



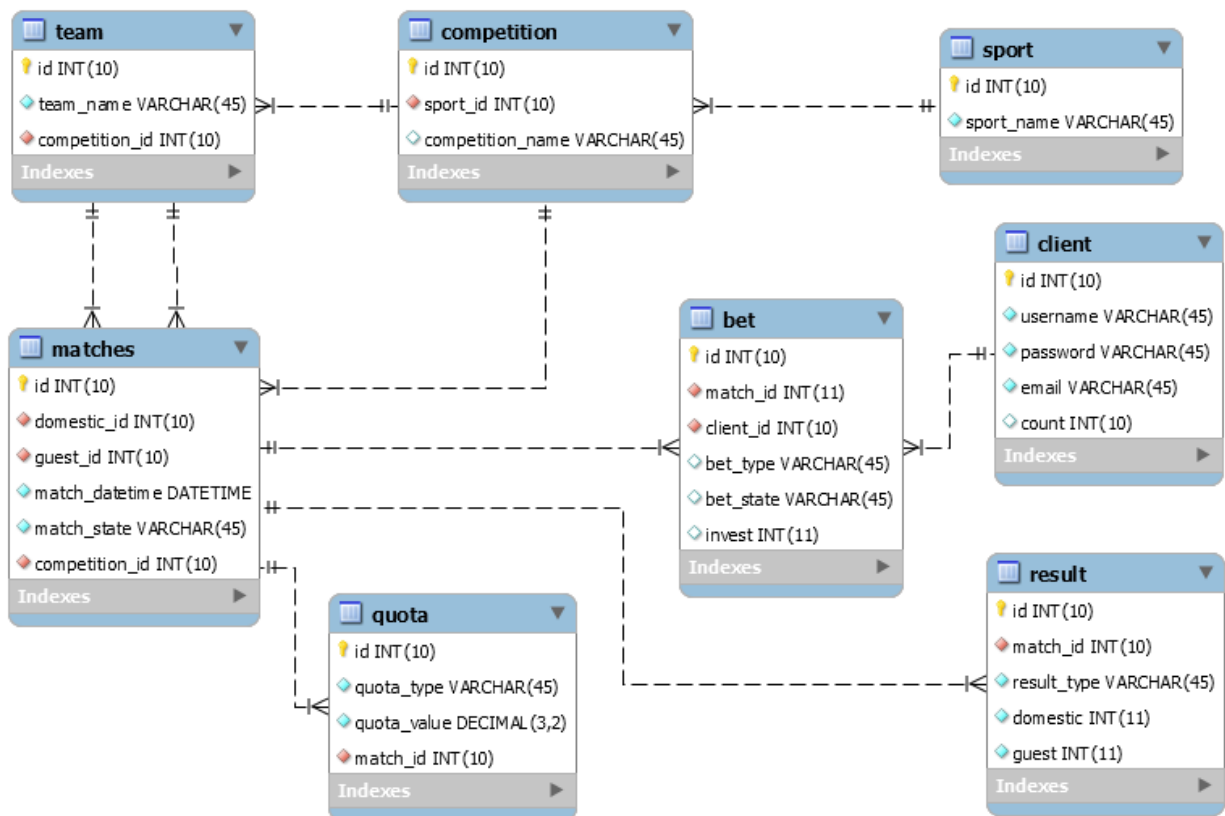
*Version 1.3*

# 1. Generalno

Aplikacija realizuje model onlajn kladionice. Trenutno nedostaje deo koji omogućava transakcije i sesije u front endu. Za back end korišćen je JdbcTemplate, a za front end korišćen je Thymeleaf, java biblioteka koja radi kao generator XML/XHTML/HTML5 šablona.

Sve je realizovano kao Spring Boot projekat.

## 2. Baza podataka



E-R Dijagram baze **betting\_base**

1. Postoje sportovi:

**sport**(**id**, sport\_name )

2. Postoje takmičenja:

**competition**(**id**, **sport\_id**, competition\_name ).

Takmičenja znaju kom sportu pripadaju.

3. Postoje timovi:

**team( id, team\_name, competition\_id ).**

Timovi znaju kom takmičenju pripadaju. Postoje situacije da jedan tim učestvuje u više takmičenja (na primer: u nacionalnom šampionatu, u nacionalnom kupu i u nekom internacionalnom takmičenju) i to se rešava tako što se jedan tim može u tabeli team pojaviti više puta, sa različitim competition\_name stranim ključem i, naravno, sopstvenim identifikatorom id. Posle određenog razmatranja (i probanja) zaključio sam da je to manje skupo rešenje nego posebna vezna tabela između timova i takmičenja. Posebna vezna tabela implicira bar još jedan JOIN u maltene svakom upitu koji radi sa takmičenjima, mečevima i sl, a ovo tekuće zahteva samo još jedan dodatan uslov u upitima koji eventualno rade sa ukupnom statistikom pojedinog tima, ili listaju sve timove jednog takmičenja npr...

4. Postoje mečevi:

**matches( id, domestic\_id, home\_id, match\_datetime, match\_state, competition\_id)**

Mečevi znaju kom takmičenju pripadaju, a naravno pamte i šifre timova koji ga igraju. Meč (tabela meč u bazi!!!) ne zna za svoj rezultat, kvote niti za eventualne opklade na njega.

5. Postoje rezultati:

**result( id, match\_id, result\_type, domestic, guest )**

Rezultati znaju na koji se meč odnose. Takođe, umesto da svi parametri koji se prate (rezultati četvrtina, ako postoje, rezultati poluvremena, kraja utakmice, zatim žuti i crveni kartoni i slično) imaju svoju kolonu, pamte se svi u istoj koloni (tačnije, u dve: jedna za domaći - **domestic**, a jedna za gostujući tim - **guest**). O kom se parametru radi govori nam kolona **result\_type**. Evo kako bi to izgledalo na primeru. Ispisujemo sadržaj (dela) samo tabele result:

	id	match_id	result_type	domestic	guest
▶	1	5	end game	2	1
	2	5	half time	1	0
	3	7	end game	0	0
	4	7	half time	0	0
	5	9	end game	2	3
	6	9	half time	0	2
	7	10	end game	1	1
	8	10	half time	1	1
	9	12	end game	4	1
	10	14	end game	2	1
	11	17	end game	0	1
	12	19	end game	1	5

Jedan red tabele result je samo deo rezultata jednog meča. Ako, na primer, pratimo rezultat na poluvremenu i rezultat na kraju utakmice imaćemo dva reda koji se tiču jednog istog meča.

Na slici su naznačena prva dva reda – oba se odnose na meč sa id = 5. Prvi red daje krajnji rezultat a drugi daje rezultat na poluvremenu.

Primetimo da se za mečeve sa id = 12, 14, 17 i 19, prati samo kraj utakmice, što je isto regularno.

6. Postoje opklade:

**bet(id, match\_id, bet\_type, bet )**

Opklade znaju na koji se meč odnose. Realizovane su na istom principu kao i tabela mečeva, samo je jednostavnije jer nema dva učesnika (kao dva tima u rezultatu): postoji kolona **bet\_type** koja jednostavno determiniše tip opklade i to je sve. Postoji još i kolona **bet\_state** koja opisuje stanje opklade: aktivna, završena u korist kladioničara / kladionice. Evo kako izgleda deo tabele bet. Primetimo da se više opklada može odnositi na isti meč. Takođe, treba naglasiti da, u ovom sistemu, opklada nije tiket. Posebna tabela čuvaće tikete (tj korisnike u personalizovanom sistemu) koji će grupisati opklade koje se odnose na isti tiket ili istu osobu.

id	match_id	bet_type	bet_state
1	4	1	active
2	4	1X	active
3	8	2	won
4	13	2	active
5	15	X2	active
6	15	12	active
7	18	2	active
8	23	1	active
9	25	X	active
10	25	2	active
11	5	2	lost

7. Tu su i kvote:

**quota(id, quota\_type, quota\_value, match\_id)**

Skup kvota koje se nude pamti se po istom principu kao i opklade i rezultati. Kvota zna na koji se meč odnosi.

8. Na kraju, imamo i klijente

**client(id, username, password, email, account)**

Sve kolone su jasne osim kolone count. Ona će da zameni posebnu tabelu koja bi predstavljala račun pojedinačnog klijenta.

Evo kako bi, primera radi, izgledao upit koji povezuje mečeve, timove i rezultate. Pozvaću se na njega u jednom trenutku.

*Dohvati detalje svih mečeva Prve lige sa rezultatima (poluvremena i kraja) ako postoje.*

Rezultujuća tabela je:

	match_id	domestic	guest	dati	state	domestic_end	guest_end	domestic_half	guest_half
▶	3	Partizan	Zvezda	2016-08-23 15:00:00	scheduled	NULL	NULL	NULL	NULL
	4	Partizan	Radnicki	2016-08-24 16:00:00	scheduled	NULL	NULL	NULL	NULL
	5	Partizan	OFK Beograd	2016-08-20 12:00:00	finished	2	1	1	0
	6	Zvezda	Partizan	2016-08-23 18:00:00	scheduled	NULL	NULL	NULL	NULL
	7	Zvezda	Radnicki	2016-08-19 12:00:00	finished	0	0	0	0
	8	Zvezda	OFK Beograd	2016-08-25 18:00:00	scheduled	NULL	NULL	NULL	NULL
	9	Radnicki	Partizan	2016-08-16 12:00:00	finished	2	3	0	2
	10	Radnicki	Zvezda	2016-08-19 12:00:00	finished	1	1	NULL	NULL
	11	Radnicki	OFK Beograd	2016-08-25 18:00:00	scheduled	NULL	NULL	NULL	NULL
	12	OFK Beograd	Partizan	2016-08-18 12:00:00	finished	4	1	NULL	NULL
	13	OFK Beograd	Zvezda	2016-08-26 15:00:00	scheduled	NULL	NULL	NULL	NULL
	14	OFK Beograd	Radnicki	2016-08-19 12:00:00	finished	2	1	NULL	NULL

SELECT

```
q1.match_id,
q1.domestic,
q1.guest,
q1.dati,
q1.state,
q2.domestic_end,
q2.guest_end,
q2.domestic_half,
q2.guest_half
```

FROM

```
(SELECT
    matches.id AS match_id,
    t1.team_name AS domestic,
    t2.team_name AS guest,
    matches.match_datetime AS dati,
    matches.match_state AS state
FROM
    matches
INNER JOIN team AS t1
INNER JOIN team AS t2 ON matches.competition_id = '2'
    AND matches.domestic_id = t1.id
    AND matches.guest_id = t2.id) AS q1
LEFT JOIN
```

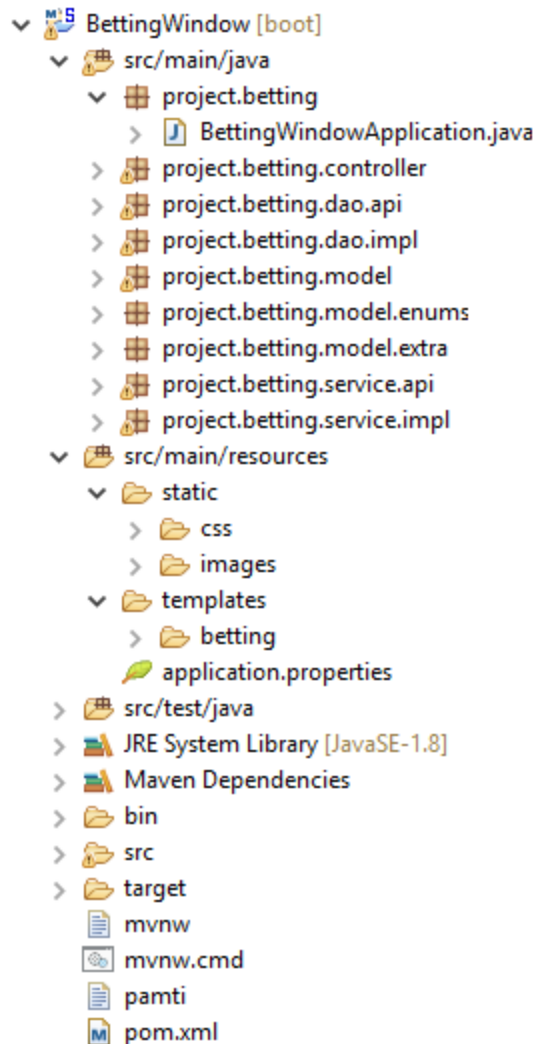
q1

```
(SELECT
    r1.match_id AS match_id,
    r1.domestic AS domestic_end,
    r1.guest AS guest_end,
    r2.domestic AS domestic_half,
    r2.guest AS guest_half
FROM
    result AS r1
LEFT JOIN result AS r2 ON r1.match_id = r2.match_id
    AND r2.result_type = 'half time'
WHERE
    r1.result_type = 'end game') AS q2
ON q2.match_id = q1.match_id;
```

q2

### 3. Struktura projekta

---



- Paket *model* sadrži domenske klase. Kao dodatak, postoje podpaketi *model.enums* sa enumeracijama i *model.extra* sa dodatnim klasama koje će biti opisane kad na njih dođe red.

- Paket *dao* sadrži odgovarajuće metode za komunikaciju sa bazom. Interfejsi su u podpaketu *dao.api*, a implementacije su u podpaketu *dao.impl*

- Paket *service*, sa odgovarajućim podpaketima koji implementiraju konkretne potrebe koje aplikacija ima prema bazi i uredjuju komunikaciju front enda i back enda.

- Paket *controller* sadrži klase sa metodama koje mapiraju pozive (linkove) sa front enda sa odgovarajućim šablonima, obezbeđujući komunikaciju sa bazom podataka.

- U fajlu *src/main/resources* nalaze se folder *static*, sa css fajlovima i korišćenim slikama, folder *templates* sa šablonima i *application.properties* fajl sa konfiguracionim podešavanjima za bazu podataka i thymeleaf.

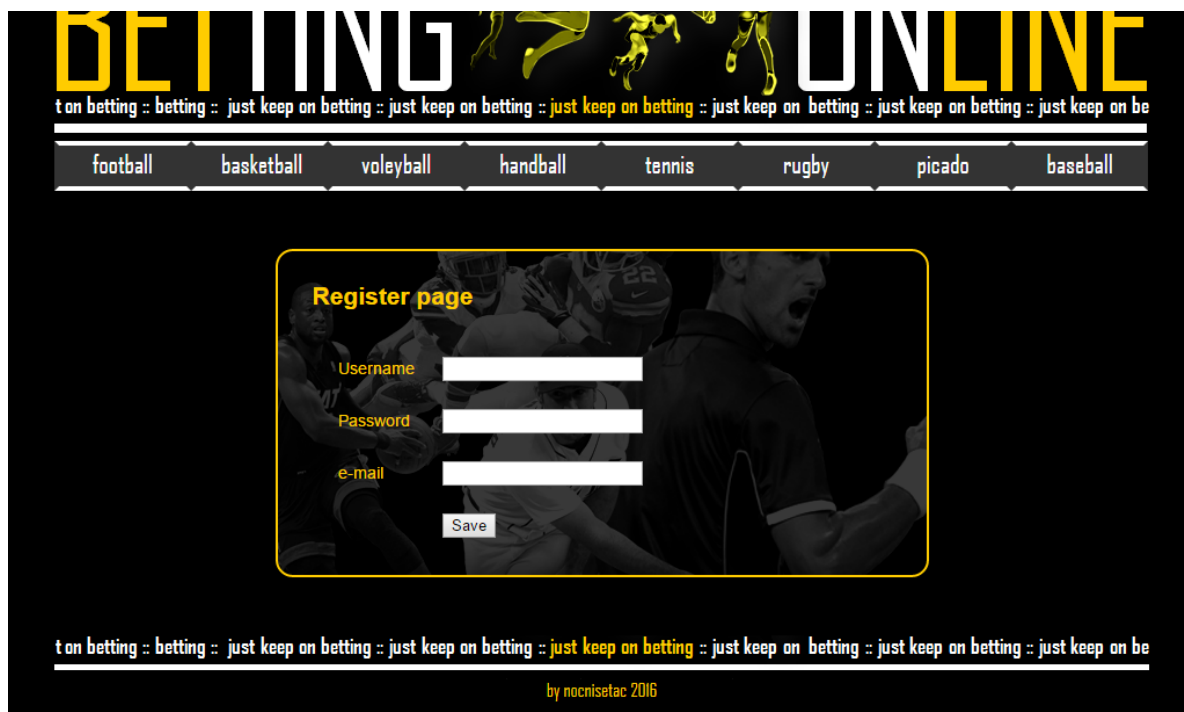
### 4. Demonstracija

---

Razmotrićemo kako aplikacija funkcioniše prikazujući sve što se dešava po dubini. Počinjemo od početne strane index.html. Ona prikazuje nekakav tekst dobrodošlice, formu za logovanje, link na stranu za registraciju i osnovni meni. Osnovni meni sadrži nabrojane sportove koji se prate. Ovo je ovako, naravno, na prvu loptu, a može da se ponudi par najpopularnijih sportova direktno, a ostalo kroz neki padajući meni ili slično. Krenimo redom.



Nismo registrovani na sistem, pa hajdemo na stranu za registraciju:



Podatke iz forme za registraciju prihvatiće instanca klase Client, sa atributima prikayanim na slici ispod:

```
1 package project.betting.model;  
2  
3 import java.util.List;  
4  
5  
6  
7  
8  
9 public class Client {  
10  
11     private long ID;  
12     @NotEmpty  
13     @Length(min=3, max=10)  
14     private String username;  
15     @NotEmpty  
16     @Length(min=3, max=10)  
17     private String password;  
18     @Email  
19     @NotEmpty  
20     private String email;  
21  
22     private int count;  
23  
24     private List<Bet> bets;  
25  
26 }
```

Klasa Client ima elementarnu validaciju. U sadejstvu sa Thymeleafom neće dozvoliti submit dok se ne unesu podaci sa zahtevanim ograničenjima.

The screenshot shows a registration form titled "Register page" with a background image of football players. It contains three input fields: "username" with the value "hh", "password" with an empty field, and "email" with the value "frggg". Each field has a corresponding error message in yellow text: "length must be between 3 and 10" for the username, "length must be between 3 and 10" for the password, and "not a well-formed email address" for the email. A "Save" button is located at the bottom of the form.

Takođe, sistem neće dozvoliti registraciju naloga sa šifrom (password) koja već postoji u bazi:

The screenshot shows a registration form titled "Register page" with a background image of football players. At the top, there is a red error message: "The password that you have chosen is already in use. Please, choose another." Below the message, there is a "Username" field with the value "veljo".

Ilustracije radi, isprat ćemo ceo proces registracije - po dubini:



Link REGISTER na index.html strani je oblika: `th:href="@{'/betting/register'}"`, traži stranu betting/register i to u formi podrazumevanog **GET** – zahteva. Kao što se vidi, reagovala metoda register birajući šablon register i prosleđujući mu instancu klase Clijent.

```
@RequestMapping(value = "register", method = RequestMethod.GET)
public String register(ModelMap modelMap) {
    modelMap.put("client", new Client());
    return "betting/register";
}
@RequestMapping(value = "register", method = RequestMethod.POST)
public String register(
    @ModelAttribute("client") @Valid Client account,
    BindingResult bindingResult,
    ModelMap modelMap) {
    if(bindingResult.hasErrors()) {
        modelMap.put("client", account);
        return "betting/register";
    } else {
        Err err = null;
        int exists = clientService.doesPasswordExist(account.getPassword());
        if(exists==1){
            err = new Err();
            modelMap.put("err", err);
            return "betting/register";
        }
        else {
            modelMap.put("client", account);
            modelMap.put("err", err);
            clientService.enterNewClient(account);
            return "betting/success";
        }
    }
}
```

Submit forma, pak ima drugačiji pristup: `<form th:action="@{/betting/register}" th:object="${client}" method="post">` pa će reagovati druga register metoda, koja implementira i validaciju, proverava unikatnost šifre i, na kraju, vrši upis u bazu.

`clientService.enterNewClient(account);`

**Kontroler**

```
@Override
public void enterNewClient(Client client) {
    clientDao.enterNewClient(client);
}
```

**Servis**

@Override

DAO

```
public long enterNewClient(Client client) {
    String sql =
        "INSERT INTO client " + "(username,password,email,count) VALUES (?, ?, ?, ?)";
    KeyHolder keyHolder = new GeneratedKeyHolder();

    jdbcTemplate.update(new PreparedStatementCreator() {
        public PreparedStatement createPreparedStatement(
            Connection connection) throws SQLException {
            PreparedStatement ps =
                connection.prepareStatement(sql, new String[] { "id" });

            int index = 1;
            ps.setString(index++, client.getUsername());
            ps.setString(index++, client.getPassword());
            ps.setString(index++, client.getEmail());
            ps.setInt(index++, 500);
            return ps;
        }
    }, keyHolder);

    client.setID((long) keyHolder.getKey());

    return client.getID();
}
```

Da preletimo ostale mogućnosti servisa. Kada je klijent ulogovan, dobiće sledeću stranicu:

The screenshot shows a betting website interface. At the top, there's a navigation bar with tabs for different sports: football, basketball, volleyball, handball, tennis, rugby, picado, and baseball. Below this, there's a user status bar showing "Hello veljo.", "competitions and teams", "bet !!!", "my bets", and "Log out". The main section is titled "Football - all matches, results and quotas". It contains a table with the following data:

competition	datetime	home	away	h	a	state	1	X	2
Prva Liga S	23-08-2016 03:00:00	Partizan	Zvezda	-	-	SCHEDULED	1.2	1.5	1.8
Prva Liga S	24-08-2016 04:00:00	Partizan	Radnicki	-	-	SCHEDULED	1.2	2.0	1.66
Prva Liga S	20-08-2016 12:00:00	Partizan	OFK Beograd	2	1	FINISHED	1.5	1.41	1.2
Prva Liga S	23-08-2016	Zvezda	Partizan	-	-	SCHEDULED	1.5	1.8	2.0

Podrazumevani sport je fudbal. Ono što se odmah ispisuje su svi mečevi koji se prate, sa rezultatima ako su gotovi, i sa kvotama na tipove koji se nude. Nekakav podmeni koji se pojavljuje nudi opciju competitions and teams, koja daje detalje o takmičenjima i pripadajućim timovima, zatim opciju bet!!!, koja nudi spisak mečeva na koje je moguće kladiti se i omogućava samo kladenje, i na kraju, opciju my bets gde klijent može da vidi svoje opklade, aktivne i završene, da pogleda ostvareni dobitak i slično...

Opcija competitions and teams:

The screenshot shows a betting website interface. At the top, there's a navigation bar with sports categories: football, basketball, volleyball, handball, tennis, rugby, picado, and baseball. Below this, there's a user greeting 'Hello veljo:', a 'competitions and teams' button (highlighted in orange), a 'bet !!!' button, a 'my bets' button, and a 'Log out' button. The main heading is 'Football competitions and teams'. Below this, there's a table with columns: competition, team, and details.

competition	team	details
Prva Liga S	Partizan	some link
	Zvezda	some link
	Radnicki	some link
	OFK Beograd	some link

Napominjem da je ovo moglo da se i kulturnije uradi pomoću DOM-a i Javascripta, bez nekih problema – sem jednog: grozne sintakse koja karakteriše komunikaciju thymeleafa i Javascripta/DOM-a (jer reč je o dinamičkim strukturama, tj tabelama koje Thymeleaf ispisuje dinamički, pa i npr identifikatore pojedinih elemenata i formi generiše sam, a i prosledjivanje argumenata postane nepregledno). Nije mi to bilo u fokusu, ali verovatno ću ispraviti. Inače kad sam već tu: postoje odlični tutoriali na youtube koji, ako se detaljno prorade, ekspresno uvode u priču sa Tajmlifom, ali za detalje nije lako naći pravi primer. Mora da se kopa po dokumentaciji, a oni njihovi tekstovi na sajtu imaju veliku manu što praktično traže da sve proradiš, pa opet nema sve tamo...

Opcija bet !!!:

The screenshot shows the same betting website interface, but with the 'bet !!!' button highlighted. The main heading is 'Football - bet right now !'. Below this, there's a table with columns: competition, datetime, home, away, 1, X, 2, unlock, tip, and invested.

competition	datetime	home	away	1	X	2	unlock	tip	invested
Prva Liga S	23-08-2016 03:00:00	Partizan	Zvezda	1.2	1.5	1.8	<input type="checkbox"/>	Choose tip ▼	
Prva Liga S	24-08-2016 04:00:00	Partizan	Radnicki	1.2	2.0	1.66	<input checked="" type="checkbox"/>	X ▼	40
Prva Liga S	23-08-2016 06:00:00	Zvezda	Partizan	1.5	1.8	2.0	<input type="checkbox"/>	Choose tip ▼	
Prva Liga S	25-08-2016 06:00:00	Zvezda	OFK Beograd	1.5	1.8	1.66	<input type="checkbox"/>	Choose tip ▼	
Prva Liga S	25-08-2016	Radnicki	OFK Beograd	1.2	1.11	1.8	<input type="checkbox"/>	Choose tip ▼	

Izlistava mečeve na koje je moguće kladiti se, sa kvotama. Tek kada se meč otključa, opcije tip i invested postaju aktivne. Takođe, obe opcije su obavezne, kada se opcija otključa (a može i da se ponovo deaktivira), tako da su propusti isključeni.

unlock	tip	invested
<input type="checkbox"/>	Choose tip ▾	
<input checked="" type="checkbox"/>	X ▾	40

<input checked="" type="checkbox"/>	X ▾	40
<input checked="" type="checkbox"/>	Choose tip ▾	40
<input type="checkbox"/>	Cho	
<input type="checkbox"/>	Choose tip	

! Please select an item in the list.

Posle submit, ispisaće se poruka o prihvaćenim opkladama kao i mečevi na koje se korisnik kladio, ulog, tip...

Na kraju, opcija my bets, daće podatke o mečevima na koje se do tada klijent kladio, eventualni dobitak koji je ostvario i sl...

Football
Basketball
Volleyball
Handball
Tennis
Rugby
Picudo
Baseball

Hello veljo:)

competitions and teams

bet !!!

my bets

Log out

My bets

bet ID	home	away	datetime	bet tip	bet state	invest	gain
67	Partizan	Radnicki	24-08-2016 04:00:00	1	ACTIVE	50	60.000004
69	Zvezda	OFK Beograd	25-08-2016 06:00:00	2	ACTIVE	150	249.0
70	OFK Beograd	Zvezda	26-08-2016 03:00:00	2	ACTIVE	100	150.0
72	Juventus	Milan	28-08-2016 11:00:00	2	WON	200	230.0
74	Napoli	Inter	29-08-2016	1	LOST	40	-