

Napredne veb tehnologije

Projektni zadatak za 2016. godinu

1. Opis zadatka

Potrebno je napraviti sistem koji služi za praćenje grešaka koje se događaju u realnom vremenu tokom životnog veka nekih aplikacija. Sistem treba da bude implementiran kao klijent-server aplikacija koja ima otvoren api za prijavu grešaka registrovanih aplikacija. Registrovane aplikacije tada mogu da koriste api kako bi prijavile bilo kakve greške ili izuzetke, bez obzira na jezik u kom je pisana ili tip aplikacije.

Sistem se može zatim uvezati sa projektom koji zadat na predmetu Konstrukcija i testiranje softvera, ali to nije neophodno.

Back end

Baza podataka koja će se koristiti je MongoDB. Sistem implementiran sa NodeJS i Express back end-om.

Potrebno je omogućiti:

Registracija i login korisnika na sistem

Registraciju i login **korisnika** na sistem. Svaki korisnik može da bude povezan sa više različitih registrovanih aplikacija. Korisnik je u mogućnosti da registruje nove aplikacije i otvara mogućnost pristupa toj aplikaciji, ostalim korisnicima.

Korisnici imaju informacije, poput email-a (koji takođe može da bude i korisničko ime), imena i prezimena, informacije o pridruženim aplikacijama.

Registracija aplikacije

Potrebno je omogućiti kreiranje novih **aplikacija**. Aplikacije imaju jednog korisnika kao odgovornu osobu koja može da uključuje druge osobe na istu aplikaciju. Svaka aplikacija ima ime, zatim opis tehnologije o kojoj je reč (npr "AngularJS front end", "Java Spring back end", "NodeJS, Express backend" isl). Zatim najskoriju verziju (ako postoji) i link ka repozitorijumu na kojem se nalazi kod kao opciono polje. Aplikacije za sebe imaju vezano više događaja koji predstavljaju greške koje te aplikacije šalju na server.

Prilikom registracije, potrebno je definisati i domen sa kojeg će aplikacija slati greške odnosno DSN (Domain Source Name) koji će jedinstveno određivati aplikaciju koja šalje grešku i biti dozvoljeno registrovanje tog događaja.

Prijem grešaka

Sve greške na sistem pristižu tako što ih pošalje jedna od registrovanih aplikacija. Greške se registruju kao **događaji** vezani su za specifičnu aplikaciju. Svaka greška za sebe može da ima vezano opciono polje koje govori o verziji aplikacije na kojoj se ta greška dogodila. Na ovaj način će se vršiti i automatsko postavljanje najskorije verzije aplikacije. Događaj takođe sadrži *stack* greške odnosno

izuzetka kojem odgovara, vreme kada je on nastao, kao i neke proizvoljne podatke u JSON formatu. Potrebno je da svaki događaj ima polje *fragment* koje će određivati o kom se delu aplikacije radi. U Spring projektu ovakvo polje bi na primer moglo da govori o kontroleru, servisu ili pak url-u na kojem se izuzetak dogodio. Isto važi i za front-end Angular aplikacije. Fragment služi kako bi se isti ili slični događaji mogli grupisati.

Prilikom pristizanja greške na server, potrebno je izvršiti notifikaciju svih korisnika koji su vezani za aplikaciju putem mail-a.

Potrebno je formirati modele podataka takvi da oni budu pogodni za upotrebu u okviru Mongo DBMS-a.

Front end

Klijentska aplikacija čiji je zadatak da vrši prikaz događaja u zavisnosti od prava pristupa logovanog korisnika.

Potrebno pružiti korisniku mogućnost da bira između prikaza svih događaja vezanih za neku aplikaciju, ili događaja grupisanih po *fragment* polju, uz jasno naznačen broj događaja vezanih za taj fragment.

Omogućiti prikaz i filtriranje događaja po verziji aplikacije.

Na stranici koja prikazuje sve podatke vezane za jedan događaj prikazati vremensku raspodelu u pristizanju izuzetaka iz istog fragmenta.

Omogućiti postavljanje komentara na sve događaje kojima korisnik ima pristup, kao i komentara na komentare formirajući kompozitnu strukturu.

2. Primer sistema

Jedan od najpoznatijih sistema za praćenje grešaka je [Sentry](#).