Matematički fakultet Beograd

Profesor:

Saša Malkov

Asistent:

Anđelka Zečević

Studenti:

Danijel Marjanović 1121/2010

Filip Luković 1048/2013

Gerawork Anyekulu 1004/2012

Uroš Milenković 164/2010

Stefan Isidorović 1014/2013

Manthano project

Projekat iz predmeta Programiranje za Veb

# **Sadržaj**

[**Sadržaj** 1](#_Toc378812266)

[**Uvod** 2](#_Toc378812267)

[**Opis arhitekture sistema** 3](#_Toc378812268)

[**Realizacija modela podataka** 4](#_Toc378812269)

[**Upustvo za instalaciju** 5](#_Toc378812270)

[**Opis izvornog koda** 5](#_Toc378812271)

[**Organizacija baze podataka** 6](#_Toc378812272)

# **Uvod**

Informacioni sistem Mantano je predviđen da pomogne organizaciju nastavnih i vannastavnih aktivnosti u okviru neke obrazovne ustanove, u našem slučaju Matematičkog fakulteta u Beogradu.

Osnovna ideja je bila da se podstakne samostalna organizacija raznih radionica od strane samih studenata, uz moguću pomoć nastavnog osoblja fakulteta. Kada je počela realizacija projekta, ideja je proširena tako da se informacioni sistem može koristiti i za realizacju obaveznih kurseva u okviru nekog nastavanog programa. Sistem sada čine dve velike celine, **realizovane aktivnosti** i **predlozi aktivnosti** u kojima postoji opcija glasanja za predavače, kao i slanje predloga drugim korisnicima. Takođe ideja je da se stvori fleksibilan sistem tako da trenutni ili završeni kursevi mogu da se iskoriste u realizaciji novih kurseva.

Prilikom izrade sistema predvideli smo da se isti može koristi prilikom organizovanja sledećih aktivnosti:

* manjih radionica sa ograničenim brojem predavanja
* regularnih nastavnih kurseva,
* sekcija studenata koje nemaju ograničen broj podaktivnosti
* naučnih skupova i pratećih događaja
* sportskih aktivnosti

Naše mišljenje je da u okviru našeg fakulteta ne postoje dovoljno dobro organizovane vannastavne aktivnosti, a studenti ih ne mogu sami organizovati, jer nemaju tehničke uslove i ostale potrebne resurse. Ovaj sistem bi služio prvenstveno kao veza između studenata, bivših studenata, nastavnog i tehničkog osoblja fakulteta. Kroz predloge bi se grupisali studenti istih interesovanja, a to bi bilo dostupno na uvid nastavnom kadru i malo iskusnijim studentima koji se bave istom oblašću, koji bi bili voljni da podele svoje znanje sa drugima. Pored toga bi tehničko osoblje fakulteta moglo da unapred, koristeći naš sistem, organizuje prostorije i resurse potrebne za uspešnu realizaciju aktivnosti.

Uzimajući sve gore navedeno u obzir, naše viđenje svega ovoga je da bi Manthano IS pomogao i nastavnom osoblju i studentima tako što bi podigao kvalitet studiranja na viši nivo i olakšao realizaciju navedenih aktivnosti.

# **Opis arhitekture sistema**

Manthano IS se u osnovi zasniva na klijent-server arhitekturi. Serverski deo čine baza podataka, PHP klase kao omotači baze i REST servisi koji su jedina veza korisnika se serverom, dok klijentski deo je realizovan uz pomoć HTML, AngularJS, Bootstrap, jQuery, JavaScript i AJAX tehnologija.

Slika : Klijent server arhitektura korišćena u Manthano IS

Podaci

Zahtev

Podaci

Zahtev

Baza

REST SERVISI

PHP klase kao   
 omotač baze

Klijentski deo

Kao glavnu serversku komponentu smo odabrali Apache web server zbog njegove stabilnosti i velike rasprostranjenosti, a za bazu smo koristili MySQL server zbog veoma dobre i funkcionalne integracije MySQL-a sa PHP-om. Pristup bazi podataka je vršen preko PDO-a i activ-records alata.

Za PHP framework je odabran CodeIgniter 2.1.4. U pitanju je stariji („sticky”) MVC framework, ali glavni razlozi za izbor istog su:

* relativno jednostavni alati i struktura
* velika sloboda u pristupu rešavanju problema
* fast-learning-curve

Interakcija između klijenta i servera se obavlja isključivo preko AJAX poziva koji kontaktiraju REST servise, što predstavlja značajno sigurnosno poboljšanje u odnosu na klasičnu klijent server arhitekturu, a takođe i povećava skalabilnost.

Prikaz i obrada dobijenog sadržaja se vrši pomoću **AngularJS** biblioteke, a za poboljšan vizuelni izgled i nezavisnost od uređaja na kome se stranice prikazuju koristili smo **Bootstrap**.

Takođe je započeto korišćenje Node.JS (serverski javascript) za realizaciju Veba u realnom vremenu. Node.JS je pogodniji za ovakve stvari od Apache servera zato što je zasnovan na događajima i lako može da se izbori sa mnoštvom zahteva odjednom. Pored same Node.JS tehnologije uz njega je korišćen i Socket.IO biblioteka za rad sa socketima, kao i Alertify (klijentska javascript biblioteka) za pop-up obaveštenja.

# **Realizacija modela podataka**

Manthano je realizovan tako da sadrži pet glavnih entiteta: Predlog, Aktivnost, Događaj, Materijal i Korisnik, a logički je podeljen u dva dela, deo za **postavljanje Predloga** i glasanja za isti, i deo **realizaciju Aktivnosti**.

Svaki registrovani korisnik može predložiti aktivnost koju misli da bi trebalo realizovati, predložiti koga želi za vlasnika te Aktivnosti, prijaviti druge korisnike za tu aktivnosti ili glasati za već predložene Vlasnike i prijaviti se na već postojeće predloge aktivnosti.

Posle ispunjenja određenih uslova (ispunjen dovoljan broj zainteresovanih studenata, saglasnost ljudi koji bi održavali tu aktivnost) predlog postaje aktivnost. Aktivnosti su organizovane u n-arno stablo tako da jedna Aktivnost može sadržati druge Aktivnosti (dok god to ispunjava uslove strukture stabla), kao i listu Događaja vezanih za nju.

Realizovana je višestruka hijerarhija, pri čemu se podrazumeva da Događaj može pripadati većem broju Aktivnosti u isto vreme, a takođe i Događaj može imati listu materijala vezanih za njega. Materijal takođe može pripadati većem broju Događaja u isto vreme.

Korisnici se dele na **obične korisnike** i **administratore**, dok se obični korisnici mogu razlikovati po pravima na samo određenim Predlozima, Aktivnostima i Događajima. Korisnik koji predloži Aktivnost, korisnik koji je Vlasnik neke Aktivnosti ili korisnik koji je Vlasnik nekog događaja ima veća prava za konkretnu aktivnost nego obični korisnik, ali su njegova prava na ostalim aktivnostima ista kao kod običnog korisnika. Administrator ima sva maksimalna prava na svim aktivnostima, kao da je vlasnik svih Aktivnosti.

# **Upustvo za instalaciju**

Potrebno je imati instaliran Apache web server i MySQL server na Linux ili Windows platformi, uz dodatna podešavanja:

* Prekopirati fajlove u Root Apache web servera.
* U php.ini uključiti short\_open\_tags (short\_open\_tags=On) kao i otkomentarisati ssl ekstenziju.
* Pomoću skripte za kreiranje baze napraviti bazu pa potom je napuniti podacima (postoji skripta sa predefinisanim podacima u istom folderu /db)
* Startovati Node.JS server.
  + Pokrenuti Node.JS Command Prompt
  + Otići u /nodejs folder
  + Pokrenuti server komandom „node server.js“

Nakon ovoga, instalacija se vrsi tako sto se importuje baza podataka i prekopira root folder web aplikacije na željeno mesto na serveru.

# **Opis izvornog koda**

U application/helpers se nalaze klase koje rade sa bazom podataka. Te klase su najniži sloj naše aplikacije I predstavljaju skup metoda kojima se upravljaju podaci.

To su klase koje odgovaraju entitetima našeg sistema

1. Activity
2. Event
3. Material
4. User
5. Proposal

Kao što je već rečeno svaka od klasa sadrži statički interfejs za upravljanje podataka koji su bili potrebni u nekom trenutku u okviru implementacije.

U application/controllers se nalaze kontroleri CodeIgniter frameworka kao I naši REST servisi to su:

1. REST\_activity\_b
2. REST\_event\_b
3. REST\_proposal
4. REST\_user
5. Materials

Svaki od ovih servisa nasleđuje MY\_Controller klasu, koja je u okviru CodeIgniter framework I omogućava lako rutiranje do samih servisa.

Servisi sadrže metode koje određuju sa kojim podacima se radi. Poziv servisa ide /ime\_servisa/ime\_metode/argumenti . Kao i kod klasa svaki servis sadrži implementirane metode koje koristeći helper klase skupljaju i pakuju podatke I prosleđuju klijentu odnosno vrše promene na serveru.

Za generisanje strana na klijentskoj strani koristili smo AngularJS. Sama organizacija skripti je prosta. Postoji glavni AngularJS module manthanoApp (/assets/scripts/routeModule.js) koji se brine o templejtima. Uz to u njega su implementirani servisi za konekciju na Node.JS (servis socket) kao I servis za slanje mejlova (servis mail). U okviru ovog modula postoji još i kontroler koji se brine za obaveštenja.

manthanoApp modul implementira $route servis koji u zavisnosti koja je adresa uneta (koriste se # adrese ) učitava određen šablon. Šabloni su smešteni u /assets/html\_fragments/. Za svaki šablon je jasno navedeno koji kontroler je zadužen za njega. Dalje su moduli podeljeni prema entitetima I svaki modul implementira potrebne kontrolere za upravljanje I kreiranje interfejsa potrebnog za taj entitet.

Application:

- konfiguracije

- modeli

- kontroleri

- view-ovi

- helper bibilioteke

- proširenja frejmworka

- itd...

System - CI core

Html\_fragments - html fragmenti strana

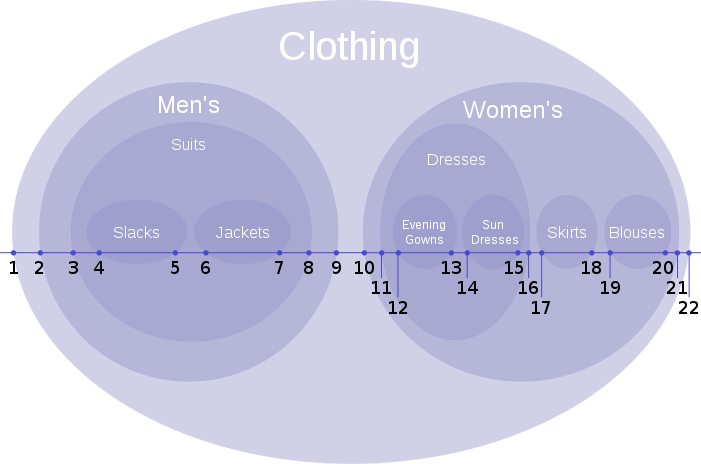
Assets:

- statičnii fajl resursi: js skripte, CSS, statične slike...

- dinamički resursi: uploadovane slike

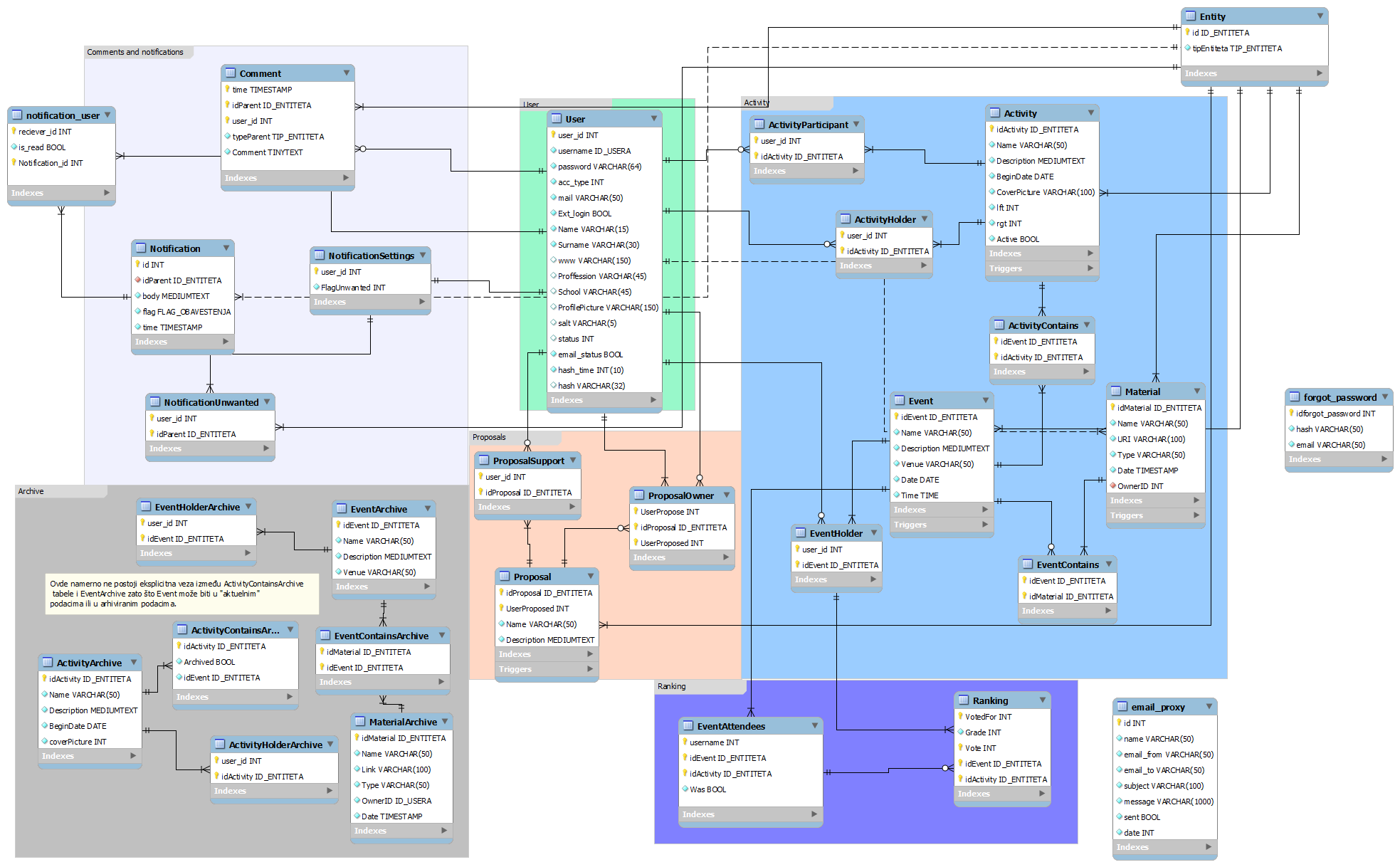
# **Organizacija baze podataka**

Za realizaciju baze podataka je korišćen MySQL. Baza je dizajnirana da što više odgovara zahtevima sistema. Podeljena je u više celina. Deo za komentare I obaveštenja, arhiva, deo sa realizovanim aktivnostima, deo za predloge, deo za glasanje. Activity tabela je realizovana pomoću Nested Set modela da bi se verno sačuvalo n-arno drvo u bazi.



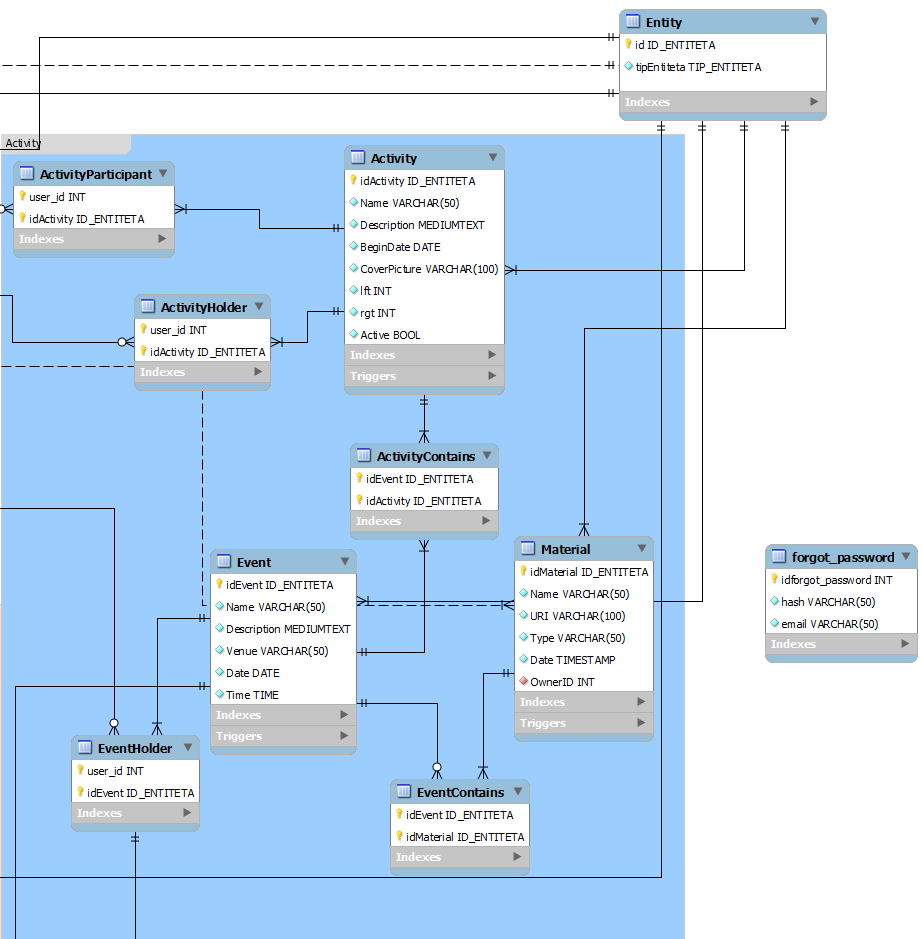
Slika - Ilustracija Nested Set Modela

Ostatak baze je urađen relacionim modelom.



Slika 1 - EER Dijagram baze podataka

Baza je projektovana tako da što vernije preslika funkcionisanje celog sistema. Podeljena je na neke logičke celine. Logičke celine su vezane za predlog i čuvanje podataka po predlozima, na realizovane Aktivnosti i sve podatke o njima, deo za arhiviranje podataka iz aktivnosti, deo za dodatne module koji bi olakšali korišćenje sistema kao što su sistem obaveštenja i komentara, kao i deo za rangiranje predavača i vođenje evidencije o dolascima na događaje.



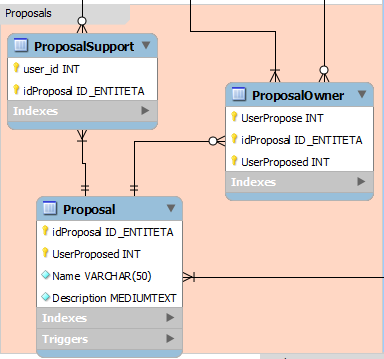
Slika 2- Realizovana aktivnost

Podatke o realizovanoj aktivnosti čuvaju ovih 8 tabela.

Activity tabela čuva podatke o specifične I jedinstvene podatke o samoj aktivnosti, kao i informacije o celom n-arnom stablu u koje su sve aktivnosti organizovane. Model korišćen za čuvanje structure kao što je n-arno stablo je Nested Set model, ili Ugnježdeni skupovi. Pogodan je za relacione baze podataka zato što su same baze zasnovane tako da rade na skupovima podataka. Svaku poziciju u stablu jedinstveno određuju indexi lft i rgt.

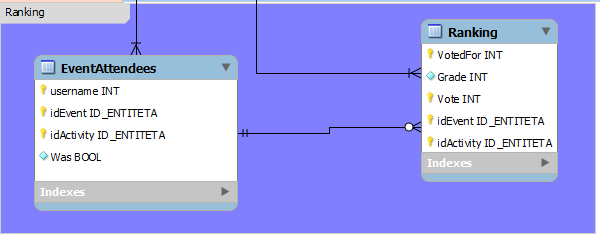
Pored toga čuvaju se i spiskovi ljudi odgovornih za tu aktivnost (ActivityHolder), spisak ljudi koji pohađaju svaku aktivnost (ActivityParticipant), kao i spisak Događaja u okviru Aktivnosti (ActivityContains).

Potreba za čuvanje spiska Događaja za svaku Aktivnost dolazi iz želje da se kreira što fleksibilniji system, pa ova tabela prati višestruku hijerarhiju koja odlikuje Manthano (Događaj može biti deo više Aktivnosti u isto vreme). Događaj ima listu materijala vezanih za njega (EventContains) kao i spisak ljudi odgovornih za taj Događaja. Postoji i tabela u kojoj se čuvaju spiskovi ljudi koji su bili na tom događaju (EventAttendees).



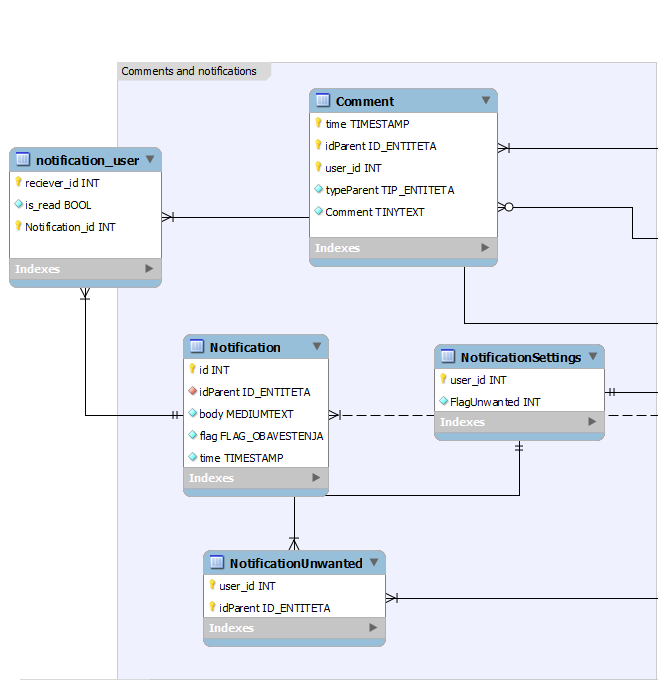
Slika 3- Proposal

Tabele koje čuvaju podatke o Predlogu su Proposal, ProposalSupport i ProposalOwner. Proposal čuva podatke koji je korisnik predložio, ime I opis, dok ostale dve tabele čuvaju podatke o ljudima koji podržavaju I ljudima koji su predloženi za vlasnika predloga, kao I broj glasova za svakoh “vlasnika”.



Slika 4- Rankiranje

EventAttendees i Ranking tabele čuvaju podatke potrebne za realizaciju sistema glasanja na nivou događaja. EventAttendees čuva podatke o korisnicima koji su bili na određenom Događaju u okviru određene Aktivnosti, dok table Ranking čuva glasove posetioca Događaja. Ovakva implementacija je da bi se onemogućilo glasanje korisnicima koji nisu bili na određenom Događaju da glasaju za ljude koji su taj Događaj održali.



Slika 5- Notifiakcije

Ove tabele čuvaju informacije za module komentara I obaveštenja. Sistem obaveštenja je zamišljen da određen Entitet (koga nasleđuju Aktivnost, Predlog, Događaj, Materijal) emituje događaj za koji će se slati obaveštenje. Pa se to obaveštenje dalje prosleđuje određenim korisnicima koji će ga pregledati u nekom trenutku. Takođe obaveštenja od određenog Entiteta mogu da se ignorišu. Obeveštenja će takođe biti podeljena u grupe, od kojih će korisnik moći da bira koju grupu obaveštenja želi da prima.

# Kratko upustvo za instalaciju.

1. Sav sadržaj se prekopira na web server Apache (sem foldera nodejs I db)
2. U folderu DB se nalazi skripta za kreiranje baze, kao i određeni podaci za punjenje baze.
3. Pokrenuti NodeJS server.js (iz NodeJS Command Prompta naredbom node server.js)
4. Početi korišćenje Manthano Sistema.