

## 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.

[A4] Pretraga email kontakata.

Pretraživanje kontakata po imenu.

Pretraživanje kontakata po prezimenu.

Pretraživanje kontakata po napomeni.

Kombinacija prethodnih parametara pretrage (BooleanQuery, omogućiti i AND i OR operator između polja)

Obezbediti podršku u poljima forme za unos PhrazeQuery i FuzzyQuery.

Pretraga je nezavisna od velikih i malih slova, kao i od ćiriličnog i latiničnog pisma.

Prilikom prikaza rezultata prikazati ime, prezime i email adresu

[A5] Ažuriranje liste direktorijuma. Direktorijumi mogu imati i pod-direktorijume.

[A6] 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.

[A7] Manuelno razvrstavanje email-ova po direktorijumima

[A8] Pretraga email-ova.

Pretraživanje email-ova po naslovu.

Pretraživanje email-ova po pošiljaocima i primaocima.

Pretraživanje email-ova po sadržaju email-a.

Pretraživanje email-ova po sadržaju PDF priloga.

Pretraživanje email-ova po direktorijumima.

Pretraživanje email-ova po email naložima.

Kombinacija prethodnih parametara pretrage (BooleanQuery, omogućiti i AND i OR operator između polja)

Obezbediti podršku u poljima forme za unos PhrazeQuery i FuzzyQuery.

Pretraga je nezavisna od velikih i malih slova, kao i od ćiriličnog i latiničnog pisma.

Prilikom prikaza rezultata kreirati dinamički sažetak (koristiti Highlighter biblioteku)

Prilikom prikaza rezultata ponuditi i link za otvaranje email-ova.

<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.

[A9] Promena lozinke.

[A10] 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
- Lucene [7] ili Elasticsearch [8] za implementaciju indeksiranja i pretraživanja email-ova

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.

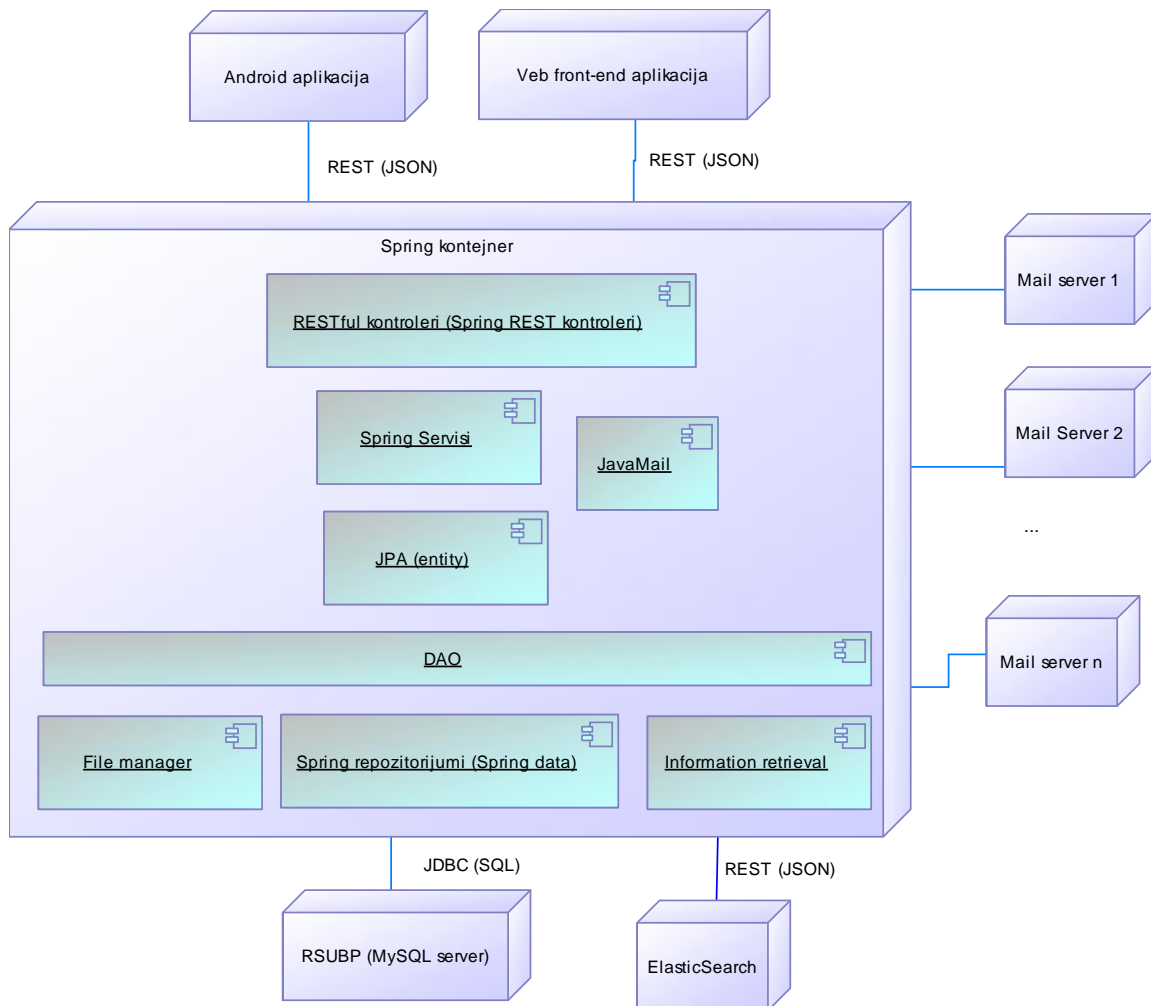
Beležiti poruke o važnim događajim koji su nastali prilikom izvršavanja aplikacije.

## **Arhitektura aplikacije**

Aplikacija je raspoređena na tri (ili četiri) uređaja: Veb brauzer ili android aplikacija, Spring kontejner (u Tomcat serveru ili pokrenut pomoću Spring Boot) i SUBP (i Elasticsearch koji nije obavezan, može se implementirati pomoću Lucene biblioteke). Dijagram rasporeda prikazan je na slici 1.

---

<sup>3</sup> Skraćeno od Sistem za Upravljanje Bazama Podataka.

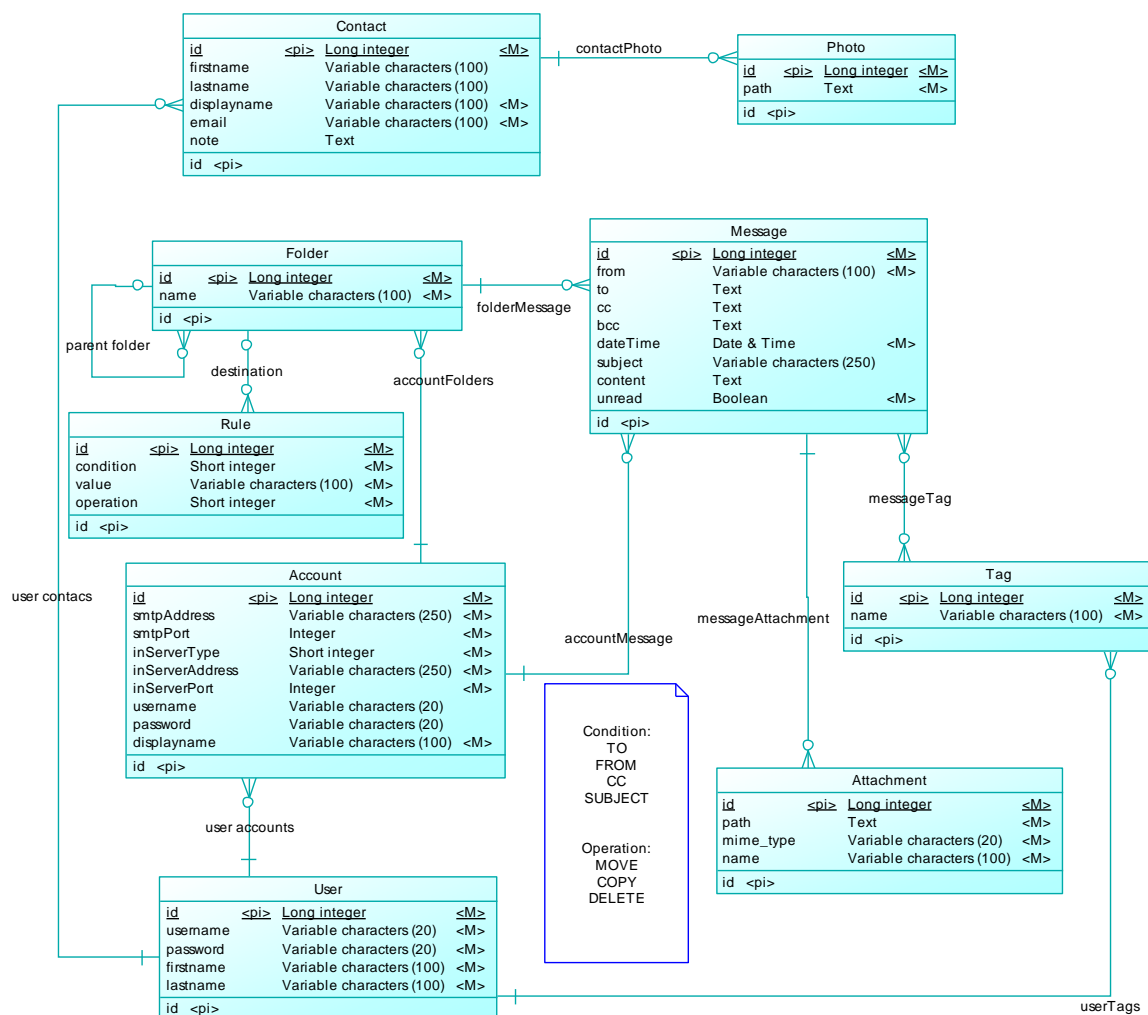


Slika 1: Dijagram rasporeda

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). File Manager skladišti i preuzima digitalne dokumente (priloga/attachment-e email-ova) iz fajl sistema. Information retrieval komponenta omogućuje indeksiranje i pretragu email-ova i njihovih priloga upotrebom Lucene IR biblioteke [7] ili ElasticSearch platforme [8].

## 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 se skladišti poljem path u entitetu Photo. 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. Inicijalno popunjavanje baze podataka može se obaviti putem SQL ili ANT scripta ili preuzimanjem email-ova sa email servera.

2. Studenti ne smeju svojevolutno da menjaju model baze podataka i mapu sajta.
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.
4. Projekti su individualni.

## 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. Apache Lucene, <https://lucene.apache.org/>
8. Elasticsearch, <https://www.elastic.co/>