

Projektni zadatak

- decembar 2019. -

Potrebno je napraviti sistem koji će omogućiti lakše dijeljenje informacija o potencijalnim opasnostima. Sistem se sastoji iz nekoliko dijelova:

- glavna aplikacija,
- administratorska aplikacija i
- sistem za pružanje pomoći.

Glavna aplikacija

Glavnu aplikaciju koriste registrovani korisnici koji treba da otvore korisnički nalog. Otvaranje naloga se obavlja na odgovarajućoj stranici gdje se unose ime, prezime, korisničko ime, lozinka (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 država u kojoj korisnik živi iz liste ponuđenih država, izabrati profilna slika i izabrati da li želi primati notifikacije o hitnim upozorenjima (unutar aplikacije ili na mail). Korisnik može da bude samo iz neke od evropskih država koje se popunjavaju konzumiranjem RESTful servisa dostupnog na <https://restcountries.eu/rest/v2/region/europe>. Korištenjem drugog RESTful servisa, a na osnovu vrijednosti *alpha2Code* za državu, automatski se popunjavaju regioni te države, a na osnovu regiona i gradovi konzumiranjem odgovarajućih servisa sa <http://battuta.medunes.net/>. Ukoliko korisnik ne izabere profilnu sliku za upload, kao profilna slika dodjeljuje mu se zastava države u kojoj živi. Zastave su takođe dostupne u samom odgovoru servisa prvog servisa, kao atribut *flag*.

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.

Nakon uspješne registracije i prijave na sistem, korisniku se prikazuje stranica na kojoj se na lijevoj strani prikazuje ime i prezime korisnika, njegov avatar, koliko puta se korisnik prijavio na sistem, kao i lista notifikacija o hitnim upozorenjima.

Na sredini stranice treba da se nalazi lista svih objava vezanih za potencijalne opasnosti. Objave mogu biti vijesti ili događaji koji se dobijaju sa RSS feed-a https://europa.eu/newsroom/calendar.xml_en?field_nr_events_by_topic_tid=151, ili objave drugih korisnika. Svaki korisnik može objaviti link, tekst sa slikama ili video (sopstveni snimak ili Youtube video). Pored objave prikazuje se avatar korisnika koji je objavio. Sve vijesti se periodično osvježavaju (svakih 30 sekundi). Objave se mogu podijeliti na društvenim mrežama (minimalno *Facebook* i *Twitter*). Korisnici mogu komentarisati objavu, pri čemu u komentar mogu dodati i sliku.

Registrovani korisnik može da unese objavu o potencijalnoj opasnosti. Pod potencijalnim opasnostima mogu da se podrazumijevaju različite stvari: srušeno drvo na putu, najavljeno olujno nevrijeme, poplava, požar i sl. Potencijalne opasnosti se dodaju u određenu kategoriju prilikom unosa (s tim da jedna objava može da bude klasifikovana u više kategorija). Kategorije opasnosti dodaje administrator. Prilikom unosa, korisnik može da izabere da se radi o hitnom upozorenju, koje se odmah nakon unosa šalje svim korisnicima koji su izabrali takvu vrstu notifikacije. Kod unosa je opciono i da korisnik doda Google maps lokaciju gdje se potencijalna opasnost nalazi.

Na desnoj strani ekrana prikazuje vremenska prognoza za tri slučajno odabrana grada iz države u kojoj korisnik živi. Prognozu preuzimati sa <https://openweathermap.org/api>.

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 odobriti registraciju korisnika, blokirati bilo kojeg korisnika, resetovati njegovu lozinku na neku slučajno odabranu vrijednost, pregledati sve objave sa sistema za pružanje pomoći 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 pružanje pomoći

Ovaj sistem se sastoji iz JSF aplikacije i RESTful servisa. Web aplikacija se koristi za kreiranje poziva za pružanje pomoći građanima u slučaju potencijalne opasnosti, koji ima naziv, vrijeme lokaciju, opis, sliku (URL) i kategoriju. Kategorije poziva za pomoć su fizička pomoć u sanaciji štete, prikupljanje odjeće, prikupljanje osnovnih životnih namirnica, transport i slično. Događaji su dostupni kao RSS feed. Pozivi za pružanje pomoći su drugim aplikacijama su dostupne preko RESTful servisa. Pozivi se čuvaju u MySQL bazi podataka, a sve izmjene (blokiranje, prijava lažnog poziva) 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 zapristup 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.