

**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.

|  |  |
| --- | --- |
| Zahtev | Opis |
| C01 | Korisnik ce moci da se uloguje na korisnicki interfejs I ima pristup konfiguraciji samog sistema. |
| C02 | Ulogovani korisnik ima pristup interfejsu za postavljanje aplikacija na sistem. |
| C03 | Ulogovani korisnik ima pristup interfejsu za izmenu konfiguracionih parametara postavljenih aplikacija |
| C04 | Korisnik dobija poruku o uspesnosti postavljanja aplikacije na sistem |
| C05 | Korisnik dobija poruku o uspesnosti promene konfiguracionih parametara apklikacije |
| C06 | Korisnik ima uvid u trenutno postavljenje aplikacije. |
| C07 | Korisnik ima indikator o tome da li je trenutka aplikacija pokrenuta na sistemu |
| C08 | Korisnik ima pristup pretrazi trenutno postavljenih aplikacija |

**Tabela 3.1.1 Funkcionalni zahtevi korisnika**

|  |  |
| --- | --- |
| Zahtev | Opis |
| G01 | Server na koji je postavljen sistem treba da ima pristup internetu |
| G02 | Server na koji je postavljen sistem treba da ima pristup podlezucim tehnologijama Node.js I Python |
| G03 | Lokalna mreza servera treba da ima preusmerene portove 80 I 30000-30100 ka masini na kojoj je instaliran sistem |

**Tabela 3.1.2 Generalni funkcionalni zahtevi**

|  |  |
| --- | --- |
| Zahtev | Opis |
| F01 | Sistem mora da ima obezbedjen korisnicki interfejs u vidu autentikacije preko korsinickog imena I sifre |
| F02 | Sistem log-uje sve HTTP pristupe postavljenim aplikacijama u fajl kojem korisnik ima pristup |
| F03 | Sistem pravi log fajl za svaki novi dvadesetocasovni period radi lakseg pregleda |
| F04 | Server treba da ima obezbedjen SSH interfejst  za slucaj kvara |

**Tabela 4.1.1 Nefunkcionalni zahtevi**