

Zadaci za vežbe iz predmeta

Softverski obrasci i komponente

GitLab i git aktivnost

SADRŽAJ

1	Kreiranje timova i repozitorijuma	2
1.1	Kreiranje timova	2
1.2	Kreiranje repozitorijuma	2
2	GitLab aktivnost	2
2.1	Kreiranje milestone-a	2
2.2	Kreiranje labela	3
2.3	Kreiranje issue-a	3
3	Git aktivnost	4
3.1	Priprema inicijalnog projekta	4
3.1.1	Komit inicijalnog projekta	4
3.1.2	Preuzimanje inicijalnog projekta	5
3.2	Kreiranje grana	5
3.3	Rešavanje konflikata	6
3.4	Zatvaranje issue-a iz komita	8

1 KREIRANJE TIMOVA I REPOZITORIJUMA

1.1 KREIRANJE TIMOVA

Rasporediti se u timove od dva do tri člana. Timovi ne moraju da budu isti kao za projektni zadatak. Odrediti ko će biti prvi član tima, koji će da kreira novi repozitorijum, a ko će da bude drugi član tima koji će da uradi inicijalni komit projekta.

1.2 KREIRANJE REPOZITORIJUMA

Prvi član tima treba na GitLab sajtu da kreira novi repozitorijum tako što otvori stranicu:

Projects->Your projects

i izbere opciju New project.

Potrebno je uneti ime projekta i pritisnuti Create project opciju. Otvoriti kreiran projekat i otići na opciju:

Settings -> Members

i dodati drugog i trećeg člana u repozitorijum, dodeliti im prava Maintainer.

2 GITLAB AKTIVNOST

2.1 KREIRANJE MILESTONE-A

U kreiran projekat potrebno je da svaki član tima doda nove milestone-ove tako što ode na opciju: Issues->Milestones

i izabere opciju New milestone. Svaki član tima treba da doda poseban milestone:

- Prvi član tima
 - za naziv milestone-a da stavi Razvoj plugina
 - za opis milestone-a da stavi Razvoj plugina za učitavanje i vizuelizaciju
 - za početni datum milestone-a trenutni datum, a za krajnji datum da stavi datum nakon nedelju dana od početnog datuma
- Drugi član tima
 - za naziv milestone-a da stavi Razvoj zajedničkih funkcionalnosti
 - za opis milestone-a da stavi Razvoj zajedničkih funkcionalnosti Core modula
 - za početni datum milestone-a isti datum koji je prvi član tima stavio kao krajnji datum, a za krajnji datum da stavi datum nakon nedelju dana od svog početnog datuma
- Treći član tima
 - za naziv milestone-a da stavi Provera projekta
 - za opis milestone-a da stavi Testiranje funkcionalnosti i dodavanje class i component diagrama

- za početni datum milestone-a isti datum koji je drugi član tima stavio kao krajnji datum, a za krajnji datum da stavi datum nakon nedelju dana od svog početnog datuma

Potvrditi kreiranje milestone-a sa Create milestone. Svaki član tima treba da proveriti da može da vidi sve milestone-ove.

2.2 KREIRANJE LABELA

Za svaki issue se može dodeliti labela koja određuje kakav tip issue se kreira: enhancement, bug, critical ili neki drugi tip. Za jedan issue se može postaviti više različitih labela.

Drugi član tima treba da otvori stranicu za labele sa opcijom:

Issues->Labels

i kreira početni skup labela pritiskom dugmeta Generate a default set of labels.

2.3 KREIRANJE ISSUE-A

Za kreiranje novih issue-a otići na stranicu: Issues->List
i pritisnuti dugme New issue.

- Prvi član tima
 - za naziv issue-a da stavi Dodati novi modul modul1.py
 - za opis issue-a da stavi Dodati novi modul modul1.py i ubaciti kod za ispis
 - sebi da dodeli issue za polje Assignee
 - za milestone izabрати Razvoj plugina
 - za labele izabрати enhancement
 - za datum staviti trenutni datum
- Drugi član tima
 - za naziv issue-a da stavi Dodati novi modul modul2.py
 - za opis issue-a da stavi Dodati novi modul modul2.py i ubaciti kod za ispis
 - sebi da dodeli issue za polje Assignee
 - za milestone izabрати Razvoj plugina
 - za labele izabрати enhancement
 - za datum staviti trenutni datum
- Drugi član tima treba da doda još jedan issue za inicijalni komit
 - za naziv issue-a da stavi Dodati inicijalni projekat
 - za opis issue-a da stavi Dodati novi projekat Projekat1 i modul glavni.py
 - sebi da dodeli issue za polje Assignee

- za milestone izabrati Razvoj plugina
- za labele izabrati enhancement i critical
- za datum staviti trenutni datum
- Treći član tima
 - za naziv issue-a da stavi Dodati novi modul modul3.py
 - za opis issue-a da stavi Dodati novi modul modul3.py i ubaciti kod za ispis
 - sebi da dodeli issue za polje Assignee
 - za milestone izabrati Razvoj plugina
 - za labele izabrati enhancement
 - za datum staviti trenutni datum

Svaki član tima treba da proveri da može da vidi sve issue na stranici:

Issues->List

Svaki član tima treba da otvori svoj issue i da proveri jedinstveni identifikator svog issue-a. Jedinstveni identifikator issue-a je određen oznakom #broj_issue-a. Jedinstveni identifikator zapamtiti da bi mogli da referencirate issue iz komita i da zatvarate issue iz komita.

3 GIT AKTIVNOST

3.1 PRIPREMA INICIJALNOG PROJEKTA

3.1.1 KOMIT INICIJALNOG PROJEKTA

Pre nego što svi članovi krenu sa razvojem svojih funkcionalnosti, potrebno je da se napravi inicijalni projekat i pošalje na GitLab repozitorijum, kako bi ostali članovi tima mogli da preuzmu taj projekat u koji će dodavati nove funkcionalnosti.

- Drugi član tima
 - treba da klonira GitLab repozitorijum sa komandom **git clone https_link_do_repozitorijuma**
 - unutar kloniranog repozitorijuma treba da napravi novi Pycharm projekat sa nazivom Projekat1
 - da doda novi modul glavni.py unutar projekta
 - da u modulu doda naredbu za ispis na konzoli sa tekстом "Inicijalni projekat"
 - u direktorijum projekta dodati i .gitignore fajl
 - konfigurišati ime i email za klonirani repozitorijum sa komandama:


```
git config --local user.name "ime prezime"
git config --local user.email "email adresa"
```
 - sve dodate fajlove pripremiti za komit sa komandom **git add .**
 - komitovati ove izmene sa komandom **git commit -m "Inicijalni projekat"**

- poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**
- otvoriti issue sa nazivom Dodati inicijalni projekat i zatvoriti issue pritiskom dugmeta Close issue

3.1.2 PREUZIMANJE INICIJALNOG PROJEKTA

Nakon što je drugi član komitovao i poslao taj komit za inicijalni projekat na GitLab repozitorijum, ostali članovi tima treba da preuzmu te izmene sa komadnom:

git clone https_link_do_repozitorijuma

Unutar kloniranog repozitorijuma potrebno je podesiti ime i email sa komandama:

git config --local user.name "ime prezime"

git config --local user.email "email adresa"

3.2 KREIRANJE GRANA

Kada svi članovi imaju lokalni repozitorijum sa inicijalnim projektom, mogu dodati potrebne module, navedene u kreiranim issue-ima.

- Prvi član tima
 - kreirati novu granu razvoja sa komadnom **git branch grana1**
 - proveriti koje sve grane postoje sa komandom **git branch -a**
 - preći na kreiranu granu sa komandom **git checkout grana1**
 - u direktorijum projekta dodati novi modul modul1.py i u njemu naredbu za ispis na konzoli sa tekstom "Ispis sa grane grana1"
 - pripremiti ove izmene za komit sa komandom **git add .**
 - komitovati ove izmene sa komandom **git commit -m "Dodat modul modul1.py #jedinstveni_identifikator_issue-a_koji_je_dodeljen_prvom_članu"**
 - preći na master granu sa komandom **git checkout master**
 - spojiti granu grana1 sa master granom koristeći komandu **git merge grana1**
 - poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**
- Drugi član tima
 - kreirati novu granu razvoja sa komadnom **git branch grana2**
 - proveriti koje sve grane postoje sa komandom **git branch -a**
 - preći na kreiranu granu sa komandom **git checkout grana2**
 - u direktorijum projekta dodati novi modul modul2.py i u njemu naredbu za ispis na konzoli sa tekstom "Ispis sa grane grana2"

- pripremiti ove izmene za komit sa komandom **git add**.
 - komitovati ove izmene sa komandom **git commit -m "Dodat modul modul2.py #jedinstveni_identifikator_issue-a_koji_je_dodeljen_drugom_članu"**
 - preći na master granu sa komandom **git checkout master**
 - spojiti granu grana2 sa master granom koristeći komandu **git merge grana2**
 - poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**
- Treći član tima
 - kreirati novu granu razvoja sa komandom **git branch grana3**
 - proveriti koje sve grane postoje sa komandom **git branch -a**
 - preći na kreiranu granu sa komandom **git checkout grana3**
 - u direktorijum projekta dodati novi modul modul3.py i u njemu naredbu za ispis na konzoli sa tekstem "Ispis sa grane grana3"
 - pripremiti ove izmene za komit sa komandom **git add**.
 - komitovati ove izmene sa komandom **git commit -m "Dodat modul modul3.py #jedinstveni_identifikator_issue-a_koji_je_dodeljen_trećem_članu"**
 - preći na master granu sa komandom **git checkout master**
 - spojiti granu grana3 sa master granom koristeći komandu **git merge grana3**
 - poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**

Kada budete slali izmene na GitLab repozitorijum, član tima koji prvi bude pokrenuo push komandu uspeće da pošalje izmene na GitLab, a ostali članovi tima dobiće poruku da postoje izmene koje nemaju u lokalnom repozitorijumu i moraće prvo da preuzmu izmene sa komadnom:

git pull

a zatim ponovo da pokušaju da pošalju izmene sa komandom: **git push -u origin master**

Svaki član tima je potrebno da ponovi pull i push komande dok ne bude video modul1.py i modul2.py, a ako ima tri člana u timu onda bi trebalo da bude vidljiv i modul3.py.

Otvoriti issue-e na GitLab i proveriti da li su za odgovarajuće issue-e prikazani komiti iz kojih su referencirani.

3.3 REŠAVANJE KONFLIKATA

Konflikti nastaju kada menjate u istom modulu istu liniju u kodu.

- Prvi član tima

- u modulu glavni treba da doda novu naredbu za ispis na konzoli sa tekstem "Prvi clan tima", ispod već postojeće naredbe za ispis sa tekstem "Inicijalni projekat"
- pripremiti ovu izmenu za komit sa komandom **git add**.
- komitovati ovu izmenu sa komandom **git commit -m "Dodat ispis u modul glavni.py prvi clan"**
- poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**
- Drugi član tima
 - u modulu glavni treba da doda novu naredbu za ispis na konzoli sa tekstem "Drugi clan tima", ispod već postojeće naredbe za ispis sa tekstem "Inicijalni projekat"
 - pripremiti ovu izmenu za komit sa komandom **git add**.
 - komitovati ovu izmenu sa komandom **git commit -m "Dodat ispis u modul glavni.py drugi clan"**
 - poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**
- Treći član tima
 - u modulu glavni treba da doda novu naredbu za ispis na konzoli sa tekstem "Treci clan tima", ispod već postojeće naredbe za ispis sa tekstem "Inicijalni projekat"
 - pripremiti ovu izmenu za komit sa komandom **git add**.
 - komitovati ovu izmenu sa komandom **git commit -m "Dodat ispis u modul glavni.py treci clan"**
 - poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**

Kada pokušate da pošaljete izmene, samo član tima koji prvi pokrene push komandu uspeće da pošalje izmene. Ostali članovi će morati da preuzmu izmene sa pull komandom, a zatim će verovatno dobiti poruku da je nastao Conflict. Da biste razrešili konflikt:

- Svaki član tima kod kojeg postoji konflikt
 - izmeni fajl u kojem postoji konflikt, tako da ostane kod bez dodatnog teksta koji dodaje git da bi označio linije koje su izazvale konflikt
 - potvrditi da je konflikt razrešen i pripremiti ove izmene za komit sa komandom **git add**.
 - komitovati ove izmene sa komandom **git commit -m "Razresen konflikt"**
 - poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**

Korake za pull, push i razrešavanje konflikata je potrebno ponoviti sve dok svi članovi tima ne budu videli sve komande za ispis u modulu glavni.py.

3.4 ZATVARANJE ISSUE-A IZ KOMITA

Kada završite sa razvojem funkcionalnosti navedene u odgovarajućem issue-u, možete taj issue zatvoriti sa posebnom komandom za zatvaranje issue direktno iz poruke samog komita. Komadne koje možete navesti za zatvranje issue su:

- Close, Closes, Closed, Closing, close, closes, closed, closing
- Fix, Fixes, Fixed, Fixing, fix, fixes, fixed, fixing
- Resolve, Resolves, Resolved, Resolving, resolve, resolves, resolved, resolving

i zatim se navodi jedan ili više jedinstvenih identifikatora issue-a razdvojenih zarezom. Za dodatna objašnjenja pogledati https://docs.gitlab.com/ee/user/project/issues/automatic_issue_closing.html.

- Prvi član tima
 - u modulu modul1.py dodati novu naredbu za ispis na konzoli sa tekstom "Završen modul1.py"
 - pripremiti ovu izmenu za komit sa komandom **git add** .
 - komitovati ovu izmenu sa komandom **git commit -m "Završen modul1.py closes #jedinstveni_identifikator_issue-a_koji_je_dodeljen_prvom_članu"**
 - poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**
 - proveriti da je zatvoren issue sa nazivom Dodati novi modul modul1.py
- Drugi član tima
 - u modulu modul2.py dodati novu naredbu za ispis na konzoli sa tekstom "Završen modul2.py"
 - pripremiti ovu izmenu za komit sa komandom **git add** .
 - komitovati ovu izmenu sa komandom **git commit -m "Završen modul2.py closes #jedinstveni_identifikator_issue-a_koji_je_dodeljen_drugom_članu"**
 - poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**
 - proveriti da je zatvoren issue sa nazivom Dodati novi modul modul2.py
- Treći član tima
 - u modulu modul3.py dodati novu naredbu za ispis na konzoli sa tekstom "Završen modul3.py"
 - pripremiti ovu izmenu za komit sa komandom **git add** .

- komitovati ovu izmenu sa komandom **git commit -m "Završen modul1.py closes #jedinstveni_identifikator_issue-a_koji_je_dodeljen_trecem_članu"**
- poslati ovaj komit na GitLab repozitorijum sa komandom **git push -u origin master**
- proveriti da je zatvoren issue sa nazivom Dodati novi modul modul3.py

Član koji prvi pokrene push komandu će uspeti da pošalje izmene, a ostali članovi tima moraju prvo da preuzmu nove izmene sa pull komandom i zatim da opet pokušaju da pošalju sa push komandom.