

## Završni test

Korišćenjem radnih okvira Spring Boot, AngularJS i Bootstrap razviti Web aplikaciju za platni promet banke.

0.1 Obezbediti kreiranje minimalnog skupa test podataka (bilo iz koda, bilo SQL srkiptom)

1.1 Aplikacija treba da obezbedi rad sa sledećim entitetima:

### Banka:

- Id - identifikator
- Naziv - tekstualna vrednost (jedinствена, obavezna)
- Sredstva banke - numerička vrednost

### Tip računa:

- Id - identifikator
- Naziv – tekstualna vrednost (obavezna, jedinstvena)
- Provizija (procenat provizije na prenos novca) - numerička vrednost
- Banka - veza sa instancom klase banka (U banci može biti više tipova računa, dok jedan tip računa može biti samo u jednoj banci). Veza je dvosmerna.

### Račun:

- Id – Identifikator
- Ime i prezime - tekstualna vrednost (obavezna)
- JMBG - tekstualna vrednost
- Broj računa - tekstualna vrednost
- Stanje na računu - numerička vrednost
- Tip računa - veza sa instancom klase tip računa (Više računa može biti istog tipa, dok jedan račun može biti jednog tipa). Veza je dvosmerna.
- Banka - veza sa instancom klase banka (U banci može biti više računa, dok jedan račun može biti samo u jednoj banci). Veza je dvosmerna.

1.2 Pomoću radnog okvira Spring Boot implementirati sledeći REST API:

- GET /api/banke - preuzimanje svih banaka
- GET /api/banke/{id}/tipovi-racuna - preuzimanje svih tipova računa u nekoj banci
- GET /api/racuni - preuzimanje svih računa (**paginirano**)
- GET /api/racuni/{id} - preuzimanje računa po zadatom id-u
- POST /api/racuni - dodavanje novog računa
- PUT /api/racuni/{id} - izmena postojećeg računa
- DELETE /api/racuni/{id} – brisanje postojećeg računa

- Voditi računa o tome da, koliko je to moguće, REST metode vraćaju preporučeni HTTP status za odgovarajući rezultat neke operacije.

1.3 Na nivou API-ja validirati sledeće stavke:

- “JMBG” u okviru računa je tekst dužine 13 karaktera
- “Broj računa” je neprazan tekst

# Računi u bankama

Ime i prezime

JMBG

Banka

Broj računa





Tip računa

Dodaj

Prenos

Prethodna

Sledeća

Ime i prezime	JMBG	Stanje	Broj računa	Tip računa	Banka	Akcije
Marko Marković	1112967820010	1500 din.	340-12332111	Standard	Erste	 
Marija Markić	1312967825011	2500 din.	150-12342111	Premium	AIK	 

Slika 1.

## Nalog za prenos

Broj računa uplatioca

Broj računa primaoca

Iznos

Prenesi

Slika 2.

2.1) Obezbediti unos novog računa na **stranici za unos i prikaz**. Nakon unosa vrednosti u polja prikazana na slici 1 i klika na dugme dodaj, račun se preko API-ja dodaje u aplikaciju i potom se stranica osvežava. **Stanje na računu se ne unosi prilikom kreiranja računa.**

Voditi računa o tome kako se prilikom dodavanja bira tip računa. Drop-down meni za odabir tipa računa je prvo skriven. Kada korisnik odabere banku (na promenu odabira možete reagovati direktivom **ng-change**), tek tada se sa back end-a dobavljaju podaci o tipu računa, popunjavaju drop-down i postaju vidljivi korisniku kako bi mogao da odabere odgovarajući.

2.2) Obezbediti izmenu računa na **zasebnoj stranici**. Nakon klika na dugme izmeni, podaci o računu se prikazuju u zasebnoj stranici koja omogućava čuvanje izmena. Ukoliko korisnik odabere da sačuva podatke, izmenjeni podaci se čuvaju u aplikaciji i prelazi se na stranicu za unos i prikaz računa. Dizajn zasebne stranice za izmenu stanice prepušten je polazniku kursa. **Nije potrebno omogućiti promenu banke računu, niti tip računa. Stanje na računu se je moguće menjati prilikom izmene.**

2.3) Obezbediti pretragu računa, preko forme za filtriranje. Pretraga je moguća po JMBG i banci (preko drop-down-a, gde se vidi ime banke, a u pozadini je njen id). Pronalaze se i prikazuju računi koji zadovoljavaju sve kriterijume. Ukoliko korisnik neko polje ne unese, vrednost tog polja se ignoriše u pretrazi. Filtriranje se vrši na *back-end* delu aplikacije. **Pored forme za unos je dugme koje briše unete parametre pretrage, i osvežava tabelu.**

2.4) Omogućiti brisanje računa pritiskom na dugme **Obriši**. Nakon što je korisnik pritisao dugme za brisanje, potrebno je osvežiti tabelu sa podacima. **Brisanje računa je moguće samo ukoliko je stanje na računu 0 (u suprotnom back-end vraća 403 Forbidden, a front-end prikazuje poruku korisniku).**

2.5) Obezbediti paginirani prikaz podataka. Dugme *Prethodno* i dugme *Sledeće* iznad tabele za prikaz računa omogućuju promenu stranice. Ukoliko se korisnik nalazi na prvoj stranici onemogućiti dugme prethodno, a ukoliko se nalazi na poslednjoj onemogućiti dugme sledeće. Paginacija se vrši na back-end delu aplikacije.

3. Implementirati prenos sredstava sa računa na račun:

- Klikom na dugme "Nalog za prenos", prelazi se na novu stranicu na kojoj se nalazi forma za nalog za prenos (slika 2.) **Nije potrebno kreirati novi entitet radi naloga za prenos.** Da bi se kreirao nalog za prenos, potrebno je uneti:
  - ◆ Broj računa **sa** kojeg se prenose sredstva (uplatioca)
  - ◆ Broj računa **na** koji se prenose sredstva (primaoca)
  - ◆ Iznos
- Izvršavanje naloga za prenos, tako što se sredstva prebacuju sa jednog računa na drugi ako za to postoje uslovi. Procenat od iznosa - provizija, koja zavisi od tipa računa sa kojeg se šalju sredstva, se prebacuje u polje "Sredstva banke" banke u kojoj je navedeni račun uplatioca. Maksimalna vrednost provizije je 1000 dinara - što znači da procenat zavisao od tipa računa prestaje da bude relevantan u trenutku kada je kalkulisana vrednost veća od te cifre.