

## Email klijent

Potrebno je implementirati aplikaciju za rad sa email-ovima (u daljem tekstu EmailClient).

Korisnik aplikacije ima na raspolaganju sledeću funkcionalnost:

- [A1] Registracija korisnika
- [A2] Prijava i odjava sa sistema.
- [A3] Ažuriranje<sup>1</sup> liste email naloga. Dodavanje inicijalne strukture direktorijuma.
- [A3] Ažuriranje<sup>2</sup> liste email kontakata. Upload slika vezanih za kontakt je potrebno podržati.
- [A5] Ažuriranje liste direktorijuma. Direktorijumi mogu imati i pod-direktorijume.
- [A6] Ažuriranje liste pravila. Pravila se mogu izvršavati automatski prilikom primanja email-ova ili eksplicitnim pozivom.
- [A7] Ažuriranje liste tagova.
- [A8] Primanje email-ova sa servera. Različito označavanje pročitanih i nepročitanih email-ova. Otvaranje email-a za čitanje i preuzimanje attachment-a. Sortiranje liste email-ova po subject-u, pošiljaocu, datumu. Filtriranje liste po dodeljenim tagovima.
- [A9] Slanje email-a. Odabir naloga sa kojeg se šalje email.
- [A10] Manuelno razvrstavanje email-ova po direktorijumima
- [A11] Pozivanje definisanih pravila
- [A12] Dodavanje tagova email-ovima
- [A13] Pretraga email-ova.
- [A14] Promena lozinke.
- [A15] Promena ličnih podataka.

Za implementaciju aplikacije iskoristiti sledeće softverske pakete:

- Spring framework
- Apache Tomcat (ne mora biti posebno integrisan, može Spring Boot)
- MySQL i
- JavaMail biblioteku [6] za komunikaciju sa mail serverima

Podatke kojima upravlja aplikacija organizovati uz oslonac na SUBP<sup>3</sup>.

### Nefunkcionalni zahtevi

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

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

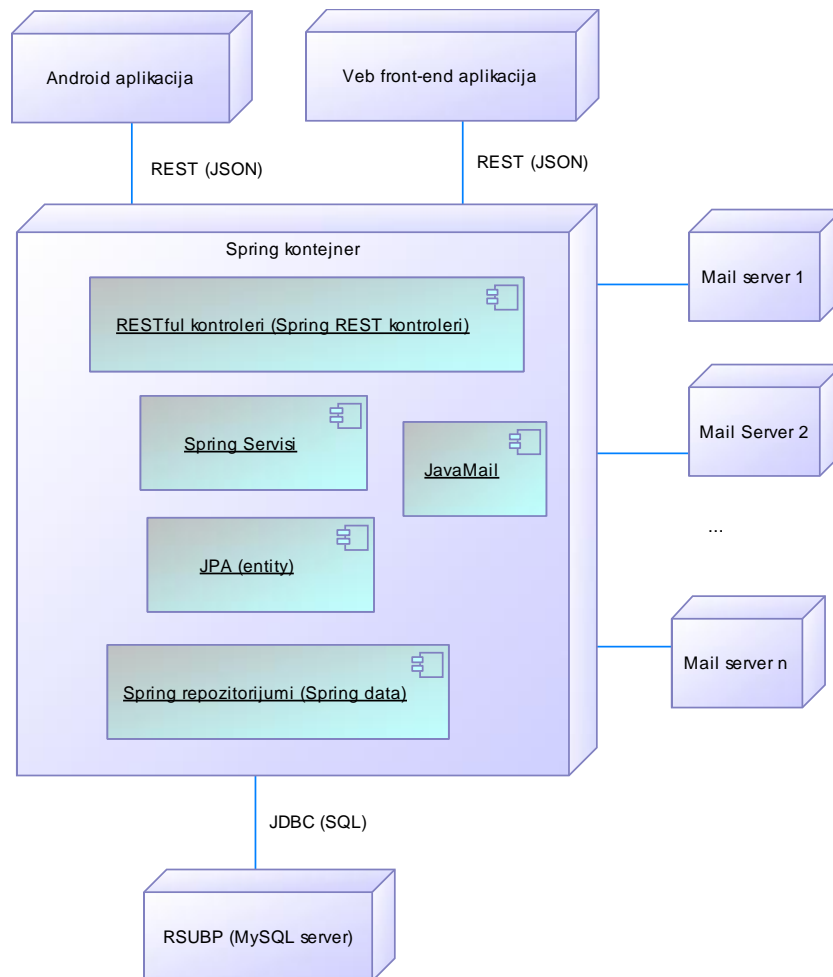
<sup>1</sup> Pod ažuriranjem se podrazumevaju aktivnosti vezane za dodavanje, promenu i uklanjanje odgovarajućih pojava entiteta. Većinu informacionih sistema karakteriše neograničen period čuvanja podataka te se aktivnost uklanjanja odgovarajućih pojava entiteta retko koristi.

<sup>2</sup> 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.

<sup>3</sup> 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.

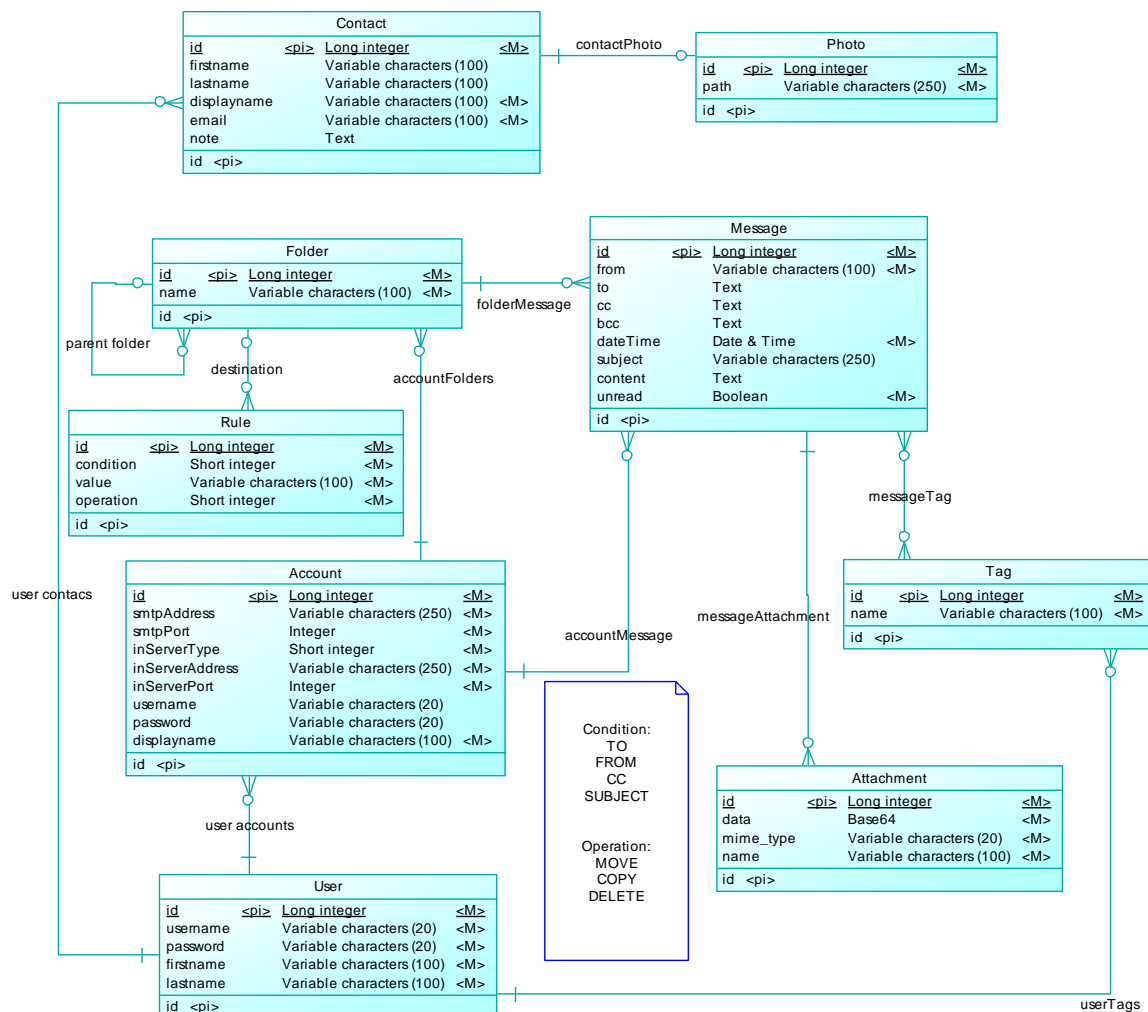


**Slika 1:** Arhitektura aplikacije – dijagram rasporeda

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. Za komunikaciju sa mail serverima koristiti JavaMail[6] ili neku drugu biblioteku (nije potrebno implementirati od nule mail protokol).

## Model podataka

Na slici 2 prikazan je model podataka EmailClient aplikacije. Entitet Message predstavlja poruku, ako postoje attachment-i sa tom porukom oni su skladišteni pomoću entiteta Attachment. Takođe, poruka može biti označena jednim ili više tagova (važno, hitno, itd.). Poruke su razvrstane po folderima koji mogu imati hijerarhiju (entitet Folder i rekurzivna veza). Entitet rule opisuje pravila za brisanje, kopiranje ili premeštanje poruka iz foldera u folder. Kontakti su opisani entitetima Contact i svaki kontakt može imati jednu ili više slika. Putanja do slike može biti skladištena poljem path u entitetu Photo ili se može modifikovati model da slika bude skladištena kao BLOB polje. Entitet Account je namenjen skladištenju parametara za konekciju na mail server. Entitet User je namenjen skladištenju podataka o korisnicima sistema koji se koriste za autentifikaciju i autorizaciju.



Slika 2: Model podataka

## Napomene

1. Studenti ne smeju svojevóljno da menjaju model baze podataka.
2. 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.
3. Projekat se može raditi u timovima od tri studenta

## 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/>