

Portal za objave vesti

Potrebno je implementirati aplikaciju za objave vesti (u daljem tekstu PostsPortal).

Aplikacija PostsPortal prepoznaje tri tipa korisnika:

- administrator
- objavljiivač
- komentator

Administrator aplikacije ima na raspolaganju sledeću funkcionalnost:

[A1] Prijava i odjava sa sistema.

[A2] Ažuriranje¹ liste objava. Upload slika vezanih za objavu, kao i vezivanje tagova za objave (ključnih reči) je potrebno podržati. Sortiranje liste objava po datumu objave, broju komentara i po popularnosti (broju like-ova/dislike-ova). Filtriranje liste objava po autoru i tagovima.

[A3] Ažuriranje liste komentara. Sortiranje liste komentara po datumu i popularnosti.

[A4] Promena lozinke.

[A5] Promena ličnih podataka. Upload slike je potrebno podržati.

[A6] Ažuriranje liste korisnika (administratora i objavljiivača, komentator se sam registruje). Upload slika korisnika je potrebno podržati.

Objavljiivač ima na raspolaganju sledeću funkcionalnost:

[O1] Prijava i odjava sa sistema.

[O2] Ažuriranje liste objava objavljiivača (samo svoje objave može ažurirati – menjati i brisati, a može i dodavati nove objave). Upload slika vezanih za objavu, kao i vezivanje tagova za objave (ključnih reči) je potrebno podržati. Sortiranje liste objava po datumu objave, broju komentara i po popularnosti (broju like-ova/dislike-ova).

[O3] Pregled postojećih objava i komentarisanje - dodavanje komentara, like-ovanje i dislike-ovanje komentara i objava. Filtriranje liste objava po autoru i tagovima. Sortiranje liste komentara po datumu i popularnosti.

[O4] Promena lozinke.

[O5] Promena ličnih podataka. Upload slike je potrebno podržati.

Komentator ima na raspolaganju sledeću funkcionalnost:

[K0] Registracija – korisnik se sam registruje, nakon toga se može prijaviti na sistem i učestvovati u komentarisanju. Bez prijave na sistem moguće je samo pregledati objave i komentare.

[K1] Prijava i odjava sa sistema.

[K2] Pregled postojećih objava i komentarisanje - dodavanje komentara, like-ovanje i dislike-ovanje komentara i objava. Filtriranje liste objava po autoru i tagovima. Sortiranje liste komentara po datumu i popularnosti.

¹ Pod ažuriranjem se podrazumevaju aktivnosti vezane za dodavanje, promenu i uklanjanje odgovarajućih pojava entiteta. Većinu informacionih sistema karakteriše neograničen period čuvanja podataka te se aktivnost uklanjanja odgovarajućih pojava entiteta retko koristi.

[K3] Promena lozinke.

[K4] Promena ličnih podataka. Upload slike je potrebno podržati.

Za implementaciju aplikacije iskoristiti sledeće softverske pakete:

- Spring framework
- Apache Tomcat (ne mora biti posebno integrisan, može Spring Boot)
- MySQL i
- potrebne Java alate i biblioteke.

Podatke kojima upravlja aplikacija organizovati uz oslonac na SUBP².

Nefunkcionalni zahtevi

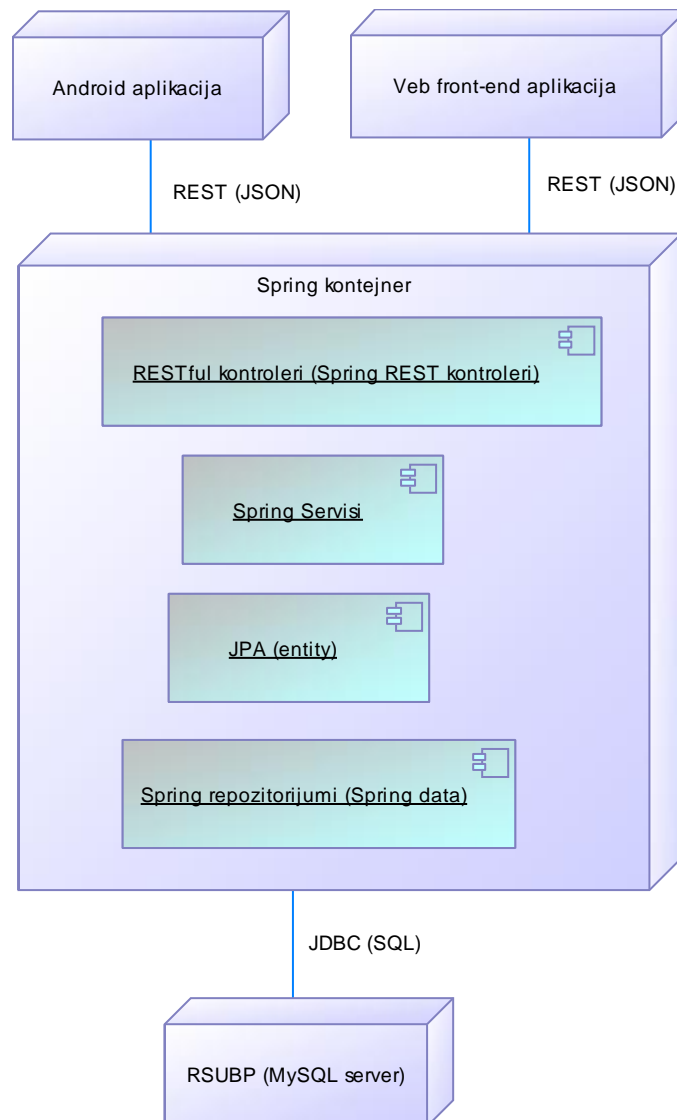
Podržati autentifikaciju korisnika upotrebom email-a (korisničkog imena) i lozinke i autorizaciju korisnika upotrebom mehanizma tokena.

Beležiti poruke o važnim događajim koji su nastali prilikom izvršavanja veb aplikacije.

² Skraćeno od Sistem za Upravljanje Bazama Podataka.

Arhitektura aplikacije

Veb aplikacija raspoređena je na tri računara: Veb brauzer ili android aplikacija, Spring kontejner (u Tomcat serveru ili pokrenut pomoću Spring Boot) i SUBP. Dijagram rasporeda prikazan je na slici 1.

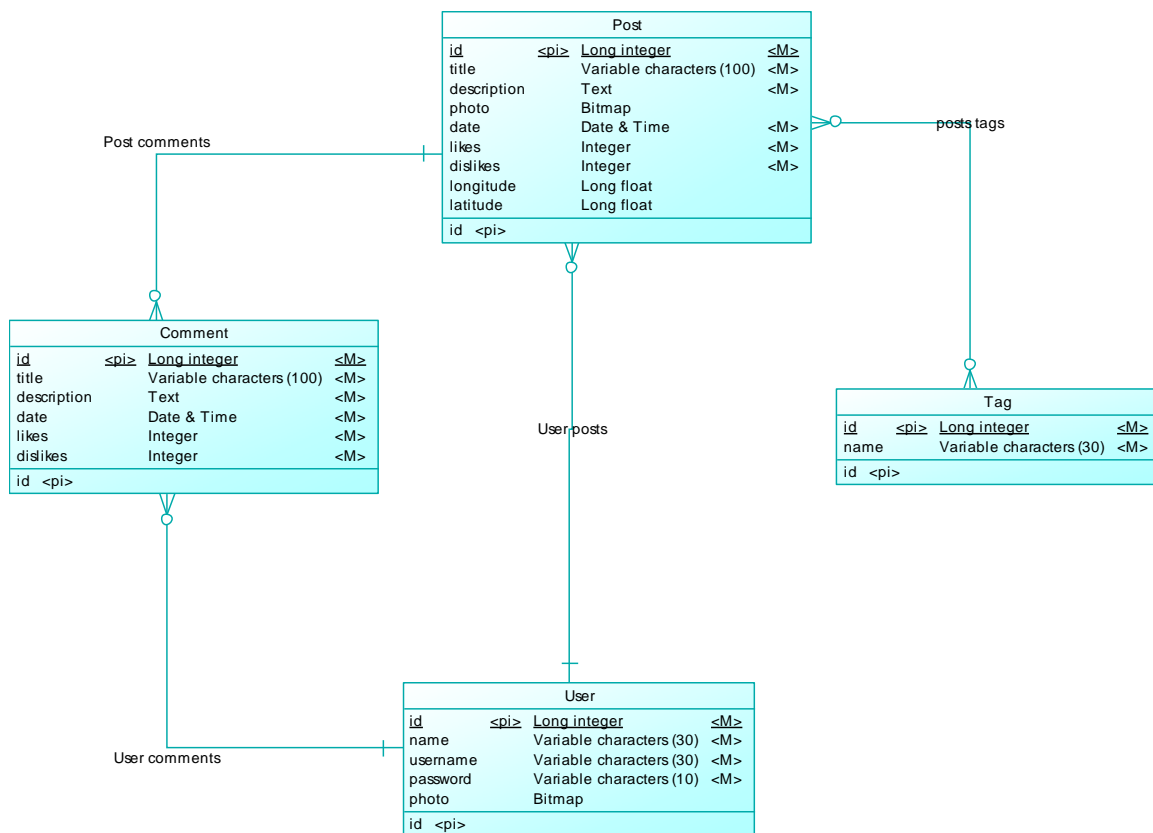


Slika 1: Arhitektura aplikacije – dijagram rasporeda

Back-end aplikaciju implementirati upotrebom Spring framework-a [1], dozvoljeno je koristiti i Spring Boot [2]. Front-end aplikacija mora postojati i komunicira sa back-end aplikacijom putem ekspozovanih RESTful servisa. Dozvoljeno je front-end aplikaciju implementirati kao Android aplikaciju ili kao veb aplikaciju upotrebom HTML, CSS, JavaScript, JQuery, AngularJS, ili React tehnologije. Kao SUBP koristiti MySQL Server [3] ili neki drugi relacioni SUBP. Za beleženje poruka koristiti log4j API [4]. Za izgradnju softvera koristiti Apache Ant [5] ili neki drugi alat, a dozvoljeno je i koristiti Spring Boot i na taj način konfigurisati i pokretati aplikaciju.

Model podataka

Na slici 2 prikazan je model podataka PostsPortal aplikacije. Entitet Post opisuje objavu koja se nalaze u bazi podataka aplikacije. Svaka objava mora biti opisan sledećim obeležjima: id, title, description, datum postavke (date), broj like-ova i dislike-ova (na početku je 0), korisnik (entitet User) koji je objavljivač. Pored ovih obeležja objava ima opciono (ne obavezno) obeležja: photo (BLOB polje u bazi podataka), longitude i latitude (koordinate vezane za objavu). Entitet User opisuje korisnike aplikacije. Svaki korisnik mora biti opisan sledećim obeležjima: id, name, username (preporuka je da to bude email), password, a ima i opciono (ne obavezno) obeležje photo. Entitet Tag predstavlja šifarnik ključnih reči, i mora biti opisani sledećim obeležjima: id, name. Entitet Comment predstavlja komentar objave i mora biti opisan sledećim obeležjima: id, title, description, datum (date), broj like-ova i dislike-ova (na početku je 0).



Slika 2: Model podataka

Napomene

1. Napraviti skript za unos inicijalnog skupa podataka, uključujući i tagove (entitet Tag).
2. Studenti ne smeju svojevolejno da menjaju model baze podataka.
3. U delu Opis funkcionalnosti zahtevi koji su obojeni **žutom** bojom označavaju minimalne zahteve za ocenu 6, zahtevi obojeni **zelenom** bojom označavaju minimalne zahteve za ocenu 8, a zahtevi obojeni **plavom** bojom označavaju minimalne zahteve za ocenu 10.

Reference

1. Spring Framework, <https://spring.io/>
2. Spring Boot, <https://projects.spring.io/spring-boot/>
3. MySQL, <https://www.mysql.com/>
4. Log4j, <https://logging.apache.org/log4j/2.x/>
5. Apache Ant, <http://ant.apache.org/>