****

**Projektovanje informacionih sistema**

**Projektni zadatak**

**Tema:**

# Informacioni sistem plivačkog kluba

**Profesor: Student:**

**Prof. dr. Vladislav Miškovic Nikola Trišić**

# Uvod

Da bi se određeni plivacki klub mogao održati, potrebno je imati i funkcionalan sistem za prijavu. Potrebno je provjeriti sve prijave, obezbijediti da ne bude veliki broj korisnika plivackog kluba, te da se nepotpune prijave odbace.

## Cilj razvoja

Uvođenje informacionog sistema će olakšati održavanje plivackog kluba, lakše vođenje evidencije korisnika, te njihove prethodne rezultate. Informacioni sistem plivackog kluba projektovan je u svrhu lakše i jednostavnije prijave za određene mototike. Takođe preko informacionog sistema može se registrovati korisinik ako prvi put pristupa informacionom sistemu .

## Obim sistema

Sistem obuhvata primanje prijava i mogućnost registracije , te kasniju provjeru podataka na osnovu koje će se donijeti odluka o prihvatanju ili odbijanju prijave odnosno registracije.

## Rečnik

**IS** – *Informacioni sistem* je integrisani skup komponenti za sakupljanje, snimanje, čuvanje, obradu i prenošenje informacija.

Informacioni sistem – integrisani skup komponenti za skupljanje, snimanje i čuvanje podataka.

Server –poslužitelj je namjensko računalo ili softver koji šalje i prima podatke od mnogostrukih klijenata.

Web-hosting – tip internet hostinga koji dozvoljava organizacijama i fizičkim licima da naprave svoj web-sajt kome se može pristupiti preko interneta.

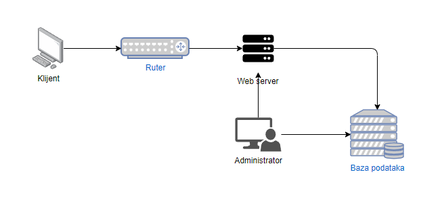
## Reference

<https://pksharks.com/>

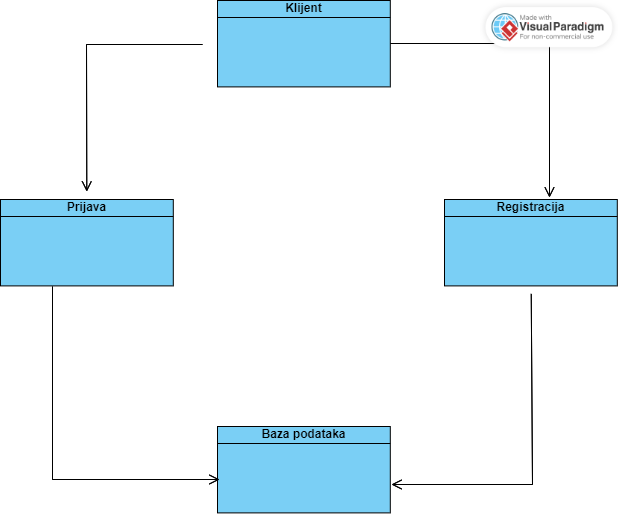
<https://www.pkcrvenazvezda.org.rs/>

# Dijagram isporuke (Deployment Diagram)

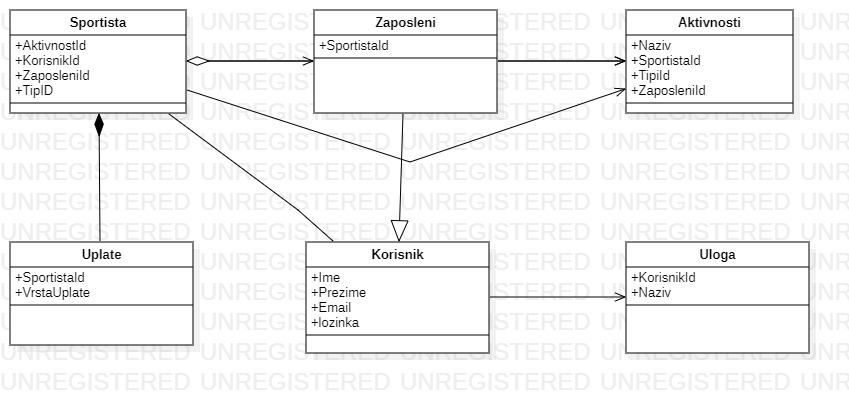
Klijent preko rutera pristupa web serveru. Nakon zavrsene prijave, pristupa se bazi podataka gdje se prijava skladišti. Takođe administrator ima pristup i web serveru i bazi podataka.



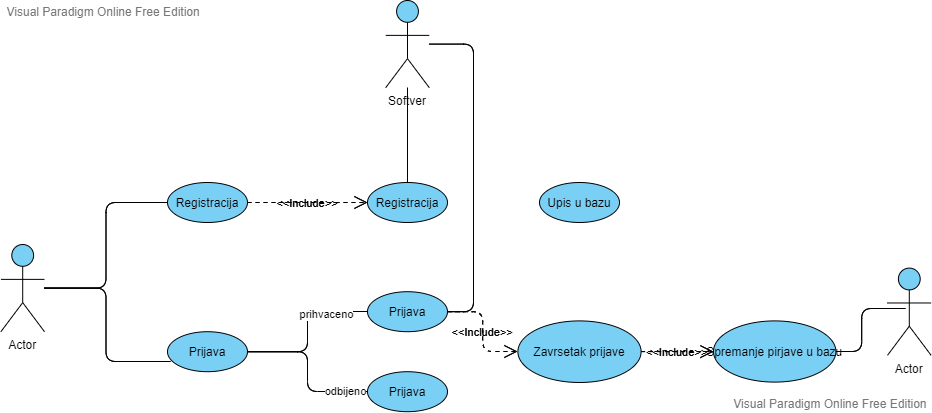
# Projekt arhitekture sistema (Architectural Design)



# Projekt strukture podataka (Data Structure Design)

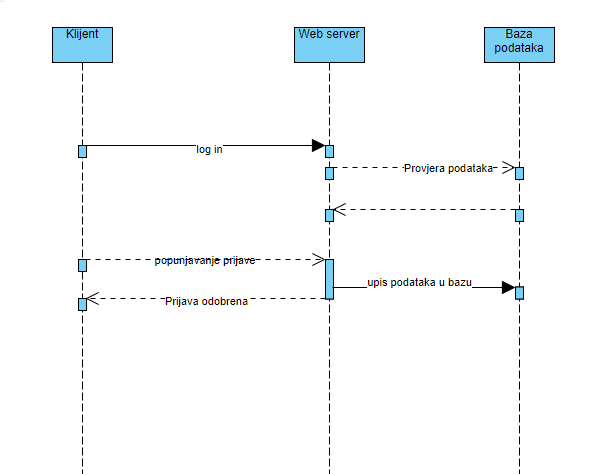


# Projekt komponenti sistema (Use Case Realizations)





Dijagram aktivnosti

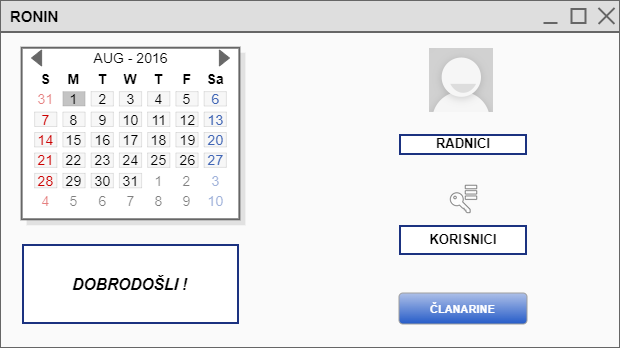


Dijagram sekvenci

Scenario :

Klijent pristupa web serveru kako bi se prijavio u plivacki klub .Web server provjerava da li postoji u bazi podataka uneseni user name i password. Nakon log in-a slijedi popunjavanje prijave. Web server provjerava da li korisnik ispunjava uslove za prijavu, te njegovu prijavu bilježi u bazi podataka. Na kraju vraća korisniku poruku o uspješnoj prijavi.

# Projekt korisničkog interfejsa (User Interface Design)

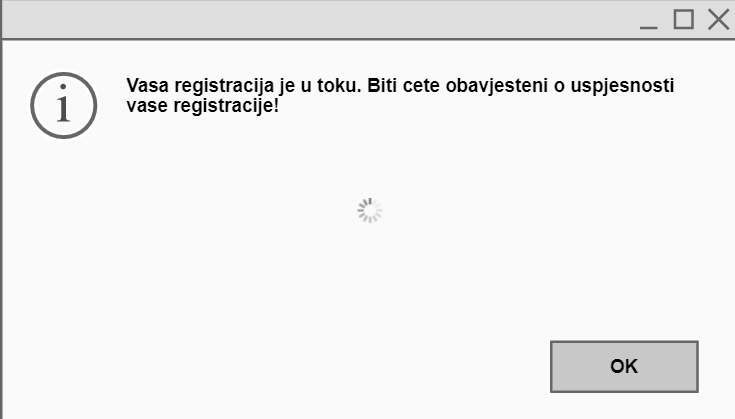


**Interfejs za zaposle**

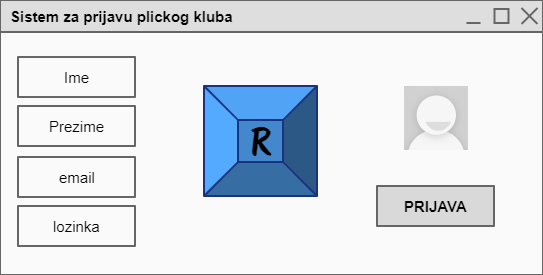


**Interfejs za registraciju na sistem**

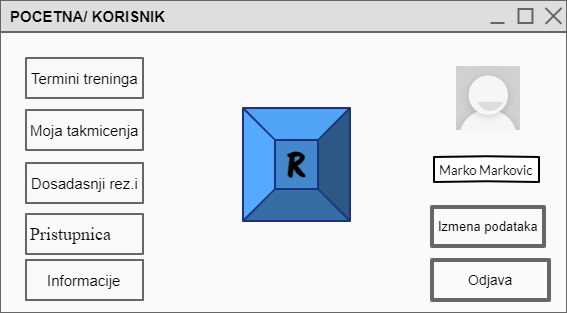
**(registracija)**



**Interfejs za registraciju na sistem**

****

**Interfejs za prijavu korisnika na sistem**

****

**Interfejs za podatke o korisniku**

# Ostali elementi sistema

Informacioni sistem zahtjeva konstatno kontrolisanje svih elemenata u cilju obezbeđenja nesmetanog funkcionisanja. IS mora da obezbjedi bezbjednost, raspoloživost i pouzdanost. IS treba da omogućiti brzu reakciju na korisničke zahtjev.

**Prilog: *Izvorni kod softverskog sistema***

* ***GENERISANI KOD KLASE UNOS PODATAKA***

\* This code has been generated by the Rebel: a code generator for modern Java.

\*

\* Drop us a line or two at feedback@archetypesoftware.com: we would love to hear from you!

\*/

package Model;

import java.util.\*;

import java.time.\*;

// ----------- << imports@AAAAAAGF34kSILyCcHo= >>

// ----------- >>

// ----------- << class.annotations@AAAAAAGF34kSILyCcHo= >>

// ----------- >>

public class Baza Podataka {

// ----------- << attribute.annotations@AAAAAAGF34lSFLys/UE= >>

// ----------- >>

private void UpisUBazu;

// ----------- << attribute.annotations@AAAAAAGF34pnXLzAWeQ= >>

// ----------- >>

private void BrisanjeIzBaze;

// ----------- << attribute.annotations@AAAAAAGF34tSP73a+Hg= >>

// ----------- >>

private Softver ;

public void getUpisUBazu() {

return UpisUBazu;

}

public void getBrisanjeIzBaze() {

return BrisanjeIzBaze;

}

public Softver get() {

return ;

}

public void setUpisUBazu(void UpisUBazu) {

this.UpisUBazu = UpisUBazu;

}

public void setBrisanjeIzBaze(void BrisanjeIzBaze) {

this.BrisanjeIzBaze = BrisanjeIzBaze;

}

public void set(Softver ) {

this. = ;

}

public void link(Softver \_) {

if (\_ != null) {

\_.unlink();

\_.set(this);

}

unlink();

set(\_);

}

public void unlink() {

if (get() != null) {

get().set(null);

set(null);

}

}

// ----------- << class.extras@AAAAAAGF34kSILyCcHo= >>

// ----------- >>

}

***GENERISANI KOD KLASE KORISNIK***

/\*

\* This code has been generated by the Rebel: a code generator for modern Java.

\*

\* Drop us a line or two at feedback@archetypesoftware.com: we would love to hear from you!

\*/

package Model;

import java.util.\*;

import java.time.\*;

// ----------- << imports@AAAAAAGF34TCQruUWr0= >>

// ----------- >>

// ----------- << class.annotations@AAAAAAGF34TCQruUWr0= >>

// ----------- >>

public class Korisnik {

// ----------- << attribute.annotations@AAAAAAGF34aKKLvDK0Q= >>

// ----------- >>

private String Ime;

// ----------- << attribute.annotations@AAAAAAGF34bbk7vUHww= >>

// ----------- >>

private String Prezime;

// ----------- << attribute.annotations@AAAAAAGF34cPO7vbAzg= >>

// ----------- >>

private int Godine;

// ----------- << attribute.annotations@AAAAAAGF34dXxLvibic= >>

// ----------- >>

private void Attribute1;

// ----------- << attribute.annotations@AAAAAAGF34svK7zIkU8= >>

// ----------- >>

private Softver ;

public String getIme() {

return Ime;

}

public String getPrezime() {

return Prezime;

}

public int getGodine() {

return Godine;

}

public void getAttribute1() {

return Attribute1;

}

public Softver get() {

return ;

}

public void setIme(String Ime) {

this.Ime = Ime;

}

public void setPrezime(String Prezime) {

this.Prezime = Prezime;

}

public void setGodine(int Godine) {

this.Godine = Godine;

}

public void setAttribute1(void Attribute1) {

this.Attribute1 = Attribute1;

}

public void set(Softver ) {

this. = ;

}

public void link(Softver \_) {

if (\_ != null) {

\_.unlink();

\_.set(this);

}

unlink();

set(\_);

}

public void unlink() {

if (get() != null) {

get().set(null);

set(null);

}

}

// ----------- << method.annotations@AAAAAAGF34eRabvoALs= >>

// ----------- >>

public void DodajKorisnika() {

// ----------- << method.body@AAAAAAGF34eRabvoALs= >>

// ----------- >>

}

// ----------- << method.annotations@AAAAAAGF34fpk7vwf3A= >>

// ----------- >>

public void UkloniKorisnika() {

// ----------- << method.body@AAAAAAGF34fpk7vwf3A= >>

// ----------- >>

}

// ----------- << class.extras@AAAAAAGF34TCQruUWr0= >>

// ----------- >>

} /\*

***GENERISANI KOD KLASE SOFTVER***

\* This code has been generated by the Rebel: a code generator for modern Java.

\*

\* Drop us a line or two at feedback@archetypesoftware.com: we would love to hear from you!

\*/

package Model;

import java.util.\*;

import java.time.\*;

// ----------- << imports@AAAAAAGF34MHbrtL/i8= >>

// ----------- >>

// ----------- << class.annotations@AAAAAAGF34MHbrtL/i8= >>

// ----------- >>

public class Softver {

// ----------- << attribute.annotations@AAAAAAGF34Nit7t2q6Y= >>

// ----------- >>

private String ProvjeraPodataka;

// ----------- << attribute.annotations@AAAAAAGF34O+u7t+7HE= >>

// ----------- >>

private String PrihvacanjePrijave;

// ----------- << attribute.annotations@AAAAAAGF34QJdLuFkvw= >>

// ----------- >>

private String OdbijanjePrijave;

// ----------- << attribute.annotations@AAAAAAGF34RjNbuMjao= >>

// ----------- >>

private String PrihvacanjeRegistracije;

// ----------- << attribute.annotations@AAAAAAGF34g3Qbv5iZo= >>

// ----------- >>

private Softver ;

// ----------- << attribute.annotations@AAAAAAGF34g3Qbv4Fd8= >>

// ----------- >>

private Softver ;

// ----------- << attribute.annotations@AAAAAAGF34svK7zJ7A4= >>

// ----------- >>

private Korisnik ;

// ----------- << attribute.annotations@AAAAAAGF34tSP73btr8= >>

// ----------- >>

private Baza Podataka ;

public String getProvjeraPodataka() {

return ProvjeraPodataka;

}

public String getPrihvacanjePrijave() {

return PrihvacanjePrijave;

}

public String getOdbijanjePrijave() {

return OdbijanjePrijave;

}

public String getPrihvacanjeRegistracije() {

return PrihvacanjeRegistracije;

}

public Softver get() {

return ;

}

public Softver get() {

return ;

}

public Korisnik get() {

return ;

}

public Baza Podataka get() {

return ;

}

public void setProvjeraPodataka(String ProvjeraPodataka) {

this.ProvjeraPodataka = ProvjeraPodataka;

}

public void setPrihvacanjePrijave(String PrihvacanjePrijave) {

this.PrihvacanjePrijave = PrihvacanjePrijave;

}

public void setOdbijanjePrijave(String OdbijanjePrijave) {

this.OdbijanjePrijave = OdbijanjePrijave;

}

public void setPrihvacanjeRegistracije(String PrihvacanjeRegistracije) {

this.PrihvacanjeRegistracije = PrihvacanjeRegistracije;

}

public void set(Softver ) {

this. = ;

}

public void set(Softver ) {

this. = ;

}

public void set(Korisnik ) {

this. = ;

}

public void set(Baza Podataka ) {

this. = ;

}

public void link(Softver \_) {

if (\_ != null) {

\_.unlink();

\_.set(this);

}

unlink();

set(\_);

}

public void link(Softver \_) {

if (\_ != null) {

\_.unlink();

\_.set(this);

}

unlink();

set(\_);

}

public void link(Korisnik \_) {

if (\_ != null) {

\_.unlink();

\_.set(this);

}

unlink();

set(\_);

}

public void link(Baza Podataka \_) {

if (\_ != null) {

\_.unlink();

\_.set(this);

}

unlink();

set(\_);

}

public void unlink() {

if (get() != null) {

get().set(null);

set(null);

}

}

public void unlink() {

if (get() != null) {

get().set(null);

set(null);

}

}

public void unlink() {

if (get() != null) {

get().set(null);

set(null);

}

}

public void unlink() {

if (get() != null) {

get().set(null);

set(null);

}

}

// ----------- << class.extras@AAAAAAGF34MHbrtL/i8= >>

// ----------- >>

}