

**IZVEŠTAJ O URAĐENOM DOMAĆEM ZADATKU**

**Predmet: SE201 UVOD U SOFTVERSKO INŽENJERSTVO**

**Školska godina** 2019/20

**Domaći zadatak br. 2**

**Ime i prezime studenta:** Nikola Tasic

**Broj indeksa:** 3698

**TEKST DOMAĆEG ZADATKA**

Za sistem koji ste odabrali u domaćem zadatku 1, uradite sledeće:

* Specificirajte zahteve koje sistem treba da zadovolji.
* Definišite zadatke koji se moraju uraditi da bi se projekat razvoja softvera realizovao.
* Funkcionalni i ne funkcionalni zahtevi
* Uraditi Requirement Model u okviru Power Designer CASE alata ili uzeti IEEE standard za pisanje zahteva u okviru Word dokumenta (pogledajte primere koji se nalaze u zahtevima za projekat)

**SPECIFIKACIJA ZAHTEVA**

**1. UVOD**

Ovaj dokument daje pregled specifikacije softvera, što uključuje sve procese prikupljanja zahteva i projektovanja sistema pod nazivom Goose. Za početak, predstavljena je svrha dokumenta.. Zatim je definisan opis projekta sa posebnim fokusom na ono što će biti doprinos novokreiranog sistema kao i svih relevantnih sistema povezanih sa njim.

**1.1 NAMENA**

Svrha ovog dokumenta je da obezbedi objasnjenje funkcionalnih i nefunkcionalnih zahteva sistema Goose. Dokument takodje pruza informacije u interfejsima odnosno servisima koji ce ovaj sistem pruzati krajnjim korisnicima. Korsnici ovog dokumenta s druge strane bice razvojni tim koji ce implementirati ovaj sistem i koristice ga kao referencu u toku same implementacije.

**1.1 OBIM**

Potreba za ovim sistemom javlja se kod pogrmera koji zele da brzo mogu da testiraju svoj softrver na realnim serverima. Cilj ovog sistema je da im u par klikova omoguci da iporuce softervrsko resenja na fizicki server i pri tome imaju uvid u to kao on radi.

**1.2.1 OSNOVNI PREGLED SISTEMA**

Predlozeni sistem treba da bude lak za korscenje i treba da omoguci postavljanje vise razlicitih aplikacija na jedan fizicki (ili virtuelni server) na jednom domenu i omogucava pristup tim aplikacijama preko odredjenih pod-domena koje korisnik definise za svaku aplikaciju.

Aplikacije su postavljene na server preko URL-a do git repozitorijuma i na osnovu tipa aplikacije server zna na koji nacin treba da konfigurise okruzenje u kome treba da pokrene aplikaciju iz datog repozitorijuma. Sistem na primer treba da ima sposobnost da serve-uje staticke HTML/CSS projekte kao vebsajtove. Svaka aplikacija koja radi pod ovim sistemom radi na jedinstvenom port-u za tu aplikaciju i sistem obzirom na to da prihvata HTTP zahteve vrsi ‚proxy‘ svih zahteva do odgvarajuce aplikacije na osnovu pod-domena.

**1.2.2 BENEFITI**

Korisnik je i sam administrator sopsvtenog servera i ima uvid u sve aplikacije koje su trenutno postavljene od strane sistema. Administrator moze da pokrece i zaustavlja aplikacije, vrsi njihovo brisanje, promenu konfiguracionih parametara i vrsi azuriranje preko predefinisanog git repozitorijuma.

**2. OPSTI OPIS**

Ovo poglavlje definiše opis sistema Goose. Konkretno, proizvod je stavljen u perspektivu korisnika, hardvera, softvera.

**2.1 INTERFEJSI**

**2.1.1 SISTEMSKI INTERFEJSI**

Sistem ce se postavlja na postojeci Linux server. Windows serveri su takodje podrzani ali u tom slucaju ne postoji opcija kontejnerizacije postavljenih aplikacija.

**2.1.2 KORISNICKI INTERFEJSI**

Korisnicki interfejs zahteva autentikaciju preko korisnickog imena i sifre. Korisnik sistema ima pristup upravljackom interfejsu preko Web GUI-a. Tu korisnik moze da pokrece i zaustavlja aplikacije, vrsi njihovo brisanje, promenu konfiguracionih parametara i vrsi azuriranje preko predefinisanog git repozitorijuma.

**2.1.3 SOFRVERSKI INTERFEJSI**

Goose podatke o trenutno postavljenim aplikacijama cuva u redudantnim JSON fajlovima radi jednostavnosti. Goose ima mogucnost integracije postojeceg sistema Ccont koji omogucava kontejnerizaciju postavljenih aplikacija radi povecanje sigurnosti samog servera i podataka na njemu. Samim tim se postize da svaka aplikacija ima virtuelizovani administratorski pristup fajl-sistemu ako je to potrebno.

**2.1.4 ZAHTEVI**

|  |  |
| --- | --- |
| Zahtev | Opis |
| 1.1 | **Konfiguracija sistema** Ulogovani korisnik ima porstup interfejsu za konfiguraciju sistema |
| 1.1.1 | **Osnovna konfiguracija** Korisnik u konfiguracionom interfejsu ima pristup izmeni parametara vezanih za, Goose proxy, Goose deployer i izmenu korisnickog imena i sifre. |
| 1.1.2 | **Obavestenje o izmeni** Korisnik dobija obavestenja da li su nova podesavanja validna i uspesno sacuvana. |
| 1.2 | **Postavljanje aplikacija** Ulogovani korisnik ima pristup interfejsu za postavljanje aplikacija |
| 1.2.1 | **Interfejs za postavljanje** Interfejs za postavljanje sadrzi polja za unos linka do repozitorijuma aplikacije, bekend tehnologiju koja ce pokretati tu aplikaciju na Goose serveru i hostname koji omogucava pristup aplikciji preko web-a. |
| 1.2.2 | **Obavestenje o postavljanju** Korisnik dobija obavestenje o uspesnosti postavljanja aplikacije na server i mogucim greskama koje su dosle u toku posavljanja. |
| 1.3 | **Konfiguracija aplikacija**  Korisnik ima pristup interfejsu za izmenu konfiguracionih parametara za svaku postavljenu aplikaciju |
| 1.3.1 | **Interfejs za konfiguraciju**  Interfejs za konfiguraciju sadrzi unosna polja za promenu parametara koji su bili opcije kod postavljanja aplikacije. |
| 1.3.2 | **Obavestenje o izmeni** Korisnik dobija obavestenje o uspenosti izmene konfiguracionih parametara aplikacije |
| 1.4 | **Pregled aplikacija** |
| 1.4.1 | **Osnovni pregled** Korisnik u interfejsu vidi listu aplikacija koje su trenutno postavljene na server. |
| 1.4.2 | **­­­­­­Status aplikacije** Korisnik u listi pored imena aplikacije ima prikazanu ikonicu koja oznacava da li je aplikacija pokrenuta ili ne. |
| 1.4.3 | **Pokretanje aplikacije** Korisnik u interfejsu za pregled ima pristup dugmetu za pokretanje aplikacije, a ako je aplikacija vec pokrenuta, dugmetu za gasenje aplikacije. |
| 1.4.4 | **Azuriranje aplikacije** Korisnik ima pristup dugmetu za azuriranje aplikacije ako aplikacija nije pokrenuta. |
| 1.4.5 | **Pregled logova aplikacije** Korisnik ima u bilo kom trenutku pristup pregeledu logova bilo kog koraka postavljanja aplikacije ili njenog pokretanja. |

**Tabela 3.1.1 Funkcionalni zahtevi korisnika**

|  |  |
| --- | --- |
| Zahtev | Opis |
| 2.1 | **Bezbednost** |
| 2.1.1 | **Login** Korisnicki interfejs mora biti obezbedjen login sistemom pomocu korisnickog imena i sifre |
| 2.1.2 | **API** API Goose sistema mora biti pristupacan samo ulogovanim korisnicima. Ogranicavanje pristupa se ostvaruje preko JWT tokena. |
| 2.2 | **Logging** |
| 2.2.1 | **HTTP**  Sistem loguje sve HTTP zahteve u fajl kojem korisnik ima pristup. |
| 2.2.2 | **Segmentisani logging**  Logovi HTTP zahteva se cuvaju u fajlovima za svaki pojedinacni dan. |
| 2.3 | **Pristup serveru** |
| 2.3.1 | **SSH** Server na kome je pokrenut Goose sistem treba da ima omogucen SSH servis radi pristupa zbog odrzavanja ili u slucaju kvara. |

**Tabela 3.1.2 Nefunkcionalni zahtevi**

|  |  |
| --- | --- |
| Zahtev | Opis |
| 3.1 | **Pristup internetu** Fizicki server na kome je postavljen sistem mora imati pristup internetu. |
| 3.2 | **Dependencies**  Server treba da ima instalirane runtime alate i kompajlere za Nodejs (v10.6.0 i novije), Python (v3.6 i novije) i Go (v1.14 i novije). |
| 3.3 | **Routing**  Ruter lokalne mreze treba da ima preusmeren port 80 ka masini na kojoj je postavljen Goose sistem. |

**Tabela 3.1.3 Opsti funkcionalni zahtevi**