

Specifikacija projekta softvera za studentsku službu - dopunjeno

Softver za studentsku službu se sastoji od serverske strane koja obezbeđuje skladištenje podataka i poslovnu logiku rada sa podacima na jednom fakultetu i klientske strane koju koristi studentska služba.

1. Model podataka

Osnovni Entiteti u sistemu su: Student, Predmet, Studijski program, Nastavnik, Ispit, Ispitni rok, Upis godine, Obnova godine, Uplata. Pored ovih entiteta postoje i drugi sa njima povezani. Svi entiteti u sistemu treba da imaju identifikator (id) koji se automatski dodeljuje od strane baze podataka.

Entitet Student

Podaci o studentu se samo jednom unose u sistem, a taj student može biti tokom vremena upisan na različite studijske programe (na primer prvo na RI, pa u drugoj godini pređe na SI, zatim posle završetka osnovnih studija se upiše na master studije) i na svim tim studijskim programima ima različiti broj indeksa.

Podaci o studentu koji se unose u sistem: ime, prezime, srednje ime, jmbg, datum rođenja, mesto rođenja, država rođenja, državljanstvo, nacionalnost, pol, adresa prebivališta (mesto, ulica i broj), broj telefona, email (fakultetski i jedan privatni), broj lične karte i od koga je izdata, završena srednja škola (treba da postoji šifarnik srednjih škola u koji mogu da se dodaju škole, i za studenta se bira škola iz šifarnika, za svaku srednju školu se pamti njen naziv, mesto i vrsta - gimnazija, srednja stručna škola,...), uspeh iz srednje škole i sa prijemnog (ako se student upisao na prvu godinu osnovnih studija), prelaz sa druge visokoškolske ustanove (ako je student prešao, čuva se naziv visokoškolske ustanove sa koje je prešao, takođe kao šifarnik, slično kao za srednje škole).

Studentski indeks ima godinu upisa, broj i studijski program. Student može da ima više indeksa, neki od njih mogu biti aktivni, a neki ne (na primer kada se prebaci sa jednog na drugi program, onaj sa kog se prebacio više nije aktivan, ali se ne briše iz sistema). Pamti se i datum od kog je indeks aktivan.

Entiteti Predmet i Studijski program

Studijski program sadrži oznaku (RI, RN, SI...), naziv, godinu kada je akreditovan, naziva zvanja koje se dobija završetkom studijskog program, trajanje u semestrima i vrstu studija koja se bira iz šifarnika. Za svaku vrstu studija čuva se skraćeni i pun naziv, na primer OAS - osnovne akademske studije, MAS - master akademske studije i slično.

Indeksi studenata su vezani za studijski program, student je za studijski program vezan preko indeksa, ne direktno.

Studijski program sadrži predmete, za svaki predmet se čuva njegova šifra, naziv, opis, broj ESPB poena, semestar u kom se sluša po studijskom programu, fond časova predavanja i vežbi.

Entitet Nastavnik

Za nastavnika se čuva ime, prezime, srednje ime, email, podaci o obrazovanju (visokoškolske ustanove na kojoj je završio sve nivoe studija) i svaki nastavnik u toku svog rada može imati različita saradnička i nastavnička zvanja (prvo asistent, zatim docent, vanredni profesor, ...). Za nastavnika treba upamtiti sva zvanja u kojima je bio i to za svako zvanje se čuva datum izbora, uža naučna oblast i sam naziv zvanja (asistent, docent,...).

Entiteti Ispit i Ispitni rok

Za ispitni rok se pamte datumi početka i završetka i ispiti koji se u njemu održavaju. Za svaki ispit se čuva datum održavanja, predmet na koji se ispit odnosi, nastavnik koji drži ispit, vreme početka i da li je ispit zaključen (uneti poeni od strane nastavnika).

Školska godina i veze nastavnika, predmeta i studenta, ispita

U sistemu treba da postoji entitet školska godina, od kojih je jedna školska godina uvek aktivna. Školskom godinom će biti određene veze između nastavnika i predmeta koje drže, kao i studenata i predmeta koje slušaju i polažu.

Jedan nastavnik predaje više predmeta u jednoj školskoj godini, a jedan predmet može da predaje više nastavnika. Student u školskoj godini sluša predmete kod određenih nastavnika, svaki predmet kod jednog nastavnika. Za predmet se definišu predispitne obaveze u školskoj godini. Svaka predispitna obaveza za predmet sadži vrstu (test, kolokvijum) i maksimalan broj

poena koje nosi ta **obaveza**. Student u jednoj **školskoj** godini osvaja određeni broj **predispitnih poena** na predispitnim obavezama predmeta koje sluša. Ispitni rokovi su takođe **vezani za školsku godinu**.

Aktivnosti studenta

Student prijavljuje ispite i izlazi na njih, a može i da prijavi, a da ne izađe. Za prijavu se pamti datum prijave i ispit koji se prijavljuje. Za svaki izlazak na ispit čuva se broj osvojenih poena (povlače se osvojeni poeni na predispitnim obavezama i unose se poeni sa ispita), napomena i informacija da li student poništava ispit. **Ukoliko student izađe na ispit i položi ga treba da mu se sačuva predmet kao položen** (predmet koji je polagao na ispitu) i ocena koja se generiše na osnovu osvojenih poena. **Student može imati položen predmet koji nije polagao na ispitu nego mu je predmet priznat sa druge visokoškolske ustanove i to treba prikazati u modelu.**

U sistemu treba čuvati tok studija za studenta na studijskom programu, tu spadaju upis godine (čuva se godina koja se upisuje, preneti predmeti), obnova godine (čuva se godina koja se obnavlja, i predmeti koje upisuje u obnovljenu godinu), za sve ove aktivnosti pamte se i datum i napomena.

2. Operacije koje je potrebno podržati

Pored osnovnih CRUD operacija u koje spadaju snimanje entiteta, preuzimanje na osnovu ključa, brisanje, potrebno je podržati i sledeće funkcionalnosti (grupisano po entitetima).

Student

- selekcija studenta (njegovih ličnih podataka) preko broja indeksa
- selekcija svih položenih ispita za broj indeksa studenta, paginirano
- selekcija svih nepoloženih ispita za broj indeksa studenta, paginirano
- pregled svih upisanih godina za broj indeksa
- upis studenta na godinu (dodati novi upis za studenta i predmete koje sluša, a koji će inicijalno biti nepoloženi)
- pregled obnovljenih godina za broj indeksa
- obnova godine za studenta gde je potrebno omogućiti da se, pored nepoloženih, izaberu predmeti iz naredne godine. Maksimalni zbir ESPB poena može biti 60. Dodaje se nova obnova godine za studenta i predmeti koje sluša, a koji će inicijalno biti nepoloženi.

- dodavanje nove uplate (čuva se datum uplate, iznos u dinarima i srednji kurs). Trenutni srednji kurs ne treba da ima predefinisane vrednosti, nego ga treba dohvatiti API pozivom: <https://kurs.resenje.org/api/v1/currencies/eur/rates/today>
- selekcija preostalog iznosa za uplatu u evrima i dinarima. Iznos školarine je predefinisana vrednost od 3000e
- selekcija studenata na osnovu imena i/ili prezimena (može samo ime, ili samo prezime ili oba da se unesu), paginirano
- selekcija svih upisanih studenata koji su završili određenu srednju školu

Studijski program i predmet

- spisak predmeta na studijskom programu
- spisak svih predmeta sa svim podacima o predmetu
- dodavanje novog predmeta na studijski program, zadaje se šifra predmeta koja mora biti jedinstvena
- prosečna ocena studenata na predmetu za raspon godina (na primer zadaje se godine 2015-2018 i vraća se prosečna ocena svih studenata koji su položili predmet u tom periodu)

Ispit

- svi prijavljeni studenti za ispit
- prosečna ocena na ispitu
- prijava ispita za izabrani ispitni rok i predmet koji student sluša
- dodavanje izlaska na ispit sa osvojenim poenima i čuvanje predmeta u položenima ako je zbir predispitnih obaveza i rezultata na ispitu minimalno 51 poen
- rezultati ispita u vidu sortiranog spiska studenata (sortirati prvo po stud programu, zatim po godini upisa, pa po broju) sa ukupnim brojem poena na ispitu (predispitne obaveze plus poeni sa ispita)
- selekcija ostvarenih poena na predispitnim obavezama za studenta na određenom predmetu u školskoj godini

- za datog studenta i predmet vratiti koliko je puta student polagao predmet (izašao na ispit)

3. Arhitektura

U implementaciji aplikacije je obavezno ispoštovati Layered arhitekturu koja podrazumeva da postoje tri osnovna sloja: Controller (API sloj), Service (servisni sloj sa biznis logikom) i Repository (Data Base sloj za rad sa podacima). Controller služi da pozove servisni sloj u kome treba da bude implementirana sva biznis logika aplikacije. Servis treba da pozove Repository koji će dohvatiti ili izmeniti podatke u bazi. Potrebno je koristiti DTO (Data Transfer Object) za dvosmernu komunikaciju izmedju Controller-a i Servisnog sloja. Dakle, potrebno je implementirati Entity Mapper-e kao zasebne komponente, koji će služiti za mapiranje Entity-ja na DTO i obrnuto. Potrebno je da se baza napuni inicijalnim podacima preko Command Line Runner-a. Za svaku operaciju je potrebno da se pripreme testovi kojima će se na odbrani demonstrirati rad tih operacija.

4. Tehnički zahtevi

Za implementaciju projekta potrebno je koristiti sledeće tehnologije:

- Spring boot
- JPA Hibernate za perzistenciju podataka i upite nad bazom
- Postman (ili sličan alat) za testiranje REST API-ja

5. Raspodela poena

Operacije nad studentima – 17p

Operacije nad predmetima i studijskim programima – 5p

Operacije nad ispitima i ispitnim rokovima - 8p

6. Rok za predaju i odbranu

Projekat se radi u paru i rok za završetak je termin odbrane. Projekat se brani u prvoj kolokvijumskoj nedelji, donosi se na odbranu i pokazuje ispitivaču, nije potrebno slati ga ranije. Raspored odbrana će biti poslat na mejling listu i objavljen na materijalima. Studenti mogu, ukoliko to žele, da projekat brane i ranije u dogovoru sa asistentom.

