Implementacija 3-tier arhitekture u PHP tehnologiji

Jelena Ćosović
Elektrotehnički fakultet
Istočno Sarajevo, Bosna i Hercegovina
e-mail: jelena-cosa@hotmail.com

Sadržaj—**Razvoj Web aplikacija danas predstavlja jedan od najunosnijih i najpopularnijih poslova u IT sektoru. Na tržištu se danas nalazi veliki broj tehnologija kojim je programerima omogućeno kreiranje Web aplikacija. Ovaj rad je posvećen jednoj tehnologiji, PHP-u. Kako posao razvijanja aplikacije nije jednostavan posao, posebno ako je riječ o složenim sistemima, pa se u radu posebna pažnja posvećuje implementaciji 3-tier arhitekture, kojom se programerima olakšava posao ukoliko primijene određena pravila koje zahtijeva ova arhitektura. U radu će biti dati primjeri praktične realizacije Web aplikacije za informacioni sistem fotografske radnje.**

Ključne riječi-PHP, 3-tier arhitektura

I. UVOD

Današnje vrijeme ubrzanog razvoja Internet tehnologija i tehnike uopšte nameće drugačije načine poslovanja kompanijama, organizacijama i pojedincima koji žele da povećaju svoj udio na tržištu i reklamiraju i popularizuju svoje usluge i proizvode. Najefikasniji pristup je svakako izrada Web stranica upotrebom različitih tehnologija koje su danas dostupne. Na samom početku, potrebno je postaviti ciljeve, analizirati sve dostupne podatke, naći odgovarajuću strategiju, kako željeni rezultat na kraju ne bi izostao. Nakon ovih koraka, prelazi se na implementaciju, tj. izradu Internet prezentacije. Da bi se napravilo kvalitetno rješenje, mora se poznavati tehnologija i alati koji se koriste za izradu.

U ovom radu predstavljeni su tehnički aspekti izrade Web aplikacije, tj. Internet softvera za informacione sisteme upotrebom PHP-a (*eng*. Hypertext Preprocessor) kao jedne od najpopularnijih tehnologija za izradu Internet softvera. Posebna pažnja je posvećena implementaciji 3-tier arhitekture u PHP-u, sa detaljno objašnjenim zadacima sva tri dijela ove arhitekture.

II. N-TIER I 3-TIER ARHITEKTURA

U softverskom inženjerstvu, N-tier ili multi-tier arhitektura je klijent-server arhitektura u kojoj su funkcije za prikaz, za obradu i upravljanje podacima logički odvojene. Od svih multi-tier arhitektura, najviše se koristi 3-tier arhitektura.[1]

N-Tier arhitektura ima daleko više nivoa, ali je arhitektura obične *desktop* i Web aplikacije svedena na osnovna tri – nivo podataka (*eng.* data layer), logika (*eng*. bussines logic) i prezentacioni nivo (*eng*. presentation layer). Koncept se zasniva na tome da svaki nivo radi svoj posao, da je svaki nivo "učauren" i da svaki od njih može biti korišten sa više strana (npr. istu logiku i pristup podacima mogu da koriste Web i Windows aplikacija) što omogućava modularnost. Takođe, pri izvršavanju programa nivoi se ne mogu preskakati (sem u izuzetnim slučajevima) već proces ide korak po korak, kao što je prikazano na sl. 1. [1]

A. Nivo podataka

U ovom nivou se samo određuje pristup podacima i definišu CRUD operacije (*eng*. Create, Retrieve, Update i Delete), bilo da su u pitanju uskladištene (*eng*. stored) procedure ili živi upiti. Konekcija na bazu se takođe vrši iz ovog nivoa. Dakle, bilo šta sto se radi sa bazom podataka se nalazi u ovom sloju.

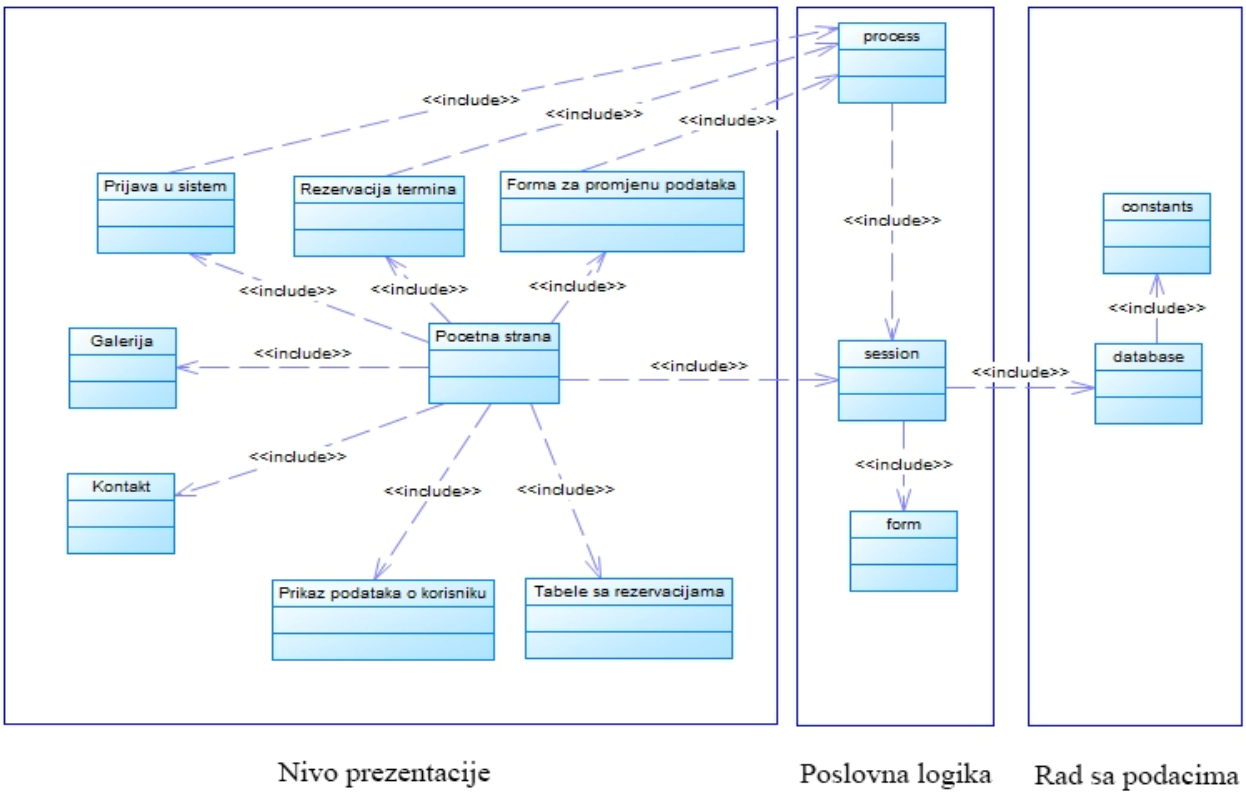
B. Poslovna logika

Ovaj sloj definiše šta se radi sa podacima koji dolaze/odlaze iz aplikacije/baze podataka. Ispituju se pravila upisa, konvertuju se podaci i jednog tipa u drugi i tome sl. Ovdje bi trebala da se nalazi sva "pamet" aplikacije.

C. Prezentacioni nivo

Ovo je zapravo samo korisnički interfejs, tj. prozori i Web stranice. To je najviši nivo arhitekture. On komunicira sa ostalim nivoima tako što šalje podatke ostalim nivoima (tier-ovima) u mreži.

Slika 1. 3-tier arhitektura na primjeru prikaza termina koje je klijent rezervisao



Slika 2. Klasni dijagram za korisnika tipa klijent sa odvojenim nivoima prema 3-tier arhitekturi

III. IMPLEMENTACIJA 3-TIER ARHITEKTURE U PHP TEHNOLOGIJI

Web aplikacija, koju je autor kreirao u praktičnom dijelu rada, predstavlja softver za informacioni sistem fotografske radnje kreiran upotrebom PHP tehnologije. Korisniku aplikacije, odnosno klijentu je omogućeno da manipuliše podacima u bazi podataka, tj. da unosi nove zapise, mijenja postojeće zapise kao i da ima uvid u postojeće zapise u bazi podataka. Aplikacija je krerana upotrebom 3-tier arhitekture, pa su u nastavku predstavljeni svi dijelovi aplikacije, prema 3-tier arhitekturi.

Kao tehnologija, PHP ne pruža različite klase za svaki nivo 3-tier aruhitekture, već je zadatak logičkog razdvajanja ostavljen programeru. Tako, u praktičnom dijelu rada, nivoi su logički odvojeni upotrebom klasa, kao što je prikazano na sl. 2, što će u nastavku biti detaljnije objašnjeno.

A. Nivo podataka

Nivo podataka u okviru 3-tier arhitekture je predstavljen samim podacima koji se nalaze u bazi podataka i obavljanjem CRUD operacija nad tim podacima. Unutar realizovane Web aplikacije nalaze se dvije klase, koji zajedno predstavljaju nivo podataka, tj. *constants* and *database* klase (sl. 2).

1) Baza podataka

Prilikom realizacije informacionog sistema za fotografsku radnju, korištena je MySQL baza podataka, pa će u nastavku biti riječi o ovoj vrsti baze podataka.

MySQL je danas jedan od najpopularnijih sistema za upravljanje bazom podataka. Razvijen je sredinom ’90-ih godina i predstavlja razvijenu tehnologiju koja opslužuje većinu današnjih Internet destinacija. Jedan od razloga za toliki uspjeh je činjenica da je, kao i PHP, ova tehnologija besplatna. Pored toga, veoma je snažna i brza i može se pokrenuti na velikom broju platformi.[2]

Baza podataka predstavlja strukturiranu kolekciju zapisa ili podataka u računarskom sistemu koja je organizovana na takav način da se može jednostavno pretraživati i da se informacije brzo dobiju.[2]

MySQL baza podataka sadrži jednu ili više tabela, od kojih svaka sadrži tzv. zapise ili redove. Unutar ovih redova nalaze se različite kolone ili polja koja sadrže same podatke. Na sl. 3. prikazan je primjer jedne tabele.

ime	prezime	kor_ime	lozinka	kor_id
admin	admin	admin	21232f297a57a5a743894a0e4a801fc3	681582d344cd50c802f8eb7f59c2f961
Jelena	Cosovic	jelena	652b2e2b7133229a89650de27ad7fc41	01063d108710121f55021d026535da5e
Marko	Markovic	marko	c28aa76990994587b0e907683792297c	6b190e3974e7c2569084b2147856e32a
Nikola	Peric	nikola	9365ea12b2d910e1aceaac190fbc97a5	556b63fae7af07c0c3a7a3669fccb742
Nikola	Nikolic	nikolaa	bad72a09881f1122ffa670fe0ef2794c	*NULL*
Verica	Jankovic	verica	497578480d35ff207fddf3f8baf38a35	0beacd186301e1f8046b8d6edbb5f66e
Vladimir	Cosovic	vlado	aa9a5c43ae288f76e3a54b249dab4fc5	9cf17e45dba57cfa95969062e13ff887

Slika 3. Tabela “Korisnici” u bazi podataka

2) Povezivanje sa bazom podataka

Većina Web stranica koja se razvija pomoću PHP-a sadrži više datoteka koje se povezuju sa MySQL bazom podataka i zato su im potrebne informacije o korisničkom imenu i lozinki za pristup bazi. Zbog toga se najčešće kreira zasebna datoteka u kojoj su smješteni ovi podaci i po potrebi još neki dodatni. U informacionom sistemu fotografske radnje ta datoteka se naziva *constants.php*. Kada se kreira ova datoteka, ona se uključuje unutar bilo koje druge PHP datoteke kojoj je potreban pristup bazi podataka.

3) Kreiranje i izvršavanje SQL upita

Direktna komunikacija sa podacima i manipulacija tim podacima je smještena u klasi *database*, koja, kao što je prethodno rečeno, predstavlja drugu klasu unutar nivoa podataka.

Slanje upita MySQL-u iz PHP programa je veoma jednostavno i obavlja se upotrebom funkcije `mysql_query()`:

```
<?php
$query = "SELECT ime,prezime,broj_rezervacija FROM
korisnici WHERE kor_ime = '$username'";
$red= mysql_query($query);
?>
```

Varijabla `$query` određuje upit koji se treba izvršiti nad bazom podataka. U ovom slučaju se uzimaju određeni podaci iz tabele "korisnici". Funkcije `mysql_query()` vraća rezultat upita, koji se smješta u varijablu `$red`.

4) Manipulacija rezultatima SQL upita

Nakon što se izvrši SQL upit, rezultati formiraju tzv. resurs (*eng*. resource) koji sadrži sve podatke koji zadovoljaju kriterijum upita. Dobijanje podataka iz ovog resursa se može izvršiti upotrebom nekoliko PHP funkcija. U nastavku biće riječi o `mysql_fetch_array()` funkciji, koja je načešće korištena u izradi praktičnog dijela rada.

Funkcija `mysql_fetch_array()` služi za preuzimanje jednog reda iz resursa sa rezultatima upita, kojeg pretvara u asocijativni ili numerički niz. [3]

U prethodnom primjeru prikazano je kako se u varijablu `$red` smješta resurs sa rezultatima koji zadovoljavaju uslove upita. Preuzimanje podataka iz ove varijable se obavlja na sljedeći način:

```
$dbarray = mysql_fetch_array($red);
$ime = $dbarray['ime'];
$prezime = $dbarray['prezime'];
$broj = $dbarray['broj_rezervacija'];
```

Ovdje vidimo da se prvo kreira niz `$dbarray` u koji se smješta red podataka iz dobijenog resursa. Dalje preuzimanje podataka se vrši upotrebom ovog niza. Tako se u ovom primjeru dobijaju podaci u vezi sa imenom, prezimenom i brojem rezervacija određenog korisnika u sistemu, smješteni u varijablama `$ime`, `$prezime` i `$broj`, respektivno. Ove varijable će se kasnije proslijediti prezentacionom nivou, koji će ih prikazati na odgovarajući način.

B. Poslovna logika

Poslovna logika unutar 3-tier arhitekture se ponaša kao srednji sloj između zahtjeva koji dolaze od Web pretraživača ili drugih HTTP klijenata i baza podataka na HTTP serverima. Kada se klijent poveže sa serverom i kreira HTTP zahtjev, taj zahtjev može biti različitog tipa tj. može biti u različitim metodama, a najčešće se radi i GET i POST metodama. [2]

Unutar realizovane Web aplikacije nalaze se dvije klase, koji zajedno predstavljaju poslovnu logiku aplikacije, tj. *process* and *session* klase (sl. 2).

```
function procFotoNarudzba() {

global $session, $form;

$retval=$session->fotoNarudzba($_POST['datum'],
$_POST['vrijeme'], $_POST['trajanje'], $_POST['adresa'],
$_POST['komentar']);
…
}
```

Na ovom primjeru prikazan je prvi dio zadatka koji obavlja *process* klasa. Naime, ona preuzima podatke koje je korisnik unio i koji pristižu u HTTP zahtjevu, i to POST metodom. Preuzimanje se vrši navođenjem odgovarajućeg naziva podatka unutar zagrada, npr. datum, vrijeme itd.. Nakon toga, te podatke prosljeđuje *session* klasi, koja te podatke dalje obrađuje.

Session klasa preuzima proslijeđene parametre iz *process* klase i obavlja zadatke sljedećim redoslijedom:

1. provjerava li je korisnik unio sve potrebne podatke
2. provjerava da li su ti podaci ispravno uneseni.
3. ukoliko su podaci ispravno uneseni, prosljeđuje ih nivou za podatke, za dodatne provjere ispravnosti podataka.
4. ukoliko su podaci ispravni, ponovo ih prosljeđuje nivou za podatke, kako bi se izvršile CRUD operacije nad bazom podataka.
5. provjerava rezultat CRUD operacija, na osnovu kojih kreira rezultat koji vraća *process* klasi.

Funkcija `fotoNarudzba()` u klasi *session*, koja se poziva u klasi *process* iz prethodnog primjera, obavlja sljedeće:

```
function fotoNarudzba($datum, $vrijeme, $trajanje,
$adresa, $komentar){
…
$field = "vrijeme";
if(!$vrijeme){
   $form->setError($field, "* Nedostaje vrijeme");
}
else{
   $vrijeme = stripslashes($vrijeme);
   if($vrijeme < 0 || $vrijeme > 24){
       $form->setError($field, "* Pogresno vrijeme");
   }
…
if($form->num_errors > 0){
      return 1;
}
```

Ovdje se vidi da se prvo vrši provjera postojanja podataka u formi, nakon čega se vrši provjera ispravnosti unosa tih podataka. Ukoliko postoji greška, funkcija vraća vrijednost 1, koja će se kasnije tumačiti u *process* klasi.

Primjer provjere ispravnosti podataka u bazi je sljedeći, gdje se provjerava datum fotografisanja, koje mora biti zakazano namjenje tri dana unaprijed:

```
function fotoNarudzba($datum, $vrijeme, $trajanje, $adresa, $komentar){
…
if($database->provjera($datum, $vrijeme, $trajanje) && $GLOBALS['greska']==2){
   $field = "datum";
   $form->setError($field, "* Rezervacija je moguca najmanje tri dana od danasnjeg datuma.");
   return 1;
   } …
}
```

Ukoliko su svi podaci ispravni i nakon provjere u bazi podataka, *session* klasa prosljeđuje sve unesene podatke nivou za podatke, koji obavlja odgovarajuće CRUD operacije nad bazom podataka. Posljednji zadatak *session* klase je tumačenje rezultata tih CRUD operacija, na osnovu kojih kreira rezultat koji vraća *process* klasi:

```
function fotoNarudzba ($datum, $vrijeme, $trajanje, $adresa, $komentar){
…
if($database->fotoNarudzba ($this->kor_ime, $datum, $vrijeme, $trajanje, $adresa, $komentar)){
     return 0;
   }else
   {
    return 2;
   } …
}
```

U dijelu o *process* klasi bio je predstavljen samo prvi od ukupno dva osnovna zadatka koje ta klasa obavlja. Drugi zadatak ove klase je namjerno izostavljen, jer on služi za tumačenje rezultata koje dobije od *session* klase. Na osnovu tih rezultata se poziva odgovarajući prezentacioni nivo, tj. kreira se odgovarajući izgled Web stranice koja se prikazuje korisniku:

```
function procFotoNarudzba() {
…
if($retval == 0){
   $_SESSION['regsuccess'] = true;
   header("location: ../index.php?strana=foto_narudzba");
}
else if($retval == 1){
   $_SESSION['value_array'] = $_POST;
   $_SESSION['error_array'] = $form->getErrorArray();
   header("location: ../index.php?strana=foto_narudzba");
}
else if($retval == 2){
$_SESSION['regsuccess'] = false;
header("Location: ../index.php?strana=foto_narudzba");
} …
}
```

C. *Prezentacioni nivo*

Prezentacioni nivo je zadužen za kreiranje izgleda Web stranice koja se prikazuje krajnjem korisniku. Sve Web stranice se kreiraju dinamički, upotrebom PHP-a. PHP stranica predstavlja Web stranicu koja sadrži elemente HTML-a, tj. HTML tag-ove. Pored HTML-a, PHP stranica može sadržati i PHP tag-ove koji dozvoljavaju serveru da ubaci dinamički sadržaj u stranicu. PHP tag-ovi se mogu koristiti u različite svrhe, kao što je povrat informacija iz baze podataka. Kada korisnik zatraži određenu PHP stranicu, server izvršava PHP elemente i spaja rezultate sa statičkim dijelovima stranice i tako dinamički kreiranu stranicu šalje nazad Web pretraživaču. [2]

Kada je riječ o komunikaciji između prezentacionog nivoa i poslovne logike u okviru 3-tier arhitekture, podaci se razmjenjuju u oba smjera. U prvom slučaju, kada se vrši slanje podataka od Web stranice ka poslovnoj logici, kreira se `request` objekat koji sadrži informacije o podacima koje je unio korisnik i dodatne informacije koje se implicitno kreiraju. Ovaj objekat se dalje obrađuje u klasama poslovne logike, kako bi se dobili željeni podaci.

U drugom slučaju, kada se podaci šalju od poslovne logike ka Web stranicama, PHP stranica koristi PHP tag-ove kako bi dobila podatke od poslovne logike. Stranica za kreiranje narudžbe fotografisanja je najbolji primjer upotrebe ove akcije.

Da bi korisnik kreirao jednu takvu narudžbu, potrebno je da popuni formu, kao što je prikazano na slici 4.

Slika 4. Prikaz stranice za unos nove narudžbe za fotografisanje

Kada korisnik unese podatke o fotografisanju, kreira se HTTP zahtjev koji se prosljeđuje klasama poslovne logike, unutar kojeg se nalaze i uneseni podaci.

Kao što je već prethodno objašnjeno, ovi podaci se provjeravaju u poslovnoj logici, i u zavisnosti od povratne vrijednosti koju prezentacioni nivo dobije od nje, kreira se i odgovarajući izgled stranice za krajnjeg korisnika, kao što je prikazano na slikama 5, 6. i 7. Tumačenje povratne informacije koju prezentacioni nivo dobija od poslovne logike, tačnije od *process* klase, obavlja se unutar početnog dijela PHP stranice:

```
if(isset($_SESSION['regsuccess'])){
  if($_SESSION['regsuccess']){
   echo "<div class=\"main grid_17\">
   <h2 class=\"title\">Uspjesna rezervacija</h2>
   <h2 class=\"infopodnaslov\" align=\"center\" >Hvala Vam, <b>".$_SESSION['kor_ime']."</b>, Vasi podaci su pohranjeni u nasoj bazi podataka.<br/> <br/>
    Povratak na <a href=\"index.php\">glavnu stranicu</a>.</h2>";
   }
   else{
```

```
    echo "<div class=\"main grid_17\">
    <h2 class='title'>Rezervacija nije uspjela</h2>";
    echo  "<h2  class=\"infopodnaslov\"  align=\"center\"
>Zao nam je, desila se greska i Vasa rezervacija, se ne
moze izvrsiti.<br/> Molimo pokusajte kasnije. </h2>";
    }
}…
```

Slika 5. Prikaz stranice za slučaj kada nisu uneseni svi podaci

Slika 6. Prikaz stranice za slučaj kada uneseni podaci nisu ispravni

Slika 7. Prikaz stranice u slučaju uspješne rezervacije

IV. ZAKLJUČAK

Danas se može reći da je PHP tehnologija veoma popularna Web tehnologija, a jedan od razloga je i jednostavno korištenje i kreiranje aplikacije. Međutim, takav način rada često dovodi do kreiranja aplikacija čiji je kod veoma složen i komplikovan, i čije tumačenje zahtijeva dosta vremena. Osnovni cilj prikazane aplikacije je bio da se prikaže, kako se razdvajanjem na dijelove sa definisanim zadacima pojednostavljuje izrada Web aplikacije.

U radu je pažnja posvećena realizaciji dijelova 3-tier arhitekture u PHP tehnologiji u Web aplikaciji za fotografsku radnju. Kada je riječ o nivou za podatke, prikazano je kako se, primjenom klasa vrši povezivanje sa bazom podataka i obavljanje CRUD operacija nad podacima u bazi. U dijelu za poslovnu logiku, predstavljene su klase čiji je osnovni zadatak da usaglase rad svih dijelova 3-tier arhitekture. PHP stranice su opisane u okviru prezentacionog dijela, gdje je akcenat stavljen na metode koje PHP stranice koriste za preuzimanje i predstavljanje podataka koje dobiju od poslovne logike.

LITERATURA

[1] Microsoft. "Layered Application Guidelines" Internet: http://msdn.microsoft.com/en-us/library/ee658109.aspx, 2013. god. [29.Jul 2013.].

[2] Robin Nixon, *Learning PHP, MySQL and JavaSript*, vol. 1, Sebastopol, United States: O'Reilly Media, 2009.

[3] W3Schools. "PHP mysql_fetch_array() Function" Internet: http://www.w3schools.com/php/func_mysql_fetch_array.asp, 2013. god. [04. Avg. 2013.]

ABSTRACT

Web application development is today one of the most profitable and popular activities in the IT sector. On the market today there is a large number of technologies which enable developers to create Web applications. This paper is devoted to one of those technologies, called PHP. Development of business applications is not an easy job, especially when it comes to complex systems, and in the paper, special attention is given to the implementation of 3-tier architecture, which simplifies work for developers if they apply certain rules required by this architecture. This paper will give examples of the practical realization of Web application information system of photographic shop.

IMPLEMENTATION OF 3-TIER ARCHITECTURE IN PHP TECHNOLOGY

Jelena Ćosović