

Društvena mreža

Potrebno je implementirati aplikaciju društvene mreže. Kao inspiraciju možete koristiti društvene mreže koje poznajete poput *Facebook-a*, *Instagram-a*, *Twitter-a* i sl.

Korisnik aplikacije ima na raspolaganju sledeće funkcionalnosti:

[K1] Registracija korisnika. Administrator sistema 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 nije moguće pristupiti ostatku aplikacije.

[K3] Rukovanje objavama.

. Obavezno polje objave su sadržaj tj. tekst te objave i PDF dokument sadržaja objave.

[K4] Ažuriranje komentara. Moguće je odgovoriti (eng. *reply*) na komentar. Moguć je proizvoljan broj odgovora na komentar (jedan ili više) i moguće je ići u proizvoljnu dubinu sa odgovorima na komentar (*reply* na *reply* na *reply*...).

[K5] Korisnicima sistema je omogućeno reagovanje na objave i komentare (*like*, *dislike*, *heart*).

[K6]

[K8] Rukovanje grupama. Bilo koji registrovan korisnik može da kreira grupu i automatski postaje administrator date grupe. Prilikom kreiranja grupe korisnik upload-uje PDF dokument sa opisom grupe. Jedino administrator sistema ima pravo da suspenduje (logički obriše) grupu.

Svaki korisnik može da napravi zahtev da postane član grupe. Zahtev odobrava Administrator grupe (vidi M2). Svaki član grupe može da prati i kreira objave unutar grupe.

[K9]

[K10]

[K11] Promena lozinke. Pri promeni lozinke prvo se unosi trenutna lozinka i dva puta se unosi nova lozinka.

[K12] Promena dodatnih podataka na profilu - moguće je podesiti ime koje se prikazuje umesto korisničkog imena i postaviti opis svog profila, [REDACTED]. Na profilu se takođe vidi i spisak svih grupa kojima korisnik pripada.

[K13] Pretraga korisnika na sistemu s ciljem pronalaska novih prijatelja. Pretraga se vrši na serverskoj strani na osnovu imena i prezimena korisnika. Korisnik sistema može da pošalje zahtev za prijateljstvo svakom korisniku sistema. Prijateljstvo se uspostavlja onda kada druga strana odobri zahtev. Svaki zahtev se može i odbiti.

Administrator grupe ima na raspolaganju sve funkcionalnosti koje ima i registrovan korisnik, kao i:

[M1] [REDACTED]

[M2] (nastavak na K8) Administrator grupe ima pregled svih pristiglih zahteva za pridruženje grupi. Tek nakon što administrator grupe odobri zahtev korisnik postaje deo grupe. Administrator grupe takođe, može svaki pristigli zahtev i da odbije.

Administrator sistema ima na raspolaganju sve funkcionalnosti koje imaju administrator grupe i registrovan korisnik, ali i:

[A1] [REDACTED]

[A2] Suspendovanje grupe. Prilikom suspendovanja grupe administrator sistema navodi tekstualan opis za razlog suspendovanja. Aplikacija automatski uklanja sve administratore grupe sa suspendovane grupe.

UES DEO:

Omogućiti indeksiranje grupa i objava u *Elasticsearch*-u, omogućiti čuvanje dokumenata u *MinIO* bazi.

[S1] Pretraga grupa:

- Pretraživanje grupa po nazivu
- Pretraživanje grupa po opisu
- Pretraživanje grupa po opisu iz zakačenog PDF fajla
- Pretraživanje grupa po opsegu broja objava (od - do), gde može biti zadata donja i/ili gornja granica opsega

[REDACTED]

- 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

[REDACTED]

- Pretraživanje po pravilima grupe
- Prilikom prikaza rezultata prikazati naziv grupe, broj objava, prosečan broj lajkov

[S2] Pretraga objava:

- Pretraživanje objava po naslovu
- Pretraživanje objava po tekstu
- Pretraživanje objava po tekstu iz zakačenog PDF fajla
- Pretraživanje objava po broju lajkova (od – do, računaju se lajkovi na samoj objavi + broj lajkova u komentarima), 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
- Prilikom prikaza rezultata prikazati naslov objave, tekst objave

Za implementaciju aplikacije iskoristiti sledeće softverske pakete:

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

Podatke kojima upravlja aplikacija organizovati uz oslonac na SUBP (Sistem Za Upravljanje Bazom Podataka). Indekse čuvati u Elasticsearch sistemu a dokumente čuvati u MinIO bazi.

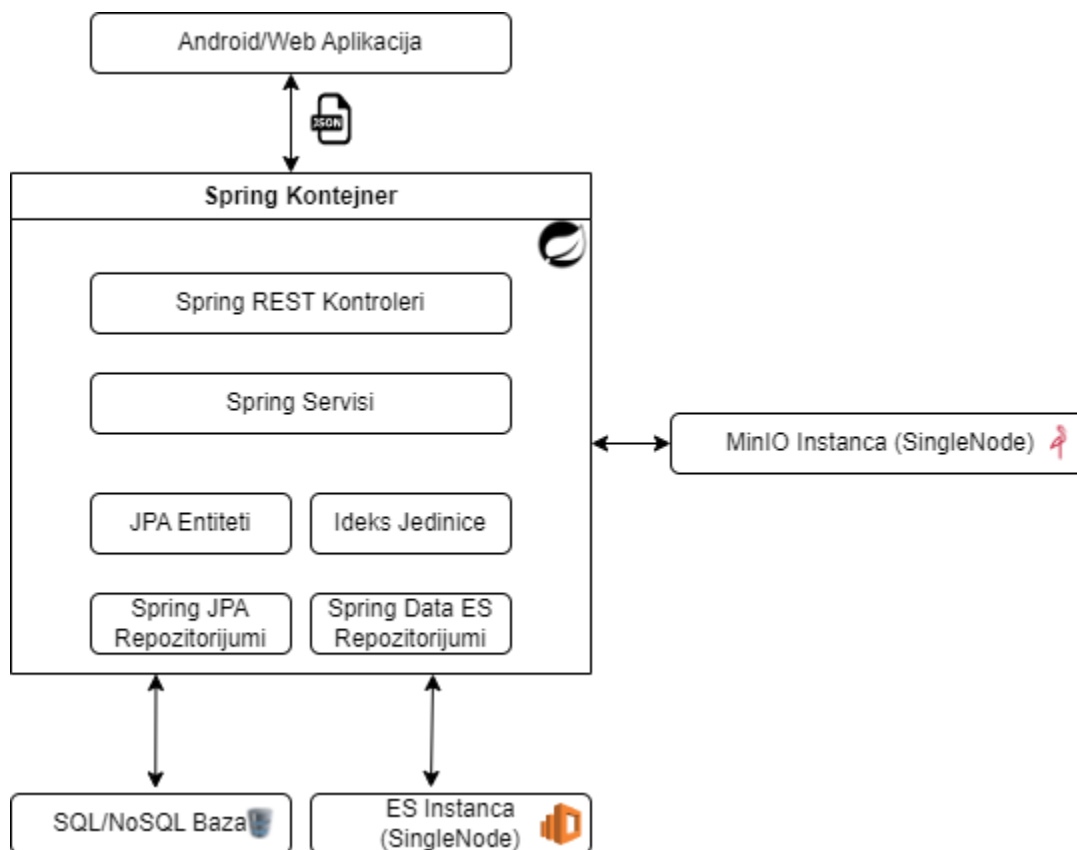
Nefunkcionalni zahtevi

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.

Arhitektura aplikacije

Aplikacija je raspoređena na tri uređaja: Veb brauzer ili android aplikacija, Spring kontejner (u Tomcat serveru ili pokrenut pomoću Spring Boot), SUBP, Elasticsearch i MinIO. 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 Maven [5] ili neki drugi alat, a dozvoljeno je i koristiti Spring Boot i na taj način konfigurisati i pokretati aplikaciju.

Model podataka

Na [linku](#) prikazan je model podataka aplikacije. Entitet User predstavlja registrovanog korisnika aplikacije i namenjen je skladištenju podataka koji se koriste za autentifikaciju i autorizaciju. Neregistrovan korisnik ima jedino mogućnost registrovanja na aplikaciju. Korisnik takođe može biti administrator sistema ili administrator grupe. Administrator grupe održava određenu grupu, dok administrator sistema rukovodi aplikacijom i ima mogućnost uklanjanja grupa. Objave su opisane entitetom Post i one su tekstualne, ali mogu sadržati i slike. Entitet Comment predstavlja komentar u datoj aplikaciji. Entitet Reaction predstavlja reakciju na određene objave ili komentara. Ukoliko sadržaj krši pravila grupe ili aplikacije, kreira se entitet Report koji se odnosi na objavu, komentar ili korisnika. Entitet Group predstavlja grupu koji sadrži objave i komentare i njoj pripadaju korisnici, a održava je administrator grupe.

Napomene

1. Studenti ne smeju svojevolsjno 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 **ljubičastom** 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. Maven, <https://maven.apache.org/>
6. Firebase, <https://firebase.google.com/>