

SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 440

**RADNO OKRUŽENJE ZA UPRAVLJANJE
WEB SADRŽAJEM**

Željko Tepšić

Zagreb, siječanj, 2009.

Sadržaj

Sadržaj	4
1. Uvod.....	6
1.1 Sustav za upravljanje sadržajem	7
2. Radno okruženje za upravljanje sadržajem	8
2.1 Koncept okruženja za upravljanje sadržajem.....	8
2.2 Odabir tehnologije.....	9
2.3 Jezgra radnog okruženja za upravljanje web sadržajem	13
2.3.1 Autentifikacija i autorizacija.....	16
2.3.2 Komponente i moduli (tipovi i instance)	19
2.4 Podatkovni model.....	20
3. Upute za korištenje	23
3.1 LAMP instalacija (Apache, PHP i MySQL)	24
3.2 Instalacija i podešavanje radnog okruženja za upravljanje web sadržajem	28
3.3 Korisničke upute	35
3.3.1 Prijava na sustav	36
3.3.2 Postavke web sjedišta	38
3.3.3 Korisnici.....	40
3.3.4 Uloge.....	42
3.3.5 Tipovi komponenata i komponente	44
3.3.6 Tipovi modula i moduli	48
3.3.7 Grupe modula	51
3.3.8 Stranice	53
3.3.9 Izbornici	57
3.3.10 Instalacija paketa.....	60
3.4 Upute za izradu proširenja radnog okruženja za upravljanje web sadržajem	61
3.4.1 Struktura paketa	61
3.4.2 Izrada komponente.....	65
3.4.3 Izrada modula	68
4. Zaključak	73
5. Literatura.....	75

6.	Naslov, sažetak i ključne riječi	76
7.	Title, Summary and keywords	77
8.	Dodatak A – Popis korištenih engleskih riječi i pojmova	77
9.	Dodatak B – Popis popularnih sustava za upravljanje sadržajem	78

1. Uvod

Tim Bernes-Lee je izumio WWW (*engl. World Wide Web*) kao jednostavan medij koji je koristio mogućnosti Interneta za objavljivanje akademskih publikacija. U ranim počecima Interneta web sjedišta su bila vrlo jednostavna, samo tekstualni članci koji se nisu toliko često mijenjali, a korisnik je bio samo pasivni promatrač. Razvijanjem Interneta razvija se i WWW te postaje globalni hipermedijski informacijski sustav.

Godine 2004. pojavljuje se termin „Web 2.0“ koji postavlja korisnika u središte WWW-a. Termin „Web 2.0“ se ne odnosi na novu verziju WWW-a, niti tehnologije, već na promijeni načina na koji korisnici koriste web, a time i načinu izrade web aplikacija. Web 2.0 koncept vodi prema evoluciji web sjedišta kao što su danas web sjedišta društvenih mreža, portali, dijeljene video zapisa, wiki-ji, blogovi. Drugim riječima sada svaki korisnik može samostalno generirati i objavljivati sadržaj bez imalo znanja o programiranju ili HTML jezika.

Možemo reći da je pred – milenijski web karakteriziralo ručno unošenje i izmjena sadržaja, dok poslije – milenijski web karakterizira značajna automatizacija objave sadržaja i informacija.

Mnoge tvrtke se muče sa održavanjem takvih web sjedišta. Često je sadržaj zastario, slaba je kontrola nad dizajnom i navigacijom, te je prisutan nedostatak nadležnosti. Da bi web sjedišta uistinu bila „Web 2.0“ potrebno je posjedovati web aplikaciju za upravljanje sadržajem. Sa programerske strane potrebno je stvoriti takvo radno okruženje koje će se moći vrlo brzo razviti i izbaciti na tržište, čije će proširenje biti lako izvedivo bez promjena na postojećoj infrastrukturi i ponovna iskoristivost koda.

1.1 Sustav za upravljanje sadržajem

CMS (*engl. Content Management System*) je sustav za upravljanje sadržajem. Postoji nekoliko kategorija sustava, no međutim najznačajnija je ona Web CMS-a. U načelu CMS je računalna web aplikacija čija je glavna svrha stvaranje, uređivanje i upravljanje različitim digitalnim medijima i elektroničkim tekstom. CMS sustavi se učestalo koriste za spremanje, kontroliranje i objavljivanje sadržaja, kao što su članci, novosti, multimedija itd.. Sustav omogućuje korisnicima da bez imalo znanja programiranja uspješno koriste navedeni sustav za objavljivanje sadržaja.

Aplikacija CMS-a se nalazi na web poslužitelju, te koristi bazu podataka za spremanje sadržaja. Korisnici koriste aplikaciju preko web preglednika što je velika prednost, jer je CMS aplikaciju moguće koristiti sa praktički bilo kojeg računala.

Značajke CMS-a su:

- Kontinuitet – zadržavanje informacija kroz seriju korisničkih interakcija
- Upravljanje korisnicima – kontroliranje korisnika kroz neki oblik autentifikacije
- Kontrola pristupa – pristup temeljem RBAC(*engl. Role – Based Access Control*). Dozvole se pridružuju ulogama, a uloge entitetima
- Upravljanje proširenjima – lako i jednostavno proširenje sustava sa dodatnim funkcionalnostima
- Izbornici – upravljanje i generiranje izbornika
- Jezici – višejezična web sjedišta

2. Radno okruženje za upravljanje sadržajem

Cilj ovog završnog rada je specificirati i oblikovati programsko okruženje za razvoj dinamičkih web sjedišta, usluga i primjena. Takvo okruženje treba smanjiti neproduktivne aktivnosti vezane uz tipičan razvoj web primjena korisnicima okoline i programerima tako da mogu jednostavno proširiti funkcionalnost okoline, bez ponovnog programiranja iz početka.

U sljedećim poglavljima biti će predstavljene tehnologije, obrasci i paradigme u programskom inženjerstvu koje omogućuju izradu web aplikacije i realizaciju radnog okruženja za upravljanje web sadržajem. Programska implementacija biti će objašnjena idejno, gdje se neće u detalje opisivati implementacija svakog razreda, jer je to preopširna tema.

2.1 Koncept okruženja za upravljanje sadržajem

Web stranice je potrebno ispuniti sadržajem. To možemo ostvariti tako da omogućimo sustavu za upravljanje sadržajem da uključi module, gdje svaki modul može generirati neku vrstu sadržaja. Dio administratorske funkcionalnosti je odrediti gdje će moduli generirati sadržaj i na kojem bloku web stranice. Moduli su zapravo manja proširenja koja obično stvaraju male okvire sa korisnim informacijama.

Kada govorimo o proširenjima sustava za upravljanje sadržajem, tada je modul programski kod. Nema razloga zašto isti modul ne bi mogli iskoristiti više puta na web sjedištu ili čak web stranici. Modulima je moguće dati određene parametre, te ako su parametri različiti tada će i generirani sadržaj biti drugačiji. Jedna od mogućih uloga modula bi bilo prikazivanje liste posljednje objavljenih novosti.

Komponenta kao dio proširenja, predstavlja cjelovitu web aplikaciju. Komponenta obrađuje ulazne podatke korisnika, te isto tako generira veliku količinu sadržaja kao što su članci na portalima i slično. Mogućnosti komponente obično definira problem radi kojeg se razvija komponenta.

2.2 Odabir tehnologije

Radno okruženje za upravljanje web sadržajem temeljiti će se na otvorenom kodu. Smisao ovakvog pristupa leži u činjenici da korištenjem jeftinih i moćnih programskih rješenja, koja su dostupna svima, u kombinaciji sa znanjem programskog inženjerstva, moguće ostvariti vrlo kvalitetnu web aplikaciju.

Za aplikaciju web poslužitelja odabran je Apache HTTP poslužitelj koji je jedan od najpoznatijih i najčešće korištenih HTTP poslužitelja na Linux/Unix sustavima. Apache podržava mnoge značajke, a mnoge od njih su implementirane i prevedene kao moduli koji nasljeđuju funkcionalnost jezgre. Ti moduli su najčešće *server-side* programski jezici (programski jezici koji se izvršavaju na poslužiteljskoj strani) poput PHP-a, obrasci za autentifikaciju itd. Apache se najviše koristi za prezentiranje statičkih i dinamičkih web stranica na WWW(*World Wide Web*).

Radna okolina, odnosno web aplikacija temeljiti će se na PHP(*Hypertext Preprocessor*) programskom jeziku. PHP je široko korišten skriptni jezik opće namjene, koji vrlo dobro odgovara WEB razvoju i može se vrlo lagano kombinirati sa HTML-om. PHP se najčešće pokreće na web poslužitelju, koji koristeći PHP kod generira web stranice. Većina funkcija PHP-a je dizajnirano za Linux sustave. Za perzistenciju podataka koristit ćemo MySQL relacijsku bazu podataka. Ta baza vrlo uspješno funkcionira u kombinaciji sa PHP-om.

Nakon navedenih svih tehnologija koje su nužne za ostvarenje radnog okruženja za upravljanje web sadržajem koristiti ćemo već razvijene programske podrške napisane nad funkcionalnošću PHP-a. Takve programske podrške su poznate pod imenom *framework*, odnosno programska okolina. Framework je programska podrška koja implementira općenito rješenje za općeniti problem.

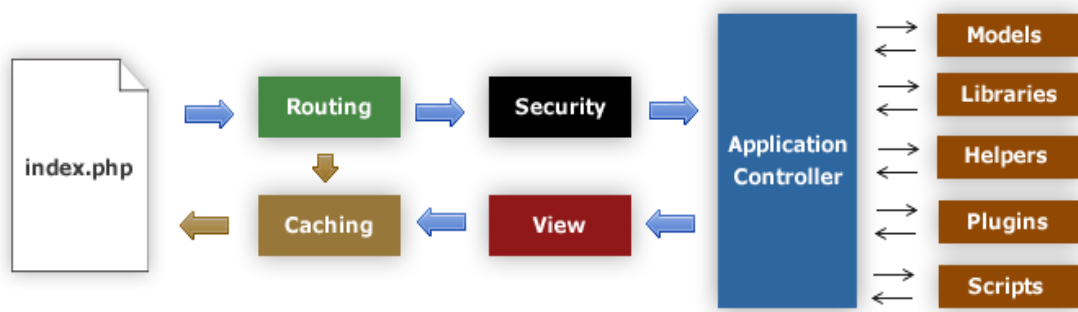
Framework za PHP koji smo koristili je CodeIgniter verzije 1.7. CodeIgniter je moćan PHP *framework* koji je razvijen za PHP programere koji trebaju jednostavan i elegantan alat za stvaranje potpuno funkcionalnih web aplikacija. Omogućuje brzo stvaranje aplikacija pružajući bogate kolekcije knjižnica za često korištene probleme,

ali i jednostavno sučelje i logičnu strukturu za pristup tim knjižnicama. Temelji se na objektno orijentiranoj arhitekturi. Jedna velika prednost ovog *frameworka* je korištenje *MVC (model-view-controller)* obrasca. MVC je arhitekturni obrazac (temelj na *design* obrascu) koji se koristi u programskom inženjerstvu. U kompleksnim računalnim aplikacijama koje korisniku prezentiraju veliku količinu podataka, programeri često žele razdvojiti podatke (*model*) i probleme korisničkih sučelja (*view*), tako da promjene vezane na korisničko sučelje nema utjecaja na upravljanje podacima, te da se podaci mogu prepoznati bez mijenjanja korisničkog sučelja. MVC rješava taj problem tako da razdvaja pristup podacima i poslovnu logiku od prezentacije podataka i korisničke interakcije, uvodeći posredničku komponentu kontroler (*controller*).

- **Model** – Model obuhvaća specifičnu domensku reprezentaciju informacija s kojima aplikacija upravlja. Domenska logika daje značaj neobrađenim podacima (npr. računanje da li je danas nečiji rođendan itd.). Pod model spadaju mehanizmi komuniciranja sa sustavima stalne pohrane podataka (npr. baze podataka).
- **View** – Prevodi model u oblik pogodan za interakciju, tipično element korisničkog sučelja. Za jedan model može postojati nekoliko pogleda, svaki sa određenom svrhom.
- **Controller** – Obrađuje i odgovara na događanje (engl. *events*), tipično korisničke akcije koje mogu uzrokovati promjene na modelu.

Kod web aplikacija, *view* je zapravo HTML stranica, dok je *controller* kôd koji prikuplja dinamičke podatke i generira sadržaj unutar HTML-a. Na kraju, *model* je predstavljen sa aktualnim sadržajem, obično spremljen u bazu podataka ili XML datoteke.

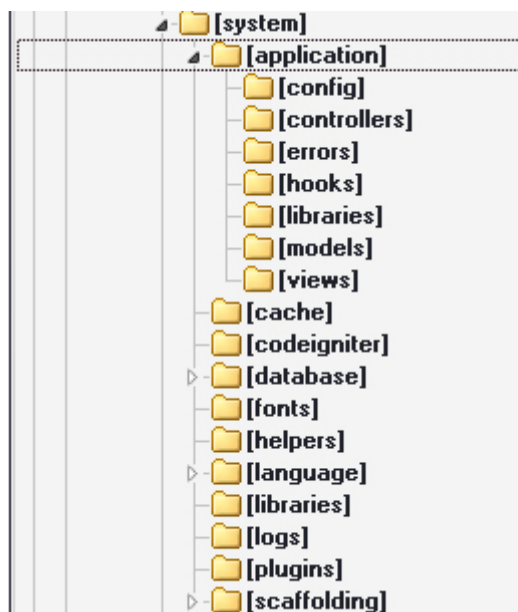
Na slijedećoj slici prikazan je protok podataka u Codeigniter okolini:



Slika 1 - Protok podataka u Codeigniter okolini

- *index.php* je prednji kontroler koji inicijalizira osnovne resurse potrebne za pokretanje CodeIgniter-a,
- *Router* pregledava HTTP zahtjev da bi odredio što treba s njime napraviti.
- Ako postoji *cache* datoteka, ona se direktno šalje Internet pregledniku, zaobilazeći pri tome normalno izvođenje sustava.
- *Sigurnost*. Prije nego se učita aplikacijski kontroler, HTTP zahtjev i svi korisnički podaci koji su predani se filtriraju radi sigurnosti.
- Kontroler učitava model, jezgrine knjižnice, te sve ostale resurse koji su potrebni za procesuiranje specifičnog zahtjeva.
- Finalizirani *view* se generira i šalje se Internet pregledniku. Ako je međuspremnik (*cache*) uključen, *view* se prvo sprema u međuspremnik kako bi se naredni zahtjevi mogli poslužiti.

Struktura direktorija CodeIgniter je sljedeća:



Slika 2 - struktura direktorija CodeIgniter frameworka

Najvažniji je direktorij *application*, odnosno njegov sadržaj. U direktoriju *config* nalaze se konfiguracijske datoteke kao što je datoteka za bazu podataka, te datoteka za općenite postavke CodeIgniter frameworka. Direktoriji *controllers*, *models* i *views* su namijenjeni za MVC. Direktorij *libraries* služi za spremanje knjižnica koje smo sami razvili. Direktorij *hooks* sadrži datoteke koje proširuju funkcionalnosti same jezgre CodeIgnitera bez potrebe za modificiranjem koda jezgre. Direktorij *errors* sadrži datoteke za prikaz i obradu grešaka.

U nastavku ćemo se baviti rješavanjem konkretnog problema u izradi web aplikacije. Dodatne detalje vezane za korištenje frameworka nećemo objašnjavati, osim na mjestima gdje je to potrebno za razumijevanje određenog rješenja.

2.3 Jezgra radnog okruženja za upravljanje web sadržajem

U prijašnjim poglavljima govorili smo o komponentama i modulima, ali za bolje razumijevanje u kratko ćemo ponoviti. Dakle u načelu komponenta je web stranica koja generira glavni sadržaj na stranici ovisno o njezinoj zadaći, a moduli generiraju manji sadržaj koji može i ne mora biti povezan sa sadržajem kojeg generira komponenta i prikazuje se na istoj stranici gdje i sadržaj komponente. Jedna od osnovnih značajki modula je ta da je moguće zadati mjesto na stranici i redoslijed prikazivanja u odnosu na druge module.

Isto tako od sustava bi željeli da su komponente ponovno iskoristive, primjerice da postoje više stranica koje generiraju različiti sadržaj na jednaki način.

U današnje vrijeme veliku važnost prilikom objave sadržaja preuzima optimizacija sadržaja na tražilice. Objavljivanje sadržaja nema nikakav značaj ukoliko nitko ne zna za vas, odnosno ne može vas pronaći putem tražilica. Jednu važnu komponentu u takvom optimiziranju za tražilice čine kvalitetno izrađene poveznice. To zapravo znači da poveznice moraju biti logične i čovjeku razumljive, a to vole tražilice. Primjerice, poveznica <http://www.example.com/file.php?p=1&n=43> ne daje jasnu predodžbu što bi takva stranica prikazivala, a osobito parametri u parametarskom djelu URI-a. Kvalitetna poveznica bi izgledala kao <http://www.example.com/stranica/12/zavrsni-rad>. Iz navedene poveznice čovjeku je jasno što će prikazivati takva stranica, dakle prikazati će stranicu koja će generirati sadržaj o završnom radu.

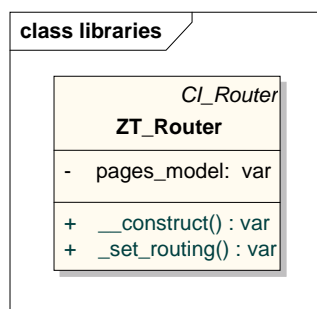
Codeigniter okolina omogućuje takvo kreiranje poveznica, ali poveznice su ograničene nazivima stranica i podstranica, odnosno kontroler razreda i pripadajućih metoda, a to ne želimo. Mi želimo da korisnik može sam definirati naziv stranice, odnosno poveznice.

Implementacija naše komponente lako bi se dala realizirati implementacijom kontrolera iz Codeigniter okoline. Ali kako ćemo znati koji modul se treba prikazati u tom kontroleru i na kojem mjestu, a posebice ako želimo dozvoliti korisniku da definira nazive kontrolera i metoda?

Ono što Codeigniter omogućuje je definiranje obrasca, odnosno regularnog izraza koji kada se prepozna u traženoj poveznici, korisnika preusmjerava na temelju rute na željeni kontroler. Takav način definiranja moguć je jedino ako se ugradi u sam programski kod, a korisnik treba imati programerska znanja da bi to znao promijeniti.

Zbog problema prepoznavanja tražene stranice, odlučili smo se za stvaranje tablice *zt_pages* u bazi podataka u kojoj se nalaze sve stranice sa definiranim regularnim izrazima i rutama. Zbog toga što Codeigniter okolina, odnosno razred *Route* koji usmjerava korisnika na željenu komponentu, ne podržava dohvaćanje podataka iz baze implementiran je vlastiti razred *ZT_Router* koji obavlja sljedeći zadatak:

- Korisnik upiše traženu adresu
- *ZT_Router* dohvati URI i šalje upit bazi podataka: koja stranica ima obrazac koji odgovara traženom URI-u
- Baza vraća odgovor
- Ukoliko postoji n-torka koja odgovara upitu tada se pamti identifikator stranice i preusmjerava korisnika na nadležni kontroler, a ukoliko ne postoji tražena n-torka tada korisniku se pokazuje 404 poruka o nepostojanju tražene stranice.

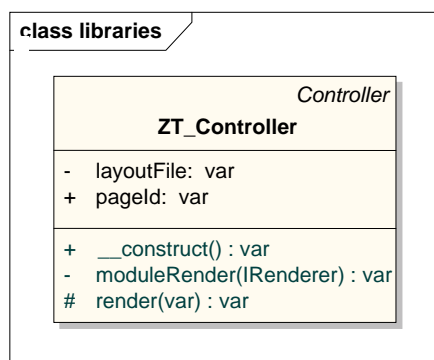


Slika 3 - Class diagram razreda *ZT_Router*

Na ovaj način saznali smo koju točno stranicu korisnik zahtjeva, odnosno dobili smo identifikator stranice preko kojeg ćemo dohvaćati module koji se trebaju prikazati na stranici, te smo dobili korisnički definirane URI-e koji ne moraju točno odgovarati nazivu kontrolera i njegovih metoda.

Sada ostaje problem, kako učitati i prikazati module koji se prikazuju na toj stranici, odnosno kontroleru/komponenti. Odgovor leži u činjenici da PHP može dinamički učitati, odnosno instancirati objekte.

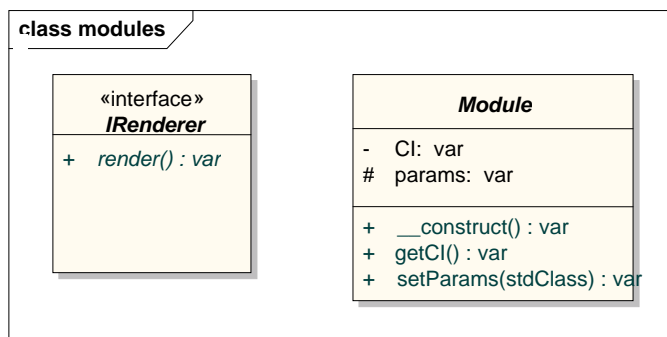
Standardno svaki se kontroler u Codeigniter okolini razvija tako da naslijedi razred *Controller*. Zbog prethodno spomenutog problema sa učitavanjem, odnosno prikazivanjem modula napravljen je *ZT_Controller* razred koji nasljeđuje Codeigniter Controller razred, a pak naš kontroler, odnosno komponenta treba nasljediti *ZT_Controller*.



Slika 4 - Class dijagram *ZT_Controller* razreda

Razred *ZT_Controller* obavlja sve isto kao i standardni Codeigniter Controller ali uz par razlika. Svaka metoda kontrolera, odnosno stranica svoj izgenerirani sadržaj šalje render metodi naslijeđenoj iz *ZT_Controller* razreda. Ta će metoda dohvatiti identifikator stranice iz sjednice, dohvatiti će module koji se trebaju prikazati na stranici, njih će iscrtati zajedno sa sadržajem koji je predan toj metodi.

Da bi se moduli znali iscertati, svaki modul koji se stvara mora naslijediti *Module* razred i implementirati *IRenderer* sučelje.



Slika 5 - Class dijagram Module i IRenderer

2.3.1 Autentifikacija i autorizacija

Svaki sustav za upravljanje web sadržajem mora imati mogućnost prijave korisnika na sustav, odnosno nekakvu vrstu autentifikacije. Kada je korisnik prepoznat potrebno ga je autorizirati, odnosno dodijeliti mu odgovarajuće uloge.

Codeigniter okolina ne dolazi sa implementacijama za autentifikaciju i autorizaciju, stoga je to bilo potrebno posebno razviti.

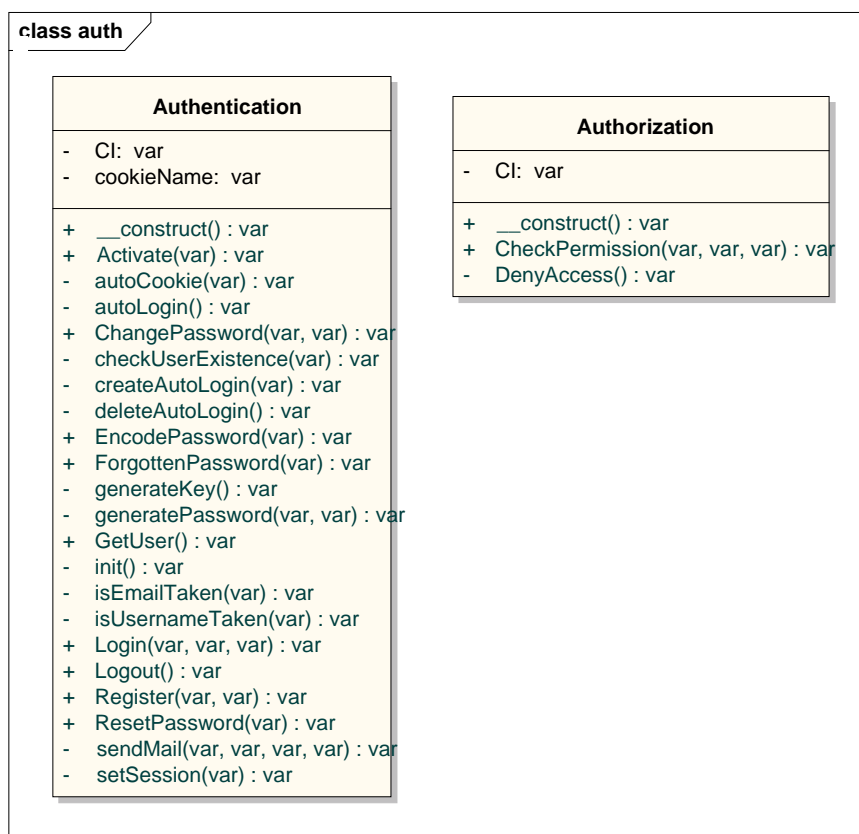
Princip autentifikacije poprilično je jednostavan. Korisnik unosi korisničko ime i lozinku. Lozinka se kodira nekom hash funkcijom i obavlja se upit na bazu sa zadanim podacima. Ukoliko korisnik postoji, identifikator korisnika i korisničko ime zapisuju se u sjednicu. Važno je napomenuti da je sjednica implementacija Codeigniter okoline, a funkcionira tako da se podaci spremaju u bazu podataka, a identifikator sjednice u cookie. Odjava korisnika se obavlja tako da se sjednica uništi. Trajanje sjednice moguće je podesiti, a osnovno trajanje je 2 sata.

Jedno od poželjnih mogućnosti je automatska autentifikacija. Automatska autentifikacija je kada korisnik posjeti stranicu a sustav ga automatski autentificira. Automatska autentifikacija ostvarena je na način da se prilikom prve prijave na sustav ponudi opcija za odabir automatske autentifikacije. Ukoliko korisnik odabere

tu opciju stvara se jedinstveni ključ koji se zapisuje u cookie i u bazu podataka sa identifikacijskim brojem korisnika. Pa kada korisnik posjeti stranicu učitava se njegov cookie, te ako za zadani ključ iz cookie-a se dohvati identifikator korisnika iz baze tada je korisnik autentificiran. Ukoliko se korisnik odjavi briše se automatska autentifikacija.

Razred *Authentication* implementira navedenu autentifikaciju. Od važnijih metoda valja spomenuti:

- Login i Logout – metode za prijavu i odjavu korisnika
- Register – registracija korisnika
- EncodePassword – kodiranje lozinke
- setSession – postavljanje podataka u sjednicu
- autoLogin – automatska prijava



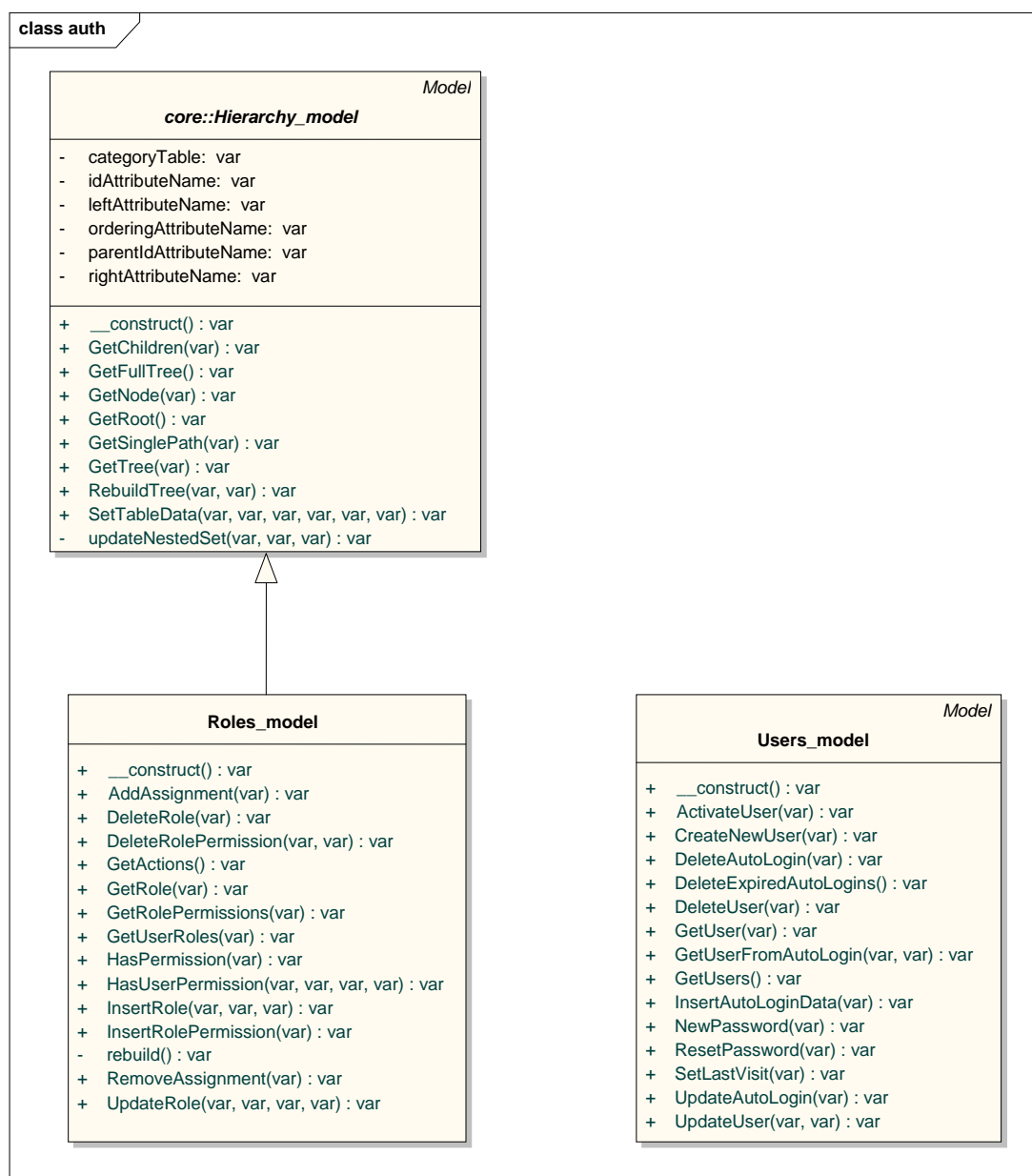
Slika 6 Class dijagram razreda za autentifikaciju i autorizaciju

Autorizacija se zasniva na RBAC (*engl. Role-based access control*), odnosno upravljanje pristupom na načelu uloga. Ukratko da ponovimo, uloge se dodjeljuju entitetima koji zahtijevaju obavljanje neke akcije nad nekim entitetom, a ulogama se dodjeljuju dozvole za obavljanje akcija nad nekim entitetom. Pa ukoliko neki entitet zatraži obavljanje neke akcije nad nekim entitetom, tada mu se to omogućava ukoliko mu je dodijeljena uloga koja ima pravo obavljanja te akcije. Važno je napomenuti da je u implementaciji ove web aplikacije ostvarena hijerarhija uloga, odnosno da nadređena uloga ima sve dozvole za obavljanjem akcija nad entitetima na temelju podređene uloge. Dakle, nije potrebno eksplicitno definirati dozvole nadređenoj ulozi ukoliko te dozvole posjeduje podređena uloga.

Implementacija razreda *Authorization* ostvarena je tako da se metodi *CheckPermission* predaju parametri kao što je tražena akcija i entitet nad kojim se želi obaviti akcija. Zatim metoda dohvaća korisnika preko razreda za autentifikaciju te postavlja upit na bazu. Ukoliko postoji dozvola metoda vraća *true*, inače *false*.

U ovim razredima RBAC se koristi isključivo za dozvole korisnika, ali moguće je koristiti strukturu RBAC-a na bilo koje entitete, odnosno na objekte, tablice ili pojedine zapise u tablicama.

Na sljedećoj slici prikazan je dijagram razreda koji predstavlja model uloga i korisnika.



Slika 7 - Modeli uloga za pristup bazi podataka

Hijerarhija uloga ostvarena je primjenom modela liste susjedstva i ugniježđenih skupova.

2.3.2 Komponente i moduli (tipovi i instance)

Zahtjev koji se postavlja nad radnim okruženjem za upravljanje web sadržajem je mogućnost proširenja. Proširivost je poželjna jer na lak način proširuje postojeću

funkcionalnost bez potrebe za mijenjanjem trenutnog programskog koda, te daje mogućnost ponovne iskoristivosti. Proširivost ponajviše pogoduje programerima jer svoj kod mogu iskoristiti više puta, lako ga je održavati i stvarati novi.

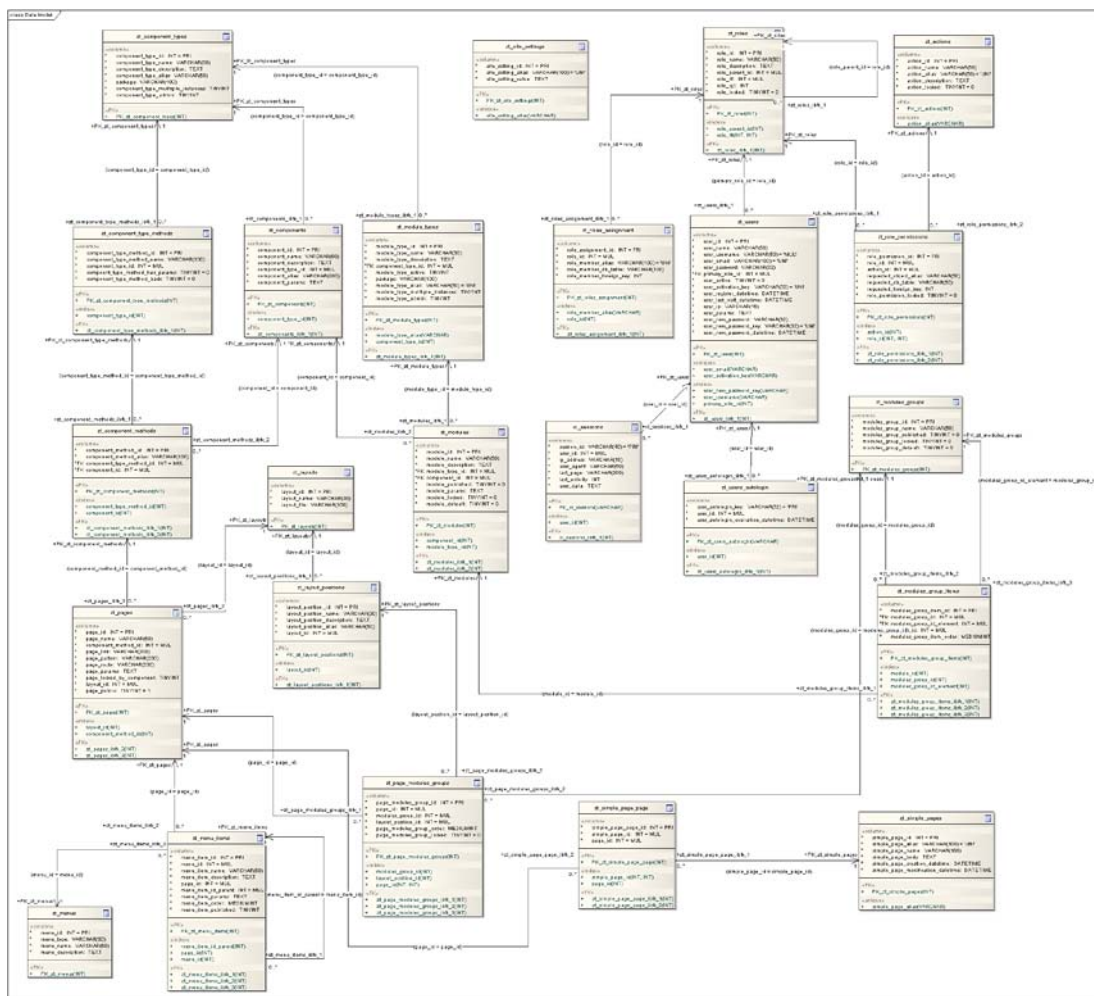
Proširenja su ostvariva preko komponenata i modula, odnosno kontrolera i programskih knjižnica Codeigniter-a. Takav programski kod predstavlja tip komponente ili modula. Tipovi komponenata i modula su zapravo vrlo slični tipovima podataka u programiranju, odnosno razredima. Dakle, moguće je kreirati instancu nekog razreda sa određenim parametrima, te će se upravo zbog tih parametara razlikovati od druge instance toga tipa. Konkretno na našem primjeru sustava za upravljanje sadržajem, to bi značilo da je moguće kreirati više modula iz jednog tipa modula. Primjerice, imamo definiran modul koji je implementacija izbornika/navigacije web stranice, a željeli bi imati više različitih izbornika na stranici. Pa upravo zbog tipova modula moguće je kreirati više izbornika gdje će svaki sadržavati drugačije poveznice. Jednako vrijedi i za komponente.

Takav problem više je problem organizacije i interpretacije podataka u bazi podataka, nego izrade kontrolera ili knjižnica. Kontroleri i knjižnice obrađuju dane podatke na jednaki način, jedino za što se moramo pobrinuti su podaci kakve će taj programski kod dobiti.

Više o načinu izrade komponenta i modula u narednim poglavljima.

2.4 Podatkovni model

Zbog kompleksnosti baze podataka, koja broji preko dvadesetak tablica, prikazanu sliku podatkovnog modela teško je očitati, stoga je slika velike rezolucije priložena u ovaj završni rad, te se nalazi na CD-u.



Slika 8 - Prikaz podatkovnog modela

Ovdje ćemo samo nabrojati tablice i opisati njihovu namjenu:

- **ci_sessions** – sadrži podatke o sjednici
- **zt_actions** – sadrži akcije koje se mogu obaviti nad nekim entitetima, npr. čitanje, stvaranje, uređivanje i brisanje
- **zt_component_methods** – sadrži preimenovane nazive metoda komponenata
- **zt_component_type_methods** – sadrži originalne nazive metoda komponenata
- **zt_component_types** – sadrži tipove komponenata
- **zt_components** – sadrži instance komponenata
- **zt_layout_positions** – sadrži pozicije u grafičkom rasporedu stranice
- **zt_layouts** – sadrži grafičke rasporede stranice
- **zt_menu_items** – sadrži elemente izbornika
- **zt_menus** – sadrži izbornike

- `zt_module_types` – sadrži tipove modula
- `zt_modules` – sadrži instance modula
- `zt_modules_group_items` – sadrži elemente grupe modula
- `zt_modules_groups` – sadrži grupe modula
- `zt_page_modules_groups` –sadrži grupe modula koje se prikazuju na stranici
- `zt_pages` – sadrži popis svih stranica sustava
- `zt_role_permissions` – sadrži dozvole za uloge
- `zt_roles` – sadrži uloge
- `zt_roles_assignment` – sadrži entitete kojima su dodijeljene uloge
- `zt_site_settings` – sadrži postavke stranice
- `zt_users` – sadrži korisnike
- `zt_users_autologin` – sadrži informacije o automatskoj prijavi korisnika

3. Upute za korištenje

U slijedećim podnaslovima govoriti ćemo o pripremi okruženja na kojemu će se pokretati sustav za upravljanje sadržajem, odnosno programsko okruženje za razvoj dinamičkih web sjedišta, usluga i primjena. Okruženje obuhvaća instalaciju web poslužitelja *Apache* sa podrškom za programski jezik *PHP*, te pripadajućom bazom podataka *MySQL*. Taj cijeli paket je poznat pod nazivom *LAMPP*, namijenjen za *unix/linux* sustave. Sličan paket postoji i za *windows* operacijske sustave, te je i sam postupak instalacije sličan. No međutim mi ćemo u nastavku opisati instalaciju za *unix/linux* operacijske sustave.

Isto tako biti će objašnjena instalacija programskog okruženja za razvoj dinamičnih web sjedišta, kao i upute za korištenje navedene web aplikacije od strane korisnika sa minimalnim tehničkim znanjem, kao i upute programerima za razvoj programskih paketa čija je svrha proširenje funkcionalnosti sustava uz poznavanje *Codeigniter* razvojne PHP okoline.

U slijedećoj tablici su prikazane preporučene tehničke značajke sustava koji će moći pokretati programsko okruženje za razvoj dinamičkih web sjedišta:

Tabela 1 - Preporučene tehničke značajke sustava

RAČUNALO	<ul style="list-style-type: none">• 1 GHz 32-bit (x86)• 1 GB RAM• 20 GB disk• Ethernet kartica• Pristup Internet-u
OPERACIJSKI SUSTAV	<ul style="list-style-type: none">• Linux operacijski sustav, kernel verzije 2.6 na više,• Windows operacijski sustav, od XP verzije na više
HTTP POSLUŽITELJ	<ul style="list-style-type: none">• Apache http poslužitelj sa podrškom za

	programski jezik PHP, od verzije 2.0 na više
PODRŠKA ZA PROGRAMSKI JEZIK	<ul style="list-style-type: none">• PHP programski jezik, verzije 5.0 na više
BAZA PODATAKA	<ul style="list-style-type: none">• MySql baza podataka, verzije 5.0 na više sa podrškom za InnoDB
INTERNET PREGLEDNICI	<ul style="list-style-type: none">• Preglednici sa mogućnošću izvršavanja JavaScript-a• Opera, verzije 9.0 na više• Mozilla Firefox, verzije 2.0 na više• Internet Explorer, verzije 7.0 na više• Google Chrome, verzije 1.0 na više

3.1 LAMP instalacija (Apache, PHP i MySql)

U ovome poglavlju objasniti ćemo instalaciju *LAMPP* paketa koji obuhvaća *Apache*, *PHP* i *MySql* poslužitelje, za *unix/linux* operacijske sustave. Instalacije operacijskog sustava *unix/linux*, te podizanje i podešavanje sigurnosnih stijena (*engl. firewall*) i sličnih mehanizama zaštite poslužitelja neće biti objašnjene. Sigurnosna ostvarenja opisana u ovome dokumentu su ona najosnovnija, te se preporuča da kasnije poboljšate sigurnosne postavke. Preporuča se da korisnik prilikom instalacije navedenog paketa bude prijavljen u sustavu kao super korisnik.

Programski paket *LAMPP* omogućuje jednostavnu instalaciju *Apache* web poslužitelja, *PHP* poslužitelja, te *MySql* baze podataka. Prije same instalacije *LAMPP* paketa potrebno je paket preuzeti sa web adrese <http://www.apachefriends.org/en/xampp-linux.html>.

Preuzeti paket zapakiran je u obliku tar.gz arhive, koju je potrebno otpakirati. Otpakiravanje se obavlja izvršavanjem sljedeće naredbe:

```
# tar -xzf xampp-linux-1.6.5a.tar.gz -C /opt
```

I to je sve, program je instaliran i nalazi se u direktoriju `/opt/lampp`. Prije nego li pokrenemo *Apache* poslužitelj potrebno je provjeriti da li će se kod pokretanja *Apache* poslužitelja učitati modul *mod_rewrite* koji je potreban da bi programsko okruženje za razvoj dinamičkih web sjedišta, usluga i primjena moglo ispravno raditi.

To ćemo napraviti tako da se u nekom uređivaču teksta otvori datoteka `/opt/lampp/etc/httpd.conf`. Nakon toga je potrebno pronaći slijedeću liniju:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

Ukoliko se na početku navedene linije nalazi `#` znak, tada je potrebno obrisati taj znak. Ukoliko znaka nema to znači da će se modul učitati prilikom pokretanja *Apache* poslužitelja.

Web poslužitelj pokreće se naredbom:

```
# /opt/lampp/lampp start
```

I nakon toga dobije se ispis poput ovog:

```
Starting XAMPP 1.6.5a...
LAMPP: Starting Apache...
LAMPP: Starting MySQL...
LAMPP started.
```

Također potrebno je podesiti lozinke za pristup *Apache* poslužitelju, a to se radi tako da se pokrene slijedeća naredba:

```
# /opt/lampp/lampp security
```

I nakon toga dobije se slijedeći ispis:

```
XAMPP: Quick security check...
XAMPP: Your XAMPP pages are NOT secured by a
password.
XAMPP: Do you want to set a password? [yes]
```

Ovdje je potrebno pritisnuti tipku ENTER, te nakon toga unijeti lozinku i ponoviti ju još jednom, a ispis će biti sličan ovome:

```
XAMPP: Password:
XAMPP: Password (again):
XAMPP: Password protection active. Please use
'lampp' as user name!
```

Nakon toga dobiva se ispis da nije podešena lozinka za *MySQL* bazu podataka. Ispis je sličan ovome:

```
XAMPP: MySQL is accessible via network.
XAMPP: Normally that's not recommended. Do you want
me to turn it off? [yes]
XAMPP: The MySQL/phpMyAdmin user pma has no password
set!!!
XAMPP: Do you want to set a password? [yes]
XAMPP: Password:
XAMPP: Password (again):
XAMPP: Setting new MySQL pma password.
```

```
XAMPP: Setting phpMyAdmin's pma password to the new
one.
```

```
XAMPP: MySQL has no root password set!!!
```

```
XAMPP: Do you want to set a password? [yes]
```

```
XAMPP: Write the password somewhere down to make
sure you won't forget it!!!
```

```
XAMPP: Password:
```

```
XAMPP: Password (again):
```

```
XAMPP: Setting new MySQL root password.
```

```
XAMPP: Setting phpMyAdmin's root password to the new
one.
```

Na kraju još treba podesiti lozinku za FTP. Ispis je sličan ovome:

```
XAMPP: The FTP password is still set to 'lampp'.
```

```
XAMPP: Do you want to change the password? [yes]
```

```
XAMPP: Password:
```

```
XAMPP: Password (again):
```

```
XAMPP: Reload ProFTPD...
```

```
XAMPP: Done.
```

LAMPP sustav se gasi naredbom:

```
# /opt/lampp/lampp stop
```

Poželjno je da se program `lampp` pokreće prilikom pokretanja samog sustava, a to se radi na sljedeći način:

1. Odrediti koji je vaš podrazumijevani *runlevel*, sa naredbom

```
# egrep :initdefault: /etc/inittab
```

U najviše slučajeva to je broj 3 ili 5.

2. Otići u direktorij u kojem se podešavaju opcije za podrazumijevani *runlevel*, dakle u našem slučaju je to direktorij `/etc/init.d/rc3.d`
3. Otipkati sljedeće naredbe u Linux ljusci:

```
# ln -s /opt/lampp/lampp S99lampp
# ln -s /opt/lampp/lampp K01lampp
```

Ovime završava instalacija i podešavanje *LAMPP* paketa.

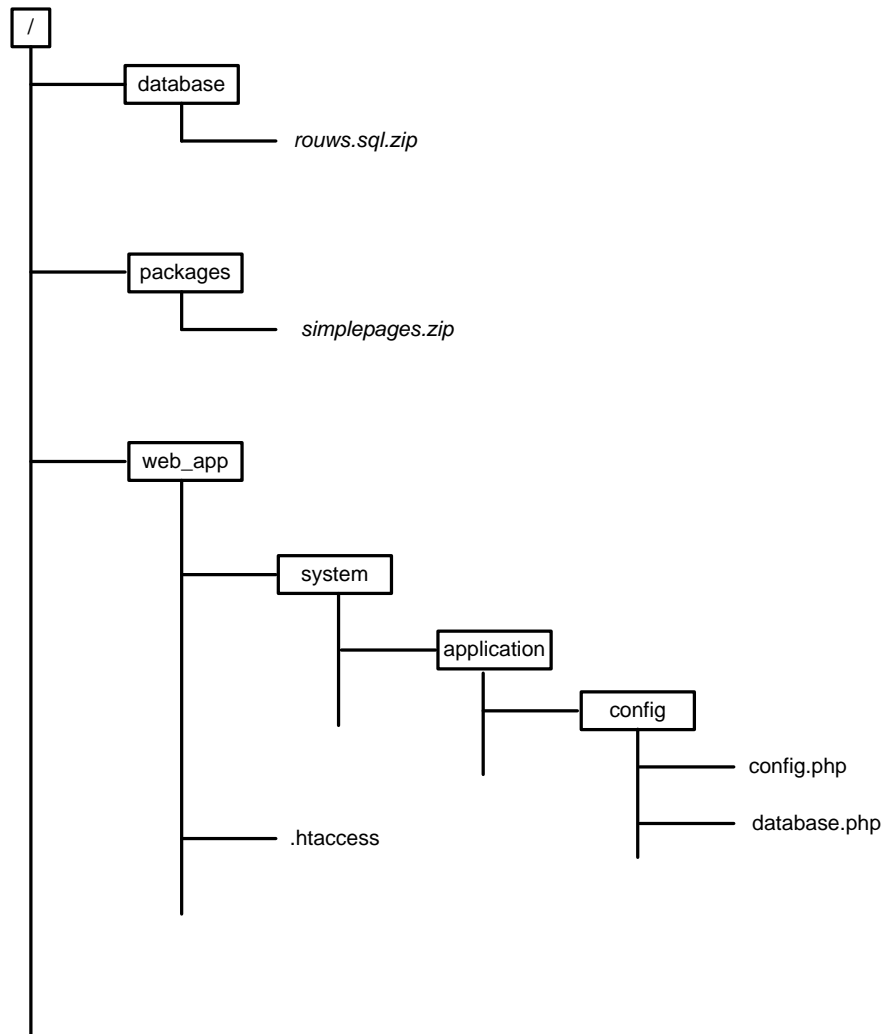
3.2 Instalacija i podešavanje radnog okruženja za upravljanje web sadržajem

U ovome poglavlju biti će objašnjena instalacija i podešavanje radnog okruženja za upravljanje web sadržajem, na temelju prethodno instaliranog *LAMPP* paketa. Postupak instalacije i podešavanja radnog okruženja za upravljanje web sadržajem biti će podijeljen na nekoliko koraka.

Korak 1.

Sve datoteke aplikacije „radno okruženje za upravljanje web sadržajem“ nalaze se u datoteci *rouws.zip* koja je priložena uz ovaj završni rad.

Na slijedećoj slici prikazana je struktura direktorija nakon raspakiravanja *rouws.zip* datoteke, te datoteke koje će biti potrebno urediti.



Slika 9- Struktura direktorija i datoteka za podešavanje aplikacije

Nakon raspakiravanja *rouws.zip* datoteke dobivamo tri direktorija:

- **database** – direktorij sadrži *rouws.sql.zip* datoteku koja sadrži prethodno pripremljenu bazu podataka koju je potrebno instalirati da bi radna okolina za upravljanjem sadržaja mogla funkcionirati. Kako kreirati bazu i instalirati pripremljenu bazu u koraku 2.
- **packages** – direktorij sadrži *simplepages.zip* arhivu koja sadrži paket proširenja za radnu okolinu za upravljanjem sadržaja. U načelu radi se o komponenti i pridruženom modulu koji omogućuju jednostavno kreiranje i objavljivanje sadržaja, odnosno web stranica. Kako samostalno izraditi paket proširenja biti će objašnjeno u poglavlju 3.4.

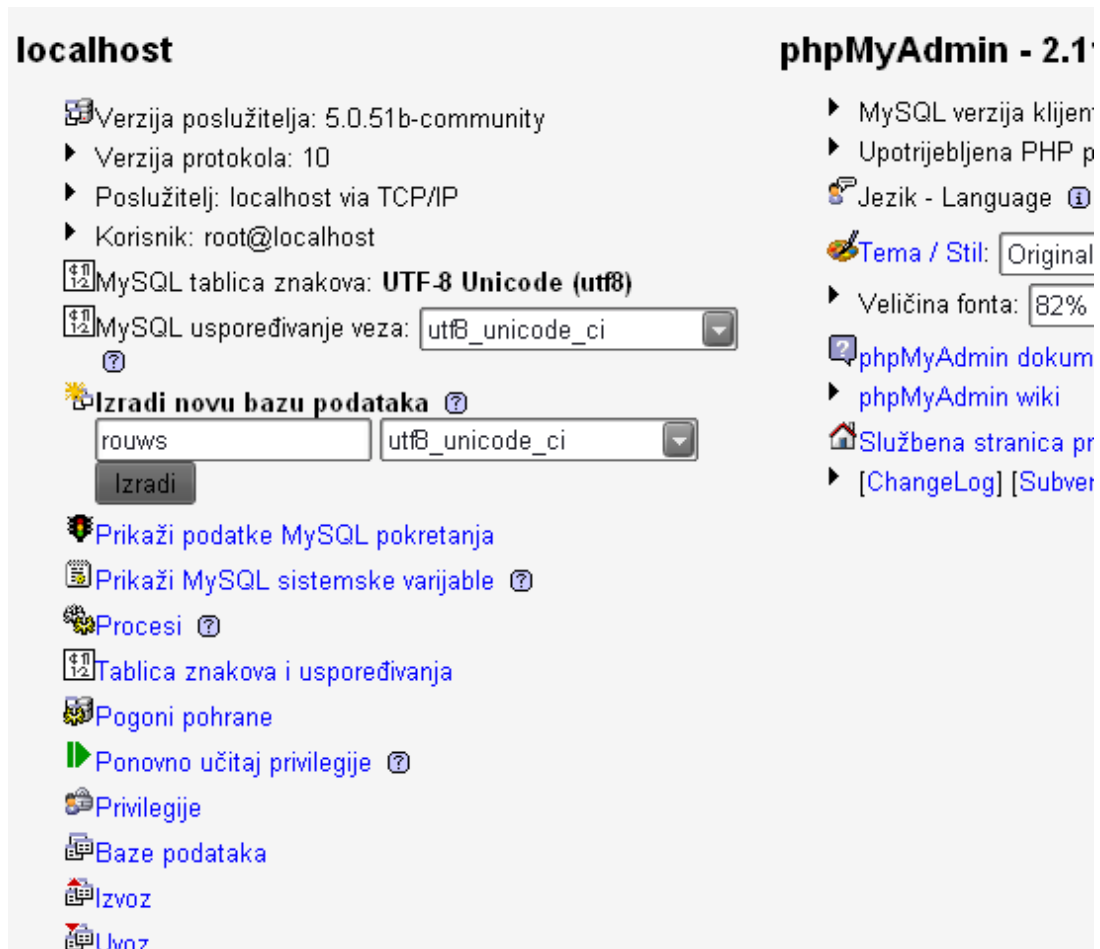
- **web_app** – direktorij sadrži aplikaciju za upravljanje web sadržajem. Struktura direktorija unutar web_app direktorija je standardna struktura *Codeigniter PHP* okoline. Posebno su naznačene datoteke koje će biti potrebno urediti radi podešavanja radne okoline za uređivanje sadržaja. U koraku 3. biti će objašnjeno što treba napraviti sa navedenim direktorijem.

Korak 2.

Da bi radno okruženje za upravljanje web sadržajem funkcioniralo, potrebno je prirediti bazu podataka kako bi se radno okruženje uspješno instaliralo. Sve što je potrebo je stvoriti bazu podataka, u koju će se odpakirati prethodno pripremljena baza podataka sa osnovnim postavkama.

Stvaranje baze podataka kao i odpakiravanje prethodno pripremljene baze podataka obaviti će se preko web aplikacije *phpmyadmin* koja standardno dolazi u *LAMPP* paketu.

U web pregledniku upišite slijedeću web adresu kako bi pristupili *phpmyadmin* aplikaciji <http://localhost/phpmyadmin>. Prikazati će se stranica, kao na slijedećoj slici, gdje se u središnjem dijelu upisuje naziv baze podataka. U ovom slučaju smo napisali za naziv baze podataka *rouws*.



Slika 10 - Stvaranje baze podataka

Nakon što smo stvorili bazu podataka, potrebno je kreirati tablice i preddefinirane podatke. To ćemo uraditi tako da kliknemo na poveznicu Uvoz (*engl. import*) gdje je potrebno preko web forme odabrati *rouws.sql.zip* datoteku koja se nalazi u *database* direktoriju kojeg smo dobili otpakiravanjem *rouws.zip* datoteke.

Uvoz

Datoteka za uvoz

Lokacija tekstualne datoteke: (Najv: 16,384 kB)

Tablica znakova za datoteku:

Kompresija uvezene datoteke bit će automatski prepoznata iz: bez kompresije, gzip, zip

Djelomičan uvoz

☒ Dopusti prekid uvoza u slučaju da skripta otkrije blizinu vremenskog ograničenja. Ovo bi mogao biti problem s velikom datotekom, ali može prekinuti transakcije.

Broj zapisa (upita) koje je potrebno preskočiti od početka:

Oblikovanje uvezene datoteke

Slika 11 - Odabir i instalacija prethodno pripremljene baze podataka

Nakon odabira potrebno je kliknuti na tipku Kreni kako bi počeo proces otpakiravanja prethodno pripremljene baze podataka.

phpMyAdmin

Poslužitelj: localhost ▶ Baza podataka: rouws

Strukturu SQL Traži Upit Izvoz Uvoz Kreator Operacije

Baza podataka

rouws (22)

rouws (22)

- oi_sessions
- zt_actions
- zt_components
- zt_component_methods
- zt_component_types
- zt_component_type_methods
- zt_layouts
- zt_layout_positions
- zt_menus
- zt_menu_items
- zt_modules
- zt_modules_groups
- zt_modules_group_items
- zt_module_types
- zt_pages
- zt_page_modules_groups
- zt_roles
- zt_roles_assignment
- zt_role_permissions
- zt_site_settings
- zt_users
- zt_users_autologin

Uvoz

Datoteka za uvoz

Lokacija tekstualne datoteke: (Najv: 16,384 kB)

Tablica znakova za datoteku:

Kompresija uvezene datoteke bit će automatski prepoznata iz: bez kompresije, gzip, zip

Djelomičan uvoz

☒ Dopusti prekid uvoza u slučaju da skripta otkrije blizinu vremenskog ograničenja. Ovo bi mogao biti problem s velikom datotekom, ali može prekinuti transakcije.

Broj zapisa (upita) koje je potrebno preskočiti od početka:

Oblikovanje uvezene datoteke

☐ DocSQL

☒ SQL

Opcije

Način rada SQL kompatibilnosti:

Slika 12 - Prikaz instalirane prethodno pripremljene baze podataka

Korak 3.

U trećem koraku potrebno je sadržaj direktorija *web_app* prekopirati u *htdocs* direktorij (na nekim poslužiteljima taj direktorij se još naziva i *public_html*). Htdocs direktorij je root direktorij koji vide korisnici koji pristupaju određenoj stranici i tu se nalaze datoteke za generiranje i prikaz web stranica. U našem konkretnom slučaju *htdocs* direktorij se nalazi u */opt/lampp/* direktoriju.

Korak 4.

Sljedeći korak obuhvaća uređivanje konfiguracijskih datoteka koje su sastavni dio *Codeigniter* PHP okoline. Te konfiguracijske datoteke se nalaze u *system/application/config* direktoriju unutar *htdocs* direktorija.

Prvo ćemo urediti *database.php* datoteku. Datoteka *database.php* drži neke standardne postavke vezane za upravljanje bazom podataka. Nakon što smo otvorili datoteku potrebno je **obavezno** promijeniti vrijednosti varijabla prikazanim u sljedećoj tablici:

Tabela 2 - Značenje pojedinih varijable u database.php datoteci

Deklaracije varijabli i vrijednosti	Objašnjenja
<code>\$db['default']['hostname'] = "domena_ili_ip";</code>	Odnosi se na domenu ili ip adresu poslužitelja na kojemu se nalazi sustav MySql baze podataka. Najčešće se na to mjesto piše <i>localhost</i> .
<code>\$db['default']['username'] = "korisnicko_ime";</code>	Korisničko ime korisnika za pristup bazi podataka.
<code>\$db['default']['password'] = "lozinka";</code>	Lozinka korisnika za pristup bazi podataka.
<code>\$db['default']['database'] = "ime_baze_podataka";</code>	Ime baze podataka koja će se koristiti za radnu okolinu za upravljanje web

sadržajem. U našem primjeru to će biti *rouws*.

Korak 5.

Nakon uređivanja *database.php* datoteke potrebno je urediti *config.php* datoteku. Config.php datoteka sadrži standardne postavke vezane za *Codeigniter* PHP okolinu. U sljedećoj tablici biti će prikazane postavke koje je potrebno promijeniti, ostale ostavljamo na izbor (proučiti dokumentaciju Codeigniter). Veći dio postavki je već podešen, no prilikom nadogradnje na noviju verziju Codeigniter okoline potrebno ih je opet podesiti.

Tabela 3 - Značenje pojedinih varijabli u config.php datoteci

Deklaracije varijabli i vrijednosti	Objašnjenja
<code>\$config['base_url'] = "http://localhost/</code>	Osnovni URL web aplikaciju sa pratećim slash znakom. Koristi se u aplikaciji za stvaranje hiperveza.
<code>\$config['index_page'] = "";</code>	Svo procesiranje u Codeigniter framework-u obavlja se preko indeks.php datoteke, čiji naziv se piše u navedenu postavku. No međutim radi estetike linkova ovdje ostavljamo prazno.
<code>\$config['sess_expiration']= 7200;</code>	Vrijeme trajanja sjednice u sekundama.
<code>\$config['sess_use_database']= TRUE;</code>	Potrebno je postaviti na TRUE za spremanje podataka iz sjednice u bazu podataka.

3.3 Korisničke upute

U ovome dijelu biti će opisane upute korisnicima na koji način mogu koristiti sustav za upravljanje sadržajem. Sve što korisnik mora imati od programske potpore je Internet preglednik (jedan on onih sa popisa preporučenih).

Sam sustav za upravljanje web sadržajem sastoji se od djela koji je namijenjen korisnicima sjedišta, te administratorskog djela koji je namijenjen objavljivačima sadržaja i administratorima.

Ukoliko ste uspješno obavili instalacijske procese opisane u prethodnim poglavljima, tada će vam se upisivanjem adrese <http://localhost> u Internet pregledniku prikazati sljedeća stranica:



Slika 13 - Početna stranica web aplikacije za upravljanjem sadržaja

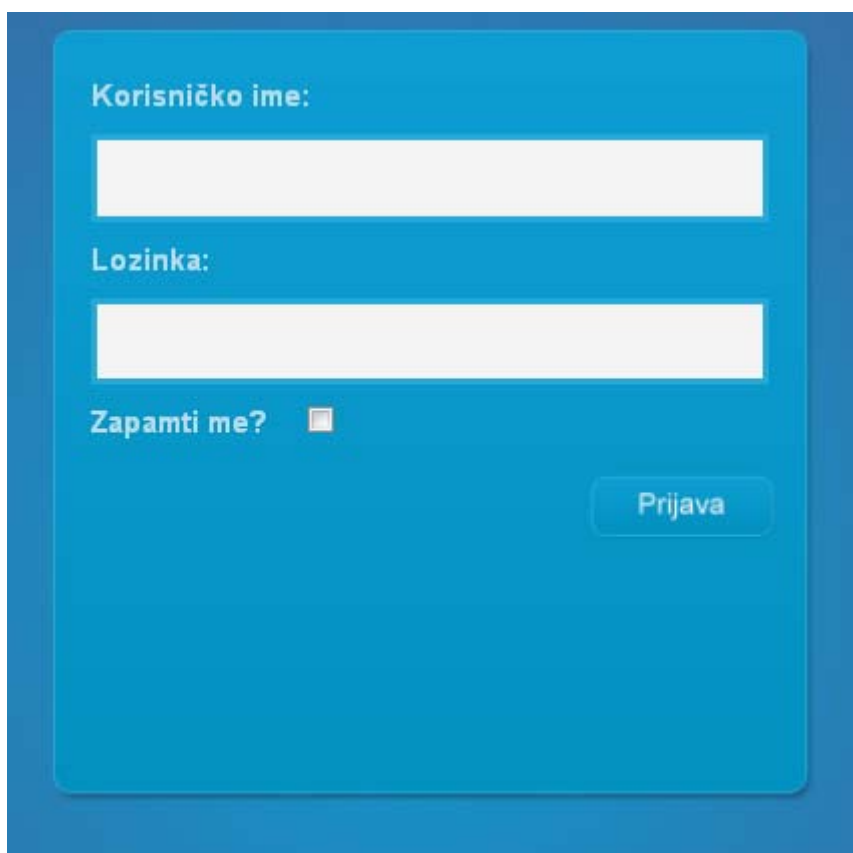
Najviše ćemo opisivati administratorski dio sjedišta, jer je to najkompleksniji dio.

3.3.1 Prijava na sustav

Korisnici se prijavljuju na sustav preko sljedeće adrese:

<http://localhost/admin>

Upisom navedene adrese otvara se stranica za unos korisničkog imena i lozinke kako bi se prijavili na sustav.



Slika 14 - Prijava na administratorski sustav

Prilikom svježe instalacije prijavu je moguće obaviti sa sljedećim podacima:

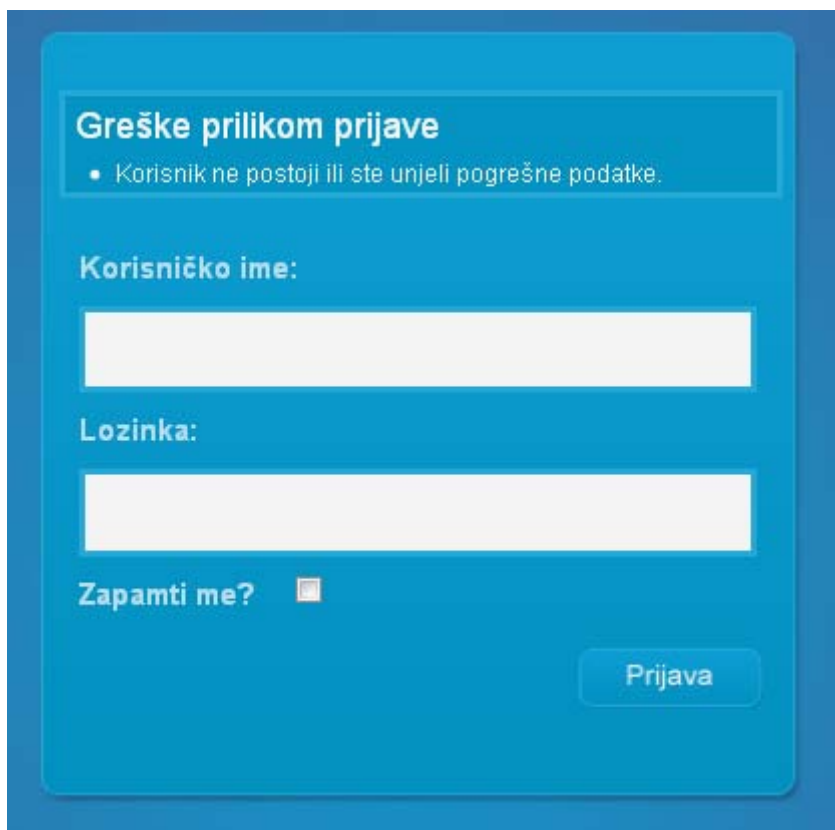
- **Korisničko ime** – admin
- **Lozinka** – 12345678

Kasnije je te podatke moguće izmijeniti.

Važno je napomenuti kako sustav omogućuje dodjelu ulogu korisnicima, te svaka uloga ima određene dozvole za obavljanje akcija na sustavu. Korisnik sa korisničkim

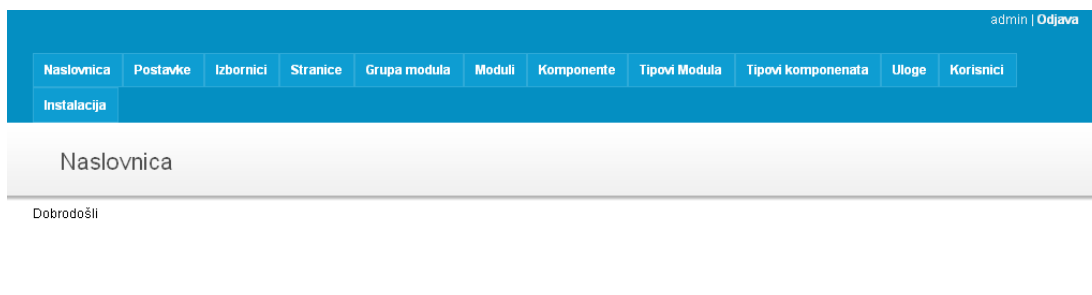
imenom *admin* ima ulogu *Super administratora* koji ima sve dozvole, te mu nije potrebno eksplicitno definirati nikakve dozvole. No više o tome u narednim poglavljima.

Prilikom neuspješne prijave, bilo da su neispravni korisničko ime i lozinka, ili je korisnik ne aktivan prikazati će se informacija o grešci.

The image shows a login interface with a blue background. At the top, there is a white box with the title "Greške prilikom prijave" (Errors during login). Below this title, a red bullet point indicates the error: "Korisnik ne postoji ili ste unjeli pogrešne podatke." (User does not exist or you entered incorrect data). Below the error message, there are two input fields: "Korisničko ime:" (Username) and "Lozinka:" (Password). Below the password field, there is a checkbox labeled "Zapamti me?" (Remember me?). At the bottom right, there is a blue button labeled "Prijava" (Login).

Slika 15 - Neuspješna prijava na sustav

Kada se korisnik uspješno prijavi na sustav dočekati će ga naslovna strana administratorskog djela sustava.



Slika 16 - Naslovna stranica administratorskog djela

Na samom vrhu stranice nalazi se glavni izbornik sa poveznicama na druge dijelove sustava. Svaki element izbornika biti će objašnjen u nastavku.

U gornjem desnom kutu prikazano je ime korisnika, te poveznica za odjavu sa sustava.

3.3.2 Postavke web sjedišta

Prilikom objašnjenja svih elemenata glavnog izbornika i opisom mogućnosti radne okoline za objavu web sadržaja, započeti ćemo sa postavkama web sjedišta.

U načelu postavke web sjedišta su vrlo osnovne i trivijalne. Dakle za svako web sjedište potrebno je definirati naslov web sjedišta, kao i opis što prikazuje određena web stranica. Isto tako moguće je definirati neke ključne riječi koje su karakteristične za sadržaj stranice.

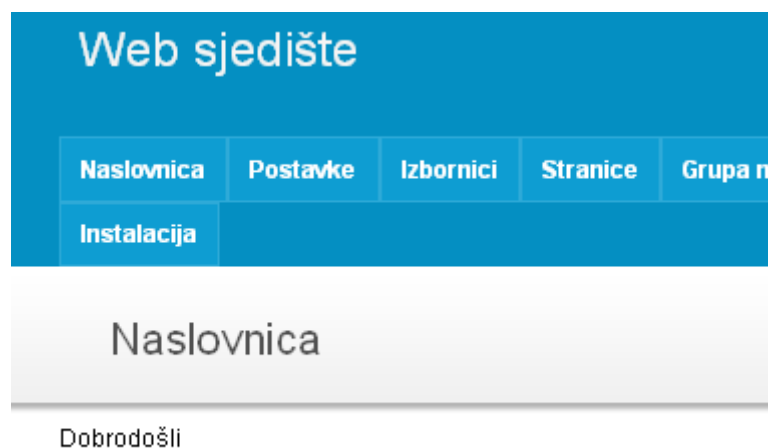
Ove postavke vrijede globalno za svaku stranicu web sjedišta. Važno je napomenuti da svaka komponenta, odnosno stranica web sjedišta ima mogućnost vlastite

implementacije ovih postavki, no ukoliko one nisu definirane tada se prikazuju ove globalne postavke.

The screenshot shows a web management interface with a blue header bar containing navigation tabs: 'Naslovnica', 'Postavke', 'Izbornici', 'Stranice', 'Grupa modula', 'Moduli', and 'Komponente'. Below this is a sub-header 'Postavke stranice'. The form contains three input fields: 'Naziv stranice' with the value 'Web sjedište', 'Opis stranice' with a multi-line text 'Web sjedište za prikaz mogućnosti radne okoline sustava za upravljanje web', and 'Ključne riječi' with the value 'web, sjedište, okolina, upravljanje, php, primjeri'. At the bottom are two buttons: 'Spremi' and 'Odustani'.

Slika 17 - Prikaz stranice sa postavkama web sjedišta

Nakon spremanja vidljive su promjene. Tako se, iznad glavnog izbornika administracije, pokazao naziv web sjedišta.



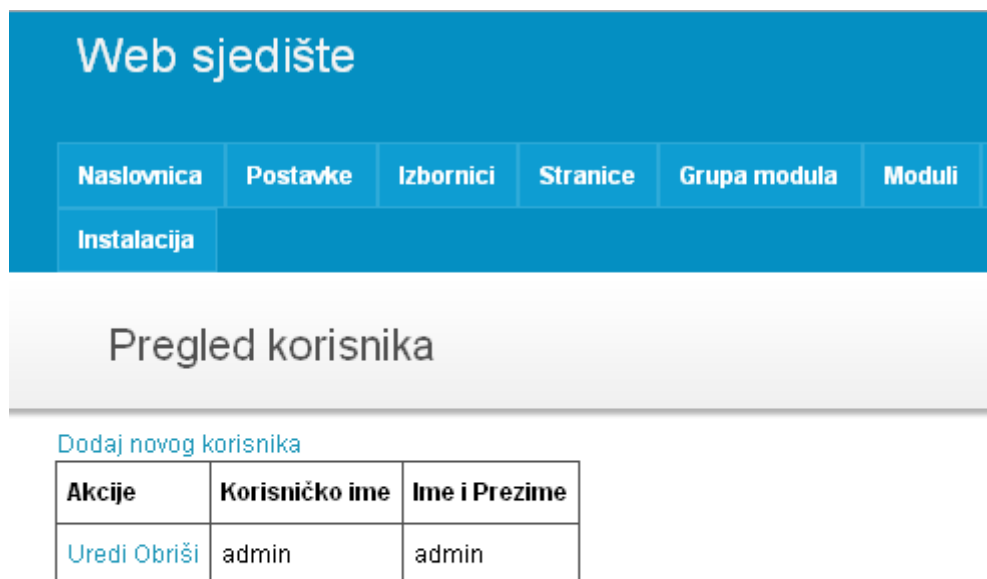
Slika 18 - Prikaz rezultata spremanja postavki

Isto tako ako se pogleda izvorni html kod stranice, vidjeti ćemo naslov i meta podatke koje su izgenerirani iz postavki.

```
<title>Web sjedište | Naslovnica</title>
<meta name="description" content="Web sjedište za prikaz mogućnosti
radne okoline sustava za upravljanje web sadržajem." />
<meta name="keywords" content="web, sjedište, okolina, upravljanje,
php, primjeri" />
```

3.3.3 Korisnici

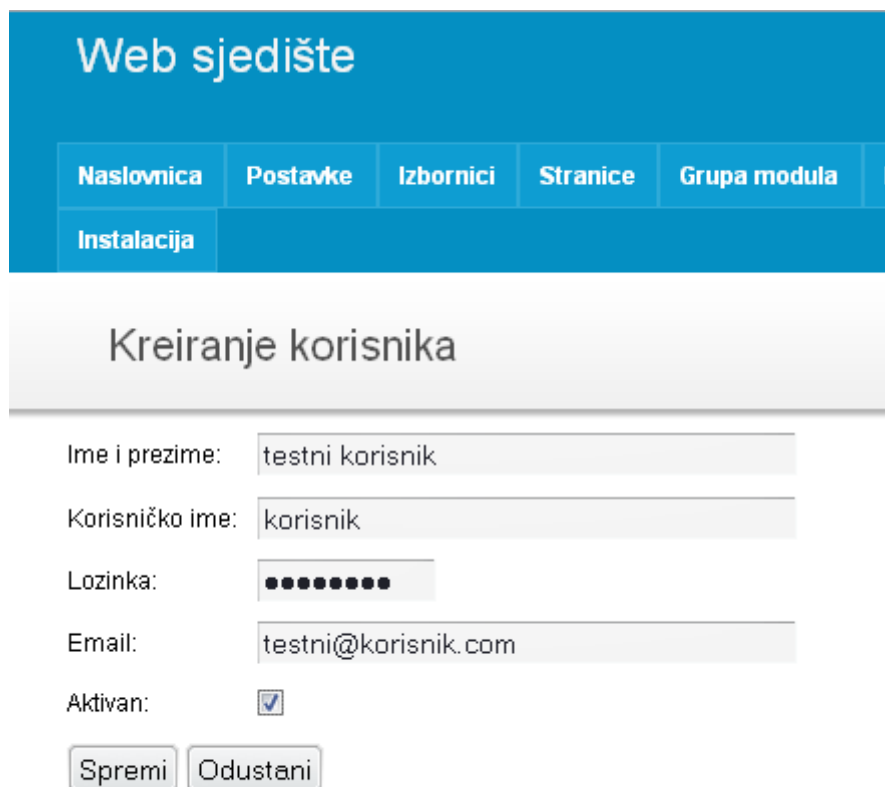
Odabirom poveznice *Korisnici* iz glavnog izbornika administracije sustava moguće je obavljati operacije upravljanja korisnicima. Početna stranica sadrži pregled korisnika koji su registrirani u sustavu. Prilikom svježe instalacije vidljiv je korisnik *admin* preko kojeg smo se i prijavili na sustav.



Akcije	Korisničko ime	Ime i Prezime
Uredi Obriši	admin	admin

Slika 19 - Pregled korisnika sustava

Sustav omogućuje uređivanje, brisanje, te stvaranje novih korisnika. Pa da bi smo stvorili novog korisnika potrebno je kliknuti na poveznicu *Dodaj novog korisnika*. Nakon toga otvara nam se stranica kao na sljedećoj slici.



The screenshot shows a web application interface with a blue header titled 'Web sjedište'. Below the header is a navigation menu with buttons: 'Naslovnica', 'Postavke', 'Izbornici', 'Stranice', 'Grupa modula', and 'M'. Below the navigation menu is a section titled 'Kreiranje korisnika' (Create user). This section contains a form with the following fields: 'Ime i prezime:' (Name and surname) with the value 'testni korisnik', 'Korisničko ime:' (Username) with the value 'korisnik', 'Lozinka:' (Password) with masked characters, 'Email:' with the value 'testni@korisnik.com', and 'Aktivan:' (Active) with a checked checkbox. At the bottom of the form are two buttons: 'Spremi' (Save) and 'Odustani' (Cancel).

Slika 20 - Kreiranje korisnika sustava

Da bi smo kreirali novog korisnika potrebno je popuniti web formu sa sljedećim podacima:

- Ime i prezime korisnika
- Korisničko ime – ime s kojim se korisnik prijavljuje na sustav. Korisničko ime mora biti jedinstveno
- Lozinka – korisnička lozinka za prijavu na sustav
- Email – e-mail adresa korisnika. E-mail adresa mora biti jedinstvena.
- Aktivan – označava aktivaciju korisnika. Ukoliko korisnik nije aktivan neće se moći prijaviti na sustav, sve dok ga nadležna osoba ponovno ne aktivira.

Sustav omogućuje i uređivanje podataka o korisniku, te mu osim standardnih podataka koje smo vidjeli prilikom stvaranja korisnika, pruža mogućnost dodjele uloge u sustavu. O ulogama u sustavu u narednim poglavljima. Dakle, korisniku je moguće dodijeliti više uloga. Postupak dodjele uloga se obavlja tako da stavimo kvačicu na onu ulogu koju želimo dodijeliti korisniku. Ukoliko želimo neku ulogu obrisati potrebno je maknuti kvačicu.

The screenshot shows a web application interface with a blue header titled 'Web sjedište'. Below the header is a navigation bar with tabs: 'Naslovnica', 'Postavke', 'Izbornici', 'Stranice', 'Grupa modula', and 'Moduli'. The 'Postavke' tab is active. Below the navigation bar is a sub-header 'Uređivanje korisnika'. The form contains the following fields and options:

- Ime i prezime: testni korisnik
- Korisničko ime: korisnik
- Lozinka: (empty field)
- Email: testni@korisnik.com
- Aktivan: ☒
- Dodjela uloga:
 - ☐ Administrator
 - ☒ Moderator
- Buttons: 'Spremi' and 'Odustani'

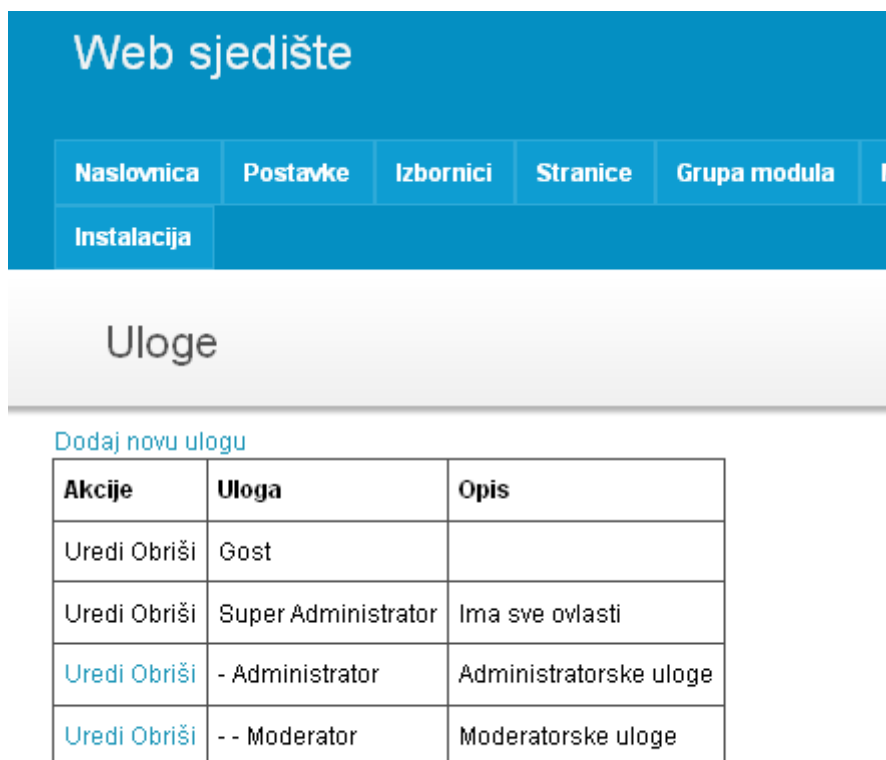
Slika 21 - Izmjena podataka korisnika sustava

3.3.4 Uloge

Svaki pravi sustav za upravljanje web sadržajem ima neku vrstu raspodjele odgovornosti, odnosno uloga. Ovo radno okruženje za upravljanje web sadržajem pruža hijerarhijske uloge. To znači da svaka uloga može imati nadređenu ulogu. Odnosno sve ovlasti koje ima podređena uloga, automatski ima i nadređena uloga, te

nije potrebno eksplicitno definirati dozvole nadređenoj ulozi. Više o dozvolama u narednim poglavljima.

Odabirom poveznice *Uloge* prikazuje nam se stranica sa prikazom uloga na sustavu. Na sljedećoj slici prikazane su već preddefinirane uloge sustava.

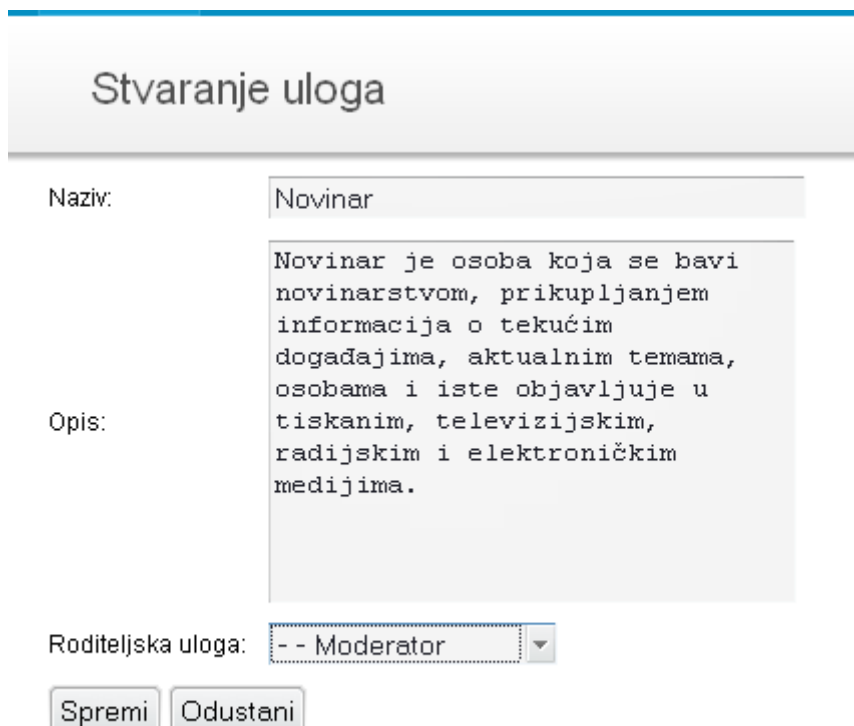


Slika 22 - Pregled uloga u sustavu

Nad ulogama moguće je obavljati akcije uređivanja, brisanja i dodavanja novih uloga. Važno je napomenuti da grupe kao što su *Gost* i *Super Administrator* nije moguće uređivati niti brisati jer su zaključane od strane sustava.

Da pojasnimo, uloga *Gost* je dodijeljena svakom korisniku/posjetitelju web sjedišta, sve dok se taj korisnik ne prijavi na sustav, odnosno ne autentificira. Uloga *Super Administrator* je uloga koja je po hijerarhiji najviša u lancu uloga, odnosno ima sva prava obavljanja akcija nad nekim entitetima i nadležnosti nad ostalim ulogama.

Ukoliko želimo kreirati novu ulogu ili postojeću urediti tada nam se otvara stranica i pripadajuća web forma kao na sljedećoj slici:



Stvaranje uloga

Naziv: Novinar

Opis: Novinar je osoba koja se bavi novinarstvom, prikupljanjem informacija o tekućim događajima, aktualnim temama, osobama i iste objavljuje u tiskanim, televizijskim, radijskim i elektroničkim medijima.

Roditeljska uloga: -- Moderator

Spremi Odustani

Slika 23 - Stvaranje i uređivanje uloga

Podatke koje je potrebno unijeti u web formu su sljedeći:

- Naziv – naziv uloge
- Opis – opis uloge
- Roditeljska uloga- uloga koja je nadređena ulozi koja se stvara ili uređuje.

3.3.5 Tipovi komponenata i komponente

Komponente su u načelu ono što nazivamo web stranicama. Sa tehničkog stajališta to je program koji na temelju podataka koje je korisnik unio/napravio generira sadržaj. Svaka komponenta generira drugačiji ili drugačiju vrstu sadržaja. U načelu postoji pravilo da veći broj komponenti povećava funkcionalnu mogućnost radne okoline za upravljanjem sadržajem.

Tip komponente je razvijeni program koji obavlja nekakvu zadaću. Iz tipa komponente moguće je stvoriti instancu komponente koja će za zadane parametre

ostvarivati navedene zadatke. Ukoliko tip komponente to dozvoljava, moguće je stvarati višestruke instance komponenata iz određenog tipa komponente.

Prikazati ćemo to na primjeru. Neka postoji tip komponente blog čija je zadaća objavljivanje i uređivanje sadržaja bloga. Ukoliko postoji više korisnika na sustavu i za svakog od njih je potrebno kreirati vlastiti blog, tada je potrebno kreirati instancu bloga, odnosno komponentu za svakog člana.

Odabirom poveznice *Tipovi komponenata* otvara se stranica gdje su prikazani tipovi komponenata sa podacima, kao što je opis komponente, te da li tip komponente dozvoljava višestruke instance, odnosno stvaranje novih komponenata/stranica koje će obavljati neku zadaću.

Tipove komponenata je moguće i obrisati, no u tom postupku mora se biti pažljiv jer brisanjem tipa komponente se briše i instanca toga tipa, odnosno komponenta ili stranica sa svim sadržajem.

Dodavanje novog tipa komponente obavlja se instalacijom paketa, ali o tome u narednim poglavljima.

Web sjedište

Naslovnica

Postavke

Izbornici

Stranice

Grupa modula

Moduli

Instalacija

Tipovi komponenata

Akcije	Id	Tip komponente	Opis komponente	Višestruke instance
Deinstaliraj	7	FrontPage	Početna stranica	Ne

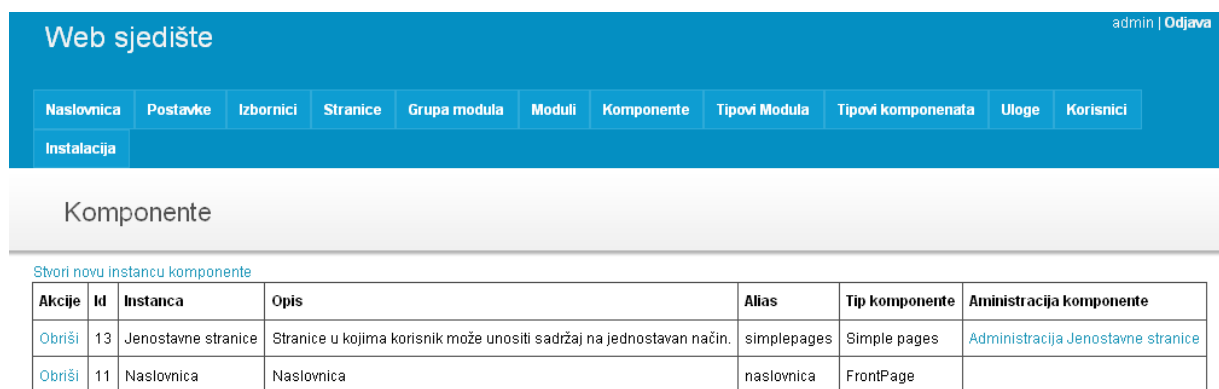
Slika 24 - Prikaz tipova komponenata

Odabirom poveznice *Komponente* iz glavnog izbornika administratorskog djela web sjedišta prikazati će nam se stranica sa prikazom komponenti na sustavu.

Dakle da ponovimo, komponenta je jedna ili skup stranica koje generiraju ili primaju unos sadržaja.

Komponente je moguće stvarati, te brisati. Kod brisanja komponente treba biti pažljiv jer se briše sav sadržaj sa tom komponentom. Na navedenoj stranici prikazane su osnovne informacije o komponenti kao što je naziv, opis, alias, tip komponente i poveznica za administraciju komponente.

Alias komponente je naziv početne stranice komponente preko koje se pristupa podstranicama komponente. Dakle ukoliko bi upisali <http://localhost/naslovnica> tada bi nam se otvorila početna stranica web sjedišta. Administracija komponente omogućuje unos podataka koji su bitni za generiranje sadržaja od strane te komponente. Sa sljedeće slike, vidljivo je da za *Naslovnicu*, te poveznice nema jer to nije omogućeno za taj tip komponente, a za *Jednostavne stranice* ona postoji.



Web sjedište											admin Odjava
Naslovnica	Postavke	Izbornici	Stranice	Grupa modula	Moduli	Komponente	Tipovi Modula	Tipovi komponenata	Uloge	Korisnici	
Instalacija											
Komponente											
Stvori novu instancu komponente											
Akcije	Id	Instanca	Opis	Alias	Tip komponente	Administracija komponente					
Obriši	13	Jednostavne stranice	Stranice u kojima korisnik može unositi sadržaj na jednostavan način.	simplepages	Simple pages	Administracija Jednostavne stranice					
Obriši	11	Naslovnica	Naslovnica	naslovnica	FrontPage						



Slika 25 - Prikaz komponenata na sustavu

Moguće je stvoriti novu instancu komponente, te se u tom slučaju prikazuje stranica kao na sljedećoj slici:

The screenshot shows a web application interface for managing a website. At the top, there is a blue header with the text 'Web sjedište'. Below the header is a navigation bar with several tabs: 'Naslovnica', 'Postavke', 'Izbornici', 'Stranice', 'Grupa modula', and 'Moduli'. The 'Stranice' tab is currently selected. Below the navigation bar is a large light gray box with the title 'Stvaranje instance komponente'. Under this title is a form with the following fields:

- Naziv:** A text input field containing 'Jenostavne stranice'.
- Opis:** A text area containing the text 'Stranice u kojima korisnik može unositi sadržaj na jednostavan način.'.
- Alias:** A text input field containing 'simplepages'.
- Tip komponente:** A dropdown menu with 'Simple pages' selected.

At the bottom of the form are two buttons: 'Sljedeći korak' (Next step) and 'Odustani' (Cancel).

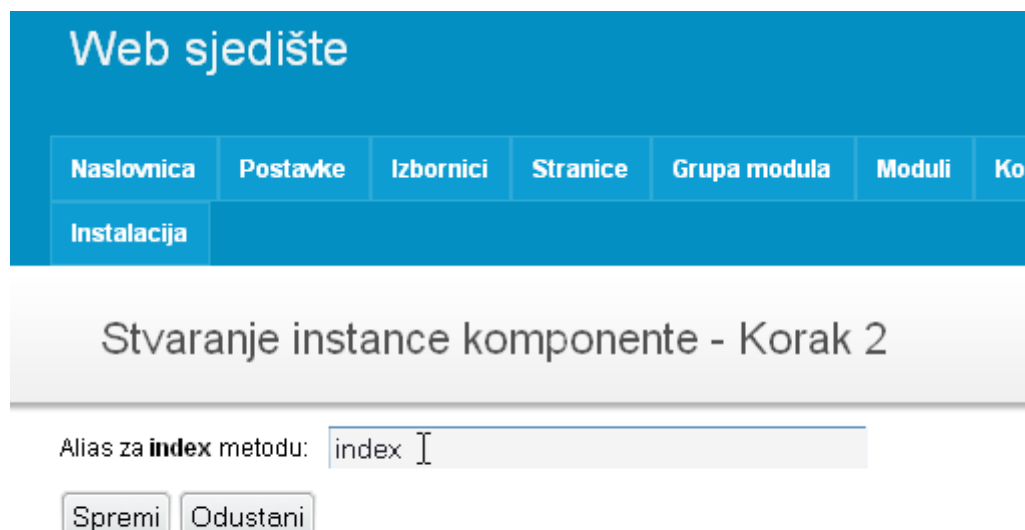
Slika 26 - Stvaranje instance komponente

Forma za stvaranja instance komponente zahtjeva sljedeće podatke:

- Naziv – naziv instance komponente
- Opis – opis instance komponente
- Alias – ime preko kojeg će se otvarati početna stranica komponente
- Tip komponente – tip na temelju kojeg će se stvoriti instanca komponente

Važno je napomenuti dio za tip komponente. Ukoliko neki tip komponente dozvoljava samo jednu instancu, tada će biti samo prikazani oni tipovi koji dozvoljavaju višestruke instance ili oni sa jednostrukim instancama a da do sada nisu instancirani.

Pritiskom na tipku sljedeći korak, otvara se stranica za definiranje naziva podstranica, kao na sljedećoj slici.



Slika 27 - Prikaz stranice za definiranje naziva podstranica komponente

3.3.6 Tipovi modula i moduli

Moduli su manji dijelovi stranice, koji ne generiraju glavni sadržaj, ali pružaju korisne informacije ili su dijelovi korisničkog sučelja kao što su izbornici. Moduli mogu biti samostalni dijelovi stranice kao što je to izbornik, ili vezani za neku komponentu. Ako uzmemo primjer bloga iz prethodnog poglavlja, tada će modul koji je vezan za komponentu bloga prikazivati informacije o posljednjim člancima objavljenim na blogu. Isto tako jedna od glavnih značajki modula je da se mogu pojavljivati na različitim stranicama i različitim pozicijama, ali više o tome u narednim poglavljima.

Tipovi modula imaju u načelu jednako značenje kao i tipovi komponenti. Dakle, tipovi modula predstavljaju program sa određenim zadatkom. A moduli su instance nekog tipa modula, sa postavkama karakterističnim za tu instancu.

Tipove modula je moguće deinstalirati, ali u tom slučaju treba biti pažljiv jer se brišu sve instance modula proizašle iz tog tipa.

Web sjedište

Naslovnica

Postavke

Izbornici

Stranice

Grupa modula

Moduli

Komponente

Instalacija

Tipovi modula

Akcije	Id	Tip modula	Opis modula	Višestruke instance	Tip komponente
Deinstaliraj	1	Menus	Izbornik	Ne	

Slika 28 - Pregled tipova modula

Na stranici *Moduli* omogućen je pregled modula, te operacije uređivanja, brisanja i stvaranja novih instanci modula.

Web sjedište

Naslovnica

Postavke

Izbornici

Stranice

Grupa modula

Moduli

Komponente

Instalacija

Moduli

[Stvori novu instancu modula](#)

Akcije	Id	Instanca	Opis	Tip modula	Vezana komponenta	Objavljeno
Uredi Obriši	1	Glavni izbornik	Navigacija	Menus		Ne

Slika 29 - Pregled modula

Prilikom stvaranja modula potrebno je unijeti podatke:

- Naziv – naziv modula
- Opis – opis modula
- Tip modula – tip modula temeljem kojeg se stvara instanca

The screenshot shows a web application interface for managing a website. At the top, there is a blue header with the text 'Web sjedište'. Below the header is a navigation bar with five tabs: 'Naslovnica', 'Postavke', 'Izbornici', 'Stranice', and 'Grupa modula'. The 'Izbornici' tab is currently selected. Below the navigation bar is a sub-header 'Stvaranje modula'. The main form contains the following fields:

- 'Naziv:' (Name) with a text input field containing 'Glavni izbornik'.
- 'Opis:' (Description) with a text input field containing 'Navigacija'.
- 'Tip modula:' (Module type) with a dropdown menu showing 'Menus'.
- Two buttons at the bottom: 'Spremi' (Save) and 'Odustani' (Cancel).

Slika 30 - Prikaz stranice za stvaranje modula

Na stranici za uređivanje modula definirana je forma sa podacima koji su korišteni prilikom kreiranja, te dodatni podaci koji su karakteristični za tip modula temeljem kojeg je nastala instanca modula.

U ovom slučaju odabire se izbornik koji će se prikazivati preko te instance modula.

Web sjedište

Naslovnica Postavke Izbornici Stranice Grupa modula

Instalacija

Uređivanje modula

Naziv: Glavni izbornik

Opis: Navigacija

Tip modula: Menu

Objavljen: ☒

Odaberite izbornik koji će predstavljati instanca modula:

Glavni izbornik ▼

Spremi Odustani

Slika 31 - Prikaz stranice za uređivanje instance modula

3.3.7 Grupe modula

Kako se moduli mogu prikazivati na različitim pozicijama i redoslijedu na stranici definiraju se grupe modula. Tako ako se prije neka grupa prikazivala na vrhu stranice moguće je definirati da se prikazuje na dnu stranice, a promjena će se odnositi na sve module unutar te grupe.

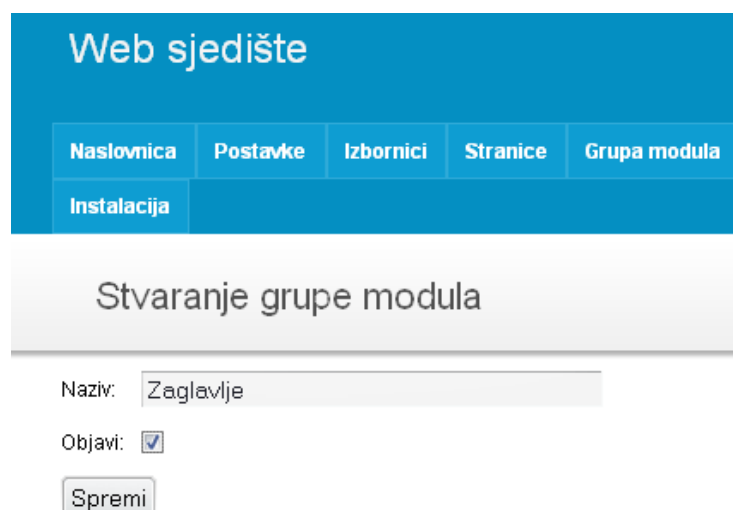
Odabirom poveznice *Grupe modula* otvara se stranica sa prikazom grupe modula. Moguće je stvarati, uređivati i brisati grupe modula, te dodjeljivati elemente, odnosno module grupi modula.



Slika 32 - Prikaz grupe modula

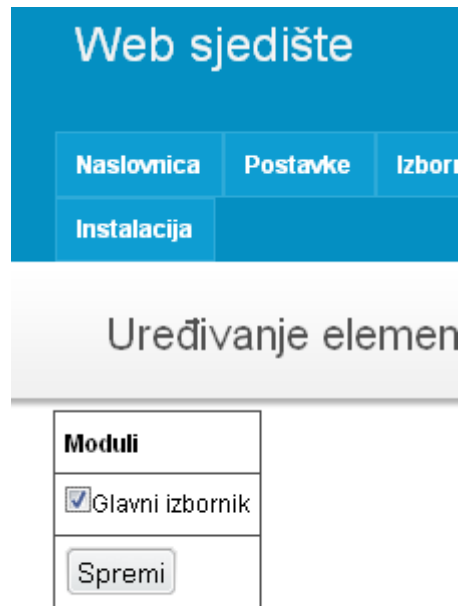
Prilikom stvaranja grupe modula ponuđena je web forma za unos i uređivanje sljedećih podataka:

- Naziv – naziv grupe modula
- Objavi – da li je grupa objavljena, tj. da li se prikazuje na stranicama



Slika 33 - Prikaz stranice za stvaranje i uređivanje grupe modula

Na stranici za uređivanje elemenata grupe modula, moguće je postavljanjem kvačice na modul dodavati module grupi modula, ili brisanjem kvačice brisati module iz grupe modula.

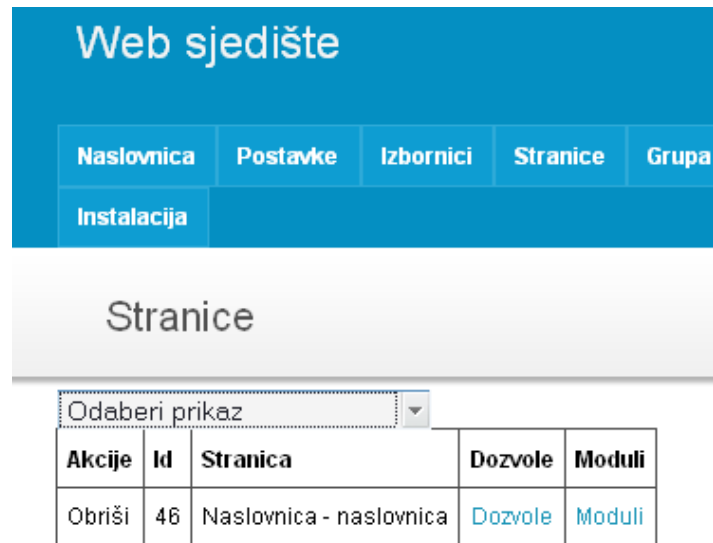


Slika 34 - Prikaz stranice za uređivanje elemenata grupe modula

3.3.8 Stranice

Odabirom poveznice *Stranice* iz glavnog izbornika administratorskog djela stranice prikazuje se stranica koja sadrži prikaz svih raspoloživih, generiranih stranica od strane radne okoline za upravljanje sadržaja.

Stranica nudi izbornik za filtriranje stranica. Filtrirati je moguće administratorske stranice, kao i one javne. Dakle javne stranice su one stranice koje su namijenjene širem krugu ljudi, dok su one administratorske za objavljivanje sadržaja i administratore.



Slika 35 - Pregled stranica sustava

Stranice je moguće obrisati ukoliko nisu zaključane od strane sustava, odnosno komponente.

Kako bi se obavile određene akcije na stranici, kao što je čitanje, stvaranje, uređivanje ili brisanje tada je potrebno dodijeliti dozvole za obavljanje tih akcija ulogama. Dakle, oni korisnici kojima je dodijeljena određena uloga, a ta uloga ima dozvolu izvođenja akcije čitanja, tada će ti korisnici moći pogledati navedenu stranicu.

Sljedeća slika prikazuje formu za pridruživanje ili brisanje dozvola za uloge.

Dozvole za stranicu

Uloga	Akcije
Gost	<input checked="" type="checkbox"/> Čitanje <input type="checkbox"/> Stvaranje <input type="checkbox"/> Ažuriranje <input type="checkbox"/> Brisanje
Administrator	<input type="checkbox"/> Čitanje <input type="checkbox"/> Stvaranje <input type="checkbox"/> Ažuriranje <input type="checkbox"/> Brisanje
Moderator	<input type="checkbox"/> Čitanje <input type="checkbox"/> Stvaranje <input type="checkbox"/> Ažuriranje <input type="checkbox"/> Brisanje

SpremiOdustani

Slika 36 - Pregled stranice da dodjeljivanje dozvola za stranicu

Bitno je napomenuti, kako je dozvole potrebno dodijeliti i za administratorske stranice, tako da svaki korisnik ne može obavljati određene akcije u administratorskom djelu.

Kako se moduli prikazuju na stranicama i takav prikaz je moguć na različitim pozicijama, postoji stranica u kojoj se definira koja će se grupa modula prikazati, na kojem području i po kojem redoslijedu ukoliko se više grupa pokazuje na istom području stranice.

Web sjedište

Naslovnica

Postavke

Izbornici

Stranice

Grupa modula

Moduli

Komponente

Tipovi Modula

Instalacija

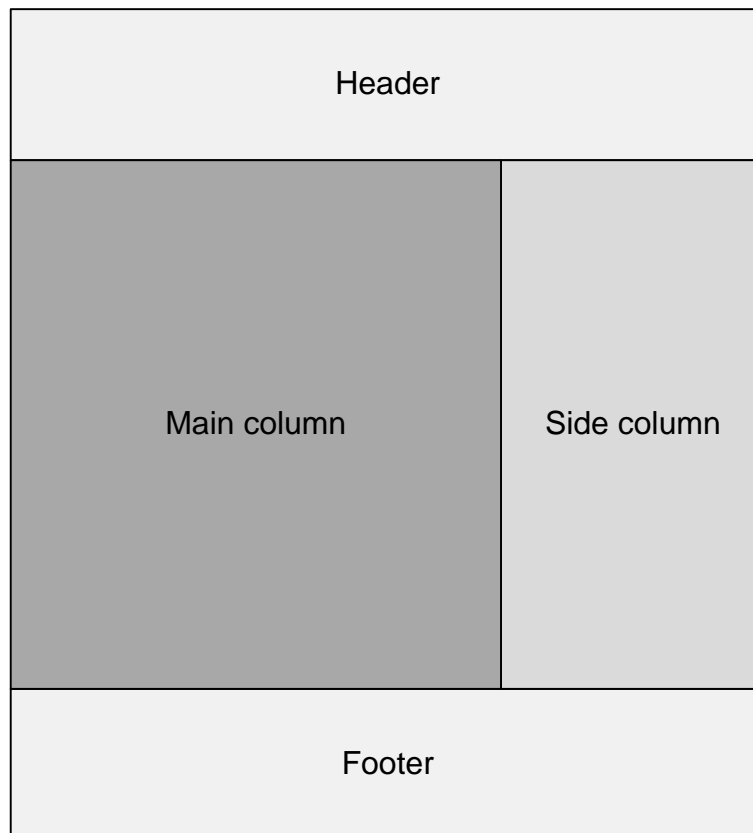
Moduli na stranici

Grupa modula	Prikazuje se na stranici	Područje na layout-u	Redoslijed grupe u zadanom području layout-a
Zaglavlje	<input checked="" type="checkbox"/>	Header	1

Spremi
Odustani

Slika 37 - Prikaz stranice za objavljivanje modula na stranici

Na sljedećoj slici prikazana su područja na kojim je moguće postaviti grupu modula za prikaz. Područja za razmještaj modula su Header, Side Column i Footer. U Main column se prikazuju sadržaj komponentata.



Slika 38 - Područja za razmještaj na stranici

3.3.9 Izbornici

Odabirom poveznice *Izbornici* otvara se stranica za pregled izbornika.

Izbornici je stranica povezana sa tipom modula *Menu*, čija je zadaća stvaranje izbornika. Dakle, prilikom stvaranja instance izbornika moguće je odabrati koji izbornik će se prikazivati u navedenoj instanci modula.



Slika 39 - Pregled izbornika

Izbornike je moguće stvarati, uređivati i brisati, te dodavati elemente izbornika. Elementi izbornika nisu ništa drugo nego poveznice na postojeće stranice sustava.

Prilikom stvaranja izbornika potrebno je zadati sljedeće informacije:

- Naziv – naziv izbornika
- Opis – opis izbornika

The screenshot shows the 'Web sjedište' (Website) management interface. At the top, there is a blue header with the title 'Web sjedište' and a navigation menu with items: 'Naslovnica', 'Postavke', 'Izbornici', 'Stranice', and 'Grupa mod'. Below the navigation menu, there is a sub-header 'Stvaranje izbornika' (Menu Creation). The form contains two input fields: 'Naziv' (Name) with the value 'Sporedni izbornik' and 'Opis' (Description) with the value 'izbornik'. At the bottom of the form, there are two buttons: 'Spremi' (Save) and 'Odustani' (Cancel). A mouse cursor is visible over the 'Odustani' button.

Slika 40 - Stvaranje novog izbornika

Za svaki izbornik potrebno je definirati njegove elemente. Klikom na poveznicu *Elementi* prikazuje se stranica sa prikazom svih elemenata toga izbornika.

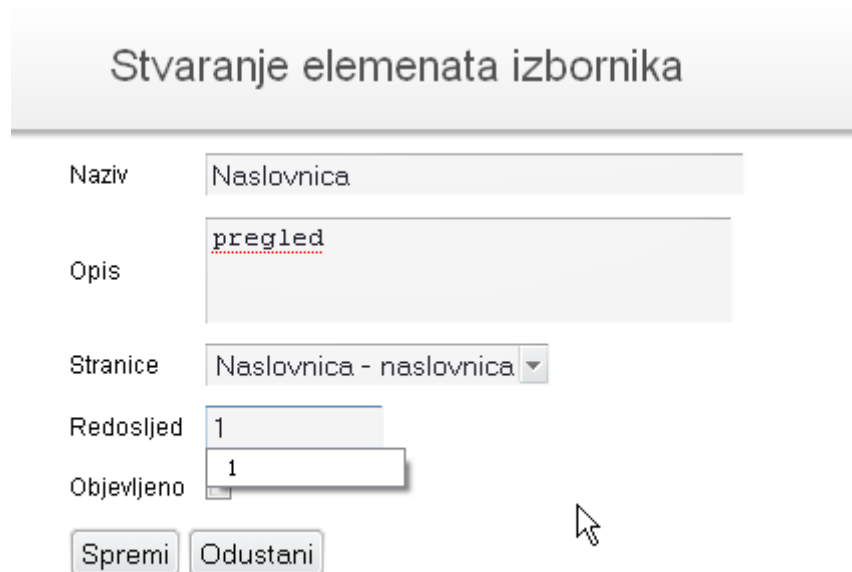
The screenshot shows the 'Web sjedište' (Website) management interface. At the top, there is a blue header with the title 'Web sjedište' and a navigation menu with items: 'Naslovnica', 'Postavke', 'Izbornici', 'Stranice', and 'G'. Below the navigation menu, there is a sub-header 'Uređivanje izbornika' (Menu Editing). Below the sub-header, there is a link 'Dodaj novi element izbornika' (Add new menu element). Below the link, there is a table with two columns: 'Akcije' (Actions) and 'Element izbornika' (Menu element). The table contains one row with the value 'Uredi Obriši' (Edit Delete) in the 'Akcije' column and 'Naslovnica' (Homepage) in the 'Element izbornika' column. A mouse cursor is visible over the 'Uredi Obriši' link.

Akcije	Element izbornika
Uredi Obriši	Naslovnica

Slika 41 - Pregled elemenata izbornika

Prilikom stvaranja novog elementa izbornika, potrebno je postaviti sljedeće podatke:

- Naziv – naziv elementa izbornika
- Opis – opis elementa izbornika
- Stranica – stranica na koju će element izbornika pokazivati
- Redosljed – redosljed elementa u izborniku u odnosu na druge elemente izbornika
- Objavljeno – da li će se element izbornika prikazivati u izborniku.



Slika 42 - Stvaranje elementa izbornika

Na slijedećoj slici je prikaz instance modula izbornika, prikazan na vrhu stranice.

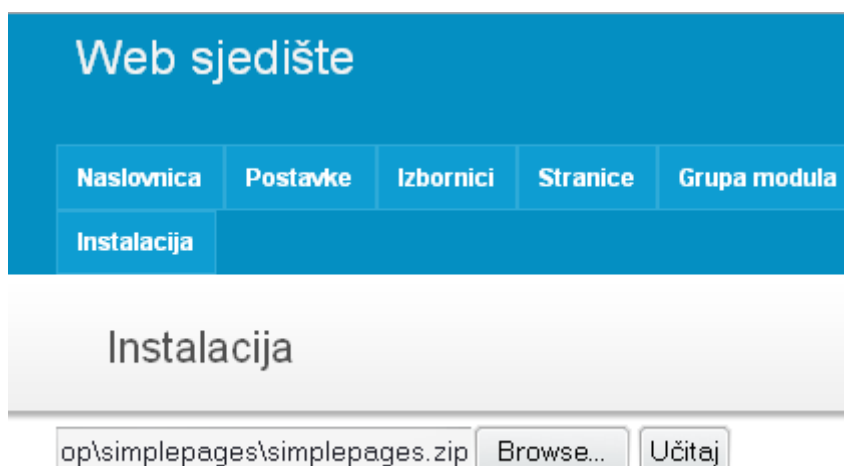


Slika 43 - Generiran izbornik

3.3.10 Instalacija paketa

Radno okruženje za upravljanjem sadržaja omogućuje instalaciju paketa koji mogu uključivati komponentu ili modul, ili oboje ako su povezani.

Odabirom poveznice *Instalacija* otvara se stranica koja pruža web formu za učitavanje zip arhive koja predstavlja paket koji želimo instalirati.



Slika 44 - Stranica za instalaciju paketa

Pritiskom na tipku Učitaj započinje učitavanje navedene datoteke sa paketom, te instalacija sadržaja u paketu.

Nakon instalacije, pojaviti će se novi tip komponente ili novi tip modula, ili oboje ako su povezani. Te se nakon toga može započeti sa postupkom instanciranja navedenih tipova.

3.4 Upute za izradu proširenja radnog okruženja za upravljanje web sadržajem

Da bi radna okolina za upravljanjem sadržaja zaista bila okolina, potrebno je takvu okolinu proširiti dodatnim komponentama i modulima. U nastavku će biti objašnjene upute za stvaranje paketa za radnu okolinu. Od čitatelja će se zahtijevati poznavanje programskog jezika PHP i rada na *Codeigniter* PHP okolini.

3.4.1 Struktura paketa

Paket mora biti zapakiran isključivo u zip arhivi. Struktura paketa prikazana je na slici Slika 44 - Struktura sadržaja paketa. Postoji nekoliko glavnih direktorija čiji sadržaj će biti objašnjen u nastavku:

- **component** – direktorij koji sadrži php datoteku koja predstavlja komponentu, odnosno kontroler (*engl. controller*) u *Codeigniter* PHP okolini, te xml datoteku istog naziva sa parametrima koji su bitni za instalaciju i rad same komponente. Ukoliko postoji administriranje za navedenu komponentu onda unutar tog direktorija postoji direktorij *admin* u kojem se također nalazi komponenta/kontroler za administratorski dio. Ukoliko se u paketu nalazi samo modul, tada *component* direktorij ne treba postojati. Opširnije o izradi komponente u narednim poglavljima.
- **database** – direktorij koji sadrži sql datoteku naziva paketa. Sql datoteka sadrži sve potrebne podatke za komponentu, modul ili oboje. Ukoliko nema potrebe za bazom podataka navedeni direktorij se može izostaviti.
- **libraries** – direktorij koji sadrži sve potrebne knjižnice koje su nužne za ispravno funkcioniranje komponente, modula ili oboje, a ne nalaze se u standardnoj radnoj okolini. Tijekom instalacije knjižnice se prebacuju u direktorij:

```
system/application/libraries/packages/aliasPaketa/libraries/.
```
- **models** – direktorij sadrži pripadajuće datoteke sa modelima za komponentu, modul ili oboje. Ukoliko ne postoji potreba za modelima, a to je najčešće povezano sa bazom podataka, tada se taj direktorij može izostaviti. Tijekom

instalacije modeli se prebacuju u direktorij:

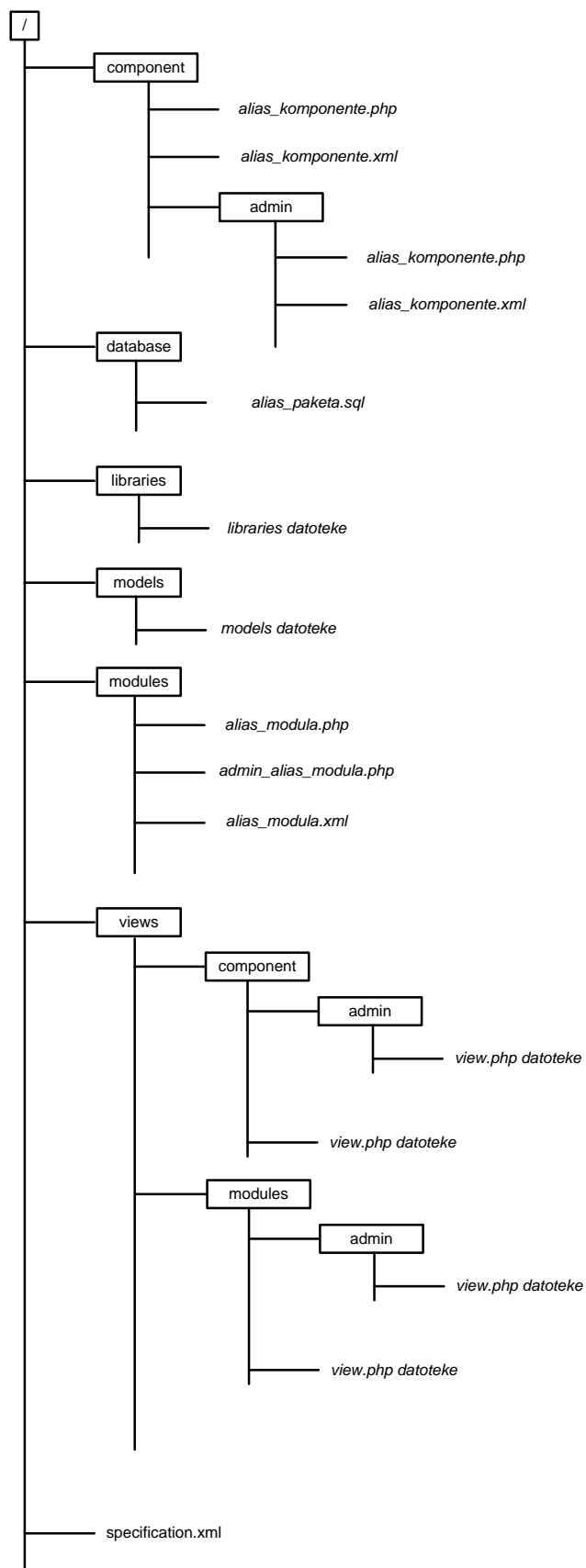
`system/application/modules/packages/aliasPaketa/`

- **modules** – direktorij koji sadrži php datoteke koji predstavljaju module. Dakle moguće je da se u tome direktoriju nalazi više modula. Za svaki modul osim same php datoteke potrebno je definirati xml datoteku koja sadrži podatke koji se koriste prilikom instalacije paketa sa modulom, te php datoteka sa *admin_* prefiksom koja služi za upravljanje podataka koji su vezani za modul prilikom stvaranja instance modula. Tijekom instalacije moduli se prebacuju u direktorij:

`system/application/libraries/packages/aliasPaketa/modules/`.

- **views** – direktorij koji sadrži php datoteke za prikaz sadržaja koji je izgeneriran od strane komponente, modula ili oboje. Unutrašnja struktura datoteka je proizvoljna, ali preporuča se da se u direktorij *component* stavljaju datoteke vezane za komponentu, a u *modules* za module. Tijekom instalacije pogledi se prebacuju u direktorij:

`system/application/views/packages/aliasPaketa/`



Slika 45 - Struktura sadržaja paketa

U samom korijenu paketa nalazi se datoteka naziva *specification.xml* koja sadrži informacije što je sve sadržano u paketu.

Izgled xml datoteke definiran je na sljedeći način:

```
<?xml version="1.0" encoding="UTF-8"?>

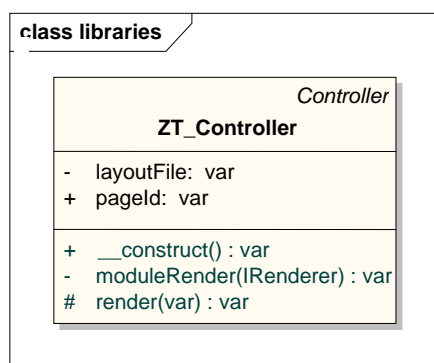
<package>
  <name>Simple pages</name>
  <alias>simplepages</alias>
  <component_spec_file>simplepages.xml</component_spec_file>
  <modules_spec_files>
    <module_spec_file>ipsumbox.xml</module_spec_file>
  </modules_spec_files>
  <database_file>simplepages.sql</database_file>
</package>
```

Svaka *specifiacion* xml datoteka započinje korijenskim elementom *package*. Unutar kojeg su definirani sljedeći elementi:

- *name* – sadrži naziv paketa
- *alias* – skraćeno jedinstveno ime paketa, te služi za lakši pristup određenim dijelovima paketa. Alias ne smije sadržavati praznine i ne tipične znakove. Preporuča se koristiti slova engleske abecede te *_* znak.
- *component_spec_file* – sadrži naziv xml datoteke koja se odnosi na komponentu. Ukoliko paket ne sadrži komponentu tada se navedena oznaka izostavlja.
- *modules_spec_files* – oznaka unutar koje se pojavljuju oznake za module koji su sadržani u paketu. Ukoliko se ne nalazi niti jedan modul u paketu navedena oznaka se izostavlja.
- *module_spec_file* – sadrži naziv xml datoteke koja se odnosi na modul.
- *database_file* – sadrži naziv sql datoteke bilo za komponentu, modul ili oboje. Ukoliko nema potrebe za bazom tada se ta oznaka izostavlja.

3.4.2 Izrada komponente

Gledajući sa strane web aplikacije, mi u biti ne razvijamo komponentu, već tip komponente, dok je sa tehničke strane to komponenta. U načelu svaka komponenta koja se razvija za radno okruženje za upravljanje web sadržajem je kontroler u *Codeigniter* PHP okolini te je moguće koristiti sve značajke navedene okoline. Međutim postoji razlika. U *Codeigniter* PHP okolini svaki kontroler nasljeđuje *Controller* razred. Dok kod izrade komponente svaki kontroler nasljeđuje *ZT_Controller* razred. Za svaki konstruktor u razvijenoj komponenti potrebno je pozvati roditeljski konstruktor.



Slika 46 - Class dirjagram *ZT_Controller-a*

Nasljeđivanjem *ZT_Controller-a* dobiva se naslijeđena metoda *render(\$data)* koja omogućuje generiranje html prikaza. Sve metode *pogleda(engl. view)* koje se pozivaju u javnim metodama razreda, moraju biti pozvane sa parametrom *true* za spremanje međurezultata generiranja html prikaza u međuspremnik.

Izgenerirani html prikaz komponente pohranjuje se u element polja `$data['main_column']`. Ukoliko postoji potreba za generiranjem podataka za *head* element html dokumenta, tada se u element polja `$data['head']` sprema `std_class` objekt *head* koji sadrži članske varijable *title*, *meta*.

Vrlo važno svojstvo komponenta u radnoj okolini za upravljanje web sadržajem, je da se prilikom instanciranja komponente može promijeniti sam naziv komponente,

odnosno kontrolera i pripadajućih metoda. Zato da bi komponenta znala izgenerirati poveznice na svoje metode *ZT_Controller* posjeduje javnu člansku varijablu *\$methodNamees* koja je zapravo asocijativno polje sa preimenovanim nazivima metoda. Dakle indeks elementa polja je naziv originalne metode, a vrijednost je novi naziv metode.

Kod pozivanja metoda *load* koje su sastavni dio *Codeigniter* okoline, mora se navesti staza dohvata datoteka na sljedeći način:

```
packages/alias_package/naziv_php_datoteke
```

Alias package je skraćeni naziv paketa koji se definira u *specifications.xml* datoteci.

Ostatak se razvija na klasičan način koji uvjetuje *Codeigniter* razvojna okolina.

Primjer programskog koda komponente:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Komponenta extends ZT_Controller {

    public function __construct() {
        parent::__construct();

        // učitavanje knjižnica i modela

    }

    public function index(){
        // nekakvo procesiranje i obrada

        $renderData= $this->load->view ("packages/alias_paketa/component/
neki_view", $dataForView, true);
        $data['main_column'] = $renderData;

        $head->title = "naslov";
        $head->meta = "";
        $data['head'] = $head;

        $this->render($data);
    }
}
```

Ukoliko je potrebno ostvariti unos podataka za navedenu komponentu sa administratorskog djela web sjedišta, tada je potrebno ostvariti administratorski kontroler koji će omogućiti navedene operacije. Naziv datoteke mora biti identičan onome komponente koju razvijamo, a smješta se u admin direktorij direktorija *controllers*.

Sve dodatne knjižnice koju koristi razvijena komponente će se nalaziti u `system/application/libraries/packages/aliasPaketa/libraries`.

Nakon što je programski kod komponente dovršen potrebno je izraditi xml datoteku istog naziva kao i naziv php datoteke komponente, uz razliku ekstenzije. Navedena datoteka osim informacija koje su potrebne za instalaciju komponente, sadrži i informacije o metodama i načinima na koji se način te metode mogu mapirati u drugačiji oblik.

Oblik takve xml datoteke je sljedeći:

```
<?xml version="1.0" encoding="UTF-8"?>

<component>
  <name>Simple pages</name>
  <description>Jednostavne stranice</description>
  <alias>simplepages</alias>
  <allow_instances>0</allow_instances>
  <admin>0</admin>
  <methods>
    <method>
      <name>index</name>
      <params>1</params>
      <back>0</back>
      <description>glavna index metoda</description>
      <route>simplepages/index/$1</route>
      <pattern>^{controller}/{method}/{ $1}$</pattern>
    </method>
  </methods>
</component>
```

Svaka xml datoteka komponente započinje korijenskim elementom *component*. Unutar kojeg su definirani sljedeći elementi:

- `name` – sadrži puni naziv komponent
- `description` – sadrži opis komponente

- `alias` – skraćeno jedinstveno ime komponente, te služi za lakši pristup određenim dijelovima komponente. Alias ne smije sadržavati praznine i ne tipične znakove. Preporuča se koristiti slova engleske abecede te `_` znak.
- `allow_instances` – sadrži 1 ukoliko komponenta dozvoljava višestruke instance, 0 inače
- `admin` – ukoliko je razvijena komponenta za administratorski dio
- `methods` – sadrži method elemente koji pobliže opisuju svaku javnu metodu komponente.
- `method` – element koji opisuje javne metode komponente
 - `name` – naziv metode
 - `params` – da li metoda prima kakve parametre
 - `back` – dali metoda služi samo za pozadinsku obradu podataka, odnosno ne ispisuje ništa
 - `description` – opis metode
 - `route` – ruta do kontrolera, odnosno metode. Termin koji se koristi u *Codeigniter* okolini, te se obično definira u `system/application/config/routes.php` datoteci i služi da nekakav obrazac usmjeri na željeni kontroler¹
 - `pattern` – obrazac za prepoznavanje, sadrži regularni izraz. Kada se prepozna obrazac tada se poziva metoda kontrolera preko rute.²

3.4.3 Izrada modula

Moduli se izrađuju na sličan način kao što se izrađuju korisničke knjižnice u *Codeigniter* okolini. Dakle moguće je koristiti sve mogućnosti *Codeigniter* okoline na standardan način. Ali postoji par razlika, od osnovne *Codeigniter* knjižnice.

Svi moduli moraju naslijediti apstraktni razred *Module* i implementirati sučelje *irenderer*, a ukoliko je potrebno zadavati neke parametre prilikom instanciranja

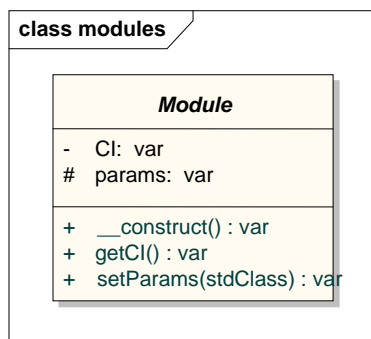
¹ Pogledati *Codeigniter* dokumentaciju za upravljanje rutama, te načinima oblikovanja ruta

² Pogledati *Codeigniter* dokumentaciju za upravljanje rutama, te načinima oblikovanja obrazaca

modula tada je potrebno implementirati knjižnicu koja će omogućiti unos tih parametara, odnosno implementirati sučelje *iadmin*. Navedeni razred i sučelja nalaze se u `system/application/libraries/core/modules` direktoriju. Sam modul koji razvijamo smješta se u direktorij:

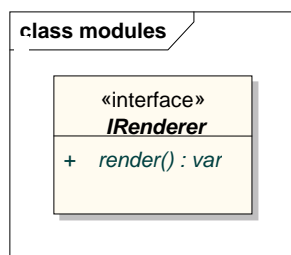
```
system/application/libraries/packages/aliasPaketa/modules  
/
```

Razred *Module* ima dvije javne metode, `getCI()` koja vraća *CodeIgniter* super objekt, koji je nužan da bi smo koristili obilježja *Codeigniter* razvojne okoline, te metodu `setParams($params)` koja prima kao parametar serijalizirane parametre iz baze i postavlja privatnu člansku varijablu `$params` te na taj način modul može koristiti parametre spremljene u bazi. Svaki modul koji razvijamo mora u svojem konstruktoru pozvati konstruktor roditeljskog razreda `parent::__construct()`;



Slika 47 - Class dijagram razreda *Module*

Svaki modul mora implementirati sučelje *IRenderer* koje sadrži metodu `render()` koja vraća podatak string koji sadrži view html prikaz za iscrtavanje modula.



Slika 48 - Class dijagram sučelja Irenderer

Dakle, pozivom metode `render()` će se pokrenuti postupak obrade podataka, te na kraju navedena metoda vraća izgenerirani prikaz u obliku stringa. Slijedi primjer implementacije navedene metode:

```
public function render(){
    $data['menuItems'] = $this->collectData();
    $data['pageLink'] = $this->getCI()->uri->uri_string();
    return $this->getCI()->load->view('packages/menu/modules/menu_' . $this->params->layout_position . '_view', $data, true);
}
```

Važno je napomenuti to da se moduli mogu prikazivati na različitim pozicijama na web stranici, stoga je potrebno osigurati sve prikaze za te razne pozicije. Svaka datoteka prikaza mora biti nazvana na slijedeći način:

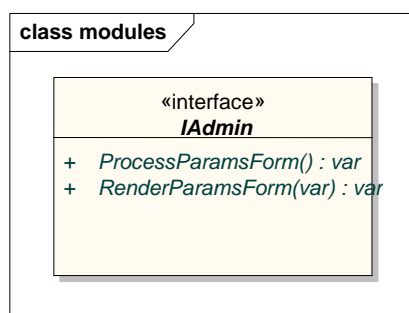
aliasModula_pozicija_view.php

A staza do direktorija gdje je potrebno smjestiti datoteke prikaza modula nalazi se u standardnom view direktoriju, a pristupa se pozivom `view` metode na način:

`packages/aliasPaketa/modules/`

Ukoliko modul prima parametre, odnosno parametre koji se definiraju kod instanciranja modula u administratorskom djelu web sjedišta, tada je potrebno implementirati tzv. admin modul.

Radi se razredu koji ima naziv `admin_aliasModula.php` i nalazi se u istom direktoriju gdje se nalazi i modul. Navedeni razred mora implementirati `iadmin` sučelje koje sadrži dvije metode, `ProcessParamsForm()` koja obrađuje podatke primljene preko web forme iz administratorskog djela, te `RenderParamsForm($params)` koja prima sve podatke o instanciranom modulu iz baze, te na temelju tih podataka generira web formu za prikaz u administraciji modula.



Slika 49 - Class dijagram iadmin sučelja

Sve dodatne knjižnice koju koristi razvijena komponente će se nalaziti u `system/application/libraries/packages/aliasPaketa/libraries`.

Nakon što je programski kod modula dovršen potrebno je izraditi xml datoteku istog naziva kao i naziv php datoteke modula, uz razliku ekstenzije. Navedena datoteka sadrži informacija nužne za uspješnu instalaciju modula.

Oblik takve xml datoteke je slijedeći:

```
<?xml version="1.0" encoding="UTF-8"?>

<module>
  <name>Ipsum box</name>
  <description>ipsum box</description>
  <alias>ipsumbox</alias>
  <allow_instances>1</allow_instances>
  <admin>0</admin>
</module>
```

Svaka xml datoteka komponente započinje korijenskim elementom *modula*. Unutar kojeg su definirani sljedeći elementi:

- `name` – sadrži puni naziv modula
- `description` – sadrži opis modula
- `alias` – skraćeno jedinstveno ime modula, te služi za lakši pristup određenim dijelovima modula. Alias ne smije sadržavati praznine i ne tipične znakove. Preporuča se koristiti slova engleske abecede te `_` znak.
- `allow_instances` – sadrži 1 ukoliko modul dozvoljava višestruke instance, 0 inače
- `admin` – ukoliko je razvijen modul za administratorski dio

4. Zaključak

Prva cjelina rada uvodi pojam Web 2.0 i temeljne probleme upravljanja sadržajem, te probleme razvijanja i održavanja web sjedišta. Kao rješenje se predstavlja radno okruženje za upravljanje web sadržajem, koje treba smanjiti neproduktivne aktivnosti vezane uz tipičan razvoj web primjena. Prezentiraju se osnovni koncepti i značajke sustava za upravljanje sadržajem.

Dalje je obrađen koncept i implementacija programskog okruženja za razvoj dinamičkih web sjedišta, usluga i primjena. Predstavljene su tehnologije, obrasci i paradigme u programskom inženjerstvu koje omogućuju izradu web aplikacije i realizaciju radnog okruženja za upravljanje web sadržajem. Prikazani su i pojedinačno obrađeni bitni elementi potrebni za razvoj web aplikacije za upravljanje web sadržajem, kao što je jezgra radnog okruženja, autentifikacija korisnika, upravljanje pristupom i proširenjima sustava.

Konačno ono što zaokružuje koncept i implementaciju programskog okruženja za razvoj dinamičkih web sjedišta, usluga i primjena je praktično korištenje i razvijanje proširenja sustava. Dane su upute za instalaciju računalnih aplikacija (poslužitelja) za pokretanje web aplikacije, kao i opis instalacije i podešavanja same web aplikacije. Navedene su upute za korištenje radnog okruženja za upravljanje web sadržajem, kao i upute programerima za proširenje postojećeg sustava.

Cijela implementacija radnog okruženja za upravljanjem web sadržaja temeljena je na otvorenom kodu. Smisao ovakvog pristupa leži u činjenici da korištenjem jeftinih i moćnih programskih rješenja, koja su dostupna svima, u kombinaciji sa znanjem programskog inženjerstva, moguće ostvariti vrlo kvalitetnu web aplikaciju. Zbog rapidnog načina programiranja i tipičnih programskih problema korištena je Codeigniter razvojna okolina, koja implementira općenito rješenje za općeniti problem. Codeigniter razvojna okolina pokazala se odličnom prilikom razvoja web aplikacija pisanih u programskom jeziku PHP, jer u dobroj mjeri prikriva nedostatke

programskog jezika PHP uvođenjem objektno – orijentirane paradigme, te primjenom MVC arhitekturnog obrasca.

Sustav za upravljanjem sadržaja pokazao se kao se kao obavezno rješenje bilo kojeg ozbiljnijeg web sjedišta, jer u prvome redu olakšava i automatizira postupak objavljivanja i upravljanja web sadržajem, a sa programerskog stajališta povećava fleksibilnost jer je sustav lako proširiv, lako ga je održavati, te je programski kod moguće jednostavno iskoristavati.

5. Literatura

1. *Martin Brampton: PHP5 CMS - Framework Development*
2. *Web content management system*
URL: http://en.wikipedia.org/wiki/Web_content_management_system
3. *CodeIgniter User Guide Version 1.7*
URL: http://codeigniter.com/user_guide
4. *Role-based access control*
URL: http://en.wikipedia.org/wiki/Role-based_access_control
5. *Managing Hierarchical Data in MySQL*
URL: <http://dev.mysql.com/tech-resources/articles/hierarchical-data.html>

6. Naslov, sažetak i ključne riječi

Naslov:

Radno okruženje za upravljanje web sadržajem

Sažetak:

U završnom radu čitatelja se uvodi kroz pojam Web 2.0 i problema upravljanja sadržajem, nedostatak nadležnosti, problem razvijanja i održavanja web sjedišta. Kao rješenje se predstavlja radno okruženje za upravljanje web sadržajem, koje treba smanjiti neproduktivne aktivnosti vezane uz tipičan razvoj web primjena. U radu su opisani koncepti i implementacija programskog okruženja za razvoj dinamičkih web sjedišta usluga i primjena kroz jezgru radnog okruženja za upravljanje sadržajem, autentifikaciju korisnika, upravljanje pristupom i proširenjima sustava. Dane su upute za instalaciju računalnih aplikacija (poslužitelja) za pokretanje web aplikacije, kao i opis instalacije i podešavanja same web aplikacije. Navedene su upute za korištenje radnog okruženja za upravljanje web sadržajem, kao i upute programerima za proširenje postojećeg sustava.

Ključne riječi:

računarstvo, programiranje, aplikacija, web, Internet, CMS, sustav, baza podataka, mysql, php, codeigniter, framework, okolina, upravljanje

7. Title, Summary and keywords

Title:

Web content management system

Summary:

Within this thesis the reader is introduced to the concept of Web 2.0 and the problem of content management, lack of authorization, the problem of development and maintenance of a website. A web content management system is presented as a solution, which is to decrease unproductive activities associated with typical web application development. Both the concept and implementation of a programming environment for development of dynamic websites, services and applications are described within this thesis through the core of the content management system, user authentication, access control and system expansions. Instructions for installation of computer applications that run this web application (web servers) are also given, as well as the description of installation and setup of the web application itself. Directions for using the content management system are presented, as well as the directions for expanding the existing system.

Keywords:

computing, programming, application, web, Internet, CMS, system, database, mysql, php, codeigniter, framework, management

8. Dodatak A – Popis korištenih engleskih riječi i pojmova

U tablici je dan popis svih engleskih pojmova sa prijevodom, koje su korištene u ovome redu

Tabela 4 - Popis prijevoda engleskih pojmova

Pojam na engleskom	Pojam na hrvatskom
Content Management System	Sustav za upravljanje sadržajem

Role – Based Access Control	Kontrola pristupa temeljena na ulogama
events	Događaji
controller	Programska komponenta Codeigniter okoline koja ima ulogu posredovanja
View	Programska komponenta Codeigniter okoline koja ima ulogu pogleda.
Model	Programska komponenta Codeigniter okoline koja ima ulogu modela.
Firewall	Sigurnosna stijena
Framework	Razvojna okolina
Import	uvoz
Router	usmjernik

9. Dodatak B – Popis popularnih sustava za upravljanje sadržajem

U tablici je dan popis besplatnih popularnih sustava za upravljanje sadržajem napisanih u programskom jeziku PHP.

Tabela 5 - Popis popularnih sustava za upravljanje sadržajem

CMS	Kategorija	Platforma	Podržane baze podataka
Drupal (http://drupal.org/)	Portal	PHP	MySQL ili PostgreSQL
eZ publish (http://ez.no/)	Portal	PHP	MySQL, PostgreSQL ili Oracle
Joomla! (http://www.joomla.org/)	Portal	PHP	MySQL
Wordpress (http://wordpress.org/)	Blog	PHP	MySQL