Završni test

Korišćenjem radnih okvira Spring Boot, AngularJS i Bootstrap razviti Web aplikaciju za komunikaciju kućnih saveta.

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

1.1 Aplikacija treba da obezbedi rad sa sledećim entitetima:

Zgrada:

Id - identifikator
Adresa - tekstualna vrednost (jedinstvena, obavezna)
Predsednik KS – tekstualna vrednost (obavezna)
Broj stanova - celobrojna vrednost
Broj stanara - celobrojna vrednost (obavezna)

Poruka:

Id - identifikator
Naslov - tekstualna vrednost (obavezna)
Tip - tekstualna vrednost
Potreban procenat - numerička vrednost
Prihvaćen - vrednost logičkog tipa - **(ne unosi korisnik, izračunava se)**
Opis - tekstualna vrednost (obavezna)
Zgrada - veza sa instancom klase zgrada (poruka može biti upućena samo stanarima jedne zgrade, a može im biti biti upućeno više poruka). Veza je dvosmerna.

Glas:

Id – Identifikator
Predlog podržan - tekstualna vrednost
Poruka - veza sa instancom klase poruka (jedan glas može biti vezan samo za jednu poruku, a za jednu poruku može biti vezano više glasova). Veza je dvosmerna.

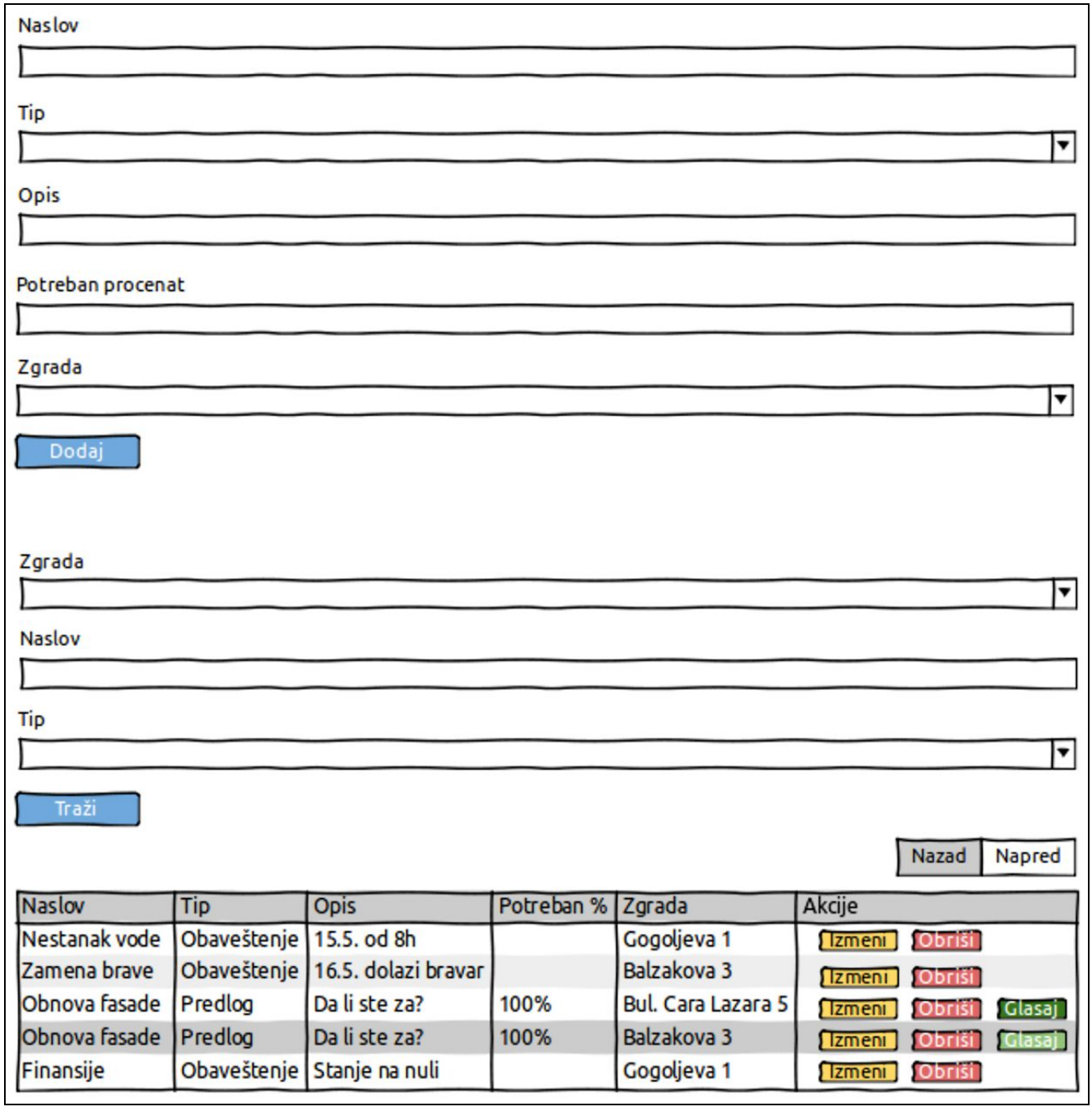
Aplikacija bi trebalo da vodi računa o komunikaciji kućnih saveta. Kućnom savetu se upućuje poruka, koja po tipu može biti "***obaveštenje***" ili "***predlog***". Tip poruke je **tekstualna vrednost**, NE novi entitet, niti enumeracija!
Kako se za predloge glasa, entitet poruka ima i polje *potreban procenat* (stanara te zgrade potreban da bi predlog bio prihvaćen) i polje *prihvaćen,* koji je inicijalno na *false*, a kada navedeni procenat stanara da pozitivan glas, postane *true.*

1.2 Pomoću radnog okvira Spring Boot implementirati sledeći REST API:

GET /api/zgrade - preuzimanje svih zgrada
GET /api/poruke - preuzimanje svih poruka **(paginirano)**
GET /api/poruke/{id} - preuzimanje poruke po zadatom id-u
POST /api/poruke - dodavanje nove poruke
PUT /api/poruke/{id} - izmena postojeće poruke
DELETE /api/poruke/{id} – brisanje postojeće poruke
GET /api/zgrade/{id}/poruke – dobavljanje poruka jedne zgrade

- 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:
Da je *naslov* unet - neprazan tekst
Da je *potreban procenat* minimalno 0, a maksimalno 100



Slika 1.

2.1) Obezbediti unos nove poruke na stranici za **unos, pretragu i prikaz** (slika 1). Nakon unosa vrednosti u polja prikazana na slici 1 i klika na dugme dodaj, poruka se preko API-ja dodaje u aplikaciju i potom se podaci u tabeli osvežavaju i polja u formi prazne. **Polje *"Prihvaćen"* ne unosi korisnik.**

2.1a) U okviru forme za unos, obezbediti da se polje *"Potreban procenat"* vidi kada je odabran tip poruke "*Predlog*", a ne vidi kada je odabran tip poruke "*Obaveštenje*" (videti sliku 2).

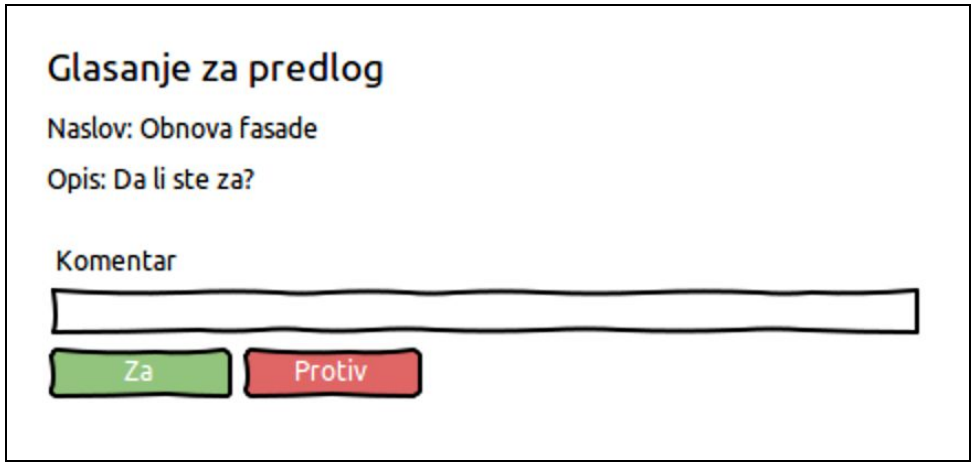
2.2) Obezbediti izmenu poruke na **zasebnoj stranici**. Nakon klika na dugme izmeni, podaci o poruci se prikazuju u zasebnoj stranici koja omogućava čuvanje izmena. Ukoliko korisnik odabere da sačuva podatke, izmenjeni podaci se čuvaju u aplikaciji i prelazi se na stranicu za unos i prikaz poruke. Treba omogućiti i promenu zgrade poruci. Dizajn zasebne stranice za izmenu poruke prepušten je polazniku kursa. **Polje *"prihvaćen"* se ne može promeniti.**

2.3) Obezbediti pretragu poruka preko forme za filtriranje (slika 1). Prilikom pretrage korisnik može da odabere *zgradu* (preko opcije za selekciju, *dropdown*), naslov ili tip poruke (takođe preko opcije za selekciju, *dropdown*). Pronalaze se i prikazuju poruke koje zadovoljavaju sve kriterijume. Ukoliko korisnik neko polje ne unese, vrednost tog polja se ignoriše u pretrazi. Filtriranje se vrši na *back-end* delu aplikacije.

2.4) Obezbediti paginirani prikaz podataka. Dugme prethodno i dugme sledeće iznad tabele za prikaz poruke (slika 1) 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 glasanje stanara za predlog.
   - ➔ Klikom na dugme "Glasaj" (slika 1), prelazi se na novu stranicu za glasanje, sa formom za unos glasa(slika 3). Kada korisnik unese podatke, i klikne na željeno dugme, podaci se odgovarajućom API metodom. Logika za glasanje implementira se na nivou servisa na back-end-u:
     - ◆ Ako je za predlog glasao dovoljan broj stanara (što zavisi od potrebnog procenta), vrednost polja "***prihvaćen***" za poruku postaje *true*.
     - ◆ Ne može biti veći broj glasova od broja stanara

Slika 2. Dva izgleda forme za dodavanje, s obzirom na polje “**Tip**”



Slika 3