

Temporalno upravljanje dokumentima
(verzioniranje, XML dokumenti, napredne
mogućnosti pretraživanja) - Temporalne +
Prostorne baze podataka - PostgreSQL +
web grafičko sučelje po želji
Seminarski rad

Nikola Buković
br. indeksa: 44443/15-R

Mentor:
Doc. dr. sc. Markus Schatten

Varaždin, 27. lipnja 2016.

Sadržaj

1	Uvod	1
2	Temporalne i popopćene baze podataka	2
3	Model podataka	3
4	Implementacija	5
5	Primjer korištenja	9
6	Zaključak	14
7	Literatura	15

Poglavlje 1

Uvod

Cilj ovog seminarskog rada je napraviti aplikaciju za verzioniranje. Aplikaciju je potrebno napraviti korištenjem popćenih i temporalnih baza podataka. Kako bi izradio bazu podataka korišten je alat MicroOlapu, te je model baze podataka prikazan iz MicroOlapa, server je lokalno na Pgadmin-u, razvojno okruženje za programiranje php-a je NetBeans, te su korišteni HTML i CSS za izgled stranice. Sustav nam omogućuje da kroz vrijeme pratimo svoje projekte, ako ažuriramo projekt, stara verzija projekta odlazi u tablicu verzija, dok se novo ažurirana verzija tada nalazi u tablici projekt, tako da se u svakom trenutku možemo vratiti na prijašnju verziju. Kako bi uveli popćene baze podataka korišten je vlastiti tip, detaljnije će biti pojašnjeno u kasnijem djelu rada. Temporalne baze podataka koristimo praćenjem vremena unosa projekta, te ažuriranjem projekta gdje se onda u verziji zapisuje novo vrijeme tako da znamo kada je bila izmjena.

Poglavlje 2

Temporalne i popopćene baze podataka

Temporalne baze podataka

Skoro pa u svakoj bazi podataka pohranjuje se trenutna vrijednost, ako se ta vrijednost kroz vrijeme mijenja gubi se ona prva originalna vrijednost, ponekad bi nam te prve vrijednosti dobro došle, i htjeli bi se vratiti na njih, u tom slučaju pomažu temporalne baze podataka da kroz vrijeme pratimo kako se neka vrijednost mijenjala u bazi.

Temporalne baze podataka su baze podataka koje imaju ugrađene vremenske aspekte, te tako omogućavaju rad s takvim podacima putem prilagođenog jezika (temporal SQL). Aspekti obično uključuju tri vremena, a to su:

- Validno vrijeme-koje na govori u kojem je vremenskom periodu neka činjenica istinita

- Transakcijsko vrijeme- to je vrijeme tijekom kojeg je činjenica pohranjena

- Bitemporalno vrijeme- kombinacija validnog i transakcijskog vremena

Vremenski podaci koji su mogući u PostgreSQL-u su timestamp, timestampz, interval, date i time te timetz.

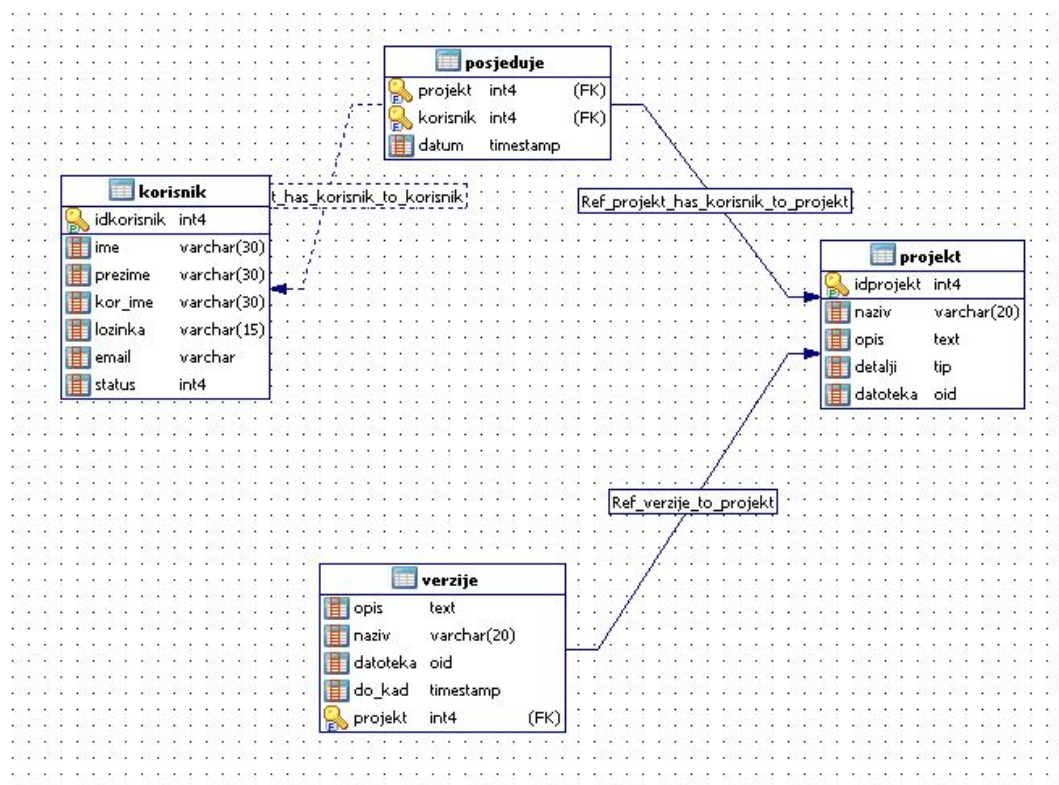
Popopćene baze podataka omogućavaju nam korištenje složenijih objekata, isto tako omogućuju nam da sami napravimo željeni tip.

Objekti koje možemo koristiti su: BLOB koristimo ih za pohranu binarnih podataka, za takve objekte koristimo tip podatka OID, u ovome radu je taj tip korišten. Sljedeće se mogu koristiti jednodimenzionalna i dvodimenzionalna polja, pobrojene vrijednosti ENUM, te na kraju složeni tipovi kod kojih složimo svoj tip podatka, također složeni tipovi su korišteni u radu.

Poglavlje 3

Model podataka

Model baze za sustav verzioniranja sastoji se od 3 jaka entita i jednog slabog. Jaki entiteti su korisnik, verzije, projekt te kao slabi entitet posjeduje. U entitet korisnik nalaze se atributi koji opisuju korisnika, te su nam oni potrebni za prijavu na aplikaciju. Korisnik može kreirati više projekta, te također na projektu može biti više korisnika, kada je korisnik kreirao projekt, u tablicu posjeduje dodaje su ključevi, te se zapisuje datum. Ako ažuriramo projekt na novu verziju, tada se stara verzije seli u entitet verzije, što je ostvareno pomoću trigera koji će biti prikazan u radu, te se tamo zapisuje atribut do kad da znamo do kad je verzija bila aktivna.



Slika 3.1: Prikaz relacijskog modela (ERA)

Model baze podatak ne izgleda kompleksno, što i nije, ali nam omogućuje sve funkcionalnosti temporalnih i popoćenih baza podataka. Lako možemo pratiti verzije naših projekata, te ih po potrebi

preuzeti.

Naziv entiteta	Atributi	Opis
korisnik	idkorisnik ime prezime kor_ime lozinka email status	Tablica u kojoj se nalaze registrirani korisnici, koji mogu koristiti aplikaciju.
posjeduje	projekt korisnik datum	Tablica koja se puni kada korisnik doda svoj novi projekt, ili ako korisnik želi dodat nekog korisnika na svoj projekt.
projekt	idprojekt naziv opis detalji datoteka	Tablica u kojoj se nalaze svi projekti, od svih korisnika.
verzije	opis naziv datoteka do_kad projekt	Tablica koja se puni kada se ažurira postojeći projekt, puni se pomoću trigera.

Slika 3.2: Opis ERA modela

Na slici su prikazani entiteti, vidimo opis entiteta iz era modela, te popisane atribute. Da detaljnije znamo što čemu služi koji entitet.

Poglavlje 4

Implementacija

U ovome poglavlju opisati ćemo kako izgleda implementacija aplikacije za verzioniranje, te će biti prikazani neki djelovi koda.

```
CREATE TABLE "posjeduje" (  
    "projekt" int4 ,  
    "korisnik" int4 NOT NULL ,  
    "datum" timestamp  
);  
CREATE TABLE "verzije" (  
    "opis" text ,  
    "naziv" varchar(20),  
    "datoteka" oid ,  
    "do_kad" timestamp ,  
    "projekt" int4  
);  
CREATE TABLE "projekt" (  
    "idprojekt" SERIAL NOT NULL ,  
    "naziv" varchar(20),  
    "opis" text ,  
    "detalji" tip ,  
    "datoteka" oid ,  
    PRIMARY KEY("idprojekt")  
);  
CREATE TABLE "korisnik" (  
    "idkorisnik" SERIAL NOT NULL ,  
    "ime" varchar(30) NOT NULL ,  
    "prezime" varchar(30),  
    "kor_ime" varchar(30),  
    "lozinka" varchar(15),  
    "email" varchar ,  
    "status" int4 DEFAULT 1 ,  
    PRIMARY KEY("idkorisnik")
```

```
);  
ALTER TABLE "posjeduje"  
  ADD CONSTRAINT "Ref_projekt_has_korisnik_to_projekt"  
  FOREIGN KEY ("projekt")  
    REFERENCES "projekt"("idprojekt")  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
    NOT DEFERRABLE,  
  ADD CONSTRAINT "Ref_projekt_has_korisnik_to_korisnik"  
  FOREIGN KEY ("korisnik")  
    REFERENCES "korisnik"("idkorisnik")  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
    NOT DEFERRABLE;  
ALTER TABLE "verzije"  
  ADD CONSTRAINT "Ref_verzije_to_projekt" FOREIGN KEY ("projekt")  
    REFERENCES "projekt"("idprojekt")  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
    NOT DEFERRABLE;
```

Iz ERA dijagrama koji je prikazn u prethodnom poglavlju izgeneriran je sljedeći sql kod. Ovdje je prikazan sql kod za bazu podataka aplikacije za verzioniranje.

Sljedeći kod nam prikazuje trigger, koji je napravljen nad tablicom projekt.

```
create or replace function stara_verzija()  
returns trigger  
as $$  
begin  
if (TG_OP='UPDATE') then  
insert into verzije select OLD.opis,OLD.naziv,OLD.datoteka,NOW(),  
OLD.idprojekt;  
end if;  
return null;  
end;  
$$ language plpgsql;  
  
create trigger projekt_promjena  
after update on projekt  
for each row execute procedure stara_verzija();
```

Ovdje je prikazn trigger koji se koristi kada se tablica projekt ažurira novom verzijom, tada se prijašnja verzija iz tablice projekt stavlja u tablicu verzije pomoću prethodnog triggera.

Ovdje je prikazan dio php koda koji nam služi za ispis naših projekata. Napravljena je funkcija koja se kasnije poziva, a unutar funkcije je sql upit koji spaja tri tablice korisnik, projekt i posjeduje kako

bi došli do korskika i uspoređuje ga s trenutno prijavljenim korisnikom kako bi ispisali samo njegove projekte. Zatim je napravljen ispis u tabličnom prikazu.

```
function mojiprojekti() {
$baza = new baza();
$ispis="";
$korisnik = isset($_SESSION["PzaWeb"]) ? $_SESSION["PzaWeb"]->
get_id() : "";
$upit = "SELECT distinct idprojekt , naziv , opis , ( detalji ). svrha ,
( detalji ). kreatori , ( detalji ). cilj , datoteka , datum FROM
projekt , posjeduje , korisnik where idprojekt=projekt and korisnik=
'{$korisnik}' order by datum desc";
$rezultat = $baza->selectDB($upit);
$ispis .= "<caption>Moji projekti </caption>";
$ispis .= "<thead><tr><th>Naziv</th><th>Opis</th><th>Svrha</th>
<th>Kreatori</th><th>Cilj</th><th>Datoteka</th><th>Datum</th>
<th>Verzija</th><th>Obrisi</th><th>Preuzmi</th><th>Osobe</th></thead>";
while (list($idprojekt , $naziv , $opis , $svrha , $kreatori , $cilj ,
$datoteka , $datum)= pg_fetch_array($rezultat)) {
$ispis.="<tr>";
$ispis.="<td>$naziv</td>";
$ispis.="<td>$opis</td>";
$ispis.="<td>$svrha</td>";
$ispis.="<td>$kreatori</td>";
$ispis.="<td>$cilj</td>";
$ispis.="<td>$datoteka</td>";
$ispis.="<td>$datum</td>";
$ispis.="<td><a href='editiraj.php?akcija=26&idprojekt=$idprojekt'>
Nova</a></td>";
$ispis.="<td><a href='editiraj.php?akcija=27&idprojekt=$idprojekt'>
Obrisi</a></td>";
$ispis.="<td><a href='editiraj.php?akcija=28&idprojekt=$idprojekt'>
Preuzmi</a></td>";
$ispis.="<td><a href='editiraj.php?akcija=29&idprojekt=$idprojekt'>
Osobe</a></td>";
$ispis.="</tr>";}
$ispis = $ispis . "<tbody>";
$ispis.="</tbody>";
echo $ispis;
}
```

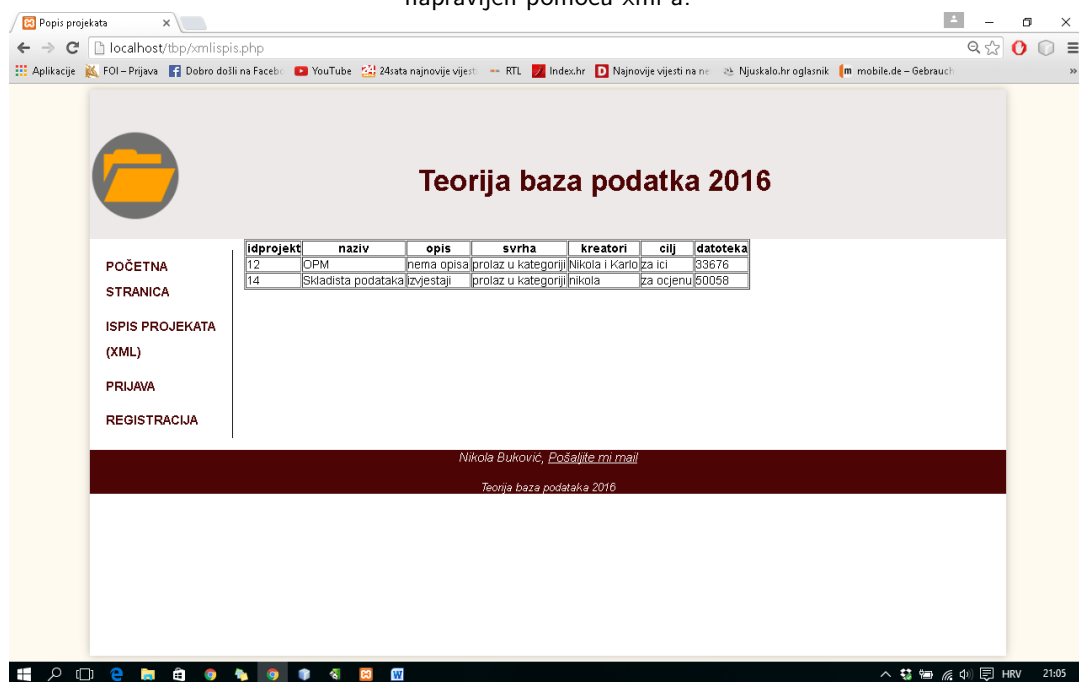

Temporalno upravljanje dokumentima (verzioniranje, XML dokumenti, napredne mogućnosti pretraživanja) - Temporalne + Poopćene baze podataka - PostgreSQL + web grafičko sučelje po želji – Seminarski rad

Poglavlje 5

Primjer korištenja

U ovome poglavlju pokazati ću izgled aplikacije za verzoiniranje. Na prvoj slici prikazan je popis svih projekata koji se nalaze na sustavu.

Ako nismo registrirani na aplikaciji tada možemo viditi koji su projekti na sustavu, ovaj pregled je napravljen pomoću xml-a.



Slika 5.1: Prikaz projekata

Temporalno upravljanje dokumentima (verzioniranje, XML dokumenti, napredne mogućnosti pretraživanja) - Temporalne + Poopćene baze podataka - PostgreSQL + web grafičko sučelje po želji – Seminarski rad

Ako osoba koja želi koristiti aplikaciju nije registrirana, tada se prvo treba registrirati.

The screenshot shows a web browser window with the URL `localhost/tbp/registracija.php`. The page title is "Teorija baza podataka 2016". On the left, there is a sidebar with four menu items: "POČETNA", "STRANICA", "PRIJAVA", and "REGISTRACIJA", with "REGISTRACIJA" being the active one. The main content area contains a registration form with the following fields: "Ime:" (text input), "Prezime:" (text input), "Korisničko ime:" (text input), "Lozinka:" (text input), and "Email:" (text input). Below these fields are two buttons: "Registriraj se" and "Resetiraj". At the bottom of the page, there is a footer with the text "Nikola Buković, Pošaljite mi mail".

Slika 5.2: Obrazac za registraciju

Kada je osoba registrirana tada se može prijaviti na sustav. te započeti s korištenjem aplikacije.

The screenshot shows a web browser window with the URL `localhost/tbp/prijava.php`. The page title is "Teorija baza podataka 2016". On the left, there is a sidebar with four menu items: "POČETNA", "STRANICA", "PRIJAVA", and "REGISTRACIJA", with "PRIJAVA" being the active one. The main content area contains a login form with the following fields: "Korisničko ime:" (text input), "Lozinka:" (text input), and "Zapamti me:" (checkbox). Below these fields is a button labeled "Prijava". At the bottom of the page, there is a footer with the text "Nikola Buković, Pošaljite mi mail" and "Teorija baza podataka 2016".

Slika 5.3: Prijava na aplikaciju

Temporalno upravljanje dokumentima (verzioniranje, XML dokumenti, napredne mogućnosti pretraživanja) - Temporalne + Poopćene baze podataka - PostgreSQL + web grafičko sučelje po želji – Seminarski rad

Kada smo se prijavili na aplikaciju, tada možemo kreirati novi projekt, kojega ćemo pratiti pomoću aplikacije.

Dodaj projekt

Naziv :

Opis :

Kreator :

Svrha :

Cilj :

Datoteka :

Tip :

[Nastavi](#)

Slika 5.4: Dodavanje novog projekta

Kako bi vidjeli na kojim projektima radimo za to nam služi stranica moji projekti, što možemo vidit na sljedećoj slici. Kada želimo dodat novu verziju na projektu kliknemo na Nova te nam se otvara novi prozor u kojem unosimo izmjene te datoteku, ako želimo možemo izbrisat projekt, te imamo mogućnost vidit ko sve radi na projektu to možemo uz opciju Osobe. Također ako želimo možemo preuzeti projekt.

Sustav za verzioniranje

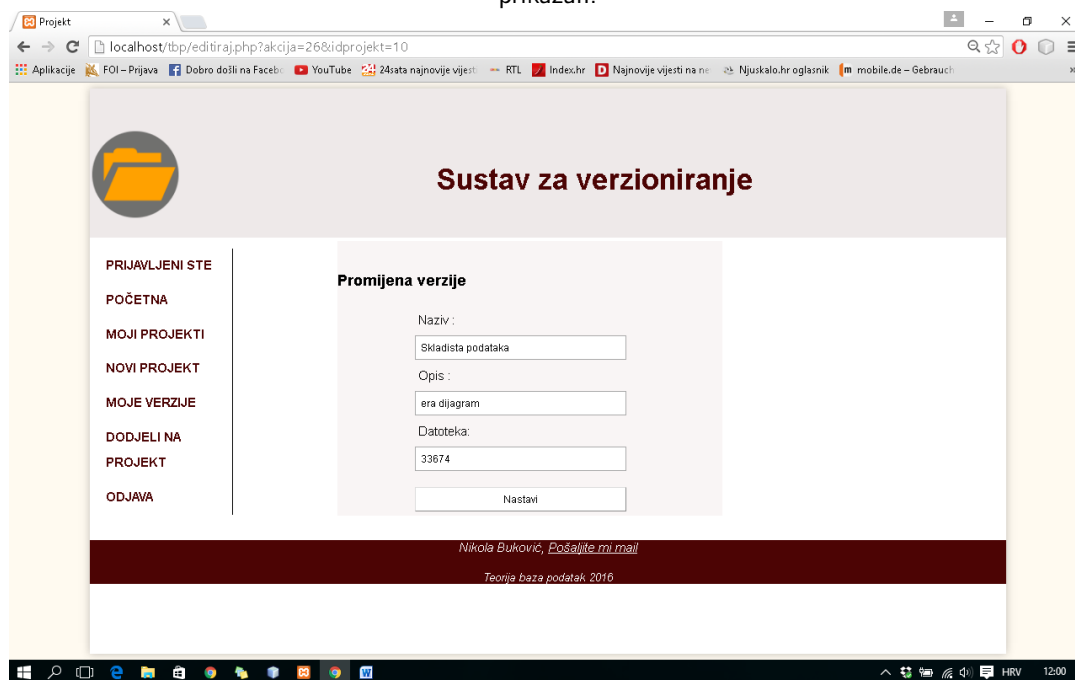
Moji projekti

Naziv	Opis	Svrha	Kreator	Cilj	Datoteka	Datum	Verzija	Obriši	Preuzmi	Osobe
Skladista podataka	era dijagram	nikola	nema je	ocijena 5	33674	2016-06-20 17:36:07.508	Nova	Obriši	Preuzmi	Osobe
Dodaj novi										

Slika 5.5: Prikaz mojih projekata

Temporalno upravljanje dokumentima (verzioniranje, XML dokumenti, napredne mogućnosti pretraživanja) - Temporalne + Poopćene baze podataka - PostgreSQL + web grafičko sučelje po želji – Seminarski rad

Promjena verzije, gdje možemo dodati novu izmjenjenu datoteku. Kada smo dodali novu verziju tada stara verzija iz tablice projekti seli u tablicu verzije, što je ostvareno pomoću trigera koji je prethodno prikazan.



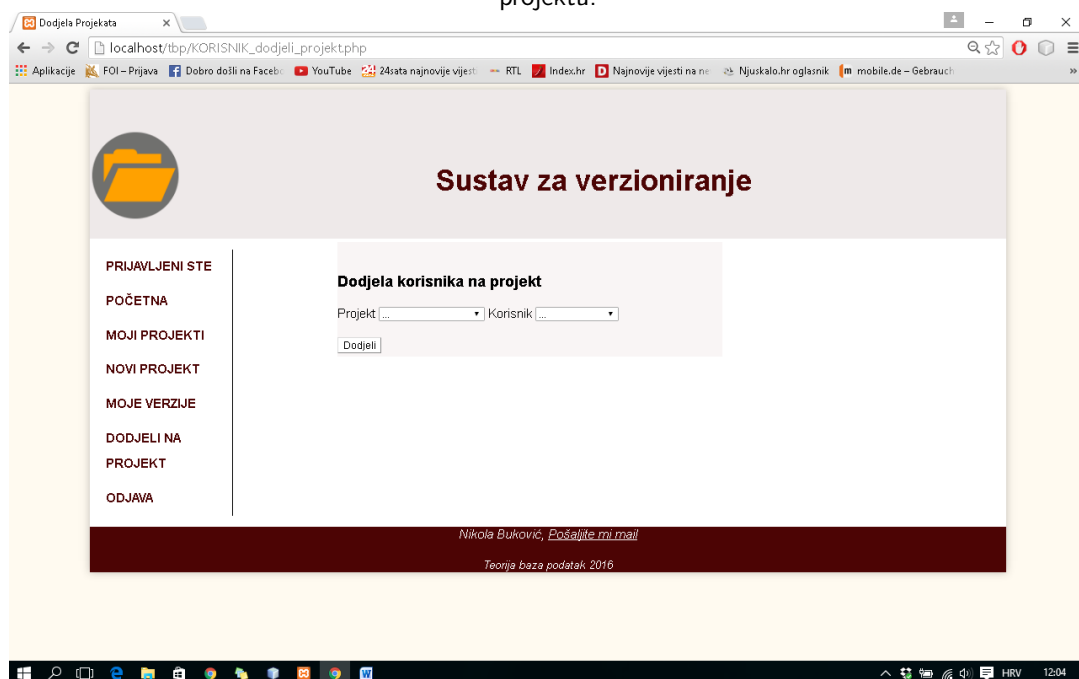
Slika 5.6: Nova verzija

Prikaz svih verzija projekta.



Slika 5.7: Verzije projekta

Kada smo prijavljeni na aplikaciju, te imamo kreiran projekt, na poveznici dodjeli projekt, otvara nam se stranici na kojoj se nalaze dvije padajuće liste, na prvoj su naši projekti, a na drugoj svi korisnici aplikacije odabirom projekta i korisnika, dodajemo korisnika na taj projekt, te i on može raditi na tome projektu.



Slika 5.8: Dodjeli osobu na projekt

Poglavlje 6

Zaključak

U sklopu kolegija Teorija baza podataka izrađena je aplikacija za verzioniranje projekata. Aplikacija je rađena u više alata, te je napravljeno web sučelje. Pomoću ove aplikacije možemo pratiti svoje projekte te s ostalim korisnicima raditi na njima i stavljati nove verzije projekata. Kada je netko od korisnika stavio novu verziju, tada si drugi korisnik može skinuti tu verziju te nastaviti nadograđivati projekt.

Izrađeni sustav za verzioniranje nam može pomoći kod verzioniranja naših projekata, te kod timskog rada, cijeli tim može raditi zajedno na projektu, a da fizički nisu skupa.

Poglavlje 7

Literatura

1. Temporalne baze podataka, Markus Schatten, dostupno 27.Lipnja 2016 na
<http://autopoiesis.foi.hr/wiki.php?name=Baze+Podataka+-+FOIparent=NULLpage=tempbp>
2. Poopćene baze podatka, Markus Schatten dostupno 27.Lipnja 2016 na
<http://autopoiesis.foi.hr/wiki.php?name=Baze+Podataka+-+FOIparent=NULLpage=poopbp>
3. Temporalne baze podataka, Andrea Perčinlić dostupno 27.Lipnja 2016 na
<http://digre.pmf.unizg.hr/4304/1/diplomski.pdf>