05. Arhitekturni projekat

eSports Management System

tim: **eSports Systems**

1. Bojan Adamović **13194**

2. Dejan Stošić **13466**

3. Milutin Stanković **14161**

4. Mario Žalac **14232**

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 29.04.2014. | 1.0 | Inicijalna verzija | eSports system tim |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 3

2. Opseg dokumenta 3

3. Reference 3

4. Pregled slučajeva korišćenja 3

*5.* Profili korisnika 5

*6.* Opis slučajeva korišćenja 6

*7.* Dodatni zahtevi 11

8. Pogled na logičku arhitekturu sistema 12

9. Pogled na procese 13

10. Pogleda na raspoređivanje sistema 14

11. Pogled na implementaciju sistema 15

12. Performane 18

13. Kvalitet 18

Specifikacija zahteva

# Cilj dokumenta

Cilj ovog dokumenta je specifikacija zahteva u pogledu detaljnog opisa slučajeva korišćenja eSMS-a.

# Opseg dokumenta

Dokument se odnosi na eSMS koji će biti razvijen od strane eSMS tima. eSMS predstavlja skraćenicu za eSports Management System. Namena sistema je obavljanje svih administrativnih aktivnosti za eSports turnire i mečeve.

# Reference

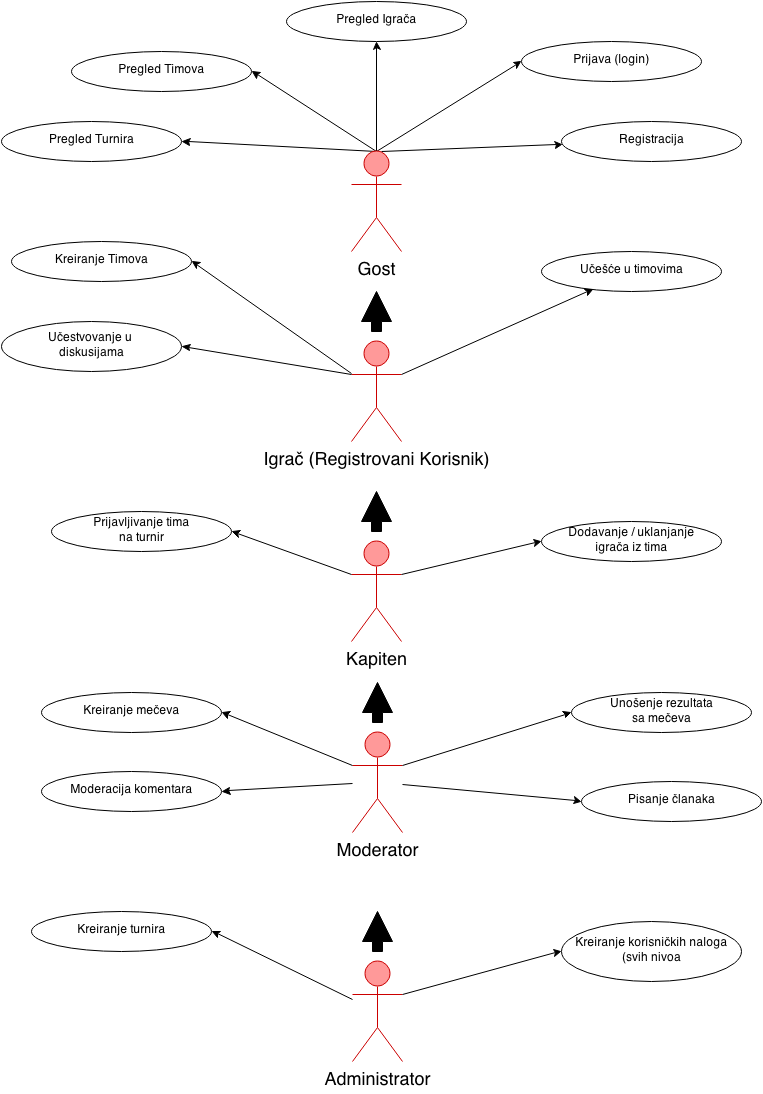
Spisak korišćene literature:

1. eSMS – Predlog projekta, V1.0, 2014
2. eSMS – Raspored aktivnosti, V1.0, 2014
3. eSMS – Plan realizacije projekta, V1.0, 2014

MS – Vizija sistema, V1.0, 2014

# Pregled slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja eSMS-a prikazan je na slici 01:



**SLIKA 01**

# Profili korisnika

Identifikovani su sledeći profili korisnika eSMS

## Gost (nivo 0)

Gost (neregistrovani korisnik) može vršiti pregled sadržaja na sajtu. Može pratiti turnire, rezultate o turnirima, može čitati informacije o timovima i njihovim statistikama i može gledati profile igrača i kapitena, kao i njihove pojedinačne učinke. Gost se može besplatno registrovati i time postati „igrač“.

## Igrač (registrovani korisnik – nivo 1)

Igrač se može prijaviti za pristup timovima, može biti odabran od strane kapitena timova za učešće u timu, može postaviti svoj lični profil sa informacijama i detaljima, i može kreirati novi tim čime postaje kapiten tog tima. Registrovani korisnik može komentaristati i učestvovati u diskusijama na postovima i na forumu.

## Kapiten (nivo 2)

Kapiten je praktično moderator svog tima. Kapiten može prijaviti svoj tim za učešće u otvorenim turnirima, može dodavati i brisati igrače iz timova i može unaprediti ostale igrače u timu u ulogu kapitena. Kapiten može menjati informacije (naziv, logo i opis) svog tima

## Moderator (nivo 3)

Nalog moderatora može kreirati samo administrator, ili može unaprediti neki od postojećih naloga u ulogu moderatora. Moderator može unositi rezultate i statistike sa mečevea, može kreirati nove mečeve. Moderator može pisati članke na blogu i može brisati i/ili cenzurisati komentare. Takođe je moguća zabrana pristupa nekom od nižih nivoa korisnika na određeni vremenski period („banovanje).

## Administrator (nivo 4)

Vlasnik sistema sa potpunim pravima pristupa celom sistemu. Može neograničeno kreirati timove, korisnike svih nivoa i turnire. Ima potpunu mogućnost brisanja i menjanja svih komentara i menjanja podešavanja sistema.

# Opis slučajeva korišćenja

## Informacije o timovima/igračima/turnirima

**Kratak opis:**

Pregled informacijama o timovima/igračima/turnirima

**Akteri:**

Gost, Igrač, Kapiten, Moderator, Administrator

**Preduslovi:**

Nema.

**Osnovni tok:**

1. Akter bira određenu stavku u meniju timovi/igrači/turniri
2. U novom pogledu (view) se pojavljuje lista timova/igrača/turnira
3. Korisnik može a i ne mora primeniti određene filtere nad prikazanim podacima
4. Korisnik bira tim/igrača/turnir.

**Izuzeci:**

Nema.

**Posledice:**

Informacije su prikazane.

## Registracija

**Kratak opis:**

Registracija gosta i pravljenje novog korisničkog naloga nivoa1.

**Akteri:**

Gost

**Preduslovi:**

Nema.

**Osnovni tok:**

1. Gost bira iz menija formu za registraciju
2. Gost unosi svoje podatke i svoju email adresu.
3. Sistem automatski šalje email na adresu gosta u kom se nalazi ključ za aktivaciju naloga
4. Korisnik sada može aktivirati svoj nalog unosom ključa u sistemi.

**Izuzeci:**

Nema.

**Posledice:**

Kreiran novi nalog u bazi.

## Prijavljivanje na sistem

**Kratak opis:**

Prijavljivanje korisnika na sistem u cilju korišćenja aplikacije.

**Akteri:**

Gost, Igrač, Kapiten, Moderator, Administrator

**Preduslovi:**

Nema.

**Osnovni tok:**

1. Korisniku bira formu za prijavljivanje iz menija
2. Korisnik unosi korisničko ime i lozinku, te nakon toga aktivira komandu *Prijavi se*.
3. Uneti podaci se proveravaju [izuzetak: pogrešno korisničko ime i/ili lozinka].
4. Prikazuje se određena forma u zavisnosti od privilegija korisnika sa određenom kombinacijom korisničko ime – lozinka.

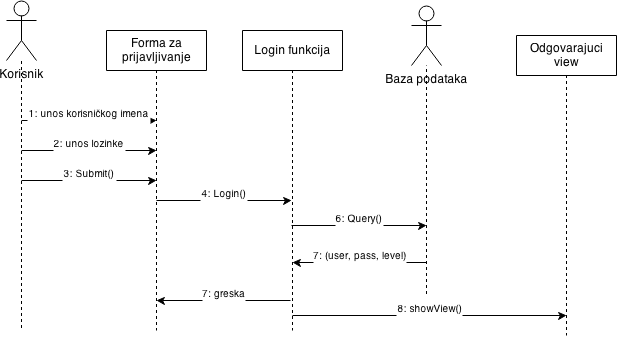
**Izuzeci:**

[*Pogrešno korisničko ime i/ili lozinka*] Prikazuje se obaveštenje na formi za prijavljivanje.

**Posledice:**

Korisnik je prijavljen i može koristiti aplikaciju.

**Dijagram sekvence:**



## Kreiranje tima

**Kratak opis:**

Kreiranje novog tima u bazi

**Akteri:**

Igrač, Kapiten, Moderator, Administrator

**Preduslovi:**

Nema.

**Osnovni tok:**

1. Korisnik bira formu za kreiranje tima
2. Unosi podatke o timu (naziv, logo, o timu)

**Izuzeci:**

[*Postoji tim sa istim imenom*] Korisnik dobija informaciju da unese novo ime tima.

**Posledice:**

Kreiran novi tim u bazi u bazi.

## Zahtev za pristup timu

**Kratak opis:**

Igrač može da pošalje zahtev da pristupi timu.

**Akteri:**

Igrač

**Preduslovi:**

Mora postojati mesto u timu.

**Osnovni tok:**

1. Igrač otvara profil tima.
2. Na profilu postoji dugme, na koje se klikom šalje zahtev kapitenu datog tima.
3. Ukoliko prihvati, igrač je sada član tima.

**Izuzeci:**

Nema.

**Posledice:**

Igrač dodat u tim.

## Dodavanje igrača u tim

**Kratak opis:**

Dodavanje igrača u postojeći tim.

**Akteri:**

Kapiten, Moderator, Administrator

**Preduslovi:**

Nema.

**Osnovni tok:**

1. Korisnik otvara profil bilo kog aktivnog igrača.
2. Na profilu postoji dugme, na koje se klikom šalje pozivnica datom igraču da pristupi timu.
3. Ukoliko prihvati, igrač je sada član tima.

**Izuzeci:**

Nema.

**Posledice:**

Igrač dodat u tim.

## Unošenje rezultata mečeva

**Kratak opis:**

Nakon završetka mečeva moguće je ažuriranje i unošenje informacija u sistem.

**Akteri:**

Moderator, Administrator

**Preduslovi:**

Nema

**Osnovni tok:**

1. Korisnik iz posebnog administracionog panela bira meč
2. Korisnik može uneti informacije za dati meč (rezultat, statistiku, izabrane karaktere...)

**Izuzeci:**

Nema.

**Posledice:**

Informacije o meču unete u bazu

## Pisanje članaka

**Kratak opis:**

Dodavanje novih članaka na sajt.

**Akteri:**

Moderator, Administrator

**Preduslovi:**

Nema

**Osnovni tok:**

1. Korisnik iz posebnog administracionog panela bira „Blog“
2. Korisnik unosi tekst i naslov posta.

**Izuzeci:**

Nema.

**Posledice:**

Informacije o postu unete u bazu i vidljive na frontendu.

## Kreiranje turnira

**Kratak opis:**

Kreiranje novog turnira

**Akteri:**

Administrator

**Preduslovi:**

Nema

**Osnovni tok:**

1. Korisnik iz posebnog administracionog panela bira formu za kreiranje turnira
2. Korisnik unosi podatke o turniru i format takmičenja
3. Korisnik bira timove ili postavlja turnir na „open“ varijantu (moguća slobodna prijava svih registrovanih timova)

**Izuzeci:**

Nema.

**Posledice:**

Novi turnir kreiran i može se videti na frontendu.

## Kreiranje korisničkih naloga

**Kratak opis:**

Kreiranje novog korisničkog naloga od strane administratora.

**Akteri:**

Administrator

**Preduslovi:**

Nema

**Osnovni tok:**

1. Korisnik iz posebnog administracionog panela bira panel za upravljanje korisnicma
2. Unosi podatke o novom korisniku kao i nivo pristupa (0 – 5)

**Izuzeci:**

Nema.

**Posledice:**

Novi korisnik dodat u bazu

## Brisanje korisničkih naloga

**Kratak opis:**

Brisanje korisničkog naloga od strane administratora.

**Akteri:**

Administrator

**Preduslovi:**

Nema

**Osnovni tok:**

1. Administrator iz posebnog administracionog panela bira panel za upravljanje korisnicma
2. Administrator bira korisnika kog želi da ukloni

**Izuzeci:**

Nema.

**Posledice:**

Korisnik izbačen iz baze

# Dodatni zahtevi

## Funkcionalnost

U ovom odeljku su specificirani funkcionalni zahtevi koji su zajednički za više slučajeva korišćenja.

eSMS portal ne zahteva nikakve dodatne funkcionalnosti.

## Upotrebivost

U ovom odeljku su specificirani zahtevi koji su vezani za, ili utiču na upotrebivosti sistema koji se razvija.

**Korisnički interfejs prilagođen korisniku:**

Korisnički interfejs eSMS portala će biti dizajniran tako da bude omogućeno jednostavno i intuitivno korišćenje bez potrebe za organizovanjem dodatne obuke.

## Pouzdanost

U ovom odeljku su specificirani zahtevi u pogledu pouzdanosti sistema koji se razvija.

**Dostupnost:**

eSMS portal će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%.

**Srednje vreme između otkaza:**

Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.

## Performanse

U ovom odeljku su specificirani zahtevi koji definišu željene performanse sistema koji se razvija.

**Broj korisnika koji simultano pristupaju:**

Sistem će da podrži do 100000 simultanih pristupa korisnika portalu.

**Vreme odziva za pristup bazi podataka:**

Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da veće od 5 sekundi.

## Podrška i održavanje

U ovom odeljku su specificirani zahtevi koji treba da poboljšaju stepen podrške i mogućnost održavanja sistema koji se razvija.

eSMS portal ne zahteva posebnu podršku i održavanje.

## Ograničenja

U ovom odeljku su specificirana ograničenja kojih se treba pridržavati pri projektovanju sistema koji se razvija.

**Hardverska platforma:**

Klijentski deo sistema treba da ima mogućnost izvršavanja na bilo kom PC računaru sa minimum Pentium II procesorom i 128 MB RAM memorije.

# Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

Logički pogled na eSMS portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži PHP kod zadužen za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži PHP kod koji predstavlja interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## 8.1 Pregled arhitekture – organizacija paketa i podsistema u slojeveC:\Users\Goran Zalac\Desktop\Use Case Diagram1.jpg

### 8.1.1 Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs aplikacije. U njemu su sadržane forme i funkcionalnosti potrebne za administrativno koriscenje.Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa pristupa podacima.

### 8.8.2 Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi eSMS portala. Sadrži PHP kodove koji realizuju funkcionalnost karakterističnu za domen primene sistema i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

### Ovaj sloj zavisi od sloja za pristup podacima i PHP kodova

### 8.8.3 Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži PHP kodove zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MySQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavistan od paketa PHP i MySQL baza podataka.

### HTML

tehnologija

Tehnologija HTML-a definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### 8.8.5 PHP

tehnologija

Tehnologija PHP-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte mogu da generišu HTML kod koji realizuje korisnički interfejs i pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### MySQL

DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju eSMS aplikacije.

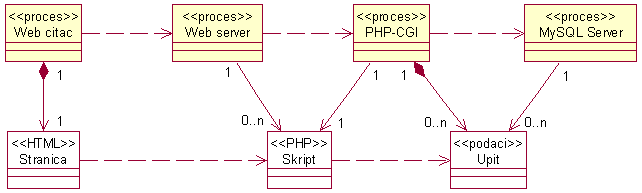
# Pogled na procese

U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

Web aplikacije zasnovane na PHP-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta PHP Web aplikacije nije potrebno voditi računa o načinu rada Web servera i načinu izvršavanja skripti.

9.1 Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje eSMS portala kao Web aplikacije.

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju eSMS sistema. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na PHP u i MySQL bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu HTML stranicu.

Web čitač zavisi od Web servera koji generiše i vraća odgovarajuću HTML stranicu na zahtev.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Ukoliko je zahtevana stranica PHP skript, Web server inicira izvršenje PHP-CGI procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj PHP-CGI procesa.

### PHP-CGI

PHP-CGI proces obavlja posao obrade zadatog PHP skripta i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web čitaču. Za izvršenje PHP skripta ovaj proces može da zahteva usluge MySQL servera-a. Komunikacija između PHP-CGI procesa i MySQL servera se obavlja preko prosleđivanja upita i vraćanja rezultat.

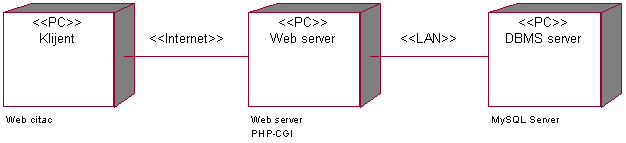
### MySQL Server

MySQL Server je proces koji izvršava funkcionalnost MySQL sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja eSMS sistema.



## Klijent

Pristup eSMS sistemu se obavlja preko klijentskih računara na kojima se izvršava aplikacija. Za povezivanje između klijenta i servera baze podataka koristi se Internet (ili LAN) infrastruktura tako da nema ograničenja u pogledu lokacije klijenta (osim u offline LAN slucaju).

## Aplikacija

Računar na kome se izvršava aplikacija opslužuje jednog klijenata koji pristupa preko Interneta(LAN-a). U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa aplikacijom u lokalnoj mreži (LAN).

## DBMS server

DBMS server je računar na kome se izvršava MySQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju eSMS sistem ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

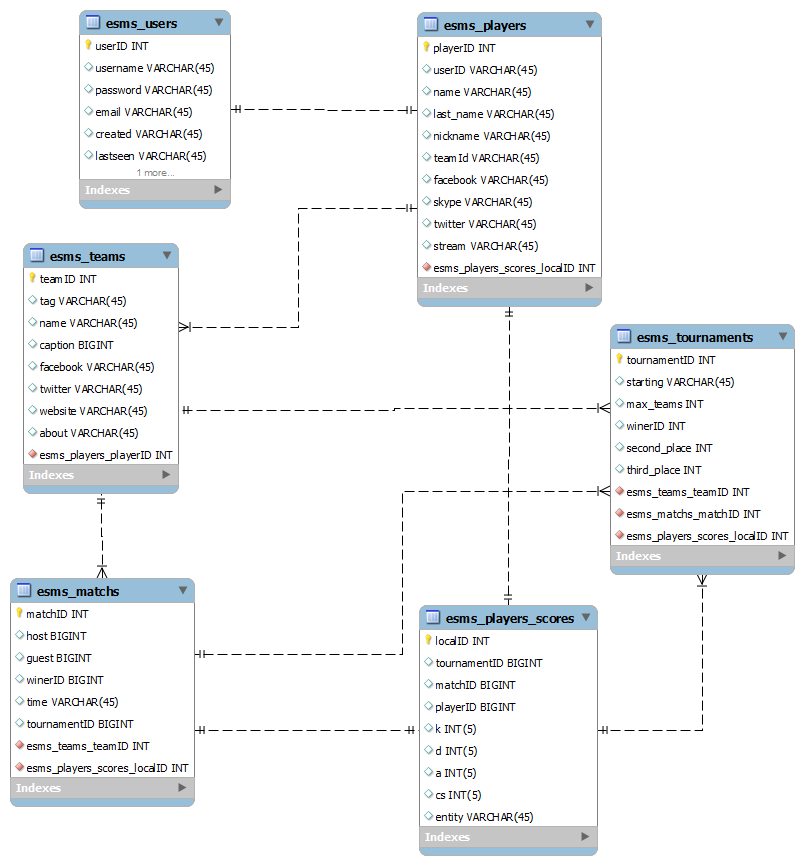
## Model domena

Model domena za koji se eSMS sistem projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (PHP skriptu) koje će biti implementirane.

## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *MS Access*-a, dok je migracija na MySQL obavljena pomoću *MySQL Migration Toolkit*-a.



## Komponente sistema

Komponente sistema eSMS portala su PHP skripti čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je PHP skript prikazan kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

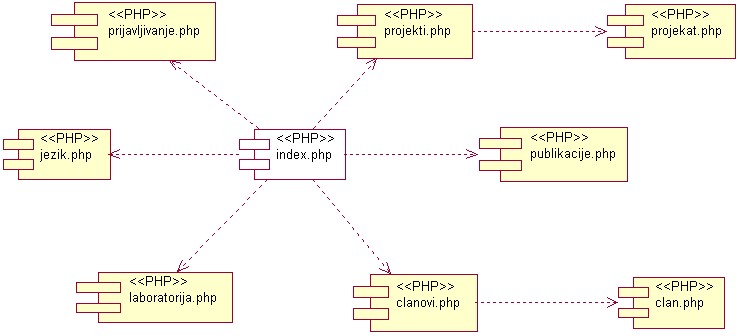


Komponenta **index.php** je implementira stranicu portala čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

Komponenta **main.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.php** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:

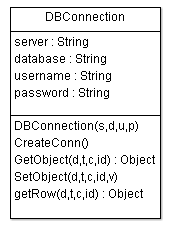


### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru PHP skripta **db.php**. Pomenuti skript se uključuje na početku index.php-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.



Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



Navedene funkcije obavljaju sledeće zadatke:

* DBConnection – inicijalizacija klase za konekciju sa bazom podataka
* CreateConn – kreira konekciju ka bazi podataka sa parametrima koji su u atributima instance
* getObject – vraca objekt zadate baze, tabele kolone i id-a
* setObject – postavlja objekt zadate baze, tabele kolone i id-a
* getRow – vraca red zadate baze, tabele i id

# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi:

1. Sistem će da podrži do 1000 simultanih pristupa korisnika portalu.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 sekundi.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme [5].

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [5]:

1. PeNcIL portal će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.