

1.1 Vežbe 23.12.2020.

Ideja ovog predmeta je da se napravi sistem za kontrolu verzija, po uzoru na github, gitlab ili neki drugi sajt koji radi istu stvar. Ono što treba da napravimo je nekakva platforma na kojoj imamo više različitih korisnika koje možemo podeliti po nekim timovima, grupama, gde postoje nekakvi projekti, taskovi i postoje milestones. Te taskove možemo da otvaramo, zatvaramo, menjamo stanja. Taskovi mogu da se grupišu u milestones , i taskovi naravno pripadaju nekom projektu. Takođe možemo da dodajemo ljude na projekat, tako da oni čine nekakav tim. Svaki task može da radi jedan ili više ljudi i taskovi mogu da prolaze kroz različita stanja.

1.1.1 Task

Razgovor na osnovu gitlaba. Ako odemo na prvi deo pod nazivom issues, ovde imamo različite varijante taskova koje mi možemo da dopustimo. Pored toga, ovde imamo različite filtere ili grupe taskov. Task može biti ili otvoren ili zatvoren ili možemo prikazati sve taskove. Ako otvorimo jedan task, unutar taska možemo da vidimo kada je task napravljen, koje mu je stanje, kom milestone-u pripada i možemo da dodamo različite tagove za taj task. Naravno, može da se pogleda i istorija, možemo da dodelimo neku težinu, koliko je task težak ili koliko je komplikovan. Možemo da dodamo informaciju kada je task otvoren , naravno prilikom zatvaranja taska mora se zabeležiti i to vreme. Na taj način možemo da računamo koliko prosečno vremena treba za taskove da budu otvoreni, ili koliko vremena treba da se taskovi zatvore, tako da možemo da iscrtamo nekakve grafike. Pored toga možemo da dodamo neki task, za koji moramo da dodamo ime, opis, kome se dodeljuje, u početku taj task nije dodeljen nikome. Ljudi kasnije uzimaju taskove. Mogu se dodati informacije o milestones, da li pripada ili ne pripada nekom, isto tako se mogu dodati neke labele. Labele se mogu ručno praviti, možemo da imamo nekakve predefinisane. U opisu taska možemo navesti neke proste informacije i naravno možemo zakačiti nekakav fajl, ako smo bili u diskusiji sa menadžerima ili sa klijentom, nešto smo crtali, dijagrame to možemo okačiti u okviru tog opisa. Kao fajl se može zakačiti slika, beleška u vidu tekstualnog fajla.

1.1.2 Milestone

Što se tiče milestones, oni grupišu takove u nekakvu celinu. Možemo dodati različite informacije, koliko informacija imamo, koji issues su zatvoreni i koji ljudi učestvuju na tom milestone i tako dalje. Možemo specificirati grafiku, odnosno koliko issues imamo i koliko njih je završeno. Tako da uvek možemo da pratimo progres dokle je taj milestone stigao. Mora se specificirati i kraj milestone. Tako da onda možemo da ispratimo procenat završenih poslova, da li možda taskove organizujemo da ima previše taskova unutar milestone. Možda treba praviti manje milestone i manje taskova unutar toga i naravno milestone može da bude otvoren ili zatvoren, u smislu da li još uvek nije isteklo vreme za taj milestone ili jeste. Moguće je imati i milestone koji nije vremenski ograničen, npr ispravke grešaka ili testiranje, koje traje značajno duže i uglavnom se ne može ograničiti vremenski. Ako hoćemo da kreiramo novi milestone, treba nam naslov, opis, može se zakačiti nekakav fajl ista stvar kao i sa taskovima, imamo početno i krajnje vreme.

1.1.3 Labele

Mogu se kreirati labele. Labele su nekakvi tagovi koji označavaju ili samo dodatno daju nekakvu semantiku našim taskovima ili milestoneovima. Taskove možemo da kreiramo, pretražujemo, pregledamo ili obrišemo. Milestones isto tako možemo da kreiramo, brišemo, pretražujemo, crud operacije isto treba implementirati. Labele se isto mogu pregledati, dodati, obrisati, odnosno crud operacije treba da implementiramo. Labele možemo da dodamo na taskove, labele možemo da dodamo na milestones. Taskovi se sastoje od osnovnih operacija, plus ljudi koji se nalaze na tom tasku, plus labele. Milestones se sastoje od ljudi, taskova i labela.

Što se tiče različitih stanja, ne moramo da komplikujemo dovoljno je da imamo open, close, done, in progress. Bilo bi dobro omogućiti da se prati istorija taskova, odnosno za svaki task kako je on u vremenu, znači može da je kreiran, može da ga je neko preuzeo, neko je krenuo da radi, pa je završen. Treba da imamo informaciju o životnom ciklusu taska.

Na osnovu svih tih stvari, možemo da crtamo nekakve grafike. Tako da možemo da pratimo koliko taskova je otvoreno, koliko zatvoreno, koji ljudi imaju najviše komita, koji ljudi imaju najviše otvorenih taskova, koji imaju najviše zatvorenih taskova, koliko je prosečno vreme po taskovima otvoreno, zatvoreno, možemo da vidimo koji su najaktivniji korisnici, možemo da vidimo istoriju komita, kada najviše komituju, kada najmanje, broj komita po čoveku. Različiti grafici mogu na osnovu toga da se kreiraju.

Ono što ne moramo da implementiramo jesu funkcionalnosti vezane za git, u smislu da naša aplikacija ne treba da ima istoriju izmena, nećemo praviti git storage. Međutim ono što treba da vidimo jesu komiti i istorija komita. Postoje dva načina kako to možemo da implementirati. Prva varijanta bi bila da u delu koji se zove commit, možemo iz našeg repozitorijuma da prosto kopiramo has nekog komita i informacije. Opcija dva je da iskoristimo neku od postojećih biblioteka i da svučemo komite direktno u našu aplikaciju. Ne treba da čuvamo kod, da pravimo razlike, da upravljamo granama. Ono što je potrebno da se uradi za veću ocenu je da se omogući prikaz grana koje imamo, koje ljude imamo po granama. Opet postoje dve opcije kako to možemo da implementiramo. Možemo ručno da dodajemo te grane ili da iskoristimo neku biblioteku koja je sposobna da povuče grane sa gitlaba, githuba, itd. Moramo omogućiti da vidimo koji ljudi su na kom projektu i onda opet možemo da crtamo različite grafike, da vidimo ko kako komituje, koje je najbolji. Istorija grana je komplikovanija za uraditi, to isto može da se uradi tako što ćemo povući neku od biblioteka vezano za git. Možemo da imamo i viki stranicu gde možemo da upišemo osnovne informacije vezano za projekat.

1.1.4 Modelovanje

Što se tiče modelovanja, potrebno je da koristimo uml dijagrame, odnosno dijagram klasa. Dobro je da se sačuva i kao slika, kako bi moglo da se pogleda.

1.1.5 Testiranje

Na kraju moramo da napišemo testove za aplikaciju. Svaki član tima treba da istestira u potpunosti jednu funkcionalnost koju je radio. U smislu kreiranje taska, otvaranje, zatvaranje, promena ljudi na projektu, tasku. To se može posmatrati kao jedan test.

Stvar oko koje ne moramo previše da se trudimo je security, autentifikacija, autorizacija, na to ne treba da trošimo vreme pošto asistent to neće toliko pregledati i ocenjivati. Druga stvar je stilizovanje i dizajn, nije bitno da bude lepo i da se posvetimo detaljima i na to se ne dobijaju dodatni poeni.

1.1.6 Pitanje

1. Da li može pojašnjenje kako bi izgledalo importovanje projekata u taj naš zadatak?

Nije potrebno da importujemo projekte u smislu da napravimo vezu između git repozitorijuma i naše aplikacije. Mi možemo da napravimo new project u našoj aplikaciji. Međutim, ono što treba da importujemo su recimo komiti, postoje razne biblioteke, i github i gitlab imaju svoj api gde mi možemo iz nekog projekta da izvučemo komite. Ali ni to ne moramo, ono što možemo je da napravimo nekakav deo naše aplikacije koji se zove commits i onda možemo ručno da napravimo commit i prepíšemo informacije. Ili da prepíšemo informacije ili commit povučemo sa nekog repozitorijuma.

2. Da li je dozvoljeno za izradu viki stranice koristiti neku biblioteku za neki editor?

Naravno, ne moramo da pravimo stavri od nule, ako znamo za neku biblioteku koja nam omogućava da implementiramo bilo koji deo ovoga znatno brže, jednostavnije to možemo koristiti. Na odbrani projekta treba samo da kažemo šta smo koristili i zašto. Bilo koja biblioteka može, samo treba da se napomene i kaže zašto to koristimo.

3. Korisnik pristupi aplikaciji, i tu može da se registruje ili uloguje na svoj nalog. I kada pristupi svom nalogu ima listu nekih projekata, i on može da uđe na te projekte i vidi taskove, milestones, grafike itd. Znači on ima dugme dodaj task, labelu, ..

Tako je. Sve te taskove, lebele, projekte sve to mi kreiramo. Sve se dodaje ručno, jedino što možemo da sinhronizujemo su komiti, ali opet ne mora ni to, može i to ručno. To se može ostaviti za kraj, ako budemo imali vremena. Ali ostale stavri treba da napravimo, znači prikaz korisnika, taskova, milestones, labela, viki, dodavanje, zatvaranje, otvaranje taskova, milestones, i grafike koji su gore spomenuti. Za crtanje tih grafikona možemo iskoristiti već neke postojeće biblioteke.

Postoji biblioteka koja nam može pomoći da izvučemo iz git repozitorijuma, komite, grane. I onda možemo to da probamo da implementiramo, jedino jednostavnije je to sve izvući iz tog git. Fajla i to može lokalno da bude kod nas i onda je potrebno da ga isparsiramo. Ali ono što je nezgodnije je to crtanje, ako nismo vični css i tim stavrima onda je ovo patnja. Ali ako imamo vremena možemo probati, ali to nije obavezno.

Što se tiče korisnika, potrebno je omogućiti da se korisnici za svaki projekat vide, ali prava koja postoje na gitlab-u, to ne treba, sistem prava ne treba. Treba da omogućimo da pogledamo korisnika, da vidimo njegovo ime, prezime, sliku, projekte na kojima je, poslednji komit kada je bio. Grafik koliko je bio aktivan ne mora.

Rok je kraj februara ili početak marta, ali će to još uvek javiti. Postoje i kasnije rokovi, kada god budemo hteli, ali je jedan zvaničan. Postoji jedna ili 2 kontrolne tačke, na kojima ćemo pokazati šta smo radili, ako imamo neki problem, ..

O dockeru ćemo detaljnije pričati sledeći put, ali je ideja da naš kod bude zapakovan u kontejner, da uradimo push na docker hub, i da onda uradimo neku varijantu diploja, što se toga tiče tu ćemo imati nekoliko varijanti koje možemo da izaberemo, može lokalno, neki cloud provajder koji će dopustiti da radimo besplatno. Ali o svemu tome detaljnije sledeće nedelje.