



## **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 softvrer 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**

Predloženi sistem treba da bude lak za korscenje i treba da omogući postavljanje više različitih aplikacija na jedan fizički (ili virtuelni server) na jednom domenu i omogućava pristup tim aplikacijama preko određenih pod-domena koje korisnik definiše za svaku aplikaciju.

Aplikacije su postavljene na server preko URL-a do git repozitorijuma i na osnovu tipa aplikacije server zna na koji način treba da konfiguriše okruženje u kome treba da pokrene aplikaciju iz datog repozitorijuma. Sistem na primer treba da ima sposobnost da serve-uje statičke HTML/CSS projekte kao veb sajtove. Svaka aplikacija koja radi pod ovim sistemom radi na jedinstvenom port-u za tu aplikaciju i sistem obzirom na to da prihvata HTTP zahteve vrši ,proxy' svih zahteva do odgovarajuće 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 može da pokrene i zaustavlja aplikacije, vrši njihovo brisanje, promenu konfiguracionih parametara i vrši azuriranje preko predefinisano g 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 će se postavljati na postojeći Linux server. Windows serveri su također podržani ali u tom slučaju ne postoji opcija kontejnerizacije postavljenih aplikacija.

#### **2.1.2 KORISNICKI INTERFEJSI**

Korisnički interfejs zahteva autentikaciju preko korisničkog imena i sifre. Korisnik sistema ima pristup upravljačkom interfejsu preko Web GUI-a. Tu korisnik može da pokrene i zaustavlja

aplikacije, vrši njihovo brisanje, promenu konfiguracionih parametara i vrši azuriranje preko predefinisano gita repozitorijuma.

### 2.1.3 SOFRVERSKI INTERFEJSI

Goose podatke o trenutno postavljenim aplikacijama čuva u redundantnim JSON fajlovima radi jednostavnosti. Goose ima mogućnost integracije postojećeg sistema Ccont koji omogućava kontejnerizaciju postavljenih aplikacija radi povećanje sigurnosti samog servera i podataka na njemu. Samim tim se postize da svaka aplikacija ima virtualizovani administratorski pristup fajl-sistemu ako je to potrebno.

### 2.1.4 ZAHTEVI

Zahtev	Opis
1.1	<b>Konfiguracija sistema</b> Ulogovani korisnik ima pristup interfejsu za konfiguraciju sistema
1.1.1	<b>Osnovna konfiguracija</b> Korisnik u konfiguracionom interfejsu ima pristup izmeni parametara vezanih za, Goose proxy, Goose deployer i izmenu korisnickog imena i sifre.
1.1.2	<b>Obavestjenje o izmeni</b> Korisnik dobija obavestjenja da li su nova podešavanja validna i uspesno sacuvana.
1.2	<b>Postavljanje aplikacija</b> Ulogovani korisnik ima pristup interfejsu za postavljanje aplikacija
1.2.1	<b>Interfejs za postavljanje</b> Interfejs za postavljanje sadrži polja za unos linka do repozitorijuma aplikacije, backend tehnologiju koja će pokrenuti tu aplikaciju na Goose serveru i hostname koji omogućava pristup aplikaciji preko web-a.
1.2.2	<b>Obavestjenje o postavljanju</b> Korisnik dobija obavestjenje o uspesnosti postavljanja aplikacije na server i mogućim greskama koje su dosle u toku posavljanja.
1.3	<b>Konfiguracija aplikacija</b> Korisnik ima pristup interfejsu za izmenu konfiguracionih parametara za svaku postavljenu aplikaciju
1.3.1	<b>Interfejs za konfiguraciju</b> Interfejs za konfiguraciju sadrži unosna polja za promenu parametara koji su bili opcije kod postavljanja aplikacije.
1.3.2	<b>Obavestjenje o izmeni</b> Korisnik dobija obavestjenje o uspenosti izmene konfiguracionih parametara aplikacije
1.4	<b>Pregled aplikacija</b>

1.4.1	<b>Osnovni pregled</b> Korisnik u interfejsu vidi listu aplikacija koje su trenutno postavljene na server.
1.4.2	<b>Status aplikacije</b> Korisnik u listi pored imena aplikacije ima prikazanu ikonicu koja označava da li je aplikacija pokrenuta ili ne.
1.4.3	<b>Pokretanje aplikacije</b> Korisnik u interfejsu za pregled ima pristup dugmetu za pokretanje aplikacije, a ako je aplikacija već pokrenuta, dugmetu za gasenje aplikacije.
1.4.4	<b>Azuriranje aplikacije</b> Korisnik ima pristup dugmetu za azuriranje aplikacije ako aplikacija nije pokrenuta.
1.4.5	<b>Pregled logova aplikacije</b> 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	<b>Bezbednost</b>
2.1.1	<b>Login</b> Korisnicki interfejs mora biti obezbedjen login sistemom pomocu korisnickog imena i sifre
2.1.2	<b>API</b> API Goose sistema mora biti pristupacan samo ulogovanim korisnicima. Ogranicavanje pristupa se ostvaruje preko JWT tokena.
2.2	<b>Logging</b>
2.2.1	<b>HTTP</b> Sistem loguje sve HTTP zahteve u fajl kojem korisnik ima pristup.
2.2.2	<b>Segmentisani logging</b> Logovi HTTP zahteva se cuvaju u fajlovima za svaki pojedinačni dan.
2.3	<b>Pristup serveru</b>
2.3.1	<b>SSH</b> Server na kome je pokrenut Goose sistem treba da ima omogucen SSH servis radi pristupa zbog održavanja ili u slučaju kvara.
2.4	<b>Karakteristike servera</b>
2.4.1	<b>Pristup internetu</b> Fizicki server na kome je postavljen sistem mora imati pristup internetu.
2.4.2	<b>Dependencies</b> 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).
2.4.3	<b>Routing</b> Ruter lokalne mreže treba da ima preusmeren port 80 ka masini na kojoj je postavljen Goose sistem.

**Tabela 3.1.2 Nefunkcionalni zahtevi**