

ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU



WEB APLIKACIJA ZA PRAVLJENJE RASPOREDA ZA LAB VEŽBE

–Diplomski rad–

Kandidat:

Zorana Stefanović 2009/114

Mentor:

doc. dr Zoran Čiča

Beograd, Oktobar 2014.

SADRŽAJ

SADRŽAJ	2
1. UVOD	3
2. PROJEKTNI ZAHTEVI	4
2.1. PROJEKTNI ZAHTEVI SA STANOVIŠTA STUDENATA	4
2.2. PROJEKTNI ZAHTEVI SA STANOVIŠTA ADMINISTRATORA	4
3. OPIS REALIZACIJE	5
3.1. FORMIRANJE GRUPA	5
3.1.1. <i>Formiranje grupa 2+1</i>	5
3.1.2. <i>Formiranje grupa 1+1+1</i>	7
3.1.3. <i>Pravljenje liste termina formiranih grupa</i>	7
3.2. DODELA TERMINA	9
4. UPUTSTVO ZA KORIŠĆENJE APLIKACIJE	15
4.1. STUDENTSKI DEO APLIKACIJE	15
4.2. ADMINISTRATORSKI DEO APLIKACIJE	19
5. ZAKLJUČAK	27
LITERATURA	28

1. UVOD

U početku osmišljen kao sredstvo za razmenu informacija između naučnih instituta, prvobitni WWW (*World Wide Web*) je predstavljao skup hipertekst stranica. Ove veb strane su bile statične pa je najviši nivo interakcije korisnika sa veb sajtovima bilo najobičnije pregledanje strana. Međutim, komercijalizacija WWW, a samim tim i ogroman broj novih korisnika, stvorili su potrebu za dinamičkim veb sajtovima što je dovelo do razvoja nove oblasti informacionih tehnologija – veb dizajna.

Veb dizajn je jedna od najperspektivnijih i najbrže rastućih industrija današnjice. Podrazumeva projektovanje i izradu dinamičkih veb aplikacija koje se oslanjaju na baze podataka i pružaju korisnicima mogućnost izmene sadržaja veb strana ili čuvanja i unosa podataka radi njihove dalje obrade.

Dizajn bilo koje veb aplikacije može se rastaviti na dve celine:

- Deo koji se izvršava na serverskoj strani komunikacije
- Deo koji se izvršava na klijentskoj strani komunikacije

Za izradu obe celine postoji širok dijapazon programskih jezika. Aplikacija diskutovana u ovom radu napravljena je korišćenjem PHP (*Personal Home Page*) zajedno sa MySQL (*My Structured Query Language*) bazom podataka, za serverski deo, dok je korisnički deo napisan pomoću HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) i *JavaScript* jezika.

Glavni cilj ovog rada je da se studentima obezbedi mogućnost da izaberu termine za izradu laboratorijskih vežbi koji su u skladu sa njihovim rasporedom. Pošto se studentima delimično prepušta odluka o terminu u kome će obaviti laboratorijsku vežbu na određenom predmetu, na ovaj način se teorijski u potpunosti može izbeći preklapanje termina sa različitim predmeta i olakšati studentima ispunjavanje obaveza tokom školske godine. Takođe, svrha aplikacije je na jednom mestu pružiti studentima uvid u raspored laboratorijskih vežbi sa svih predmeta koje slušaju.

Nastavak rada sastoji se iz četiri celine. U prvom poglavlju rada će se govoriti o postavljenim zahtevima projekta. Drugo poglavlje obuhvata kompletan opis tehničkog rešenja sa posebnim naglaskom na kodove odgovorne za formiranje grupa i dodelu termina. Takođe će biti dat kratak pregled mogućih proširenja i unapređenja performansi aplikacije. Treće poglavlje predstavlja detaljno uputstvo za korišćenje, najpre sa stanovišta studenta, a potom i sa stanovišta administratora. Na samom kraju, u zaključku biće napravljen kratak rezime rada sa mogućnostima primene aplikacije.

2. PROJEKTNI ZAHTEVI

Kao što je navedeno u uvodnom poglavlju, ovde će ukratko biti predstavljene postavke za realizaciju veb aplikacije.

Osnovni projektni zahtev aplikacije je da izvrši dodelu termina grupama. Takođe je potrebno prethodno formirati grupe tako da se sastoje od tačno određenog broja studenata. Pored osnovnog, postoje i projektni zahtevi sa stanovišta korisnika.

Oni se mogu podeliti na dve grupe:

- Projektni zahtevi sa stanovišta studenata
- Projektni zahtevi sa stanovišta administratora

2.1. PROJEKTNI ZAHTEVI SA STANOVIŠTA STUDENATA

Omogućiti studentima da se preko svojih naloga prijavljuju na laboratorijske vežbe za određeni predmet. Studenti imaju opciju da prijave najviše dvoje kolega sa sobom u grupi. Potom, student iz spiska termina definisanih za dati predmet bira najpovoljnije i rangira svaki na skali od jedan do tri, gde tri predstavlja najviši nivo prioriteta. Nakon inicijalnog prijavljivanja, izabrani termini, kao i njihov prioritet, se mogu menjati sve dok ne istekne rok prijave.

2.2. PROJEKTNI ZAHTEVI SA STANOVIŠTA ADMINISTRATORA

Administrator ima potpunu kontrolu nad prijavljenim grupama. Može zasebno obrisati članove grupe ili čitavu grupu, kao i izmeniti njihovu listu i rangiranje izabranih termina. Takođe je zadužen za unos termina na svim predmetima i dodavanje studenata na liste predmeta. Nakon isteka roka prijave, administrator zaključava obrasce za prijavu na određenom predmetu.

3. OPIS REALIZACIJE

Ovo poglavlje posvećeno je implementiranom rešenju. Pošto će korisnički deo biti detaljno opisan u narednom poglavlju, ovde će se najviše pažnje posvetiti programskom kodu za formiranje grupa i dodelu termina. Pre nego što se pređe na odeljke koji će objasniti svaki od njih zasebno, potrebno je prvo dati par reči o kriterijumima koji su korišćeni za odlučivanje.

To su prosečna ocena i opterećenost studenata, odnosno grupe kojoj pripadaju. Pod pojmom opterećenost podrazumeva se broj predmeta koje student sluša odnosno prosečan broj predmeta koje grupa sluša. O tome kako bi se ovaj kriterijum mogao dinamičnije primeniti u budućim verzijama biće reči na samom kraju poglavlja.

3.1. FORMIRANJE GRUPA

Prvi problem koji se morao rešiti pre početka dodele termina, jeste formiranje grupa od najviše tri člana. Naime, u nekim slučajevima studenti će prijaviti samo sebe ili grupu od dva člana, pa je radi efikasnijeg iskorišćenja vremenskog intervala predviđenog za realizaciju vežbi neophodno najpre napraviti grupe od po troje. Ovaj problem se može rastaviti na dva dela:

- Formiranje grupa 2+1
- Formiranje grupa 1+1+1

3.1.1. Formiranje grupa 2+1

Pri formiranju ovog tipa grupe, glavni uticaj dat je kriterijumu opterećenosti a prednost je data grupama sa dva člana. Iz baze podataka se prvo izaberu dvočlane grupe i poređaju po opterećenosti, a zatim se za svaku redom traži jednočlana grupa čiji se izbor termina najviše poklapa sa izborom termina date grupe. Kod koji traži najveći broj poklapanja prikazan je ispod zajedno sa komentarima na ključnim delovima.

```
# $termini2 i $termini1 su nizovi izabranih termina dvočlane i jednočlane grupe
respektivno

# indeksi ovih nizova su id termina u tabeli termina, a vrednosti su prioriteti
koje im je grupa dodelila

$termini_poklapanje=array_intersect_key($termini1,$termini2);

if ($broj>1)
{
    # tražimo broj poklapanja izabranih termina

    $c=count($termini_poklapanje);

    $t=array_keys($termini_poklapanje);
```

```

$p=0;

# tražimo sada poklapanje dodeljenih prioriteta među izabranim terminima
grupa

# i na osnovu visine prioriteta svakom poklapanju dodajemo
određeni broj poena

foreach($t as $value)
{
    if ($termini1['.$value.']==$termini2['.$value.'])
    {
        if ($termini1['.$value.']==3)
        {
            $p=$p+2;
        }
        else if ($termini1['.$value.']==2)
        {
            $p=$p+1.5;
        }
        else if ($termini1['.$value.']==1)
        {
            $p=$p+1;
        }
    }
    else if (abs($termini1['.$value.']-$termini2['.$value.'])==1)
    {
        $p=$p+0.5;
    }
}

unset($value);

$poklapanje=$c+$p;

# ukoliko je konačan broj poklapanja veći od broja poklapanja prethodne
jednočlane grupe

# trenutna jednočlana grupa postaje najpovoljniji kandidat za spajanje

if ($poklapanje>$broj_poklapanja)
{
    $broj_poklapanja=$poklapanje;

    $gp=$grupal;

    $tp=$termini_poklapanje;

    $termini_grupel=$termini1;
}

```

```

    $prosek_grupel=$ispis1['prosek_grupe'];

    $opterecenost_grupel=$ispis1['opterecenost_grupe'];

    $clanovi_grupel=$ispis1['clanovi'];

    $vreme_prijave_grupel=$ispis1['vreme_prijave'];

}

}

#ako je pronađeno više od jedne jednočlane grupe, vršiće se ispitivanje
poklapanja

#u suprotnom se ovaj deo preskače i odmah se postojeća jednočlana grupa
proglašava za najpovoljniju

else if ($broj1==1)
{
    $gp=$grupal;

    $tp=$termini_poklapanje;

    $termini_grupel=$termini1;

    $prosek_grupel=$ispis1['prosek_grupe'];

    $opterecenost_grupel=$ispis1['opterecenost_grupe'];

    $clanovi_grupel=$ispis1['clanovi'];

    $vreme_prijave_grupel=$ispis1['vreme_prijave'];

}

```

3.1.2. *Formiranje grupa 1+1+1*

Traženje najvišeg broja poklapanja između tri grupe je kompleksniji problem u odnosu na obavljanje istog postupka za dve grupe. Stoga primenjeno jednostavno spajanje grupa po redu a pritom se i dalje pridržava kriterijuma opterećenosti.

3.1.3. *Pravljenje liste termina formiranih grupa*

Nakon nalaženja najvećeg broja poklapanja i spajanja grupa, dolazimo do sledećeg problema: za novonastalu grupu treba napraviti listu izabраниh termina i njihovih prioriteta.

Termini koji su jedinstveni za jednu grupu, kao i termini čiji se prioriteta poklapaju za sve grupe, ostaju nepromenjeni i odmah se dodaju nizu termina nove grupe. Na prioritete ostalih termina mora se primeniti logika odlučivanja.

Kod koji sledi predstavlja logiku odlučivanja u slučaju spajanja dve grupe. Za slučaj spajanja tri grupe, kod će biti priložen samo u elektronskom obliku zbog svoje obimnosti.

```

#tražimo termine koji ne pripadaju drugoj, odnosno prvoj grupi, respektivno
$termini_razl=array_diff_key($termini_grupe1,$termini_grupe2);
$termini_raz2=array_diff_key($termini_grupe2,$termini_grupe1);
# pravimo niz jedinstvenih termina
$termini_razlika=$termini_razl+$termini_raz2;
$termini_poklapanje=array_intersect_key($termini1,$termini2);
$tp=$termini_poklapanje;
$termini_pok=array();
$tp0=array_keys($tp);
foreach ($tp0 as $value)
{
    # ako se prioriteti termina poklapaju on ostaju neizmenjeni

    if ($termini_grupe1['.$value.']==$termini_grupe2['.$value.'])
    {
        $termini_pok['.$value.']=$termini_grupe2['.$value.'];
    }

    # ukoliko se prioriteti termina maksimalno razlikuju, uvek se dodaje
    # prioriteta tri

    # na taj način se trudimo da što bolje očuvamo najpovoljnije termine obe
    # grupe

    else if (abs($termini_grupe1['.$value.']-$termini2['.$value.'])==2)
    {
        $termini_pok['.$value.']=3;
    }
    else
    {
        # u slučaju kada je razlika između prioriteta jedan, prednost dajemo
        # grupi sa većom opterećenošću

        if($opterecenost_grupe1>$opterecenost_grupe2)
        {
            $termini_pok['.$value.']=$termini_grupe1['.$value.'];
        }
        else if ($opterecenost_grupe1<$opterecenost_grupe2)
        {
            $termini_pok['.$value.']=$termini_grupe2['.$value.'];
        }
        else if ($prosek_grupe1<$prosek_grupe2)

        # ukoliko je kriterijum neodlučan po opterećenosti, gleda se
        # prosečna ocena grupa
        {
            $termini_pok['.$value.']=$termini_grupe2['.$value.'];
        }
        else if ($prosek_grupe1>$prosek_grupe2)
    }
}

```



```

    {
        $termini_pok[''.$value.'']=$termini_grupe1[''.$value.''];
    }

    # u slučaju da je kriterijum i ovde neodlučan, prednost ima grupa sa
    više članova
    else
    {
        $termini_pok[''.$value.'']=$termini_grupe2[''.$value.''];
    }
}

unset($value);

$termini=$termini_pok+$termini_razlika;

```

Ovim smo kompletirali formiranje grupa. Naglašavamo da bi trebalo da buduće verzije aplikacije budu unapređene prvenstveno da se za slučaj grupa 1+1+1 takođe vrši pretraživanje za najvećim brojem poklapanja termina. Isto tako, mogli bi se, npr. uvesti međunivoi prioriteta kako bi se što efikasnije i približnije očuvale želje svake grupe koja učestvuje u formiranju nove.

3.2. DODELA TERMINA

Sada kada smo obavili formiranje grupa, možemo preći na dodelu termina. Glavna ideja je da se grupe sortiraju prema određenom kriterijumu, a potom im se na osnovu njihovih želja i dostupnosti termina dodeli najpovoljniji mogući termin za izradu laboratorijskih vežbi.

Prvo će se formirati multidimenzionalni niz od podnizova koji predstavljaju grupe. Svaki podniz sadržaće dve vrednosti, prosečnu ocenu i prosečnu opterećenost grupe. Kod zadužen za ovaj deo je prikazan ispod.

```

$upit="SELECT * FROM grupe WHERE predmet=$predmet";

$provera=mysql_query($upit)or die(mysql_error());

while($ispis=mysql_fetch_assoc($provera))
{
    $grupe[$ispis['grupaid']][1]=$ispis['prosek_grupe']*100;
    $grupe[$ispis['grupaid']][2]=$ispis['opterecenost_grupe']*100;
}

```

Sortiranje niza vrši se pomoću sledeće funkcije:

```

function grp($a, $b)
{
    if ($a['2'] == $b['2'])
    {
        return ($a['1'] < $b['1']) ? 1 : -1;;
    }
    return ($a['2'] < $b['2']) ? 1 : -1;
}

#aktiviramo funkciju sortiranja
uasort($grupe, 'grp');

```

Iz priloženih kodova može se primetiti da je primarni kriterijum prosečna opterećenost grupe. Naravno, moguće je obrnuti prioritete kriterijuma.

Nakon sortiranja grupa, počinje i sama dodela termina. Na isti način kao u slučaju formiranja grupa, pravi se niz izabranih termina grupe, kao i niz dostupnih termina na datom predmetu.

```
$upit1="SELECT terminid, status_t, termin, prioritet FROM termini, grupe_termini  
WHERE (predmet=$predmet AND status_t='N') AND grupa=$grupa";
```

```
$provera1=mysql_query($upit1)or die(mysql_error());
```

```
while ($ispis1=mysql_fetch_assoc($provera1))  
{  
    $termini[''.$ispis1['terminid'].'']=$ispis1['status_t'];  
    $termini_grupe[''.$ispis1['termin'].'']=$ispis1['prioritet'];  
}
```

Zatim se vrši poređenje ova dva niza:

```
$dostupni_termini=array_intersect_key($termini_grupe,$termini);
```

Ukoliko nijedan od izabranih termina grupe nije dostupan, izvršava se sledeći kod:

```
$sucestalost=1;
```

```
foreach($termini as $k=>$v)
```

```
{  
    # pretražujemo izabrane termine ostalih grupa i poredimo ih sa dostupnim terminima
```

```
$upit10="SELECT COUNT(*) AS count10 FROM grupe_termini WHERE termin=$k";
```

```
$provera10 = mysql_query($upit10) or die(mysql_error());
```

```
$ispis10=mysql_fetch_assoc($provera10);
```

```
# ukoliko se među dostupnim terminima nadje neki koji nije izabran od strane nijedne grupe, automatski se dodeljuje trenutnoj
```

```
if ($ispis10['count10']==0)
```

```
{  
    $td=$k; break;  
}
```

```
# ukoliko se ne nadje takav termin, traži se najmanje biran termin i što moguće nižeg prioriteta dodeljenog mu od strane ostalih grupa
```

```
else
```

```
{  
    $sucestalost0=$ispis10['count10'];
```

```
$upit11="SELECT COUNT(*) AS count11 FROM grupe_termini WHERE  
termin=$k AND prioritet='1'";
```

```
$provera11 = mysql_query($upit11) or die(mysql_error());
```

```

$ispis11=mysql_fetch_assoc($provera11);

# tražimo najmanje učestao termin prioriteta jedan

if ($ispis11['count11']>1)
{
    $sucestalost11=$ispis11['count11'];

    $sucestalost_prioriteta=$sucestalost11/$sucestalost0;

    if ($sucestalost_prioriteta<=$sucestalost)
    {
        $sucestalost=$sucestalost_prioriteta;

        $td=$k;
    }
}
else if ($ispis11['count11']==1)
{
    $td=$k;break;
}
else
{
    $upit12="SELECT COUNT(*) AS count12 FROM grupe_termini WHERE
termin=$k AND prioritet='2'";

    $provera12 = mysql_query($upit12) or die(mysql_error());

    $ispis12=mysql_fetch_assoc($provera12);

    # tražimo najmanje učestao termin prioriteta dva

    if ($ispis12['count12']>1)
    {
        $sucestalost12=$ispis12['count12'];

        $sucestalost_prioriteta=$sucestalost12/$sucestalost0;

        if ($sucestalost_prioriteta<=$sucestalost)
        {
            $sucestalost=$sucestalost_prioriteta;

            $td=$k;
        }
    }
    else if ($ispis12['count12']==1)
    {
        $td=$k;break;
    }
    else
    {
        $upit13="SELECT COUNT(*) AS count13 FROM grupe_termini
WHERE termin=$k AND prioritet='3'";

        $provera13 = mysql_query($upit13) or die(mysql_error());

        $ispis13=mysql_fetch_assoc($provera13);
    }
}

```

```

# tražimo najmanje učestao termin prioriteta tri
if ($ispis13['count13']>1)
{
    $sucestalost13=$ispis13['count13'];

    $sucestalost_prioriteta=$sucestalost13/$sucestalost0;

    if ($sucestalost_prioriteta<=$sucestalost)
    {
        $sucestalost=$sucestalost_prioriteta;

        $td=$k;
    }
}
else if ($ispis13['count13']==1)
{
    $td=$k;break;
}
else
{
    echo 'Greška.';
}
}
}
}
}

```

Ako među izabranim terminima grupe postoje dostupni, od njih se izdvajaju oni sa najvišim mogućim nivom prioriteta.

```

$prioritet3 = array_filter($dostupni_termini, "tri");

if (count($prioritet3)!=0)
{
    if (count($prioritet3)==1)
    {
        $td=key($prioritet3);
    }
    else
    {
        $prioritet=$prioritet3;
    }
}
else
{
    $prioritet2=array_filter($dostupni_termini, "dva");

    if (count($prioritet2)!=0)
    {
        if (count($prioritet2)==1)
        {
            $td=key($prioritet2);
        }
        else
        {
            $prioritet=$prioritet2;
        }
    }
}
}

```

```

    }
}
else
{
    $prioritet1=array_filter($dostupni_termini, "jedan");

    if (count($prioritet1)!=0)
    {
        if (count($prioritet1)==1)
        {
            $td=key($prioritet1);
        }
        else
        {
            $prioritet=$prioritet1;
        }
    }
}
}
}

```

Funkcije pozivane u prethodnom delu koda:

```

function tri($var3)
{
    if($var3==3)
    {
        return($var3);
    }
}

function dva($var2)
{
    if($var2==2)
    {
        return($var2);
    }
}

function jedan($var1)
{
    if($var1==1)
    {
        return($var1);
    }
}

```

Na kraju, na identičan način kao u slučaju kada među izabranim terminima grupe nije bilo dostupnih termina, traži se termin najređe rangiran nivoom prioriteta koji mu je dodelila tekuća grupa. Ovakva logika pravljenja rasporeda prvenstveno ima za cilj da u što moguće većoj meri grupama dodeli jedan od termina kojima su dale najviši nivo prioriteta tako što će im dati upravo onaj koji je od strane ostalih grupa najređe rangiran istim nivoom prioriteta.

Može se zapaziti da opterećenost grupe ima glavni uticaj na formiranje grupa i dodelu termina. Pošto je suštinski glavna svrha aplikacije organizacija vremena, jednostavno je izgledalo prirodno dati prednost studentima sa više obaveza i prilagoditi postupak pravljenja rasporeda njihovim potrebama. Kao što je rečeno na početku poglavlja, ovaj kriterijum se uz nešto

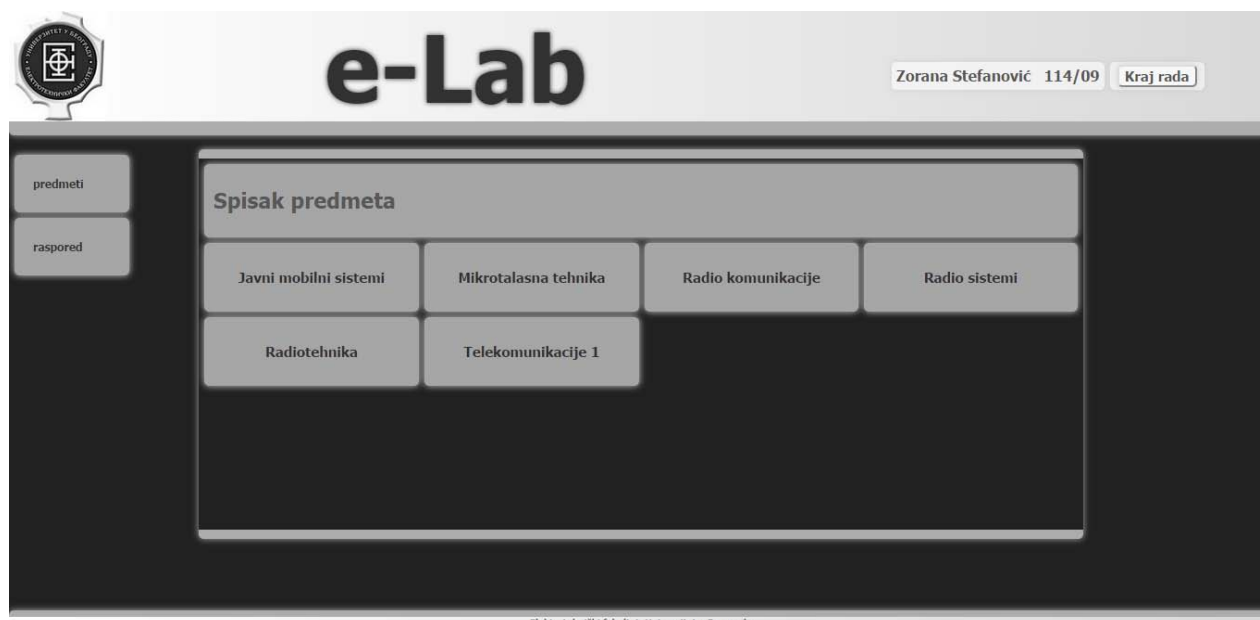
kompleksniju implementaciju može učiniti dinamičnijim u smislu da umesto ukupnog broja predmeta, predstavlja broj predmeta na kojima je student trenutno prijavljen za laboratorijske vežbe. Takođe, ovako definisana opterećenost može biti i potpuno zaseban kriterijum u budućim verzijama aplikacije.

4. UPUTSTVO ZA KORIŠĆENJE APLIKACIJE

U ovom delu rada čitalac će se upoznati sa korisničkim interfejsom aplikacije. Prvo će biti prikazan deo namenjen studentima, a potom administratorski deo aplikacije.

4.1. STUDENTSKI DEO APLIKACIJE

Po prijavljivanju na svoj nalog, studentu se otvara početna stranica sa tabelom predmeta koje sluša u tekućoj školskoj godini (slika 4.1).

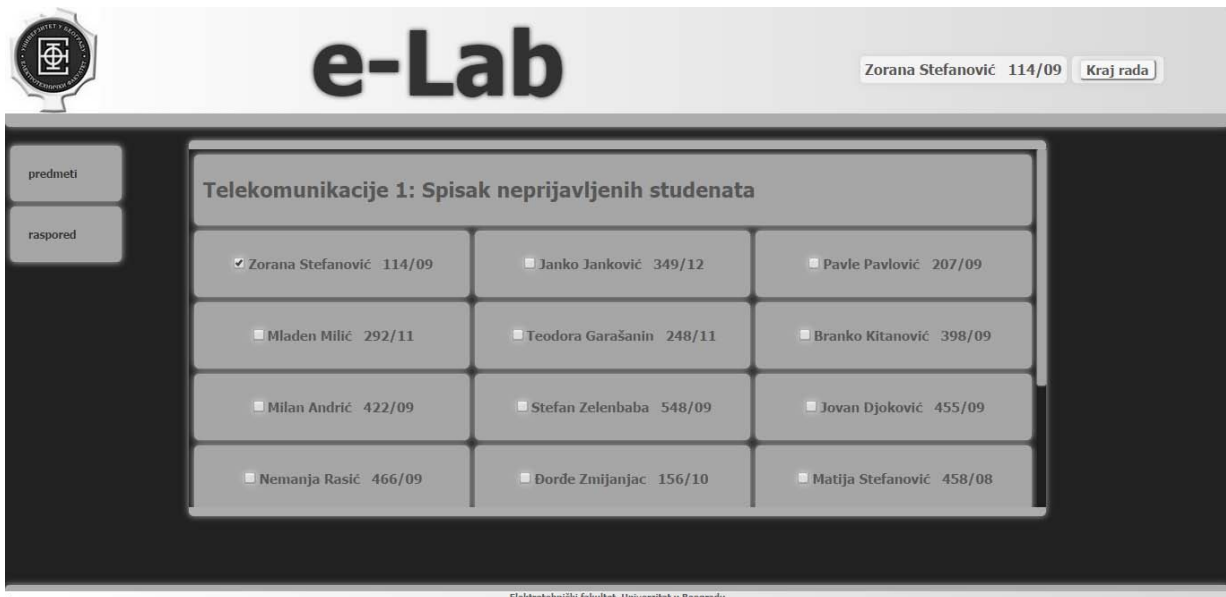


Slika 4.1 - Izgled početne strane studenta

Sa leve strane se nalazi navigacija, link **predmeti** vodi na početnu stranu, dok link **raspored** otvara stranicu na kojoj student može videti sve grupe u kojima je prijavljen, kao i izabrane termine tih grupa. O ovome će biti više reči nešto kasnije u tekstu.

Klikom na naziv predmeta, otvoriće se različite stranice u zavisnosti od statusa studenta na tom predmetu – da li je već prijavljen ili ne, kao i od toga da li je istekao rok prijave.

Ukoliko student nije prijavio laboratorijske vežbe, a prijave su još u toku, otvara se stranica sa spisikom svih studenata koji se još uvek nisu prijavili (slika 4.2). Studentovo ime je automatski čekirano.



Slika 4.1 - Izgled stranice sa spisakom neprijavljenih studenata

Iz spiska sa slike 4.2 student može izabrati maksimalno još dvoje kolega i prijaviti grupu. Prijava se vrši klikom na dugme koje se nalazi na dnu tabele.

Nakon prijavljivanja grupe otvara se stranica sa terminima prikazana na slici 4.3.



Slika 4.2 - Izgled stranice sa spisakom termina

Iz tabele ponuđenih termina student bira najpovoljnije i određuje im prioritet čekiranjem odgovarajućeg polja, imajući u vidu da je tri najviši nivo prioriteta. Svaki termin se prijavljuje posebno klikom na dugme prijavi.

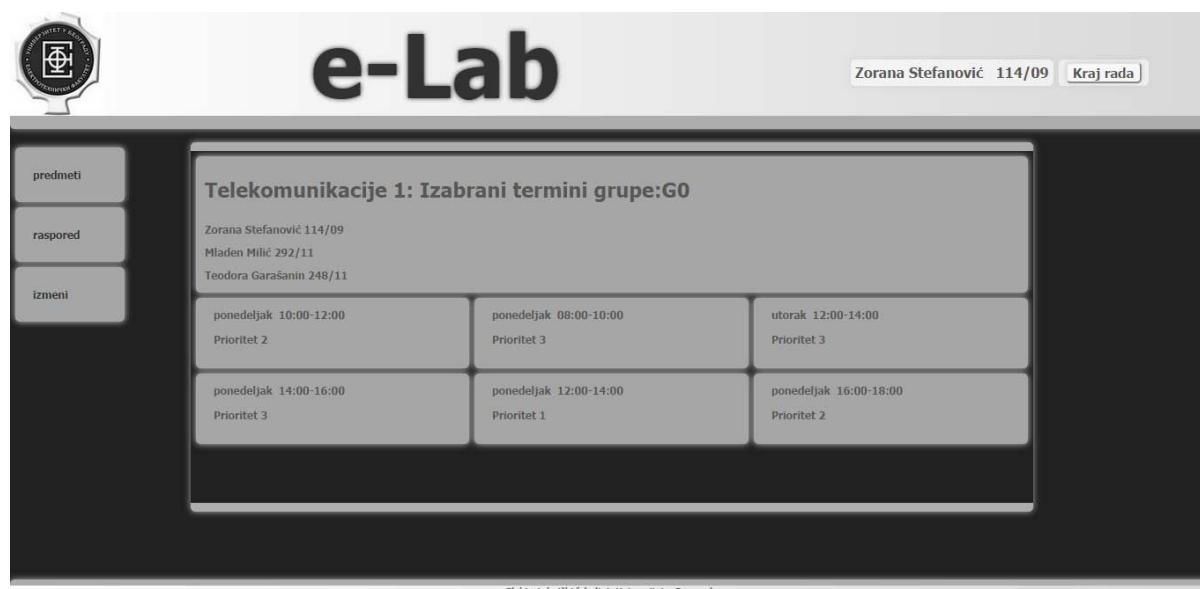
Nakon prijave termina, njegovo polje dobija izgled kao na slici 4.4.



Slika 4.3 - Izgled stranice sa spisakom termina nakon prijavljivanja

Sada je čekirano polje prioriteta koji je dodeljen datom terminu, a takođe se pojavljuju dugmići **Sačuvaj** i **Odjavi**. Pomoću njih je moguće promeniti prioritet izabranog termina ili odjaviti termin.

Po završetku prijave, student se može odjaviti klikom na dugme **Kraj rada** u gornjem desnom uglu ekrana. U slučaju da želi da nastavi sa radom, klikom na link **predmeti** se može vratiti na početnu stranicu. Ukoliko bi sada ponovo klinuo na naziv prijavljenog predmeta, u našem slučaju Telekomunikacije 1, otvorila bi se stranica prikazana na slici 4.5.



Slika 4.4 - Izgled stranice prijavljenog predmeta

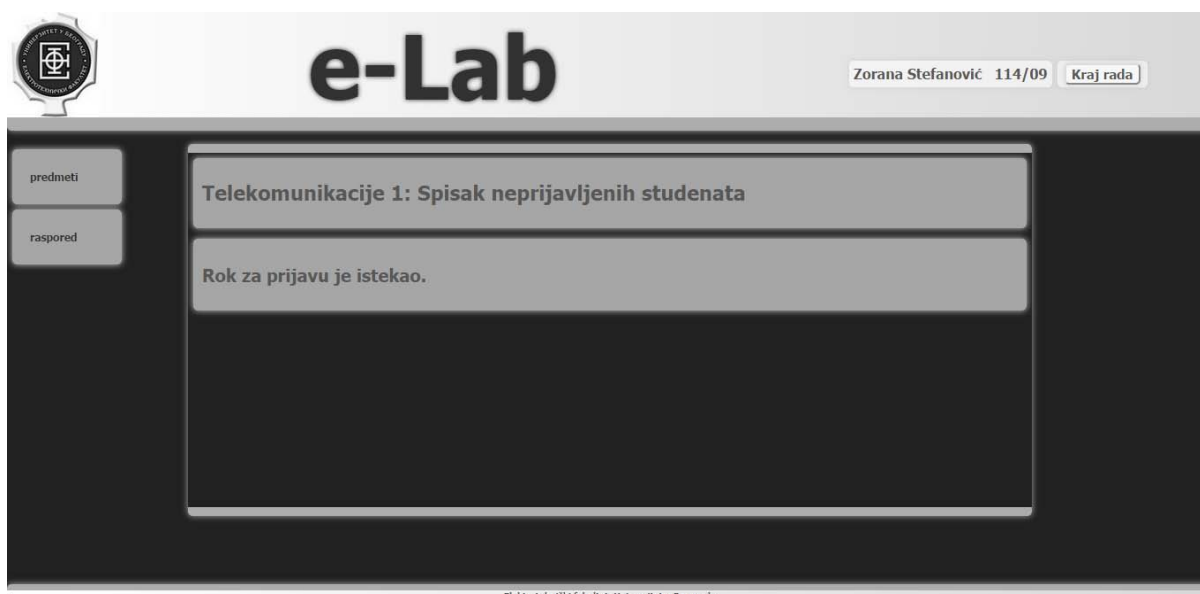
U zaglavlju tabele se nalaze imena grupe i njenih članova, dok su u tabeli prikazani izabrani termini grupe i njihovi prioriteti. Primetićemo da se u navigacionom delu na levoj strani pojavio novi link – **izmeni**. Klikom na **izmeni** otvara se nova stranica prikazana na slici 4.6.



Slika 4.5 - Izgled stranice za izmenu termina

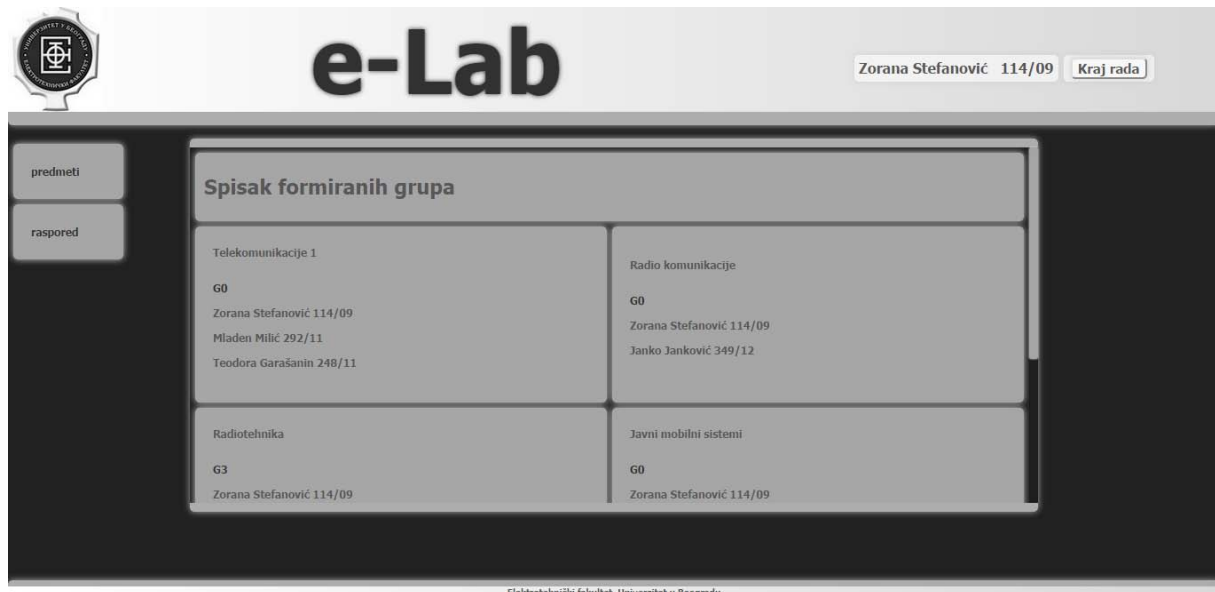
Ova stranica podeljena je u dve sekcije, sa leve strane se nalaze termini koje je grupa izabrala, a sa desne preostali termini predviđeni za dati predmet. Studenti su u mogućnosti da menjaju termine sve dok ne istekne rok prijave.

Ukoliko prijavljivanje za laboratorijske vežbe nije u toku ili je rok za prijavu istekao, otvoriće se stranica prikazana na slici 4.7.



Slika 4.6 - Izgled stranice u slučaju isteka roka prijave

Klikom na link **raspored** u navigacijskom delu, student dobija tabelu svih grupa u kojima je prijavljen, kao i imena ostalih članova grupe i predmeta na kojima je prijavljen (slika 4.8).

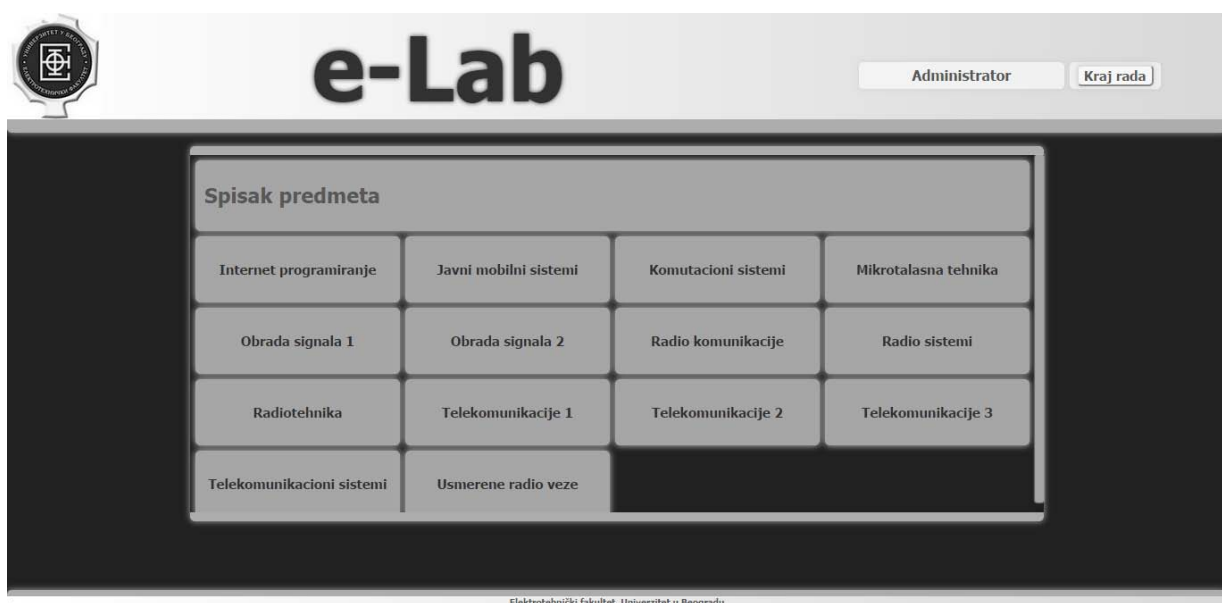


Slika 4.7 - Izgled stranice raspored

Klikom na naziv grupe otvoriće se stranica identična stranici sa slike 4.6. ukoliko je prijavljivanje još u toku, u suprotnom se prikazuje strana sa slike 4.7.

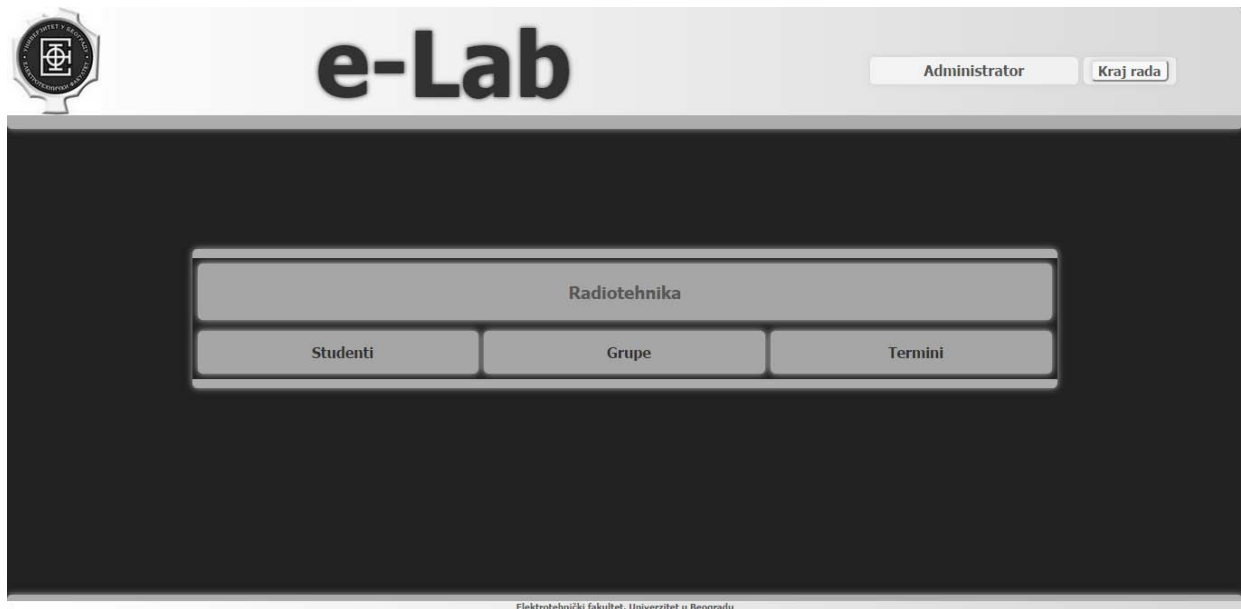
4.2. ADMINISTRATORSKI DEO APLIKACIJE

Početna stranica administratorskog dela je potpuno ista kao i za studentski deo, s tim što se u tabeli nalaze svi predmeti (slika 4.9).



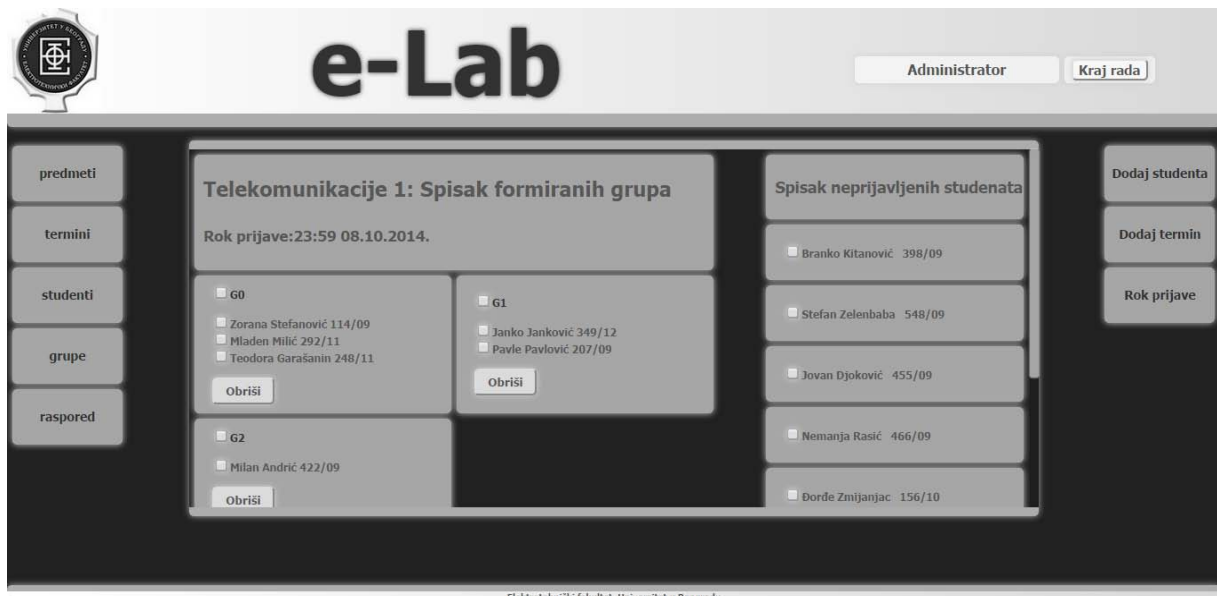
Slika 4.8 - Izgled početne stranice administratorskog dela

Klikom na naziv bilo kog od predmeta otvara se sledeća navigaciona stranica prikazana na slici 4.10.



Slika 4.9 - Izgled glavne navigacione stranice administratorskog dela

Odabirom linka **Studenti** učitava se stranica sa spiskom neprijavljenih studenata, kao i spiskom formiranih grupa (slika 4.11).

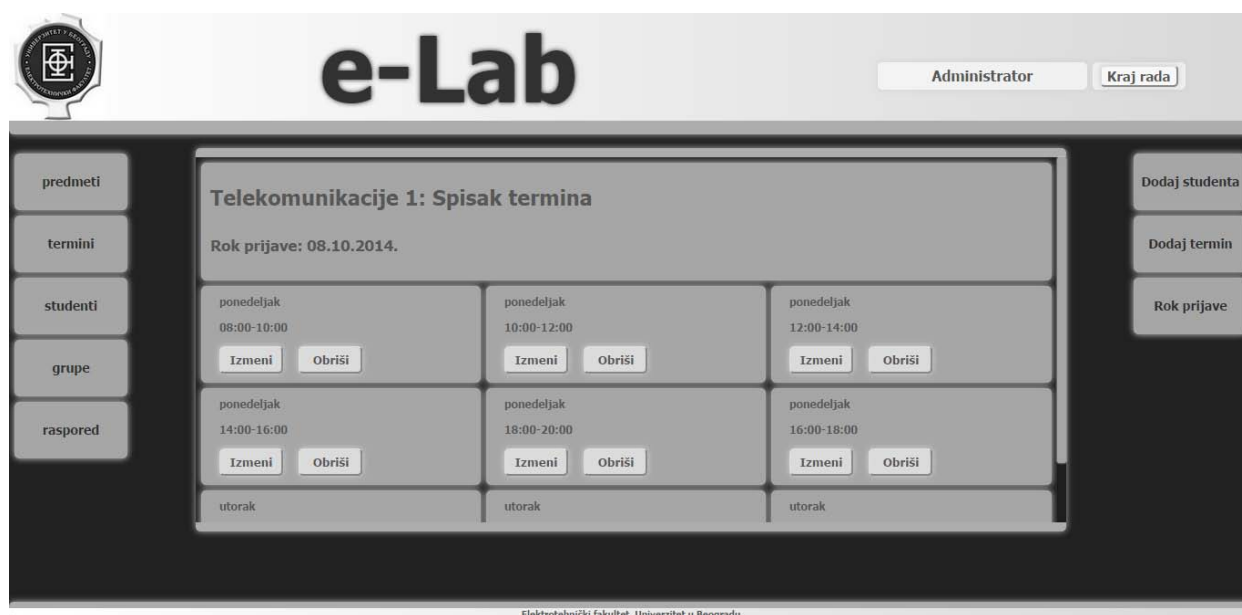


Slika 4.10 - Izgled administratorske stranice studenti

Iz spiska sa leve strane mogu se brisati pojedini članovi ili čitava grupa čekiranjem odgovarajućih polja i klikom na dugme **Obriši**. Spisak neprijavljenih studenata može biti menjan na dva načina. Mogu se brisati studenti ili prijavljivati nove grupe, naravno, i dalje uz ograničenje od najviše tri člana po grupi. Ovo se postiže čekiranjem polja pored imena studenta i klikom na odgovarajuće dugme na dnu spiska.

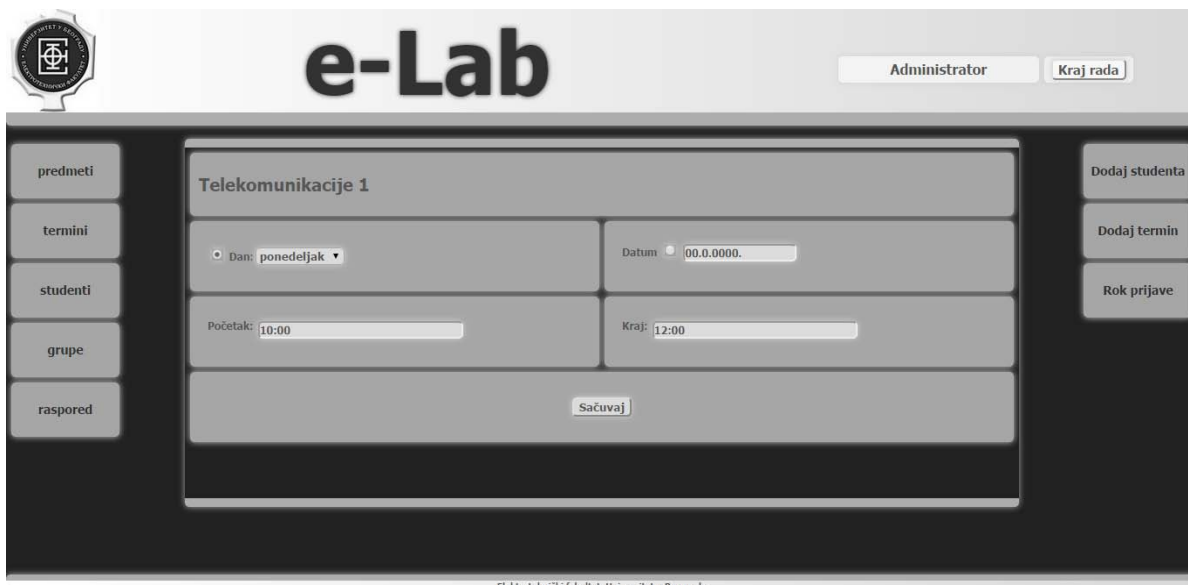
Klikom na ime grupe otvara se strana identična strani sa slike 4.6. Ovde administrator može menjati izabrane termine grupe i njihove prioritete. Uočavamo da levo i desno od centralnog dela stranice postoje dva navigaciona menija. Meni na levoj strani sadrži linkove ka stranicama **predmeti, termini, studenti, grupe i raspored**.

Klikom na link **termini**, otvara se strana sa tabelom svih termina definisanih za dati predmet (slika 4.12).



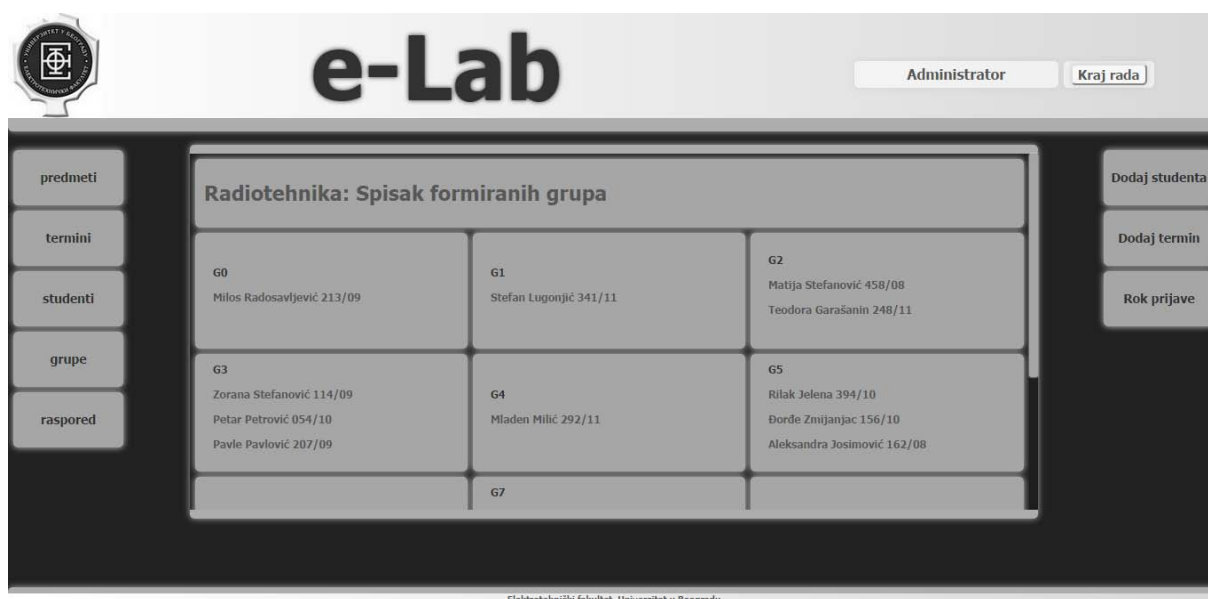
Slika 4.11 - Izgled administratorske stranice termini

Ispod datuma i satnice termina nalaze se dugmići **Izmeni** i **Obriši**. Klik na dugme **Obriši** briše termin iz baze podataka, a klik na dugme **Izmeni** vodi na stranicu prikazanu na slici 4.13.



Slika 4.12 - Izgled stranice za izmenu datuma/satnice termina

Izborom linka **grupe** iz navigacionog menija otvara se tabela svih prijavljenih grupa na predmetu kao i njihovih članova (slika 4.14).



Slika 4.13 - Izgled administratorske stranice grupe

Klikom na ime grupe učitava se strana identična strani sa slike 4.6 i omogućava administratoru manipulaciju nad izabranim terminima grupe.

Stranama **termini** i **grupe** je moguće direktno pristupiti i sa glavne navigacione stranice prikazane na slici 4.10.

U desnom navigacionom meniju nalaze se linkovi **dodaj studenta**, **dodaj termin** i **rok prijave**.

Klikom na link **dodaj studenta** dobija se prozor prikazan na slici 4.15.

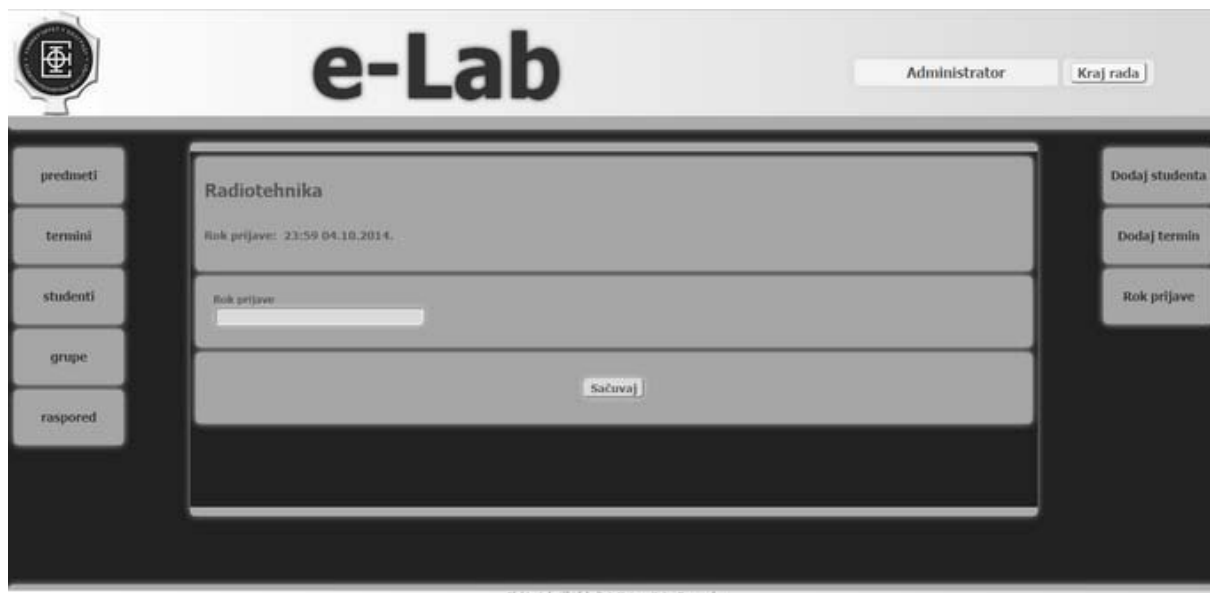


Slika 4.14 - Izgled stranice dodaj studenta

Unosom imena i prezimena i broja indeksa student se dodaje na spisak neprijavljenih za određeni predmet.

Klikom na link **dodaj termin** otvara se strana slična strani sa slike 4.13, s tim što su polja formulara prazna. Bira se da li je termin tačno definisanog datuma ili se održava periodično određenog dana u nedelji. U polja početak i kraj unosi se satnica termina.

Link **rok prijave** vodi na stranicu prikazanoj na slici 4.16.



Slika 4.15 - Izgled stranice rok prijave

Ovde administrator unosi ili menja rok prijave laboratorijskih vežbi.

Na samom kraju uputstva, govorićemo o stranici **raspored**. Do nje možemo doći klikom na istoimenu link u meniju na levoj strani. Na slici 4.17 prikazan je izgled stranice ukoliko formiranje grupa i dodela termina još nisu izvršeni i rok prijave još nije istekao.



Slika 4.16 - Izgled stranice raspored kada nije izvršeno dodeljivanje termina i prijave su još uvek u toku

Kada istekne rok prijave na određenom predmetu, u meniju sa desne strane pojavljuju se još dva dugmeta (slika 4.18).



Slika 4.17 - Izgled stranice raspored kada dodela termina još uvek nije obavljena i nakon isteka roka prijave

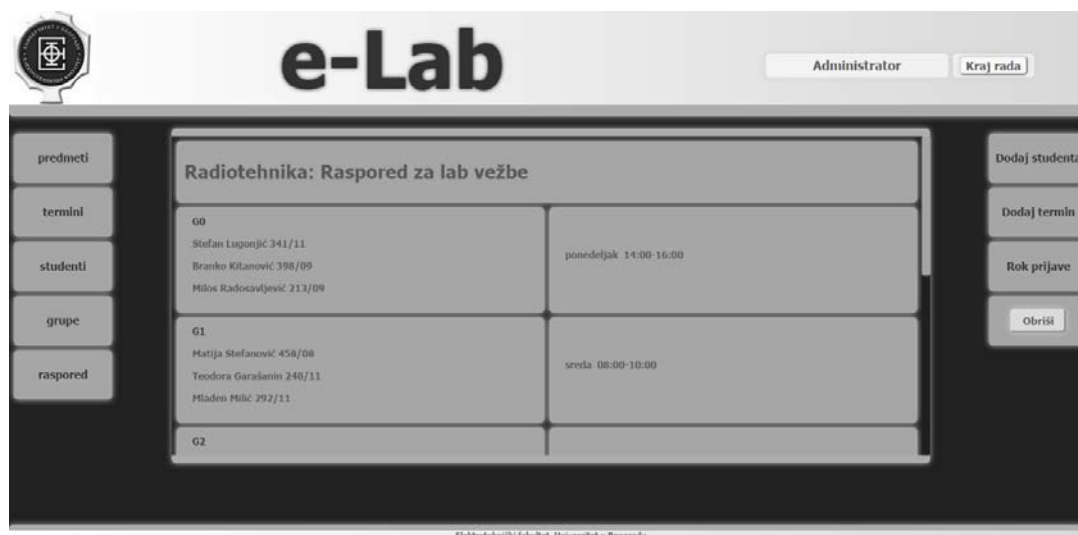
Dugmad **Grupe** i **Termini** služe za formiranje grupa i dodelu termina, respektivno. Klikom na dugme **Grupe** vrši se formiranje grupa i aplikacija nas vodi na stranicu sa slike 4.11 gde se može, ukoliko je to potrebno, naknadno izvršiti korekcija grupa.

Ukoliko sad odemo na stranicu **raspored** ona će izgledati kao što je prikazano na slici 4.19.



Slika 4.18 - Izgled stranice raspored nakon formiranja grupa

Dugme **Grupe** je sklonjeno, preostalo je samo dugme **Termini**. Klikom na njega se započinje dodela termina. Nakon obavljene dodele termina, na stranici **raspored** biće prikazana tabela grupa i termina koji je svaka od njih dobila.



Slika 4.19 - Izgled stranice raspored nakon dodele termina

Pojavilo se novo dugme, **Obrisi**, koje služi za brisanje čitavog rasporeda.

Glavna odlika celokupnog korisničkog dela jeste da se sve promene vrše u okviru predmeta izabranog na početnoj strani. Takođe, obrasci se zaključavaju automatski nakon isteka roka prijave,

pa nije potrebno učešće od strane administratora. Ipak, on može uticati na vreme zaključavanja tako što će izmeniti rok prijave.

5. ZAKLJUČAK

U ovom radu je realizovana veb aplikacija za izradu rasporeda za lab vežbe. Aplikacija omogućava da studenti na jednostavan način prijavljuju lab vežbe, kao i termine izrade lab vežbi u cilju da postignu što optimalniji raspored svojih obaveza. Takođe, sami nastavnici su rasterećeni obaveza oko evidencije prijave studenata što im olakšava njihove nastavne obaveze. Cilj aplikacije je da se omogući automatizovan proračun rasporeda lab vežbi za veći broj predmeta čime bi se izbegli i problemi preklapanja termina lab vežbi za istog studenta što je takođe čest problem s kojim se susreću studenti.

Naravno, ovo je samo prva verzija veb aplikacije koja ima prostora za dalja unapređenja. Jedno od unapređenja se može odnositi na poboljšani unos termina koji bi definisao maksimalan broj studenata po terminu pa bi na taj način u jedan termin moglo da stane više grupa. Ovaj pristup je podržan i u trenutnoj verziji veb aplikacije, ali je unos nešto komplikovaniji. Drugo unapređenje je dodatna optimizacija algoritma za spajanje grupa u termin, pre svega sa stanovišta slučaja 1+1+1. Pored navedenog, bilo bi potrebno dodatno anketiranje nastavnika da bi se veb aplikacija prilagodila u potpunosti njihovim potrebama i zahtevima. Ali, važno je navesti da ova verzija veb aplikacije predstavlja dobru osnovu za dalji razvoj i optimizaciju koja bi mogla doprineti poboljšanju i olakšavanju ispunjavanja obaveza kako studenata, tako i nastavnika.

LITERATURA

- [1] Larry Ullman “*PHP 6 and MySQL 5*”, Peachpit Press, 2008
- [2] <http://stackoverflow.com/>
- [3] <http://www.w3schools.com/>
- [4] <http://php.net/>
- [5] <http://dev.mysql.com/>