  
25     no usages  
26     @NotBlank(message = "Pitanje ne moze biti prazno")  
27     private String pitanje;  
28     no usages  
29     @PastOrPresent(message = "Datum pitanja ne moze biti u buducnosti")  
30     private LocalDate datumPitanja;  
31     no usages  
32     private String odgovor;  
33     no usages  
34     @PastOrPresent(message = "Datum odgovora ne moze biti u buducnosti")  
35     private LocalDate datumOdgovora;  
36     no usages  
37     @Enumerated(EnumType.STRING)  
38     private TipMolbe tipMolbe;  
39 }
```

Slika : Klasa Molba

```
10     @Setter  
11     @EqualsAndHashCode  
12     @ToString  
13  
14     @Entity(name = "nepolozeni_predmet")  
15     public class NepolozeniPredmet {  
16         no usages  
17         @Id  
18         @GeneratedValue(strategy = GenerationType.AUTO)  
19         private Long id;  
20  
21         no usages  
22         @ManyToOne  
23         private Student student;  
24         no usages  
25         @ManyToOne  
26         private Predmet predmet;  
27     }
```

Slika : Klasa NepolozeniPredmet

```
10  @NoArgsConstructor  
11  @AllArgsConstructor  
12  @Builder  
13  @Getter  
14  @Setter  
15  @EqualsAndHashCode  
16  @ToString  
17  
18  @Entity(name = "obavestenje")  
19  public class Obavestenje {  
20  
    no usages  
21  @Id  
22  @GeneratedValue(strategy = GenerationType.AUTO)  
23  private Long id;  
24  
    no usages  
25  private LocalDate datum;  
26  
    no usages  
27  @NotBlank(message = "Sadrzaj obavestenja ne sme biti prazno")  
28  @Size(max = 1000, message = "Sadrzaj obavestenja ne sme prelaziti 1000 karaktera")  
29  @Column(length = 1000)  
30  private String sadrzaj;  
31  
32 }
```

Slika : Klasa Obavestenje

```
19  @Entity(name = "polaganje")
20  public class Polaganje {
21      no usages
22      @Id
23          @GeneratedValue(strategy = GenerationType.AUTO)
24          private Long id;
25
26          no usages
27          @Past(message = "Polaganje mora da se desilo u prolosti")
28          private LocalDate datum;
29
30          no usages
31          @Min(value = 5, message = "Ocena ne moze biti manja od 5")
32          @Max(value = 10, message = "Ocena ne moze biti veca od 10")
33          private Integer ocena;
34
35          no usages
36          @ManyToOne
37          private Student student;
38      }
```

Slika : Klasa Polaganje

```
11  @NoArgsConstructor
12  @AllArgsConstructor
13  @Builder
14  @Getter
15  @Setter
16  @EqualsAndHashCode
17  @ToString
18
19  @Entity(name = "predmet")
20  public class Predmet {
21      no usages
22      @Id
23          @GeneratedValue(strategy = GenerationType.AUTO)
24          private Long id;
25
26          no usages
27          private String naziv;
28
29          @Min(value = 1, message = "Predmet mora vredeti barem 1 ESPB")
30          @Max(value = 30, message = "Predmet ne moze vredeti vise od 30 ESPB")
31          private Integer ESPB;
32  }
```

Slika : Klasa Predmet

```
1 package rs.fon.studentska_sluzba.domain;
2
3 import org.springframework.security.core.GrantedAuthority;
4
5  * VeljkoBlagojevic
6 public enum Role implements GrantedAuthority {
7
8     1 usage
9     USER,
10    1 usage
11    ADMIN;
12
13     * VeljkoBlagojevic
14    @Override
15    public String getAuthority() { return name(); }
16}
```

Slika : Enum Role

```
1 package rs.fon.studentska_sluzba.domain;
2
3  * VeljkoBlagojevic
4 public enum StatusMolbe {
5
6     3 usages
7     U_OBRADI,
8     4 usages
9     RAZRESENA
10}
```

Slika : Enum StatusMolbe

```
27  @Entity(name = "student")
28  public class Student implements UserDetails {
29
30      no usages
31      @Id
32      @GeneratedValue(strategy = GenerationType.AUTO)
33      private Long id;
34      no usages
35      @NaturalId
36      @Pattern(regexp = "\d{4}/(19|20)\d{2}")
37      private String indeks;
38      no usages
39      @NotBlank(message = "Ime ne moze biti prazno")
40      private String ime;
41      no usages
42      @NotBlank(message = "Prezime ne moze biti prazno")
43      private String prezime;
44
45      no usages
46      @Size(min = 13, max = 13, message = "JMBG mora imati tacno 13 cifara")
47      private String jmbg;
48      no usages
49      private LocalDate datumRodjenja;
50      no usages
51      @Pattern(regexp = "+381\\d{9}", message = "Mobilni telefon treba da bude u formatu '+3816...'")
52      private String brojTelefona;
```

Slika : Klasa Student

```
1 package rs.fon.studentska_sluzba.domain;
2
3     ✎ VeljkoBlagojevic
4
5     public enum TipMolbe {
6
7         no usages
8         PROMENA_PODATAKA_O_STUDENTU,
9         no usages
10        PONISTAVANJE_OCENE
11    }
```

Slika : Enum TipMolbe

Lista konkretnih repozitorijuma korišćenih u projektu se nalazi u nastavku:

```
1 package rs.fon.studentska_sluzba.repository;
2
3 import ...
4
5
6     ▲ VeljkoBlagojevic
7 @Repository
8 public interface GradRepository extends JpaRepository<Grad, Long> {
9 }
```

Slika : GradRepository

```
1 package rs.fon.studentska_sluzba.repository;
2
3 import ...
4
5
6     ▲ VeljkoBlagojevic
7
8 public interface MolbaRepository extends JpaRepository<Molba, Long> {
9     3 usages ▲ VeljkoBlagojevic
10    List<Molba> findByStudent(Student student);
11
12    1 usage ▲ VeljkoBlagojevic
13    Optional<Molba> findByStudentAndId(Student student, Long id);
14 }
```

Slika : MolbaRepository

```
1 package rs.fon.studentska_sluzba.repository;
2
3 import ...
4
5
6     ▲ VeljkoBlagojevic
7 @Repository
8 public interface NepolozeniPredmetRepository extends JpaRepository<NepolozeniPredmet, Long> {
9
10    1 usage ▲ VeljkoBlagojevic
11    List<NepolozeniPredmet> findByStudentAndTrenutnoSlusa(Student student, Boolean trenutnoSlusa);
12    1 usage ▲ VeljkoBlagojevic
13    Page<NepolozeniPredmet> findByStudentAndTrenutnoSlusa(Student student, Boolean trenutnoSlusa, Pageable pageable);
14    2 usages ▲ VeljkoBlagojevic
15    List<NepolozeniPredmet> findByTrenutnoSlusa(Boolean trenutnoSlusa);
16    1 usage ▲ VeljkoBlagojevic
17    Page<NepolozeniPredmet> findByTrenutnoSlusa(Boolean trenutnoSlusa, Pageable pageable);
18    1 usage ▲ VeljkoBlagojevic
19    List<NepolozeniPredmet> findByStudent(Student student);
20    1 usage ▲ VeljkoBlagojevic
21    Page<NepolozeniPredmet> findByStudent(Student student);
22 }
```

Slika : NepoloženiPredmetRepository

```
1 package rs.fon.studentska_sluzba.repository;
2
3 import ...
6
7
8 @Repository
9 public interface ObavestenjeRepository extends JpaRepository<Obavestenje, Long> {
10 }
```

Slika : ObavestenjeRepository

```
1 package rs.fon.studentska_sluzba.repository;
2
3 import ...
11
12 public interface PolaganjeRepository extends JpaRepository<Polaganje, Long> {
13
14     List<Polaganje> findByStudent(Student student);
15
16     Optional<Polaganje> findByStudentAndId(Student student, Long id);
17
18     List<Polaganje> findByStudentAndAndPolozio(Student student, Boolean polozi);
19
20     Page<Polaganje> findByStudentAndAndPolozio(Student student, Boolean polozi, Pageable pageable);
21
22     List<Polaganje> findByPolozio(Boolean polozi);
23     Page<Polaganje> findByPolozio(Boolean polozi, Pageable pageable);
24 }
```

Slika : Polaganje Repository

```
1 package rs.fon.studentska_sluzba.repository;
2
3 import ...
6
7
8 @Repository
9 public interface PredmetRepository extends JpaRepository<Predmet, Long> {
10 }
```

Slika : PredmetRepository

```
1 package rs.fon.studentska_sluzba.repository;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16 }
```

Slika : StudentRepository

Lista konkretnih kontrolera korišćenih u projektu se nalazi u nastavku:

```
16 public class GradController {
17
18     5 usages
19     private final GradService gradService;
20
21     5 usages
22     private final GradMapper gradMapper;
23
24     □
25     □
26     □
27     □
28
29     □
30     □
31     □
32     □
33
34     no usages
35     @GetMapping("/{id}")
36     □
37 }
```

Slika : Klasa GradController

```
14  @RequestMapping("/api/v1/molbe")
15  public class MolbaController {
16
17      9 usages
18      private MolbaService molbaService;
19
20      10 usages
21      private MolbaMapper molbaMapper;
22
23      ▲ VeljkoBlagojevic
24      public MolbaController(MolbaService molbaService, MolbaMapper molbaMapper) {
25          this.molbaService = molbaService;
26          this.molbaMapper = molbaMapper;
27      }
28
29      ▲ VeljkoBlagojevic
30      @GetMapping
31      public ResponseEntity<List<MolbaDTO>> getAll() {
32          return ResponseEntity.ok(molbaMapper.entitiesToDTOs(molbaService.findAll()));
33      }
34
```

Slika : MolbaController

```
13  @RequestMapping("api/v1/obavestenja")
14  public class ObavestenjeController {
15
16      5 usages
17      private final ObavestenjeService obavestenjeService;
18
19      5 usages
20      private final ObavestenjeMapper obavestenjeMapper;
21
22      ▲ VeljkoBlagojevic
23      public ObavestenjeController(ObavestenjeService obavestenjeService, ObavestenjeMapper obavestenjeMapper) {
24          this.obavestenjeService = obavestenjeService;
25          this.obavestenjeMapper = obavestenjeMapper;
26      }
27
28      no usages ▲ VeljkoBlagojevic
29      @GetMapping
30      public ResponseEntity<List<ObavestenjeDTO>> getSvaObavestenja() {
31          return ResponseEntity.ok(obavestenjeMapper.entitiesToDTOs(obavestenjeService.getSvaObavestenja()));
32      }
33
34      no usages ▲ VeljkoBlagojevic
35      @GetMapping("/{id}")
36      public ResponseEntity<ObavestenjeDTO> getObavestenje(@PathVariable Long id) {
37          return ResponseEntity.ok(obavestenjeMapper.entityToDTO(obavestenjeService.getObavestenjeSaId(id)));
38      }
```

Slika : ObavestenjeController

```
15  @RequestMapping("api/v1/polaganja")
16  public class PolaganjeController {
17
18      8 usages
19      private final PolaganjeService polaganjeService;
20
21      8 usages
22      private final PolaganjeMapper polaganjeMapper;
23
24      ✎ VeljkoBlagojevic
25      public PolaganjeController(PolaganjeService polaganjeService, PolaganjeMapper polaganjeMapper) {
26          this.polaganjeService = polaganjeService;
27          this.polaganjeMapper = polaganjeMapper;
28      }
29
30      no usages ✎ VeljkoBlagojevic
31      @GetMapping
32      public ResponseEntity<List<PolaganjeDTO>> getSvaPolaganja() {
33          return ResponseEntity.ok(polaganjeMapper.entitiesToDTOs(polaganjeService.getSvaPolaganja()));
34      }
35      no usages ✎ VeljkoBlagojevic
36      @GetMapping("/{id}")
37      public ResponseEntity<PolaganjeDTO> getPolaganje(@PathVariable Long id) {
38          return ResponseEntity.ok(polaganjeMapper.entityToDTO(polaganjeService.getPolaganjeById(id)));
39      }
40
```

Slika : PolaganjeController

```
17  @RequestMapping("api/v1/predmeti")
18  public class PredmetController {
19
20      5 usages
21      private final PredmetService predmetService;
22
23      4 usages
24      private final NepolozeniPredmetMapper nepolozeniPredmetMapper;
25
26      3 usages
27      private final PredmetMapper predmetMapper;
28
29      ▲ VeljkoBlagojevic
30      @Autowired
31      public PredmetController(PredmetService predmetService, NepolozeniPredmet
32          this.predmetService = predmetService;
33          this.nepolozeniPredmetMapper = nepolozeniPredmetMapper;
34          this.predmetMapper = predmetMapper;
35      }
36
37      no usages ▲ VeljkoBlagojevic
38      @GetMapping("/nepolozeni")
39      public ResponseEntity<List<NepolozeniPredmetDTO>> getSveNepolozenePredmet
40          return ResponseEntity.ok(nepolozeniPredmetMapper.entitiesToDTOs(predm
41      }
42
43      no usages ▲ VeljkoBlagojevic
44      @GetMapping("/slusa")
```

Slika : PredmetController

```
15  @RequestMapping("api/v1/prijava")
16  public class PrijavaController {
17      5 usages
18      private final PrijavaService prijavaService;
19
20      4 usages
21      private final PredmetMapper predmetMapper;
22
23      ▲ VeljkoBlagojevic
24      public PrijavaController(PrijavaService prijavaService, PredmetMapper predmetMapper) {
25          this.prijavaService = prijavaService;
26          this.predmetMapper = predmetMapper;
27      }
28
29      no usages ▲ VeljkoBlagojevic
30      @GetMapping
31      public ResponseEntity<Set<PredmetDTO>> getPrijava() {
32          return ResponseEntity.ok(new HashSet<>(predmetMapper.entitiesToDTOs(prijavaService.getPrijava().stream().toList())));
33      }
34
35      no usages ▲ VeljkoBlagojevic
36      @GetMapping("/{id}")
37      public ResponseEntity<PredmetDTO> getPrijavaSaId(@PathVariable Long id) {
38          return ResponseEntity.ok(predmetMapper.entityToDTO(prijavaService.getPrijavaSaId(id)));
39      }
40
```

Slika : PrijavaController

```
17  @RequestMapping("/api/v1/studenti")
18  public class StudentController {
19
20      3 usages
21      private final StudentService studentService;
22
23      3 usages
24      private final StudentMapper studentMapper;
25
26      ▲ VeljkoBlagojevic
27      @Autowired
28      public StudentController(StudentService studentService, StudentMapper studentMapper) {
29          this.studentService = studentService;
30          this.studentMapper = studentMapper;
31      }
32
33      no usages ▲ VeljkoBlagojevic
34      @GetMapping
35      public ResponseEntity<List<StudentDTO>> getSviStudenti() {
36          return ResponseEntity.ok(studentMapper.entitiesToDTOs(studentService.findAll()));
37      }
38
39      no usages ▲ VeljkoBlagojevic
40      @GetMapping("/trenutni")
41      public ResponseEntity<StudentDTO> getTrenutniStudent() {
42          return ResponseEntity.ok(studentMapper.entityToDTO(studentService.getTrenutniStudent()));
43      }
44
```

Slika : StudentController

## Frontend

Frontend deo je implementiran u programskom jeziku Javascript, preciznije u biblioteci React ovog programskog jezika, korišćenjem VSCode razvojnog okruženja.

```
31  function App() {
32    if(!user.Admin){
33      return [
34        <div class='Page'>
35          <BrowserRouter className='App'>
36            <div class='NavBar'><Navbar/></div>
37            <div class='Main'>
38              <div class='SideBar'><Sidebar/></div>
39              <div class='CenterComponent'>
40                <Routes>
41                  <Route path='/' element={<Login/>} />
42                  <Route path='/krajRada' element={<Login/>} />
43                  <Route path='/ispitiUspesno' element={<IspitiUspesno/>} />
44                  <Route path='/ispitiNeuspesno' element={<IspitiNeuspesno/>} />
45                  <Route path='/prijavaIspita' element={<PrijavaIspita/>} />
46                  <Route path='/molbeUObradi' element={<MolbeUObradi/>} />
47                  <Route path='/molbeIzdate' element={<MolbeIzdate/>} />
48                  <Route path='/student' element={<Student/>} />
49                  <Route path='/unesiMolbu' element={<UnosMolbe/>} />
50                  <Route path='/biranjePredmeta' element={<BiranjePredmeta/>} />
51                  <Route path='/mojiPredmeti' element={<MojiPredmeti/>} />
52                  <Route path='/obavestenja' element={<Obavestenja/>} />
53                  <Route path='/oNama' element = {<ONama/>}/>
54                  <Route path='/analitika' element={<Analitika/>}></Route>
55                </Routes>
56              </div>
57            </div>
58          </BrowserRouter>
59        </div>
60      );
61    }
62
63  else{
64    return [
65      <div class='Page'>
66        <BrowserRouter className='App'>
67          <div class='NavBar'><Navbar/></div>
68          <div class='Main'>
69            <div class='SideBar'><SidebarAdmin/></div>
70            <div class='CenterComponent'>
71              <Routes>
72                <Route path='/' element={<Login/>} />
73                <Route path='/krajRada' element={<Login/>} />
74                <Route path='/upisiOcenu' element={<UpisiOcenu/>} />
75                <Route path='/molbeUObradi' element={<MolbeAdmin/>} />
76                <Route path='/molbeIzdate' element={<MolbeIzdate/>} />
77                <Route path='/student' element={<Student/>} />
78                <Route path='/obavestenja' element={<ObavestenjaAdmin/>} />
79                <Route path='/oNama' element = {<ONama/>}/>
80                <Route path='/unosObavestenja' element={<UnosObavestenja/>} />
81                <Route path='/izmeniPodatke' element={<IzmeniPodatke/>}></Route>
82                <Route path='/podaciOStudentu' element={<PodaciOStudentu/>}></Route>
83              </Routes>
84            </div>
85          </div>
86        </BrowserRouter>
87      </div>
88    );
89  }
90}
```

```

79         <Route path="/onama" element={<Onama/>} />
80         <Route path="/unosObavestenja" element={<UnosObavestenja/>} />
81         <Route path="/izmeniPodatke" element={<IzmeniPodatke/>} />
82         <Route path="/podaciOStudentu" element={<PodaciOStudentu/>} />
83         <Route path="/admin" element={<Admin/>} />
84     </Routes>
85   </div>
86 </div>
87 </BrowserRouter>
88 </div>
89 </>;
90 }
91 }
92
93 export default App;
94

```

Slika : App.js fajl

```

4 const Admin = () => {
5   const [IPAdresa, setIPAdresa] = useState("");
6   const [dugme1Text, setDugme1Text] = useState("Prikazi trenutnu javnu IP adresu");
7   const [dugme2Text, setDugme2Text] = useState("Prikaži dodatne informacije o adresi");
8
9   const [IPInfo, setIPInfo] = useState({
10     asn: '',
11     city: '',
12     continent_code: '',
13     country: '',
14     country_area: '',
15     country_calling_code: '',
16     country_capital: '',
17     country_code: '',
18     country_code_iso3: '',
19     country_name: '',
20     currency: '',
21     region: '',
22     timezone: '',
23   })
24
25   const dugme1Action = () => {
26     if(dugme1Text === "Prikazi trenutnu javnu IP adresu"){
27       axios.get("https://api.ipify.org/?format=json").then((response) =>{
28         setIPAdresa(response.data.ip)
29         setDugme1Text("Sakrij trenutnu javnu IP adresu")

```

Slika : Admin React komponenta (deo)

```

7  class Analitika extends React.Component {
8      state = {
9          data: []
10     }
11
12     componentDidMount() {
13         axios({
14             method: 'get',
15             url: '/api/v1/studenti/trenutni',
16             baseURL: 'http://localhost:8080',
17             data: {},
18             headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
19         }).then((response) => {
20             console.log(response);
21             this.setState({data: response.data})
22             //alert(JSON.stringify(response.data.mestoRodjenja.naziv));
23
24         }, (error) => {
25             console.log(error);
26         });
27     }
28 }

```

Slika : Komponenta Analitika

```

29     render() {
30         return (
31             <>
32                 <h3>Podaci o polaganju ispita studenta: </h3>
33                 <h4>{this.state.data?.ime} {this.state.data?.prezime}</h4>
34                 <br><br>
35                 <h3>Sa brojem indeksa: </h3>
36                 <h4>{this.state.data?.indeks}</h4>
37                 <div className='container'>
38                     <div className = "child">
39                         <Grafikon ></Grafikon>
40                     </div>
41                     <div className = "child">
42                         <GrafikonOcene></GrafikonOcene>
43                     </div>
44                 </div>
45             </>
46         );
47     }
48 }
49
50
51 export default Analitika;

```

Slika : Komponenta Analitika (nastavak)

```

5  class BiranjePredmeta extends React.Component {
6
7    state = {
8      data: []
9    }
10
11   builder(){
12     this.forceUpdateHandler = this.forceUpdateHandler.bind(this);
13   };
14
15   forceUpdateHandler(){
16     this.forceUpdate();
17   };
18
19   componentDidMount() {
20
21     axios({
22       method: 'get',
23       url: '/api/v1/predmeti/nepolozeni',
24       baseURL: 'http://localhost:8080',
25       data: {},
26       headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
27     ).then((response) => {
28       console.log(response);
29       this.setState({data: response.data})

```

Slika : Komponenta BiranjePredmeta

```

29       this.setState({data: response.data});
30       console.log(this.state.data);
31     }, (error) => {
32       console.log(error);
33     });
34
35
36   }
37
38   izaberipredmete(){
39     for(let i=0;i<this.state.data.length;i++){
40       axios({
41         method: 'patch',
42         url: '/api/v1/predmeti/slusa',
43         baseURL: 'http://localhost:8080',
44         data: {
45           id: this.state.data[i].id,
46           predmet: this.state.data[i].predmet,
47           trenutnoSlusa: this.state.data[i].trenutnoSlusa
48         },
49         headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
50       ).then((response) => {
51         console.log(response);
52       }, (error) => {
53         console.log(error);

```

Slika : Komponenta Biranje Predmeta – Nastavak

```

4
5  export default class Grafikon extends PureComponent {
6    static demoUrl = 'https://codesandbox.io/s/two-simple-pie-chart-otx9h';
7
8    state = {
9      data: {}
10 }
11
12  componentDidMount() {
13    axios({
14      method: 'get',
15      url: '/api/v1/polaganja',
16      baseURL: 'http://localhost:8080',
17      data: {},
18      headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
19    }).then((response) => {
20      console.log(response);
21      let brojPolozenih = 0;
22      let brojPalih = 0;
23      response.data.forEach(zapis=>{
24        if(zapis.polozio === true){
25          brojPolozenih++;
26        }else brojPalih++;
27      });
28      this.setState({
29        data: {
30          ...
31        }
32      })
33    })
34  }

```

Slika : Komponenta Grafikon (deo)

```

5  class GrafikonOcene extends React.Component {
6    state = {
7      data: {}
8    }
9    pieData = [0,0,0,0,0];
10
11   componentDidMount() {
12     axios({
13       method: 'get',
14       url: '/api/v1/polaganja/uspesna',
15       baseURL: 'http://localhost:8080',
16       data: {},
17       headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
18     }).then((response) => {
19       console.log(response);
20       let brojSestica = 0;
21       let brojSedmica = 0;
22       let brojOsmica = 0;
23       let brojDevetki = 0;
24       let brojDesetki = 0;
25       response.data.forEach(zapis => {
26         switch(zapis.ocena){
27           case 6: brojSestica++; break;
28           case 7: brojSedmica++; break;
29           case 8: brojOsmica++; break;
30           case 9: brojDevetki++; break;
31           case 10: brojDesetki++; break;
32         }
33       })
34       this.setState({
35         data: {
36           ...
37         }
38       })
39     })
40   }

```

Slika : Komponenta GrafikonOcene (deo)

```
5  class Ispiti extends React.Component {
6      state = {
7          data: [],
8          dataNeuspescno: [],
9          dataUspescno: [],
10         tip: false
11     }
12
13     componentDidMount() {
14         axios({
15             method: 'get',
16             url: '/api/v1/polaganja/uspesna',
17             baseURL: 'http://localhost:8080',
18             data: {},
19             headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
20         }).then((response) => {
21             console.log(response);
22             this.setState({ dataUspescno: response.data })
23             console.log(this.state.dataUspescno);
24         }, (error) => {
25             console.log(error);
26         });
27         axios({
28             method: 'get',
29             url: '/api/v1/polaganja/neuspescno',
30             baseURL: 'http://localhost:8080',
31             data: {}
```

Slika : React komponenta Ispiti

```
77  render(){
78    return (
79      <div className='Ispiti'>
80        <h1>Ispiti</h1>
81        <div class="status">
82          <form>
83            <button class="statusMolbe" onClick={this.plsRadi}>Uspesna polaganja</button>
84            <button class="statusMolbe" onClick={this.plsRadi}>Neuspesna polaganja</button>
85          </form>
86        </div>
87        <table class="table">
88          <thead class="thead-dark">
89            <tr>
90              <th scope="col">#</th>
91              <th scope="col">Predmet</th>
92              <th scope="col">ESPB</th>
93              <th scope="col">Ocena</th>
94            </tr>
95          </thead>
96          <tbody>
97            {(this.state.tip ? this.state.dataUspesno : this.state.dataNeuspesno).map(el =>
98              <tr>
99                <td>{el.ocena}</td>
100               <td>{el.predmet}</td>
101               <td>{el.espb}</td>
102               <td>{el.ocena}</td>
103             </tr>
104           )
105         </tbody>
106       </table>
107     </div>
108   )
```

Slika : React komponenta Ispiti, render metoda

```
6  class IspitiNeuspesno extends React.Component {
7    constructor(){
8      super();
9      this.state = {
10        data: [],
11        trenutnaStrana:1,
12      }
13      this.setTrenutnaStranica = this.setTrenutnaStranica.bind(this)
14    }
15    state = {
16    }
17
18    indeksPoslednjegPredmeta = 4;
19    indeksPrvogPredmeta = 1;
20    trenutniPredmeti = [];
21
22    elementiPoStrani= 4;
23
24    setTrenutnaStranica(brojStranice) {
25      this.state.trenutnaStrana = brojStranice;
26      this.forceUpdate();
27    }
28  }
```

```

29  </>    componentDidMount() {
30    <div>
31      axios({
32        method: 'get',
33        url: '/api/v1/polaganja/neuspesna',
34        baseURL: 'http://localhost:8080',
35        data: {},
36        headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
37      ).then((response) => {
38        console.log(response);
39        this.setState({data: response.data});
40        console.log(this.state.data);
41      }, (error) => {
42        console.log(error);
43      });
44    </div>
45  </>    render(){
46      this.indeksPoslednjegPredmeta = this.state.trenutnaStrana * this.elementiPoStrani;
47      this.indeksPrvogPredmeta = this.indeksPoslednjegPredmeta - this.elementiPoStrani;
48      this.trenutniPredmeti = this.state.data.slice(this.indeksPrvogPredmeta, this.indeksPoslednjegPredmeta);
49      return (
50        <div className='Ispiti'>
51          <h1>Ispiti</h1>
52          <table className="table">
53            <thead className="thead-dark">
54              <tr>
55                <th scope="col">#</th>
56                <th scope="col">Predmet</th>
57                <th scope="col">ESPB</th>
58                <th scope="col">Ocena</th>
59                <th scope="col">Datum</th>
60              </tr>
61            </thead>
62            <tbody>
63              {this.trenutniPredmeti.map((el, index) => <tr><td>{index+1}</td><td>{el.predmet}</td>
64                </tbody>
65            </table>
66            <Paginacija ukupniPredmeti = {this.state.data.length}
67            elementiPoStrani = {this.elementiPoStrani}
68            setTrenutnaStranica = {this.setTrenutnaStranica}/>
69        </div>
70      );
71    </>  
```

Slika : React komponenta IspitiNeuspesno

```

5   class IspitiUspesno extends React.Component {
6     state = {
7       data: [],
8       prosek: 0,
9       ukupnoESPB: 0
10    }
11
12  componentDidMount() {
13    axios({
14      method: 'get',
15      url: '/api/v1/polaganja/uspesna',
16      baseURL: 'http://localhost:8080',
17      data: {},
18      headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
19    }).then((response) => {
20      console.log(response);
21      this.setState({data: response.data});
22      console.log(this.state.data);
23      this.setState({prosek: (this.state.data.map(el => el.ocena).reduce((partialSum, a) => partialSum + a) / this.state.data.length)});
24      this.setState({ukupnoESPB: (this.state.data.map(el => el.predmet.ESPB).reduce((partialSum, a) => partialSum + a))});
25    }, (error) => {
26      console.log(error);
27    });
28  }
29
30  render(){
31    this.indeksPoslednjegPredmeta = this.state.trenutnaStrana * this.elementiPoStrani;
32    this.indeksPrvogPredmeta = this.indeksPoslednjegPredmeta - this.elementiPoStrani;
33    this.trenutniPredmeti = this.state.data.slice(this.indeksPrvogPredmeta, this.indeksPoslednjegPredmeta);
34    return (
35      <div className='Ispiti'>
36        <h1>Ispiti</h1>
37        <table className="table">
38          <thead className="thead-dark">
39            <tr>
40              <th scope="col">#</th>
41              <th scope="col">Predmet</th>
42              <th scope="col">ESPB</th>
43              <th scope="col">Ocena</th>
44              <th scope="col">Datum</th>
45            </tr>
46          </thead>
47          <tbody>
48            {this.trenutniPredmeti.map((el, index) => <tr><td>{index+1}</td><td>{el.predmet}</td>
49            <td>{el.ESPB}</td>
50            <td>{el.ocena}</td>
51            <td>{el.datum}</td>
52          </tr>)}
53        </tbody>
54      </table>
55      <Paginacija ukupniPredmeti = {this.state.data.length}>
56        elementiPoStrani = {this.elementiPoStrani}
57        setTrenutnaStranica = {this.setTrenutnaStranica}/>
58      </Paginacija>
59    </div>
60  );
61}

```

Slika : komponenta IspitiUspesno

```
6  function Login() {
7    const [username, setUsername] = useState('');
8    const [password, setPassword] = useState('');
9    const [poruka, setPoruka] = useState('');
10
11   const handleChangeUsername = (event) => {
12     setUsername(event.target.value);
13   };
14
15   const handleChangePassword = (event) => {
16     setPassword(event.target.value);
17   };
18 }
```

Slika : Funkcija Login

```
33  componentDidMount() {
34    axios({
35      method: 'get',
36      url: '/api/v1/predmeti/slusa',
37      baseURL: 'http://localhost:8080',
38      data: {},
39      headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
40    }).then((response) => {
41      console.log(response);
42      this.setState({data: response.data})
43      console.log("DATA SET " + this.state.data);
44      //this.setState({trenutnaStrana : 1})
45      //console.log("TRENUTNA STRANA SET" + this.state.trenutnaStrana)
46
47    }, (error) => {
48      console.log(error);
49    });
50  }
51 }
```

Slika : Komponenta MojiPredmeti (samo axios poziv)

```

5  function Molba() {
6    const [data, setData] = useState([]);
7    useEffect(() => {
8      axios({
9        method: 'get',
10       url: '/api/v1/molbe/uobradi',
11       baseURL: 'http://localhost:8080',
12       data: {},
13       headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
14     ).then((response) => {
15       console.log(response);
16       setData(response.data);
17       console.log(data);
18     }, (error) => {
19       console.log(error);
20     });
21   }, []);
22
23   return (
24     <div className='Molba'>
25       <h1>Molbe</h1>
26       <div class="status">
27         <form>
28           <button class="statusMolbe" formaction="/unesiMolbu">U obradi</button>
29           <button class="statusMolbe" formaction="/unesiMolbu">Izdate</button>
30           <button class="statusMolbe" formaction="/unesiMolbu">Podnesi molbu</button>
31         </form>
32       </div>
33       <table class="table">
34         <thead class="thead-dark">
35           <tr>
36             <th scope="col">Tip molbe</th>
37             <th scope="col">Pitanje</th>
38             <th scope="col">Odgovor</th>
39           </tr>
40         </thead>
41         <tbody>
42           </tbody>
43         </table>
44       </div>
45     );
46   }
47 }
48
49 export default Molba

```

Slika : Komponenta Molba

```

5   function MolbeIzdate() {
6     const [data, setData] = useState([]);
7     useEffect(() => {
8       axios({
9         method: 'get',
10        url: '/api/v1/molbe/razresene',
11        baseURL: 'http://localhost:8080',
12        data: {},
13        headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
14      ).then((response) => {
15        console.log(response);
16        setData(response.data);
17        console.log(data);
18      }, (error) => {
19        console.log(error);
20      });
21    }, []);
22
23    return (
24      <div className='Molba'>
25        <h1>Molbe</h1>
26        <div class="status">
27          <form>
28            <button class="statusMolbe" formaction="/molbelObradi">U obradi</button>
29            <button class="statusMolbe" formaction="/molbeIzdate">Izdate</button>
30            <button class="statusMolbe" formaction="/unesiMolbu">Pôdneši molbu</button>
31          </form>
32        </div>
33        <table class="table">
34          <thead class="thead-dark">
35            <tr>
36              <th scope="col">Tip molbe</th>
37              <th scope="col">Pitanje</th>
38              <th scope="col">Odgovor</th>
39            </tr>
40          </thead>
41          <tbody>
42
43            </tbody>
44          </table>
45        </div>
46      );
47    }
48  }
49  export default MolbeIzdate

```

Slika : Komponenta Molbelzdate

```

5  function MolbeUObradi() {
6    const [data, setData] = useState([]);
7    useEffect(() => {
8      axios({
9        method: 'get',
10       url: '/api/v1/molbe/uobradi',
11       baseURL: 'http://localhost:8080',
12       data: {},
13       headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
14     }).then((response) => {
15       console.log(response);
16       setData(response.data);
17       console.log(data);
18     }, (error) => {
19       console.log(error);
20     });
21   }, []);
22
23   return (
24     <div className='Molba'>
25       <h1>Molbe</h1>
26       <div class="status">
27         <form>
28           <button class="statusMolbe" formaction="/molbeUObradi">U obradi</button>
29           <button class="statusMolbe" formaction="/molbeIzdate">Izdate</button>
30           <button class="statusMolbe" formaction="/unesiMolbu">Podnesi molbu</button>
31         </form>
32       </div>
33       <table class="table">
34         <thead class="thead-dark">
35           <tr>
36             <th scope="col">Tip molbe</th>
37             <th scope="col">Pitanje</th>
38             <th scope="col">Odgovor</th>
39           </tr>
40         </thead>
41         <tbody>
42           </tbody>
43         </table>
44       </div>
45     );
46   }
47 }
48
49 export default MolbeUObradi

```

Slika: Komponenta MolbeUObradi

```
1 import React from 'react';
2 import { Link } from 'react-router-dom';
3
4 function TopNavbar() {
5     return (
6         <div className="TopNavbar2">
7             <nav class="TopNavbar2">
8
9
10            <div class="Fon"></img></div>
11
12
13        </nav>
14    </div>
15 )
16 }
17
18 export default TopNavbar
```

Slika : Komponenta TopNavbar

```
5 class Obavestenja extends React.Component {
6
7     state = {
8         data: []
9     }
10
11     componentDidMount() {
12         axios({
13             method: 'get',
14             url: '/api/v1/obavestenja',
15             baseURL: 'http://localhost:8080',
16             data: {},
17             headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
18         ).then((response) => {
19             console.log(response);
20             this.setState({data: response.data})
21             console.log(this.state.data);
22         }, (error) => {
23             console.log(error);
24         });
25     }
26
27     render(){
28         return (
29             <div className='Obavestenja'>
30                 <h1>Obavestenja</h1>
```

Slika : komponenta Obavestenja

```
27 |     render(){
28 |       return (
29 |         <div className='Obavestenja'>
30 |           <h1>Obavestenja</h1>
31 |
32 |           {this.state?.data.map(el =>
33 |             <div class="post">
34 |               <h3>{el.datum}</h3>
35 |               <p>{el.sadrzaj}</p>
36 |             </div>
37 |
38 |           )}
39 |         </div>
40 |       );
41 |     }
42 |
43 |     export default Obavestenja
```

Slika: render metoda komponente Obavestenja

```
src > components > ObradiMolbu.jsx > ...
1  import React from 'react';
2  import { Link } from 'react-router-dom';
3  import axios from 'axios';
4
5  function UnosMolbe() {
6    function odgovor(){
7      alert('Ubaci odobravanje molbe');
8    }
9    return (
10      <div className='UnosMolbe'>
11        <h1>Unos molbe</h1>
12        <h5>Tip molbe:</h5>
13        <div class="form-group">
14          <label for="exampleFormControlTextareal">Odgovor</label>
15          <textarea class="form-control" id="text" rows="3"></textarea>
16          <br><br>
17          <button type="button" class="btn btn-danger" onClick={odgovor}>Potvrdi</button>
18        </div>
19      </div>
20    );
21  }
22
23  export default UnosMolbe
```

Slika: KomponentaUnosMolbe

```

3  function ONama() {
4    return (
5      <>
6        <h2 className='title'>Fakultet organizacionih nauka</h2>
7        <br></br>
8        <p>
9          Fakultet organizacionih nauka je osnovan 1969. godine. Sa radom je počeo 3 godine nakon svog
10         Na samom početku svog postojanja, na fakultetu su postojala tri smera: menadžmentsko opšti,
11         smer za upravljanje kvalitetom.
12        </p>
13        <p>
14          Nešto kasnije je otvoren i smer za kibernetiku, koji je kasnije prerastao u smer
15          ISiT, (Informacioni sistemi i tehnologije). Danas fakultet ima brojne smerove i podsmere
16          master i doktorskim studijama, i tako studentima pruža široko znanje u oblastima menadžmenta
17          upravljanja projektima, informacionih sistema, programiranja, data science-a, matematike, i
18          srodnih akademskih polja. Fakultet se nalazi na adresi Jove Ilića 154, na Voždovcu, i deo je
19        </p>
20        <br></br>
21        <iframe width="400" height="350" id="gmap_canvas" src="https://maps.google.com/maps?q=Jove%20Ilić"
22        >/>
23    )
24  }
25
26 export default ONama

```

Slika: komponenta ONama

```

4  const Paginacija = ({ukupniPredmeti, elementiPoStrani, setTrenutnaStranica}) => {
5    console.log(ukupniPredmeti)
6    console.log(elementiPoStrani)
7    console.log(Math.ceil(ukupniPredmeti/elementiPoStrani))
8
9    let stranice = [];
10
11   for(let i = 1;i<=Math.ceil(ukupniPredmeti/elementiPoStrani);i++){
12     stranice.push(i);
13   }
14
15   return (
16     <div>{stranice.map((stranica, index)=>{
17       return <button className="paginationBtn" margin = "10px" key = {index}
18       onClick = {()=>{
19         setTrenutnaStranica(stranica);
20       }}>{stranica}</button>
21     })
22     </div>
23   )
24 }
25
26 export default Paginacija

```

Slika : Paginacija

```

4  const PrijavaIspita = () => {
5    const[data, setData] = useState([]);
6    const[dataPrijavljeni, setDataPrijavljeni] =
7      useState([]);
8
9    useEffect(() => {
10      axios({
11        method: 'get',
12        url: '/api/v1/predmeti/slusa',
13        baseURL: 'http://localhost:8080',
14        data: {},
15        headers: { 'Authorization': 'Bearer ' +localStorage.getItem('token') }
16      }).then((response) => {
17        console.log(response);
18        setData(response.data)
19        console.log(response.data);
20      }, (error) => {
21        console.log(error);
22      });
23      axios({
24        method: 'get',
25        url: '/api/v1/prijave',
26        baseURL: 'http://localhost:8080',
27        data: {},
28        headers: { 'Authorization': 'Bearer ' +localStorage.getItem('token') }
29      ).then((response) => {
30        setDataPrijavljeni(response.data)
31        console.log(response.data);
32      }, (error) => {
33        console.log(error);
34      });
35    });
36  })
37
38  function prijaviIspit(index){
39    axios({
40      method: 'post',
41      url: '/api/v1/prijave',
42      baseURL: 'http://localhost:8080',
43      data: {
44        id: data[index].id,
45        naziv: data[index].naziv,
46        espb: data[index].espb
47
48      },
49      headers: { 'Authorization': 'Bearer ' +localStorage.getItem('token') }
50    }).then((response) => {
51      }, (error) => {
52        console.log(error);
53      });
54    }
55  }

```

Slika : Komponenta Prijavalspita (1)

```
52         console.log(error);
53     });
54
55 }
56 function odjaviliSpit(id){
57     axios({
58         method: 'delete',
59         url: '/api/v1/prijave/'+id,
60         baseURL: 'http://localhost:8080',
61         data: {
62             },
63             headers: { 'Authorization': 'Bearer '+localStorage.getItem('token')}
64         }).then((response) => {
65             }, (error) => {
66                 console.log(error);
67             });
68 }
69
70 return (
71     <div className='PrijavaIspita'>
72         <h1>Prijava ispita</h1>
73         <table class="table">
74             <thead class="thead-dark">
75                 <tr>
76                     <th scope="col">#</th>
77                 <tr>
78                     <th scope="col">#</th>
79                     <th scope="col">Predmet</th>
80                     <th scope="col">ESPB</th>
81                     <th scope="col">Prijava</th>
82                 </tr>
83             </thead>
84             <tbody>
85                 {data?.map((el, index) => <tr><td>{index+1}</td><td>{el.naziv}</td><td>{el.ESPB}</td>
86                 </tbody>
87             </table>
88             <h1>Prijavljeni ispiti</h1>
89             <table class="table">
90                 <thead class="thead-dark">
91                     <tr>
92                         <th scope="col">#</th>
93                         <th scope="col">Predmet</th>
94                         <th scope="col">ESPB</th>
95                     </tr>
96                 </thead>
97                 <tbody>
98                     {dataPrijavljeni?.map((el, index) => <tr><td>{index+1}</td><td>{el.naziv}</td><td>{el.ESPB}</td>
99                 </tbody>
100             </table>
101         </div>
102     )
```

Slika : Komponenta Prijavalspita (2)

```

4  function Sidebar() {
5    return (
6      <div className='Sidebar2'>
7        <div className='Sidebar'>
8          <nav id="sidebarMenu" className="collapse d-lg-block sidebar collapse bg-white">
9            <div className="position-sticky">
10              <div className="list-group list-group-flush mx-3 mt-4">
11                <Link
12                  to="/obavestenja"
13                  className="list-group-item list-group-item-action py-2 ripple"
14                  aria-current="true"
15                >
16                  <i className="fas fa-tachometer-alt fa-fw me-3" style="color: #007bff;"></i><span>Obavestenja</span>
17                </Link>
18                <Link
19                  to="/prijavaIspita"
20                  className="list-group-item list-group-item-action py-2 ripple"
21                  aria-current="true"
22                >
23                  <i className="fas fa-tachometer-alt fa-fw me-3" style="color: #007bff;"></i><span>Prijava ispita</span>
24                </Link>
25                <Link to="/biranjePredmeta" className="list-group-item list-group-item-action py-2 ripple">
26                  <i className="fas fa-chart-area fa-fw me-3" style="color: #007bff;"></i><span>Biranje predmeta</span>
27                </Link>
28                <Link to="/ispitiUspesno" className="list-group-item list-group-item-action py-2 ripple">
29                  <i className="fas fa-lock fa-fw me-3" style="color: #007bff;"></i><span>Uspesna polaganja</span>
30                </Link>
31                <Link to="/ispitiNeuspesno" className="list-group-item list-group-item-action py-2 ripple">
32                  <i className="fas fa-lock fa-fw me-3" style="color: #dc3545;"></i><span>Neuspesna polaganja</span>
33                </Link>
34                <Link to="/molbeUObradi" className="list-group-item list-group-item-action py-2 ripple">
35                  <i className="fas fa-chart-line fa-fw me-3" style="color: #007bff;"></i><span>Molba</span>
36                </Link>
37                <Link to="/mojiPredmeti" className="list-group-item list-group-item-action py-2 ripple">
38                  <i className="fas fa-chart-pie fa-fw me-3" style="color: #007bff;"></i><span>Moji predmeti</span>
39                </Link>
40                <Link to="/student" className="list-group-item list-group-item-action py-2 ripple">
41                  <i className="fas fa-chart-pie fa-fw me-3" style="color: #007bff;"></i><span>Podaci o studentu</span>
42                </Link>
43                <Link to="/analitika" className="list-group-item list-group-item-action py-2 ripple">
44                  <i className="fas fa-chart-pie fa-fw me-3" style="color: #007bff;"></i><span>Analitika</span>
45                </Link>
46                <Link to="/oNama" className="list-group-item list-group-item-action py-2 ripple">
47                  <i className="fas fa-chart-pie fa-fw me-3" style="color: #007bff;"></i><span>O Fakultetu</span>
48                </Link>
49                <Link to="/krajRada" className="list-group-item list-group-item-action py-2 ripple">
50                  <i className="fas fa-chart-pie fa-fw me-3" style="color: #007bff;"></i><span>Kraj rada</span>
51                </Link>
52              </div>
53            </div>
54          </nav>

```

Slika : Komponenta Sidebar (1)

```

30                <Link to="/ispitiNeuspesno" className="list-group-item list-group-item-action py-2 ripple">
31                  <i className="fas fa-lock fa-fw me-3" style="color: #dc3545;"></i><span>Neuspesna polaganja</span>
32                </Link>
33
34                <Link to="/molbeUObradi" className="list-group-item list-group-item-action py-2 ripple">
35                  <i className="fas fa-chart-line fa-fw me-3" style="color: #007bff;"></i><span>Molba</span>
36                </Link>
37
38                <Link to="/mojiPredmeti" className="list-group-item list-group-item-action py-2 ripple">
39                  <i className="fas fa-chart-pie fa-fw me-3" style="color: #007bff;"></i><span>Moji predmeti</span>
40                </Link>
41
42                <Link to="/student" className="list-group-item list-group-item-action py-2 ripple">
43                  <i className="fas fa-chart-pie fa-fw me-3" style="color: #007bff;"></i><span>Podaci o studentu</span>
44                </Link>
45
46                <Link to="/analitika" className="list-group-item list-group-item-action py-2 ripple">
47                  <i className="fas fa-chart-pie fa-fw me-3" style="color: #007bff;"></i><span>Analitika</span>
48                </Link>
49
50                <Link to="/oNama" className="list-group-item list-group-item-action py-2 ripple">
51                  <i className="fas fa-chart-pie fa-fw me-3" style="color: #007bff;"></i><span>O Fakultetu</span>
52                </Link>
53
54            </div>
55          </div>
56        </nav>

```

Slika : Komponenta Sidebar (2)

```
6  class Student extends React.Component {
7      state = {
8          data: []
9      }
10 }
11
12 componentDidMount() {
13     axios({
14         method: 'get',
15         url: '/api/v1/studenti/trenutni',
16         baseURL: 'http://localhost:8080',
17         data: {},
18         headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
19     }).then((response) => {
20         console.log(response);
21         this.setState({data: response.data});
22         //alert(JSON.stringify(response.data.mestoRodjenja.naziv));
23
24     }, (error) => {
25         console.log(error);
26     });
27 }
28 }
```

Slika : Komponenta Student (1)

```
29 render() {
30     return (
31         <div className='Student'>
32             <table class="table">
33                 <tbody>
34                     <tr>
35                         <th><h1>Podaci o studentu</h1></th>
36                     </tr>
37                     <tr>
38                         <th>Broj indeksa</th>
39                         <td>{this.state.data?.indeks}</td>
40                     </tr>
41                     <tr>
42                         <th>Ime i prezime</th>
43                         <td>{this.state.data?.ime + " " + this.state.data?.prezime}</td>
44                     </tr>
45                     <tr>
46                         <th>Username</th>
47                         <td>{this.state.data?.username}</td>
48                     </tr>
49                     <tr>
50                         <th>Ime roditelja</th>
```

Slika : Komponenta Student (2)

```

50          <th>Ime roditelja</th>
51          <td>{this.state.data?.imeRoditelja}</td>
52      </tr>
53      <tr>
54          <th>JMBG</th>
55          <td>{this.state.data?.jmbg}</td>
56      </tr>
57      <tr>
58          <th>Mesto rodjenja</th>
59          <td>{this.state.data?.mestoRodjenja}</td>
60      </tr>
61      <tr>
62          <th>Licni email</th>
63          <td>{this.state.data?.licniEmail}</td>
64      </tr>
65      <tr>
66          <th>Studentski email</th>
67          <td>{this.state.data?.studentskiEmail}</td>
68      </tr>
69      <tr>
70          <th>Broj telefona</th>
71          <td>{this.state.data?.brojTelefona}</td>
72      </tr>
73
74  </tbody>

```

Slika : Komponenta Student (3)

```

5  function UnosMolbe() {
6      function podnesiMolbu(){
7          axios({
8              method: 'post',
9              url: '/api/v1/molbe/uobradi',
10             baseURL: 'http://localhost:8080',
11             data: {
12                 //sadrzaj: document.getElementById("text").value
13                 pitanje: document.getElementById("text").value
14             },
15             headers: { 'Authorization': 'Bearer '+localStorage.getItem('token')}
16         ).then((response) => {
17             }, (error) => {
18                 console.log(error);
19             });
20     }
21     return (
22         <div className='UnosMolbe'>
23             <h1>Unos molbe</h1>
24             <h5>Tip molbe:</h5>
25             <select name="language" id="language">
26                 <option value="javascript">Promena podataka o studentu</option>
27             </select>
28             <div className="form-group">
29                 <label htmlFor="exampleFormControlTextarea1">Obrazlozenje</label>

```

Slika : komponenta UnosMolbe (1)

```

24     <h5>Tip molbe:</h5>
25     <select name="language" id="language">
26       <option value="javascript">Promena podataka o studentu</option>
27     </select>
28     <div class="form-group">
29       <label for="exampleFormControlTextarea1">Obrazlozenje</label>
30       <textarea class="form-control" id="text" rows="3"></textarea>
31       <br><br>
32       <button type="button" class="btn btn-danger" onClick={podnesiMolbu}>Potvrdi</button>
33     </div>
34   </div>
35 );
36 }
37
38 export default UnosMolbe

```

Slika : komponenta UnosMolbe (2)

```

5  function IzmeniPodatke() {
6    const [data, setData] = useState([]);
7    useEffect(() => {
8      axios({
9        method: 'get',
10       url: '/api/v1/studenti',
11       baseURL: 'http://localhost:8080',
12       data: {},
13       headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
14     }).then((response) => {
15       console.log(response);
16       setData(response.data);
17       console.log(data);
18     }, (error) => {
19       console.log(error);
20     });
21   }, []);
22
23   function Izmeni(id){
24     localStorage.setItem('student', id);
25     window.location.href = "/podaciOStudentu";
26   };
27
28   return (
29     <div className='Molba'>

```

Slika : Komponenta IzmeniPodatke (1)

```

    return (
      <div className='Molba'>
        <h1>Molbe</h1>
        <table className="table">
          <thead className="thead-dark">
            <tr>
              <th scope="col">#</th>
              <th scope="col">Student</th>
              <th scope="col">Indeks</th>
              <th scope="col">Izmeni</th>
            </tr>
          </thead>
          <tbody>
            {data.map((el) => <tr><td>{el.id}</td><td>{el.ime}</td><td>{el.indeks}</td><td><button>Izmeni</button></td></tr>)}
          </tbody>
        </table>
      </div>
    );
}

export default IzmeniPodatke

```

Slika: komponenta IzmeniPodatke

```

5   function MolbeUObradi() {
6     const [data, setData] = useState([]);
7     useEffect(() => {
8       axios({
9         method: 'get',
10        url: '/api/v1/molbe',
11        baseURL: 'http://localhost:8080',
12        data: {},
13        headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
14      ).then((response) => {
15        console.log(response);
16        setData(response.data);
17        console.log(data);
18      }, (error) => {
19        console.log(error);
20      });
21    }, []);
22 //OMOGUCI DA SE UPISE ODGOVOR NA MOLBU I POTVRDI UNOS
23   return (
24     <div className='Molba'>
25       <h1>Molbe</h1>
26       <div className="status">
27       </div>
28       <table className="table">
29         <thead className="thead-dark">

```

Slika : koponentaMolbeUObradi (Admin, 1)

```

30         <tr>
31             <th scope="col">Tip molbe</th>
32             <th scope="col">Pitanje</th>
33             <th scope="col">Odgovor</th>
34             <th scope="col">Obradi molbu</th>
35         </tr>
36     </thead>
37     <tbody>
38
39     </tbody>
40   </table>
41 </div>
42 );
43 }
44
45 export default MolbeUObradi

```

Slika : koponentaMolbeUObradi (Admin, 2)

```

5  class ObavestenjaAdmin extends React.Component {
6
7    state = {
8      data: []
9    }
10
11   componentDidMount() {
12     axios({
13       method: 'get',
14       url: '/api/v1/obavestenja',
15       baseURL: 'http://localhost:8080',
16       data: {},
17       headers: { 'Authorization': 'Bearer ' +localStorage.getItem('token') }
18     ).then((response) => {
19       console.log(response);
20       this.setState({data: response.data})
21       console.log(this.state.data);
22     }, (error) => {
23       console.log(error);
24     });
25   }
26
27   render(){
28     return (
29       <div className='Obavestenja'>

```

Slika: komponenta Obavestenja (Admin, 1)

```

27  render(){
28    return (
29      <div className='Obavestenja'>
30        <h1>Obavestenja</h1>
31
32        {this.state?.data.map(el =>
33          <div class="post">
34            <h3>{el.datum}</h3>
35            <p>{el.sadrzaj}</p>
36          </div>
37        )}
38        <br><br>
39        <button type="button" class="btn btn-outline-danger" onClick={() => window.location.href =
40          el.link}
41        >Pogledaj</button>
42      );
43    }
44  export default ObavestenjaAdmin

```

Slika : komponenta Obavestenja (Admin, 2, render metoda)

```

6  class Student extends React.Component {
7    state = {
8      data: [],
9      student: []
10    }
11
12    componentDidMount() {
13      axios({
14        method: 'get',
15        url: '/api/v1/studenti',
16        baseURL: 'http://localhost:8080',
17        data: {},
18        headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token')}
19      ).then((response) => {
20        console.log(response);
21        this.setState({data: response.data});
22        //alert(localStorage.getItem('student'));
23        //alert(response.data.filter(el => el.id === localStorage.getItem('student'))[0]);
24        this.setState({student: response.data.find(el => '' + el.id === localStorage.getItem('student'))});
25        //alert(response.data[0].id);
26        //alert(JSON.stringify(response.data.mestoRodjenja.naziv));
27
28      }, (error) => {
29        console.log(error);
30      });
31      //this.state.data.find(el => el.id === localStorage.getItem('student'));

```

Slika : komponenta Student (1, Admin)

```

34    render() {
35      return (
36        <div className='Student'>
37          <table class="table">
38            <tbody>
39              <tr>
40                <th><h1>Podaci o studentu</h1></th>
41              </tr>
42              <tr>
43                <th>Broj indeksa</th>
44                <td><input type="text" value={this.state.student?.indeks}/></td>
45              </tr>
46              <tr>
47                <th>Ime i prezime</th>
48                <td><input type="text" value={this.state.student?.ime + " " + this.state.stu
49              </tr>
50              <tr>
51                <th>Username</th>
52                <td><input type="text" value={this.state.student?.username}/></td>
53              </tr>
54              <tr>
55                <th>Ime roditelja</th>
56                <td><input type="text" value={this.state.student?.imeRoditelja}></td>

```

Slika : komponenta Student (2, Admin, deo render metode)

```

8   class Prijavljspita extends React.Component {
9
10    state = {
11      data: [],
12      dataPrijavljeni: []
13    }
14
15    builder(){
16      this.forceUpdateHandler = this.forceUpdateHandler.bind(this);
17    };
18
19    forceUpdateHandler(){
20      this.forceUpdate();
21    };
22
23    componentDidMount() {
24      axios({
25        method: 'get',
26        url: '/api/v1/predmeti/slusa',
27        baseURL: 'http://localhost:8080',
28        data: {},
29        headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
30      }).then((response) => {
31        console.log(response);
32        this.setState({data: response.data})

```

Slika : komponenta Prijavljspita (1)

```

33     console.log(this.state.data);
34 }, (error) => {
35     console.log(error);
36 });
37 axios({
38     method: 'get',
39     url: '/api/v1/prijave',
40     baseURL: 'http://localhost:8080',
41     data: {},
42     headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
43 }).then((response) => {
44     console.log(response);
45     this.setState({dataPrijavljeni: response.data});
46     console.log(this.state.data);
47 }, (error) => {
48     console.log(error);
49 });
50 }
51 prijaviIspit(index){
52     axios({
53         method: 'post',
54         url: '/api/v1/prijave',
55         baseURL: 'http://localhost:8080',
56         data: {
57             id: this.state.data[index].id,
58             naziv: this.state.data[index].naziv,

```

Slika : komponenta Prijavalspita (2)

```

56     data: {
57         id: this.state.data[index].id,
58         naziv: this.state.data[index].naziv,
59         espb: this.state.data[index].espb
60     },
61 },
62     headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
63 ).then((response) => {
64 }, (error) => {
65     console.log(error);
66 });
67     this.forceUpdateHandler();
68 }
69 odjaviIspit(id){
70     axios({
71         method: 'delete',
72         url: '/api/v1/prijave/' + id,
73         baseURL: 'http://localhost:8080',
74         data: {
75         },
76         headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
77     }).then((response) => {
78 }, (error) => {
79     console.log(error);
80 });
81     this.forceUpdateHandler();
82 }

```

Slika : komponentea Prijavalspita (3)

```
81     |     this.forceUpdateHandler();
82   }
83   render(){
84
85
86   return (
87     <div className='PrijavaIspita'>
88       <h1>Prijava ispita</h1>
89       <table class="table">
90         <thead class="thead-dark">
91           <tr>
92             <th scope="col">#</th>
93             <th scope="col">Predmet</th>
94             <th scope="col">ESPB</th>
95             <th scope="col">Prijavi</th>
96           </tr>
97         </thead>
98         <tbody>
99           {this.state?.data.map((el, index) => <tr><td>{index+1}</td><td>{el.naziv}</td><td>{el.espb}</td><td>{el.prijavi}</td></tr>)};
100       </tbody>
101     </table>
102     <h1>Prijavljeni ispiti</h1>
103     <table class="table">
104       <thead class="thead-dark">
105         <tr>
106           <th scope="col">#</th>
```

Slika: komponenta Prijavalspita (4)

```
102     <h1>Prijavljeni ispiti</h1>
103     <table class="table">
104       <thead class="thead-dark">
105         <tr>
106           <th scope="col">#</th>
107           <th scope="col">Predmet</th>
108           <th scope="col">ESPB</th>
109         </tr>
110       </thead>
111       <tbody>
112         {this.state?.dataPrijavljeni.map((el, index) => <tr><td>{index+1}</td><td>{el.naziv}</td><td>{el.espb}</td></tr>)};
113       </tbody>
114     </table>
115   </div>
116 };
117 }
118
119 export default PrijavaIspita
```

Slika: komponenta Prijavalspita (5)

```

4  function SidebarAdmin() {
5    return (
6      <div className='Sidebar2'>
7        <div className='Sidebar'>
8          <nav id="sidebarMenu" className="collapse d-lg-block sidebar collapse bg-white">
9            <div className="position-sticky">
10              <div className="list-group list-group-flush mx-3 mt-4">
11                <Link
12                  to="/obavestenja"
13                  className="list-group-item list-group-item-action py-2 ripple"
14                  aria-current="true"
15                >
16                  <i className="fas fa-tachometer-alt fa-fw me-3" style="font-size: 1.5em;"></i><span>Obavestenja</span>
17                </Link>
18                <Link
19                  to="/upisiOcenu"
20                  className="list-group-item list-group-item-action py-2 ripple"
21                  aria-current="true"
22                >
23                  <i className="fas fa-tachometer-alt fa-fw me-3" style="font-size: 1.5em;"></i><span>Upisi ocenu</span>
24                </Link>
25
26                <Link to="/molbeUObradi" className="list-group-item list-group-item-action py-2 ripple">
27                  <i className="fas fa-chart-line fa-fw me-3" style="font-size: 1.5em;"></i><span>Molba</span>
28                </Link>
29                <Link to="/student" className="list-group-item list-group-item-action py-2 ripple">
30                  <i className="fas fa-user fa-fw me-3" style="font-size: 1.5em;"></i><span>Student</span>
31                </Link>

```

Slika : komponenta SidebarAdmin (deo)

```

7  function UnosObavestenja() {
8    function potvrdiUnos(){
9      axios({
10        method: 'post',
11        url: '/api/v1/obavestenja',
12        baseURL: 'http://localhost:8080',
13        data: {
14          sadrzaj: document.getElementById("text").value
15        },
16        headers: { 'Authorization': 'Bearer ' +localStorage.getItem('token')}
17      }).then((response) => {
18      }, (error) => {
19        console.log(error);
20      });
21
22      window.location.href = "/obavestenja"
23    }
24    return (
25      <div className='UnosMolbe'>
26        <h1>Unos obavestenja</h1>
27        <div className="form-group">
28          <textarea className="form-control" id="text" rows="3" style="height: 100px;"></textarea>
29        </div>
30        <br><br>
31        <button type="button" className="btn btn-outline-danger" onClick={potvrdiUnos}>Potvrdi</button>

```

Slika : komponenta UnosObavestenja (1)

```

24     return (
25         <div className='UnosMolbe'>
26             <h1>Unos obavestenja</h1>
27             <div class="form-group">
28                 <textarea class="form-control" id="text" rows="3"></textarea>
29             </div>
30             <br><br>
31             <button type="button" class="btn btn-outline-danger" onClick={potvrdiUnos}>Potvrди</button>
32         </div>
33     );
34 }
35
36 export default UnosObavestenja

```

Slika: komponenta UnosObavestenja (2)

```

8  class UpisiOcenu extends React.Component {
9
10    state = {
11      data: [],
12      dataPrijavljeni: []
13    }
14
15    builder(){
16      this.forceUpdateHandler = this.forceUpdateHandler.bind(this);
17    };
18
19    forceUpdateHandler(){
20      this.forceUpdate();
21    };
22
23    componentDidMount() {
24      //POSLATI AXIOS ZAHTEV ZA SVE PRIJAVE
25      axios({
26        method: 'get',
27        url: '/api/v1/prijave',
28        baseURL: 'http://localhost:8080',
29        data: {},
30        headers: { 'Authorization': 'Bearer ' + localStorage.getItem('token') }
31      }).then((response) => {
32        console.log(response);

```

Slika : Komponenta UpisiOcenu (1)

```
    32     console.log(response);
  33     this.setState([dataPrijavljeni: response.data])
  34     console.log(this.state.data);
  35   }, (error) => {
  36     console.log(error);
  37   });
  38 }
 39 render(){
 40   //OMOGUCI DA SE UPISE OCENA I POTVRDI UNOS
 41   return (
 42     <div className='PrijavaIspita'>
 43       <h1>Prijava</h1>
 44       <table class="table">
 45         <thead class="thead-dark">
 46           <tr>
 47             <th scope="col">#</th>
 48             <th scope="col">Predmet</th>
 49             <th scope="col">Student</th>
 50             <th scope="col">Ocena</th>
 51             <th scope="col">Potvrdi</th>
 52           </tr>
 53         </thead>
 54         <tbody>
 55           {this.state?.data.map((el, index) => <tr><td>{index+1}</td><td>{el.naziv}</td><td>{el.ocena}</td><td>{el.potvrda}</td></tr>)};
 56       </tbody>
 57     </table>
 58   </div>
 59 )
```

Slika: Komponenta UpisiOcenu (2)