

## Projektni zadatak

- decembar 2018. -

Potrebno je napraviti sistem koji će omogućiti lakše dijeljenje informacija i materijala tokom studija. Sistem se sastoji iz nekoliko dijelova:

- glavna aplikacija,
- administratorska aplikacija,
- sistem za upravljanje vijestima i događajima.

### Glavna aplikacija

Glavnu aplikaciju koriste studenti koji treba da otvore korisnički nalog. Otvaranje naloga se obavlja na odgovarajućoj stranici gdje se unose ime, prezime, korisničko ime, lozika (2 puta zbog provjere) i *mail* adresa. Ako je korisničko ime slobodno, a *mail* nije već korišten, korisnik se uspješno registruje i preusmjerava na stranicu za izmjenu profila. Na toj stranici se osim osnovnih podataka može izabrati fakultet iz liste ponuđenih fakulteta, unijeti studijski program, godina studija, izabrati profilna slika i popuniti polje za opis interesovanja.

Korisnici glavne aplikacije treba da se prijave na sistem da bi ga koristili, a bez prijave na sistem jedine dostupne stranice su stranica za prijavu i stranica za registraciju.

Početna stranica prijavljenih korisnika prikazuje sa lijeve strane sve studente sa kojima je prijavljen student povezan. Na spisku treba da se nalazi njihova mala slika (avatar), ime i prezime. U slučaju da je broj povezanih studenata veći od visine stranice potrebno je prikazati *scroll*.

Na sredini stranice treba da se nalazi lista svih objava. Objave mogu biti vijesti ili događaji koji se dobijaju sa posebnog sistema, ili objave drugih korisnika. Svaki korisnik može objaviti link, tekst ili Youtube video. Samo povezani prijatelji mogu vidjeti objave nekog korisnika, dok su objave koje se dobijaju sa odgovarajućeg sistema vidljive svim korisnicima sistema. Svaka objava ima opcije *like* i *dislike*, bez obzira na izvor sa kojeg dolazi. Na gornjem dijelu liste objava prikazuje se pet najpopularnijih objava (*total likes* - *total dislikes*), dok se u donjem dijelu prikazuju ostale objave sortirane po vremenu kreiranja. Pored objave prikazuje se avatar korisnika koji je objavio. Sve vijesti se periodično osvježavaju (svakih 30 sekundi).

Na desnoj strani ekrana prikazuje se dio za blog i dio za dijeljene materijale. Blog predstavlja mjesto gdje studenti mogu diskutovati na neku temu. Svaki korisnik može pokrenuti blog tako što će napisati neki sadržaj, dok ostali korisnici mogu ostavljati komentare. Blogovi su vidljivi svim korisnicima sistema, a čuvaju se u MongoDB bazi podataka zajedno sa komentarima. Lista blogova treba biti sortirana u opadajućem redoslijedu po vremenu

kreiranja bloga. Dio za dijeljenje materijala predstavlja mjesto gdje studenti mogu da urade *upload* nekog dokumenta proizvoljnog formata koji se čuva na serveru, a ostali korisnici sistema mogu da ga preuzmu. Prilikom *upload*-a fajla studenti unose opis. Na listi fajlova prikazuju se avatar korisnika i opis fajla. Klikom na avatar otvara se profilna stranica tog korisnika.

Studenti imaju stranicu za povezivanje. Na toj stranici nalazi se prikaz svih korisnika sistema (avatar, ime i prezime) i opcija za povezivanje. Na vrhu liste nalazi se filter u kojem se može izabrati fakulteta, pa se na listi prikazuju samo oni studenti koji studiraju na izabranom fakultetu. Podrazumijevana vrijednost je fakultet na kojem studira trenutno prijavljeni student. Kada student aktivira opciju za povezivanje, tada se drugom studentu pošalje obavještenje u kojem on može da prihvati ili odbije zahtjev za povezivanjem. Osim toga, potrebno je prikazati i listu trenutno povezanih studenata sa prijavljenim pri čemu se uz svakog povezanog studenta nalazi i opcija za brisanje te veze.

Osim navedenih opcija, studenti imaju i profilnu stranicu na kojoj mijenjaju sve podatke (ista stranica koja je opisana u procesu registracije), pri čemu se za promjenu šifre koristi druga forma. Studenti mogu vidjeti tuđu profilnu stranicu samo ako su povezani sa tim studentom, pri čemu ne mogu mijenjati podatke.

Podaci se čuvaju u MySQL bazi podataka, a aplikacija mora biti implemenirana pomoću JSP M2 modela, HTML5, JS i AJAX. Po potrebi moguće je koristiti servlete. Aplikacija mora imati dizajn koji se prilagođava upotrebi na mobilnim uređajima.

## Administratorska aplikacija

Ova aplikacija treba da se koristi za monitoring ostalih dijelova sistema. Administratori su jedini korisnici ove aplikacije i moraju biti prijavljeni na sistem da bi mogli da ga koriste. Nalozi se kreiraju direktno u bazi podataka.

Aplikacija na početnoj stranici prikazuje broj trenutno aktivnih korisnika glavne aplikacije, ukupan broj registrovanih korisnika i grafikon na kojem je prikazan broj korisnika po satima u prethodnih 24h.

Osim toga, administrator može blokirati bilo kojeg korisnika, resetovati njegovu lozinku na neku slučajno odabranu vrijednost, pregledati sve objave sa sistema za upravljanje vijestima i događajima i obrisati bilo koju objavu tog sistema.

Sistem koristi istu bazu podataka kao i glavna aplikacija, a implementacija mora biti pomoću JSF-a. Grafikon se može implementirati pomoću Chart.JS biblioteke.

## Sistem za upravljanje vijestima i događajima

Ovaj sistem se sastoji iz JSF aplikacije i RESTful servisa. Web aplikacija se koristi za kreiranje događaja koji ima naziv, vrijeme održavanja, opis, sliku (URL) i kategoriju. Kategorije događaja su ispiti, predavanja, promocije, prakse, putovanja i ostalo. Događaji su dostupni kao RSS feed. Osim toga, web aplikacija treba da ima formu koja se koristi za kreiranje vijesti. Vijesti imaju datum kreiranja, naslov i sadržaj, a drugim aplikacijama su

dostupne preko RESTful servisa. Vijesti i događaji se čuvaju u MySQL bazi podataka, a sve izmjene (blokiranje, glasanje like/dislike) dostupni su preko odgovarajućih REST servisa. **Dozvoljeno je dodati još neka polja koja poboljšavaju kvalitet rada.**

Za realizaciju je potrebno poštovati MVC principe. U svim situacijama, gdje to ima praktičnog smisla, potrebno je implementirati dinamičke elemente interfejsa korištenjem AJAX tehnologije. Svi podaci za koje nije naglašen način čuvanja, čuvaju se u relacionoj bazi podataka. Model aplikacije koji se ne odnosi na NoSQL bazu mora biti realizovan korištenjem DAO patern-a za pristup podacima.

Napomene:

- Projektni zadatak se radi pojedinačno.
- Odbrana projektnog zadatka obavlja se u laboratoriji. Na odbranu je potrebno donijeti kompletan izvorni kod, model podataka, kao i model aplikacija.
- Pri realizaciji projektnog zadatka zabranjeno je korištenje generatora koda, osim onih koji su korišteni na laboratorijskim vježbama ili predavanjima.
- Interfejs prema korisniku treba da bude organizovan na jednoobrazan način tj., sve stranice treba da imaju sličan izgled i intuitivne kontrole i prikaz. Detalji dizajna se ocjenjuju, a studenti mogu koristiti gotove biblioteke poput Bootstrap-a, Material Lite, vlastite implementacije ili neke besplatne teme.
- **Svaki vid serverske validacije potrebno je da bude što efikasnije realizovan.** Obavezno napraviti klijentsku validaciju.
- Pri razvoju je potrebno koristiti MySQL server baze podataka i odgovarajući JDBC konektor za dio vezan za rad sa relacionom bazom podataka.
- Nije dozvoljeno koristiti stored procedure, funkcije i okidače (triggere). Dakle, baza podataka ne smije imati bilo kakvu logiku osim definisanih tabela i ograničenja koja važe među njima.
- Studenti koji su završili projektni zadatak su dužni da se jave predmetnom asistentu prije termina ispita koji su prijavili kako bi se blagovremeno organizovala odbrana.
- Ovaj projektni zadatak vrijedi do objave novog projektnog zadatka.