

**Prolećni semestar 2019/2020**

*Goose Deployment System*

*SE201*

*Uvod u softversko inženjerstvo*

*Projektna dokumentacija*

***Student****:*

Nikola Tasić 3698

***Profesor****:*

Dragan Domazet

***Asistent****:*

Jovana Jović

Sadržaj

[Uvod 4](#_Toc30886523)

[Svrha dokumenta 4](#_Toc30886524)

[Terminologija 4](#_Toc30886525)

[Opis 5](#_Toc30886526)

[Funkcije softvera 5](#_Toc30886527)

[Korisnici sistema 5](#_Toc30886528)

[Opis sistema za korisnika 5](#_Toc30886529)

[Metodologija izrade 6](#_Toc30886530)

[Specifikacija zahteva 6](#_Toc30886531)

[Funkcionalni i nefunkcionalni zahtevi 6](#_Toc30886532)

[Slučajevi upotrebe 9](#_Toc30886533)

[Slučaj upotrebe postavljanja aplikacije 9](#_Toc30886534)

[Slučaj upotrebe ažuriranja aplikacije 10](#_Toc30886535)

[Slučaj upotrebe brisanja aplikacije 11](#_Toc30886536)

[Slučaj upotrebe promene parametara aplikacije 12](#_Toc30886537)

[Dijagrami sekvenci 13](#_Toc30886538)

[Dijagram postavljanja aplikacije 13](#_Toc30886539)

[Dijagram ažuriranja aplikacije 14](#_Toc30886540)

[Dijagram brisanja aplikacije 15](#_Toc30886541)

[Dijagram promene parametara aplikacije 15](#_Toc30886542)

[Deteljan dijgram postavljanja aplikacije 16](#_Toc30886543)

[Klasni dijagram 16](#_Toc30886544)

[Arhitektura sistema 17](#_Toc30886545)

[Opis arhitekture sistema 17](#_Toc30886546)

[Korisnički interfejs 18](#_Toc30886547)

[Login forma 18](#_Toc30886548)

[Pregled postavljenih aplikacija 18](#_Toc30886549)

[Pregled informacija o pokrenutoj aplikaciji 19](#_Toc30886550)

[Deployment interfejs 21](#_Toc30886551)

[Testiranje 22](#_Toc30886552)

[Zaključak 22](#_Toc30886553)

# 

# Uvod

Ovaj dokument služi da bolje opiše funkcionalnosti proces izrade **Goose** sistema za deployment aplikaicija. Ovaj projektni zadatak baviće se definisanjem osnovnih funkcionalnih i nefunkcionalnih zahteva koje softver mora da zadovolji kako bi se smatrao uspešnim. U dokumentu će biti detaljno opisan rad sistema o kome je reč. Za bolje razumevanje problema sa kojim se susrećemo kako bi projekat bio realizovan biće objašnjeni putem korišćenja dijagrama(Klasni, UseCase, sekvencijalni) koji bi trebalo detaljnije da objasne način i zamisao funkcionisanja sistema. Za potrebe izrade dijagrama koristiće se alata PowerDesigner. Prikaz interfejsa za korišćenje biće realna aplikacija koja već radi u produkciji i pokazuje rezultate. Korisnici ovog sistema su pojedinačni developeri ili jednostvano ljudi koji žele da imaju olakšani pregled i upravljanje svojim web aplikacijama.

Projekat je tako zamišljen da olakša proces postavljanja aplikacija na web servere, njihovo ažuriranje i da omogći lak način za održavanje vise web aplikacija na jednom domenu i jednom fizičkom serveru.

# Svrha dokumenta

Svrha dokumenta je da jasno opiše moguće funkcionalnosti i mogućnosti softvera. Dokument se sastoji od samog uvoda gde se daje kratak opis cilja za kreiranje ovog sistema. Nakon toga, dokument će se sastojati od opisa sistema njegovim mogućnostima, detalje zahteva u kojima su opisani funkcionalni i nefunkcionalni zahtevi. Dokument će se sastojati i od specifikacije dizajna u okviru kog se nalaze dijagrami slučajeva korišćenja, klasni dijagram i svih mogućih sekvencijalnih dijagrama koji su vezani za rad sistema za praćenje prisustva studenata. Nakon toga, dokument će dati opis funkcionisanja sistema u vidu slika i sam dizajn tog sistema u realnoj situaciji.

## Terminologija

Kako će se u dokumentu koristiti terminologija koja zavisi od konteksta i profesionalna terminologija prilikom objašnjavanja i razvoja sistema, potrebno je da čitaoce ovog dokumenta upoznamo sa skraćenicama i definicijama koje ćemo koristiti u razvijanju dokumenta.

* **Goose** – naziv sistema.
* **PowerDesigner** – Grafičko okruženje za modelovanje aplikacija, podataka, poslovnih procesa.
* **Web server** – Fizička mašina koja je povezana na internet i ima svoju internet adresu.
* **Aplikacija** – Ova reč će se koristiti u kontekstu web aplikacije koju korisnik želi da postavi (deploy) na web server na kome radi Goose sistem.
* **Deployment** – proces koji počinje prebacivanjem binarnih fajlova ili izvornog koda aplikacije na web server a završava se aktivnom aplikacijom na istom.
* **YAML** – struktuirani jezik za opisivanje i čuvanje podataka.
* **JSON** – struktuirani jezik za opisivanje i čuvanje podataka.
* **Go** – programski jezik u kome je razvijen backend aplikacije.
* **React** – tehnologija za izradu web stranica i interfejsa
* **Kontejner –** virtuelizovano okruzenje u kome proces tj. Web aplikacija radi.
* **Routing –** preusmeravanja zahveva na aplikacije u zavisnosti od adrese koja je zahtevana.
* **Backend** – Logika aplikacije koja se izvršava na web serveru.
* **Log –** tekstualni fajl sa opisom proteklih dešavanja u procesu.
* **Terminal, Shell –** tekstualni korisnicki interfejs
* **SSH –** protokol sigurnog upravljanja tekstualnih korisničkih interfejsa preko mreže
* **MVP** – minimum viable product
* **API** - interfejst za upravljanje aplikacijama

# Opis

Goose je sistem dizajniran da olaksa proces deploymenta aplikacija na server i upravljanje vise različitih web aplikacija na jednoj fizičkoj ili virtuelnoj mašini. Goose omogućava developšerima brz i lak način testiranja svojih web aplikacija na realnom hardveru ali takođe s druge strane omogućava platformu sa pokretanje više različitih sistema koji mogu da međusobno komuniciraju. Svaka aplikacija je bezbedna i radi u okviru svog kontejnera i u slučacju bilo kakvih katastrofalnih grešaka oprativni sistem koji sve to pokreće ostaje zaštićen.

## Funkcije softvera

* **Autentikacija korisnika sistema**
* **Deployment aplikacija na web server**
* **Podešavanje routinga aplikacija**
* **Ažuriranje aplikacija**
* **Brisanje aplikacija**
* **Konfiguracija parametara postavljenih aplikacija**
* **Pregled logova aplikacija i samog sistema**

# Korisnici sistema

Postoji samo jedan tip korisnika a to je sam developer koji koristi ovaj sistem.

## Opis sistema za korisnika

Korisnik sistema treba da samo pokrene binarnu distribuciju ovog sistema u terminalu i dobiće funkcionalan sistem sa onim sto u struci zovemo *reasonable defaults* odnosno podrazumeavim podešavanjima koja su validna i omogućavaju rad sistema a ne utiču na ostale već postojeće programe. Korisniku se u terminalu ispisuju sva trenutna podešavanja i informacije kako može da pristupi korisnijem web interfejsu preko svog pretraživavča. Posle uspešne autentikacije korisnik ima pristup interfejsu gde odmah može krenuti sa deployovanjem aplikacija. Kofiguracija sistema se vrši preko YAML fajlova za koje korisnik u temrinalu dobija informaciju gde se nalaze i kako treba izmeniti.

# Metodologija izrade

Metodologija koju ćemo koristiti za izradu ovog sistema je **agilni metod razvoja.** Za izbor ovog metoda je zaslužna činjenica da se ovaj sistem sa stoji iz više komponenti koje mogu biti razvijene nezavisno jedna od druge. Backend ovog sistema je servisno orijentisan web server koji komunicira preko JSON HTTP zahteva što znači da korisnički interfejs za čiju bazu je odabran React može biti razvijan paralelno jer su interfejsi komunikacije između ova dva sistema vec jasno predefinisani u dokumentu. Takođe sistem kontejnerizacije aplikacija na serveru je podsistem koji je na primer takođe autnoman i može se razvijati nezavisno. Agilnim metodom možemo voema brzo doći do MVP-a na primer backenda koji će biti funkcionalan bez obzira na to kakav korisnički interfejs komunicira sa njim. Svaki novi inkrement može da dodda novu funkcionalnst ili celu novu komponentu sistema.

# Specifikacija zahteva

## Funkcionalni i nefunkcionalni 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. |

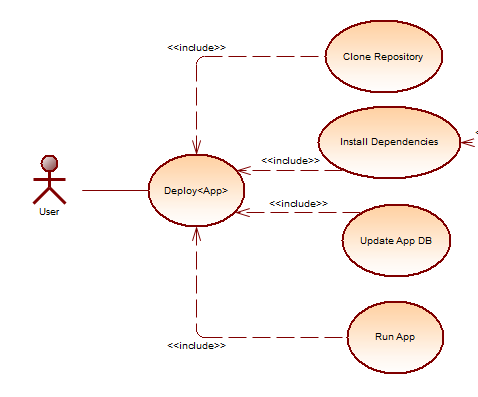
**Tabela 1 Funkcionalni zahtevi korisnika**

|  |  |
| --- | --- |
| Zahtev | Opis |
| 2.1 | **Bezbednost** |
| 2.1.1 | **Login** Korisnicki interfejs mora biti obezbedjen login sistemom pomocu korisnickog imena i šifre |
| 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. |
| 2.4 | **Karakteristike** **servera** |
| 2.4.1 | **Pristup internetu** Fizicki server na kome je postavljen sistem mora imati pristup internetu. |
| 2.4.2 | **Dependencies**  Server treba da ima instalirane runtime alate i kompajlere za Nodejs (v10.6.0 i novije), Python (v3.6 i novije), Go (v1.14 i novije) i Git. |
| 2.4.3 | **Routing**  Ruter lokalne mreze treba da ima preusmeren port 80 ka masini na kojoj je postavljen Goose sistem. |

**Tabela 2 Nefunkcionalni zahtevi**

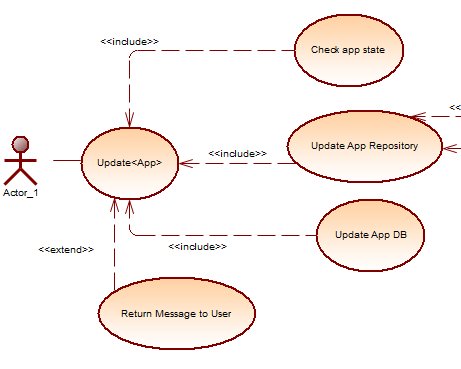
# Slučajevi upotrebe

## Slučaj upotrebe postavljanja aplikacije



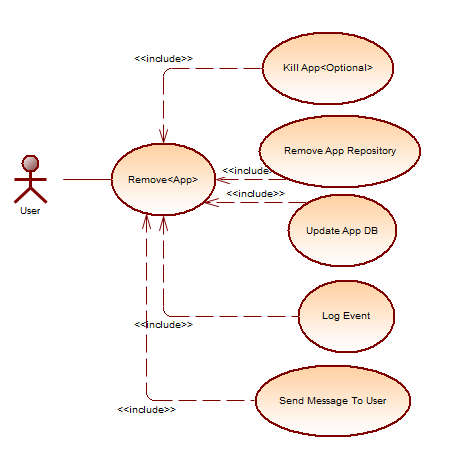
|  |  |
| --- | --- |
| Naziv | Postavljanje aplikacije na sistem |
| Aktor | Korisnik |
| Preduslovi | Nema |
| Okidač | Korisnik je izabrao opciju za postavljanje aplikacije |
| Normalni tok | 1. Korisniku se prikazuje za interfejs sa unosnim poljima za parametre za aplikaciju koju želi da postavi. 2. Korisnik posle unošenja validnih podataka ima mogućnost da pošalje zahtev za postavljanje aplikacije. 3. Korisnik potvrđuje unešene parametre i šalje zahtev za postavljanje. 4. Korisniku se prikazuje informacija da je postavljanje aplikacije zapčeto [IZ - 1] 5. Korisniku se prikazuje informacija da je aplikacija postavljena i aplikacija je dodata u listu postavljenih aplikacija. |
| Alternativni tok | Korisniku se prikazuje poruka o nevalidnim podacima koje treba da izmeni. |
| Izuzeci | [IZ - 1] Korisnik je uneo nevalidne podatke. Korisnik se vraća na interfejs za postavljanje aplikacije |
| Postuslovi | Korisnik je postavio aplikaciju |

## Slučaj upotrebe ažuriranja aplikacije



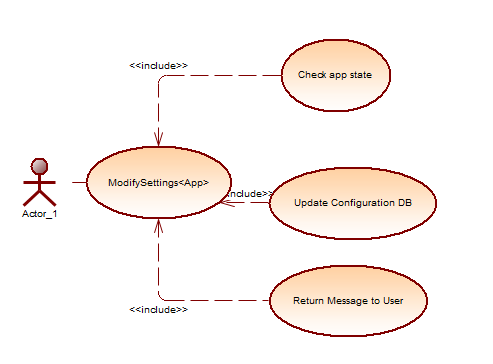
|  |  |
| --- | --- |
| Naziv | Ažuriranje postavljene aplikacije |
| Aktor | Korisnik |
| Preduslovi | Aplikacija nije aktivna |
| Okidač | Korisnik je izabrao opciju za ažuriranje |
| Normalni tok | 1. Korisniku se prikazuje informacija da je aplikacija ažurirana i u listi aplikacija se promenio datum poslednjeg ažuriranja. |
| Alternativni tok |  |
| Izuzeci |  |
| Postuslovi | Korisnik je ažurirao aplikaciju. |

## Slučaj upotrebe brisanja aplikacije



|  |  |
| --- | --- |
| Naziv | Brisanje postavljene aplikacije |
| Aktor | Korisnik |
| Preduslovi | Aplikacija nije aktivna |
| Okidač | Korisnik je izabrao opciju za brisanje |
| Normalni tok | 1. Korisniku se prikazuje informacija da je aplikacija obrisana i aplikacija je uklonjena iz liste postavljenih aplikacija. |
| Alternativni tok |  |
| Izuzeci |  |
| Postuslovi | Korisnik je obrisao aplikaciju. |

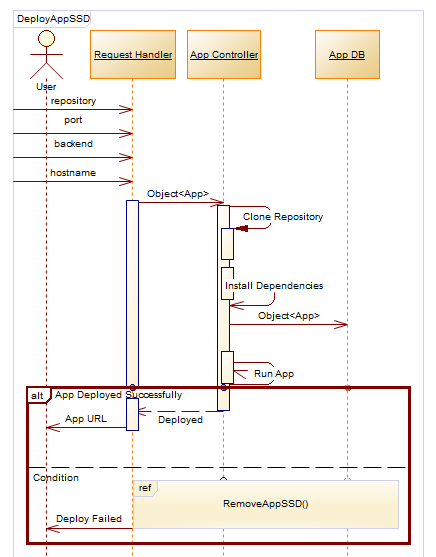
## Slučaj upotrebe promene parametara aplikacije



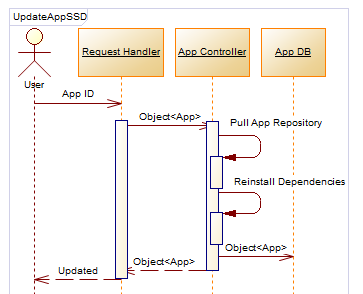
|  |  |
| --- | --- |
| Naziv | Izmena parametara postavljene aplikacije |
| Aktor | Korisnik |
| Preduslovi | Aplikacija nije aktivna |
| Okidač | Korisnik je izabrao opciju za izmenu parametara |
| Normalni tok | 1. Korisniku se prikazuje za interfejs sa unosnim poljima trenutnih parametara aplikacije. 2. Korisnik posle unošenja validnih podataka ima mogućnost da pošalje zahtev za promenu parametara. 3. Korisnik potvrđuje unešene parametre i šalje zahtev za promenu. 4. Korisniku se prikazuje informacija da su parametri promenjeni i novi parametri su azurirani u listi postavljenih aplikacija [IZ - 1]. |
| Alternativni tok | Korisniku se prikazuje poruka o nevalidnim podacima koje treba da izmeni. |
| Izuzeci | [IZ - 1] Korisnik je uneo nevalidne podatke. Korisnik se vraća na interfejs za promenu parametara. |
| Postuslovi | Korisnik je promenio parametre aplikacije. |

# Dijagrami sekvenci

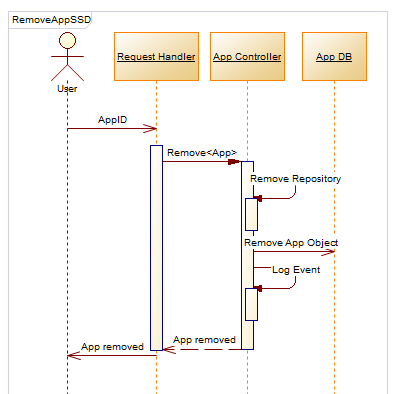
## Dijagram postavljanja aplikacije



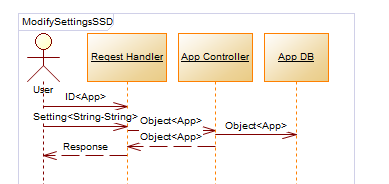
## Dijagram ažuriranja aplikacije



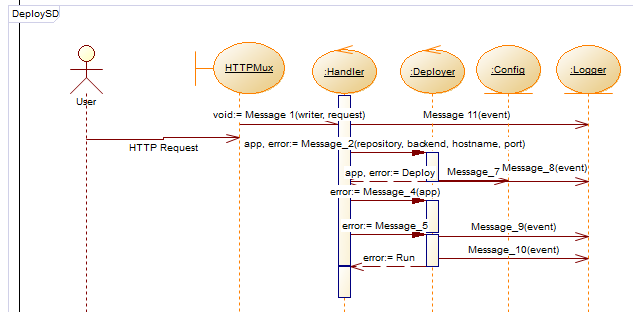
## Dijagram brisanja aplikacije



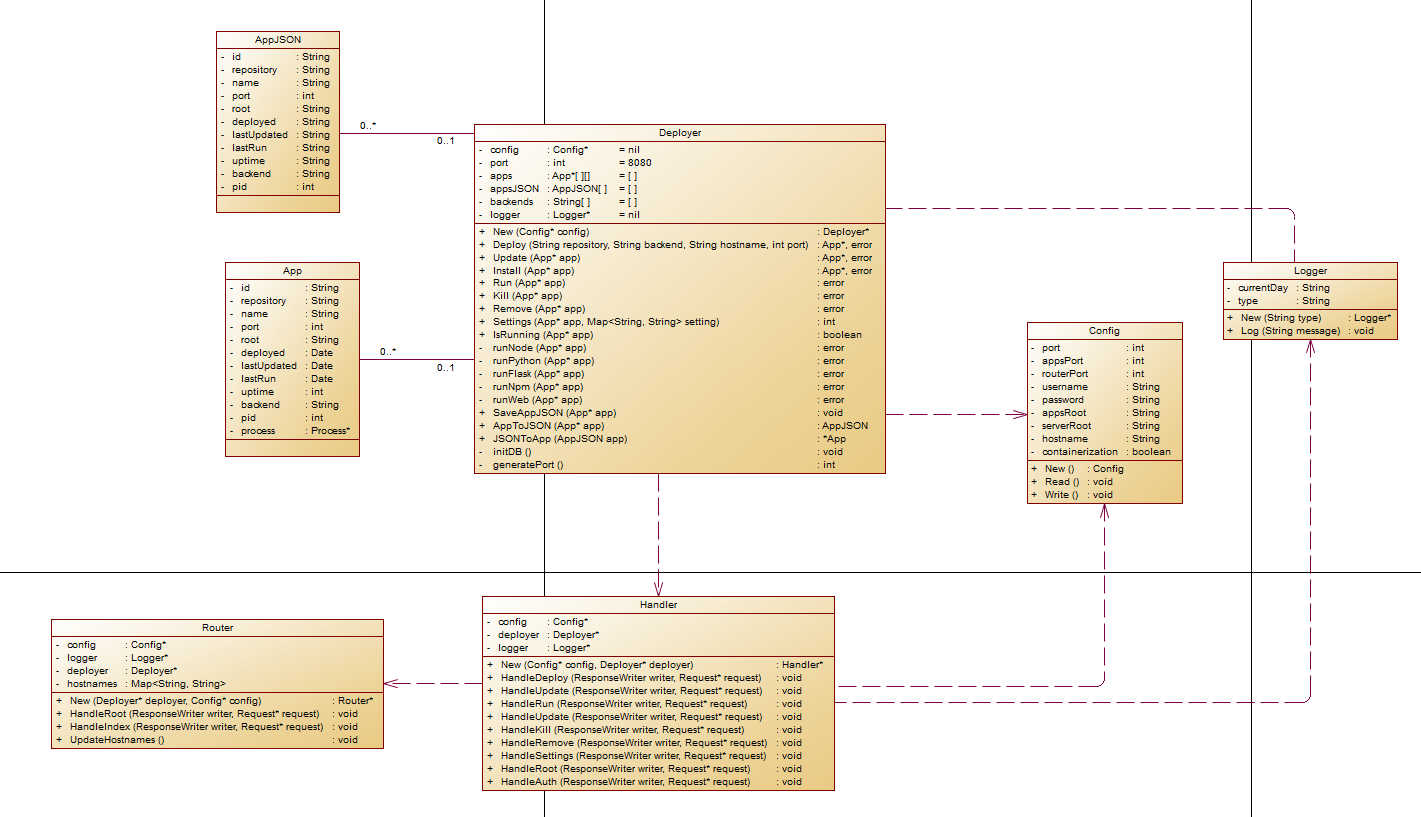
## Dijagram promene parametara aplikacije



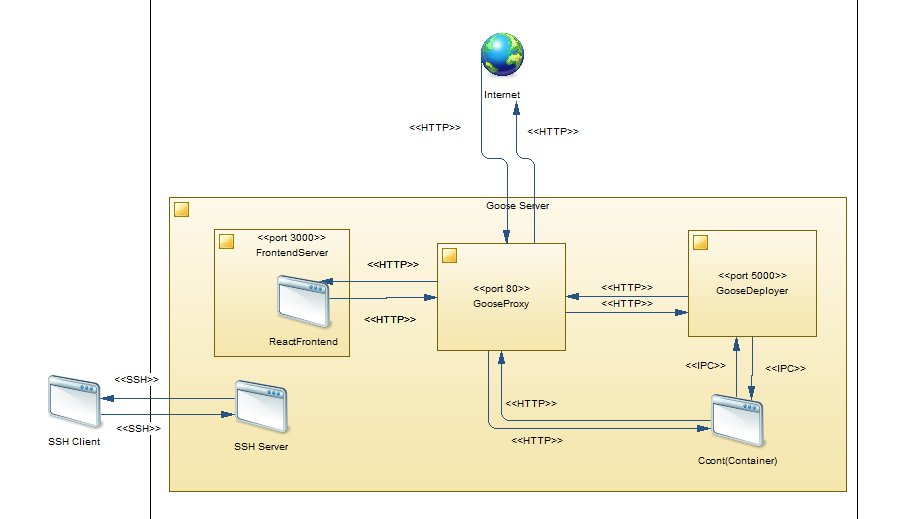
## Deteljan dijgram postavljanja aplikacije



## Klasni dijagram



# Arhitektura sistema

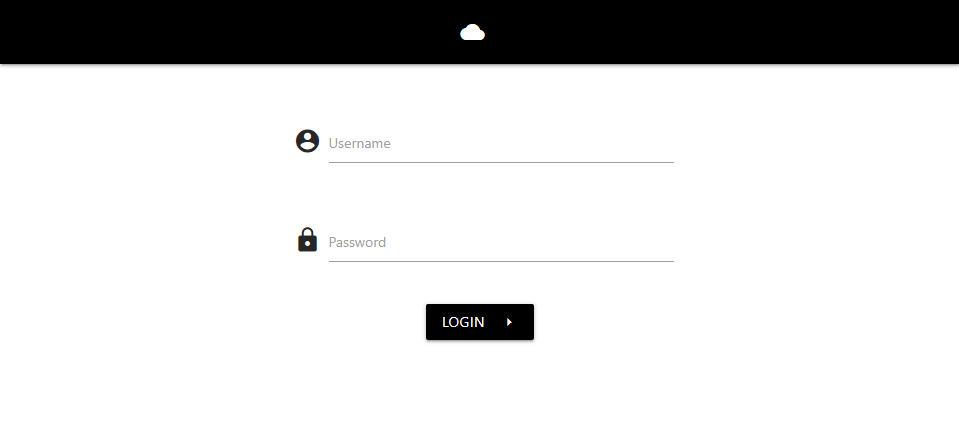


## Opis arhitekture sistema

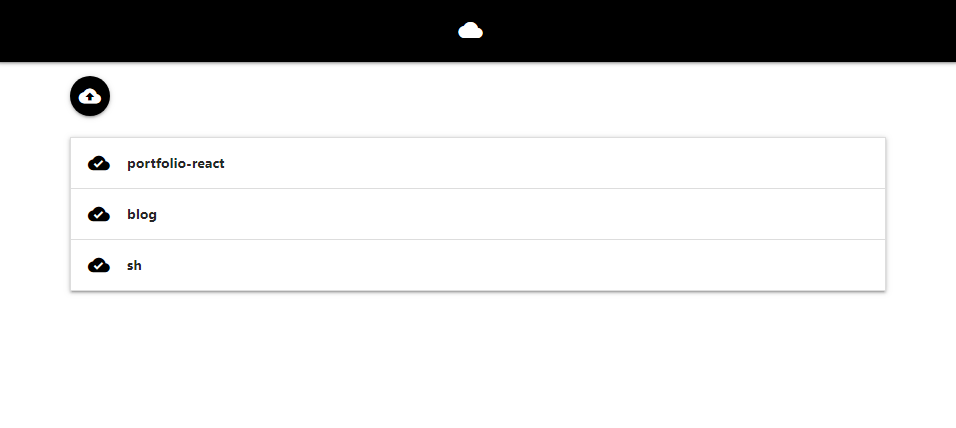
Za arhitekturu sistema odabran je repository patern arhitekture sistema gde se sve informacije cuvaju na jednom mestu a distribuirane su do korisnika preko servisno orijentisanog backenda koji sa klijentom komuncira preko HTTP-a. Cela arhitektura se nalazi iza proksi servera koji određuje da gde će se HTTP zahtev poslati u zavisnosti od hostname-a. Controller aplikacija (tj. njihovih kontejnera, Ccont) omogucava kontejnerima pristup internetu preko komunikacije sa proksijem. React frontemnd se takođe nalazi na odvojenom serveru i potpuno je autonoman. U suštini odžavamo ideju o modularnosti koju smo pomenuli ranije. Tako da na primer uklanjanjem frontenda ostaje web API koji moze koristiti bilo koji frontend interfejs ili pak neka druga aplikacija ili sistem.

# Korisnički interfejs

## Login forma

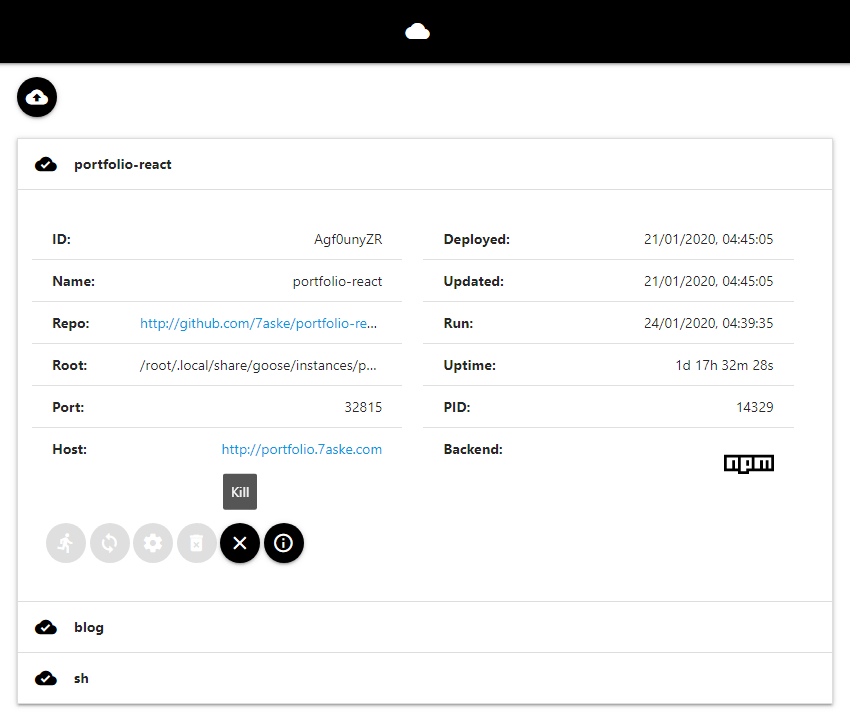


## Pregled postavljenih aplikacija



U listi vidimo postavljenje aplikacije. Stiklir u oblaku pored imena aplikacije ozačava da je aplikacija pokrenuta. Dugme iznad liste je za prostup interfejsu za deployment.

## Pregled informacija o pokrenutoj aplikaciji



U datom pregledu možemo videti sledece informacije o aplikacji:

**ID**

Nasumično generisan jedinstveni identifikator pokrenute aplikacije.

**Name**

Naziv aplikacije izvučen iz imena repozitorijuma koji je korišćen.

**Repo**

URL do Github ili sličnog repozitorijuma koji je korišćen za deployment.

**Root**

Lokacija aplikaciije na disku web servera

**Port**

Port na webserveru koji je rezervisan za aplikaciju

**Host**

URI hostname preko koga pristpupamo aplikaciji

**Deployed**

Datum i vreme deploymenta

**Updated**

Datum i vreme poslednjeg ažuriranja aplikacije

**Run**

Datum i vreme poslednjeg pokretanja aplikacije

**Uptime**

Vreme rada aplikacije od poslednjeg pokretanja

**PID**

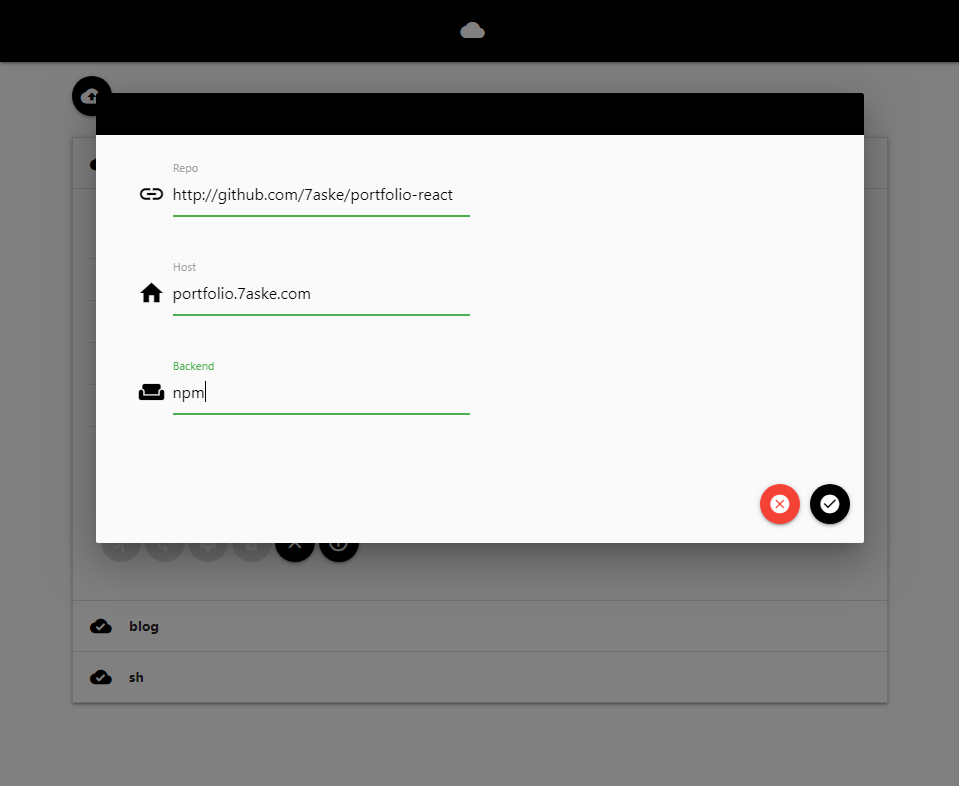
ID procesa na serveru koji je asociran aplikaciji

**Backend**

Bekend sistem koja je korišćen za deployment

U dnu liste vidimo kontrole koje predstavljaju interfejse za, redom, pokretanja ažuriranje, menjanje podešavanja, brisanje, gašenje i čitanje logova. Dugmeta za logout nema jer se svi korisnički podaci čuvaju u sesiji trenutno otvorenog tab-a tako da čim je tab zatvoren korisnik je automatski izlogovan. Interfejs se osvežava svakih 10 sekundi da bi korisnik imao koliko toliko realne informacije o status svojih aplkacija.

## Deployment interfejs



Na ovom interfejsu vidimo formu za postavljanje aplikacije na sistem.

**Repo**

URL do Github ili sličnog repozitorijuma koji je korišćen za deployment.

**Host**

URI hostname preko koga pristpupamo aplikaciji

**Backend**

Bekend sistem koja koristimo za deployment. Trenutno podržani sistemi su Node, Npm, Python i Flask.

# Testiranje

Testiranje modula proverava da li pojedinačne funkcije i komponente ispravno funkcionišu sa svim očekivanim tipovima ulaza, u skladu sa dizajnom komponente. Jedinično testiranje treba vršiti u kontrolisanom okruženju gde rezultati metoda mogu biti precizno proućen i biti im određena validnost.

Go programski jezik u sebi ima ugrađeno jedinično testiranje koje se veoma lako pokreće komandom go test i koja testira funkcije na nivou paketa. U ovom slučaju testiramo da li će nam program, preciznije funkcija za instalaciju izbaciti grešku ako koristimo nevalidan bekend koji nije podržan.

**package** deployer

**import** (

    "../internal/deployer"

    "../internal/instance"

    "testing"

)

func TestInstall(t \*testing.T) {

    bkend := "golang" *// invalid backend*

    inst := instance.New(

        "https://github.com/7aske/goose",

        "goose.7aske.com",

        instance.Backend(bkend))

    json := instance.ToJSONStruct(inst)

    err := deployer.Deployer.Install(json)

**if** err == **nil** {

        t.Errorf("Backend check failed, invalid backend %s was installed", bkend)

    }

}

U našem slučaju test prolazi i baca grešku 'backend not implemented', što je i očekivano.

PASS

ok \_/home/nik/Code/go/goose/test 0.004s

# Zaključak

Obzirom na to da je sistem već bio dizajniran pre nastanka ove dokumentacije a ponovnom revizijom usked analize za izradu iste došlo do potpunog re-dizajna koji je za rezulat imao mnogo bolji sistem primećujem da sva znanja dobjiena na ovom predmetu i tokom izrade ove dokumentacije će doprineti mnogoim drugim budućim projektima.