

RedditClone

Potrebno je implementirati aplikaciju za podršku pregledanja, objave i diskusije sadržaja (u daljem tekstu RedditClone) po uzoru na sajt [Reddit](#).

Korisnik aplikacije ima na raspolaganju sledeće funkcionalnosti:

- [K1] Registracija korisnika. Administrator je predefinisani korisnik u sistemu.
- [K2] Prijava i odjava sa sistema. Na formi za prijavljivanje na sistem postoji link za prelaz na registraciju korisnika. Kada se korisnik uspešno prijavi na aplikaciju omogućiti korisniku da se odjavi. Bez prijave na sistem moguće je samo pregledati objave i komentare unutar neke zajednice.
- [K3] Ažuriranje zajednice. Bilo koji registrovani korisnik može da kreira zajednicu i automatski postaje moderator date zajednice. Jedino administrator ima pravo da suspenduje (logički obriše) zajednicu. Zajednica može da sadrži PDF u kojem je opis zajednice.
- [K4] Ažuriranje¹ liste objava. Na stranici odabrane zajednice (eng. *community*) nalazi se ista objava. Objava može da sadrži PDF. Obavezna polja objave su naslov i tekst. Objava može da sadrži tag (eng. *flair*).
- [K5] Ažuriranje komentara. Moguće je odgovoriti (eng. *reply*) na komentar. Moguće je proizvoljan broj odgovora na komentar (*reply* na *reply* na *reply*...).
- [K6] Upvote i downvote. Moguće je oceniti objavu ili komentar pozitivno (eng. *upvote*) ili negativno (eng. *downvote*). Pored objave ili komentara vidi se ukupna karma. Karma predstavlja razliku između pozitivnih i negativnih ocena ($upvoteCount - downvoteCount$). Svaka objava ili komentar biva automatski *upvote*-ovana od strane korisnika kada je inicijalno kreira.
- [K7] Pretraga zajednica:
 - ● Pretraživanje zajednica po nazivu
 - ● Pretraživanje zajednica po opisu
 - ● Pretraživanje zajednica po opisu iz zakačenog PDF fajla
 - ● Pretraživanje zajednica po opisu pravila
 - ● Pretraživanje zajednica po opsegu broja objava (od - do), gde može biti zadata donja i/ili gornja granica opsega
 - ● Pretraživanje zajednica po prosečnoj karmi² zajednice u opsegu (od - do), gde može biti zadata donja i/ili gornja granica opsega
 - ● Kombinacija prethodnih parametara pretrage (BooleanQuery, omogućiti AND i OR operator između polja)
 - ● Pretprocesirati upit, tako da bude nezavisan od velikog i malog slova, kao i ćiriličnog ili latiničnog pisma
 - ● Obezbediti podršku u poljima forme za unos PhraseQuery i FuzzyQuery

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.

² Prosečna karma se formira kao količnik sume karmi svih objava zajednice i ukupnog broja objava u zajednici

- Prilikom prikaza rezultata prikazati naziv zajednice, broj objava, prosečnu karmu i dinamički sažetak (Highlighter)

[K8] Pretraga objava:

- Pretraživanje objava po naslovu
- Pretraživanje objava po tekstu
- Pretraživanje objava po tekstu iz zakačenog PDF fajla
- Pretraživanje objava po tekstovima u komentarima
- Pretraživanje objava po opsegu karme (od - do), gde može biti zadata donja i/ili gornja granica opsega
- Kombinacija prethodnih parametara pretrage (BooleanQuery, omogućiti AND i OR operator između polja)
- Pretprocesirati upit, tako da bude nezavisan od velikog i malog slova, kao i ćiriličnog i latiničnog pisma
- Obezbediti podršku u poljima forme za unos PhraseQuery i FuzzyQuery
- Pretraživanje objava koje imaju broj komentara u opsegu (od-do), gde može biti zadata donja i/ili gornja granica opsega
- Pretraživanje objava po tagu (eng. *flair*)
- Prilikom prikaza rezultata prikazati naslov objave, tekst objave i dinamički sažetak (Highlighter)

[K9] Prijavljivanje neprikladnog komentara ili objave. Svaki korisnik ima pravo da prijavi sadržaj kao neprikladan. Prijavljen sadržaj se pregleda od strane moderatora.

[K10] Pregled početne stranice. Registrovani i neregistrovani korisnici na početnoj stranici vide nasumične objave sa nasumičnih zajednica. Pored objave nalazi se link koji vodi ka datoj zajednici.

[K11] Promena lozinke. Pri promeni lozinke prvo se unosi trenutna lozinka.

[K12] Promena dodatnih podataka na profilu - moguće je podesiti ime koje se prikazuje umesto korisničkog imena i postaviti opis svog profila, kao i sliku. Na profilu se vidi karma (ukupna karma sa svih objava i komentara datog korisnika)

Moderator aplikacije ima na raspolaganju sve funkcionalnosti koje ima registrovan korisnik, kao i:

- [M1] Ažuriranje liste pravila. Moderator zajednice može da opiše pravila (eng. *rules*) date zajednice. Pravila su korisnicima dostupna na stranici zajednice.
- [M2] Blokiranje i odblokiranje korisnika. Blokiranje (eng. *ban*) sprečava korisnika da kreira objave, kreira komentare ili da *upvote*-uje ili *downvote*-uje objave ili komentare.
- [M3] Ažuriranje liste tagova (eng. *flairs*).
- [M4] Pregled prijavljenih komentara ili objava. Moderator može da ukloni objavu ili komentar.

Administrator aplikacije ima na raspolaganju sve funkcionalnosti koje ima registrovan korisnik, ali i:

- [A1] Uklanjanje moderatora. Uklonjen moderator postaje običan korisnik.
- [A2] Suspendovanje zajednice. Prilikom suspendovanja zajednice administrator navodi tekstualan opis za razlog suspendovanja. Aplikacija automatski uklanja sve moderatore sa suspendovane zajednice.

Za implementaciju aplikacije iskoristiti sledeće softverske pakete:

- Spring framework
- Apache Tomcat (ne mora biti posebno integrisan, može Spring Boot)
- MySQL i
- ElasticSearch

Podatke kojima upravlja aplikacija organizovati uz oslonac na SUBP³.

Nefunkcionalni zahtevi

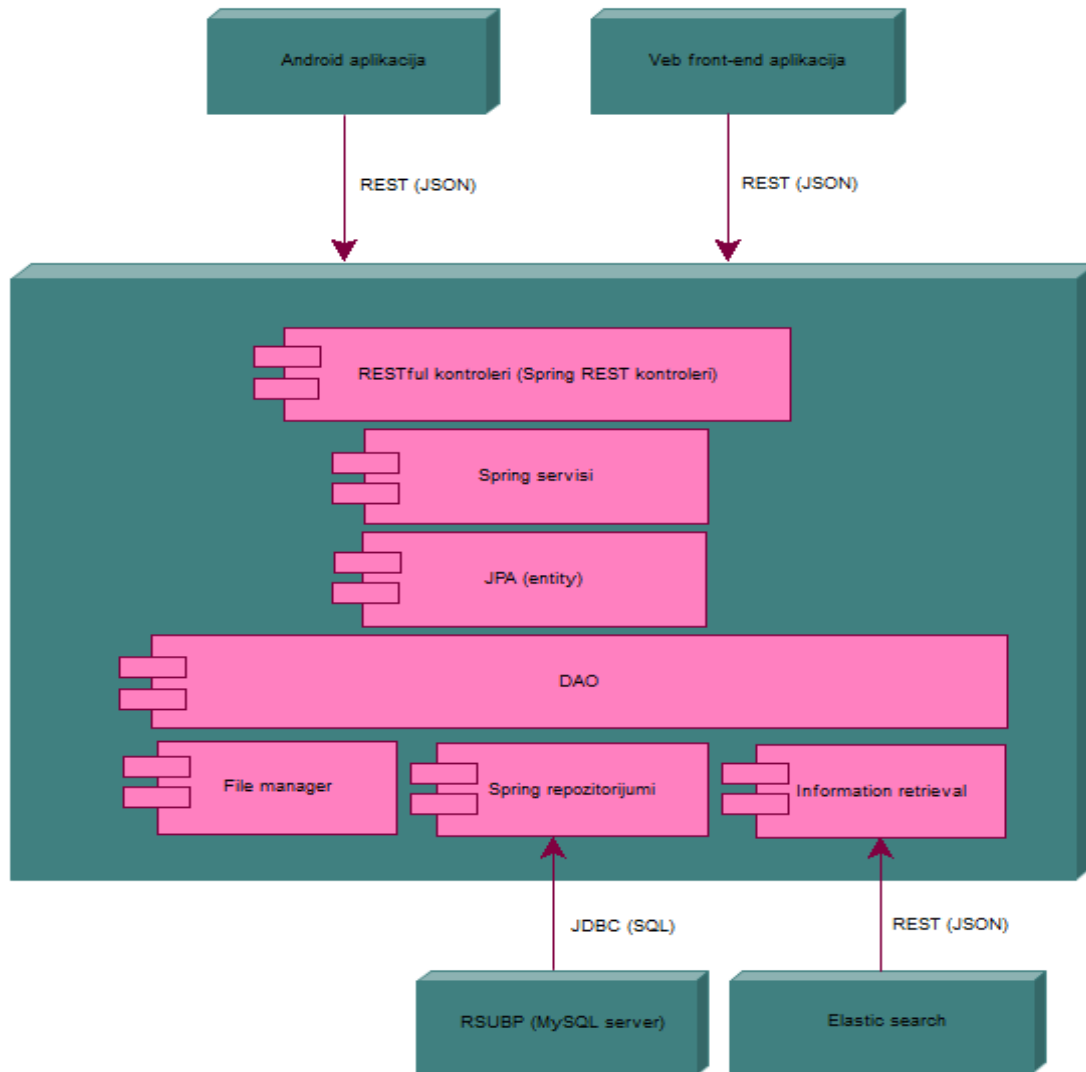
Podržati autentifikaciju korisnika upotrebom korisničkog imena i lozinke i autorizaciju korisnika upotrebom mehanizma tokena.

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

³ Skraćeno od Sistem za Upravljanje Bazama Podataka.

Arhitektura aplikacije

Aplikacija je raspoređena na tri uređaja: 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. File manager skladišti i preuzima digitalne dokumente iz fajl sistema. Information retrieval komponenta omogućuje indeksiranje i pretragu objava upotrebom ElasticSearch platforme [7].

Model podataka

Na [linku](#) prikazan je model podataka RedditClone aplikacije. Entitet User predstavlja registrovanog korisnika aplikacije i namenjen je skladištenju podataka koji se koriste za autentifikaciju i autorizaciju. Neregistrovan korisnik ima samo prava pregledanja objava i komentara na zajednicama. Korisnik takođe može biti administrator ili moderator. Moderator održava određenu zajednicu, dok administrator rukovodi aplikacijom i ima mogućnost uklanjanja zajednice. Objave su opisane entitetom Post i one su tekstualne, ali mogu sadržati i sliku. Entitet Comment predstavlja komentar u datoj aplikaciji. Entitet Reaction predstavlja *upvote* ili *downvote* određene objave ili komentara. Ukoliko sadržaj krši pravila zajednice ili aplikacije, kreira se entitet Report koji se odnosi na objavu ili komentar. Entitet Flair predstavlja tagove koje moderator održava. Entitet Community predstavlja zajednicu (subreddit) koji sadrži objave i komentare.

Rad sa fajlovima

Fajlove je potrebno čuvati u zasebnom folderu fajl sistema. Putanja do fajla može se čuvati uz objavu u bazi podataka (da bismo kasnije pristupili fajlu ako to bude potrebno).

Napomene

1. Studenti ne smeju svojevorno da menjaju model baze podataka.
2. U delu Opis funkcionalnosti zahtevi koji su obojeni **žutom** bojom označavaju zahteve za ocenu 6, zahtevi obojeni **crvenom** bojom označavaju zahteve za ocenu 7, zahtevi obojeni **zelenom** bojom označavaju zahteve za ocenu 8, zahtevi obojeni **magenta** bojom označavaju zahteve za ocenu 9, a zahtevi obojeni **plavom** bojom označavaju zahteve za ocenu 10.
3. Projekat se radi samostalno.

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/>
6. JavaMail, <https://javaee.github.io/javamail/>
7. Elasticsearch, <https://www.elastic.co/>