Collaborative Code Editor

Stefan Stankovic 18395

Uros Lazic 18229

Kontekst I cilj softverskog projekta

# Opis projekta

Real-time collaborative code editor je softver koji omogucava da vise korisnika istovremeno pisu kod na jednom projektu u stvarnom vremenu. Cilj projekta je poboljsanje timskog rada I efikasnosti kod deljenja I editovanja koda. Ovaj softver treba da omoguci korisnicima da istovremeno menjaju kod na razlicitim fajlovima projekta kao I mogucnost da prate promene u stvarnom vremenu.

Arhitekturno specificni zahtevi

Zbog prirode problema koji se resava, arhitekturno specificni zahtev je Client-Server arhitektura sa mogucnostima real-time komunikacije. Samim tim organizacija arhitekture je podeljena na:

* Klijentski deo → Jednostavan I intuitivan korisnicki interfejs za manipulaciju koda / fajlova
* Serverski deo → Zaduzen za obradu, skladistenje podataka ( fajlova ) I real-time komunikaciju
* Persistence deo → Zaduzen za trajno cuvanje podataka o projektima, fajlovima I kodu
* Cache deo (?) → Zaduzen za privremeno cuvanje podataka radi poboljsanja performansi

Glavni funkcionalni zahtevi

Glavni funkcionalni zahtevi su:

* Korisnicki interfejs → Jednostavan interfejs za manipulaciju koda I prikazivanja aktivnosti korisnika u editoru
* Real-time editovanje koda → Mogucnost da vise korisnika istovremeno menjaju/brisu/dodaju kod u odredjenim fajlovima, kao I automatsko pracenje I azuriranje za sve korisnike
* Trajno cuvanje informacija o projektima (?)

Ne-funkcionalni zahtevi

Glavni ne-funkcionalni zahtevi su:

* Performanse → Brz odaziv na promene u kodu I minimalno kasnjenje kod komunikacije izmedju korisnika
* Bezbednost → Autentifikacija I autorizacija korisnika
* Pouzdanost → Automatsko cuvanje stanja fajlova kako bi se izbegli gubitci podataka

Tehnicka I poslovna ogranicenja

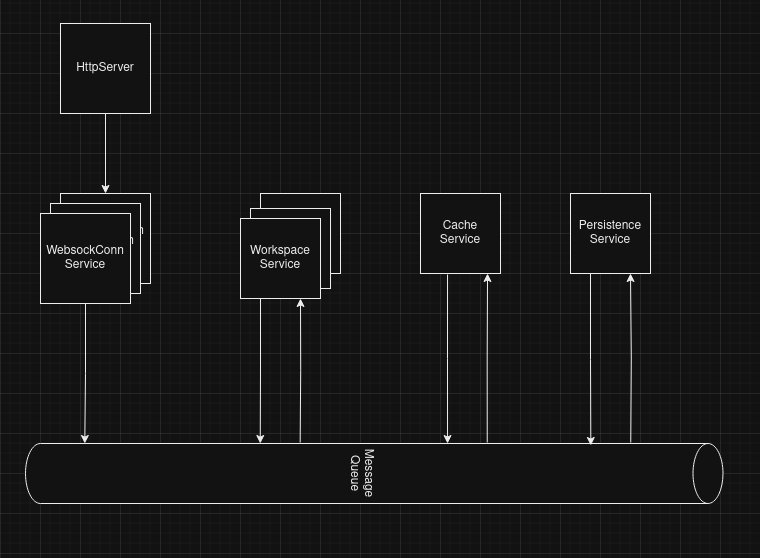
Arhitekturni dizajn

#### Arhitekturni obrazci

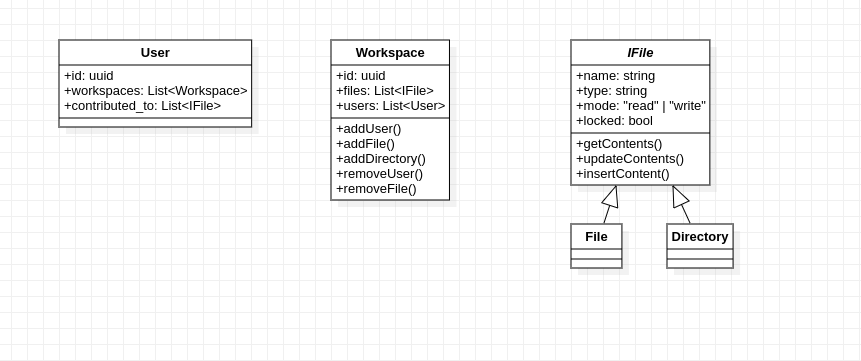
Arhitekturni obrazci koji ce biti korisceni su:

* Client-Server
* Repository
* Pub/Sub
* Service-Oriented (Microservice)

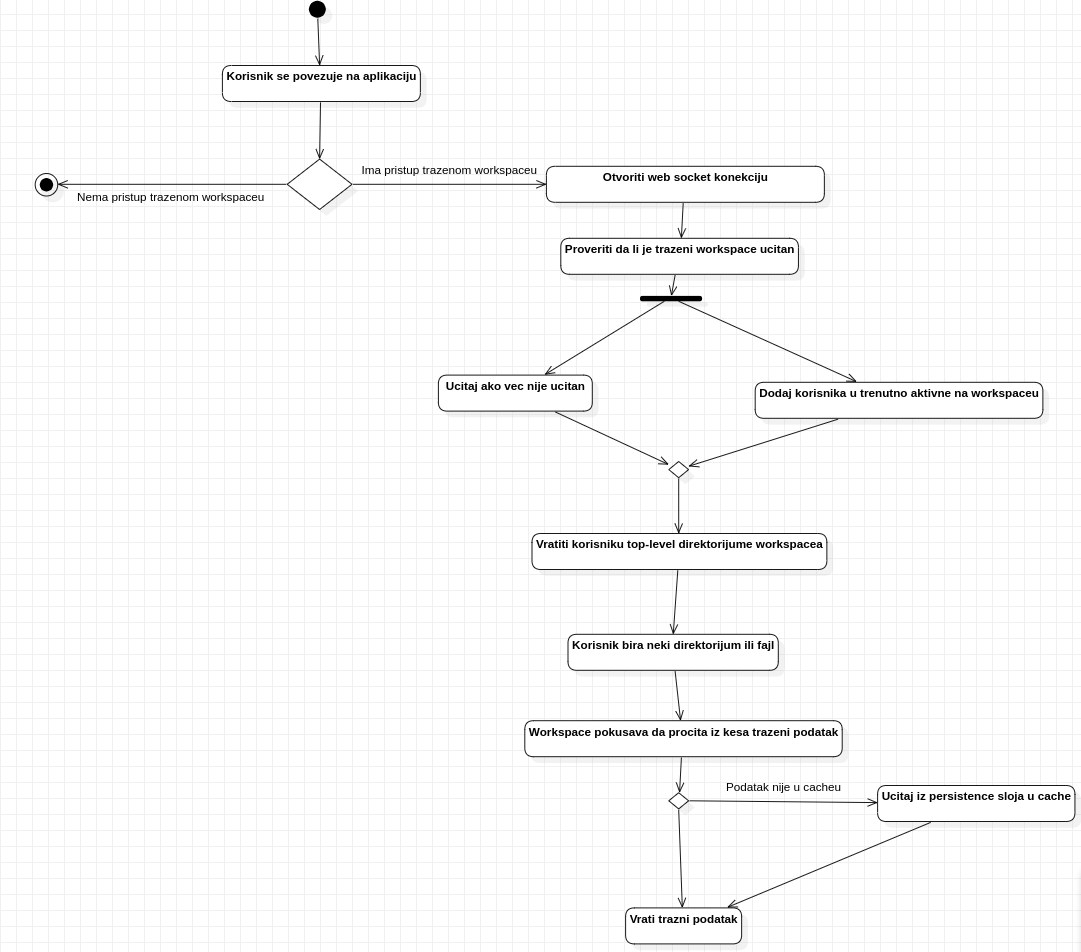
#### Generalna arhitektura

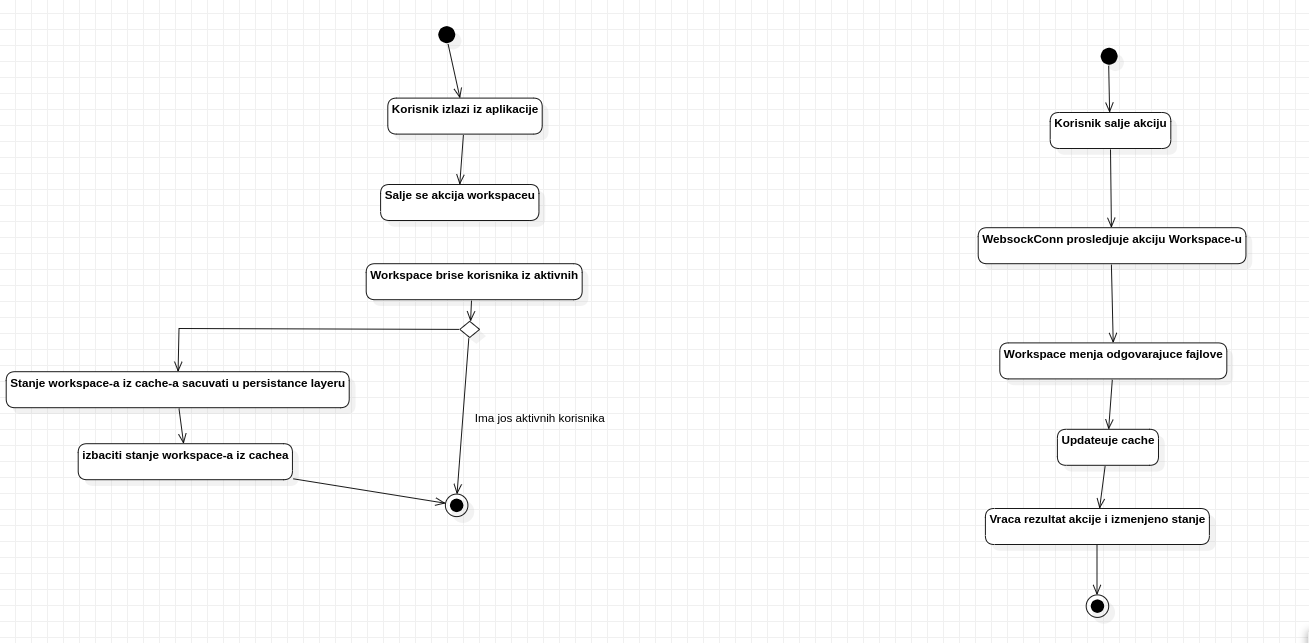


#### Klasni dijagram ( entiteti )



#### Dijagrami aktivnosti





Izabrani aplikacioni okviri ( frameworks ) I biblioteke

* Gin – Lightweight Go web framework, ne namece nikakve arhitekturne paterne, to ostavlja developerima aplikacije ( koristice se za Client-Server pattern, Pub/Sub )
* GORM – Golang Objektno Relacioni Maper, ne namece nikakve arhitekture paterne, to ostavlja developerima aplikacije ( korisice se za Repository pattern )
* Next-js ( React ) - ne namece nikakve arhitekturne paterne, to ostavlja developerima aplikacije