

# **SPECIFIKACIJA PROJEKTOG ZADATKA**

---

Osnovi informacionih sistema i softverskog inženjerstva  
2020/21

## Sadržaj

---

1. Uvod	3
2. Model sistema	4
3. Funkcionalnosti sistema	7
4. Perzistencija stanja informacionog sistema	32
5. Formiranje timova	32
6. Raspodela zadataka po studentima	33
7. Upotreba Git-a	35
8. Saradnja studenata	35
9. Konsultacije u vezi sa projektnim zadatkom	36
10. Odbrana projekta	36
11. Domaći zadatak 1.	36
12. Domaći zadatak 2.	37
Odgovori na najčešće postavljena pitanja	38

## I. Uvod

---

Neophodno je modelovati i implementirati informacijski sistem studentske službe Fakulteta tehničkih nauka. Pomenuti sistem implementirati u vidu aplikacije namenjene za upotrebu od strane referenta studentske službe. Aplikaciju implementirati upotrebom Java programskog jezika, pri čemu se grafička korisnička sprega (engl. *interface*) mora razviti upotrebom *Swing* biblioteke. Razvoj aplikacije mora biti u skladu sa MVC šablonom (engl. Model View Controller).

U nastavku specifikacije opisani su neophodni elementi informacionog sistema u vidu modela aplikacije. Nakon toga, detaljno su opisane funkcionalnosti uz prateće grafičke prikaze korisničkog interfejsa.

**Napomena:** Prikazi korisničkog interfejsa (engl. *mockups*) služe da olakšaju razumevanje informacionog sistema studentske službe.

## 2. Model sistema

---

Aplikacija rukuje sledećim entitetima, koji imaju navedene atribute:

### Student (#student)

- Prezime
- Ime
- Datum rođenja
- Adresa stanovanja
- Kontakt telefon
- E-mail adresa
- Broj indeksa
- Godina upisa
- Trenutna godina studija
- Status (tipa enum: "B"- za budžet i "S" – za samofinansiranje)
- Prosečna ocena

- Spisak položenih ispita (spisak ocena)
- Spisak nepoloženih ispita

### **Profesor (#profesor)**

- Prezime
- Ime
- Datum rođenja
- Adresa stanovanja
- Kontakt telefon
- E-mail adresa
- Adresa kancelarije
- Broj lične karte
- Titula
- Zvanje
- Spisak predmeta na kojima je profesor

### **Predmet (#predmet)**

- Šifra predmeta
- Naziv predmeta
- Semestar (letnji ili zimski)
- Godina studija u kojoj se predmet izvodi
- Predmetni profesor
- Broj ESPB bodova
- Spisak studenata koji su položili predmet
- Spisak studenata koji nisu položili predmet

## Ocena na ispitu (#ocena)

- Student koji je položio ispit
- Predmet
- Brojčana vrednost ocene (u intervalu od 6 do 10)
- Datum polaganja ispita

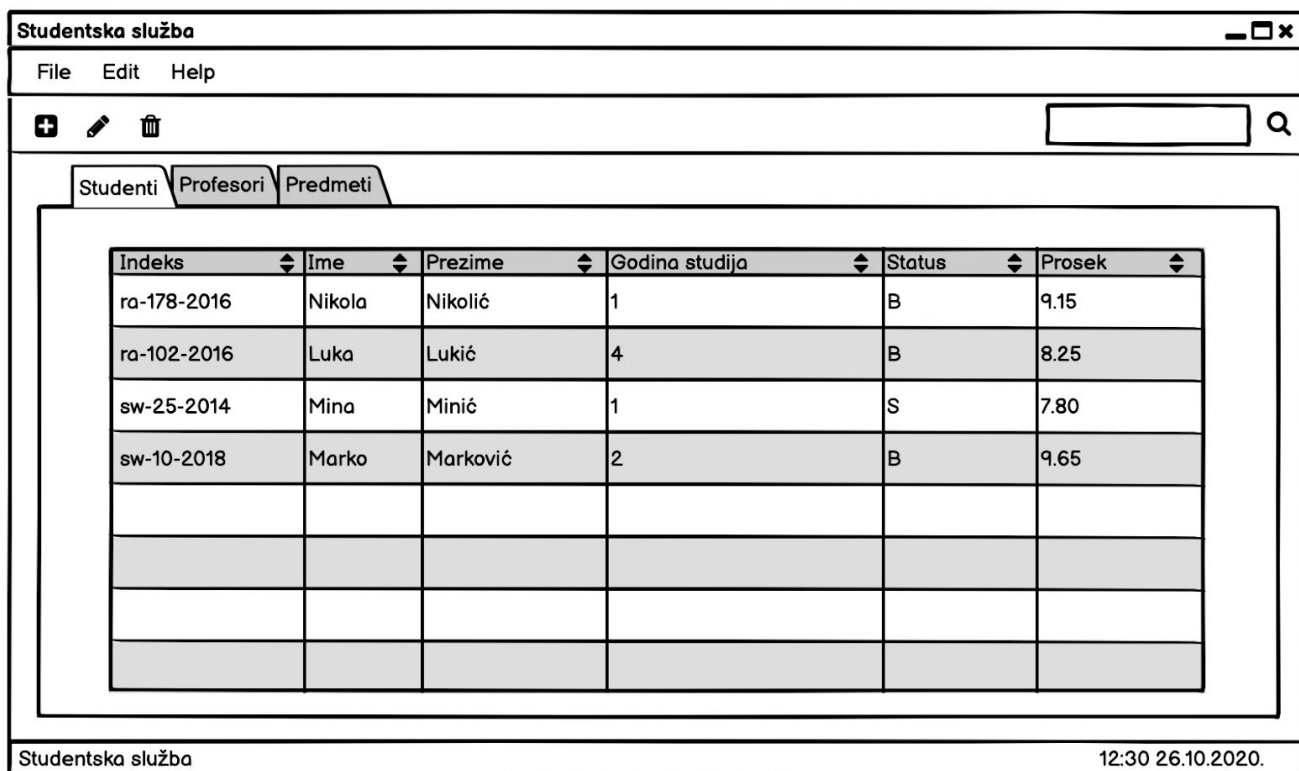
**Napomena: Navedene klase su obavezne. Moguće je proširiti postojeće klase i/ili dodavati nove.**

## 3. Funkcionalnosti sistema

### 3.1 Pokretanje aplikacije (#glavni\_prozor)

Prilikom pokretanja aplikacije, referentu je potrebno prikazati glavni prozor (slika 1). Pomenuti prozor aplikacije inicijalno treba da zauzima  $\frac{3}{4}$  dužine i širine ekrana i da bude centriran u odnosu na ekran. Glavni prozor treba da sadrži:

- Traku sa menijima (engl. *Menu Bar*) aplikacije
- Traku sa alatkama (engl. *Toolbar*) aplikacije
- Centralni panel koji sadrži tabove: *Studenti*, *Profesori* i *Predmeti*. Svaki tab treba da prikaže tabelu sa odgovarajućim entitetima.
- Statusnu traku (engl. *Status Bar*) aplikacije



Slika 1. Izgled početnog prozora

### 3.2 Menu Bar (#menu\_bar)

Na slici 2. prikazan je *Menu Bar* aplikacije koji poseduje 3 menija, a svaki od njih poseduje dolenaedene stavke, odnosno podmenije:

#### 1. File

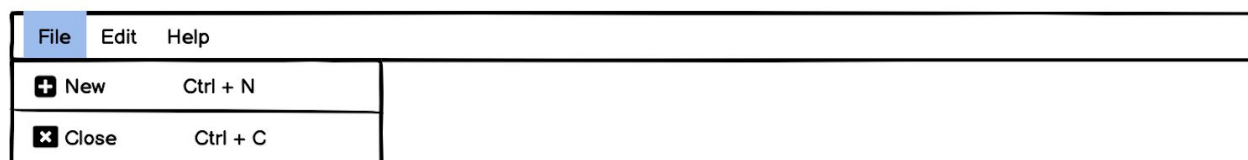
- a) *New* - Dodavanje novog entiteta u sistem.
- b) *Close* - Zatvaranje aplikacije.

#### 2. Edit

- a) *Edit* - Izmena postojećeg entiteta.
- b) *Delete* - Brisanje postojećeg entiteta.

#### 3. Help

- a) *Help* - Ova sekcija treba da sadrži detaljan opis o načinu korišćenja aplikacije. Potrebno je objasniti kako se svaka od dolenaedjenih funkcionalnosti može sprovesti u delo i to u vidu niza korisničkih akcija. Takođe, potrebno je navesti prečice (akceleratori) koje naprednim korisnicima mogu olakšati rad.
- b) *About* - Prikaz verzije aplikacije, kao i kratak opis iste. Nakon toga treba da sledi sažeta biografija svakog autora.

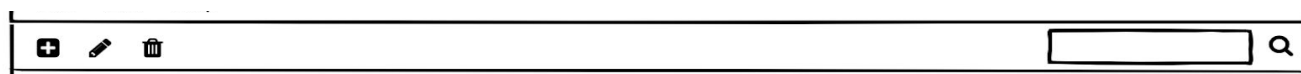


**Slika 2.** Menu Bar aplikacije

**Napomena:** Sve stavke menija treba da imaju odgovarajuću ikonicu, accelerator i mnemonik.

### 3.3 Toolbar (#toolbar)

*Toolbar* aplikacije treba da sadrži 4 dugmeta i jedno polje za unos teksta. Izgled *Toolbar*-a je dat na slici 3.



**Slika 3.** Toolbar aplikacije

Prvo dugme služi za otvaranje dijaloga za kreiranje entiteta informacionog sistema (student, profesor, predmet). Drugo dugme služi za otvaranje dijaloga za izmenu označenog entiteta, dok treće dugme briše označeni entitet iz sistema.

U tekstualno polje za pretragu korisnik može uneti kriterijum pretrage i pritiskom na dugme za pretragu pretražiti tabelu aktivnog taba. U tabeli će biti prikazani entiteti koji zadovoljavaju kriterijum pretrage. Tekstualno polje za pretragu i dugme za pretragu mogu da budu pozicionirani i iznad tabele sa entitetima, tj. nije obavezno da budu sastavni deo toolbar-a.

Toolbar može da se proširi sa ikonicama za ostale funkcionalnosti, po potrebi.

**Napomena:** Sve stavke Toolbar-a moraju imati ikonice, *tooltip*-ove i vezane akceleratora.

### 3.4 Status Bar (#status\_bar)

Status bar aplikacije (primer naveden na slici 4) treba minimalno da podrži:

- 1) trenutni datum
- 2) trenutno vreme
- 3) naziv aplikacije

Studentska služba	12:30 26.10.2020.
-------------------	-------------------

**Slika 4.** Status bar aplikacije

### 3.5 Prikaz entiteta sistema (#prikaz\_studenata, #prikaz\_profesora, #prikaz\_predmeta)

Prilikom izbora taba *studenti*, u centralnom delu panela taba potrebno je prikazati tabelu sa svim studentima (slika 1). Tabela treba da prikazuje sledeće podatke studenata:

- Indeks
- Ime
- Prezime
- Godina studija
- Status
- Prosek

Prilikom izbora taba *profesori*, prikazuje se tabela sa sledećim informacijama o profesorima:

- Ime
- Prezime
- Titula
- Zvanje



Izborom taba *predmeti*, referentu je potrebno prikazati tabelu sa sledećim informacijama predmeta:

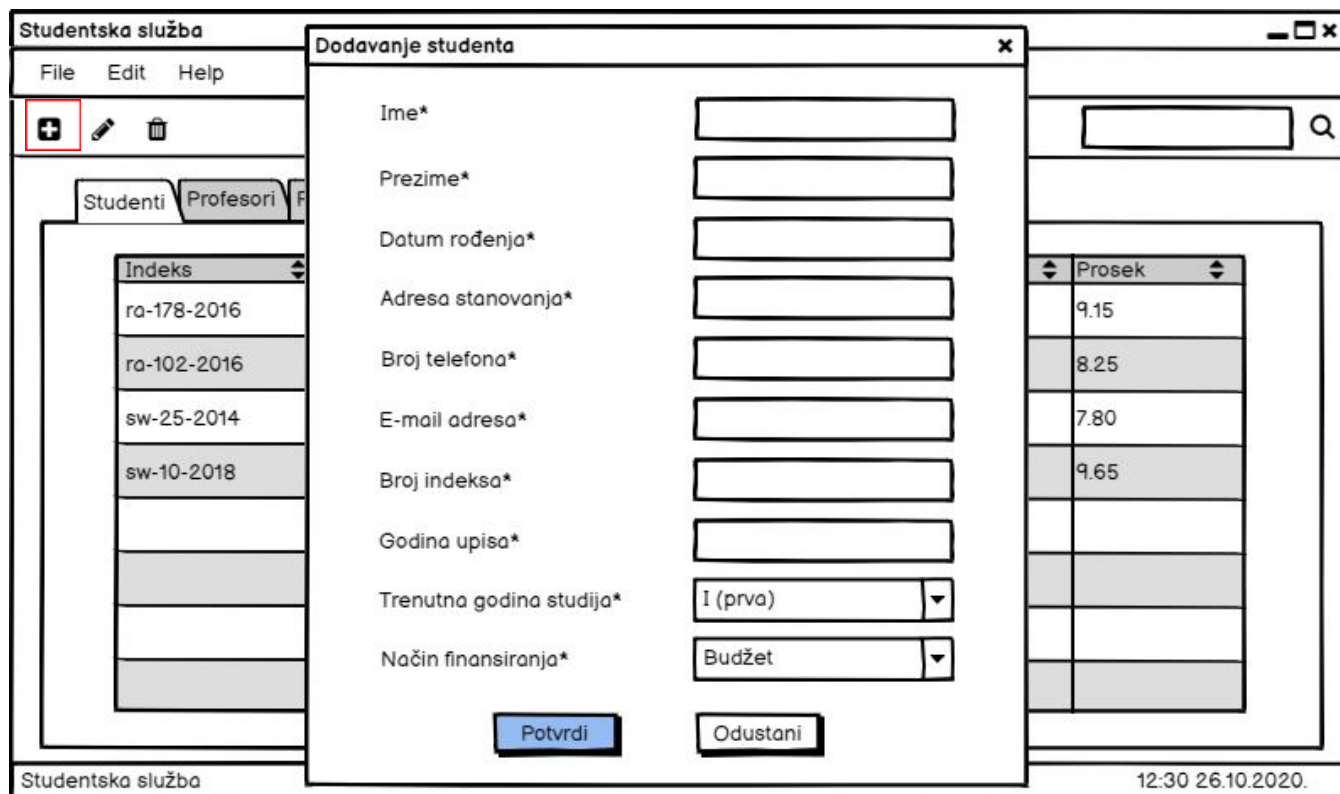
- Šifra predmeta
- Naziv predmeta
- Broj ESPB bodova
- Godina na kojoj se predmet izvodi
- Semestar u kome se predmet izvodi

### 3.6 Dodavanje novog entiteta (#dodavanje\_studenta, #dodavanje\_profesora, #dodavanje\_predmeta)

Da bi referent registrovao novog studenta u sistem, neophodno je da se na glavnom prozoru odabere tab *Studenti* (slika 1). Nakon toga, potrebno je da referent odabere jedan način za dodavanje novog entiteta (od 4 moguća načina):

1. Pritisak prvog dugmeta u *ToolBar*-u
2. Izbor stavke menija *New*
3. Upotrebom odgovarajućih mnemonika
4. Upotrebom odgovarajućeg akceleratora.

Nakon toga, referentu se prikazuje dijalog za dodavanje studenta (slika 5). Dijalog treba da bude modalan i centriran u odnosu na glavni prozor koji ga poziva. Treba da sadrži formu za unos informacija o studentu, kao i da ima dva dugmeta, koja služe za potvrdu („*Potvrdi*“) unosa, odnosno odustanak od unosa novog studenta („*Odustani*“). Sva polja moraju biti validno popunjena, u suprotnom onemogućiti pritisak dugmeta „*Potvrdi*“. Nakon pritiska pomenutog dugmeta, neophodno je ažurirati sadržaj tabele sa studentima.



**Slika 5.** Dodavanje studenta

Po sličnom principu referentu omogućiti registrovanje novog profesora, odnosno novog predmeta u sistem.

**Napomena:** Podrazumeva se i da se ostale funkcionalnosti mogu izvršiti upotrebom odgovarajućeg dugmeta toolbar-a, stavke menija, kombinacije mnemonika, odnosno akceleratora. Radi sažetosti zapisa, biće izostavljeni iz dalje specifikacije.

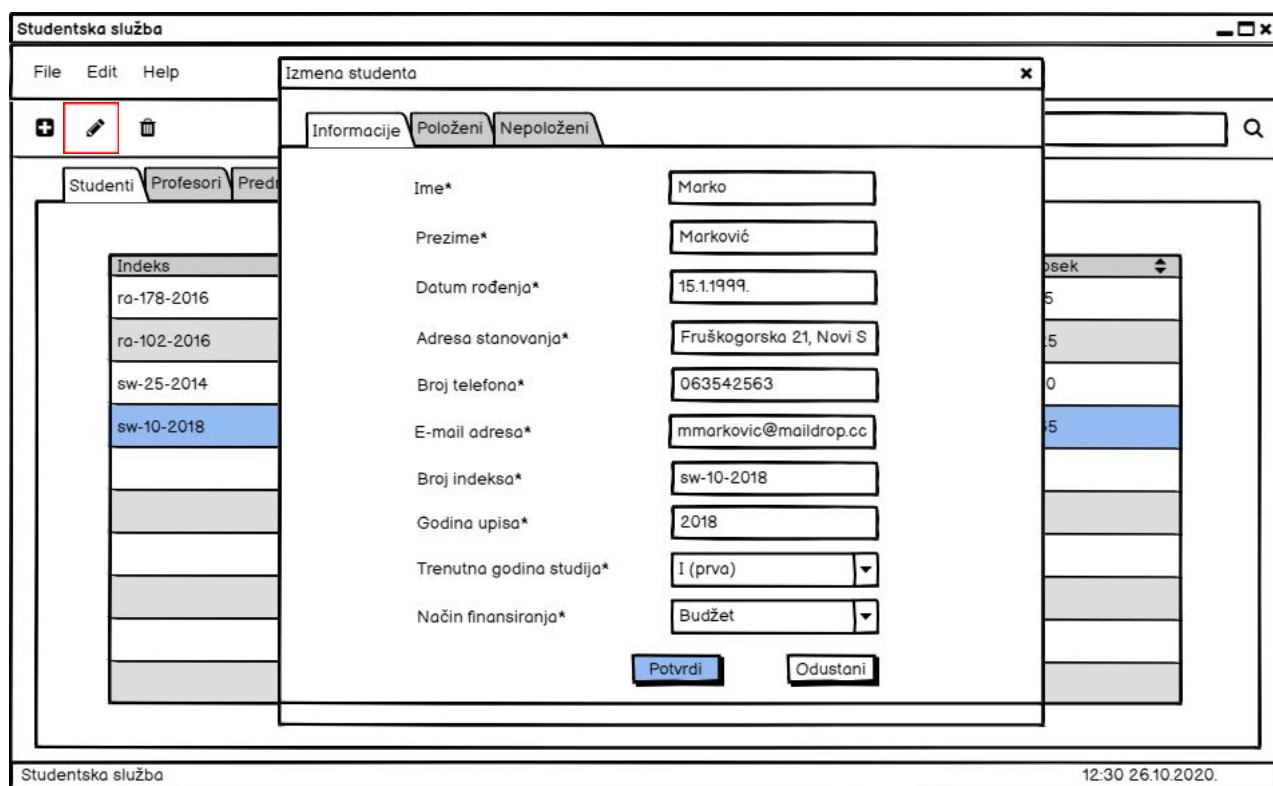
### 3.7 Izmena informacija o entitetu (#izmena\_studenta, #izmena\_profesora, #izmena\_predmeta)

Da bi referent izmenio informacije o studentu, neophodno je da izabere tab *Studenti*. Nakon toga, potrebno je da označi odgovarajući red tabele, koji sadrži sažetak informacija o konkretnom studentu. Zatim, potrebno je da izabere funkcionalnost izmene entiteta, pri čemu se otvara dijalog prikazan na slici 6. Dijalog treba da bude centriran u odnosu na glavni prozor i treba da sadrži sledeće tabove:

- *Informacije* - sa detaljnim podacima o studentu.
- *Položeni* - sa spiskom položenih predmeta.
- *Nepoloženi* - sa spiskom nepoloženih predmeta.

Tab *Informacije* treba da sadrži panel na kome je prikazana forma za izmenu podataka o studentu. Polja forme treba da budu popunjena odgovarajućim podacima. Na dnu forme nalaze se dugmad za potvrdu izmena (*Potvrdi*), odnosno odustanak od izmene (*Odustani*). Dugme *Potvrdi* može biti aktivno ukoliko su svi izmenjeni podaci u formi validni. Nakon uspešne izmene podataka, ažurirati sadržaj tabele sa studentima.

Po sličnom principu implementirati izmenu informacija o profesorima, odnosno predmetima.



Slika 6. Izmena selektovanog studenta

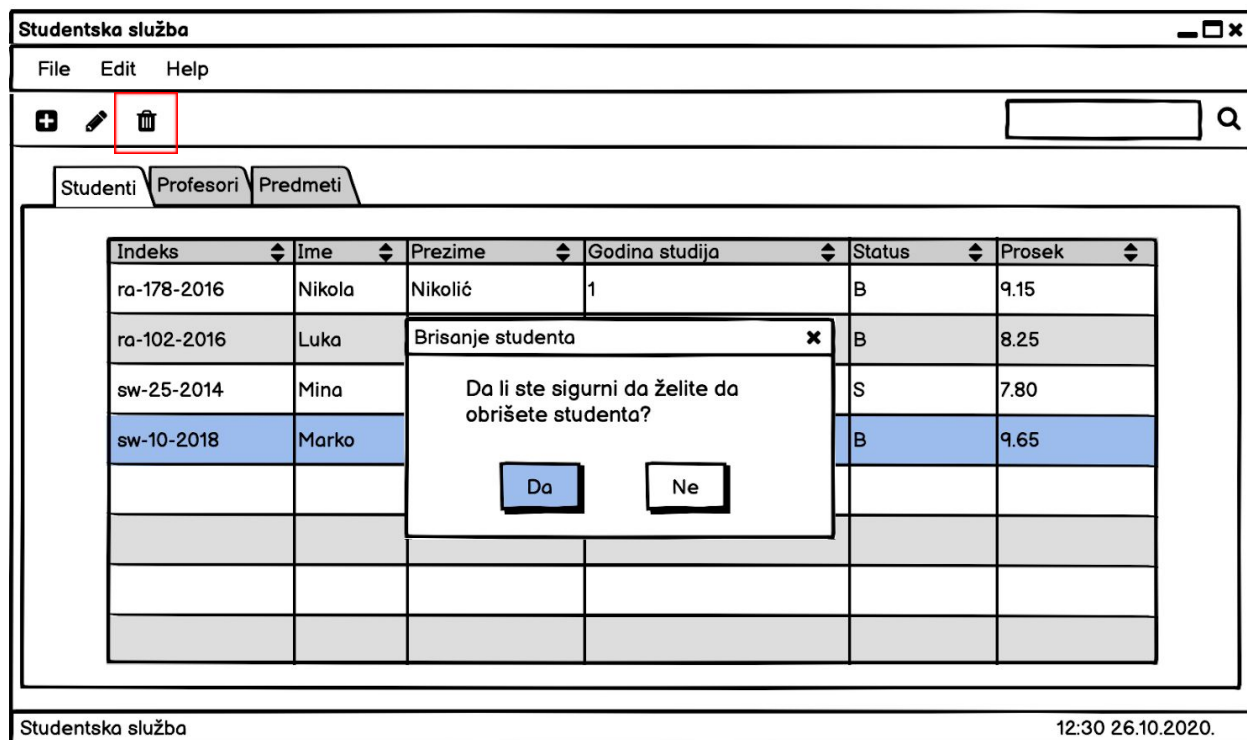
Tabovi Položeni i Nepoloženi biće detaljno objašnjeni posle funkcionalnosti za brisanje entiteta.

### 3.7 Brisanje entiteta iz sistema (#brisanje\_studenta, #brisanje\_profesora, #brisanje\_predmeta)

Da bi referent bio u mogućnosti da ukloni studenta iz sistema, neophodno je da na početnom prozoru odabere tab *Studenti*. Nakon toga, označiti red tabele kojim se reprezentuje student koga treba ukloniti iz sistema. Zatim, referent mora odabrati funkcionalnost brisanja (dugme u toolbaru, stavka menija, ...). Kao rezultat izabrane funkcionalnosti, korisniku se prikazuje dijalog sa porukom (engl. *Message Box*) prikazan na slici 7. Ukoliko se potvrdi brisanje, potrebno je ukloniti označenog studenta iz tabele.

Po sličnom principu, neophodno je implementirati brisanje profesora, odnosno predmeta iz sistema.

**Napomena:** Voditi računa o referencijalnim zavisnostima.



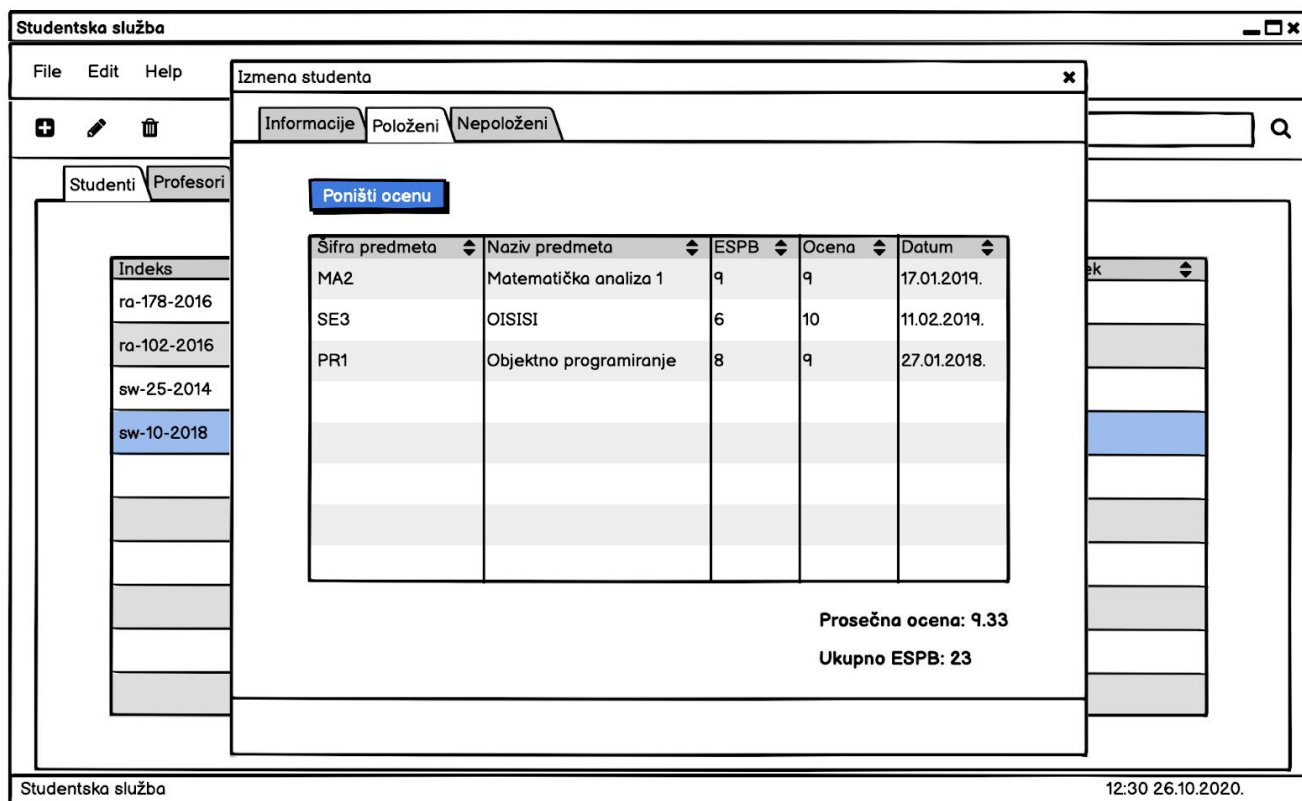
**Slika 7.** Dijalog za brisanje označenog studenta

### 3.8 Prikaz položenih ispita (#prikaz\_polozenih\_ispita)

Prilikom rada sa dijalogom za izmenu podataka o studentu (slika 6), referent može da odabere tab *Položeni*, nakon čega se u panelu taba prikazuje tabela sa položenim ispitima (slika 8).

Iznad table nalazi se dugme “Poništi ocenu”.

U donjem desnom uglu panela nalaze se informacije o prosečnoj oceni, odnosno o ukupnom broju osvojenih ESPB bodova u toku studija.

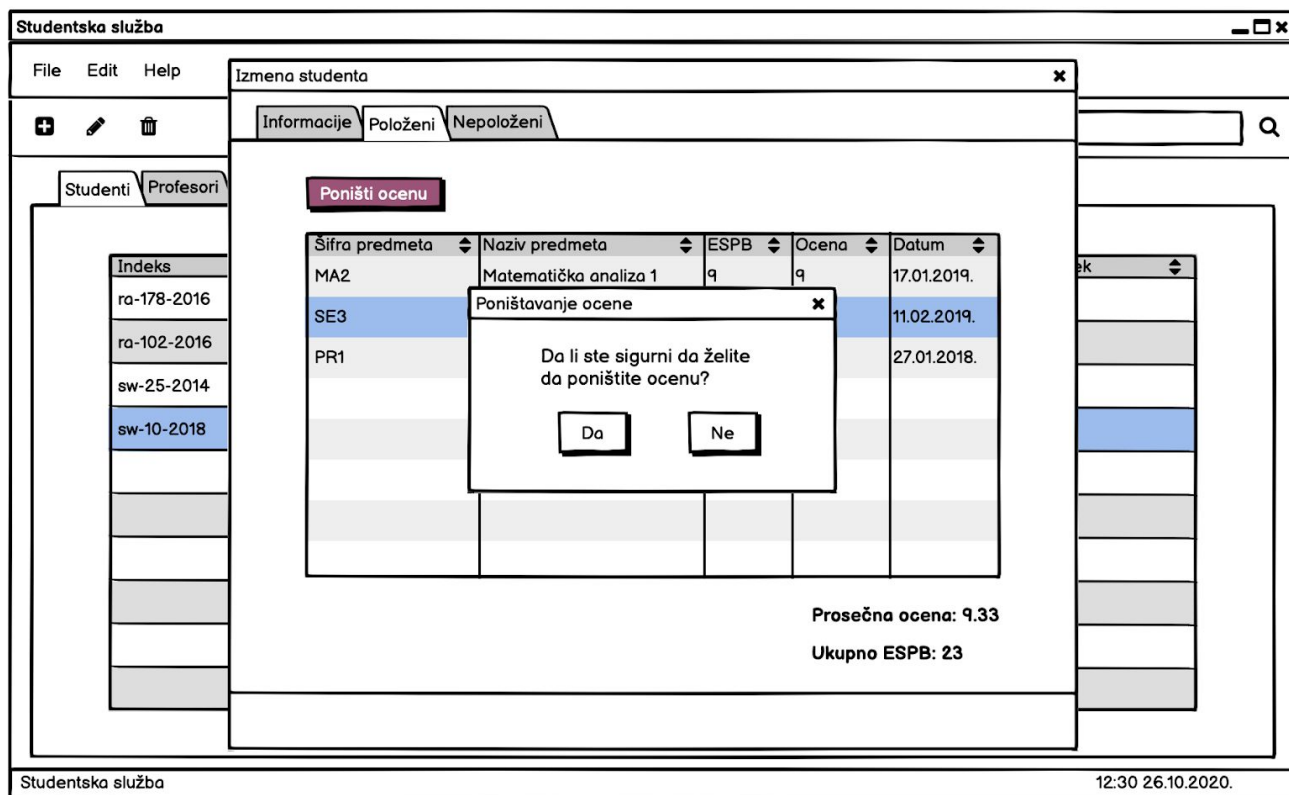


**Slika 8.** Tabela sa položenim predmetima

### 3.9 Poništavanje ocene (#ponistavanje\_ocene)

Ako se predmet nalazi u tabeli sa položenim predmetima (slika 8), njemu je dodeljena odgovarajuća ocena i moguće je poništiti. Da bi se ocena poništila, potrebno je označiti predmet i kliknuti na dugme *Poništi ocenu*. Prikazuje se novi dijalog sa porukom (slika 9) koji treba da bude modalan, centriran u odnosu na dijalog koji ga je pozvao i da sadrži dva dugmeta za potvrdu poništavanja ocene ("Da") i odustanak ("Ne").

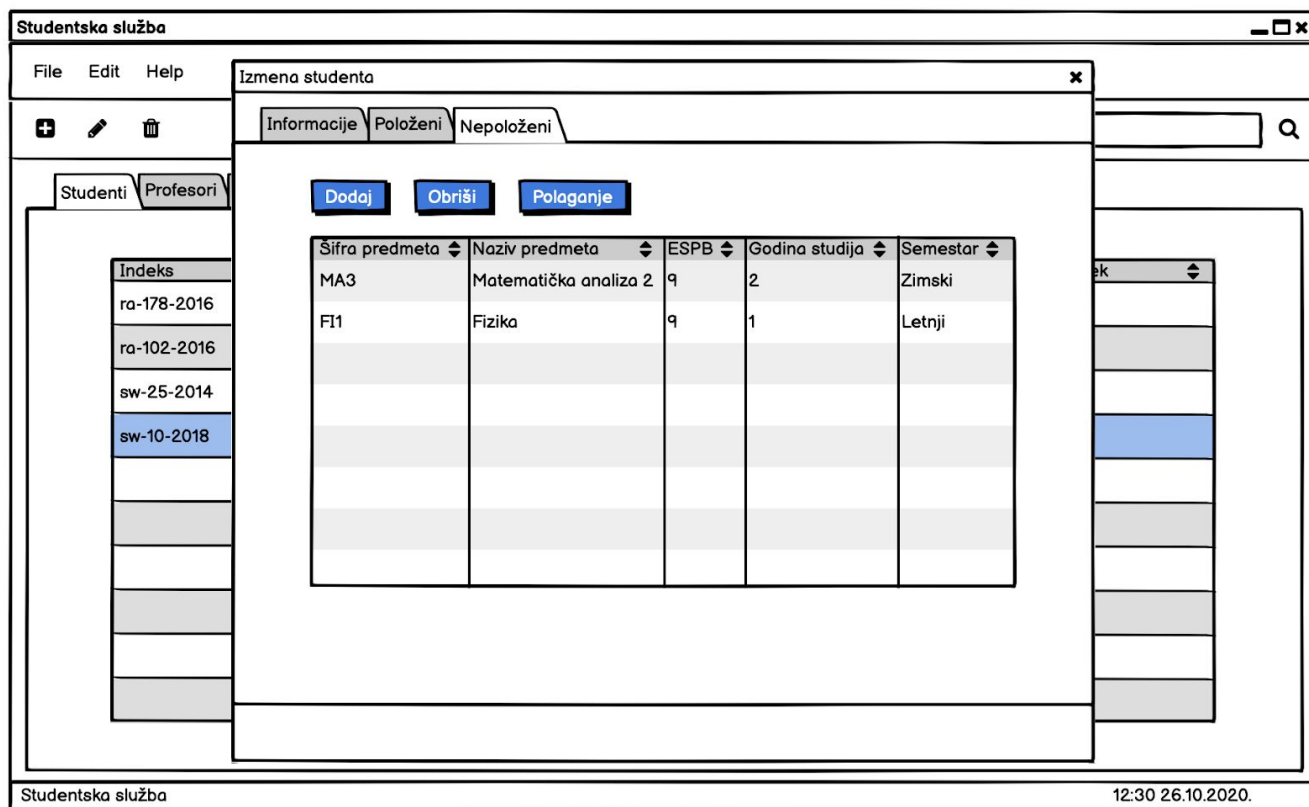
Kada se uspešno poništi ocena, predmet se iz tabele položenih predmeta prebacuje u tabelu nepoloženih predmeta. Nakon zatvaranja dijaloga za potvrdu poništavanja ocene, predmet čija je ocena poništena, više ne treba da bude vidljiv u tabeli položenih predmeta.



Slika 9. Poništavanje ocene označenog predmeta

### 3.10 Prikaz nepoloženih predmeta (#prikaz\_nepolozenih\_ispita - ispravka 04.01.2021.)

Prilikom rada sa dijalogom za izmenu podataka o studentu (slika 6), referent može da odabere tab *Nepoloženi*, nakon čega se prikazuje tabela sa svim nepoloženim predmetima, koje student sluša. Iznad tabele nalaze se 3 dugmeta: *Dodaj*, *Obriši* i *Polaganje* (slika 10).

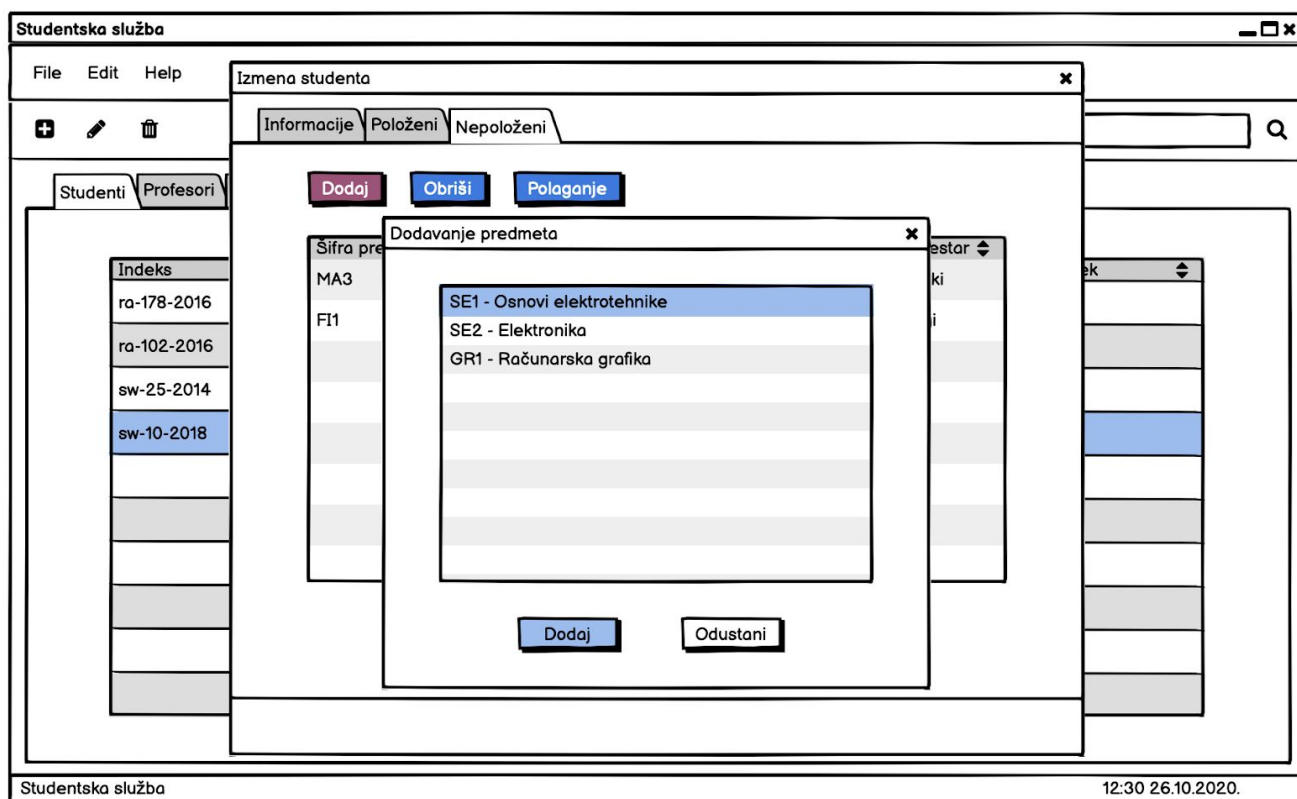


**Slika 10.** Tabela sa nepoloženim predmetima

### 3.1.1 Dodavanje studenta na predmet (#dodavanje\_predmeta\_studentu - ispravka 30.12.2020)

Da bi se dodao novi student na predmet, potrebno je pozicionirati se na tab *Nepoloženi* (prikazan na slici 10) i kliknuti dugme *Dodaj*. Nakon toga, referentu se prikazuje dijalog sa listom predmeta koje student može da sluša (slika 11). Predmeti koji se prikazuju u listi moraju da zadovolje minimum 2 uslova. Prvi uslov je da se ne nalaze ni u listi položenih, kao ni u listi nepoloženih predmeta označenog studenta. Drugi uslov je da je označeni student na odgovarajućoj godini studija (ista ili viša godina studija u odnosu na godinu na kojoj se predmet izvodi).

Izborom odgovarajućeg predmeta i klikom na dugme *Dodaj*, dodaje se novi predmet u tabelu nepoloženih predmeta, koju je ovom prilikom potrebno ažurirati.

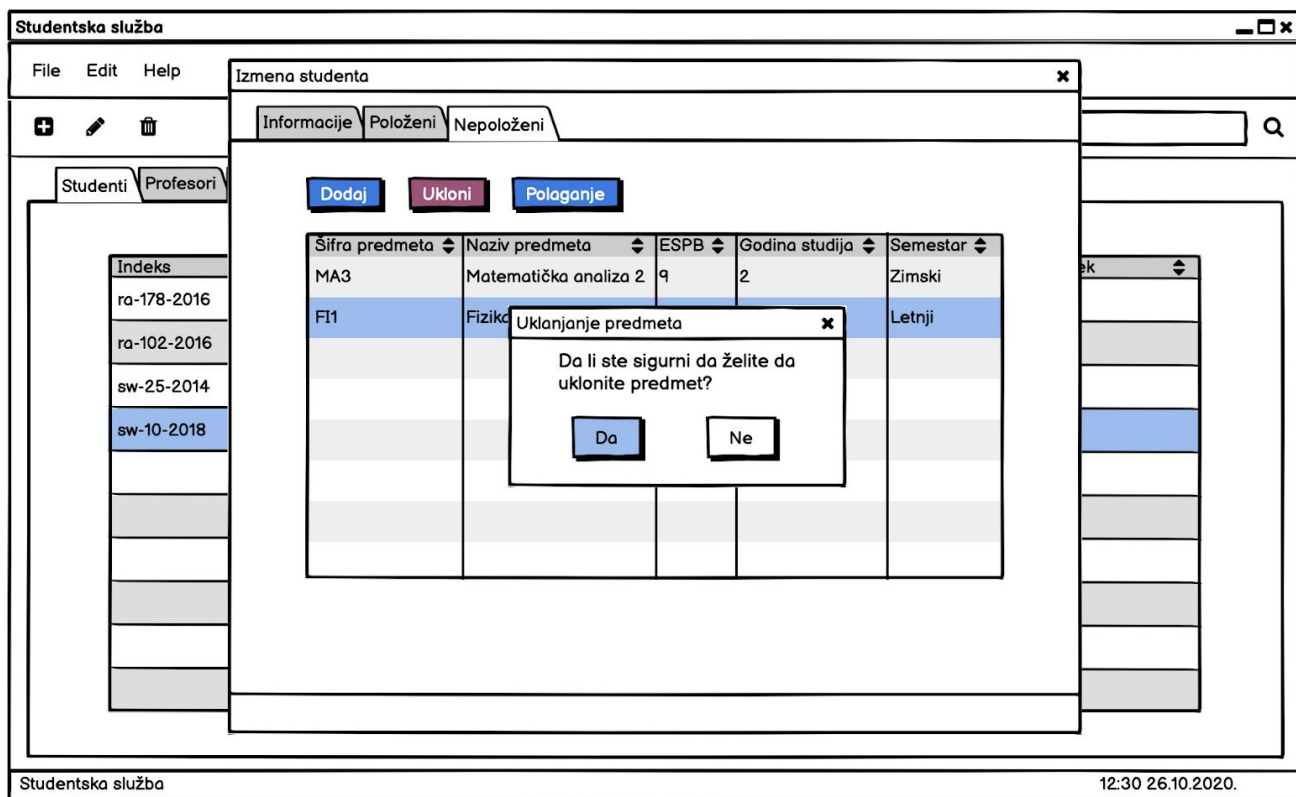


Slika 11. Dodavanje studenta na novi predmet



### 3.12 Uklanjanje studenta sa predmeta (#uklanjanje\_predmeta\_sa\_studenta)

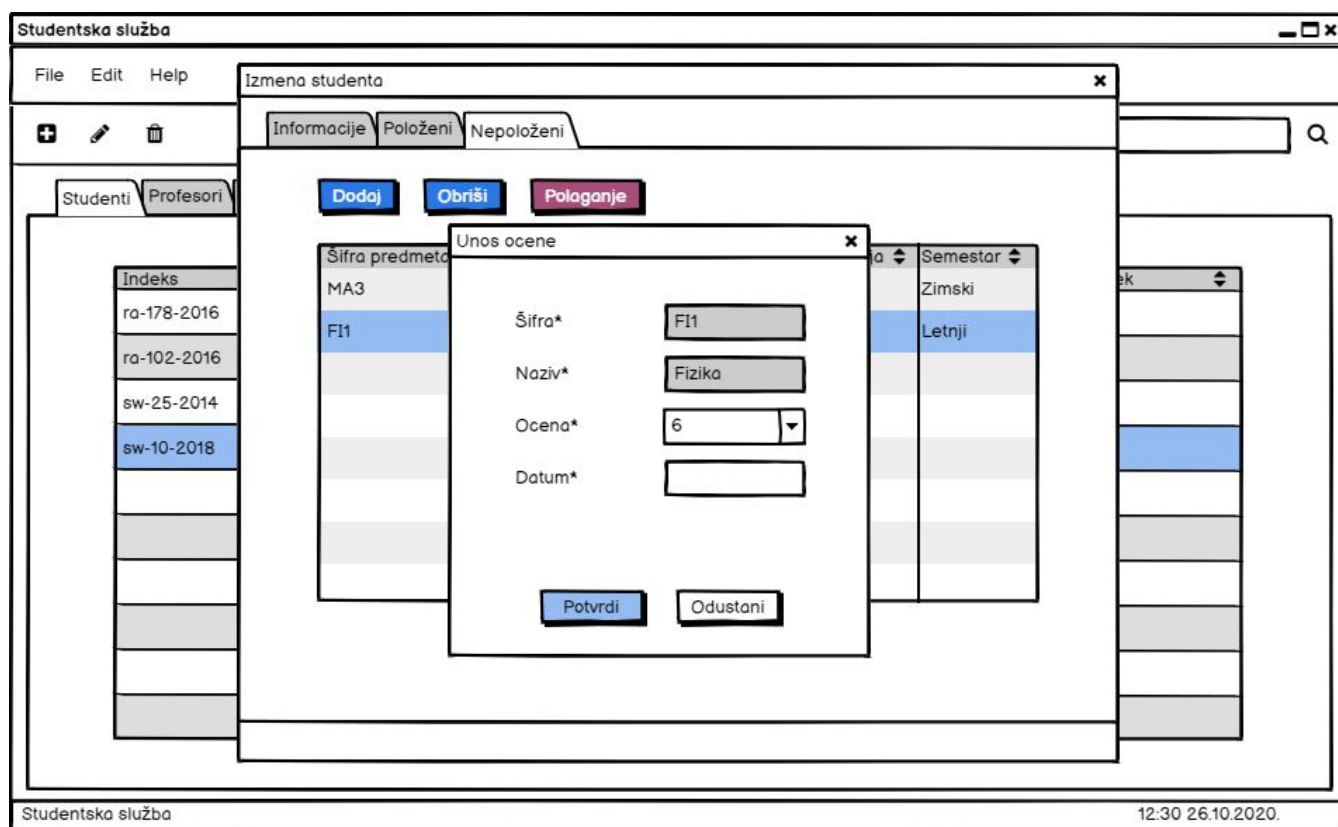
U okviru taba *Nepoloženi* moguće je ukloniti predmete, koje označeni student sluša. Nakon što se označi odgovarajući predmet i klikne na dugme *Ukloni*, prikazuje se dijalog gde je potrebno potvrditi operaciju uklanjanja predmeta (slika 12). Nakon potvrde, izabrani predmet je neophodno ukloniti iz tabele nepoloženih predmeta.



Slika 12. Uklanjanje studenta sa predmeta

### 3.13 Upis ocene (#upis\_ocene)

Tokom pregleda nepoloženih ispita (slika 10), referentu je potrebno ponuditi upis ocene na neki od nepoloženih predmeta. Da bi se pomenuta funkcionalnost sprovela, neophodno je označiti neki od predmeta iz tabele nepoloženih predmeta, nakon čega se odabirom dugmeta “*Polaganje*” kao rezultat prikazuje dijalog centriran u odnosu na dijalog sa spiskom nepoloženih predmeta. U novootvorenom dijalogu neophodno je da se nalazi forma prikazana na slici 13. U pomenutoj formi potrebno je programski upisati šifru i naziv odabranog predmeta. Pomenute podatke nije moguće izmeniti. Takođe, referentu omogućiti unos brojčane vrednosti ocene u intervalu od 6 do 10, kao i datuma polaganja ispita u odgovarajuća polja. Na samom dnu dijaloga prikazati dugmad za potvrdu, odnosno odustanak upisa ocene. Ukoliko su sva polja validno popunjena, omogućiti pritisak dugmeta *Potvrda*. Kao rezultat ove akcije referenta, neophodno je izabrani predmet izbrisati iz tabele nepoloženih predmeta i prebaciti ga u tabelu položenih predmeta.



Slika 13. Dijalog za unos ocene

### 3.14 Pretraga entiteta (#pretraga\_studenata, #pretraga\_profesora, #pretraga\_predmeta)

Potrebno je omogućiti tekstualnu pretragu svih entiteta informacionog sistema studentske službe. Prilikom izbora taba *Studenti* glavnog prozora (slika 1), referent ima mogućnost da pretraži sve studente registrovane u studentskoj službi. Kriterijum tekstualne pretrage, tj. upit, unosi se u tekstualno polje prikazano unutar *Toolbar*-a. Nakon pritiska dugmeta koje se nalazi odmah uz pomenuto polje, tabela sa studentima se ažurira tako da prikaže studente koji zadovoljavaju uneti upit.

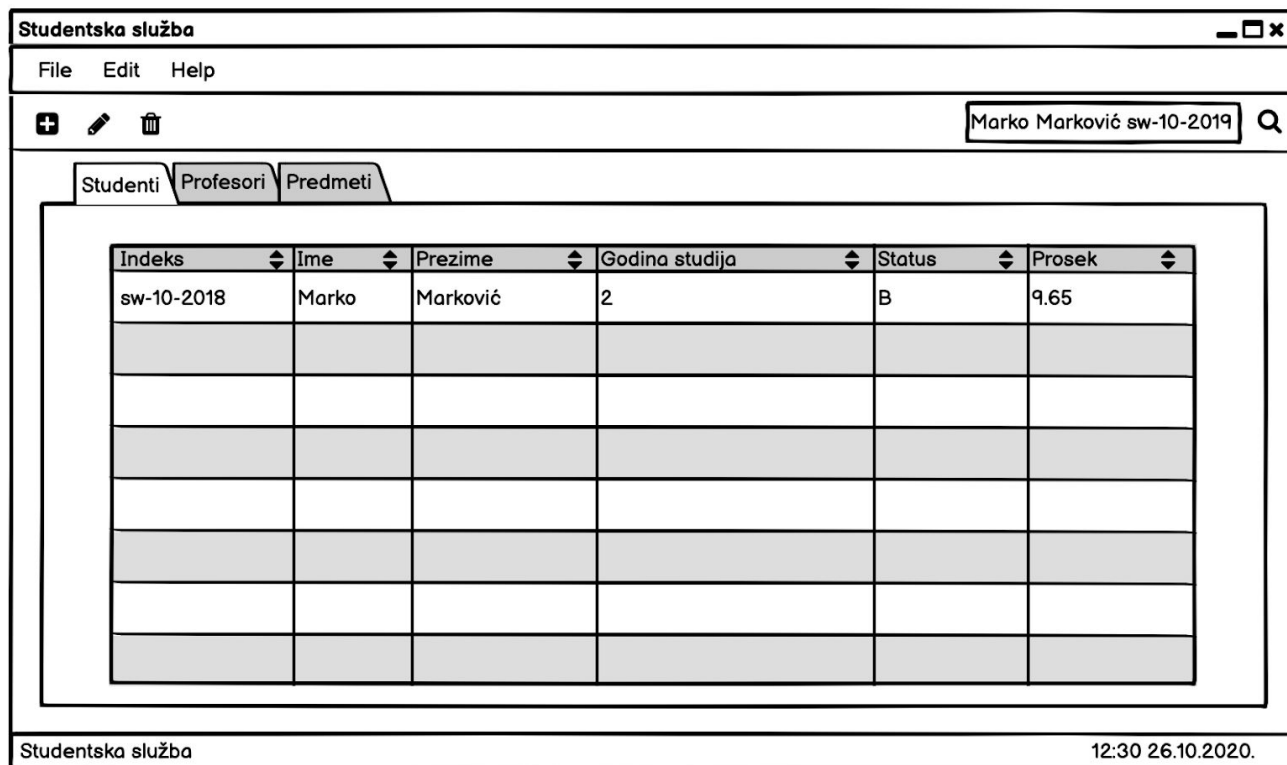
Prilikom pretrage studenata, u polje za pretragu unose se upiti koji se mogu sastojati od jedne, dve ili tri reči. Prva reč je obavezna i odnosi se na deo prezimena studenta. Druga, odnosno treća reč su opcione i odnose se na deo imena, odnosno broja indeksa studenta. Nakon unosa upita, aplikacija treba da isparsira upit, te izdvoji reči upita. Na osnovu parsiranih reči, potrebno je proveriti koji od registrovanih studenata zadovoljava kriterijum pretrage.

Ukoliko je uneta samo jedna reč, tada treba izdvojiti studente čija prezimena sadrže unetu reč. Ukoliko su unete dve reči, tada je potrebno proveriti da li se prva reč sadrži u prezimenu studenta, pri čemu se druga reč mora sadržati u imenu studenta. Ukoliko su unete sve tri reči, tada kriterijum pretrage zadovoljavaju studenti čije prezime sadrži prvu reč, pri čemu ime, odnosno broj indeksa moraju sadržati drugu, odnosno treću reč.

Primeri validnih upita su:

- Marković Marko sw-1-2019
- MARK MARK SW-1-
- Markov
- MaRkov
- vić mark sw-1

U slučaju da su studenti navedeni na slici 1. registrovani u sistemu studentske službe, tada studenti prikazani na slici 14. zadovoljavaju navedene kriterijume pretrage.



**Slika 14.** Primer pretrage studenata

Na sličan način potrebno je implementirati pretragu profesora, odnosno predmeta registrovanih u sistemu studentske službe. Pretragu profesora je moguće obaviti unosom dela prezimena, odnosno imena, dok je pretragu predmeta moguće obaviti unosom dela naziva predmeta.

### 3.1.5 Sortiranje entiteta (#sortiranje\_studenata, #sortiranje\_profesora, #sortiranje\_predmeta)

Referentu studentske službe potrebno je omogućiti sortiranje entiteta sistema po predefinisanim kriterijuma. Ukoliko referent odabere tab *Studenti* glavnog prozora, tada je potrebno omogućiti sortiranje svih studenata po vrednostima sledećih atributa:

- a) broj indeksa (rastuće/opadajuće)
- b) ime (rastuće/opadajuće)
- c) prezime (rastuće/opadajuće)
- d) godina studija (rastuće/opadajuće)
- e) status (rastuće/opadajuće)
- f) prosečna ocena (rastuće/opadajuće)

Obezbediti sortiranje i u rastućem i u opadajućem redosledu. Na slici 15 prikazani su studenti sortirani po imenu. Izbor kriterijuma za sortiranje potrebno je obaviti pritiskom na strelice prikazane u zaglavlju odgovarajuće kolone tabele.

Po sličnom principu neophodno je obezbediti sortiranje profesora, odnosno predmeta. Implementirati sortiranje profesora po vrednostima sledećih atributa:

- a) ime (rastuće/opadajuće)
- b) prezime (rastuće/opadajuće)
- c) titula (rastuće/opadajuće)
- d) zvanje (rastuće/opadajuće)

Implementirati sortiranje predmeta po vrednostima sledećih atributa:

- a) šifra predmeta (rastuće/opadajuće)
- b) naziv predmeta (rastuće/opadajuće)
- c) broj ESPB bodova (rastuće/opadajuće)
- d) godina na kojoj se predmet izvodi (rastuće/opadajuće)
- e) semestar u kome se predmet izvodi (rastuće/opadajuće)

Studentska služba						
File Edit Help						
<div> <div>+</div> <div></div> <div></div> </div> <div></div> <div>Q</div>						
<div> <div>Studenti</div> <div>Profesori</div> <div>Predmeti</div> </div>						
Indeks	Ime	Prezime	Godina studija	Status	Prosek	
ra-102-2016	Luka	Lukić	4	B	8.25	
sw-10-2018	Marko	Marković	2	B	9.65	
sw-25-2014	Mina	Minić	1	S	7.80	
ra-178-2016	Nikola	Nikolić	1	B	9.15	

Studentska služba

12:30 26.10.2020.

Slika 15. Sortiranje studenata po imenu

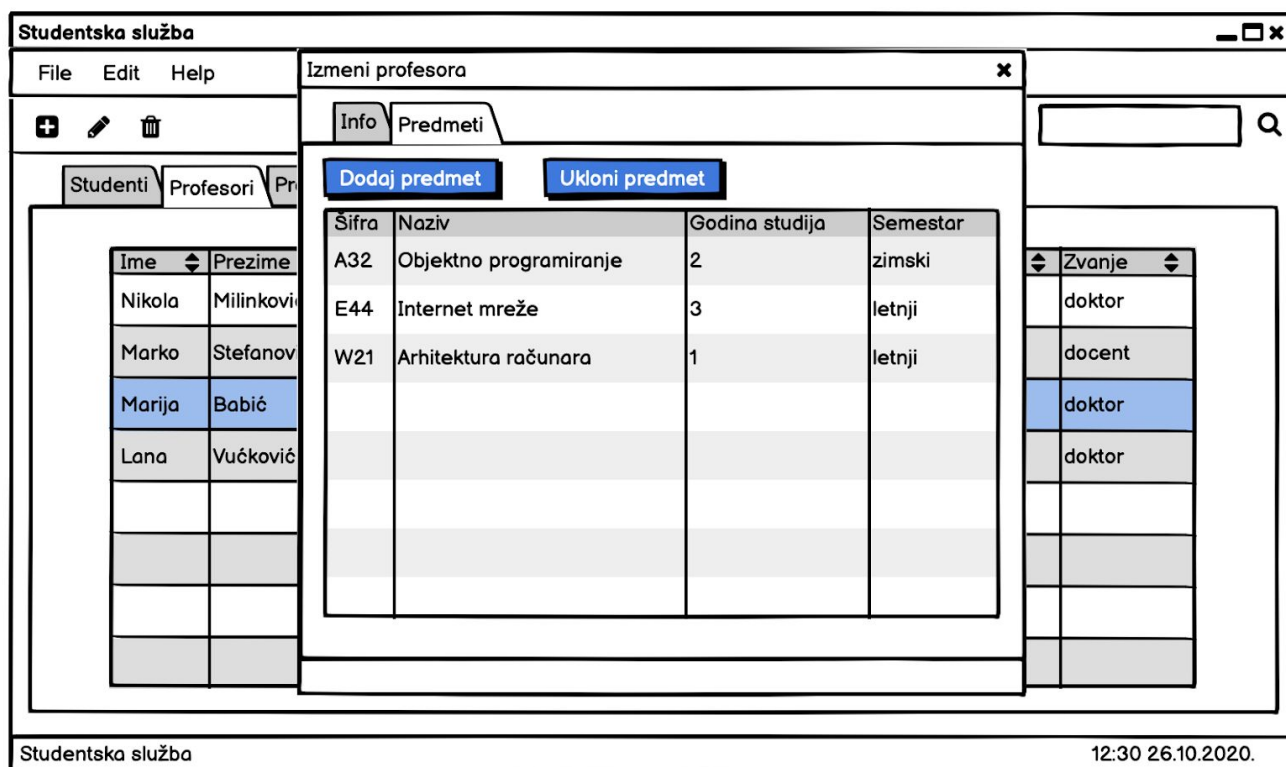
**Napomena:** Prilikom sortiranja voditi računa da li se primenjuje leksikografsko, odnosno brojčano sortiranje.

### 3.16 Prikaz predmeta koje profesor predaje (#profesor\_predaje\_predmete)

Ukoliko referent izabere tab *Profesori* glavnog prozora, tada mu se prikazuje tabela sa svim profesorima registrovanim u sistemu. Selekcijom odgovarajućeg reda i izborom funkcionalnosti za izmenu entiteta (detaljan opis dostupan u sekciji 3.7), prikazuje se dijalog *Izmena profesora*, koji treba da bude centriran u odnosu na glavni prozor. Pomenuti dijalog poseduje dva taba: *Informacije* i *Predmeti*. Prvi tab sadrži formu za izmenu podataka o profesoru (implementirati po uzoru na izmenu podataka o studentu, objašnjenu u sekciji 3.7). Drugi tab (slika 16) prikazuje tabelu sa spiskom predmeta koje profesor predaje. U tabeli prikazati sledeće podatke o predmetima:

- Šifra predmeta
- Naziv predmeta
- Godina studija na kojoj se predmet izvodi
- Semestar u kojem se predmet izvodi.

Iznad table potrebno je prikazati dva dugmeta: *Dodaj predmet* i *Ukloni predmet*, čija će upotreba biti objašnjena u nastavku.



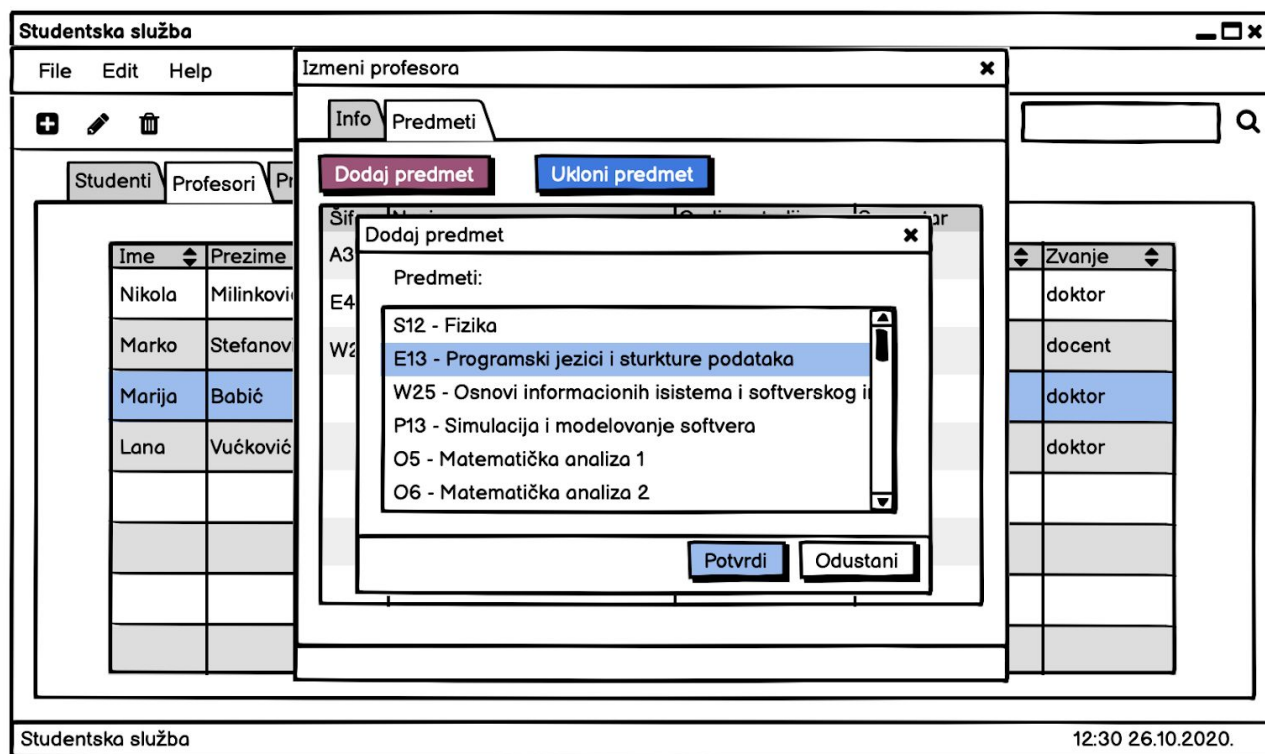
**Slika 16.** Tabela sa listom predmeta na kojima označeni profesor predaje

### 3.17 Dodavanje predmeta profesoru (#dodavanje\_predmeta\_profesoru - ispravka 04.01.2021)

Tokom rada sa dijalogom Izmena profesora (prikazanog na slici 16), referent ima mogućnost da doda profesora na predmet koji profesor u datom trenutku ne predaje. Potrebno je izabrati dugme *Dodaj predmet* (slika 16), nakon čega se prikazuje dijalog Dodaj predmet, koji je centriran u odnosu na dijalog Izmena profesora. Novootvoreni dijalog sadrži listu predmeta, koje profesor u datom trenutku ne predaje. Ispod liste nalaze se dva dugmeta za potvrdu dodavanja profesora na predmet (*Potvrdi*), odnosno odustanak od pomenute akcije (*Odustani*).

Da bi referent dodao profesora na predmet, neophodno je da označi predmet iz liste, nakon čega pritiska dugme *Potvrdi* (slika 17). Nakon uspešno odrađene akcije, dijalog *Dodaj predmet* se zatvara, pri čemu se tabela taba *Predmeti* dijaloga *Izmeni profesora* ažurira tako da prikaže novi predmet koji profesor predaje.

**Napomena:** Dodavanje jednog predmeta je neophodno. Moguće je obezbediti i dodavanje profesora na više predmeta odjednom (višestruka selekcija).



Slika 17. Dodavanje predmeta profesoru

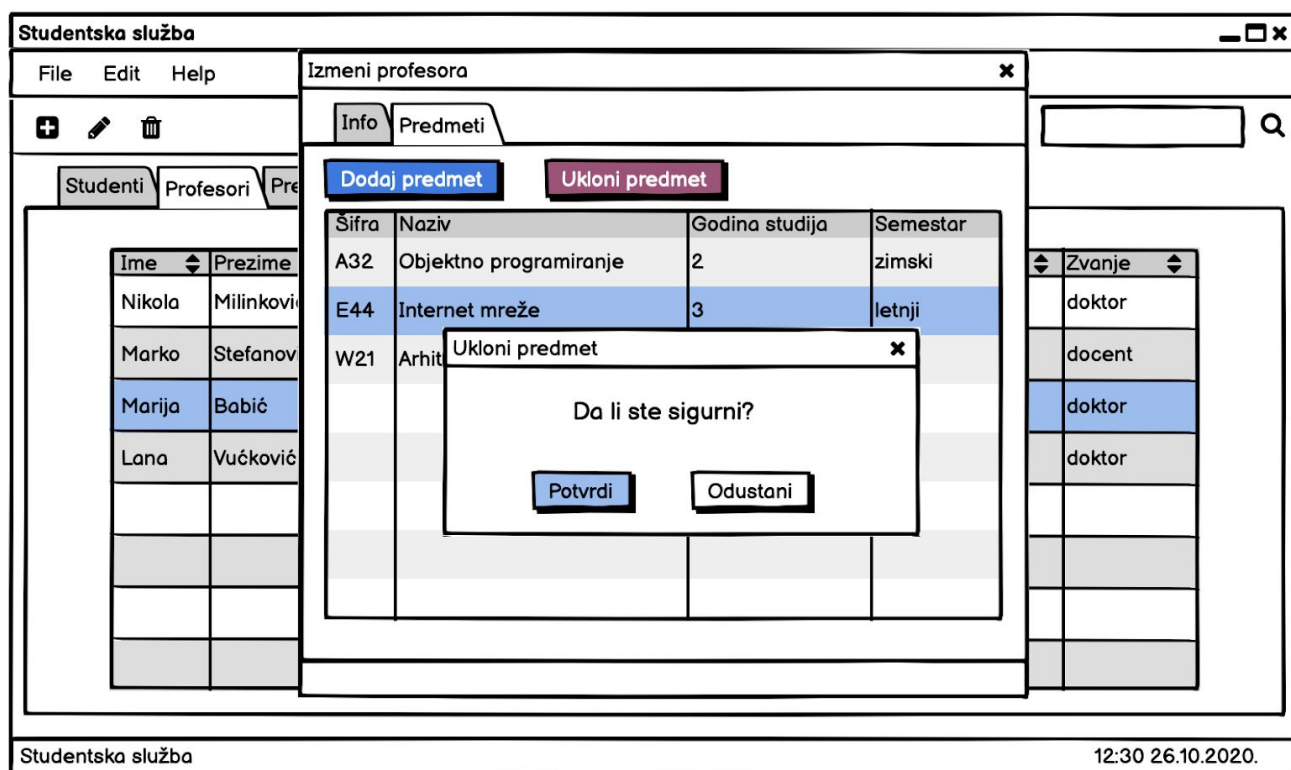


### 3.18 Uklanjanje predmeta iz liste predmeta, koje profesor predaje

(#uklanjanje\_predmeta\_sa\_profesora)

Tokom interakcije sa tabom *Profesori* dijaloga *Izmeni profesora*, referent ima mogućnost da ukloni profesora sa predmeta. Da bi to uradio, neophodno je da označi red iz tabele, koji reprezentuje predmet koji profesor predaje u datom trenutku, nakon čega bira dugme Ukloni predmet. Nakog toga, prikazuje se dijalog sa porukom o potvrdi izabrane akcije (slika 18). Kada referent potvrdi da želi da ukloni označeni predmet, predmet se uklanja iz tabele predmeta koje profesor predaje.

**Napomena:** Uklanjanje profesora sa jednog predmeta je neophodno. Po želji, omogućiti uklanjanje profesora sa nekoliko predmeta odjednom (višestruka selekcija).



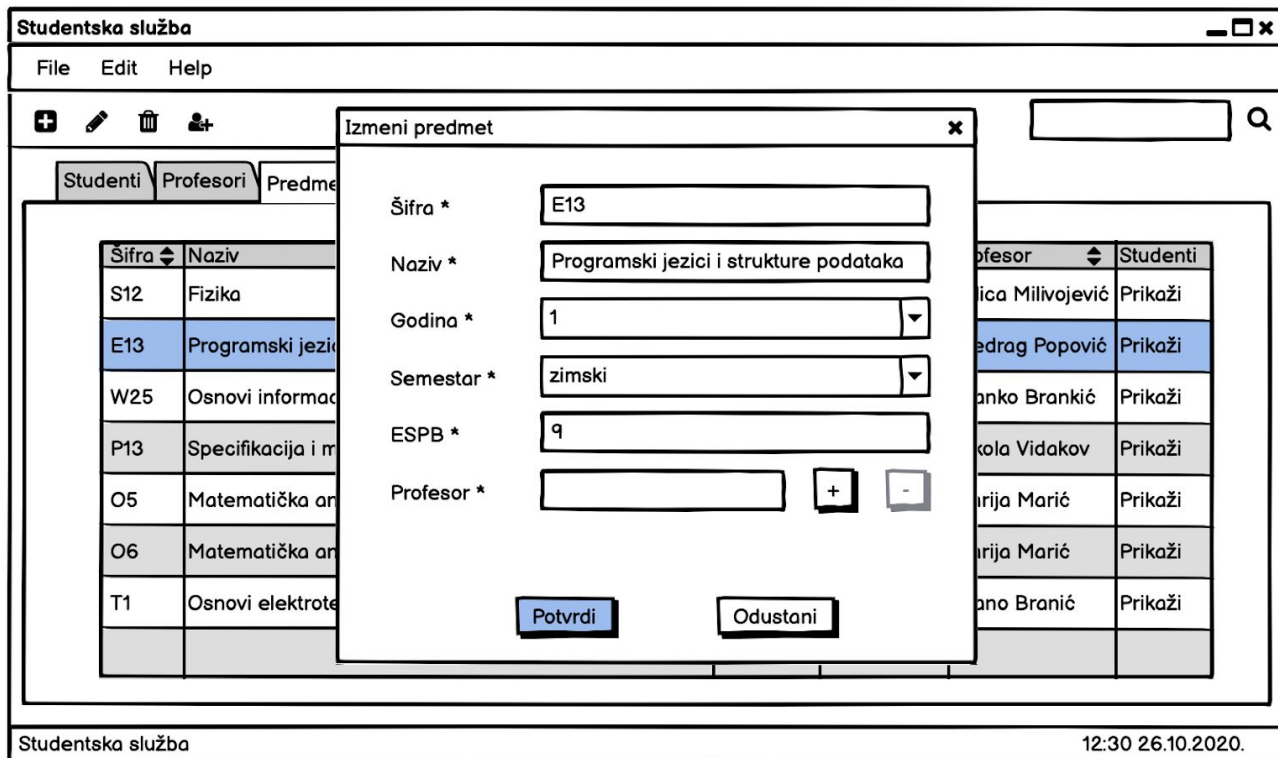
**Slika 18.** Uklanjanje predmeta iz liste predmeta, koje profesor predaje

### 3.19 Dodavanje profesora na predmet (#dodavanje\_profesora\_na\_predmet)

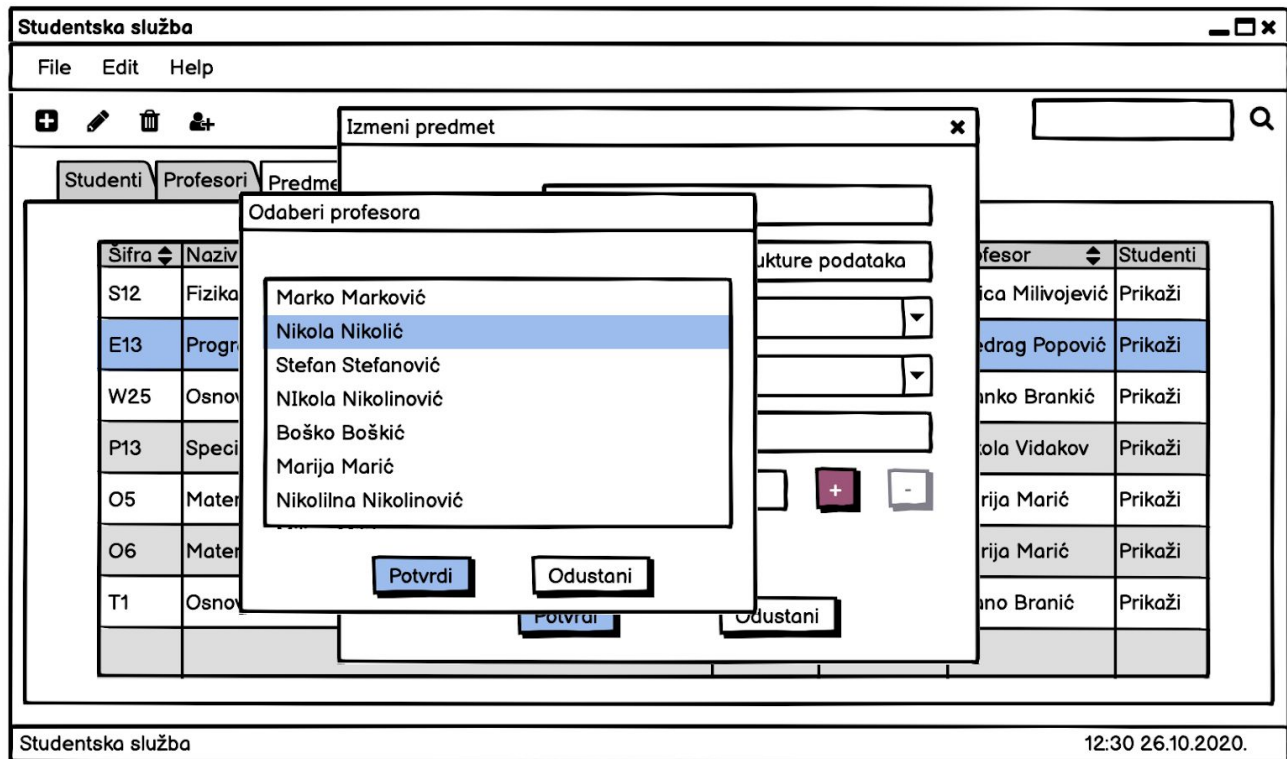
Odabirom taba *Predmeti* glavnog prozora (slika 1), referentu se prikazuje tabela sa spiskom svih predmeta registrovanih u sistemu. Označavanjem reda tabele, kojim se reprezentuje željeni predmet, i izborom funkcionalnosti za izmenu entiteta (objašnjenog u sekciji 3.7), referentu se prikazuje dijalog *Izmeni predmet* (slika 19). Dijalog treba da bude centriran u odnosu na glavni prozor. Dijalog sadrži formu kojom je moguće izmeniti podatke o predmetu na način kako je to objašnjeno u sekciji 3.7.

Na samom dnu forme nalazi se polje koje prikazuje profesora koji predaje izabrani predmet. Ukoliko nijedan profesor ne predaje predmet, tada je pomenuto polje prazno. Pored polja nalaze se dva dugmeta. Prvo dugme (označeno sa +) služi za dodavanje novog profesora na predmet i ono je moguće pritisnuti u situaciji kada nijedan profesor ne drži predmet. U suprotnom, pomenuto dugme nije moguće pritisnuti. Pored dugmeta za dodavanje profesora na predmet, nalazi se dugme za uklanjanje profesora sa predmeta (označeno sa -). Pomenuto dugme je moguće pritisnuti u situacija kada postoji profesor koji predaje izabrani predmet.

U slučaju kada nijedan profesor ne predaje predmet, tada je referentu neophodno omogućiti dodavanje profesora na predmet izborom dugmeta + prikazanog na slici 19. Nakon toga, referentu prikazati dijalog *Odaberi profesora* (slika 20), koji treba da bude centriran u odnosu na dijalog *Izmeni predmet* (zbog potpunijeg prikaza, dijalog *Odaberi profesora* nije centriran na slici 20 kako je to pređašnje opisano). Dijalog *Odaberi profesora* treba da sadrži listu sa svim profesorima registrovanim u sistemu, ispod koje se nalaze dva dugmeta *Potvrdi* i *Odustani*. Odabirom profesora iz liste i pritiskom dugmeta *Potvrdi*, dijalog *Odaberi profesora* se zatvara, pri čemu se sadržaj dijaloga *Izmeni predmet* ažurira, tako da prikaže ime i prezime profesora, koji je upravo dodat na predmet.



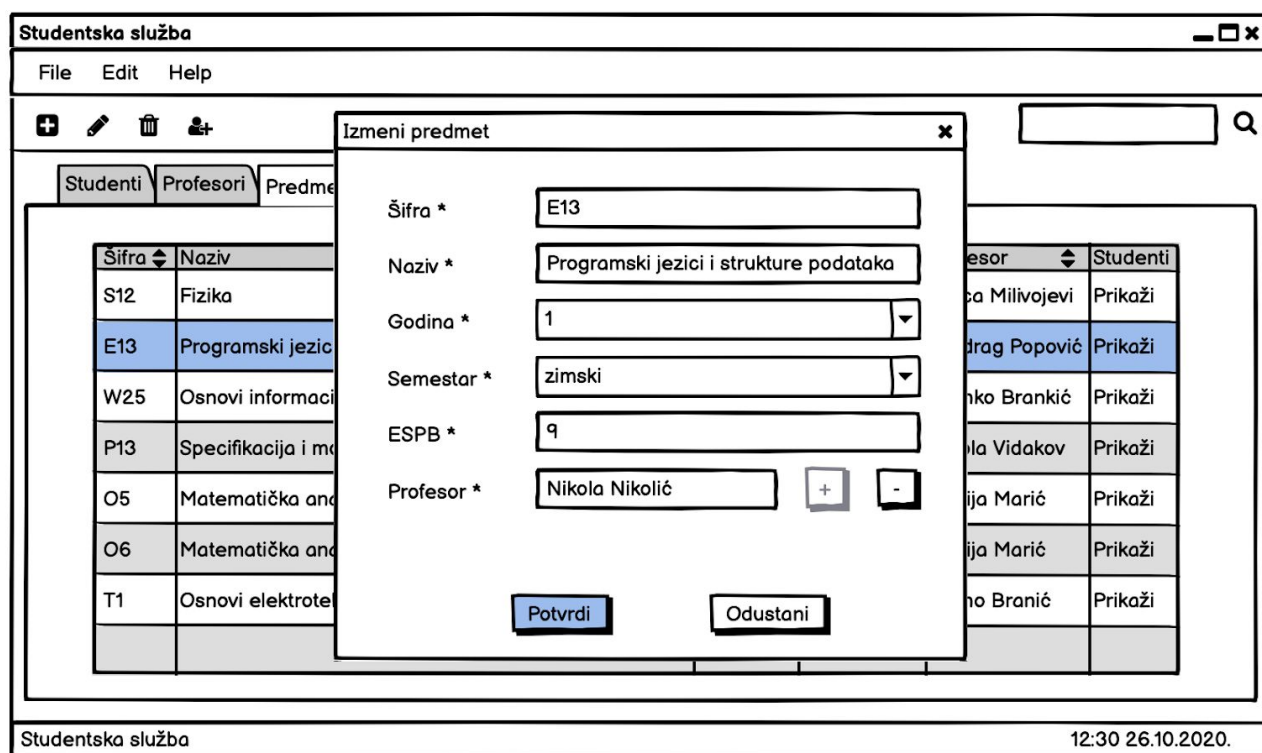
Slika 19. Dijalog za izmenu predmeta



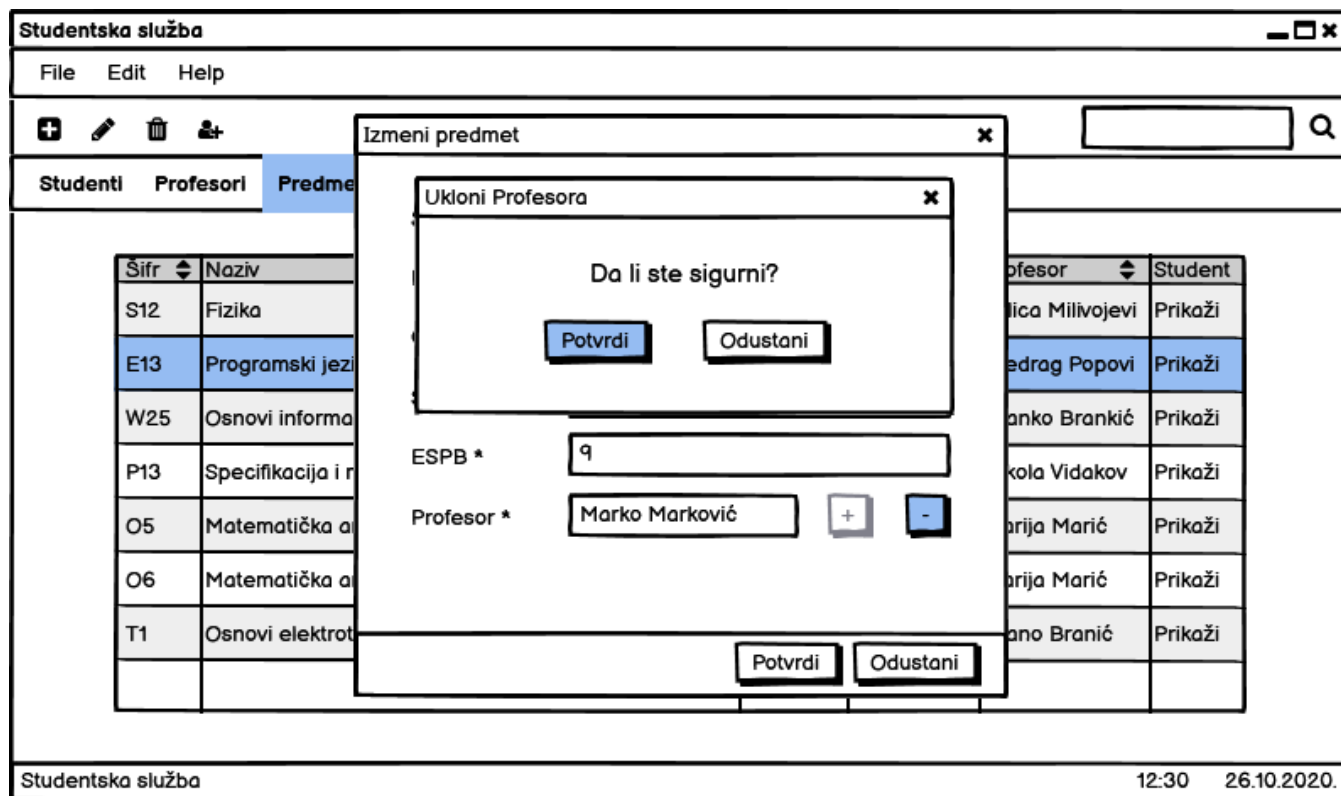
**Slika 20.** Dijalog za dodavanje profesora na predmet

### 3.20 Uklanjanje profesora sa predmeta (#uklanjanje\_profesora\_sa\_predmeta)

Prilikom interakcije sa dijalogom za izmenu predmeta, u situaciji kada postoji profesor koji drži izabrani predmet (slika 21), referentu se daje mogućnost da ukloni profesora sa predmeta. To se radi izborom dugmeta - (prikazanog na slici 21), nakon čega je potrebno prikazati dijalog sa porukom koji od korisnika traži potvrdu akcije (slika 22). Izborom dugmeta *Potvrdi*, profesor se uklanja sa predmeta, a sadržaj dijaloga *Izmeni predmet* se na odgovarajući način ažurira (slika 19).



**Slika 21.** Dijalog za izmenu predmeta (uklanjanje profesora je omogućeno)



Slika 22. Dijalog za izmenu predmeta - uklanjanje profesora

## 4. Perzistencija stanja informacionog sistema

---

Prilikom zatvaranja aplikacije, stanje informacionog sistema mora biti perzistirano na tvrdi disk (engl. *hard drive*) u vidu odgovarajuće hijerarhije datoteka (**#serijalizacija**). Prilikom narednog pokretanja aplikacije, potrebno je izvršiti učitavanje perzisistiranih podataka, te oporaviti prethodno stanje sistema (neposredno pre poslednjeg zatvaranja aplikacije studentske službe) (**#deserijalizacija**).

## 5. Formiranje timova

---

Projektni zadatak se radi u timu od 2 studenta. **Tim čini par studenata koji pripadaju istoj grupi formiranoj od strane Studentske službe Fakulteta tehničkih nauka.** Ovako formirani timovi treba da se upišu [u tab grupe kojoj pripadaju](#).

U slučaju neparnog broja studenata u grupi, student koji ne uspe da formira tim sa kolegom iz iste grupe, može formirati tim sa kolegom iz neke druge grupe koji je u istoj situaciji. Preporuka je da studenti koji na ovaj način formiraju tim slušaju vežbe kod istog asistenta, kako bi mogli nesmetano zakazati konsultacije. Ovako formirani timovi treba da se upišu [u tab "Samostalno raspoređeni"](#).

Studenti koji ne uspeju samostalno da formiraju timove, treba da se upišu [u tab "Neraspoređeni"](#), kako bi bili raspoređeni od strane predmetnih asistenata.

**Napomena: Krajnji rok za prijavu tima je 15.11.2020 (23:59:59)**

## 6. Raspodela zadatka po studentima

---

Prvi student treba da implementira stavke sa sledećim oznakama:

1. #student
2. #ocena
3. #menu\_bar
4. #status\_bar
5. #prikaz\_studenata
6. #dodavanje\_studenta
7. #dodavanje\_predmeta
8. #izmena\_studenta
9. #izmena\_predmeta
10. #brisanje\_studenta
11. #brisanje\_profesora
12. #prikaz\_polozenih\_ispita
13. #dodavanje\_predmeta\_studentu
14. #uklanjanje\_predmeta\_sa\_studenta (ispravljeno 30.12.2020)
15. #ponistavanje\_ocene
16. #pretraga\_studenata
17. #sortiranje\_studenata
18. #dodavanje\_profesora\_na\_predmet
- 19. #uklanjanje\_predmeta\_sa\_profesora (ispravka 30.12.2020)**
20. #deserijalizacija



Drugi student treba da implementira stavke sa sledećim oznakama:

1. #profesor
2. #predmet
3. #glavni\_prozor
4. #toolbar
5. #prikaz\_profesora
6. #prikaz\_predmeta
7. #dodavanje\_profesora
8. #izmena\_profesora
9. #brisanje\_predmeta
10. #prikaz\_nepolozenih\_ispita
11. #upis\_ocene
12. #pretraga\_profesora
13. #pretraga\_predmeta
14. #sortiranje\_profesora
15. #sortiranje\_predmeta
16. #profesor\_predaje\_predmete
17. #dodavanje\_predmeta\_profesoru (**ispravka 04.01.2020.**)
18. #uklanjanje\_profesora\_sa\_predmeta
19. #serijalizacija

## 7. Upotreba Git-a

---

Obavezno je korišćenje Sistema za kontrolu verzija **git**. Kao udaljeni repozitorijum, može se koristiti *Github* ili *Gitlab*. Neophodno je da projekat bude **u privatnom repozitorijumu**, na koji će **predmetni asistent biti dodat kao collaborator/reporter**. Prilikom *commit*-ovanja izvornog koda, neophodno je na početku poruke navesti oznaku stavke koju je student delimično ili u potpunosti implementirao, npr.

**git commit -m "#dodavanje\_studenta Implementacija forme za dodavanje studenta"**

U okviru ovog dokumenta, pored svake funkcionalnosti, nalaze se i oznake koje treba koristiti prilikom *commit*-ovanja.

*Commit*-i čija poruka ne počinje oznakom stavke, biće ignorisani.

Svaki projekat mora imati napisanu **README.md** datoteku, u kojoj će stajati koji je član tima radio stavke studenta 1, odnosno studenta 2. Napisati koja je verzija JDK-a, odnosno *Eclipse* razvojnog okruženja korišćenja. Ukoliko su korišćene eksterne biblioteke, moraju se navesti njihove verzije, a njihov kompajlirani kod je neophodno priložiti i povezati u okviru *Eclipse* projekta.

**Ukoliko studenti postave na git ceo projekat u jednom commit-u ili dan pred rok za predaju projekta gube pravo izlaska na odbranu.**

## 8. Saradnja studenata

---

Saradnja studenata je dozvoljena isključivo između studenata unutar tima. Nije dozvoljena saradnja studenata koji pripadaju različitim timovima. Pre odbrane projektnih zadataka, sva projektna rešenja biće dodata u bazu projekata, nad kojom će biti pokrenut softver za detekciju plagijata. Ukoliko se tom prilikom utvrde poklapanja, projekti će biti proglašeni plagijatima, te timovi koji su razvijali pomenute projektne zadatke gube pravo izlaska na odbranu.

Ukoliko kod preuzimate iz dozvoljene literature (materijali sa vežbi, *stack overflow*, *google*, ...), neophodno je referencirati ga komentaram iznad dela preuzetog koda. U suprotnom, kod će se smatrati plagiranim.

**Nije dozvoljena zamena stavki među članovima tima.** Ukoliko student odradi stavke koje nisu njemu namenjene, bodove gube oba člana tima.

## 9. Konsultacije u vezi sa projektnim zadatkom

---

Studenti po potrebi mogu zakazati konsultacije u vezi sa projektnim zadatkom. Neophodno je da se jedan od članova tima obrati predmetnom asistentu sa spiskom pitanja, kako bi se predmetni asistent pripremio za konsultacije.

## 10. Odbrana projekta

---

Odbrana projekta će se odvijati u prostorijama Fakulteta tehničkih nauka u terminu koji će biti naknadno objavljen. Projekti se nakon kloniranja repozitorijuma moraju prepoznati kao Postojeći *Eclipse* projekti prilikom *import-a* (*Existing Projects Into Workspace*). Ukoliko nakon *import-a* nije moguće pokrenuti kod zbog eventualnih grešaka (greške u *import-u*, greške prilikom kompajliranja, ...), oba studenta biće ocenjena minimalnim nenegativnim brojem bodova.

Studenti će dobiti specifikaciju odgovarajućeg stanja informacionog sistema studentske službe u kome se aplikacija mora nalaziti prilikom pokretanja na odbrani projektnog zadatka. Ukoliko se na odbrani utvrdi da je stanje neodgovarajuće, oba studenta gube pravo odbrane.

## 11. Domaći zadatak 1.

---

Domaći zadatak 1. obuhvata implementaciju sledećih stavki od strane:

- studenta 1
  - #menu\_bar
  - #status\_bar
- studenta 2
  - #glavni\_prozor (tabove dodati prilikom izrade drugog domaćeg zadatka)
  - #toolbar

Na udaljenom repozitorijumu neophodno je kreirati **granu** “**DOMACI-1**” na kojoj će se nalaziti rešenje prvog domaćeg zadatka. Ostale grane biće ignorisane prilikom pregledanja.

**Rok za izradu prvog domaćeg zadatka je 06.12.2020. (23:59:59).**

**Napomena:** Nakon što implementirate pomenute stavke, kreirajte navedenu granu. Nakon toga, na drugoj grani možete nastaviti sa izradom projektnog zadatka.

## 12. Domaći zadatak 2.

---

Domaći zadatak 2. obuhvata implementaciju sledećih stavki od strane:

- studenta 1
  - #student
  - #ocena
  - #prikaz\_studenata
  - #dodavanje\_studenta
- studenta 2
  - #profesor
  - #predmet
  - #prikaz\_profesora
  - #dodavanje\_profesora

Na udaljenom repozitorijumu neophodno je kreirati **granu** “**DOMACI-2**” na kojoj će se nalaziti rešenje drugog domaćeg zadatka. Ostale grane biće ignorisane prilikom pregledanja.

**Rok za izradu drugog domaćeg zadatka je 27.12.2020. (23:59:59).**

**Napomena:** Nakon što implementirate pomenute stavke, kreirajte navedenu granu. Nakon toga, na drugoj grani možete nastaviti sa izradom projektnog zadatka.

Za domaći 2 potrebno je ispuniti i sve nefunkcionalne zahteve, koji slede u nastavku.

**(14.12.2020.) Posebno obratiti pažnju na sledeće nefunkcionalne zahteve:**

- Podržati **UTF-8 encoding** u *Java (Eclipse)* projektu, kako biste bili sigurni da će se slova srpske cirilice/latinice korektno prikazati na svim platformama.
- Putanje do eksternih datoteka (npr. slike) ne navoditi upotrebom separatora putanja koji je karakterističan za jedan tip datotečkog sistema (engl. *File System*). Putanje graditi uz oslonac na **File.separator**.
- Izborom stavki menija **About** i **Help** prikazati odgovarajuće **dijaloge**. Dijalog **About** mora biti **u potpunosti implementiran**. **Help** dijalog mora **minimalno** da opisuje **način korišćenja trenutno implementiranih funkcionalnosti**, kao i sve **mnemonike** i **akceleratora** registrovane u aplikaciji.
- Neophodno je implementirati **README.md** datoteku koja će minimalno sadržati:
  - **imena studenata** podeljena po ulogama (**student 1** i **student 2**)
  - verziju korišćenog **JDK-a**,

- verziju korišćenog **Eclipse** razvojnog okruženja,
- **Povezati JDK sa Eclipse projektom** kako bi se projekat mogao pokrenuti nakon *import*-a bez naknadnih konfiguracija.
- Ukoliko koristite **eksterne biblioteke**, navesti ih u **README.md** datoteci, priložiti uz projekat u **lib** foleru i povezati u **Java Build Path** Eclipse projekta.
- Klase logički grupisati u **pakete**, pri čemu je potrebno imati najmanje **tri paketa** koji odgovaraju komponentama **MVC arhitekture** (*model*, *view* i *controller* paketi)
- **Ne smete postaviti klase u default paket** (direktno ispod *src* paketa).
- **Postaviti meta-podatke Eclipse projekta** kako bi se projekat mogao importovati na način objašnjen u sekciji 10.
- **Programski kod razvijati u skladu sa [propisanim konvencijama Java programskog jezika](#).**

## 13. Funkcionalnosti za samostalno istraživanje

---

Tokom rada na projektu, zainteresovani studenti mogu pristupiti samostalnom istraživanju, koje donosi mogućnost osvajanja dodatnih bodova. Pomenute bodove profesor može uzeti u obzir tokom polaganja ispita. O bodovanju funkcionalnosti navedenih u ovom poglavlju naknadno će odlučiti profesor.

**Napomena:** Studenti koji se odluče za implementaciju stavki iz ove sekciju, mogu se konsultovati sa predmetnim asistentom isključivo u vezi sa tekstom specifikacije istih. Detalje tehničkog i implementacionog nivoa dužni su da sami istraže.

**Napomena 2:** Studenti koji se odluče da rade neku od navedenih funkcionalnosti, moraju je navesti unutar README.md datoteke.

### 13.1 Lokalizacija (#lokalizacija)

Korisnicima omogućiti rad na aplikaciji na minimum dva jezika:

- srpski (#lokalizacija\_srpski - potrebno je da je implementira student 1) i
- engleski (#lokalizacija\_engleski - potrebno je da je implementira student 2).

Lokalizacija obuhvata sve elemente grafičkog korisničkog interfejsa (labele, dugmad, naslove, tooltipove, itd).

Omogućiti referentu studentske službe da u bilo kom trenutku upotrebe aplikacije izvrši promenu željenog jezika.

Kao smernice za implementaciju lokalizacije možete uzeti referentni primer sa vežbi 9.

### 13.1 Napredna pretraga (#napredna\_pretraga)

Podržati naprednu pretragu za sledeće entitete po navedenim parametrima (atributima):

- Student (#napredna\_pretraga\_studenata - treba da je implementira student 1):
  - indeks
  - ime
  - prezime
  - godina studija
  - prosek
  - predmeti koje student trenutno sluša
- Predmet (#napredna\_pretraga\_predmeta - treba da je implementira student 2)
  - šifra predmeta

- naziv predmeta
- broj ESPB bodova koje predmet nosi
- semestar u kom se predmet sluša - *LETNJI/ZIMSKI*
- godina u kojoj se predmet izvodi
- profesori koji predaju predmet

Naprednu pretragu čine upiti koji se unose u tekstualno polje opisano u sekciji 3.14, poštujući sledeći format: **<entitet\_pretrage> = (<kriterijum\_napredne\_pretrage>):**

- **<entitet\_pretrage>** - može biti **student**, odnosno **predmet**
- **<kriterijum\_napredne\_pretrage>** - sastoji se od proizvoljnog broja **uslovnih izraza**. Izrazi se kombinuju upotrebom logičkih operatora **AND/and/∧** i **OR/or/∨**, pri čemu je neophodno obezbediti upotrebu zagrada. Uzeti u obzir prioritet operatora.

### Uslovni izrazi

Uslovni izrazi imaju sledeći format:

**<atribut\_entiteta> <operator> <vrednost\_atributa\_entiteta>**

**<atribut\_entiteta>** - u zavisnosti od entiteta sistema, u obzir uzeti sledeće attribute (ključne reči):

- Student:
  - indeks
  - ime
  - prezime
  - godina
  - prosek
  - slusa\_predmete
- Predmet:
  - sifra
  - naziv
  - ESPB
  - godina
  - semestar
  - profesori
    - Profesor:
      - ime
      - prezime

- titula
- zvanje

**<operator>** - operatori koji dolaze u obzir su sledeći:

- ==
- !=
- <
- >
- <=
- >=

Za više informacija o značenju operatora pogledati navedene primere.

**<vrednost\_atributa\_entiteta>** - vrednost može biti *String*, regularni izraz ili numerička vrednost. Za više informacija pogledati primere ispod.

Ukoliko je **<atribut\_entiteta>** tipa *String*, tada se kao operatori mogu koristiti == i !=, dok **<vrednost\_atributa\_entiteta>** može biti *String*, odnosno *regularni izraz*. Primeri ovakvog uslovnog izraza su sledeći:

- **(\*) nazivAtributa == "Vrednost Atributa"** - Uslovni izraz je tačan za sve entitete čiji atribut ***nazivAtributa*** ima vrednost **"Vrednost Atributa"**
- **nazivAtributa == /regularni izraz/** - Uslovni izraz je tačan za sve entitete čiji atribut ***nazivAtributa*** odgovara regularnom izrazu **/regularni izraz/**

Ukoliko je **<atribut\_entiteta>** numeričkog tipa, tada se kao operator može koristiti proizvoljan relacioni operator (==, !=, <, <=, >, >=), dok vrednost atributa predstavlja broj. Primeri ovakvog uslovnog izraza su:

- **nazivAtributa > 100.15** - Uslovni izraz je tačan za sve entitete čiji atribut **nazivAtributa** ima vrednost veću od broja **100.15**

Ukoliko je **<atribut\_entiteta>** enumeracija, tada je dozvoljeno koristiti operator == i !=, dok se kao vrednost atributa koristi *String* reprezentacija vrednosti enumeracije. Primer ovakvog izraza bi odgovarao izrazu označenom sa (\*).

Ukoliko se **<atribut\_entiteta>** odnosi na listu referenciranih entiteta, tada se kao operatori mogu koristiti == i !=, dok se kao vrednost koristi poseban kriterijum za izdvajanje liste referenciranih objekata. Ovakav tip izraza najbolje će biti objašnjen na primeru studenta koji sluša predmete koji zadovoljavaju jedan od 3 uslova:

- šifra predmeta sadrži sekvencu karaktera **20A**



- naziv predmeta je "Analiza 1"
- godina na kojoj se predmet izvodi je veća od 2

Izraz: `slusa_predmete == { sifra == /*20A*/ or naziv == "Analiza 1" or godina > 2 }`

Slično se može koristiti i and operator.

### Primeri upita napredne pretrage:

#### Napredna pretraga studenata (#napredna\_pretraga\_studenata)

- studenti 3. godine koji se zovu Marko
  - `studenti = (ime == "Marko" and godina == 3)`
- studenti čiji indeks počinje karakterima RA i pri tom su: ili 2. godina ili imaju prosek preko 7,49
  - `studenti = (index == /^RA*/ and (godina == 2 or prosek > 7,49))`
- studenti imaju prosek preko 9 i koji slušaju predmete koji zadovoljavaju jedan od tri uslova:
  - sifra predmeta sadrži sekvencu karaktera 20A
  - naziv predmeta je "Analiza 1"
  - godina na kojoj se predmet izvodi je veća od 2
    - `studenti = (prosek > 9 and slusa_predmete == { sifra == /*20A*/ or naziv == "Analiza 1" or godina > 2 })`
- studenti koji su ili 1. godina ili čiji broj indeksa sadrži sekvencu karaktera "2020"
  - `studenti = ((godina == 1) or (index == /\w*2020\w*/))`
- studenti koji slušaju predmet koji ima 9 ESPB bodova i koji je sa 1. godine
  - `studenti = (slusa_predmete == {ESPB == 9 and godina == 1})`
- studenti koji imaju prosek između 7,20 i 8,70
  - `studenti = (prosek >= 7,20 and prosek <= 8,70)`
- studenti čiji dosije sadrži dvocifren broj nakon dvoslovnog prefiksa, koji nisu 4. godina, imaju prosek iznad 8 i slušaju predmet koji ima preko 6 ESPB poena
  - `studenti = (index == /[A-Za-z]{2}-?\d{2}/\d{4}/ and godina != 4 and prosek > 8 and slusa_predmete == {ESPB > 6})`

#### Napredna pretraga predmeta (#napredna\_pretraga\_predmeta)

- predmeti koji u imenu sadrže sekvencu karaktera *softver*
  - `predmeti = (naziv == /\w*softver\w*/)`
- predmeti koji zadovoljavaju jedan od dva uslova:
  - slušaju se u 4. semestaru i nose više od 3 ESPB poena

- slusaju se u 1. semestru i imaju manje ili jednako od 5 ESPB poena
  - predmeti = (godina == 2 and semestar == "LETNJI" and ESPB > 3 or godina == 1 and semestar == "ZIMSKI" and ESPB <= 5)
- predmeti koji imaju preko 6 bodova i slusaju se na 3. i 4. godini
  - predmeti = (ESPB > 6 and (godina == 3 or godina == 4))
  - predmeti = (ESPB > 6 and (godina > 2 and godina < 5))
- predmeti koji se slusaju u 1, 2, ili 3. godini i koje predaju profesori koji zadovoljavaju jedan od dva uslova:
  - prezime profesora se završava na završava na *vić*
  - ime profesora je "Marko"
    - predmeti = (godina <= 3 and profesori == {prezime == /[A-Z][a-z]\*vić\$/ or ime == "Marko"})
- predmeti čija šifra predmeta sadrži samo brojeve i naziv sadrži jedno od: *ika*, *elektro* ili *tele*. Pri tom se slusaju ili u 8. semestra ili ih drže profesori koji zadovoljavaju jedan od dva uslova:
  - ime profesora je branko "Branko"
  - zvanje profesora je DOCENT
  - zvanje profesora je REDOVNI\_PROFESOR
  - profesor je doktor nauka
    - predmeti = (sifra == /^d+\$/ and naziv == /(w\*(ika|elektro|tele)w\*)/) and (godina == 4 and semestar == "LETNJI" or profesori == {ime == "Branko" or zvanje == "DOCENT" or zvanje == "REDOVNI\_PROFESOR" or titula == "DR"})



## Odgovori na najčešće postavljena pitanja

---

1. Korisnička imena asistenata na GitHub platformi:

- a. Marija Kovačević - **marija-kovacevic**
- b. Lazar Jović - **LazarJovic**
- c. Balša Šarenac - **balsa-sarenac**
- d. Vladimir Inđić - **vladaindjic**

2. **Pitanje:** Da li glavni prozor mora biti *resizable*?

**Odgovor:** Da.

3. **Pitanje:** Da li na glavnom prozoru moraju postojati komponente na koje će se smestiti prikaz studenata... (**domaći 1**)?

**Odgovor:** Postavite bar panel na kome će stajati labela “**TODO:** Prikaz entiteta sistema”.

4. **Pitanje:** Da li treba za **domaci 1** da budu odrađene i funkcionalnosti za meni?

**Odgovor:** Nije potrebno vezivati *ActionListener*-e za stavke menije. Potrebno je kreirati sve menije, njihove stavke, prikazati odgovarajuće ikonice, mnemomike, akceleratora, ...

5. **Pitanje:** Da li komitujemo samo .java datoteke, ili ceo projekat?

**Odgovor:** Možete postaviti čitav projekat. Eventualno ignorišite *byte-code*. Neophodno je da na udaljenom repozitorijumu imate **src paket, meta-datoteke Eclipse projekte**, dodatne resurse (tipa slike, datoteke, itd.), kao i sve eksterne biblioteke (.jar archive) koje koristite.

6. **Pitanje:** Da li treba napraviti kontrolu unosa godine kod studenta (sa veće da ne može na manju)?

**Odgovor:** Studenti viših godina mogu slušati predmete nižih godina, obrnuto nije moguće. Obratiti pažnju na uslove navede u **sekciji 3.11**.

7. **Pitanje:** Da li koristimo bazu podataka ili datoteke?

**Odgovor:** **Datoteke!** Obratiti pažnju na **sekciju 4**.

8. **Pitanje:** Da li postoji neki šablon kako treba da pišemo komentare kojima navodimo iz koje dozvoljene literature smo preuzeli deo koda?

**Odgovor:** Iznad dela koda koji je preuzet u doc komentaru navesti

`/**REFERENCA: ... */`

9. **Pitanje:** Da li sav kod pišemo na engleskom jeziku, uključujući i nazive klasa iz modela i njihova polja?

**Odgovor:** Engleski ili srpski što se koda tiče. Labele na GUI treba da budu napisane na srpskom.

10. **Pitanje:** Dodao sam shortcut-e sa crticom (Ctrl-N), u specifikacijama pise sa plusom ('Ctrl+N')

**Odgovor:** To je u redu. Swing zapisuje akceleratora na taj način. Bitno je da u kombinaciji dugmadi dođe do okidanja akceleratora.

11. **Pitanje:** Da li moramo da uradimo samo grafički deo ili mora da se urade sve funkcionalnosti (dodavanje studenta, izmena postojećeg intetiteta ...)?

12. **Pitanje:** Da li se zna okvirno do kada je predaja celog projekta?

**Odgovor:** Poslednja nastavna nedelja tekućeg semestra.

13. **Pitanje:** Da li je problem što je ceo prvi domaći postavljen u jednom commit-u? Da li je problem što je više funkcionalnosti postavljeno u jednom commit-u?

**Odgovor:** **Postavljanje na repozitorijum u samo jednom commit-u ceo domaći zadatak ili više funkcionalnosti nije u skladu sa specifikacijom projekta.** Pored svake funkcionalnosti, u okviru ovog dokumenta, sa razlogom stoji oznaka. Oznake treba koristiti prilikom commit-ovanja.

14. **Pitanje:** Kako referencirati literaturu preuzetu sa interneta?

**Odgovor:** Navesti link do sajta sa kog je preuzet kod.

15. **Pitanje:** Da li Help dijalog treba da bude kompletan za **domaci-1**?

**Odgovor:** Opisati ukratko šta je odrađeno. Navesti iz čega se sastoji glavni pozor (MenuBar, ToolBar, StatusBar, ...). Objasniti koje stavke menija/toolbara postoje, koji su akceleratori i mnemonici registrovani. Šta bi svaka od stavki menija/toolbara trebalo da radi. Za stavke koje nisu implementirane navesti komentar da će Help dijalog biti naknadno proširen. Opisati šta se sve nalazi u status baru, ...

**Za domaći-1 nije obavezno da se otvaraju novi dijalozi kada se izabere stavka iz menija.**

16. **Pitanje:** U specifikaciji piše da dodamo akceleratora na ikonice u toolbar-u, a JButton kao takav nema metodu za postavljanje akceleratora, da li se misli na mnemomike? Nešto tipa `button.setMnemonic(KeyEvent.VK_C);` ?

**Odgovor:** Ukoliko koristite apstraktne akcije, na jednom mestu možete specificirati i akceleratora i mnemonike, koji će biti dodati od strane Swing-a tamo gde je to propisano konvencijama programiranja GUI desktop aplikacije.

17. **Pitanje:** U zadatku pise da treba dodati acelerator u okviru dugmadi *toolbar*-a, pa me sada zanima da li je poenta da to dugme odreaguje tako sto ce simulirati efekat pritiska misem, ili je poenta u tome da acelerator samo trigeruje lisener bez da simulira pritisak. Posto ja simulaciju pritiska mogu da ostvarim samo pomocu *setMnemonik* metode

**Odgovor:** Prilikom izbora određene kombinacije tastera sa tastature, doći će do prepoznavanja akceleratora, koji treba da dovede do okidanja događaja akcije (*ActionEvent*). Pomenuti događaj biće obrađen od strane odgovarajućeg osluškivača (*ActionListener*). Nema potrebe ništa simulirati. Preporuka je da se koriste apstraktne akcije (*AbstractAction*) i da se one smeštaju kao stavke menija, odnosno *toolbar*-a.

18. **Pitanje:** Da li ja smem da nakon sto odradim *push* na udaljeni repozitorijum vrsim neke izmene i nadograđivanje ako mislim da nesto treba ispraviti?

**Odgovor:** Naravno. Sve izmene napravljene do roka za DOMACI-1, DOMACI-2, predaju projektnog zadatka bice uvažene. Bitno je samo da pomenute stavke predate u vidu odgovarajuće udaljene grane.

19. **Pitanje:** Da li je u redu da apstraktnu akciju dodam na dugme, a zatim dugme na *toolbar*? Ovo je urađeno kako bih mogla da uredim ivice dugmadi *toolbar*-a.

**Odgovor:** Da.

20. **Pitanje:** Da li je potrebno i za dugme pored tekstualnog polja za pretragu entiteta podesiti akcelerator? Namestila sam da se izborom akceleratora CTRL+F kursor pojavi u pomenutom tekstualnom polju, dok se pritiskom na enter dok pomenuto polje ima fokus aktivira akcija za pretragu entiteta.

**Odgovor:** Nema potrebe postavljati akcelerator na pomenuto dugme. Pritisak pomenutog dugmeta, kao i pritisak tastera enter prilikom unosa teksta u pomenute polje treba da dovedu do okidanja istog *ActionEvent*-a, koga će odgovarajući *ActionListener* obraditi tako da obavi pretragu prikazanih entiteta.

21. **Pitanje:** Da li je potrebno podešavati *ActionEvent* za pretragu za domaći 1?

Odgovor: Nema potrebe. Dovoljno je samo vezati *ActionEvent* za prikaz dijaloga *Help*, odnosno *About*.

22. **Pitanje:** Koliko je striktna podela između #studenta1 i #studenta2? Koleginica i ja neke probleme rešimo zajedno, pa me zanima da li je u redu da ona vrši neke promene na mojim klasama i obrnuto,

**Odgovor:** Podela je napravljena kako bi se poštovala, kako od strane članova tima, tako i od strane predmetnih asistenata prilikom ocenjivanja. Prilikom implementacije pojedinih funkcionalnosti, potrebno je koristiti klase koje je koleginica/kolega implementirala/implementirao, ali ne i menjati njihovo ponašanje. Ukoliko primetite da je koleginica/kolega napravila/napravio previd prilikom implementacije neke od stavki dodoljenih njoj/njemu, dužni ste da joj/mu to prijavite kako bi popravila/popravio, ali ne i samostalno da ih menjate. Idealno bi bilo ukoliko biste korisili *GitHub issues* za evidenciju propusta.

23. **Pitanje:** Napravila sam ikonice i tooltip-ove, ali vidim u napomeni u specifikaciji da treba povezati akceleratora, pa mi nije jasno da li treba da napravim i prozore za dodavanje, izmenu i brisanje ili je to deo drugog domaćeg?

**Odgovor:** Za sada je potrebno vezati akceleratora, ali nema potrebe obrađivati događaje akcija koji oni prouzrokuju. Dovoljno je samo implementirati *ActionListener*-e zadužene za otvaranje About i Help dijaloga. Preporuka je da koristiti apstraktne akcije kako biste jednostavnije manipulirali ikonicama, menmonicima i akceleratorima.

24. **Pitanje:** Da li je za prvi domaći (pre svega mislim na toolbar) uopšte potrebno implementirati apstraktne akcije (pošto nikakve akcije još ne mogu da se izvrše)?

**Odgovor:** Potrebno ih je vezati za dugmad *toolbar*-a. Naknadno ćete implementirati logiku *actionPerformed* metoda.

25. **Pitanje:** Da li prozore koje dobijamo klikom na About i Help možemo implementirati preko *ShowMessageBox*-a?

**Odgovor:** Ne. Neophodno je koristiti *JDialog*.

26. **Pitanje:** Mi smo stavili domaći 1 na granu Domaći-1 prije nekoliko dana. Sada sam vidio da treba da se doda About i Help dijalog. U međuvremenu sam u svom projektu napravio klase Student i Ocjena. Sada bih htio da na granu domaći 1 dodam samo dijaloge za help i about. Zanima me kako to da uradim a da mi se na tu granu ne dodaju klase Student i Ocjena?

**Odgovor:** To što ste trenutno odraditi, možete *stash*-ovati ili *commit*-ovati na grani na kojoj ste trenutno (main, master, ...). Nakon toga, prebacite se na granu DOMACI-1, napravite pomenute dijaloge i *commit* napravite na toj grani. Nakon toga *push* na

istoimenu udaljenu granu. Zatim se prebacite na prethodnu granu (main, master, kako god da se zove) i odradite *git merge DOMACI-1*, kako bi ta dva dijaloga bila vidljiva i na toj grani.

27. **Pitanje:** Da li je problem ukoliko sam na nekoliko commitlsto tako u prvih nekoliko komita sam zaboravio da stavim "#". Nadam se da to neće predstavljati neki problem.

Odgovor: to se tice # u prvih nekoliko commit-a, zaista ne smem garantovati koliki ce se penali placati za to, moram se konsultovati sa ostalim kolegama.

28. **Pitanje:** Na nekim commit-ima sam zaboravio staviti #, da će ti commit-i biti uvaženi?

**Odgovor:** Pročitajte sekciju 7.

29. **Pitanje:** Kako promeniti poruku starog commit-a?

**Odgovor:** Uputstvo možete pronaći na [zvaničnoj GitHub dokumentaciji](#).

**Napomena:** Kada prepoznate slučaj koji odgovara Vašem problemu, objašnjene korake pokušajte da napravite na test-repozitorijumu, kako biste bili sigurni da metod radi i da neće dovesti do problema sa granama.

30. **Pitanje:** Prilikom menjanja prethodnih *commit* poruka primenom načina objašnjenog u prethodnom odgovoru, došao sam do zaključka da ne mogu *commit* poruku započeti sa #, jer se to prepoznaje kao komentar. Da li je u redu da stavim jedan razmak pre samog znaka #?

**Odgovor:** To je u redu. Napomenuo bih da postoji šansa da ćete prilikom rada na projektu u jednom commit-u popraviti nedostatke nekoliko funkcionalnosti. Tada commit poruka treba da sadrži nekoliko oznaka, koje će biti prepoznate prilikom automatizovanog pregledanja.

31. **Pitanje:** Da li je dovoljno da prvi domaci na github postavi jedan student iz tima ili moraju oba studenta, s obzirom da smo zajedno radile taj deo projekta?

**Odgovor:** Jedan student iz tima će napraviti granu DOMACI-1 na udaljenom repozitorijumu. Oba studenta moraju imati odgovarajuće commit-e na pomenutoj grani.

32. **Pitanje:** Da li ja kao student 1 treba da inicijalizujem svoje klase u MainFrame (za MainFrame je zaduezen student 2)?

**Odgovor:** Kolega iz tima će implementirati klasu MainFrame, a Vi ćete je po potrebi proširiti funkcionalnostima koje su Vam dodeljene.



33. **Pitanje:** Pošto za svako dugme u toolbar-u treba da stoji i tooltip da li to može da se napiše u kodu ovako: `putValue(SHORT_DESCRIPTION, "Dodaj");` ? Pomenuti iskaz bi se nalazio u klasi naslednici klase *AbstractAction*.

**Odgovor:** Da.

34. **Pitanje:** Kada pravimo toolbar, potrebno je da za svaku stavku vezemo akcelerator, posto buttoni nemaju metodu `set accelerator`, da li je dovoljno da napravimo mnemonik ili treba nekako drugacije da realizujemo akcelerator?

**Odgovor:** Preporuka je da koristite apstraktne akcije (*AbstractAction*).

35. **Pitanje:** Da li je po pravilu mvc sablona da se klasa `action_listener` nalazi u controller-u?

**Odgovor:** Registracija odgovarajućeg objekta (instance klase koja implementira *ActionListener* interfejs) obaviće se u nekoj od *view* komponenti. Zadatak `actionPerformed` metode pomenutog objekta jeste da se pozove odgovarajuća metoda kontrolera.

U zavisnosti od načina definicije klase koja implementira pomenuti interfejs (regularna klasa, anonimna klasa, ...), implementacija same *actionPerformed* metode će se nekada nalaziti unutar koda *view* komponente, a nekada van nje(ga).

Klasa kontrolera mora biti u zasebnom paketu, dok se klase koje implementiraju *listener-e* mogu smestiti u proizvoljan paket. Preporuka je da napravite poseban paket `listeners` (ukoliko ne koristite anonimne klase).

36. **Pitanje:** Da li je potrebno da uradimo merge grane DOMACI-1 pre neko što nastavimo sa radom na projektu?

**Odgovor:** Ukoliko želite da funkcionalnosti pomenute grane prebacite na drugu granu gde želite nastaviti razvoj, onda je odgovor da.

37. **Pitanje:** Da li je potrebno da pravim natpise i polja za unos teksta za ulicu, broj, mesto i poštanski broj? Odnosno, šta će nam biti zadato u okviru jedne adrese studenta ili profesora u podacima koji treba da budu stanje aplikacije pred odbranu projekta?

**Odgovor:** Dovoljno je da bude samo jedan `TextField`, tj. adresa u vidu jednog stringa.

38. **Pitanje:** Da li je prilikom izmene studenta potrebno onemogućiti da se promeni broj indeksa ili se i on može promeniti? I da li ovo važi i za odgovarajuće attribute kod predmeta i profesora?

**Odgovor:** Omogućite promenu indeksa, ali vodite računa da indeks ostane jedinstven. Isto važi i za odgovarajuće attribute kod predmeta i profesora.

39. **Pitanje:** Prilikom dodavanja predmeta, da li je neophodno da se odmah doda i profesor koji drži predmet ili je moguće da to polje ostane nepopunjeno?

**Odgovor:** Prilikom dodavanja predmeta ne morate odmah da dodate i profesora. Profesora možete i kasnije da postavite na odgovarajući predmet.

40. **Pitanje:** Možete li precizirati ponašanje dugmeta za potvrdu unosa entiteta u zavisnosti od (ispravne) popunjenosti polja unosa?

**Odgovor:** Korisničku akciju proslediti odgovarajućem kontroleru koji će model iz jednog konzistentnog stanja prevesti u drugo konzistentno stanje. Nakon toga, ažurirati komponentu za prikaz na odgovarajući način.

41. **Pitanje:** Da li sortiranje elemenata tabele sme da se radi ovom funkcijom: *setAutoCreateRowSorter(true)*?

**Odgovor:** Ukoliko funkcija dovodi do željenog rezultata, pri čemu je podržana u standardnom JDK-u (JTable), onda je odgovor da.

42. **Pitanje:** Imam pitanje u vezi entiteta student koji ima više podataka nego što treba da prikazemo u tabeli. Da li te podatke treba da zapamtimo u bazi, ali da ih ne ispisemo u tabeli?

**Odgovor:** Podatke ćete čuvati na nivou objekata u odgovarajućim atributima. U tabeli prikazujete podskup informacija o studentima, dok se preostale informacije prikazuju u dijalogu za izmenu studenta.

43. **Pitanje:** Da li možemo pri unosu studenta da stavimo da su budžet i prva godina u combobox-u unaprijed čekirani?

**Odgovor:** Da.

44. **Pitanje:** Kod pravljenja studenata, da li je moguće uneti godinu upisa različitu od one u broju indeksa; ili broj indeksa u sebi ne sadrži godinu upisa (već samo RA80 npr), pa se u prikazu tabele to spoji?

**Odgovor:** Što se formata indeksa tiče, nije potrebno konkatenerati godinu studija. Mislim da se ranijih godina brojevi indeksa nisu formalni po formatu kao danas, tako da se može desiti da negde godine ni nema. Svakako nije eksplicitno navedeno u specifikaciji, te se od Vas to neće zahtevati. Ono što je bitno jeste da je indeks jedinstven za svakog studenta (pogledati pitanje 38.).

45. **Pitanje:** Da li prikaz studenata podrazumeva da Student1 napravi funkcionalnu tabelu koju Student2 poziva i samo prikazuje u okviru tabova, ili Student1 sam poziva prikaz svoje tabele (koliko smo razumele, Student 2 je zadužen za tabove u okviru glavnog prozora, pa nam je malo nejasna podela posla ovde).

**Odgovor:** Student 2. je zadužen da napravi TabbedPane komponentu na koju će studenti dodavati tabove koji odgovaraju entitetima koje treba prikazati. Student 1 doda tab studenti, ostalo student 2.

46. **Pitanje:** Domaći 2 ne obuhvata rad sa fajlovima? Da li je ok da za ovaj korak prikaz uradimo kao u jednom od projekata sa vežbi, gde se u kodu manualno doda nekoliko početnih entiteta za prikaz?

**Odgovor:** Da!

47. **Pitanje:** Kako ste rekli da proveravamo da li smo dobro povezali jdk/eksterne biblioteke sa projektom?

**Odgovor:** Potrebno je izvršiti sledeći postupak:

- a. Klonirati projekat na novu lokaciju na datotečkom sistemu.
- b. kreirati novi Eclipse radni prostor.
- c. Importovati prethodno klonirani projekat kao postojeći Eclipse projekat
- d. Proveriti da li se javljaju greške prilikom importa.
- e. Pokrenuti projekat.

Ukoliko se prilikom opisanog postupka ne javi greška i aplikacije se uspešno pokrene, onda bi sve trebalo da je dobro uvezano.

48. **Pitanje:** Da li je spisak položenih ispita ustvari lista integer-a, koji su ustvari ocene? Kao i spisak nepoloženih ispita? Ili je ipak to ustvari lista entiteta ocena?

**Odgovor:** Lista referenci na instance klase Ocena, odnosno lista referenci na instance klase Predmet.

49. **Pitanje:** Da li datumi rođenja studenta i profesora mogu da budu tipa String ili je potrebno da budu Date pošto nije naglašeno?

Odgovor: Datum je u programskom jeziku Java predstavljen klasom Date.

50. **Pitanje:** Da li postoje specifični formati indeksa koje moramo ispoštovati, ili je u redu da koristimo svoje, koji su opisani u Help dijalogu? Na primer, zbog starih studenata da li je u redu da unos broja indeksa ograničimo na RAXX/XXXX ili moramo pokriti i slučaj za indekse koji su počinjali na slovo E pa brojevi?

**Odgovor:** Format indeksa je proizvoljan i ne treba ograničavati referenta studentske službe previše. Npr. na master studijama FTN-a indeksi imaju drugačiji format od navedenog.

51. **Pitanje:** Da li zvanje i titula profesora treba da budu tipa String ili je potrebno napraviti enum za neke određene vrednosti? Ili možemo na primer pri dodavanju profesora napraviti JComboBox da se samo neke određene vrednosti mogu odabrati?

**Odgovor:** Enumeracija.

Neke od naučnih titula: BSc, MSc, mr, dr, prof. dr...

Neka od naučnih zvanja: saradnik u nastavi, asistent, asistent sa doktoratom, docent, vanredni profesor, redovni profesor, profesor emeritus, ...

Na internetu možete potražiti detaljnu listu titula i zvanja na nivou visokoškolskih institucija Republika Srbije.

Takođe, sajt FTN je dobra lokacija od koje se potraga može započeti.

52. **Pitanje:** Kako bi trebalo da napišemo poruku za 19. tačku kod studenta 1? U specifikaciji stoji `#uklanjanje_predmeta_sa_profesora`, a u raspodeli `#uklanjanje_predmeta_profesoru`.

**Odgovor:** Hvala na ukazanom propustu. Ispravka je odrađena.

53. **Pitanje:** Postoji razlika u imenovanju jednog zadatka u specifikaciji. Konkretno zadatak za studenta 1, zadatak 13. U poglavlju 3.11 piše `#dodavanja_predmeta_studentu`, a u poglavlju 6 piše `#dodavanje_predmeta_studentu`. Kako bi bilo ispravno da nazovem commit za taj zadatak?

**Odgovor:** Hvala na ukazanom propustu. Ispravka je odrađena.

54. **Pitanje:** Nekonzistentija je u oznaci stavke 14. studenta 1.

**Odgovor:** Hvala na ukazanom propustu. Ispravka je odrađena.

55. **Pitanje:** Kog tipa treba da bude god studija obeležje iz entiteta Student?

**Odgovor:** Celobrojna vrednost iz odgovarajućeg opsega.

56. **Pitanje:** Da li mi nas dalji rad s projektom mozemo okaciti na granu domaci-2 ili da radimo na onoj main default grani?

**Odgovor:** Postavićemo obaveštenje o načinu predaje projekta. Najbolje da nastavite sa radom na master grani.

57. **Pitanje:** Greškom sam zaboravio da stavim # na par kommita pa me zanima da li to mogu da ostavim ili moram da menjam?

**Odgovor:** Trebalo bi izmeniti.

58. **Pitanje:** Da li je problem ukoliko imam vise od dva vezana ##? npr(#profesor #prikaz\_profesora, ...)?

**Odgovor:** Ukoliko ste u istom *commit*-u menjali više funkcionalnosti, potrebno je staviti sve oznake odrađenih funkcionalnosti. Baš kao što ste naveli.

59. **Pitanje:** Sta se očekuje od nas da napisemo za verziju aplikacije i biografiju kod stavke About u menu baru?

**Odgovor:** Možete pogledati konvencije verzioniranja. Dovoljno je staviti 0.0.2 (jer je drugi domaći zadatak). Kratka biografija o članovima tima.

60. **Pitanje:** Pojavio se neki čudan commit nakon pull/merge operacije i razrešavanja konflikta. Šta da radim?

**Odgovor:** To je očekivano ponašanje. Za više informacija konsultovati materijale sa vežbe, odnosno dokumentaciju git alata.

61. **Pitanje:** Ukoliko treba da implementiram brisanje profesora i zatreba mi copy konstruktor. Da li ja smijem da ga implementiram u klasi Profesor (posto je ta klasa napravljena od strane studenta 2), ili moram njega da čekam da on napravi taj konstruktor?

**Odgovor:** Trebalo bi da kolega implementira tu stavku. U dogovoru specificirate API koji kasnije obojica koriste.

62. **Pitanje:** Kod izmene predmeta, kad na dugme "+" dodamo profesora, nakon što se prikaže dijalog za odabir i nakon što se klikne na potvrdi, da li se profesor dodaje samo ako se klikne i na dugme potvrdi u dijalogu za izmenu predmeta (isto i za uklanjanje)?

**Odgovor:** Odluka je prepuštena na Vama. Neophodno je **napisati u Help-u** koji od dva metoda je implementiran. Oba predložena rešenja imaju smisla.

63. **Pitanje:** Da li kod dijaloga za izmene entiteta treba dugme *Potvrdi* da bude uvek omogućeno ili samo ako je izmena napravljena?

**Odgovor:** Inicijalno neka bude omogućen pritisak dugmeta, jer je pretpostavka da su prikazane informacije validne (inače sistem nije u konzistentnom stanju, što znači da se

negde potkrala greška). Prilikom izmene nekih polja, omogućiti validaciju identičnu kao što je opisano u sekciji 3.6 (dodavanje entiteta).

64. **Pitanje:** U specifikaciji projekta su navedena dva dodatna zadatka (lokalizacija i napredno pretraživanje). Da li možemo da biramo jedan od ta dva ili je potrebno oba da se urade da bi zamenili jedan zadatak na ispitu?

**Odgovor:** Ukoliko želite 12.5 bodova na praktičnom delu ispita, potrebno je uraditi oba dodatna zadatka.

65. **Pitanje:** Je li možemo svoje laptopove da ponesemo na odbranu?

**Odgovor:** Možete, ali nije zagarantovano da će Vam biti dozvoljeno da ih koristite.

66. **Pitanje:** Kod dijaloga za izmenu studenta, da li klikom na potvrdi treba da se ugasi ceo dijalog izmene, ili samo azuriramo promene i dopustimo dalje promene/pregled predmeta?

**Odgovor:** Odluka je prepuštena Vama. **Napisati u Help-u šta je odabrano.** Oba načina su validna.

67. **Pitanje:** Koleginica i ja smo se dogovorila da podelimo lokalizaciju tako da ona implementira pozivanje `getResourceBundle().getString` funkcija za sve dialoge a ja za sve klase view paketa. Da li je to ok, ili i tu moramo da poštujemo onu podelu iz specifikacije projekta?

**Odgovor:** Svako treba da odradi lokalizaciju komponenti koje je razvijao i da uz oznaku lokalizacije stavi oznaku funkcionalnosti na koju se komponenta odnosi. Podršku za jezik, kao i događaj koji će dovesti do aktivacije tog jezika dužni ste implementirati na način na koji je navedeno u sekciji 13.1.

68. **Pitanje:** Pošto bih ja probao da odradim dodatne zadatke a zbog sveopšte situacije je to malo teže, moje pitanje je da li postoji mogućnost da se osnovni projekat predaje do 10. A napredne stvari da imamo vremena do termina odbrane? Deluje mi kao da smo dosta kasnije dobili te zadatke nego što smo očekivali, možda mi samo nije sinula ideja i dalje ali deluje mi kao brdo posla makar za ovaj drugi dodatni zadatak.

**Odgovor:** Zbog raspoloživosti učionica nije moguće garantovati da će se odbrana projekta za sve studente odvijati u istom danu. Samim tim, pojedini studenti (koji brane ranije) imali bi manje vremena za izradu pomenutih funkcionalnosti. Proslediću kolegama Vašu molbu na razmatranje. O konačnoj odluci bićete blagovremeno obavešteni (putem specifikacije projekta, odnosno kanala predmeta).

69. **Pitanje:** Nama zadatke na ispitu mogu da zamene samo oba validno odradjena dodatna zadatak. Na primer ako bismo uradili samo lokalizaciju onda ne bismo mogli biti oslobođeni dela na ispitu?

**Odgovor:** Svaka od funkcionalnosti nosi određeni broj bodova (o tačnom broju bodova diskutovaćemo kolege i ja naknadno). Zajedno ove funkcionalnosti iznose najviše 12.5 bodova, što odgovara jednom praktičnom zadatku na ispitu. Dodatne funkcionalnosti biće Vam pregledane i ocenjene sa ***n bodova*** ( $n \in [0, 12.5]$ ). Ukoliko ste zadovoljni sa ***n bodova***, možete ih iskoristiti kao opciju da ne radite jedan zadatak. Ukoliko smatrate da znate za više, pristupate izradi zadatka, čime se ***n bodova osvojenih za zalaganje na izradi dodatnih funkcionalnosti zanemaruje***.

70. **Pitanje:** imam pitanje u vezi deserijalizacije. Radimo sa json fajlovima i iako ja iscitam iz fajla sve korektno, dobijao sam poruku u konzoli "Security framework of XStream not explicitly initialized, using predefined black list on your own risk.". Proverio sam liste u koje sam ucitao i sve su sadrzale ocekivani rezultat, i pored ove poruke. Onda sam dodao liniju `xs.addPermission(AnyTypePermission.ANY);` i ta poruka je nestala. Malo mi ovo deluje ko da sam samo zataskao neki potencijalni problem time sto sam dodao sve dozvole, umesto da ga resim, ali ne znam ni da li problem uopste postoji jer je deserijalizacija davala ocekivani rezultat i bez dodatih dozvola.

**Odgovor:** Problem se tiče bezbednosti (security), a to je tema nekog od kasnijih predmeta. Pomenuta greška ne bi trebalo da naruši sam proces (de)serijalizacije.

71. **Pitanje:** Da li dodavanje profesora na predmet treba da bude moguće samo prilikom izmene premeta, a ne i prilikom dodavanja?

**Odgovor:** Minimalno mora da bude prilikom izmjene, ukoliko želite možete dodati i prilikom dodavanja.

72. **Pitanje:** Imam pitanje šta treba da se desi kada se izvrši pretraga studenata? Da li je potrebno ponovo omogućiti pretragu i da li treba nakon završetka pretrage da se ponovo prikaže cela lista.

**Odgovor:** Kako vi smatrate da više ima smisla, može ponovna pretraga, može i pretraga originalne liste. Samo mora postojati način da se vrati na početno stanje (enter u prazno polje ili posebno polje/dugme za poništavanje pretrage).

73. **Pitanje:** Imam pitanje u vezi pretrage predmeta i profesora. Ja sam limitirao broj ukucanih reci na 3 reci i kod profesora i kod predmeta. Kod profesora referentu sam

dozvolio opciju da ukuca i broj licne karte, dok kod predmeta da li sve sto referent ukuca da se pretrazuje kao naziv predmeta. Je li je greska sto sam limitirao na tri reci svuda?

**Odgovor:** Kao što piše u specifikaciji, profesor se pretražuje po dva atributa, a predmet po jednom, U skladu sa tim trebate napraviti i validacije i ograničenja za pretragu.

74. **Pitanje:** Da li je napravljena štamparska greška u oznaci stavke 3.10 (#prikaz\_neplozenih\_ispita)

**Odgovor:** Hvala na ukazanom propustu. Ispravka je odrađena.

75. **Pitanje:** Da li pri unosu studenta sa istim indeksom treba da bude onemogućeno dugme potvrdi ili pri stiskom dugmeta potvrdi treba da izadje dijalog koji kaze da je nevalidan unos ?

**Odgovor:** Bilo bi poželjno onemogućiti pritisak pomenutog dugmeta.

76. **Pitanje:** Da li je napravljena štamparska greška u oznaci stavke 3.17 (#dodavanja\_predmeta\_profesoru)

**Odgovor:** Hvala na ukazanom propustu. Ispravka je odrađena.

77. **Pitanje:** Šta raditi u situaciji ukoliko kolega iz tima odustane od izrade projektnog zadatka?

**Odgovor:** Ukoliko ste do sada radili funkcionalnosti studenta 1, tada je potrebno implementirati funkcionalnosti koje se tiču entiteta student, ocena i predmet (možete ignorisati entitet profesor). Ukoliko ste pak do sada radili funkcionalnosti studenta 2, u tom slučaju je neophodno implementirati funkcionalnosti koje se tiču profesora i predmeta (možete ignorisati entitete student i ocena). Ukoliko se kolega pak predomisli i želi da nastavi sa radom, tada će se funkcionalnosti koje su njemu dodeljene, a implementirali ste ih vi, biti zanemarene. Preporuka je da stupite u kontakt sa kolegom i da od njega dobijete po potvrdu o odustanku. Takođe, oba studenta su dužna kontaktirati predmetnog asistenta.

78. **Pitanje:** Da li je u redu da se odabir studenta ili profesora koji će se dodati na predmet uradi preko JTable umjesto preko JListBox-a kako je u specifikaciji prikazano? Da li je problem da dodamo neke dodatne attribute u tu tabelu onda?

**Odgovor:** U redu je.

79. **Pitanje:** Zašto su funkcionalnosti studenta 1 i studenta 2 u “tolikoj meri” povezane i isprepletane.



**Odgovor:** Razlog zašto su funkcionalnosti povezane, jeste zato što je u pitanju timski projekat (a ne dva pojedinačna projekta). Jedna od bitnih stavki timskih projekata je saradnja kolega. Navedena raspodela funkcionalnosti potencira na podjednakom opterećenju studenata u timu.

80. **Pitanje:** Sortiranje Predmeta, da li je to potrebno samo u Tabu predmeti ili je potrebno na primer i unutar studenta gde prikazujemo predmete?

**Odgovor:** Možete, ako želite, u okviru svih tabela da implementirate sortiranje. Bodovaćemo samo sortiranje koje je navedeno u specifikaciji projekta, tj. sortiranja u okviru tabela studenti, profesori i predmeti (tab Studenti, tab Profesori i tab Predmeti).

81. **Pitanje:** Da li je kod dodavanja predmeta profesor obavezno polje ili može da ostane prazno, pa da se kasnije dodaje profesor preko izmene predmeta ?

**Odgovor:** Polje može inicijalno da ostane prazno, pa da se prilikom izmene predmeta doda profesor.

82. **Pitanje:** Da li je dodatno dugme na toolbar-u višak na slikama 19-22? Da li je kolona studenti sa ćelijama na kojima piše "Prikaži" na istim slikama isto višak?

**Odgovor:** Višak su.

83. **Pitanje:** Po specifikaciji, prilikom dodavanja predmeta profesoru, treba da se omogući dodavanje svih predmeta koje dati profesor u tom momentu ne predaje. Međutim, ukoliko sam dobro shvatio zadatak, jedan predmet može da predaje samo jedan profesor. Da li to znači da prilikom dodavanja predmeta profesoru treba da se obriše dodati predmet iz liste predmeta drugog profesora, ukoliko on predaje taj isti predmet?

**Odgovor:** Odluka je prepuštena Vama. Prilikom dodavanja predmeta profesoru možete prikazati samo predmete na kojima nema profesora ili sve predmete koje ne predaje pomenuti profesor, bez obzira da li ih predaje neko drugi. Ukoliko se odlučite za ovu drugu opciju, neophodno je da prikažete *confirm dialog* koji će od korisnika zatražiti potvrdu akcije, jer dolazi do implicitnog uklanjanja profesora sa predmeta.

84. **Pitanje:** Da li prilikom brisanja predmeta, predmet treba izbrisati i iz položenih predmeta studenata? Pretpostavljam da iz nepoloženih treba da se izbriše.

**Odgovor:** Imalo bi smisla.

85. **Pitanje:** U Vašoj poslednjoj objavi, spomenuli ste da je hardcode dodavanje podataka u tabelu dozvoljeno isključivo ako nemamo serijalizaciju i deserijalizaciju. Kolega i ja imamo funkcionalnu serijalizaciju i deserijalizaciju, ali planiramo namestiti da aplikacija

automatski generise default podatke iz tabele ukoliko odgovarajuć fajl sa bazom nije pronađen u direktorijumu, za svaki slučaj. Da li je ovo OK?

**Odgovor:** Studenti kojima serijalizacija/deserijalizacija radi treba da kreiraju odgovarajuću hijerarhiju datoteka na osnovu kojih će učitati test podatke u aplikaciju.

Način na koji će praviti te datoteke:

- ručno (u tekstualnom editoru)
- unos i povezivanje entiteta upotrebom sopstvene aplikacije i serijalizacijom stanja iste.

86. Pitanje: