

**SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN**

**Filip Horvat**

## **Razvoj aplikacije za elektroničko glasovanje**

**ZAVRŠNI RAD**

**Varaždin, rujan 2012.**

**SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN**

**Filip Horvat**

**Redoviti student**

**Broj indeksa: 36998/08 - R.**

**Smjer: Informacijski sustavi**

**Preddiplomski studij**

## **Razvoj aplikacije za elektroničko glasovanje**

**ZAVRŠNI RAD**

**Mentor:**

Tihomir Orehovački, mag. inf.

**Varaždin, rujan 2012.**

# Sadržaj

1. Uvod .....	1
2. Sustavi za elektroničko glasovanje .....	3
2.1. Podjela sustava za elektroničko glasovanje .....	3
2.2. Prednosti i nedostaci elektroničkog glasovanja .....	5
3. Postojeće web aplikacije za elektroničko glasovanje .....	7
3.1. eBallot .....	7
3.2. Big Pulse .....	9
3.3. Simply voting .....	11
4. Tehnologije korištene kod izrade aplikacije sa primjerima u kôdu aplikacije .....	13
4.1. HTML .....	13
4.1.1. HTML Elementi .....	14
4.1.2. Forme .....	15
4.2. PHP .....	17
4.2.1. Varijable .....	18
4.2.2. Funkcije .....	19
4.3. Smarty .....	22
4.4. JavaScript i JQuery .....	24
4.5. CSS .....	28
5. Realizacija aplikacije .....	30
5.1. Opis funkcionalnosti aplikacije .....	30
5.2. Use case .....	32
5.3. Navigacijski dijagram .....	33
6. Tehnički dio aplikacije i kôd .....	34
6.1. ERA Dijagram .....	34

6.2. Struktura web mjesta i izvršavanje aplikacije .....	36
6.3. Uređivanje glasovanja .....	37
6.4. Pregled rezultata .....	41
6.5. Glasovanje .....	42
7. Korištenje aplikacije.....	45
7.1. Početna stranica .....	45
7.2. Pregled glasovanja .....	46
7.3. Glasovanje .....	48
7.4. Rezultati.....	49
7.5. Uređivanje glasovanja .....	51
8. Zaključak.....	54
9. Literatura .....	55

# 1. Uvod

Pod pojmom elektroničko glasovanje podrazumijevamo različite vrste glasovanja. U članku „Online voting“ objavljenom 2001. godine od Britanskog parlamentarnog ureda za znanost i tehnologije elektroničko glasovanje se odnosi na korištenje računala ili elektronske opreme za bilježenje glasova kod glasovanja, a ponekad se pojam koristi specifično i odnosi se samo na glasovanja koje se odvijaju na Internetu [2]. Takvim glasovanjima će se ovaj rad najviše baviti, iako će u početku rada biti riječi i o ostalim vrstama elektroničkog glasovanja.

Glasovanje je temeljni instrument demokracije koje ljudima omogućava službeni način da izraze svoje mišljenje vladi. Tradicionalno, proces glasovanje je dosta zamoran jer korisnik mora doći osobno na neko mjesto da glasa. Ova činjenica rezultira time da je postotak izlaznosti na glasovanje niska [3]. Glasovanje preko emaila može pružiti uslugu glasovanja korisnicima koji žive previše udaljeno od glasačkog mjesta, ali ovaj način uzima previše vremena i zahtijeva od dodan trošak rada za kod primanja i zbrajanja glasačkih listića. „Sustavi za elektroničko glasovanje EVS (Electronic voting system) mogu prevladati te probleme i učiniti moderan društven život više ugodnim, efikasnim i manje skupim. Koristeći EVS u državnim izborima, korisnik može glasati na poslu ili kod kuće“ [3].

Elektroničko glasovanje, uključujući ostale usluge na razini države, se sve više koristi na elektronički način. Spomenimo na primjer e-upravu i njezine podskupine kao što su e-gradjevinska dozvola, zatim e-zdravstvo te njezina podskupina e-redovi čekanja, zatim e-bankarstvo itd. U svijetu se može pronaći mnogo primjera gdje su glasovanja za državna tijela bila provedena putem elektroničkog glasovanja kao i mnogi drugi primjeri službenih glasovanja na razini države koja su izvršena na elektronički način kao što je npr. izglasavanje zakona u hrvatskome saboru. IDEA u svome članku „Introducing Electronic Voting: Essential Considerations“ iz 2011. godine nabroja više država u kojima je glasovanje održano na elektronički način. „Za primjer se može navesti Estonija gdje je 2005. godine uvedeno elektroničko glasovanje kao jedan od načina uz tradicionalno glasovanje, zatim Venezuelu gdje je 2004. godine glasovanje provedeno putem DRE (Direct Recording Electronic) uređaja te Nizozemska gdje je do 2008. glasovanje bilo na elektronički način, ali je tada ukinuto nakon prosvjeda aktivista“ [4].

Osim standardnih glasovanja na državni razini kao što su glasovanje za saborske zastupnike, glasovanje za predsjednika države, glasovanje za lokalna tijela, referendum i tako dalje, elektronička glasovanja nalaze primjenu i u ostalim manje formalnim glasovanjima poput glasovanje uprave ili dioničara neke kompanije za različite odluke ili kao što su glasovanje na različitim natjecanjima. Primjer za to bi bilo glasovanje za najboljeg glazbenog izvođača, najboljeg sportaša itd. Primjeri koji se još ovdje mogu spomenuti su različita glasovanja za odbore ili zborove poput studentskog zbora, odbora neke političke stranke te mnoga druga.

U ovom završnom radu većina sadržaja je koncentrirana na samu aplikaciju za elektroničko glasovanje, njezinu izradu i način na koji funkcionira te tehnologije kojima je aplikacija ostvarena. Na početku rada opisane su vrste elektroničkog glasovanja, njihove prednosti i nedostaci, zajedničke karakteristike itd. Nakon toga je odabrano i analizirano nekoliko web aplikacija za elektroničko glasovanje. Poslije ovog dijela fokus rada se usmjerava na samu aplikaciju. Najprije su opisane tehnologije koje se koriste u aplikaciji i njihovo korištenje u aplikaciji. Nakon što je obrađen dio s tehnologijama slijedi dio u kojem su opisane funkcionalnosti i mogućnosti koje će aplikacija imati. Sljedeći dio rada je opis rada aplikacije kroz sam kôd i kroz dijagrame. Za kraj još preostaje opis korištenje aplikacije kroz snimke ekrana što će biti jedan veći dio rada te zaključak i popis korištene literature.

## 2. Sustavi za elektroničko glasovanje

U ovom dijelu bit će opisani sustavi za elektroničko glasovanje. Iako je ovaj rad usmjeren na web aplikacijame namijenjene elektroničko glasovanju, bit će spomenuto na koje sve načine može elektroničko glasovanje biti izvedeno. U sljedećem dijelu rada su spomenuti neki autori te je navedeno koje načine elektroničkog glasovanja oni spominju te kako su podijelili elektronička glasovanja.

### 2.1. Podjela sustava za elektroničko glasovanje

Susanne Caarls je savjetnica u Vijeću Europe te započela i razvila njihove standarde e-glasovanja te sudjelovala u timu OSCE/ODIHR koji je procjenjivao izbore u elektroničke izbore Estoniji i ima mnogo iskustva u e-demokraciji i e-glasovanju [9]. Ona u članku „Handbook on E-Voting“ objavljen od Vijeća Europe u 2010. godine spominje nekoliko načina za elektroničko glasovanje [1]:

- Pomoću DRE (direct-recording electronic) uređajima koji su instalirani na glasačkom mjestu gdje bilježe i spremaju glasove. Taj postupak se može odvijati preko zaslona osjetljivog na dodir (sa ili bez posebne olovke) ili preko jednog ili više gumba koje je potrebno pritisnuti.
- Glasovanje putem Interneta koje se može odvijati u kontroliranim uvjetima (na biračkom mjestu) ili u nekontroliranim uvjetima, kod kuće, odnosno na bilo kojem mjestu gdje postoji pristup Internetu.
- Optički i digitalni čitači koji se koriste na biračkom mjestu i direktno od birača učitavaju glasove ili pak oni koji se koriste za brojanje glasova nakon što glasovanje završi. Ovim drugim načinom pokušava se smanjiti potencijalna ljudska pogreška kod ručnog brojanja, ali kvaliteta brojanja ovisi o pravilnom označavanju odgovora i kvalitete tinte koju je glasač koristio prilikom označavanja odgovora.
- Na samom glasačkom mjestu se koristi neki medij za bilježenje glasa, koji je zatim registriran u glasačkoj kutiji ili na drugom uređaju. Ovakav način se razlikuje od DRE po tome da se ništa ne sprema u DRE uređaj te je stoga nemoguće manipulirati memorijom koja sadrži glasove.

U članku „Online voting“ objavljenom 2001. godine od Britanskog parlamentarnog ureda za znanost i tehnologije se spominje sljedeća podjela elektroničkog glasovanja [2]:

- Zbrajanje uređajem koje podrazumijeva čitanje glasačkih listića koji glasove imaju označene zaokruživanjem olovkom ili markerom određena mjesta ili pak bušenjem određenih mjesta. Ova druga vrsta bila je korištena tijekom predsjedničkih izbora u Americi 2000. godine. Nakon što se glasovi na bilo koji od ovih načina označe, oni se zatim bilježe u uređaju na glasačkom mjestu i šalju centralnom uređaju.
- Glasovanje na računalu, takozvani DRE uređaji za elektroničko glasovanje. Oni uključuje glasovanje putem zaslona osjetljivog na dodir, tipkovnice i pointera. Izvorni DRE uređaji su bili jednostavne elektroničke implementacije tradicionalnih mehaničkih uređaja dok su današnji DRE uređaji više nalik na bankomate ili osobne uređaje i imaju više mogućnosti poput prikaza fotografija i slično.
- Internet glasovanje koje može biti provedeno na 4 načina:
  - Glasovanje tj. glasački sustav zahtijeva od korisnika da posjeti biračko mjesto s osobljem i tamo obavi glasovanje nakon čega se glasovi šalju u centralno mjesto obrade i spremanja glasova.
  - Ovo glasovanje je slično prethodnom, jedino je razlika da glasač može otići na bilo koje biračko mjesto u određenoj regiji, a sustav prilikom pristupanju glasovanju provjerava da li je korisnik već glasovao i onda mu isporučuje glasački listić na temelju podataka gdje živi.
  - Glasački kiosci koji su smješteni na prigodnim mjestima kao što su trgovački centri ili pošta koji nisu nadgledani od radnika na glasačkim mjestima te takva glasovanja mogu trajati i više dana ili tjedana.
  - Glasovanje koje omogućava da korisnik glasa na bilo kojem mjestu gdje ima pristup Internetu i računalu. Glasanje se može čak provesti preko digitalnih tv-a ili mobitela. Ovakva vrsta glasovanja se može koristiti tako da zamijeni glasovanje na biračkom mjestu u potpunosti ili drugi način da glasaju ljudi koji nemaju pristup biračkom mjestu.



## 2.2. Prednosti i nedostaci elektroničkog glasovanja

Očito je da elektronička glasovanja donose mnogo prednosti jer se inače ne bi ni razvijala. Naravno da ona sa sobom donose i mnoge nedostatke. Prednosti i nedostaci koji će biti nabrojati su većinom preuzeti od IDEA (The International Institute for Democracy and Electoral Assistance) te se oni odnose većinom na politička glasovanja ali se mogu i povezati s ostalim glasovanjima. IDEA-a u svojim publikacijama savjetuje, predlaže i donosi iskustva za različite sustave za elektroničko glasovanje i zaključuje da nema točno definiranih smjernica i standarda kako bi se takav sustav trebao izraditi. Tradicionalna glasovanja imaju mnogo nedostataka koji se pokušavaju riješiti implementacijom elektroničkog glasovanja ali kako se na takav način riješe problemi tradicionalnog glasovanja oni istovremeno povlače niz drugih problema koji se vežu uz elektroničko glasovanje. U sljedećem dijelu su pobrojani nedostaci i prednosti elektroničkog glasovanja [4]:

- Prednosti:
  - Glasovanje na elektronički način ubrzava proces obrade podataka.
  - Smanjuje se rizik od ljudske pogreške.
  - Efikasno se mogu koristiti složene formule za obradu glasova.
  - Poboljšan je prikaz kompliciranih glasačkih listića.
  - Povećavanje dostupnosti, a samim time i veći odaziv.
  - Smanjenje rizika ljudske manipulacije glasovanja kod prebrojavanja ili zapisivanja broja glasova.
  - Veća i lakša dostupnost glasovanja ljudima s posebnim potrebama, onima koji su u nemogućnosti glasati zbog posla ili oni koji su previše udaljeni od glasačkog mjesta.
  - Veće mogućnosti korištenja višejezičnih sučelja kod glasovanja.
  - Manji rizik za nevažeće listiće zbog mogućnosti sustava da upozori na takve
  - Dugoročno smanjenje troškova izbora.
  - Smanjenje rizika za prodavanje glasova ili utjecaja obitelji na glas kroz mogućnost da se glasa više puta i samo zadnji glas vrijedi.

- Nedostaci:
  - Glasovanje je relativno netransparentno.
  - Ograničena otvorenost, odnosno shvaćanje sustava samo ekspertima.
  - Nedostatak standarda za elektroničko glasovanje.
  - Mogućnost povrede anonimnosti glasača .
  - Rizik od manipulacije unutarnjih ljudi koji imaju pristup sustavu ali i hakera izvana.
  - Povećanje troškova za izradu i održavanje sustava za elektroničko glasovanje.
  - Ograničena mogućnost prebrojavanja.
  - Potreba za dodatnim podučavanjem glasača.
  - Potencijalno smanjenje broja glasača zbog nepovjerenja u sustave za elektroničko glasanje.

### 3. Postojeće web aplikacije za elektroničko glasovanje

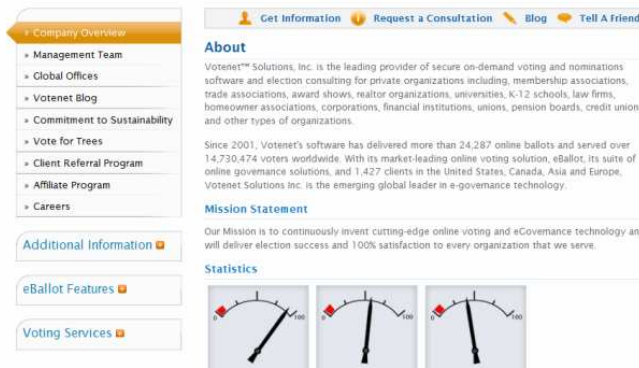
Već je prije spomenuto da postoje različiti sustavi elektroničkog glasovanja. U ovom će poglavlju biti opisane web aplikacije koje se u tu svrhu mogu koristiti. Kod opisa aplikacija je spomenuto je nešto o početnoj stranici. Nakon toga je opisano kako aplikacija podržava osnovne mogućnosti i standarde koje bi takve aplikacije trebale imati. Zatime je opisan je dio aplikacije gdje su demonstracije i tutorijali te je analizirana aplikaciju uspoređena s izrađenom. Na kraju su u par rečenica su spomenuti minusi i plusevi. Kod svake aplikacije su još priložena dva snimka zaslona i to početne stranice i primjera jednog glasovanja.

#### 3.1. eBallot<sup>1</sup>

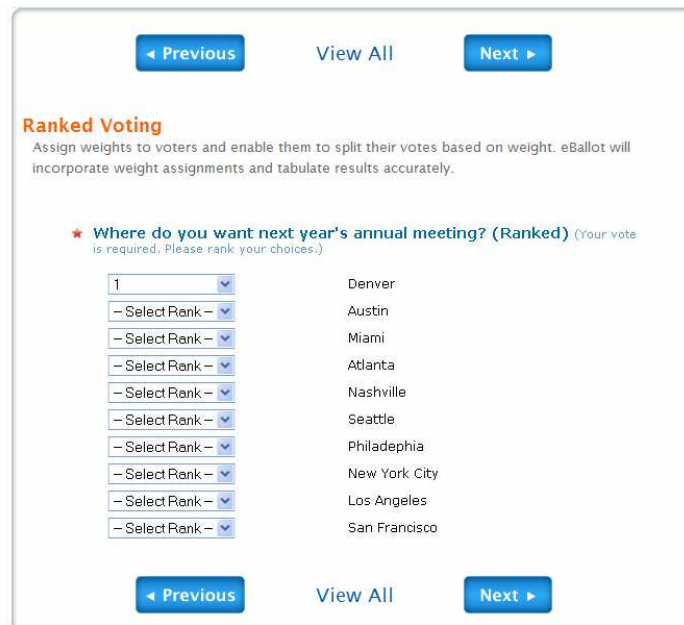
Na početnoj stranici se nalazi logo aplikacije i navedena je tvrtka koja je vlasnik glasovanja. Kada pogledamo mogućnosti sa korisnicima, aplikacija nudi više načina anonimnosti, tj. načina na koji se korisnik može povezati sa svojim glasom. Npr. jedan način je da odgovore može vidjeti samo vlasnik glasovanja, drugi da mogu vidjeti svi sudionici glasovanja, treći da vidi bilo tko itd. Aplikacije daje opciju za glasovanje takvu da vlasnik glasovanja nikad ne vidi tko je kako glasao, a korisnici mogu vjerovati jer je to aplikacija treće strane, tj. ne pripada vlasniku glasovanja. Korisnici se mogu logirati i putem Single Sign-on tehnologije. Podržano je i samostalno nominiranje korisnika i prihvaćanje nominacija od strane vlasnika. Sučelje aplikacije je napravljena u velikoj mjeri pomoću AJAX tehnologije. Glasovanje je omogućeno i putem telefona ili hibridno preko papira. Za kreirano glasovanje ili svoj korisnički račun tvrtka može postaviti prigodan tekst i logo. Kod pitanja je moguće postaviti sadržaj u obliku tekst slike, video ili zvuk koje daje dodatne informacije o glasovanju. Aplikacija ima i mogućnost slanja emailova za različite prilike kao što su poziv na glasovanje i različita upozorenja ili obavijesti. Podržana je i mobilna verzija. Aplikacija sadrži brojne upute kod korištenja, a sadrži i popis svih mogućnosti koje su dostupne korisniku. Kad bi se ova aplikacija usporedila s izrađenom može se zaključiti da ona sadrži sve mogućnosti kao izrađena. Prednost aplikacije je korištenje AJAX tehnologije u velikoj mjeri, a za minus se ništa ne ističe jer aplikacija nudi sve što se od takve aplikacije očekuje te još više.

---

<sup>1</sup> Web adresa aplikacije: <http://eballot.votenet.com/>



Slika 3.1. Početna stranica aplikacije „eBallot“



Slika 3.2. Primjer glasovanja kod aplikacije „eBallot“

## 3.2. Big Pulse<sup>2</sup>

Aplikacija na početnoj stranici sadrži logo i dodatne informacije koje opisuju njezine funkcionalnosti, ali nema spomenuto tko je vlasnik. Postoje četiri mogućnosti kako da kontroliramo tko može glasati. One su pobrojane i opisane u sljedećem dijelu:

- Generiranje glasačkih korisničkih računa. Kreiranje novih korisničkih imena i lozinke za pojedino glasovanje ili njihovo generiranje od aplikacije prema listi s email-ovima. Korisnik dobiva poziv na glasovanje putem email u kada klikne na link automatski se prijavi i može glasovati
- Link s druge web stranice, glasovanje je smješteno na Big pulse serveru. Ovaj način je prigodan kada korisnik ima account na drugoj stranici. Postoji veze s web stranice na big Pulse server i kada korisnik klikne link za glasanja, kreira mu se account.
- Glasovanje korisnika s postojećim korisničkim računom na Big pulse.
- Slanje PIN korisniku. Nakon upisivanja pina korisnik se preusmjerava na popunjavanje osobnih podataka ili direktno na glasovanje ako je već ima profil na aplikaciji.

Glasovanja koja nudi ova stranica su obična glasovanja, glasovanja s rangovima, kumulativna glasovanja gdje korisnik može svoj glas dati jednom kandidatu ili rasporediti na više, glasovanje gdje korisnik može različito rangirati glasove, glasovanje s različitim težinama glasova itd.

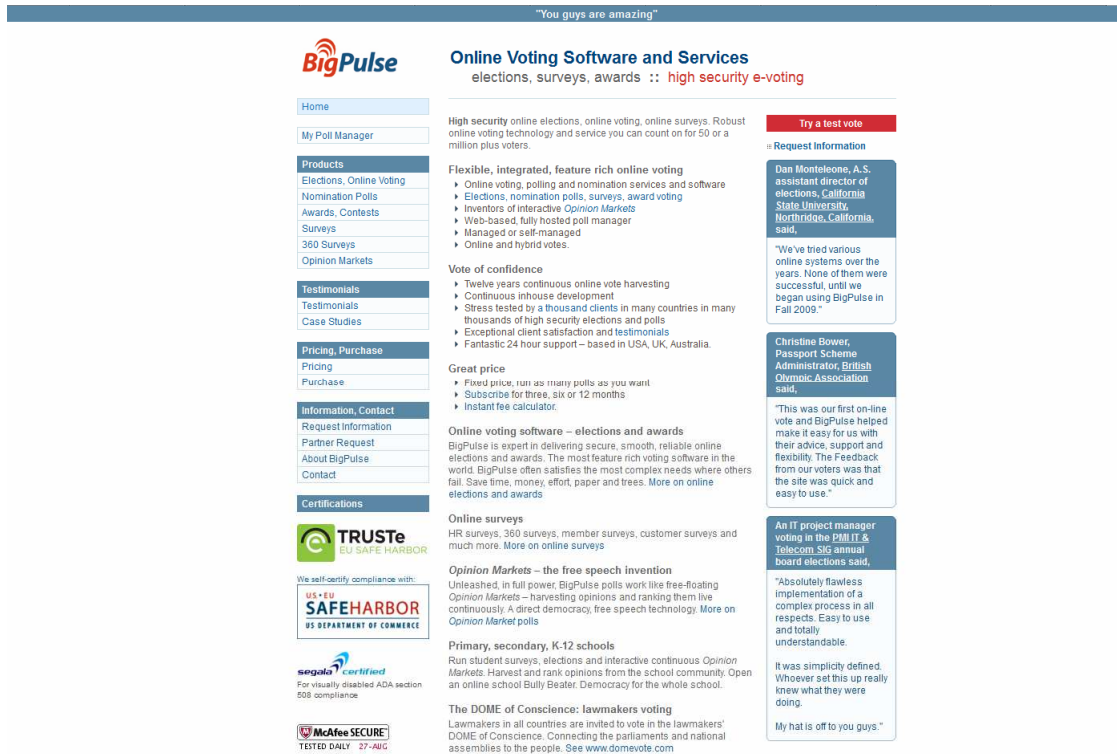
Neke od mogućnosti koje ću još spomenuti su nominacije online, grupe ljudi za različita glasovanja, više razina anonimnosti glasača, dio s informacijama o izboru gdje se može uploadati tekst, slike i mp3 itd.

Demo je omogućen tako da se na početnoj stranici zatraži testno glasovanje čiji link onda stiže na željenu email adresu. Na tom linku se nalazi više primjera različitih glasovanja.

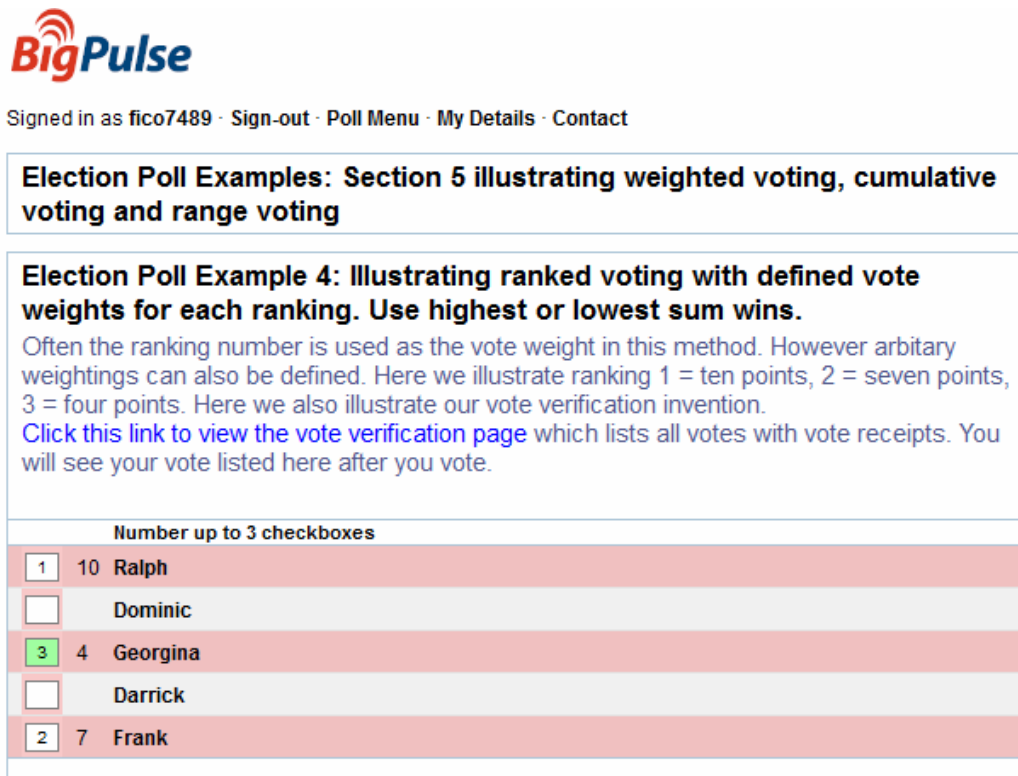
Minusi aplikacije bi bili slabo podržan AJAX te da nema demonstracije za postupak kreiranja glasovanja i odabira glasača i pregleda rezultata.

---

<sup>2</sup> Web adresa aplikacije: <http://www.bigpulse.com/>



Slika 3.3. Početna stranica aplikacije „BigPulse“



Slika 3.4. Primjer glasovanja kod aplikacije „BigPulse“

### 3.3. Simply voting<sup>3</sup>

Treća aplikacija koju ću analizirati je „Simply voting“. Na njezinoj početnoj stranici se može zaključiti čemu ona služi. Kad promatramo aplikaciju u smislu šta ona pruža u radu sa korisnicima možemo zamijetiti da nije podržana lista glasača koja bi se koristila za više glasovanja nego se za svako glasovanje kreira nova lista. Korisnici se mogu kreirati se može ručno ili se mogu učitati iz csv datoteke. Glasovanja se kreiraju tako da se prvo nazove i kreira glasovanje te se zatim dodaju pitanja. Opcije koje postoje su glasovanje s rangiranjem, glasovanje s odabirom i unos teksta. Za svaki od tih odabira postoje još dodatne opcije.

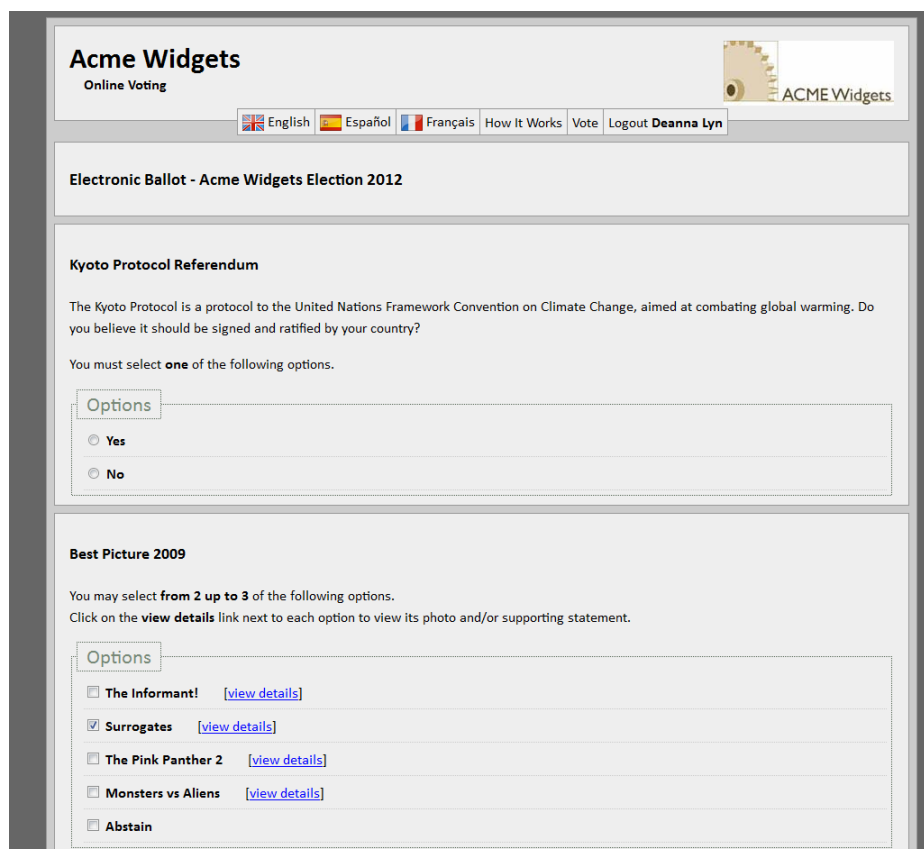
Ostale mogućnosti koje ću još biti spomenute je da se za pitanje može kreirati dodatni informativni tekst sa slikom kod kreiranja i kasnije kod uređivanja glasovanja. Dio aplikacije gdje se objašnjava funkcioniranje i način rada postoje dva načina. Tako da se besplatno registrira i onda se može besplatno kreirati testna glasovanja i vidjeti kako izgleda sučelje i drugi način, demonstracijski dio, gdje se prikazuje primjer kako izgleda popis glasovanja, glasovanje i rezultati. Popis funkcionalnosti u aplikaciji je vrlo skroman. Ova aplikacije ne podržava sve funkcionalnosti kod glasovanja, uključujući natjecateljsko pitanje, koje podržava izrađena aplikacija. To se kod nje i može spomenuti kao glavni minus.

---

<sup>3</sup> Web adresa aplikacije: <http://www.simplyvoting.com/>



Slika 3.5. Početna stranica aplikacije „Simple Voting“



Slika 3.6. Primjer glasovanja kod aplikacije „Simple Voting“



## 4. Tehnologije korištene kod izrade aplikacije sa primjerima u kôdu aplikacije

U ovom dijelu rada će biti opisane tehnologije koje su korištene kod izrade aplikacije. Tehnologije će biti opisane na primjeru odabranih dijelova kôda iz aplikacije.

### 4.1. HTML

Html je jezik oznaka (predstavljaju strukturne, prezentacijske i semanticke informacije). On se sastoji se od hipertekst-hipermedija osobina poput tekst, slika, zvuk, film i dr. koji se mogu međusobno povezati čime se dobije mreža informacija. HTML razvio Tim Berners-Lee na CERN-u, a popularnost dobiva na pregledniku Mosaic koji je razvijen na NCSA (National Center for Supercomputer Applications) [7, str. 5].

Postoje mnoge verzije ali verzija HTML5 se najčešće koristiti jer pruža mnoge napredne mogućnosti iako to još nije standardna verzija. Niti jedan od glavnijih web preglednika ne podržava HTML5 u potpunosti, ali oni u sve većem postotku pdoržavaju njegove mogućnosti [11].

Osnovna struktura HTML-a koju bi svaka web stranica trebala imati izgleda kao sljedeći primjer:

```
<!DOCTYPE html>
<html>
<head>
<title>Naslov dokumenta...</title>
</head>

<body>
Sadržaj dokumenta...
</body>

</html>
```

### 4.1.1. HTML Elementi

- HTML elementi počinju s početnim tagom.
- HTML elementi završavaju sa završnim tagom.
- Sadržaj HTML elementa je sve prije završnog i poslije početnog taga.
- Prazni elementi su zatvoreni u početnom tagu.
- Većina HTML elementa može imati attribute.
- Većina HTML elementa može biti ugniježdjena.

Pogledajmo sad primjer HTML kôda iz aplikacije:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>E-voting</title>

  <link rel='stylesheet' type='text/css' href='css/jquery.lightbox-0.5.css' />
  <link rel="stylesheet" type="text/css" href="css/ui-lightness/jquery-ui-
1.8.23.custom.css" />
  <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
  <link rel="stylesheet" type="text/css" href="css/bootstrap-responsive.min.css" />
  <link rel='stylesheet' type='text/css' href='css/style.css' />

  <script type="text/javascript" src="js/jquery-1.7.2.min.js"></script>
  <script type="text/javascript" src="js/jquery-ui-1.8.20.custom.min.js"></script>
  <script type="text/javascript" src="js/jquery.ui.datepicker-hr.js"></script>
  <script type="text/javascript" src="js/timepicker.js"></script>
  <script type="text/javascript" src="js/ui.spinner.min.js"></script>
  <script type="text/javascript" src="js/jquery.livequery.js"></script>
  <script type="text/javascript" src="js/bootstrap.min.js"></script>
</head>
```

U prethodnom HTML primjeru kôdu iz aplikacije možemo primijetiti oznaku *<!DOCTYPE html>* oznaku koja se stavlja prije bilo kojeg HTML kôda i ona označuje kako preglednik interpretira HTML kôd. U ovom slučaju ga interpretira kao HTML5. U zaglavlju možemo vidjeti meta oznaku koja daje upute browseru kako da dekodira znakove, u ovom slučaju na utf-8 način. Zatim slijedi niz od 4 link tagova koji daju web pregledniku instrukcije da učita vanjski CSS te nakon toga niz od 7 script tagova koji daju browseru upute da učita vanjske JavaScript datoteke.

Tipični primjeri HTML elementa su:

- Paragraf.

```
<p>Ovo je paragraf.</p>
```

- Slika (atribut src nam daje putanju do slike iz trenutnog direktorija).

```
</img>
```

- Naslov prve razine.

```
<h1>Naslov</h1>
```

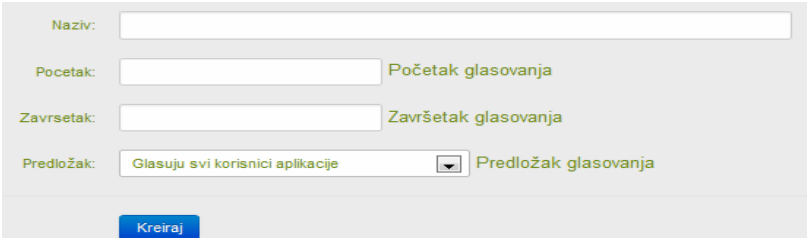
- Lista s 3 elementa liste.

```
<ul>
  <li>Lista 1</li>
  <li>Lista 2</li>
  <li>Lista 3</li>
</ul>
```

#### 4.1.2. Forme

Forme su vrlo bitan dio web stranica i HTML-a. Za primjer forme je uzeta jedna u izrađenoj aplikaciji, točnije ona za kreiranje glasovanja i ona je prikazana na slici 4.1. Iz primjera kôda ove forme, koje je prikazan u nastavku, su izbačeni *div* i *span* elementi koji se nalaze u aplikaciji samo zbog dizajna, a ovdje je cilj prikazati samo dijelove forme.

```
<form class="form-horizontal" action="" method="post">
  <input type="text" id="naziv" name="naziv" value="" class="span6"></input>
  <input type="text" name="pocetak" id="pocetak" value="" class="datepicker_mysql"></input>
  <input type="text" name="zavrsetak" id="zavrsetak" value=""
class="datepicker_mysql"></input>
  <select name="predlozak" id="predlozak" style="width: 300px;">
    <option value="1" id="predlozak-0">Glasuju svi korisnici aplikacije</option>
    <option value="2" id="predlozak-1">Glasuju samo prijatelji korisnika</option>
  </select>
  <input type="submit" value="Kreiraj" class="btn btn-primary"></input>
</form>
```



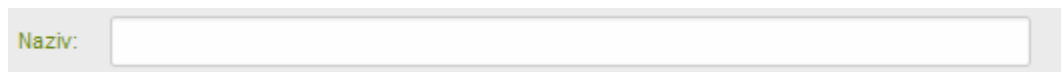
Slika 4.1. Forma za kreiranje glasovanja

U tagu `form` se nalazi atribut `action` koji predstavlja naziv skripte koja će se izvršiti kada pošaljemo formu te atribut `method` koji predstavlja način kako će se podaci poslati toj skripti. Podaci koji se šalju skripti su svi `input`, `select` i `textarea` tagovi sa svojim vrijednostima. Postoje još neki koji se automatski šalju, ali samo ovi su korištene kod aplikacije. Forma se šalje klikom na submit gumb, u ovom primjeru on ima vrijednost „Kreiraj“, slika 4.1.

U nastavku će biti pobrojani svi elementi forme koji su korišteni kod izrade aplikacije i to na način da će prvo biti napisan naziv elementa forme, zatim primjer kôda iz aplikacije te na kraju vizualni primjer iz aplikacije.

- Tekst (unos kratkog teksta)

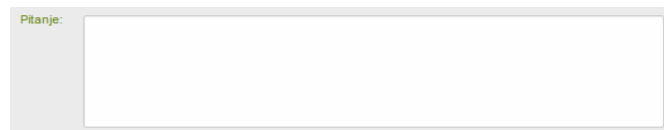
```
<input type="text" id="naziv" name="naziv" value="">input</input>
```



Slika 4.2. Primjer tekstualnog unosa

- Tekstualno polje (Unos teksta sa više redova i kolona)

```
<textarea type="text" id="pitanje" name="pitanje" value="" cols="60" rows="5"></textarea>
```

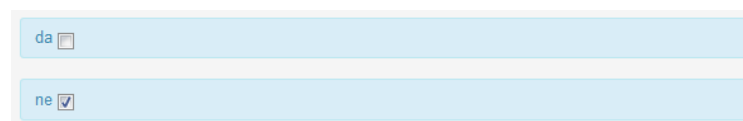


Slika 4.3. Primjer tekstualnog unosa, sa više redova

- Polje za označavanje (checkbox)

```
da <input type="checkbox" name="52" value="52" ></input>
```

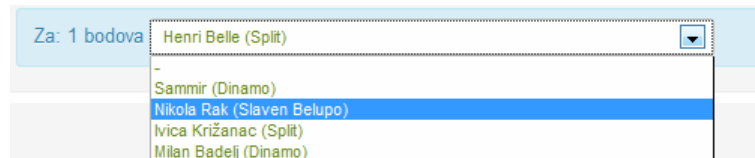
```
ne <input type="checkbox" name="53" value="53" checked="" ></input>
```



Slika 4.4. Primjer unosa sa označavanjem

- Padajući izbornik

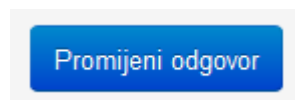
```
<select name="natjecateljski_1" class="span5">
  <option value="0"> - </option>
  <option value="35">Sammir (Dinamo)</option>
  <option value="36">Nikola Rak (Slaven Belupo)</option>
  <option value="37">Ivica Križanac (Split)</option>
  <option value="38">Milan Badelj (Dinamo)</option>
</select>
```



Slika 4.5. Primjer padajućeg izbornika

- Gumb za slanje

```
<input type="submit" class="btn" value="Promijeni odgovor"></input>
```



Slika 4.6. Primjer gumba za slanje

## 4.2. PHP

PHP je skriptni jezik koji je prigodan za izgradnju dinamičkih web stranica. Njegova primarna namjena je skriptiranje na serverskoj strani, ali se može koristiti i sučelju komandne linije i grafički samostalnim aplikacijama. PHP je skraćenica za PHP Hypertext Preprocessor. Sintaksa PHP-a je slična sintaksi C jezika, ali je pojednostavljena [5, str. 138].

PHP kôd se može ugrađivati unutar HTML-a tako da PHP kôd upišemo unutar oznaka:

```
<?
PHP kôd
?php>
```

Na sljedećem dijelu je prikazanao kako se PHP kôd ugrađuje u HTML. Ovdje bi se u tijelu HTML-a ispisalo „Hello world!“.

```
<!DOCTYPE html>
<html>
<head>
  <title>Title of the document</title>
</head>
  <body>
    <?php echo „Hello world!“ ?>
  </body>
</html>
```

#### 4.2.1. Varijable

Sve varijable u PHP-u počinju sa znakom „\$“, i u svakom trenutku bilo kojoj varijabli možemo dodijeliti bilo koju vrijednost.

Varijable u PHP-u mogu biti sljedećeg tipa [6]:

- Integer - Cjeli brojevi.
- Float - Decimalni brojevi.
- String - nizovi znakova.
- Boolean - true i false vrijednosti.
- Array - Polja.
- Object - koristi se za spremanje instance klase.
- Posebne varijable - resource i NULL.

U php-u postoje superglobalne varijabli koje su dostupne u svakom upitu i mogu imati bilo koju vrijednost od gore spomenutih. U sljedećem dijelu rada će biti spomenute samo neke koje su korištene kod izrade aplikacije, ali ih postoji više:

- `$_POST` - ovdje se spremaju vrijednosti iz forme kada je forma poslana na POST način.
- `$_GET` - ovdje se spremaju vrijednosti iz forme kada je forma poslana na GET način, tj. kada se poslani parametri nalaze u url-u.

- `$_SESSION` - u ovu varijablu spremamo sve podatke koje želimo da nam budu dostupni tijekom korisnikovog korištenja stranicom.
- `$_SERVER` - ovdje se spremaju različite vrijednosti od upita, npr. ip adresa.

#### 4.2.2. Funkcije

Sada će biti spomenute funkcije za rad s bazom (MySQL). Kod nabiranja će prvo biti prikazna funkcija iz službene dokumentacije PHP-a te nakon toga opis funkcije. Nakon popisa ovih funkcija bit će prikazan i primjer njihovog korištenja u izrađenoj aplikaciji. Postoje više funkcija, ali u sljedećem dijelu će biti nabrojane i opisane samo one koje su korištene kod izrade aplikacije [10]:

- `resource mysql_connect ( [ string $server = ini_get("mysql.default_host") [, string $username = ini_get("mysql.default_user") [, string $password = ini_get("mysql.default_password") [, bool $new_link = false [, int $client_flags = 0 ]]]]] )`

Ova funkcija nam služi za spajanje na bazu i vraća referencu na tu konekciju. Prvi argument je naziv servera, drugi username, treći password, četvrti argument je opcionalan i ne koristim ga u aplikaciji.

- `bool mysql_select_db ( string $database_name [, resource $link_identifier = NULL ] )`

Ovom funkcijom odabiremo bazu koju želimo koristiti. Prvi argument je naziv baze, drugi je referenca na konekciju

- `resource mysql_query ( string $query [, resource $link_identifier = NULL ] )`

Ovom funkcijom izvršavamo upit nad bazom. Prvi argument je upit, a drugi referenca na postojeću konekciju.

```
mysql_query("set names utf8");
```

- `array mysql_fetch_array ( resource $result [, int $result_type = MYSQL_BOTH ] )`

Ova funkcija vraća jedan red iz baze na temelju reference od upita. Prvi argument je referenca na postojeću konekciju, a drugi opcionala i ne koristim ga u aplikaciji.

- `int mysql_num_rows ( resource $result )`

Ova funkcija vraća broj redova koliko vraća dana upit. Prvi argument je referenca na postojeću konekciju.

Evo nekih primjera gdje koristim navedene funkcije u aplikaciji. Funkcija za spajanje na bazu *bazaSpoji()* koja je prikazana u nastavku teksta koristi prve tri funkcije. Prvo se spajamo na

mysql server na računalu localhost, pod korisničkim imenom „root“ i praznom lozinkom, zatim odabiremo bazu *evoting* za korištenje te na kraju izvršavamo upit kojim određujemo kako će mysql kodirati znakove.

```
function bazaSpoji(){
    mysql_connect('localhost', 'root', '') or die(mysql_error());
    mysql_select_db('evoting') or die(mysql_error());
    mysql_query("set names utf8") or die(mysql_error());
}
```

Sljedeći isječak kôda je primjer iz aplikacije gdje koristimo funkciju *mysql\_num\_rows()* koja nam u ovom slučaju vraća broj pitanja od glasovanja sa određenim id-om te primjer kako koristimo funkciju *mysql\_fetch\_array()* koja u svakom koraku petlje vraća polje sa jednim pitanje koje se onda spremaju u varijablu.

```
$sql = "SELECT * FROM pitanje WHERE id=$id";
$broj = mysql_num_rows($r);
$r = mysql_query($sql);
while($row = mysql_fetch_array($r)){
    $pitanja[] = $row;
}
```

Sada ću pobrojati ostale funkcije koje sam koristio. Prvo ću prikazati kako izgledaju u službenoj dokumentaciji PHP-a, zatim objasniti kako se koriste te ću na kraju za neke pokazati primjer u mojoj aplikaciji:

- string **date** ( string \$format [, int \$timestamp = time() ] )  
Vraća formatirano vrijeme prema formatu koji joj damo u prvom argumentu. Drugi argument ako želimo neko drugo vrijeme, a ne trenutno.
- bool **file\_exists** ( string \$filename )  
Funkcija vraća true ako postoji file čiji je naziv dat u argumetu ili false ako ne postoji
- int **fwrite** ( resource \$handle , string \$string [, int \$length ] )  
Zapisivanje u datoteku
- bool **fclose** ( resource \$handle )  
Zatvaranje veze na datoteku



- resource **fopen** ( string \$filename , string \$mode [, bool \$use\_include\_path = false [, resource \$context ] ] )  
Otvaranje veze na datoteku
- bool **session\_start** ( void )  
Funkcija koja kreira novu sesiju ako trenutna ne postoji, a ako postoji onda ju samo zapisuje u cookie
- bool **session\_destroy** ( void )  
Funkcija prekida sessiju
- bool **empty** ( mixed \$var )  
Funkcija provjerava da li varijabla prazna
- void **header** ( string \$string [, bool \$replace = true [, int \$http\_response\_code ] ] )  
Funkcija stavlja dodatne argumenta u http zahtjev. Sljedeći dio kôda iz aplikacije stavlja u http zahtjev dodatno polje da se aplikacija preusmjeri na početnu stranicu, kada je korisnik pristupio djelu stranice gdje ne smije.  

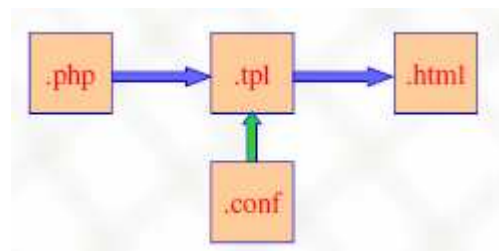
```
header( "Location: index_aplikacija.php?poruka=8" );
```
- void **exit** ( [ string \$status ] )  
Završanje skripte, nakon ove funkcije kôd se više ne izvršava
- int **strtotime** ( string \$time [, int \$now = time() ] )  
Funkcija iz formatiranog vremena mijenja u običan timestamp
- bool **isset** ( mixed \$var [, mixed \$... ] )  
Provjerava da li je varijabla postavljena
- float **round** ( float \$val [, int \$precision = 0 [, int \$mode = PHP\_ROUND\_HALF\_UP ] ] )  
Zaokružuje float vrijednost na datu u prvom argumentu na preciznost datu u drugom argumentu.

- bool **in\_array** ( mixed \$needle , array \$haystack [, bool \$strict = **FALSE** ] )

Funkcija provjerava dali broj dat u prvom argumentu postoji u polju koje je dano u drugom argumentu. Ako postoji vraća se true, ako ne vraća se false.

### 4.3. Smarty

Smarty je sustav predložaka i on služi za odvajanje logike od prezentacijskog dijela(HTML-a, CSS-a). Na taj je način proces moguće podijeliti na dva dijela [8]. Slika 4.1. prikazuje da kada radimo sa smarty-em postoje 4 vrste datoteka. PHP skripta gdje se nalazi sam kôd, tpl datoteka gdje se nalazi HTML kôd, te conf datoteka gdje se nalazi konfiguracija prema kojoj se tpl datoteka transformirati u HTML. Slika 4.7. vizualno prikazuje način rada smarty sustava.



Slika 4.7.

Smarty sustav [Kermek, Dragutin. Prezentacija sa predavanja.]

Najnovija stabilna verzija te ujedno i verzija koja je korištena u radu je Smarty 3.1. Smarty je zapravo skup skripta i biblioteka. Kada skinemo Smarty, stavimo ga u direktorij na web mjestu uključimo njegov file sa Smarty klasom u našu skriptu. Napravimo instancu te klase i njome sve radimo.

## PHP datoteka:

```
<?php
    $smarty_file = 'Smarty-3.1.11/libs/Smarty.class.php';
    include_once($smarty_file);
    $smarty = new Smarty();
    $smarty->assign(sadrzaj, 'Pozdrav od Smarty-a');
?>
```

## Tpl datoteka:

```
<html>
<head>
<title> Smarty </title>
</head>
<body>
    {$sadrzaj}
</body>
</html>
```

U body-u će se umjesti `{$sadrzaj}` zapisati „Pozdrav od Smarty-a“.

Smarty ima još mnogo mogućnosti od kojih su neke korištene za potrebe izrade web aplikacije u ovom radu. U nastavku će biti prikazani još *if* uvjet i *foreach* petlja. U sljedećem dijelu je prikazan kôd koji se nalazi u php i tpl datoteci kod primjera *if* uvjeta:

```
$smarty->assign(sadrzaj, 'Pozdrav od Smarty-a.');
```

```
$smarty->assign(sadrzaj2, 'Ovo je html.');
```

```
$smarty->assign(uvjet, 2);
```

```
<html>
<head>
<title> Smarty </title>
</head>
<body>
    {if $uvjet==2 }
        {$sadrzaj2}
    {else}
        {$sadrzaj}
    {/if}
</body>
</html>
```

Ovdje će se ispisati „Ovo je html“. U sljedećem dijelu je opet prikazan kôd koji se nalazi u php i tpl datoteci kod primjera s *foreach* petljom.

```
$polje = array('PHP', 'AJAX', 'jQuery');

<html>
<head>
<title> Smarty </title>
</head>
<body>
    {foreach from=$polje key=k item=v}
        {$v}
    {/foreach}

</body>
</html>
```

Ovdje će se ispisati 'PHP, AJAX, JQuery.

## 4.4. JavaScript i JQuery

JavaScript je programski jezik koji se najčešće koristi za programiranje na klijentskoj strani, a to znači da se izvršava na klijentovom pregledniku. Postoje mnoge JavaScript biblioteke koji uvelike skraćuju pisanje JavaScript kôda, pogotovo onaj dio koji se odnosi na AJAX, a jedna od tih biblioteka je JQuery.

Jquery je JavaScript framework koji uvelike pomaže kod pisanja JavaScript kôda i skraćuje ga. JQuery je jednostavno koristiti. Preuzmemo ga sa službene stranice i samo ga uključimo u zaglavlju stranice te ga nakon toga možemo koristiti.

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
    $(document).ready(function(){
        alert("Pozdrav. Ovo je JQuery");
    });
};
</script>
</head>
```

```
<body>
</body>
</html>
```

Sav Jquery kôd bi se treba nalaziti u funkciji `$(document).ready(function(){ });` zato jer se sav kôd unutar te funkcije počinje izvršavati tek kad je cijela stranica učitana. U ovom primjeru će se ispisati “Pozdrav. Ovo je JQuery“ u pop up prozoru kad se stranica učita.

Sada će biti prikazane neke osnove Jquery-a, a zatim i neke složenije stvari na primjeru u aplikaciji koje su korištene kod izrade aplikacije. Jquery se uglavnom bazira na selektorima i ako želimo dobro znati Jquery moramo dobro znati selektirati elemente po HTML dokumentu, ali i po XML i JSON dokumentima.

Sljedeći dio prikazuje neke primjere selektiranja u JQuery-u:

- `$()`  
Elemente selektiramo tako da u zagradu upisujemo odgovarajuće vrijednosti.
- `$("p")`  
Selektiramo sve `<p>` elemente
- `$("p.intro")`  
Selektiramo sve `<p>` sa klasom `class="intro"`.
- `$("p#intro")`  
Selektiramo sve `<p>` sa id-om `id="intro"`.
- `$(":animated")`  
Selektiramo sve elemente koji se trenutno animiraju
- `$(":button")`  
Selektiramo sve `<button>` elemente i `<input>` elemente koji imaju tip `type="button"`.
- `$(":even")`  
Selektiramo sve parne elemente

Još jedan važni dio Jquery-a koji ću pokazati su događaji. Događaja postoje mnogo više ali naravno da svi nisu korišteni u aplikaciji.

U sljedećem isječku kôda iz aplikacije je događaj *click* vezan na elemente sa `class="posalji_zahhtjev"`, a to su linkovi pokraj korisnika kod ispisa korisnika koje se nalaze

na slici 4.8. Nakon što se klikne taj link pojavlja se pop up prozor sa pitanjem da li želimo poslati zahtjev korisniku, koji je prikazana na slici 4.9. Ako se odabere odgovor „da“, željenom korisniku se šalje zahtjev za prijateljstvom, u suprotnom ne dogodi ništa.

```

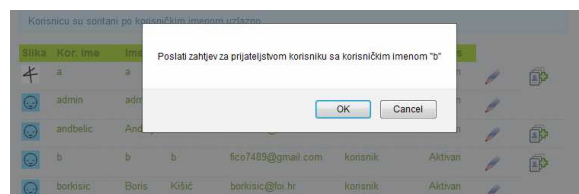
$(".posalji_zah_tjev").live('click', function(){
    var korisnik = $(this).attr("korisnik");
    var korime = $(this).attr("korime");
    var z = confirm('Poslati zahtjev za prijateljstvom korisniku sa korisničkim imenom "' +
korime + '"');

    if(z){
        kod za slanje zahtjeva;
    };
    return false;
});

```



Slika 4.8. Ispis korisnika u aplikaciji



Slika 4.9. Potvrda zahtjeva za prijateljstvom u aplikaciji

Jquery-u funkcije za rad s AJAX koje sam koristio su nabrojane u nastavku [12]:

- `getJSON()`;
- `get()`;
- `post()`;
- `ajax()`;

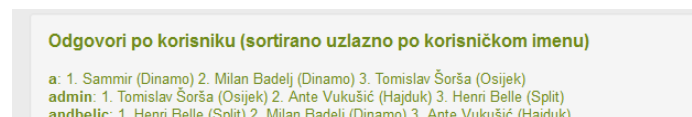
Sve tri funkcije `getJSON()`, `get()` i `post()` služe za AJAX upite i svaka je specijalizirana za posebne slučajeve dok je `ajax()` funkcija ima sve mogućnosti. U sljedećem isječku koda je prikazan primjer korištenja `getJSON()` funkcije u aplikaciji. Nakon što kliknemo gumb sa klasom `pokazi_korisnike`, a to je gumb „rezultati po korisnicima“ na slici 4.10. izvršava se AJAX upit na skriptu `dohvatiRezultate.php`. Nakon što server vrati podatke izvršava funkcija

koji se nalazi u drugom parametru funkcije *getJSON()*. U ovom slučaju se glasovi po pojedinom korisniku spremaju u varijablu *rezultati* te se zatim spremaju odgovarajući *div*.

```
$(".pokazi_korisnike").live('click', function(){
    var id_pitanje = $(this).attr("value");
    var status = $(".korisnici_rezultati[value=" + id_pitanje + "]).attr("status");
    var url = 'ajax/dohvatiRezultate.php?id_pitanje=' + id_pitanje;
    $.getJSON(url, function(data){
        var rezultati = '<span style="font-size: 20px;"><b>Odgovori po korisniku (sortirano
uzlazno po    korisničkom imenu) <br/><br/></b></span>';
        $.each(data, function(i, item) {
            var korime = item.korime;
            var odgovor = item.naziv;
            rezultati += '<b>' + korime + '</b>: ' + odgovor + '<br/>';
        });
        if(status == 1){
            $(".korisnici_rezultati[value=" + id_pitanje + "]).html(rezultati);
            $(".korisnici_rezultati[value=" + id_pitanje + "]).attr("status", 2);
            $(".korisnici_rezultati[value=" + id_pitanje + "]).show("explode");
        }
        else{
            $(".korisnici_rezultati[value=" + id_pitanje + "]).attr("status", 1);
            $(".korisnici_rezultati[value=" + id_pitanje + "]).hide();
        }
    });
    return false;
});
```



Slika 4.10. Prikaz gumba „rezultati po korisnicima“



Slika 4.11. Prikaz glasova prema korisniku

## 4.5. CSS

Netscape je 1994. godine predstavio svoj prvi komercionalni preglednik. Za razliku od drugih preglednika on je bio obogaćen novim HTML tagovima koji su omogućavali forme, tablice, povećane mogućnosti formatiranja teksta itd. Microsoft je shvatio da će web imati velik utjecaj na računalstvo i komunikacije te su oni su izdali svoj preglednik „Internet Explorer“. Tako je i „browserski rat“ između Netscape-a i Microsofta počeo te su oni sljedećih godina izrađivali mnogo novih tagova koji su dali nove mogućnosti. Ali nedostatak toga je bio da je HTML kôd postajao sve više kompleksniji i stranice su se prikazivale različitiije na drugim te je tad ideja o CSS-u rođena. Razvijanje CSS-a je W3C-ov napor da uvodi standardizirane tagove umjesto da kreatori browsera uvode nove za samo njihove browsere. CSS ne mijenja HTML nego povećava njegove mogućnosti, odnosno CSS se može smatrati kao dopuna HTML-u. To je najviše evidentno kada pogledamo kako css zapravo radi. CSS svojstva se dodaju postojećim HTML oznakama i prema njima se HTML renderira. CSS informacije mogu biti specificirane na tri načina [5, str. 137]:

- ugrađeno u header stranice
- ugrađeno u body stranice
- u vanjskoj datoteci

CSS upute u svojoj aplikaciji smjestio sam u datoteku style.css u mapu „css“. Primjer kako izgledaju CSS upute koje se koriste u aplikaciji je prikazan u sljedećem dijelu:

```
#header{
    width: 1160px;
    height: 160px;
    font-weight: bold;
    background-image: url('../slike/h3.png');
    margin-top: 50px;
    margin-bottom: 10px;
}

.ispis{
    min-width: 100px;
    color: #6D8A19;
}
```



Prvi dio odnosi se na sve HTML elemente koji *imaju id="header"* i za njih je prema CSS-u određeno da imaju duljinu 1160 pixela, širinu 160 pixela, da su slova podebljana, da je pozadinska slika h3.png, da je gornja margina 50 pixela, a donja margina 10 pixela. Ovom CSS uputom se formatira zaglavlje u izrađenoj aplikaciji.

Druga css uputa odnosi se na sve elemente koji imaju atribut *class="ispis"* i njima je minimalna duljina 100 pixela i boja teksta *6D8A19*. Ovom CSS uputom se odnose na tablicu s ispisom korisnika i glasovanja u aplikaciji.

## 5. Realizacija aplikacije

### 5.1. Opis funkcionalnosti aplikacije

Korisnik u aplikaciji može imati jedan od tri statusa, tj. postoje tri vrste korisnika. Prva vrsta je običan *korisnik*. Status običnog korisnika imaju korisnici nakon registracije. Takvi korisnici mogu pregledavati glasovanja, rezultate i glasovati. Da bi korisnik mogao kreirati i uređivati glasovanja mora postati *korisnik-admin*, što je ujedno druga vrsta korisnika. Treća vrsta korisnika je *admin* i on može sve što i prethodni korisnici te još može administrirati korisnike, tj. mijenjati njihove podatke, te mijenjati opće postavke aplikacije.

Svaki korisnik može vidjeti sve ostale korisnike i slati ima zahtjev za prijateljstvom. Nakon što korisnik pošalje zahtjev drugi korisnik ga može potvrditi za prijatelja ili odbiti.

Glasovanja mogu imati četiri statusa:

- U izradi - status koji glasovanje ima nakon što se kreira
- Aktivno - kada glasovanje ima aktivan status onda se na to glasovanje može glasovati i pregledavati rezultate
- Pasivno - kada glasovanje traje ono se privremeno može staviti u pasivno stanje i tada je ono zamrznuto, tj. ne može se glasati ni pregledavati rezultati
- Završeno - vlasnik glasovanje ručno postavlja glasovanje na status završeno kada je ono prošlo
- Obrisano - kada korisnik odustane od glasovanje i obriše ga

Svako glasovanje ima datum i vrijeme početka te predložak kako glasuju korisnici koji može biti postavljen tako da glasaju svi korisnici aplikacije ili samo oni koji su prijatelji s vlasnikom glasovanja. Dodatne opcije su način na koji će se prikazivati glasovi i mogu biti:

- Prikazuju se svima
- Prikazuje se samo sudionicima
- Prikazuje se samo vlasniku

Još postoji i opcija da li će se moći vidjeti glasovi po pojedinom korisniku. Ako je opcija postavljena na status „da“ glasove za svakog korisnika pojedinačno će moći vidjeti oni koji mogu i rezultate u suprotnom nitko ne može vidjeti.

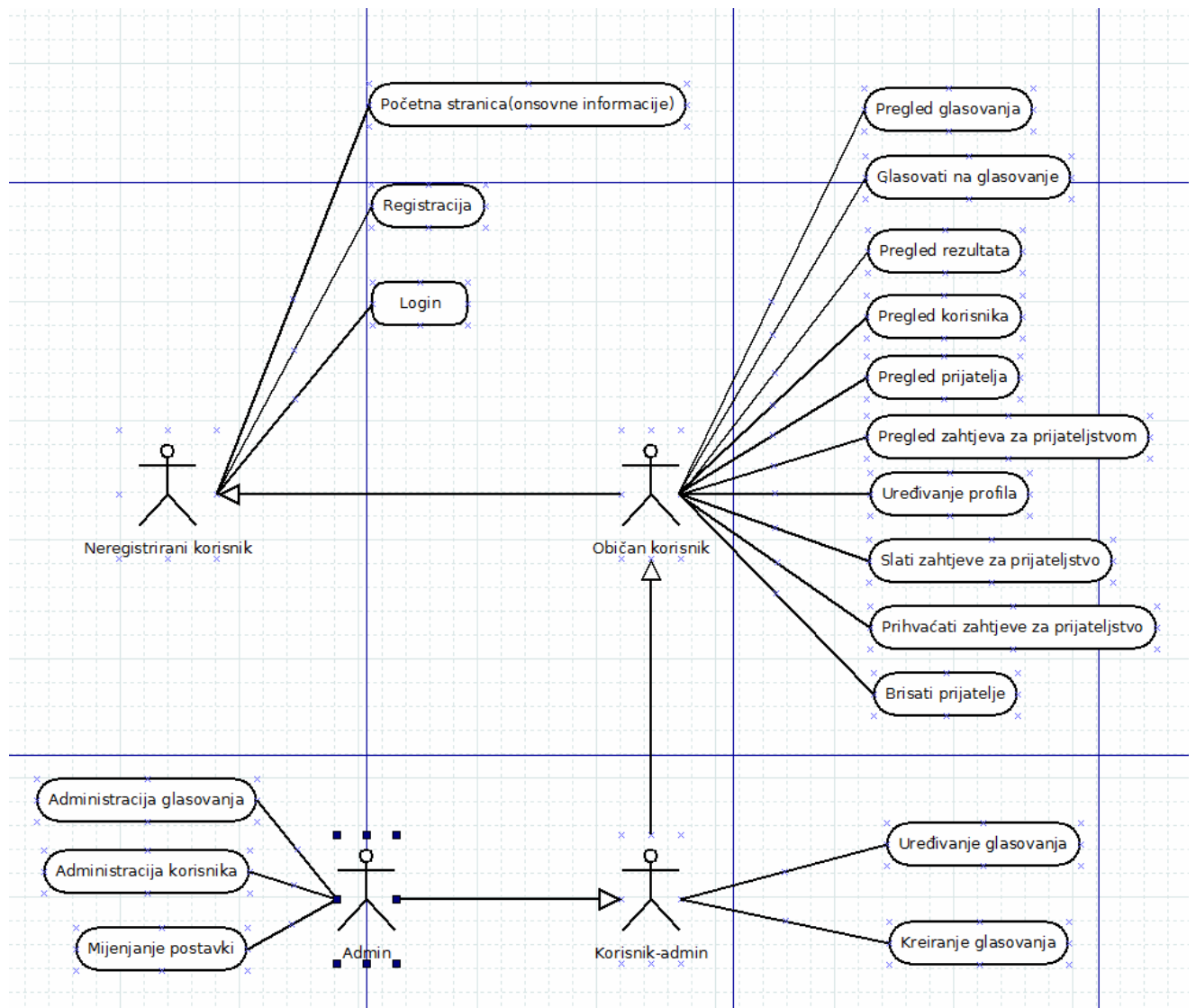
Svako glasovanje ima neograničen broj pitanja i odgovora. Vrste pitanja koje postoje su:

- Pitanja s odabirom - kod ovog pitanja korisnik može odabrati jedan ili više od ponuđenih odgovora, u zavisnosti s time koliko je dozvoljeno odabrati
- Tekstualno - Korisnik unosi tekst u odgovor
- Natjecateljsko - Korisnik odabire nekoliko izbora od ponuđenih s time da mjesto odabira ima težinu, npr prvi odabrani dobiva 10 bodova, drugi 3 i treći 1 bod

Za detaljnije razumijevanje aplikacije su napravljeni use case dijagram, koji definira interakcije između korisnika i sustava te navigacijski dijagram kojem je svrha ilustrirati navigaciju kroz web stranicu

## 5.2. Use case

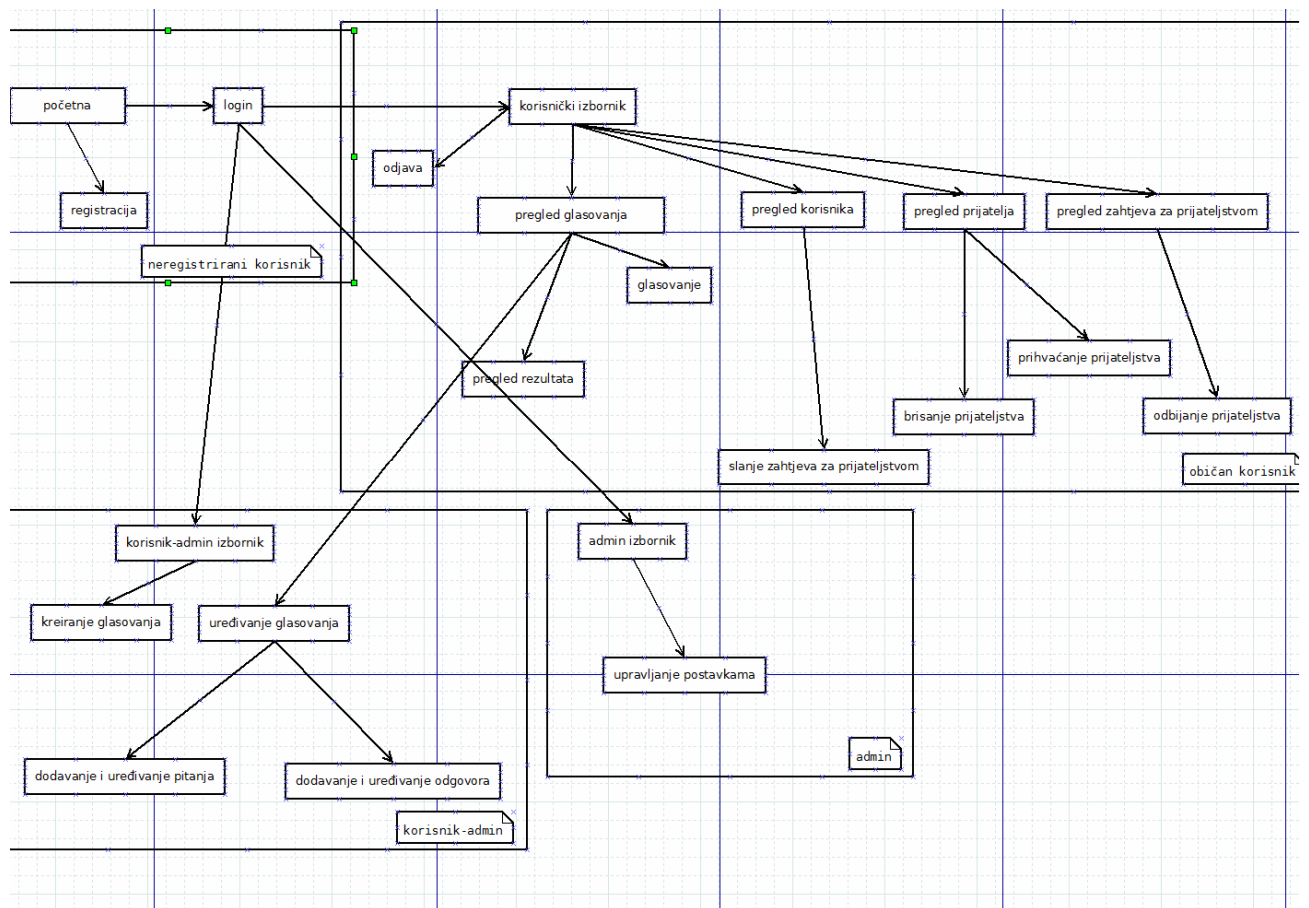
Na slici 5.1. je prikazana use case dijagram izrađene aplikacije. Na dijagramu se mogu primjetiti tri vrste korisnika koji su u prethodnom poglavlju bila spomenuta te još neregistrirani korisnik. Ovaj use case također prikazuje da „korisnik-admin“ naslijeđuje sve mogućnosti običnog korisnika, a da „admin“ naslijeđuje sve mogućnosti od korisnika sa statusom „korisnika-admin“.



Slika 5.1. Use case dijagram aplikacije

### 5.3. Navigacijski dijagram

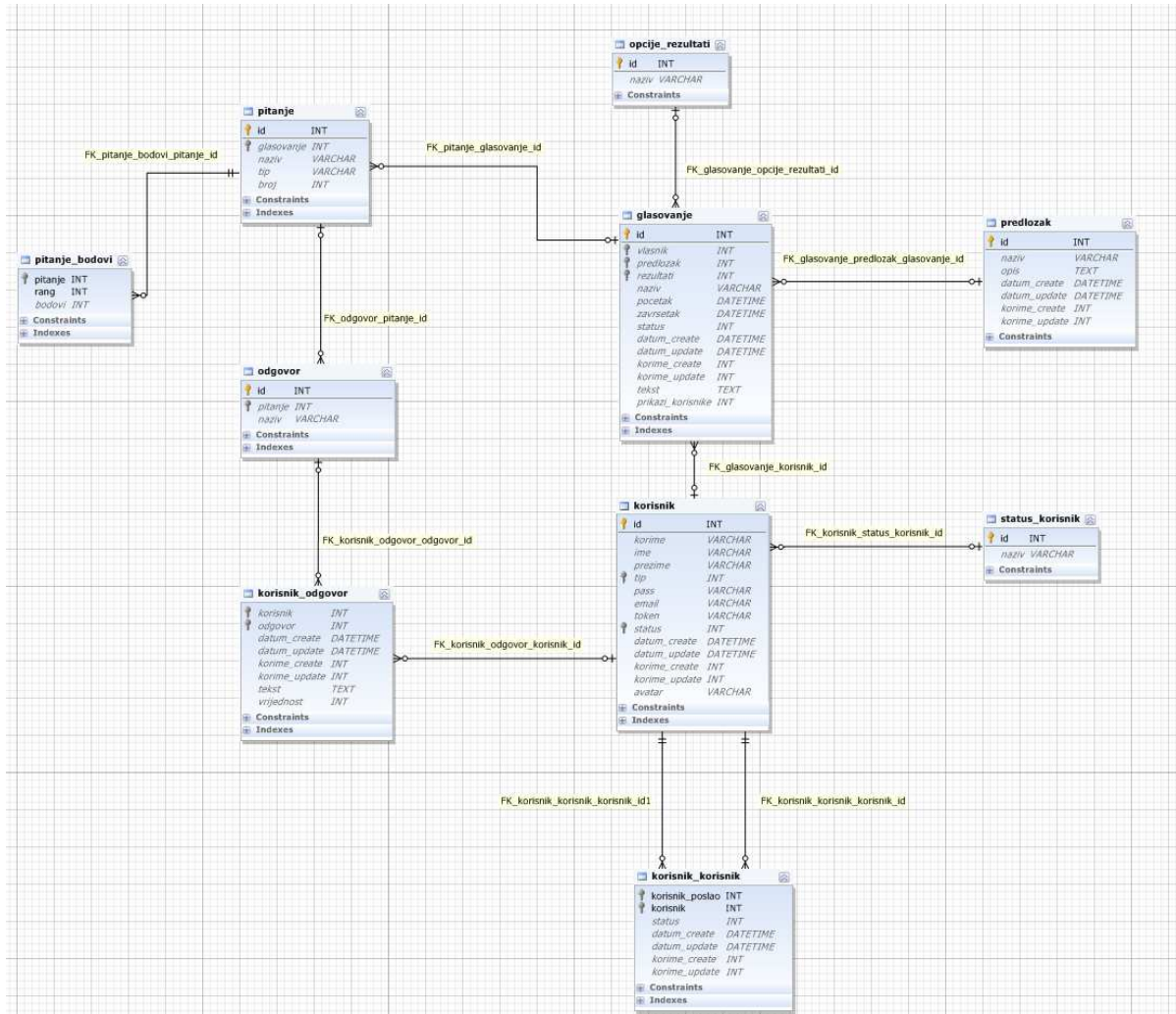
Navigacijski dijagram ove aplikacije se nalazi na slici 5.2. i ovdje se može primjetiti da aplikacija ima vertikalnu navigaciju. Slika prikazuje kako se se prijava u sustav i registracija nalaze na početnoj stranici, a sve ostale poveznice se nalaze na glavnoj navigaciji koja je podjeljena na dio za običnog korisnika, dio za korisnika-admina i dio za admina i to na način da svaki korisnik koji je viši u hijerarhiji korisnika ima izbornik i od nižih korisnika u hijerarhiji. Slika navigacijskog dijagrama još prikazuje koje sve poveznice postoje u aplikaciji i kako se do njih dolazi.



Slika 5.2. Navigacijski dijagram aplikacije

## 6. Tehnički dio aplikacije i kôd

### 6.1. ERA Dijagram



Slika 6.1. ERA dijagram baze koja je korištena u aplikaciji

Najvažnija tablica u aplikaciji je *glasovanje*. Ona ima tri vanjska ključa, *vlasnik* koji predstavlja vlasnika glasovanja, tj. id korisnika koji je kreirao glasovanje, zatim *predlozak* koji predstavlja načina kako će se glasovati i *rezultati* koji predstavlja kome će se smjeti prikazati rezultati glasovanja. U tablici *glasovanje* još imamo atribut za naziv glasovanja te attribute za početak i završetak glasovanja, za status i za način prikazivanja korisnika. Atributi *datum\_create*, *datum\_update*, *korime\_create*, *korime\_update* se nalaze u više tablica i oni

predstavljaju datum kada je koji zapis u bazi kreiran ili ažuriran te id korisnika koji ga kreirao ili ažurirao. Moram napomenuti da se ovi atributi ne zapisuju u svim tablicama negu su tu samo kao moguća buduća proširenja.

Tablica *korisnik* ima minimalan broj atributa koji se vode o korisniku, dok tablica *korisnik\_korisnik* predstavlja vezu gdje se zapisuju prijateljstva korisnika, a tablica *status\_korisnika* trenutni status korisnika, koji može biti jedan od tri spomenutih u prošlim poglavljima.

Glasovanje može imati neograničen broj pitanja koja se zapisuju u tablicu *pitanje*. Atribut *broj* u tablici pitanje označava broj dozvoljenih odgovora, atribut *tip* označuje jedan od tri mogućih tipova glasovanja dok je atribut *glasovanje* vanjski ključ na tablicu glasovanje.

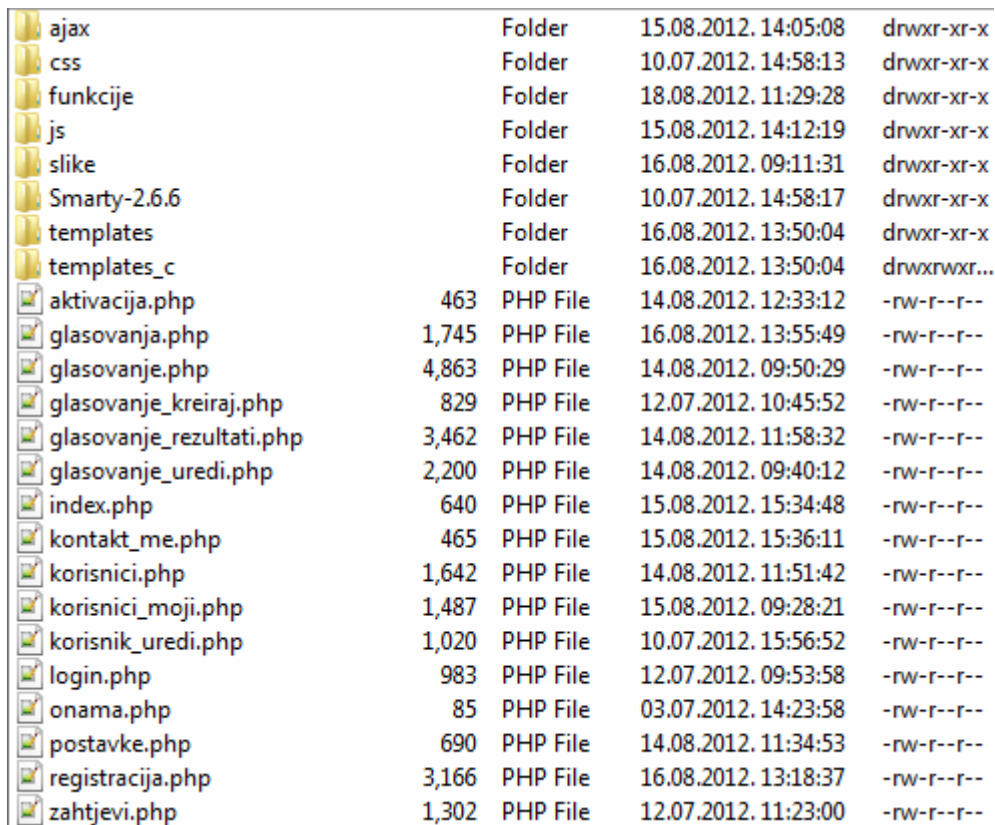
Glasovanje također može imati neograničen broj odgovora na pojedino pitanje koji se zapisuju u tablicu *odgovor* i tu se zapisuju samo vanjski ključ na glasovanje i naziv odgovora.

*Korisnik\_odgovor* je tablica gdje se spremaju odgovori korisnika, za svaku vrstu pitanja drugačije, ali taj dio je detaljnije objašnjen u dijelu „realizacija aplikacije“.

*Pitanje\_bodovi* je tablica koja služi samo za natjecateljska pitanja. Jer natjecateljska pitanja imaju različite bodove za različiti rang pa se ti rangovi i njihovi bodovi spremaju u tu tablicu.

## 6.2. Struktura web mjesta i izvršavanje aplikacije

U početnom direktoriju aplikacije se nalaze php skripte za svaki link na aplikaciji, npr. *aktivacija.php* se izvršava kada kliknemo na link za aktivaciju koji dobijemo u e-mailu, *glasovanje\_rezultati.php* se izvršava kada kliknemo na rezultate za pojedino glasovanje. U mapi ajax se nalaze skripte koje se izvršavaju samo preko AJAX zahtjeva. css i js mape su za CSS i JavaScript. Mapa funkcije služi za PHP skripte u kojim se nalaze različite funkcije za rad s aplikacijom i druge txt ili xml datoteke koje služe kao log ili datoteke za postavke. *Smarty-2.6.6* je mapa u kojem se nalaze skripte od Smarty dodatka, a mape *templates* i *templates\_c* su mape koje Smarty koristi svoj za rad.



ajax	Folder	15.08.2012. 14:05:08	drwxr-xr-x
css	Folder	10.07.2012. 14:58:13	drwxr-xr-x
funkcije	Folder	18.08.2012. 11:29:28	drwxr-xr-x
js	Folder	15.08.2012. 14:12:19	drwxr-xr-x
slike	Folder	16.08.2012. 09:11:31	drwxr-xr-x
Smarty-2.6.6	Folder	10.07.2012. 14:58:17	drwxr-xr-x
templates	Folder	16.08.2012. 13:50:04	drwxr-xr-x
templates_c	Folder	16.08.2012. 13:50:04	drwxrwxr...
aktivacija.php	463 PHP File	14.08.2012. 12:33:12	-rw-r--r--
glasovanja.php	1,745 PHP File	16.08.2012. 13:55:49	-rw-r--r--
glasovanje.php	4,863 PHP File	14.08.2012. 09:50:29	-rw-r--r--
glasovanje_kreiraj.php	829 PHP File	12.07.2012. 10:45:52	-rw-r--r--
glasovanje_rezultati.php	3,462 PHP File	14.08.2012. 11:58:32	-rw-r--r--
glasovanje_uredi.php	2,200 PHP File	14.08.2012. 09:40:12	-rw-r--r--
index.php	640 PHP File	15.08.2012. 15:34:48	-rw-r--r--
kontakt_me.php	465 PHP File	15.08.2012. 15:36:11	-rw-r--r--
korisnici.php	1,642 PHP File	14.08.2012. 11:51:42	-rw-r--r--
korisnici_moji.php	1,487 PHP File	15.08.2012. 09:28:21	-rw-r--r--
korisnik_uredi.php	1,020 PHP File	10.07.2012. 15:56:52	-rw-r--r--
login.php	983 PHP File	12.07.2012. 09:53:58	-rw-r--r--
onama.php	85 PHP File	03.07.2012. 14:23:58	-rw-r--r--
postavke.php	690 PHP File	14.08.2012. 11:34:53	-rw-r--r--
registracija.php	3,166 PHP File	16.08.2012. 13:18:37	-rw-r--r--
zahtjevi.php	1,302 PHP File	12.07.2012. 11:23:00	-rw-r--r--

Slika 6.2. Prikaz mapa i datoteka u početnoj mapi aplikacije



Na početku svake skripte koja se poziva se uključuje skripta *pocetno.php* u kojoj se započinje sesija, uključuje smarty datoteka i kreira instanca smarty klase te se još u toj skripti aplikacija spaja na bazu funkcijom *bazaSpoji()*. U skripti *pocetno.php* se nalaze još neke funkcije koje sam koristio na više mjesta kroz aplikaciju, dio kôda za logiranje i odlogiranje te kôd za zapisivanje u log. Sljedeći isječak koda prikazuje dio skripte *pocetno.php*:

```
session_start();
$smarty_file = 'Smarty-3.1.11/libs/Smarty.class.php';
include_once($smarty_file);
$smarty = new Smarty();
function bazaSpoji(){
    mysql_connect('localhost', 'root', '') or die(mysql_error());
    mysql_select_db('evoting') or die(mysql_error());
    mysql_query("set names utf8") or die(mysql_error());
}
bazaSpoji();
```

### 6.3. Uređivanje glasovanja

Kod uređivanja glasovanja podaci o glasovanju se mijenjaju na način da se post metodom preko forme za uređivanje glasovanja šalju podaci, te se oni nakon toga pomoću PHP-a ažuriraju u bazi. Izgled forme za promjenu podataka je prikazan u nastavku:

```
<form class="form-horizontal" action="" method="post" enctype="multipart/form-data">
    <input type="text" name="naziv" value="Glasovanja kapetana svih momčadi prve HNL za najboljeg igrača lige" class="span6"></input>
    <input type="text" id="naziv" name="pocetak" value="2012-08-10 10:00" size="40" class="datepicker_mysql"></input>
    <input type="text" id="zavrsetak" name="zavrsetak" value="2012-09-22 20:00" size="40" class="datepicker_mysql"></input>
    <select name="status" id="status">
        <option value="1" selected="selected" id="status-0">Aktivno</option>
        <option value="2" id="status-1">Pasivno</option>
        <option value="3" id="status-2">Završeno</option>
        <option value="9" id="status-3">Obrisano</option>
    </select>
    <select name="predlozak" id="predlozak">
        <option value="1" selected="selected" id="predlozak-0">Glasuju svi korisnici aplikacije</option>
        <option value="2" id="predlozak-1">Glasuju samo prijatelji korisnika</option>
```

```

</select>
<select name="rezultat" id="rezultat">
    <option value="1" selected="selected" id="rezultat-0">Prikaži svima</option>
    <option value="2" id="rezultat-1">Prikaži sudionicima</option>
    <option value="3" id="rezultat-2">Prikaži samo vlasniku</option>
</select>
<select name="prikazi_korisnike" id="prikazi_korisnike">
    <option name="ne" value="1" >ne</option>
    <option name="da" value="2" selected="selected" >da</option>
</select>
<input type="submit" value="Ažuriraj" class="btn btn-primary"></input>
</form>

```

Dio PHP skripte koja služi za primanje podataka iz forme za uređivanje glasovanja i ažuriranje u bazi je prikazana u nastavku:

```

if(isset($_POST['naziv'])){
    $naziv = $_POST['naziv'];
    $predlozak = $_POST['predlozak'];
    $pocetak = $_POST['pocetak'];
    $zavrsetak = $_POST['zavrsetak'];
    $status = $_POST['status'];
    $rezultat = $_POST['rezultat'];
    $prikazi_korisnike = $_POST['prikazi_korisnike'];

    $sql = "UPDATE glasovanje SET naziv='$naziv', predlozak=$predlozak, pocetak='$pocetak',
        zavrsetak='$zavrsetak', status=$status, rezultati=$rezultat,
        prikazi_korisnike=$prikazi_korisnike
        WHERE id=$id";
    $r = mysql_query($sql);
    $smarty->assign('info', "Uspješno ažurirano. Vrijeme: " . date("Y-m-d H:i:s"));
}

```

U kôdu se na početku provjerava da li je postavljena varijabla `$_POST['naziv']`), a ona je postavljena samo u slučaju kada su poslani podaci iz forme pa se ovaj kôd izvršava samo tada. U nastavku if petlje se uzimaju svi poslani podaci iz forme te se sql upitom ažuriraju podaci u bazi i na kraju se postavlja smarty varijabla `$info` sa prikladnom porukom da bi se prikazala iznad forme nakon ažuriranja.

Dodavanje pitanja i odgovora te mijenjanje svih njihovih opcije se ostvaruje pomoću javascripta, jquery-a i AJAX-a. Uzmimo za primjer dodavanje pitanja. HTML forma izgleda je prikazana na sljedećem dijelu:

```
<form class="form-horizontal" action="" method="post" >
  <div class="hero-unitt well" style="background-color: transparent;">
    <h1>
      </img>
      Dodavanje pitanja
    </h1>
  </div>

  <select name="tip_pitanja" id="tip_pitanja">
    <option name="1" value="1">odabir(1 ili više)</option>
    <option name="2" value="2">natjecateljski odabir</option>
    <option name="3" value="3">tekst</option>
  </select>
  <textarea id="pitanje" class="pitanje" name="pitanje" cols="60" rows="5"></textarea>
  <input id="dodaj_pitanje" type="submit" value="Dodaj pitanje" class="btn" ></input>
</form>
```

Prethodna forma sadrži podatke o tipu pitanja i tekstu pitanja i oni se pomoću javascripta uzimaju te preko AJAX zahtjeva šalju na server, nakon što klikne na submit gumb u ovoj formi. JavaScript i JQuery kôd u kojem je *click* događaj vezan na taj slijedi:

```
$("#dodaj_pitanje").live('click', function(){
  var pitanje = $(".pitanje").val();
  var id_glasovanje = $(".id_glasovanje").attr("value");
  var tip_pitanja = $("#tip_pitanja").val();

  var url = 'ajax/dodajPitanje.php?glasovanje=' + id_glasovanje + '&naziv=' + pitanje +
  '&tip=' + tip_pitanja;
  $.get(url, function(html){
    $(".pitanje").val("");
    osvjeziPitanja();
  });
  return false;
});
```

U ovom prethodnom dijelu kôda se uzimaju naziv pitanja, vrsta pitanja i id glasovanja te se šalju u get parametrima PHP skripti `addajPitanje.php` koja dodaje pitanje u bazi. Nakon toga se poziva funkcija JavaScript `osvjeziPitanja()` koja osvježuje podatke pitanja asinkrono pomoću AJAX-a. Cijela skripta `addajPitanje.php` slijedi:

```

<?php
    require_once("../funkcije/pocetno.php");

    $korisnik = $_SESSION['korisnik']['id'];
    $glasovanje = $_GET['glasovanje'];
    $naziv = $_GET['naziv'];
    $tip = $_GET['tip'];

    if( $glasovanje != null && (isset($_SESSION['korisnik']))){
        if(($_SESSION['korisnik']['tip'] < 3) && ($_SESSION['korisnik']['id'] !=
$glasovanje['vlasnik'])){
            echo "Neutoriziran pristup preko ajaxa!";
            exit;
        }
    }

    $sql = "INSERT INTO pitanje(glasovanje, naziv, broj, tip)
        VALUES($glasovanje, '$naziv', 1, $tip)";
    mysql_query($sql);
    $id = mysql_insert_id();

    if($tip == 2){
        $sql2 = "INSERT INTO pitanje_bodovi(pitanje, rang, bodovi)
            VALUES($id, 1, 1)";
        mysql_query($sql2);
    }
    if($tip == 3){
        $sql3 = "INSERT INTO odgovor(pitanje)
            VALUES($id)";
        mysql_query($sql3);
    }

    echo "<?xml version='1.0'?><poruka><alert>Uspješno!</alert></poruka>";
?>

```

U početku skripte se uzimaju podaci iz get parametara, nakog toga provjerava da li korisnik ima pravo dodavati pitanje te se zatim pitanje sql upitom unosi u bazu. Na kraju je još dio kôda koji služi za unos dodatnih vrijednosti u bazu ako je pitanje tekstualno ili natjecateljsko.

## 6.4. Pregled rezultata

Kod pregleda rezultata prvo se iz baze uzimaju svi podaci o glasovanju i oni će biti prikazani na početku rezultata. PHP kôd za uzimanje glasovanja iz baze slijedi:

```
$sql = "SELECT g.*, p.naziv as predlozak_naziv, s.naziv as status, p.id as predlozak,
k.korime, opcije_rezultati.naziv as rezultati_naziv
FROM glasovanje g, predlozak p, status s, korisnik k, opcije_rezultati
WHERE g.status=s.id
AND g.predlozak=p.id
AND k.id=g.vlasnik
AND opcije_rezultati.id=g.rezultati
AND g.id=$id";
$r = mysql_query($sql);
$glasovanje = mysql_fetch_array($r);
```

Nakon podataka o glasovanju potrebno je iz baze uzeti pitanja, odgovore na pitanja i korisnikove odgovore. U sljedećem dijelu kôda možemo vidjeti upit u kojem se iz baze uzimaju sva pitanja kod gledanog glasovanja. U svakom koraku *while* petlje se zatim uzima jedno pitanje glasovanja i onda unutar te petlje postoji druga *while* petlja za svaki odgovor tog pitanja. U toj drugoj petlji postoje više *if* uvjeta jer se za svaku vrstu pitanja drugačije računaju rezultati. U sljedećem dijelu kôda to je prikazano, ali je uzet samo dio gdje se računaju rezultati za pitanja sa odabirom. Taj dio se nalazi u *if* petlji gdje se provjerava da li je varijabla *\$row['tip']* jednaka 1. Za druge vrste pitanja, natjecateljsko pitanje i pitanje s odabirom, postoje još dvije *if* petlje, ali zbog jednostavnosti je ovdje ostavljena samo ova.

```
$vlasnik = $_SESSION['korisnik']['id'];
$pitanja = array();
$sql = "SELECT * FROM pitanje WHERE glasovanje=$id";
$r = mysql_query($sql);
while($row = mysql_fetch_array($r)){
    $pitanje_id = $row['id'];

    $sql2 = "SELECT *FROM odgovor WHERE pitanje=" . $row['id'];
    $r2 = mysql_query($sql2);

    while($row2 = mysql_fetch_array($r2)){
        $odgovor_id = $row2['id'];
        $sql3 = "SELECT * FROM korisnik_odgovor WHERE odgovor=$row2[id]";
        $r3 = mysql_query($sql3);

        if($row['tip'] == 1){
```

```

        $broj = mysql_num_rows($r3);
        $row2['broj'] = $broj;
        $broj_glasova = mysql_num_rows(mysql_query("SELECT * FROM odgovor
o,korisnik_odgovor ko
        WHERE o.id=ko.odgovor AND o.pitanje=" . $row['id']));
        if($broj_glasova != 0)
            $row2['postotak'] = round(($broj / $broj_glasova), 2) * 100;
        else
            $row2['postotak'] = '0';
    }
    $row['odgovori'][] = $row2;
}
$pitavanja[] = $row;
}

```

Nakon što se pitanja sa odgovorima dohvate iz baze i izračunaju rezultati, pitanje se spremaju u varijablu *\$pitavanja* i šalju u *glasovanje\_rezultati.tpl* smarty datoteku. U toj se datoteci također izvršavaju dvije petlje, kao i u PHP skripti, prvo jedna za sva pitanja te onda druga unutar nje za sve odgovore i tamo postoje if uvjeti za svaku vrstu pitanje jer se odgovori za svaku vrstu pitanja drugačije interpretiraju i prikazuju. Sljedeći isječak kôda je dio unutar smarty-ove petlje koja se izvršava za sve odgovore pojedinog pitanja i tu je prikazan način na koji se prikazuju odgovori kod tekstualnog oblika pitanja:

```

{if $pitavanja[i].tip==3}
    <li>
    <li><span class="label label-info" style="font-size: 14px;">tekst</span> <br/>
        <div class="pitanje well">{$pitavanja[i].naziv}</div>
    </li>
    <ul class="odgovori odgovor odgovori_komentar">
        {section name = j loop=$pitavanja[i].odgovori_tekstualni}
        <li>
            <div class="alert alert-info">
                {$pitavanja[i].odgovori_tekstualni[j]}
            </div>
        </li>
        {/section}
    </ul>
{/if}

```

## 6.5. Glasovanje

Na sljedećem isječku kôda je prikazana tpl datoteka gdje se prikazuje forma za odgovor na pitanja. Isto kao i kod kreiranja i uređivanja glasovanja i ovdje se izvršavaju dvije smarty for petlje, prva za pitanja i druga unutar nje za odgovore na ta pitanja. U petlji za odgovore imamo if uvjete za svaki tip pitanja. Na sljedećem isječku kôda je ostavljen i prikazan samo if uvjet za natjecateljska pitanja. Možemo primijetiti da se ova vrsta pitanja odgovaraju u obliku

padajućeg izbornika, tj. select inputa. Svaki select se jednoznačno označuje na sa id pitanja i rangom odgovora.

```

<div id="pitanja" class="well">
  <form action="" method="post" class="form-horizontal pitanja_glasanje">
    <ul>
      <hr class="hr2"/>
      {section name = i loop=$pitanja}
      <li>
      <li>
        <span class="label label-info" style="font-size: 14px;">
          {if $pitanja[i].tip==1}odabir{elseif
          $pitanja[i].tip==2}natjecateljsko{else}tekst{/if}</span> <br/>
          <span class="pitanje well" broj="{ $pitanja[i].broj}">
            { $pitanja[i].naziv}
            {if $pitanja[i].tip!=3}
<div style="margin-top: 20px; font-size: 14px;">( broj odgovora:{ $pitanja[i].broj} )</div>
            {/if}
          </span>
        </li>
      <ul class="odgovori">
        {if $pitanja[i].tip==2}
          {foreach from=$pitanja[i].bodovi key=k item=v}
          <li>
            <div class="alert alert-info odgovor">
              Za: { $v.bodovi} bodova
              <select name="natjecateljski_{ $pitanja[i].id}_{ $v.rang}" class="span5">
                {foreach from=$pitanja[i].izbori key=k2 item=v2}
                  <option value="{ $v2.id}"
                    {if $v.vrijednost==$v2.id} selected="selected"{/if}
                  >{ $v2.naziv}</option>
                {/foreach}
              </select>
            </div>
          </li>
          {/foreach}
        {/if}
      </ul>
      <hr class="hr2"/>
      {/section}
    </ul>
    <div class="form-actions" style="background-color: transparent;">
      <input type="submit" name="odgovori"
        value="{if $glasovao=='ne'}Odgovori{else}Promijeni odgovor{/if}"
        class="btn-large btn-primary glasuj_button" id="odgovori"
        id_pitanje="{ $pitanja[i].id}"></input>
      </div>
      <input type="text" name="glasaj" value="da" style="display: none;"></input>
    </form>
  </div>

```

Sljedeći isječak kôda je dio PHP skripte u kojem se spremaju odgovori. Možemo primijetiti da se ovaj dio kôda izvršava tek kad je postavljeno `$_POST['glasaj']` varijabla, a to je samo onda kada je korisnik kliknuo glasaj i poslao obrazac i kada je varijabla `$validacija` postavljen na `TRUE`, a to je samo onda kada je korisnik ispravno glasovao. Kôd kojim se provjerava ispravnost odgovora, tj. validacija odgovora, se izvršava prije ovog isječka i nije prikazan ovdje. U ovom dijelu kôda je ostavljen samo dio za natjecateljska pitanja. Za svako natjecateljsko pitanje se prvo brišu svi odgovori iz baze, ako je korisnik već prije glasovao te se u bazu spremaju oni odgovori koji su dobiveni iz obrasca.

```

if(isset($_POST['glasaj']) && ($validacija == TRUE)){
    $vlasnik = $_SESSION['korisnik']['id'];
    $sql = "SELECT * FROM pitanje WHERE glasovanje=$id";
    $r = mysql_query($sql);
    while($row = mysql_fetch_array($r)){
        $pitanje_id = $row['id'];
        $sql2 = "SELECT *FROM odgovor WHERE pitanje=" . $row['id'];
        $r2 = mysql_query($sql2);
        if($row['tip']==2){
            while($row2 = mysql_fetch_array($r2)){
                $odgovor_id = $row2['id'];
                $sql3 = "DELETE FROM korisnik_odgovor WHERE odgovor=$odgovor_id AND korisnik=$vlasnik";
                $r3 = mysql_query($sql3);
            }
        }
    }
    for($i=1; $i <= $broj_rangova; $i++){
        $odgovor = $_POST['natjecateljski_' . $pitanje_id . '_' . $i];
        $vrijeme = date("Y-m-d H:i:s");
        $sql3 = "INSERT INTO korisnik_odgovor(korisnik, odgovor, vrijednost, datum_create)
            VALUES($vlasnik, $odgovor, $i, '$vrijeme')";
        $r3 = mysql_query($sql3);
    }
}
header("Location: glasovanje_rezultati.php?id=" . $id . "&status=1");
}

```

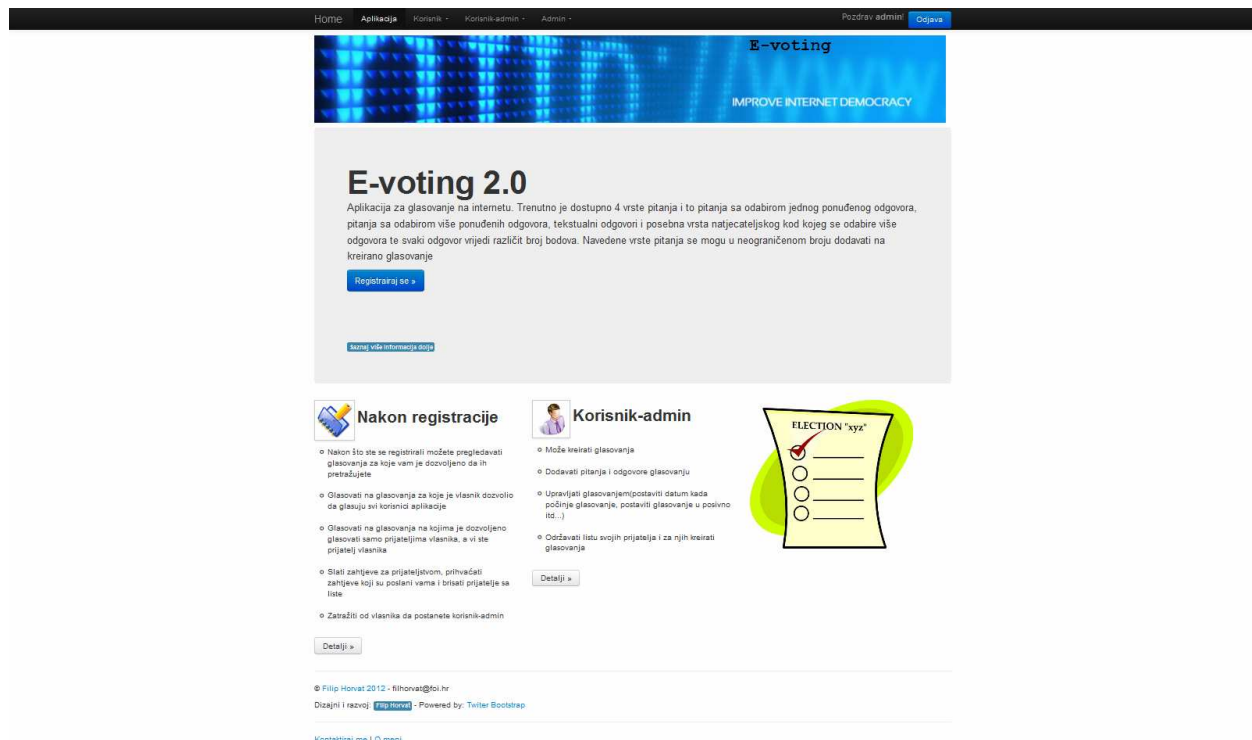


## 7. Korištenje aplikacije

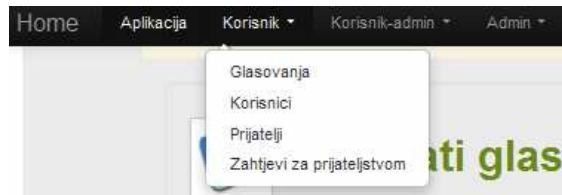
U dijelu „korištenje aplikacije“ je opisan rad aplikacije. Primjer glasovanja koji se pojavljuje kroz cijeli ovaj dio je glasovanje za najboljeg nogometaša i to tako da postoji osam izbora te prvi izbor vrijedi 5 bodova, drugi tri boda, a treći 1 bod.

### 7.1. Početna stranica

Aplikacija je zamišljena tako da ima dva dizajna. Početna stranica i demonstracije imaju jedan dizajn, dok ostali dio ima drugi. Na početnoj stranici aplikacije su prikazane neke osnovne informacije i ovdje se korisnik može prijaviti u sustav i registrirati. Slika 7.1. prikazuje početnu stranicu aplikacije, dok slika 7.2. prikazuje izbornik aplikacije nakon logiranja. Ostali izgled aplikacije bit će prikazan u sljedećim potpoglavljima.



Slika 7.1. Početna stranica aplikacije



Slika 7.2. Izbornik u aplikaciji

## 7.2. Pregled glasovanja

Krenimo sada sa djelom aplikacije gdje korisnik može pregledavati glasovanja, vidjeti informacije o glasovanjima i odabrati jedan od tri poveznice pokraj glasovanja. Te tri poveznice su da korisnik glasuje, da uređuje glasovanje i da pregledava rezultate. Korisnik može pregledavati glasovanja na dva načina. Prvi je da pregledava sva glasovanja koja postoje i da samo pregledava svoja glasovanja, a to može samo u slučaju da je korisnik-admin.

Kod pregleda svih glasovanja link za uređivanje će biti prikazan samo u slučaju da je korisnik admin, a svoja glasovanja može uređivati kod ispisa njegovih glasovanja. Napomenimo ovdje da se glasovanja sa statusom u izradi mogu vidjeti samo kod ispisa korisnikovih glasovanja. Straničenje (eng. paginacija) i pretraga glasovanje je napravljena pomoću AJAX-a. Glasovanja se mogu pretraživati po statusu i po vlasniku s time da kad se pretražuje po vlasniku postoji jquery plugin(autocomplete) koji predlaže postojeće korisnike-adminine.

Rezultati pretrage se automatski nakon mijenjanja vlasnika ili statusa glasovanja. Glasovanja se mogu sortirati po svim navedenim informacijama, silazno i uzlazno.

Gumbić „*moja glasovanja*“ služi da se u input za pretragu vlasnika upiše korisničko ime logiranog korisnika, a samim time da se pregledavaju njegova glasovanja.

Kada glasovanja nema status aktivno, link za glasovanje je onemogućen i izblijeđen, a kada smo već glasovali onda ima kvačicu preko linka kao što možemo zamijetiti kod trećeg glasovanja na slici 7.3. Slika 7.3. ujedno prikazuje ispis svih glasovanja koja postoje dok slika 7.4. prikazuje samo glasovanja logiranog korisnika. Na oba dvije prije spomenute slike je kod izbornika označeno kako se dođe na ovaj dio aplikacije.

Home Aplikacija Korisnik - Korisnik-admin - Admin - Pozdrav zvojurini! [Odjava](#)

Glasovanja  
Korisnici  
Prijatelji  
Zahtjevi za prijateljstvom

Projavi

## Glasovanja

- Glasovanja se mogu pretraživati po statusu i po vlasniku glasovanja.
- Glasovanja se mogu sortirati po svim prikazanim vrijednostima (klikom na vrijednost u zaglavlju tablice).
- Aktivna glasovanja ne znači da još traju nego da im je status aktivan, obično glasovanja imaju aktivan status nekoliko dana prije početka i nekoliko dana poslije završetka.

Status:

Vlasnik:  [Moja glasovanja](#)

Naziv	Vlasnik	Tip	Pocetak	Zavrsetak	Status
aa	admin	Glasuju svi korisnici aplikacije	2012-09-29 00:00:00	2012-09-30 00:00:00	Aktivno
Anketa o ovoj stranici	admin	Glasuju svi korisnici aplikacije	2012-08-01 00:00:00	2012-09-30 00:00:00	Aktivno
Glasovanja kapetana svih momčadi prve HNL za najboljeg igrača lige	admin	Glasuju svi korisnici aplikacije	2012-08-10 10:00:00	2012-09-22 23:22:00	Aktivno
Glasovanje 1	filhorvat	Glasuju svi korisnici aplikacije	2012-08-09 10:00:00	2012-08-31 16:00:00	Aktivno

Slika 7.3. Ispis svih glasovanja u aplikaciji

Home Aplikacija Korisnik - Korisnik-admin - Admin - Pozdrav zvojurini! [Odjava](#)

Kreiraj glasovanje  
[Moja glasovanja](#)

Prijavljeni ste kao: zvojurini

## Moja glasovanja

- Glasovanja se mogu pretraživati po statusu i po vlasniku glasovanja.
- Glasovanja se mogu sortirati po svim prikazanim vrijednostima glasovanja.

Status:

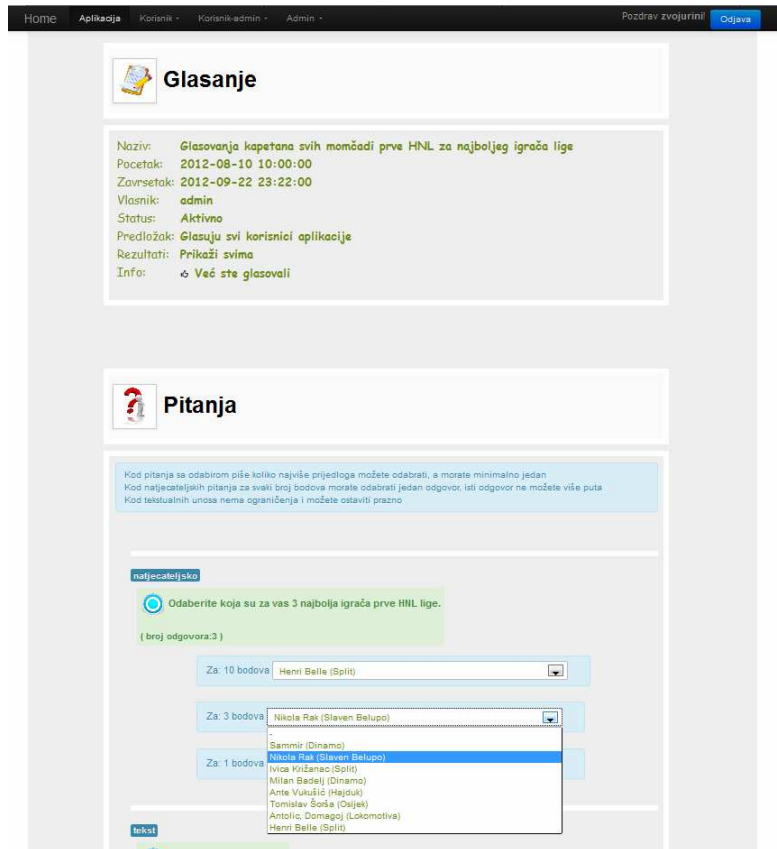
Naziv	Vlasnik	Tip	Pocetak	Zavrsetak	Status
Glasovanje za EU	zvojurini	Glasuju samo prijatelji korisnika	2012-08-01 00:00:00	2012-08-24 15:00:00	Završeno
referendum za pejaški most	zvojurini	Glasuju samo prijatelji korisnika	2012-08-04 11:00:00	2012-09-01 10:00:00	Aktivno

[Prev](#) [1](#) [Next](#)

Slika 7.4. Ispis glasovanja logiranog korisnika u aplikaciji

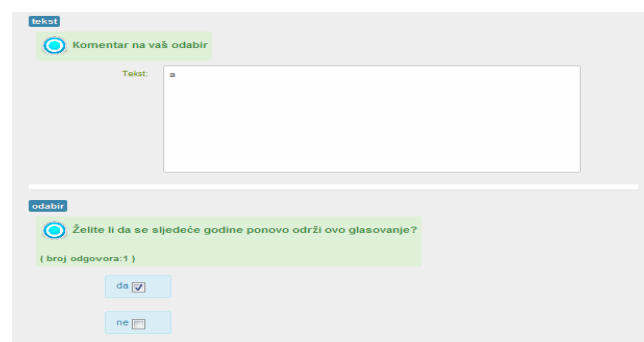
### 7.3. Glasovanje

Slika 7.5. prikazuje izgled glasovanje kada otvorimo link za glasovanje. Prvo pitanje spomenutoj slici prikazuje natjecateljsko glasovanje. U ovom slučaju pitanje ima tri odabira i glasovanje na ovo pitanje će biti uspješno tek kad odaberemo odgovore za svaki broj bodova.



Slika 7.5. Prikaz postupka glasovanja u aplikaciji

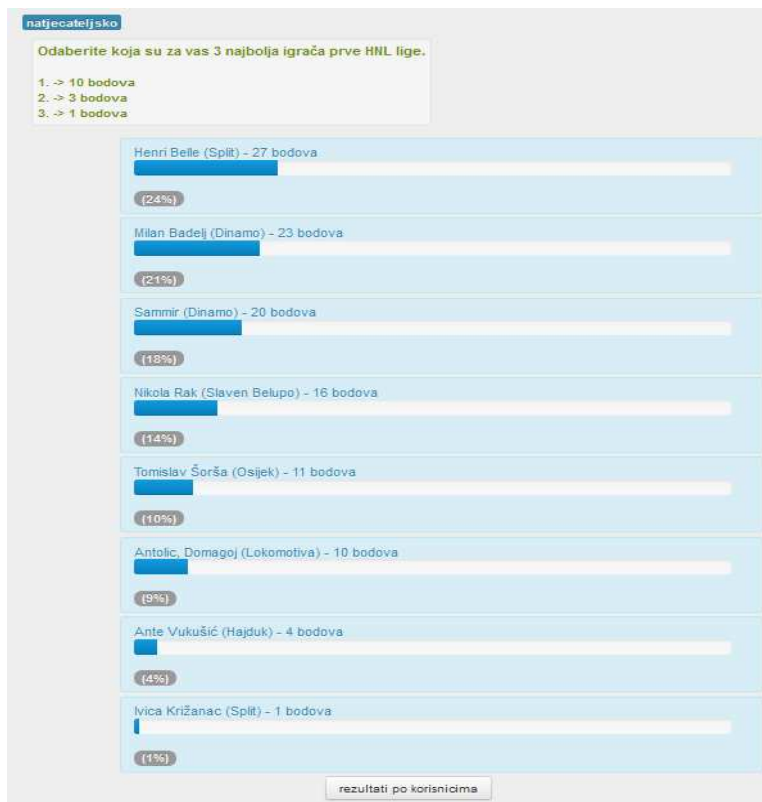
Slika 7.6. prikazuje pitanja sa komentarom i običnim odabirom:



Slika 7.6. Prikaz pitanja sa komentarom i pitanja sa odabirom

## 7.4. Rezultati

Na slici 7.7. je prikazana izgled aplikacije kada pregledavamo rezultate glasovanje za natjecateljska pitanja:



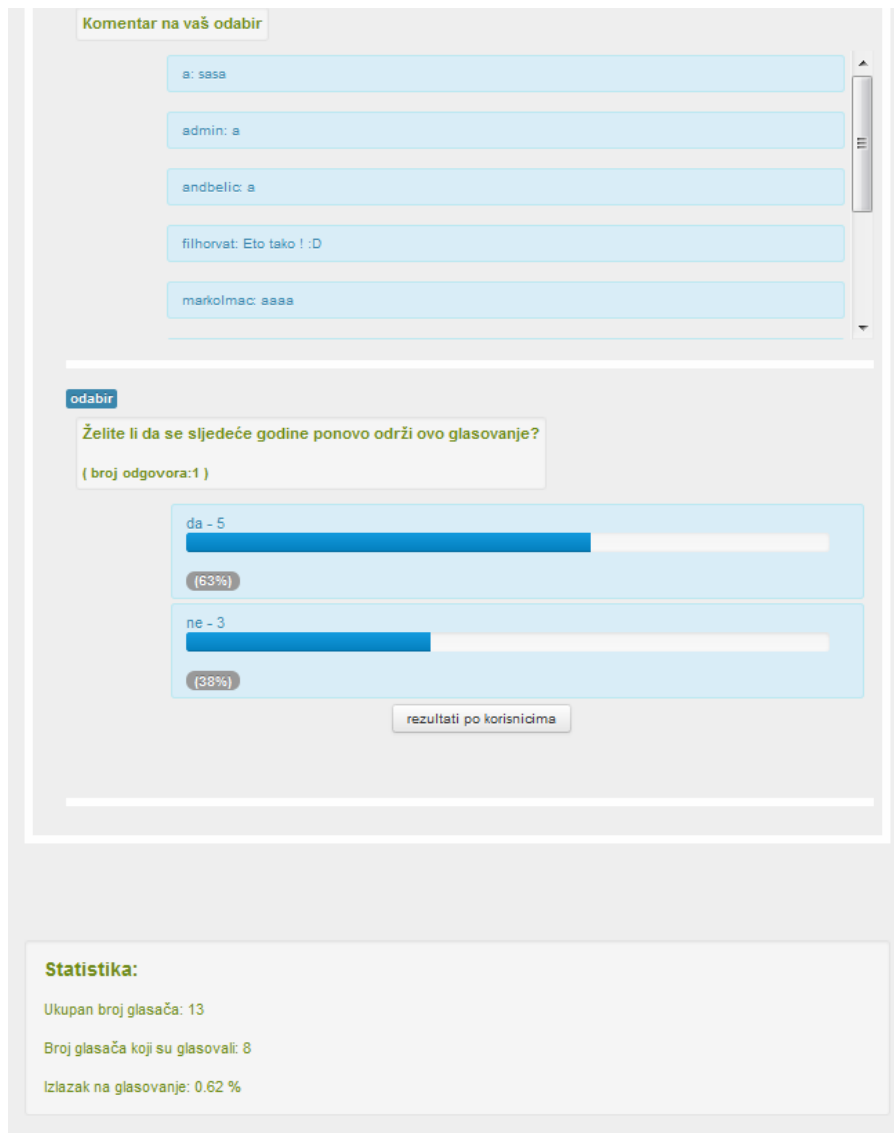
Slika 7.7. Prikaz rezultata kod natjecateljskih pitanja

Kod pregleda rezultata imamo opciju rezultati po korisnicima i to je dostupno samo ako je tako određeno u postavkama glasovanja. Kada kliknemo na gumbić „*rezultati po korisnicima*“ AJAX upitom se dohvaćaju svi korisnici koji su glasali i njihovi odabiri te se ti podaci se smještaju ispod rezultata, slika 7.8.



Slika 7.8. Prikaz rezultata po korisniku

Slika 7.9. prikazuje kako izgledaju rezultati za druge dvije vrste odgovora. Komentari se prikazuju sortirano po korisničkom imenu uzlazno u div s ograničenom širinom i ako ih je odgovore previše pojavljuje se vertikalni scroll. Na kraju glasovanje prikazuje se statistika koja uključuje broj potencijalnih glasača, broj onih koji su glasovali i postotak onih koji su glasovali.



Slika 7.9. Prikaz rezultata kod pitanja sa komentarom i pitanja sa odabirom

## 7.5. Uređivanje glasovanja

Slika 7.12. prikazuje izgled forme kod uređivanja glasovanja. Kada mijenjamo glasovanja i pritisnemo ažuriraj to se izvršava bez AJAX-a, sa slanjem forme na server. Nakon što ažuriramo glasovanje iznad će se ispisati poruka. Slika 7.10. prikazuje formu za dodavanje pitanja. Nakon što odaberemo vrstu pitanja i napišemo tekst pitanja kliknemo dodaj pitanje, putem AJAX-a se pitanje dodaje na trenutno glasovanje i forma sa pitanjima na trenutnom glasovanju se osvježi.

Home Aplikacija Korisnik - Korisnik-admin - Admin - Pozdrav zvojurini! Odjava

### Uređivanje glasovanja

- Postavite podatke o glasovanju na željeni način i kliknite "Ažuriraj".
- Nakon završene akcije prikazat će vam se obavijest s vremenom ažuriranja.

Naziv: referendum za peješki most

Pocetak: 2012-08-04 11:00

Zavrsetak: 2012-09-01 10:00

Predložak: Glasuju samo prijatelji korisnika

Rezultati: Prikaži samo vlasniku

Prikaži korisnike: da

Status: Aktivno

Ažuriraj

### + Dodavanje pitanja

- Odaberite vrstu pitanja koju želite dodati i tekst pitanja i kliknite "Dodaj pitanje".
- Nakon dodavanja pitanje će se pojaviti kod uređivanja pitanja.

Tip pitanja: odabir (1 ili više)

Pitanje: natjecateljski odabir

tekst

Dodaj pitanje

Slika 7.10. Prikaz uređivanja glasovanja

Uređivanje natjecateljskog pitanja je prikazano na slici 7.11. Unos broj odgovora nakon naziva pitanja određuje koliko odgovora želimo da pitanje ima i kada ga mijenjamo automatski se AJAX upitom on povećava u bazi te se niže bodovi osvježe da budu za broj pitanja koliko želimo. Kada želimo maknuti neki odgovor kliknemo na iksić uz njega i on se isto AJAX upitom izbriše iz baze i odgovori se odmah osvježe.

Ako želimo dodati odgovor upišemo u tekstualni unos željeni odgovor i kliknemo „*dodaj odgovor*“. Nakon liste sad odgovorima još imamo bodove i tu se mijenjaju željeni bodovi za prvo, drugo itd. mjesto. Kada se broj bodova povećeva ili smanjuje, izvršava se AJAX upit i trenutno se bodovi povećavaju u bazi.

natjecateljsko

Odaberite koja su za vas 3 najbolja igrača prve HNL lige. broj odgovora 3

- Sammir (Dinamo)
- Nikola Rak (Slaven Belupo)
- Ivica Križanac (Split)
- Milan Badelj (Dinamo)
- Ante Vukušić (Hajduk)
- Tomislav Šorša (Osijek)
- Antolic, Domagoj (Lokomotiva)
- Henri Belle (Split)

Odgovor:

Dodaj odgovor

1. bodovi: 5

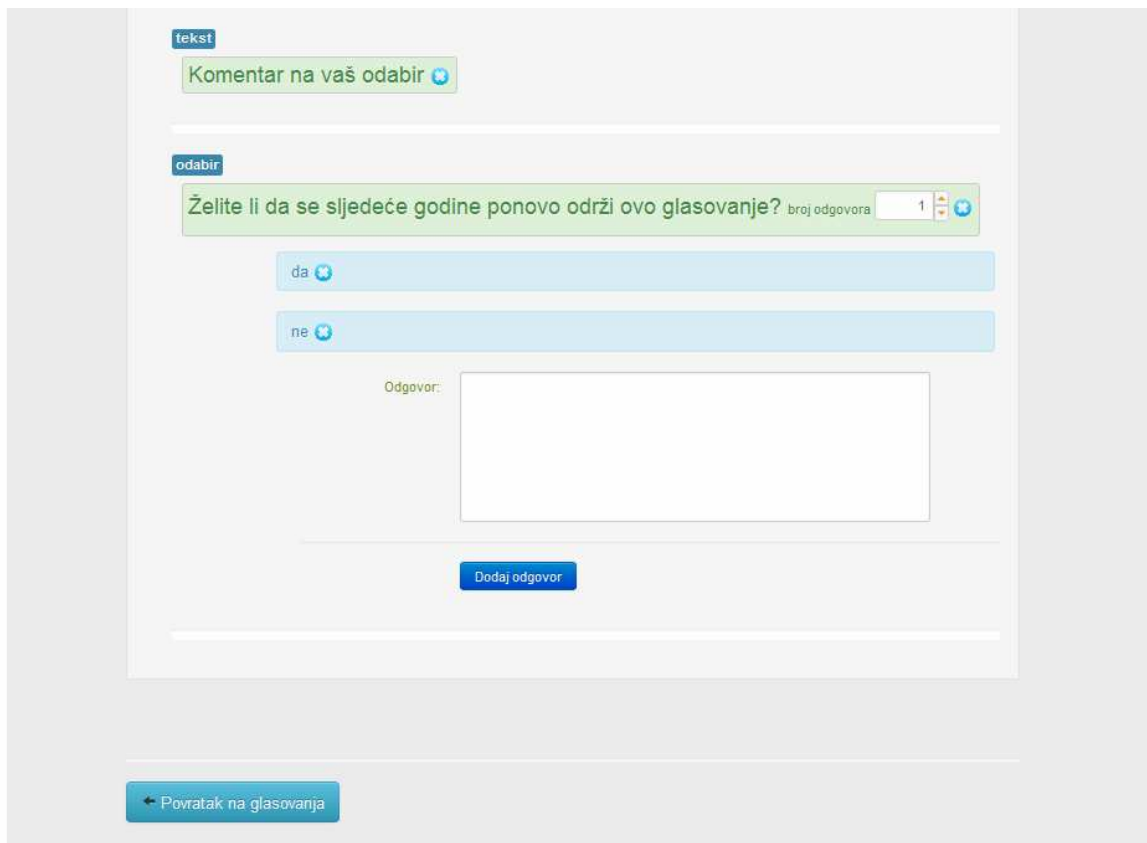
2. bodovi: 3

3. bodovi: 1

Slika 7.11. Prikaz uređivanja natjecateljskog pitanja



Na slici 7.12. su još prikazane forme za tekstualnih pitanja i pitanja s odabirom. Kod tekstualnih samo dodamo pitanje dok kod pitanja s odabirom je sve slično kao s natjecateljskim pitanjem samo da nema bodova nakon ponuđenih odgovora.



The screenshot shows a web interface for creating a question. It is divided into two main sections: 'tekst' (text) and 'odabir' (choice).

**tekst**

Komentar na vaš odabir

**odabir**

Želite li da se sljedeće godine ponovo održi ovo glasovanje? broj odgovora 1

da

ne

Odgovor:

Dodaj odgovor

Povratak na glasovanja

Slika 7.12. Prikaz uređivanja pitanja sa komentarom i pitanja sa odabirom

## 8. Zaključak

Elektroničko glasovanje mogu biti provedena na više načina, ali se ovaj rad se najviše bavi web aplikacijama za elektroničko glasovanje. U radu su spomenute i analizirane tri postojeće web aplikacije za elektroničko glasovanje. To su „BallotBin“, „eBallot“ i „Simple Voting“. Za takve aplikacije postoje neki standardi i osnovne funkcionalnosti koje bi one morale implementirati. Analiza je pokazala da neke od tih aplikacija imaju više mogućnosti, a neke manje. Kao neke od mogućnosti koje podržavaju sve navedene aplikacije možemo spomenuti listu korisnika, podržavanje više tipova glasovanja, razne opcije kod glasovanja, prikaz rezultata na različite načine itd.

U sklopu ovog rada je izrađena aplikacija za elektroničko glasovanje. Najvažnije tehnologije koje su korištene kod izrade aplikacije za ovaj rad su PHP koji je korišten za programiranje na serverskoj strani, HTML za prikaz sadržaja, CSS za formatiranje HTML-a, Smarty za odvajanje logike od prezentacije te JavaScript uz pomoć dodatne biblioteke JQuery-a za programiranje na klijentskoj strani.

Izrađena aplikacija podržava tri vste korisnika i svaki od njih naslijeđuje mogućnosti od onih koji su niže na hijerarhiji. Neke od glavnijih mogućnosti kod izradene aplikacije su kreiranje glasovanja, glasovanje i uređivanje glasovanje. Svako glasovanje ima dodatne opcije koje određuju tko smije glasovati, kako će se prikazivati rezultati itd. Kod aplikacije postoje još dodatne mogućnosti kao što je lista prijatelja koja služi da na određeno glasovanje mogu glasati samo prijatelji vlasnika glasovanja ili mogućnost upravljanja različitim postavkama aplikacije.

Aplikacija koja je izrađena u sklopu ovog radi ima mnogo mogućnosti za daljnji razvoj te predstavlja dobar temelj da se iz nje razvije ozbiljna i komercionala aplikacija. U budućnosti će biti implementirani različiti sigurnosni mehanizmi koji će sigurnost aplikacije povećati na visoku razinu. Zatim će biti implementirana open id tehnologija te napravljen „zid aplikacije“ koji će predstavljati dodatne informacije o glasovanju kao što su uvod u tematiku glasovanja, svrha glasovanja, očekivanja kod glasovanja itd. i moći će sadržavati tekst, tvuk, audio i video. Dodatne mogućnosti koje će još biti implementirane je mogućnost postavljanja glasovanja na neku vanjsku stranicu i mogućnost kreiranja grupe ljudi iz postojećih prijatelja koje će se moći onda koristiti za pojedina glasovanja.

## 9. Literatura

1. Caarls, S.(2010) Handbook on E-Voting.Strasbourg: Council of Europe. Dostupno na [http://www.coe.int/t/dgap/democracy/activities/GGIS/E-voting/E-voting%202010/Biennial\\_Nov\\_meeting/ID10322%20GBR%206948%20Evoting%20handbook%20A5%20HD.pdf](http://www.coe.int/t/dgap/democracy/activities/GGIS/E-voting/E-voting%202010/Biennial_Nov_meeting/ID10322%20GBR%206948%20Evoting%20handbook%20A5%20HD.pdf) (pristupano 01. 09.2012.)
2. Coleman S. i Cranor L.(2000) Online voting.London: Uk Parliamentary Office of science and Technology. Dostupno na <http://www.parliament.uk/documents/post/pn155.pdf> (pristupano 01. 09.2012.)
3. Ibrahim, I. et al. (2003) Secure E-Voting With Blind Signature: Introduction[online]. Malaysia: Faculty Of Computer Science & Information Technology - University Technology Of Malaysia. Dostupno na [http://eprints.utm.my/3262/1/IEEE02-EVS\\_full\\_paper\\_ver14Nov.pdf](http://eprints.utm.my/3262/1/IEEE02-EVS_full_paper_ver14Nov.pdf) (pristupano 01. 09.2012.)
4. IDEA (2011) Introducing Electronic Voting: Essential Considerations.Stockholm: IDEA(The International Institute for Democracy and Electoral Assistance). Dostupno na [http://www.idea.int/publications/introducing-electronic-voting/upload/PP\\_e-voting.pdf](http://www.idea.int/publications/introducing-electronic-voting/upload/PP_e-voting.pdf) (pristupano 01. 09.2012.)
5. Lovrenčić A, Konecki M, Orehovački T. 1957 - 2007: 50 Years of Higher Order Programming Languages. Journal of Information and Organizational Sciences, vol. 33 no. 1, pp. 135-140 (2009)
6. Welling, Luke; Thomson, Laura. PHP and MySQL web development 4th Edition. SAD: Addison-Wesley, 2009.
7. Kermek, Dragutin. "HTML jezik, osnovne oznake". Web dizajn i programiranje. Fakultet organizacije i informatike. Varaždin, 2011. [Predavanje]
8. Kermek, Dragutin. "Smarty predlošci". Web dizajn i programiranje. Fakultet organizacije i informatike. Varaždin, 2011. [Predavanje]
9. International Electoral Affairs Symposium: Speakers 2013. Dostupno na: <http://electoralforum.org/speakers.php>

10. Službena dokumentacija PHP-a. Popis i objašnjenje PHP funkcija. Dostupno na <http://www.php.net/manual/en/funcref.php> (pristupano 01. 09.2012.)
11. W3Schools web stranica za online tutorijale. HTML5. Dostupno na <http://www.w3schools.com/html5/default.asp> (pristupano 01. 09.2012.)
12. Službena stranica JQuery-a. Dokumentacija o AJAX funkcijama. Dostupno na: <http://api.jquery.com/jquery.ajax/> (pristupano 01. 09.2012.)