

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Tomaž Šifrer

**Izdelava sistema za upravljanje spletnih  
vsebin z uporabo ogrodja Django**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Mira Trebar

Ljubljana, 2013



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.





Št. naloge: 00476/2013

Datum: 12.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **TOMAŽ ŠIFRER**

Naslov: **IZDELAVA SISTEMA ZA UPRAVLJANJE SPLETNIH VSEBIN Z  
UPORABO OGRODJA DJANGO**  
**IMPLEMENTATION OF A WEB CONTENT MANAGEMENT SYSTEM  
USING THE DJANGO FRAMEWORK**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje


Tematika naloge:

Za urejanje informacij na spletnih straneh se vse pogosteje uporabljajo sistemi za upravljanje vsebin (Content Management System). Osnovne funkcionalnosti spletne strani definira ponudnik, ostalo delo pa lahko prevzamejo uporabniki sami, tako da po potrebi dodajajo, brišejo in urejajo vsebine, ki jih želijo prikazati na spletu. Na osnovi analize obstoječih sistemov, potrebnih spletnih tehnologij in orodij naj kandidat v diplomskem delu opiše zasnovo in izdelavo takšnega sistema z uporabo ogrodja Django. Rezultate svojega dela naj preveri z urejanjem vsebin spletne strani LRA in predstavi prednosti ter slabosti zasnovane rešitve.

Mentor:

  
doc. dr. Mira Trebar

Dekan:

  
prof. dr. Nikolaj Zimic





## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Tomaž Šifrer, z vpisno številko **63090296**, sem avtor diplomskega dela z naslovom:

*Izdelava sistema za upravljanje spletnih vsebin z uporabo ogrodja Django*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mire Trebar,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 5. septembra 2013

Podpis avtorja:



# Kazalo

**Povzetek**

**Abstract**

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Razvojna orodja in tehnologije</b>	<b>3</b>
2.1	Razvojna orodja . . . . .	3
2.2	Tehnologije . . . . .	3
2.2.1	Django . . . . .	4
2.2.2	SQLite . . . . .	5
<b>3</b>	<b>Načrtovanje in razvoj sistema</b>	<b>7</b>
3.1	Sistem za upravljanje s spletnimi vsebinami . . . . .	7
3.2	Ideja in načrtovanje sistema . . . . .	7
3.3	Podatkovna baza . . . . .	8
3.4	Razvoj sistema . . . . .	11
3.4.1	Struktura strani . . . . .	11
3.4.2	Objave . . . . .	13
3.4.3	Galerije . . . . .	13
3.4.4	Nastavitve . . . . .	14
3.5	Prikaz spletne strani . . . . .	17
3.6	Namestitev sistema . . . . .	18

<b>4</b>	<b>Uporaba sistema za upravljanje z vsebinami</b>	<b>19</b>
4.1	Prijava v sistem . . . . .	19
4.2	Skrbniški del sistema za upravljanje z vsebinami . . . . .	20
4.2.1	Urejanje opisa strani . . . . .	21
4.2.2	Datoteke CSS . . . . .	23
4.2.3	Datoteke Javascript . . . . .	25
4.2.4	Slike . . . . .	25
4.2.5	Komponente . . . . .	26
4.2.6	Jezikovne različice spletne strani . . . . .	28
4.2.7	Uporabniki . . . . .	29
4.3	Uredniški del sistema za upravljanje z vsebinami . . . . .	30
4.3.1	Strani oz. podstrani . . . . .	31
4.3.2	Objave (novice, članki ...) . . . . .	34
4.3.3	Galerija . . . . .	35
4.4	Predstavitev spletne strani LRA . . . . .	36
<b>5</b>	<b>Sklep in ugotovitve</b>	<b>41</b>
	<b>Dodatek A Koda HTML opisa spletne strani</b>	<b>45</b>

# Seznam uporabljenih kratic

**AJAX** ang. *Asynchronous JavaScript and XML*, asinhroni JavaScript in XML

**CMS** ang. *Content Management System*, sistem za upravljanje vsebin

**CSS** ang. *Cascading Style Sheets*, kaskadni stilski slogi

**HTML** ang. *HyperText Markup Language*, označevalni jezik

**HTTP** ang. *Hypertext Transfer Protocol*, prenosni protokol

**PHP** ang. *PHP: Hypertext Preprocessor*, programski jezik PHP

**SQL** ang. *Structured Query Language*, strukturni poizvedbeni jezik



# Povzetek

Sistem za upravljanje spletnih vsebin (CMS - Content Management System) postaja vse bolj zanimiv in uporaben za številne uporabnike, ki želijo sami urejati svoje spletne strani. V diplomski nalogi je predstavljen sistem, ki je zasnovan z uporabo ogrodja Django in omogoča enostavno postavitvev in urejanje spletne vsebine preko brskalnika. Zagotovljen je celoten nadzor nad izgledom spletne strani tako da vključuje tudi dodajanje poljubnih funkcionalnosti, ki so napisane v programskem jeziku Python. Sistem za upravljanje je namenjen dveh skupinam uporabnikov, skrbnikom in urednikom. Skrbnik vspostavi, prilagaja in ureja osnovno strukturo, ki je potrebna za izdelavo spletne strani in urejanje njene vsebine. Urednik skrbi za dodajanje, brisanje in urejanje vsebin na spletni strani. Pri tem ne potrebuje znanja spletnih tehnologij in različnih jezikov za oblikovanje, saj mu sistem ponudi vgrajen urejevalnik besedil. Vsebine se delijo na statične vsebine, objave (novice, članki ...) in galerije. Največja prednost sistema je, da lahko skrbnik izdelava spletno stran in ji dodaja potrebne funkcionalnosti kar preko brskalnika, brez kakršne koli uporabe drugih programskih orodij. Za to potrebuje znanje programskega jezika Python in ogrodja Django. Izdelan sistem za upravljanje z vsebinami je prijazen tako za uporabnike, kot tudi za izdelovalce spletnih strani. V postopku testiranja smo zasnovali in uredili spletno stran Laboratorija za računalniško arhitekturo na osnovi obstoječih podatkov, ki so že objavljeni na spletu.

**Ključne besede:** sistem za upravljanje z vsebinami, CMS, Python, Django, splet



# Abstract

Content Management Systems (CMS) are getting more interesting and useful for many users, who wish to manage their own websites. This dissertation presents a system, based on the Django framework, which enables simple website setup and web content editing in the browser. It includes adding optional functionalities, written in Python, which provides complete control over the website appearance. Content Management System is intended for two groups of users, administrators and content managers. Administrator establishes, adjusts and manages basic structure, which is necessary for creating website and managing its content. Content manager adds, erases and manages web content. That doesn't require the knowledge of web technologies and different markup languages, as the system offers built-in content editor. The web content is divided into static content, posts (news, articles...) and galleries. The biggest advantage of this system is being able to create a website and to add necessary functionalities directly through browser, without any use of other programming tools. To be able to do this, one needs knowledge of programming language Python and Django framework. The designed content management system is friendly for users as well as web page developers. In the process of testing we designed and created a website of the Laboratory for Computer Architecture, based on existing data, which are already published on the web.

**Keywords:**

content management system, CMS, Python, Django, Web



# Poglavje 1

## Uvod

Spletne strani so v zadnjih nekaj letih z razvojem informacijsko komunikacijskih tehnologij postale splošen trend današnje družbe. Ima jo že skoraj vsako podjetje in skoraj vsak posameznik. Razlikujejo se glede na vsebino. Tako poznamo predstavitvene strani oseb ali podjetij, bloge, trgovine, portale, družabne strani oz. družabna omrežja in še bi lahko naštevali. S tem ko so postale vse bolj razširjene in uporabne, se je prav tako povečalo povpraševanje po uporabniških vmesnikih, ki služijo za urejanje spletne vsebine. Nastalo je veliko najrazličnejših sistemov za upravljanje z vsebino CMS (Content Management System). Najbolj poznan in uporabljen sistem za upravljanje z vsebinami je WordPress [6], ki temelji na programskem jeziku PHP (PHP: Hypertext Preprocessor). Le-ta je namenjen velikemu številu uporabnikov, saj omogoča uporabo vtičnikov (ang. plugins), ki osnovnemu sistemu dodajo nove funkcionalnosti. Obstajajo tudi mnogi drugi sistemi za upravljanje z vsebino, ki so namenjeni samo določeni skupini uporabnikov. Nekateri služijo samo za pisanje blogov, drugi za spletno trgovino, spet tretji pa samo za urejanje slikovnih galerij.

Veliko podjetij, ki izdelujejo spletne strani, uporablja svoje sisteme za upravljanje z vsebino. Ti so razviti po njihovih željah in jih lahko enostavneje prilagodijo naročnikovim željam. Najboljša lastnost takega sistema je enostavno dodajanje novih funkcionalnosti, za katere bi pri uporabi drugih sistemov porabili bistveno več časa.

Na podlagi izkušenj s sistemi za upravljanje vsebin (WordPress, Joomla, Django CMS [12]) smo se odločili, da naredimo svoj sistem za upravljanje vsebin. Cilj je, da bi lahko uporabnik hitro in enostavno urejal spletno vsebino brez znanja programiranja. Prav tako bo izdelovalcem spletnih strani omogočal urejanje in spreminjanje oblike in dodajanje novih funkcionalnosti spletne strani. Nastal je sistem za upravljanje vsebin, vključuje:

- strani, podstrani,
- objave (novice, članki ...),
- galerije slik,
- nastavitve (oblika, jeziki, uporabniki ...).

Z vse hitrejšim razvojem spletnih tehnologij, bodo sistemi za upravljanje vsebin postajali vedno boljši in vedno več bomo lahko uredili z njimi, ne da bi potrebovali predhodno znanje programiranja. Ker večina novejših sistemov podpira vtičnike, ki dodajajo funkcionalnosti osnovnim sistemom, jih lahko uporabljamo za vedno več različnih spletnih strani.

# Poglavje 2

## Razvojna orodja in tehnologije

### 2.1 Razvojna orodja

Za pisanje programske kode smo uporabili urejevalnik besedil Sublime Text 2 [5]. Namenjen je pisanju programske kode. Podpira zelo veliko programskih jezikov, med njimi so Python, C, C++, C#, Java, HTML (HyperText Markup Language), CSS (Cascading Style Sheets) in mnoge druge. Omogoča hitrejše pisanje programske kode s svojimi predlogi in je tudi ustrezno obarvana glede na programski jezik v katerem jo pišemo.

Za razvoj sistema za upravljanje z vsebinami smo vključili veliko kode napisane v skriptnem jeziku Javascript [13]. Za odkrivanje sintaktičnih napak kode Javascript, smo v brskalniku Google Chrome [14] uporabili Chrome Developer Tools [10]. To je orodje, ki nam omogoča vpogled v to, kaj se dogaja pri prikazu posamezne spletne strani. Zagotavlja dober nadzor nad kodo Javascript in pove tudi, če je pri nalaganju spletne strani prišlo do kakšne napake.

### 2.2 Tehnologije

Sistem za upravljanje z vsebinami je izdelan s pomočjo ogrodja Django, ki temelji na programskem jeziku Python [17]. Spletna stran za prikaz uporablja HTML5 (HyperText Markup Language 5) [16], ki skrbi za prikaz vsebine. Uporablja tudi

CSS3 (Cascading Style Sheets 3) [15], ki omogoča ustrezno oblikovanje vsebine. Da smo spletni strani dodali še nekaj dodatnih funkcionalnosti, kot na primer obvestilo pri brisanju, smo uporabili knjižnico jQuery [11], ki je nekakšna razširitev programskega jezika Javascript. Uporabili smo tudi AJAX (asynchronous JavaScript and XML) [18], ki je skupina medsebojno povezanih spletnih tehnik, uporabljenih za razvoj interaktivnih spletnih aplikacij. Te tehnike omogočajo spletnim stranem oz. aplikacijam, da lahko ažuriramo del spletne strani, brez nepotrebnega nalaganja celotne spletne strani. To pripomore k boljšemu delovanju spletne strani, saj namesto da bi ponovno naložili celotno spletno stran, naložimo samo tisto, kar je spremenjeno glede na zahtevo, ki je bistveno manjše kot celotna spletna stran. To se najbolj očitno pri počasnejših internetnih povezavah.

### 2.2.1 Django

Django je odprtokodno ogrodje za izdelavo internetnih aplikacij napisano v programskem jeziku Python. Temelji na principu model-pogled-nadzornik. Glavni namen je poenostavitev izdelave velikih spletnih aplikacij in povezovanja s podatkovno bazo. Poudarek je na zmanjševanju podvajanja programske kode. To pomeni, da lahko eno funkcijo uporabimo večkrat in nam je ni potrebno kopirati [2]. Namenjen je izdelavi celovitih spletnih rešitev. Javnosti je bil predstavljen leta 2005 in njegova uporaba se je zelo hitro razširila. Sedaj pri razvoju sodeluje zelo veliko število ljudi, saj je ogrodje odprtokodno in ga razvijajo različni programerji po svetu. Django vsebuje nekaj najosnovnejših modulov za pomoč pri razvoju. Razvit ima modul za avtentikacijo uporabnika, ki poskrbi, da uporabnika enostavno prijavimo v sistem. Vsebuje še veliko ostalih modulov, ki pripomorejo k lažjemu razvoju aplikacije.

Django CMS je sistem za upravljanje s spletnimi vsebinami, katerega smo uporabili za zgled. Sistem omogoča urejanje vsebine preko brskalnika. Omogoča večjezičnost spletne strani, vendar ne omogoča drugačnih struktur spletne strani za različne jezike. Deluje na principu vtičnikov, saj je vsaka vsebina, ki jo dodamo na stran, svoj vtičnik. Vtičniki so funkcionalnosti, ki jih moramo izdelati in jih nato preko brskalnika dodati v sistem. S tem pridobimo dinamično značilnost, ker

imamo odprte možnosti za nadgradnjo sistema.

### 2.2.2 SQLite

SQLite [3] je sistem za upravljanje relacijskih baz podatkov, ki je predstavljen v zelo majhni knjižnici, napisani v programskem jeziku C. Uporablja se predvsem za shranjevanje majhnega števila podatkov, ker ne omogoča najboljšega pregleda nad vsemi podatki. Uporabljamo ga lahko tudi za veliko število podatkov, vendar je pregled nad le-timi otežen. Sistem ne potrebuje dodatne storitve za pridobivanje in shranjevanje podatkov, tako kot jo potrebujejo običajni strežniki SQL. Vse podatke zapisuje v eno datoteko na trdem disku. Tako lahko doseže pri majhni količini podatkov celo hitrejši prenos oz. iskanje podatkov, kot običajne podatkovne baze. Pri velikih količinah podatkov pa se hitrost skorajda ne razlikuje od ostalih podatkovnih baz. Za SQLite smo se odločili, ker je enostaven in ne potrebuje posebnih priprav za uporabo. Vse kar je potrebno storiti za začetek uporabe je, da v nastavitvah projekta Django povemo pot do datoteke, ki jo bomo uporabili za podatkovno bazo.



# Poglavje 3

## Načrtovanje in razvoj sistema

### 3.1 Sistem za upravljanje s spletnimi vsebinami

Sistem za upravljanje z vsebinami (ang. Content Management System - CMS) [4] je sistem, ki omogoča urejanje vsebine spletnih strani brez znanja označevalnega jezika HTML. Urednik spletne strani lahko dodaja in spreminja besedila, dodaja novice, ureja galerije in druge elemente brez posredovanja osebe ali podjetja, ki je spletno stran izdelalo. Zaradi enostavne uporabe so takšni sistemi vedno bolj popularni.

Sistemi za upravljanje z vsebinami so zelo različni. Nekateri so dinamični (WordPress, Joomla ...) in omogočajo uporabo vtičnikov (ang. plugins). To pomeni, da lahko dodajamo poljubne funkcionalnosti že obstoječemu sistemu, ki jih potrebujemo za pravi prikaz spletne strani.

### 3.2 Ideja in načrtovanje sistema

Ideja je bila narediti sistem, ki omogoča urejanje celotne spletne strani preko brskalnika. Omogočati mora urejanje vsebine, oblike in funkcionalnosti spletne strani. Podpirati mora dve skupini uporabnikov, to sta skrbnik in urednik. Skrbnik je napreden uporabnik, ki ima nadzor nad sistemom in celotno spletno stranjo. Lahko ureja vsebino, dodaja funkcionalnosti in spreminja obliko. Urednik je upo-

rabnik, ki ima omogočeno samo urejanje vsebine.

Naredili smo analizo sistemov za upravljanje z vsebinami, da bi ugotovili, kateri so osnovni elementi spletne strani. Pregledali smo sisteme WordPress [6], Joomla [9] in Django CMS [12]. Ugotovili smo, da so najbolj pomembni elementi sistema urejanje vsebine, novic, galerij in osnovne nastavitve sistema. Pod osnovne nastavitve sistema spadajo nastavitve jezikovnih različic spletne strani, urejanje skrbnikov in urednikov ter urejanje oblike spletne strani. Ugotovili smo tudi, da so sistemi precej zapleteni za uporabo. Uporabniki potrebujejo precej časa, da se naučijo uporabljati sistem tako obsežen kot so analizirani sistemi. Vsi analizirani sistemi so prosto dostopni, saj je to tudi cilj naše rešitve. Razlika med prosto dostopnimi in plačljivimi sistemi je samo v tem, da imajo plačljivi zagotovljeno podporo uporabnikom, medtem ko pri prosto dostopnih težave rešujejo uporabniki sami. Za izdelavo svojega sistema smo se odločili zato, ker želimo narediti enostaven sistem, ki bo omogočal izdelavo in urejanje celotne spletne strani preko brskalnika

### 3.3 Podatkovna baza

Ogrodje Django ima zelo dobro izdelano komuniciranje s podatkovnimi bazami in ni pomembno katerega ponudnika podatkovne baze bomo uporabili. Pomembno je samo to, da tabele definiramo v programski kodi. Nato ogrodje samo kreira definirane tabele in jih samodejno ustvari v podatkovni bazi, ki smo jo izbrali v nastavitvah ogrodja. Nastavitve so odvisne od strežnika, kjer gostimo spletno stran in podatkovno bazo.

Za ponudnika podatkovne baze smo izbrali SQLite [3]. Podatkovna baza sestoji iz 14 tabel:

- Language - v njej so shranjeni podatki o jezikovnih različicah spletne strani.
- Page - v njej so shranjeni podatki o straneh oz. podstraneh.
- Content - v njej so shranjeni podatki o vsebini.

- PageContent je povezovalna tabela med tabelo Page in tabelo Content. Pove nam katere vsebine so na kateri strani/podstrani.
- Layout - v njej so shranjeni opisi strani v kodi HTML.
- Part - v njej so shranjena vsa območja, ki so vezana na obliko strani.
- CSSFile - v njej so shranjene poti do datotek s kodo CSS.
- JSFile - v njej so shranjene poti do datotek s kodo Javascript.
- ImgFile - v njej so shranjene poti do slik, ki jih uporabljamo za obliko spletne strani.
- Control - v njej so shranjene oblike komponent in poti do datotek s programsko kodo.
- PostCategory - v njej so shranjene vse kategorije objav.
- Post - v njej so shranjene vse objave, ki so vezane na kategorijo.
- Gallery - v njej so shranjene vse galerije.
- GalleryImage - v njej so shranjene poti do slik, ki so vezane na posamezno galerijo.
- User - v njej so shranjene vsi uporabniki sistema CMS.
- Group - v njej so shranjene vse skupine uporabnikov (skrbnik in urednik).
- UserGroups - je povezovalna tabela, ki nam pove kateri uporabnik spada v katero skupino uporabnikov.

Slika 3.1 prikazuje model podatkovne baze sistema z uporabo spletnega orodja WWW SQL Designer [7]. S povezavami med tabelami so določene odvisnosti tabel med sabo. Določajo tuje ključe tabel, preko katerih nato sistem prepozna, da so nekateri podatki odvisni od drugih.



## 3.4 Razvoj sistema

Razvili smo sistem, ki omogoča urejanje celotne spletne strani. Naloga je bila zapletena, saj je sistem dinamičen, kar pomeni, da smo izdelali bolj zapletene funkcije in metode. Slika 3.2 prikazuje glavo sistema, ki smo ga poimenovali DerpCMS.



Slika 3.2: Glava sistema za upravljanje z vsebinami.

### 3.4.1 Struktura strani

Struktura strani ("Pages") služi dodajanju in urejanju spletne vsebine. Vsebino lahko pišemo v različnih jezikih, zato sistem omogoča večjezičnost spletne strani. Veliko sistemov, ki omogočajo večjezičnost, imajo narejeno tako, da se obkljuka, v katere jezike želimo prevesti vsebino. Nato se pri vnosu vsebine prikaže toliko vnosnih polj, kolikor smo izbrali jezikov in v njih vnesemo prevedeno vsebino. S takšnim načinom je zelo težko narediti dve popolnoma drugačni jezikovni spletni strani. Nekatere spletne strani ponujajo drugačne vsebine za na primer Evropo kot za na primer Ameriko. Temu imajo tudi prilagojeno spletno strukturo strani. Če bi uporabili zgoraj opisan sistem, tega ne bi mogli doseči, saj nam sistem ne pusti urejati strukture glede na jezikovno različico, ampak nam samo prikaže drugo vsebino.

Mi smo to rešili tako, da najprej določimo katere jezike želimo imeti. Ko jih določimo, lahko naredimo popolnoma novo spletno stran in nato strukturo gradimo na podlagi posamezne jezikovne različice. Slabost tega pristopa je, da imamo v primeru večjezičnih strani veliko več dela, saj moramo ponovno nastaviti celotno strukturo spletne strani.

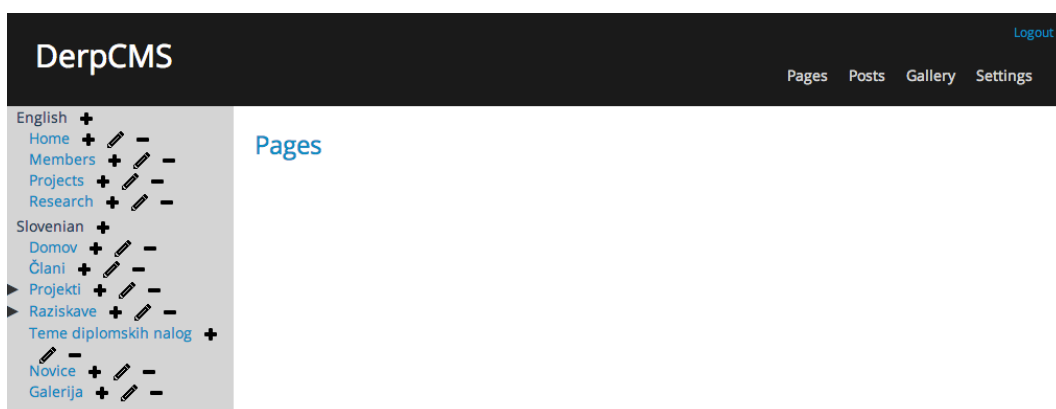
Posamezna spletna stran oz. podstran je vezana na obliko strani. S tem povemo, kako bo izgledala in katera območja bo imela, kamor bomo nato dodajali vsebino. Določiti ji je potrebno ime, ki mora biti skrbno izbrano, saj bo le to

vidno v meniju, ampak le v primeru, da jo bomo tam želeli prikazati. Napisati moramo tudi pot do le-te, saj drugače sistem ne more vedeti, kako jo prikazati. Če želimo, da je stran prikazana v meniju na spletni strani, moramo le to izbrati. Če izberemo, da je trenutna stran glavna oz. prva stran, to pomeni, da je trenutna stran vstopna spletna stran. To je stran, katero uporabnik prvič vidi, ko pride na povezavo spletne strani.

Na posamezno spletno stran moramo omogočiti dodajanje vsebine. Le-te so vezana na območja, ki smo jih definirali v obliki spletne strani. Tako jih lahko poljubno dodajamo v posamezna območja. Implementirani so štirje različni tipi:

- vsebina HTML,
- objava (novica, članek...),
- galerija in
- komponenta (ang. components).

Na sliki 3.3 vidimo glavno stran za urejanje vsebin. Ta se uporablja tudi kot vstopna stran v sistem za upravljanje z vsebinami. To pa zato, ker v praksi največkrat uporabljamo prav ta del sistema, razen v primeru portala, ki vsebuje novice. V takem primeru se največkrat uporabljajo objave.



Slika 3.3: Glavna stran za urejanje strani.

### 3.4.2 Objave

Ta del sistema služi urejanju objav ("Posts") (slika 3.4). Omogočeno je vstavljanje novic, blogov ali kakršnih koli drugih vsebin, ki so vezane na datum ali jih želimo dinamično prikazovati. Objave so lahko tudi dogodki ali koncerti, ki so vezani na datum in jih želimo vedno znova dodajati in prikazovati samo tiste, ki so atraktivni na določen dan. Ker nekateri sistemi za upravljanje z vsebinami omogočajo dodajanje samo novic ali samo bloga, smo se odločili, da naredimo tako, da bo vsaka objava spadala pod svojo kategorijo. V tem primeru lahko naredimo poljubno število kategorij, ki imajo poljubna imena. Tako lahko naredimo kategorijo, v kateri imamo novice in drugo kategorijo, ki nam služi kot blog. Nato posamezno objavo dodamo v poljubno kategorijo. Ker je vrstni red objav zelo pomemben, smo tudi omogočili urejanje zaporedja s pomočjo zaporedne številke navzdol in navzgor.



Slika 3.4: Stran za urejanje novic.

### 3.4.3 Galerije

Galerije ("Gallery") omogočajo predstavitev večjega števila fotografij. Zato so sestavljene iz imena, ki je v bistvu nekakšna kategorija in iz posameznih slik. Za nalaganje slik smo morali razviti nov vmesnik. Narejen je iz že obstoječega vtičnika jQuery [11], ki smo ga nato povezali s programsko kodo. Vtičnik za nalaganje slik omogoča dodajanje več slik hkrati in je zato nalaganje slik v galerijo precej enostavno. Vtičnik deluje tako, da ko izberemo želene slike, preko AJAX klica izvršimo metodo, ki nato poskrbi za shranitev slik na strežnik. Metoda deluje tako, da se slike najprej shranijo v začasno mapo *temp*. Po kliku na gumb shrani, se pokliče metoda, ki prepozna katere slike je uporabnik izbral in podatke o sliki

zapiše v podatkovno bazo. Ko so uspešno shranjene, jih skopiramo v mapo, kjer imamo shranjene vse galerije (mapa *gallery*), sliko iz začasne mape pa pobrišemo, da sprostimo prostor na disku. Na sliki 3.5 vidimo glavno stran galerij.



Slika 3.5: Glavna stran galerij.

### 3.4.4 Nastavitve

Nastavitve ("Settings") so namenjene urejanju oblike in funkcionalnosti spletne strani. Omogočajo urejanje jezikovnih različic, dodajanje različnih komponent in urejanje oblike. Slika 3.6 prikazuje glavo sistema s celotno navigacijo, ki jo vidi samo skrbnik.



Slika 3.6: Glava sistema z prikazano celotno navigacijo.

### Opis strani

V tem delu sistema urejamo opis spletne strani ("Layout") preko brskalnika. Spletna stran ima lahko drugačen opis na posameznih podstraneh. Zato je omogočeno dodajanje večih opisov. Opis je koda HTML, ki jo shranimo namesto v datoteko s končnico ".html" v podatkovno bazo, da jo nato lažje urejamo.

Na sliki 3.7 vidimo primer enostavnega opisa spletne strani. Razdeljen je na tri območja, ki so definirana kot "[:navigacija:]", "[:vsebina:]", "[:noga:]". Pred-

stavljajo posamezne dele spletne strani, kjer bodo nato prikazane vsebine, ki jih bomo vnesli. Tako bomo na primer v območju, ki je definirano kot "[:navigacija:]", prikazali glavni meni spletne strani, v območju "[:vsebina:]" bomo prikazali vsebino in v območju "[:noga:]" bomo dodali kontaktne informacije. Da bi lahko urejali območja in pri tem ne bi izgubili vsebin, ki bodo vezana na območja, kjer se bodo prikazovala, smo morali območja posameznega opisa shraniti v posebno tabelo v podatkovni bazi. Shraniti smo morali kateremu opisu posamezno območje pripada, kakšno je ime območja, ki se bo nato uporabljal pri dodajanju strani, kakšno ime smo uporabili v kodi HTML za označitev območja in ali je območje privzeto ali ne. V primeru da je privzeto to pomeni, da se bo v njem prikazala vsebina, ki ji ni mogoče določiti območja.

```
<html>
  <body>
    <header>
      <nav>
        [:navigacija:]
      </nav>
    </header>
    <section>
      [:vsebina:]
    </section>
    <footer>
      [:noga:]
    </footer>
  </body>
</html>
```

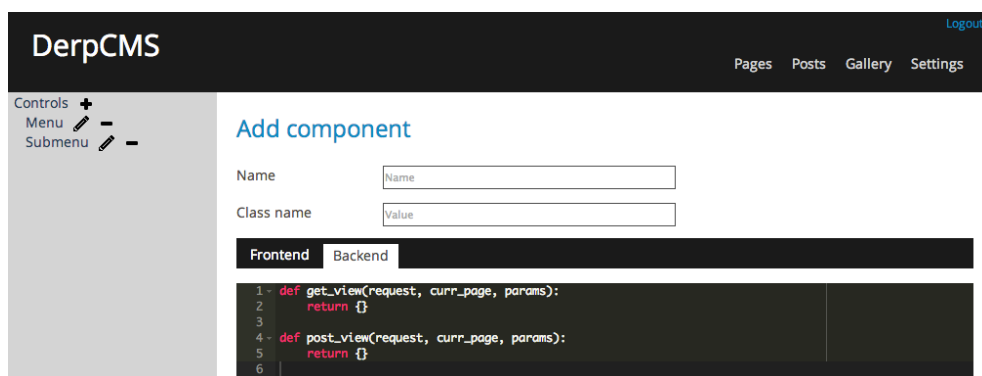
Slika 3.7: Primer kode HTML, ki definira opis spletne strani.

Za ločevanje elementov oz. kode od preostanka kode HTML smo uvedli nove označbe (ang. tag). Označbe se začnejo z "[:" in končajo z ":]". Vse kar je napisano znotraj označb, se prevede v programsko kodo.

### Komponente

Komponente ("Components") omogočajo spreminjanje funkcionalnosti sistema preko spletnega vmesnika. Funkcionalnosti, ki jih lahko dodajo obstoječemu sistemu za upravljanje z vsebinami, so odvisne od spletne strani, ki jo izdelujemo.

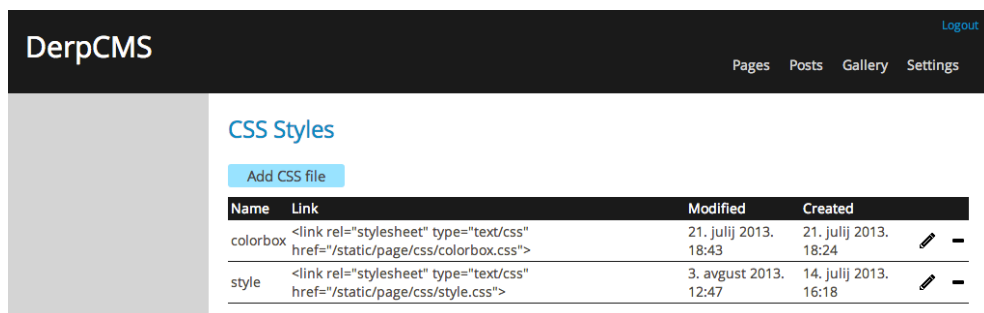
Lahko naredimo meni, kontaktni obrazec, kalkulator ali kakršno koli drugo funkcionalnost, ki jo želimo imeti. Delujejo tako, da ob prikazu spletne strani pokličemo metodo *get\_view* ali *push\_view* (slika 3.8). Sistem metodo izbere na podlagi klica spletne strani. Če stran naložimo preko klica *GET* (metoda HTTP, ki pošlje zahtevek za pridobitev vsebine iz določenega naslova) se izvede metoda *get\_view*, če pa stran naložimo preko klica *POST* (metoda HTTP, ki pošlje podatke na določen naslov za nadaljnjo obdelavo) se pokliče metoda *post\_view*. Ko se izbrana metoda izvede, sistem prikaže opis komponente s podatki, ki smo jih pridobili v metodi.



Slika 3.8: Metodi, ki se izvedeta ob prikazu komponente na spletni strani.

### Datoteke CSS, Javascript in slike

Sistem za upravljanje z vsebinami smo naredili tako, da je preko njega mogoče urejati celotno spletno stran. V to spadajo tudi datoteke s kodo CSS (slika 3.9). Za kreiranje datoteke smo uporabili urejevalnik besedil Ace [8]. Ko napišemo željeno kodo in jo shranimo, se na disku kreira nova datoteka z imenom, ki smo ga vpisali v vnosno polje. To datoteko lahko tudi kasneje urejamo. Omogočeno je tudi nalaganje že obstoječih datotek s kodo CSS, preko nalagalnega vmesnika, ki je opisano v poglavju 3.5.



Slika 3.9: Stran za dodajanje in urejanje datotek CSS.

## 3.5 Prikaz spletne strani

Ko smo realizirali vse dele sistema za upravljanje z vsebinami, smo morali vse skupaj še povezati. Povezava je implementirana metoda v programskem jeziku Python, katera najprej preveri katero stran želimo prikazati. Preko poti iz podatkovne baze izbere pravilno stran. Ker je vezana na obliko strani, enostavno dobimo želeno obliko. Nato izberemo vse vsebine in jih dodamo v območje na obliki strani, ki smo ga določili pri dodajanju vsebine. Če imamo med vsebino tudi objave, metoda to zazna in iz podatkovne baze pridobi vse objave, ki ustrezajo parametrom, ki smo jih nastavili ob dodajanju objav kot vsebino. Sistem prav tako zazna galerije in jih pridobi iz podatkovne baze. Nato jih postavi v mrežo in doda v določeno območje. Ko je vsa vsebina v potrebnih območjih, le to prikažemo uporabniku.

V primeru, da smo med vsebino dodali kakšno komponento, sistem dodajanja komponent med vsebino deluje nekoliko drugače. Sistem najprej doda vso kodo HTML v določeno območje, ki smo ga izbrali pri dodajanju vsebine oz. komponente. Nato pokliče ustrezno metodo v komponenti in prikaže potrebno vsebino.

## 3.6 Namestitev sistema

Za delovanje sistema potrebujemo ogrodje Django (verzija 1.5.1) [1]. Ko ga namestimo na strežnik, moramo namestiti še knjižnico django-dajaxice (verzija 0.5.5) [19], ki omogoča komuniciranje med strežniško in Javascript kodo preko klicov AJAX.

Za objavo sistema na spletu je najpogosteje uporabljen spletni strežnik Apache HTTP Server [20]. Za testiranje sistema smo uporabili neobsežen razvojni strežnik, ki je del ogrodja Django. Je najprimernejši za testiranje, saj ne potrebuje nobene konfiguracije ali dodatnega programa.

## Poglavje 4

# Uporaba sistema za upravljanje z vsebinami

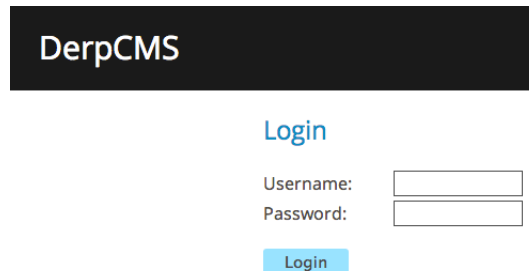
Sistem za upravljanje z vsebinami je namenjen dvema skupinama uporabnikom: skrbnikom in urednikom. Prioriteta je na urednikih vsebin, ki v večini primerov ne znajo programskih ali oblikovnih jezikov, zato je sistem prirejen tako, da namesto oblikovnih programskih jezikov, uporabljamo urejevalnike besedila, ki samodejno generirajo oblikovno programsko kodo. Uredniki nimajo omogočenega urejanja oblik in funkcionalnosti spletne strani. Del sistema, ki je namenjen skrbnikom, pa zahteva znanje programskih in oblikovnih jezikov. Skrbniki imajo popoln nadzor nad spletno stranjo.

Za primer uporabe sistema smo se odločili, da bomo prenovili spletno stran Laboratorija za računalniško arhitekturo. Obliko spletne strani smo zasnovali sami, vsebina pa je prenešana iz obstoječe spletne strani [21].

### 4.1 Prijava v sistem

Vstop v sistem za upravljanje z vsebinami je zavarovan z uporabniškim imenom in geslom. Do sistema DerpCMS dostopamo preko naslova `"/admin/"` (npr.: <http://www.primer.com/admin/>). Sistem nas avtomatsko preusmeri na stran za prijavo v sistem (slika 4.1). V polje `"Username"` vpišemo uporabniško ime in v

polje "Password" vpišemo geslo. Nato se s klikom na gumb "Login" prijavimo v sistem.



Slika 4.1: Prijava v sistem.

## 4.2 Skrbniški del sistema za upravljanje z vsebinami

V skrbniški del sistema za upravljanje z vsebinami spadajo funkcije sistema, ki pripomorejo k postavitvi, izgledu in funkcionalnosti spletne strani. Nastavimo lahko v koliko jezikov bo spletna stran prevedena. Naredimo vse postavitve in določimo območja, v katerih se bo prikazovala vsebina. Preko sistema uredimo ali naložimo vse potrebne datoteke za pravilen izgled spletne strani. Prav tako lahko izdelamo dodatne funkcionalnosti spletne strani. Za uporabo tega območja potrebujemo znanje programskega jezika Python in ostale jezike za postavitev spletne strani (HTML, Javascript, CSS...). Slika 4.2 prikazuje meni, ki se odpre skrbniku spletne strani.



Slika 4.2: Meni, ki se odpre skrbniku spletne strani.

### 4.2.1 Urejanje opisa strani

Opis strani služi izgledu spletne strani z vsebinami v kodi HTML, s pravilno umeščenimi označbami za območja, kamor se bo prikazovala.

Na sliki 4.3 vidimo spisek vseh opisov, ki so definirani. Polje "Name" nam pove, kako smo opis poimenovali. To ime je uporabljeno pri dodajanju novih strani, da lažje izberemo pravi opis. V polju "Parts" imamo spisek vseh območij, ki smo jih definirali v opisu spletne strani. Vsaka spletna stran ima ponavadi vsaj dva opisa strani. Vstopna stran (vstopni opis) in vsebinska stran (vsebinski opis). V našem primeru imamo definirani dve različni vstopni strani. Vsak opis služi za svojo jezikovno različico spletne strani.

#### Layouts

Add layout

Name	Parts	Modified	Created	
Index	Desna vsebina Meni Noga Vsebina	10. avgust 2013. 09:58	20. julij 2013. 18:29	-
Index EN	Desna vsebina Meni Noga Vsebina	11. avgust 2013. 23:58	11. avgust 2013. 23:58	-

Slika 4.3: Postavitev spletne strani.

Opis dodamo s klikom na gumb "Add layout". Odpre se nam vnosna maska za dodajanje in urejanje (slika 4.4). Sestavljena je iz treh elementov: ime, vsebina in območja.

Ime opisa "Name" moramo izbrati tako, da bomo v nadaljnji uporabi sistema vedeli, kakšen je posamezen opis strani. Pametno je izbrati ime, ki na kratko povzema opis. Na primer vstopna stran, vsebinska stran ali galerija. V primeru smo jih poimenovali *Index* in *Index EN*. *Index* je v primeru hkrati vstopni in vsebinski opis strani za slovensko jezikovno različico. *Index EN* je isti kot *Index*, vendar je prirejen za angleško jezikovno različico. Ločili smo jih zgolj zaradi jezikovne različice, saj imamo v primeru v opisu strani definiran naslov spletne strani, ki ga

## 22 POGLAVJE 4. UPORABA SISTEMA ZA UPRAVLJANJE Z VSEBINAMI

moramo prevesti v oba jezika.

Vsebina je koda HTML, ki nam pove kakšen je opis spletne strani. V njej moramo določiti vsa območja, v katerih bo prikazana vsebina. Določimo jih s pomočjo označbe "[ : ]".

Območja moramo definirati vsako posebej. Če želimo dodati območja, kliknemo na gumb "Add part". S klikom na gumb se nam v tabeli doda nova vrstica za vnos območja. Območje je sestavljeno iz treh vnosnih polj. Prvo vnosno polje je ime območja ("Name"). Ime izberemo tako, da bomo v nadaljnji uporabi vedeli, kaj to območje predstavlja. V polje "Value" vnesemo besedo, katero smo uporabili v označbi območja. Če območje označimo kot "Default" to pomeni, da se v nekaterih primerih vsebina (npr.: vsebina celotne objave) prikaže v tem območju.

### Add layout

Name

```
1 - <html>
2 -   <body>
3 -     <header>
4 -       <nav>
5 -         [:navigacija:]
6 -       </nav>
7 -     </header>
8 -     <section>
9 -       [:vsebina:]
10 -    </section>
11 -    <footer>
12 -      [:noga:]
13 -    </footer>
14 -  </body>
15 - </html>
```

Name	Value	Default
<input type="text" value="Glavni meni"/>	<input type="text" value="navigacija"/>	<input type="radio"/>
<input type="text" value="Vsebina"/>	<input type="text" value="vsebina"/>	<input checked="" type="radio"/>
<input type="text" value="Noga strani"/>	<input type="text" value="noga"/>	<input type="radio"/>

Slika 4.4: Vnosna maska za dodajanje postavitev strani z enostavnim primerom.


## 4.2.2 Datoteke CSS

Vsaka spletna stran uporablja datoteke, ki vsebujejo kodo CSS, za oblikovanje vsebine na spletni strani. V njej so določene velikosti pisave, barve, odmiki in razmiki med besedilom in slikami in drugo.

Slika 4.5 prikazuje spisek vseh datotek, ki vsebujejo kodo CSS za spletno stran. Če želimo dodati novo datoteko, kliknemo na gumb "Add CSS file". Odpre se nam vnosna maska za kreiranje ali prenos datoteke (slika 4.6). Vnosna maska je razdeljena na dva dela. Prvi del je namenjen kreiranju nove datoteke ("Create new"), drugi pa je namenjen prenosu datoteke iz računalnika ("Upload").

### CSS Styles

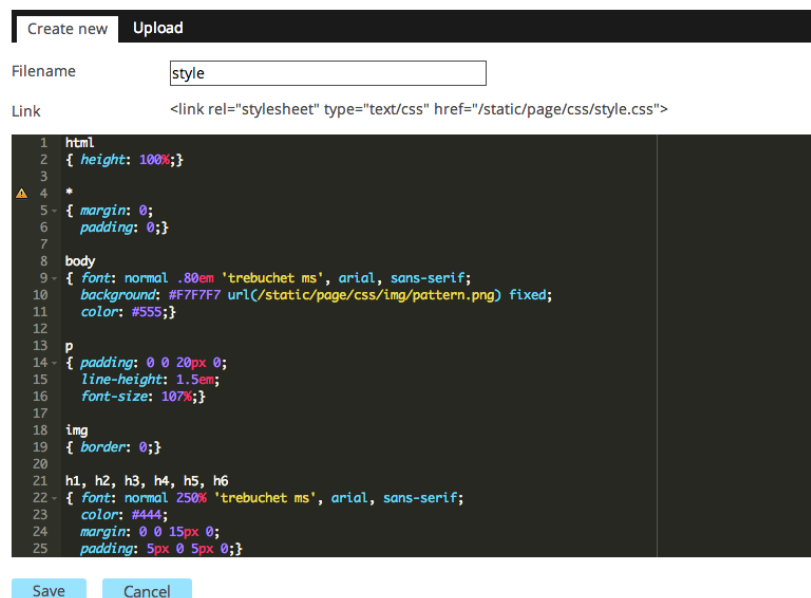
Add CSS file

Name	Link	Modified	Created		
colorbox	<link rel="stylesheet" type="text/css" href="/static/page/css/colorbox.css">	21. julij 2013. 18:43	21. julij 2013. 18:24		
style	<link rel="stylesheet" type="text/css" href="/static/page/css/style.css">	20. julij 2013. 18:52	14. julij 2013. 16:18		

Slika 4.5: Pregled obstoječih oblik spletne strani.

Pri kreiranju nove datoteke CSS moramo izbrati ime in napisati vsebino datoteke (slika 4.6). Ime datoteke ("Filename") sme vsebovati samo črke, številke in/ali znak "-". V glavno polje vnesemo kodo CSS. Ko smo končali, kliknemo na gumb "Save".

## Add CSS file



Filename

Link

```

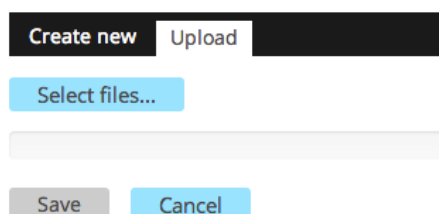
1  html
2  { height: 100%;}
3
4  *
5  { margin: 0;
6    padding: 0;}
7
8  body
9  { font: normal .80em 'trebuchet ms', arial, sans-serif;
10 background: #F7F7F7 url(/static/page/css/img/pattern.png) fixed;
11 color: #555;}
12
13 p
14 { padding: 0 0 20px 0;
15 line-height: 1.5em;
16 font-size: 107%;}
17
18 img
19 { border: 0;}
20
21 h1, h2, h3, h4, h5, h6
22 { font: normal 250% 'trebuchet ms', arial, sans-serif;
23 color: #444;
24 margin: 0 0 15px 0;
25 padding: 5px 0 5px 0;}

```

Slika 4.6: Vnosna maska za dodajanje nove datoteke CSS in primer vsebine v datoteki style.css

Če želimo prenesti datoteko iz računalnika izberemo "Upload" (slika 4.7). S pomočjo upravitelja datotek izberemo željene datoteke za samodejen prenos na strežnik. Ko je prenos končan, moramo vse skupaj potrditi s klikom na gumb "Save", drugače sistem ne bo shranil datotek v podatkovno bazo.

## Add CSS file



Slika 4.7: Vnosna maska za prenos datoteke CSS iz računalnika na strežnik.

Ko smo izdelali ali naložili novo datoteko, jo je potrebno uporabiti tudi v

opisu. Najprej iz tabele, kjer imamo prikaz vseh datotek, kopiramo pot do datoteke ("Link"). Nato gremo na željen opis in dodamo kopirano kodo (slika 4.8). Tako je datoteka uspešno uporabljena v opisu strani.

## Edit template



Slika 4.8: Opis spletne strani, ki smo mu dodali pot do datoteke CSS.

### 4.2.3 Datoteke Javascript

Sistem omogoča tudi kreiranje datotek, ki vsebujejo programsko kodo Javascript. Te datoteke se dodajajo na isti način kot datoteke CSS. To pomeni, da jih lahko prav tako ustvarimo ali pa jih naložimo iz svojega računalnika. Vnosna maska je podobna kot pri datotekah CSS, razlikuje se le v urejevalniku besedil. Ta je prilagojen za programski jezik Javascript. Na novo izdelano ali prenešeno datoteko moramo nato uporabiti v postavitvi tako kot datoteko CSS.

### 4.2.4 Slike

V datotekah s kodo CSS so velikokrat tudi prisotne poti do slik, ki se uporabljajo za ozadja ali kakšne druge namene. Slike, ki jih uporabimo na spletni strani je potrebno naložiti na strežnik.

Na sliki 4.9 vidimo podatke o slikah, ki smo jih že naložili na strežnik. Ko kliknemo na gumb "Add image" se nam odpre vnosna maska za prenos slik (slika 4.10). S pomočjo upravitelja datotek izberemo slike, ki jih želimo prenesti na strežnik. Samodejno se nam prenesejo in po kliku na gumb "Save" se nam slike shranijo še v podatkovno bazo. Nato jih je potrebno uporabiti v datotekah s kodo

CSS ali v postavitvi strani.

Images

Add image

Name	Path	Created	Modified	
Antena_3D_tmb1	/static/page/css/img/Antena_3D_tmb1.jpg	10. avgust 2013. 09:32	10. avgust 2013. 09:32	-
Citalec_RFID_ids_tmb1	/static/page/css/img/Citalec_RFID_ids_tmb1.jpg	10. avgust 2013. 09:32	10. avgust 2013. 09:32	-
Citalec_RFID_tmb1	/static/page/css/img/Citalec_RFID_tmb1.jpg	10. avgust 2013. 09:32	10. avgust 2013. 09:32	-
F2F-Ira-page	/static/page/css/img/F2F-Ira-page.jpg	10. avgust 2013. 09:25	10. avgust 2013. 09:25	-
Fonda-Ira-page8	/static/page/css/img/Fonda-Ira-page8.jpg	10. avgust 2013. 09:25	10. avgust 2013. 09:25	-

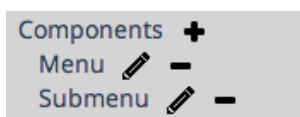
Slika 4.9: Pregled že prenesenih slik.



Slika 4.10: Prenos slik na strežnik.

## 4.2.5 Komponente

Sistem omogoča dodajanje svojih komponent. Napisane so v programskem jeziku Python in temeljijo na ogrodju Django. To so dodatne funkcionalnosti na spletni strani. Dodatne funkcionalnosti so na primer glavni meni, kontaktni obrazec, prijava na e-novice ali katera koli druga funkcionalnost spletne strani.



Slika 4.11: Vnosna maska za dodajanje nove komponente.

Komponente so sestavljene iz dveh datotek. Prva datoteka vsebuje kodo HTML, ki uporablja ukaze iz knjižnice Django (primer na sliki 4.12). Druga datoteka pa

vsebuje kodo Python (primer na sliki 4.13). Komponenta sestoji iz "frontend" kode in "backend" kode. "Frontend" koda predstavlja obliko podatkov, ki jih bo komponenta prikazala. V "backend" kodi pa naredimo funkcionalnost, ki jo bo komponenta dodala sistemu v upravljanje z vsebinami. V primeru imamo kodo, ki služi za prikaz glavnega menija na spletni strani.

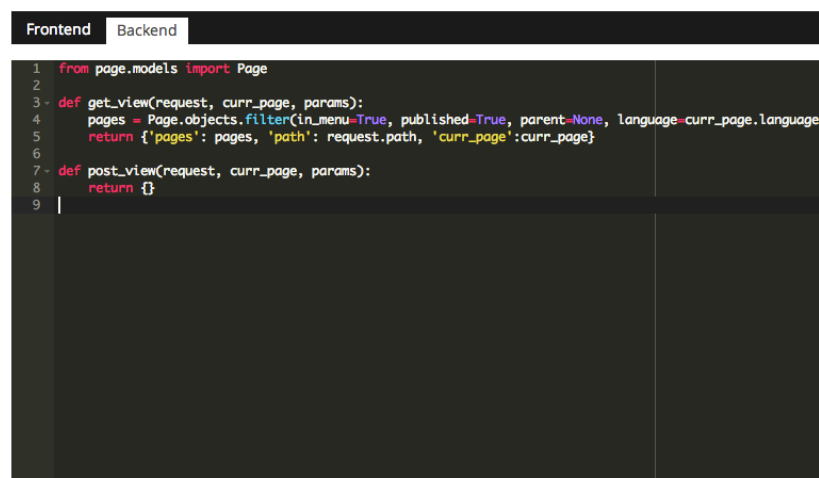
### Edit control

Name	<input type="text" value="Menu"/>
Value	<input type="text" value="menu"/>

Frontend Backend

```
1 <ul id="menu">
2   {% for page in pages %}
3     <li {% if curr_page == page %}class="current"{% endif %}><a href="{%page.language%}{%page.url%}">{
4   {% endfor %}
5 </ul>
```

Slika 4.12: Vnosna maska za dodajanje nove komponente s primerom "Frontend" kode.



```

1 from page.models import Page
2
3 def get_view(request, curr_page, params):
4     pages = Page.objects.filter(in_menu=True, published=True, parent=None, language=curr_page.language)
5     return {'pages': pages, 'path': request.path, 'curr_page': curr_page}
6
7 def post_view(request, curr_page, params):
8     return {}
9

```

Slika 4.13: Vnosna maska za dodajanje nove komponente s primerom "Backend" kode.

#### 4.2.6 Jezikovne različice spletne strani

Spletna stran je lahko prevedena v kolikor jezikov hočemo. Vsak jezik je potrebno ročno nastaviti. Na sliki 4.14 vidimo spisek jezikov. Jezikovno različico dodamo s klikom na gumb "Add language". Odpre se nam vnosna maska za dodajanje jezikovne različice (slika 4.15). V polje "Name" vnesemo ime jezika, ki ga želimo dodati. V polje "Code" vnesemo kodo države (npr. si, en, gb, hr...). Če želimo, da je jezikovna različica, ki jo dodajamo privzeta za spletno stran, obkljukamo polje "Is default". Nato shranimo izbrano jezikovno različico.

Languages

Add language

Name	Code	Default	Modified	Created	
English	en	<input type="checkbox"/>	21. julij 2013. 19:57	20. julij 2013. 18:20	
Slovenian	si	<input checked="" type="checkbox"/>	9. avgust 2013. 19:40	21. julij 2013. 19:59	

Slika 4.14: Spisek jezikovnih različic.



Add language

Name

Code

Is default

Slika 4.15: Vnosna maska za jezikovno različico s primerom.

### 4.2.7 Uporabniki

Uporabniki so osebe, ki skrbijo za urejanje spletne strani. Delimo jih na skrbnike in urednike. Ureja jih lahko samo skrbnik sistema. Na sliki 4.16 je spisek obstoječih uporabnikov, ki imajo dostop do sistema. V stolpcu "Username" so zapisana uporabniška imena, ki se jih uporablja pri prijavi v sistem. "Group" določa kakšen tip dostopa do sistema ima uporabnik. Če je uporabnik "Admin" pomeni, da je skrbnik. V primeru da je "Content Manager", je uporabnik urednik in ima omejen dostop. "Last login" je datum zadnje prijave v sistem. Ta informacija se uporablja v primeru, da želimo vedeti aktivnost uporabnikov sistema. Stolpec "Joined" je datum, kdaj je bil uporabnik dodan v sistem. Novega uporabnika dodamo s klikom na gumb "Add user".

#### Users

Username	Group	Last login	Joined	
user	Content Manager	25. avgust 2013. 18:21	17. avgust 2013. 11:38	 
admin	Admin	27. avgust 2013. 23:11	27. avgust 2013. 23:11	 

Slika 4.16: Spisek obstoječih uporabnikov sistema.

Uporabnika dodamo preko vnosne maske za dodajanje novega uporabnika (slika 4.17). Polje "Username" predstavlja uporabniško ime, ki ga uporabnik uporabi ob prijavi v sistem. Da bi lahko uporabnike sistema kontaktirali v primeru

kakšne napake, moramo podati tudi e-naslov uporabnika, ki ga zapišemo v polje "Email". Geslo vpišemo v polje "Password". Za zagotovitev, da smo pravilno vpisali geslo, ga moramo vpisati tudi v polje "Re-password". Geslo je priporočljivo imeti dolgo vsaj 6 znakov, da ga je težje ugotoviti, saj če nam kdo ukrade ali ugotovi uporabniške podatke, lahko spreminja celotno vsebino spletne strani.

### Add new user

Username	<input type="text" value="Username"/>
Email	<input type="text" value="Email"/>
Password	<input type="password" value="Password"/>
Re-password	<input type="password" value="Re-password"/>
Group	<input type="radio"/> Admin <input checked="" type="radio"/> Content manager
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

Slika 4.17: Vnosna maska za novega uporabnika.

## 4.3 Uredniški del sistema za upravljanje z vsebinami

Ta del sistema je namenjen uporabnikom, ki skrbijo za vsebine na spletni strani. V njem dodajamo nove podstrani ("Pages"), urejamo vsebine na straneh, urejamo objave ("Posts") in dodajamo galerije ("Gallery"). Slika 4.18 prikazuje meni, ki ga vidi urednik, ko je prijavljen v sistem za upravljanje z vsebinami.



Slika 4.18: Meni, ki ga vidi urednik v sistemu za upravljanje z vsebinami.

### 4.3.1 Strani oz. podstrani

Ker lahko spletna stran podpira več jezikovnih različic, so v sistemu za upravljanje z vsebinami podstrani ločene glede na jezikovno različico. Tako so lahko strani popolnoma drugačne glede na izbran jezik. Podstrani dodajamo in urejamo s pomočjo menija na levi strani okna (slika 4.19). Podstran dodamo tako, da z levim gumbom miške kliknemo na "+" zraven že obstoječe podstrani ali jezikovne različice. Odpre se nam vnosna maska za dodajanje podstrani (slika 4.20).



Slika 4.19: Meni jezikovnih različic in podstrani.

**Add page**

Name

Layout

Url

In menu

Is main page

Published

Slika 4.20: Vnosna maska za dodajanje strani oz. podstrani.

Pri dodajanju strani moramo biti pozorni, kaj vnašamo in izbiramo v vnosni

maski. Ime podstrani "Name" je uporabljeno tudi pri prikazovanju menija, zato moramo izbrati primerna imena. Nato izberemo, katero obliko bo imela podstran ("Layout"). Vsaka podstran mora imeti tudi svojo povezavo ("Url"), preko katere dostopamo do posamezne podstrani. Povezava mora biti edinstvena in se ne sme ponoviti v dveh različnih podstraneh. Izbirno polje "In menu" nam pove ali želimo, da je podstran vidna v meniju na spletni strani. Če izberemo možnost "Is main page", smo trenutno podstran nastavili za vstopno stran. Izbirno polje "Published" nam pove ali želimo strukturo objaviti na spletni strani ali ne.

Ko smo uspešno dodali podstran in jo izbrali za dodajanje vsebine, se nam prikažejo zavihki in tabela, v kateri lahko vidimo že obstoječe vsebine (slika 4.21). Zavihki predstavljajo območja na izbrani obliki spletne strani. Ko izberemo zeleno območje (zavihek) kamor želimo dodati vsebino, kliknemo na "Add content". Odpre se nam vnosna maska za dodajanje vsebine HTML (slika 4.22).

Člani

Desna vsebina				Meni				Noga				Vsebina			
Name	Order	Modified	Created												
Člani	▲▼	9. avgust 2013, 22:07	9. avgust 2013, 22:06												

Add content    Add existing content

Slika 4.21: Tabela z vsebinami na podstrani "Člani".

Pri dodajanju vsebine moramo vnesti naslov vsebine ("Title"). Če želimo, da se naslov vsebine vidi na spletni strani, moramo izbrati možnost "Show title". Izbirno polje "Published" nam pove, ali želimo celotno vsebino prikazati na spletni strani ali ne. Izbrati moramo tudi vrsto vsebine. Privzeta vrsta vsebine je navadna vsebina HTML.

Pri vrsti vsebine ("Layout") in izbiri "HTML" se nam pojavi urejevalnik besedil, v katerega vnesemo vsebino. Vsebino lahko s pomočjo urejevalnika poljubno oblikujemo (slika 4.22).

Če smo za vrsto vsebine ("Layout") izbrali "Post", moramo izbrati, katero kategorijo objav želimo prikazati. Izbrati moramo tudi način prikaza objav. Tre-

nutno sta na voljo dva različna prikaza objav. Prvi je seznam objav, ki izpiše samo naslove objav s povezavami na celotno objavo. Drugi je celotna vsebina, ki prikaže celotno objavo. Vnesti moramo tudi število objav, ki jih želimo prikazati iz trenutne kategorije. Objave bodo nato prikazane ena pod drugo, odvisno od vrstnega reda, ki ga določimo v sistemu za upravljanje z vsebinami.

Pri "Gallery", ki je tretja vrsta vsebine, moramo samo izbrati katero galerijo želimo prikazati. Na spletni strani se bo prikazala avtomatsko, njen prikaz pa je odvisen od oblike strani in shranjenih datotek slik.

Pri dodajanju komponente na podstran moramo izbrati željeno komponento. Zraven se nam pokaže polje, v katerega vnesemo parametre v primeru, da jih kontrola potrebuje. Vsebino shranimo s klikom na gumb "Save". Na sliki 4.22 vidimo primer dodane vsebine.

### Edit content

Title

Show title

Published

Content type

Source

**B** **I** **U** **S**  $x_2$   $x^2$   $I_x$   $\equiv$   $\equiv$   $\equiv$   $\equiv$   $\equiv$   $\equiv$

Raziskave vključujejo dve področji:

- RFID (ang. RadioFrequency Identification) v inteligentnih aplikacijah s uporabo metod mehkega računanja.
- CP&ES (ang. Computer Processing & Embedded Systems) pokriva računalniške arhitekture, 'software-hardware' razvoj, paralelno procesiranje, vgrajene sisteme in programabilno logiko.

body

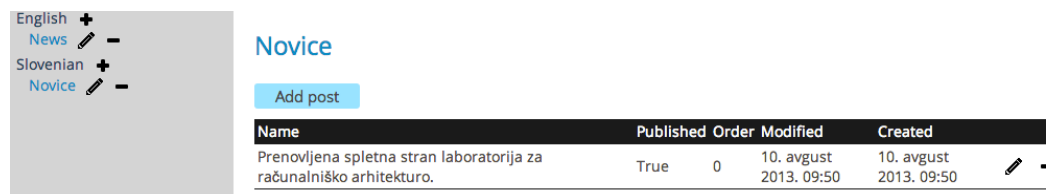
Save Cancel

Slika 4.22: Vnosna maska za dodajanje vsebine.

### 4.3.2 Objave (novice, članki ...)

Objavam je namenjen poseben del sistema za upravljanje z vsebinami. To pa zato, ker so drugačne kot navadna vsebina. Če bi hoteli imeti novice, ki bi jih naredili s pomočjo podstrani, bi morali vedno znova urejati isto statično vsebino. Pri tem pride do številnih problemov, saj se vedno kaj izgubi ali po nesreči pobriše. V našem primeru pa imamo vsako objavo ločeno od druge. Objave so kategorizirane in ločene ena od druge. Vsaki kategoriji je potrebno določiti ime in privzeto stran, katero uporabimo za prikaz celotne objave.

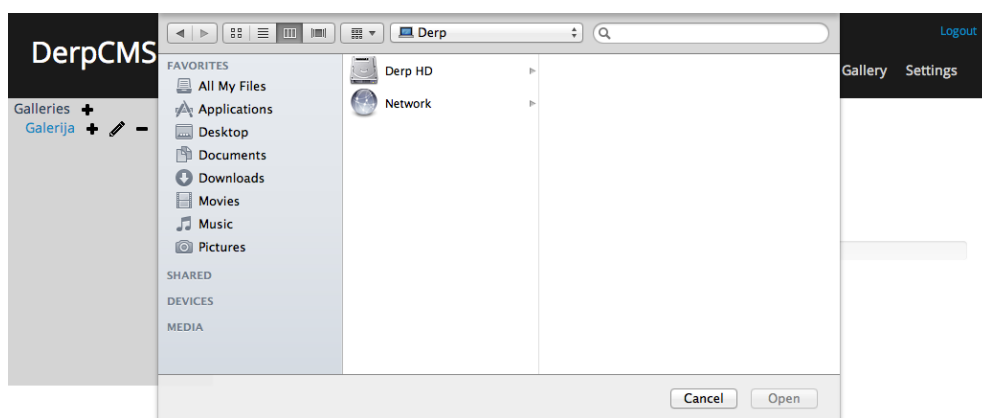
Na sliki 4.23 vidimo seznam objav, ki so dodane v kategorijo "Novice" pod slovensko jezikovno različico. Objavo dodamo s klikom na gumb "Add post". Prikaže se nam vnosna maska za vnos objave (slika 4.24). Objava sestoji iz naslova in vsebine. Izbrati moramo tudi možnost, ali je objava vidna na spletni strani ali ne. Na sliki 4.24 vidimo primer vnešene vsebine.



Name	Published	Order	Modified	Created	
Prenovljena spletna stran laboratorija za računalniško arhitekturo.	True	0	10. avgust 2013. 09:50	10. avgust 2013. 09:50	-

Slika 4.23: Seznam objav pod izbrano kategorijo.





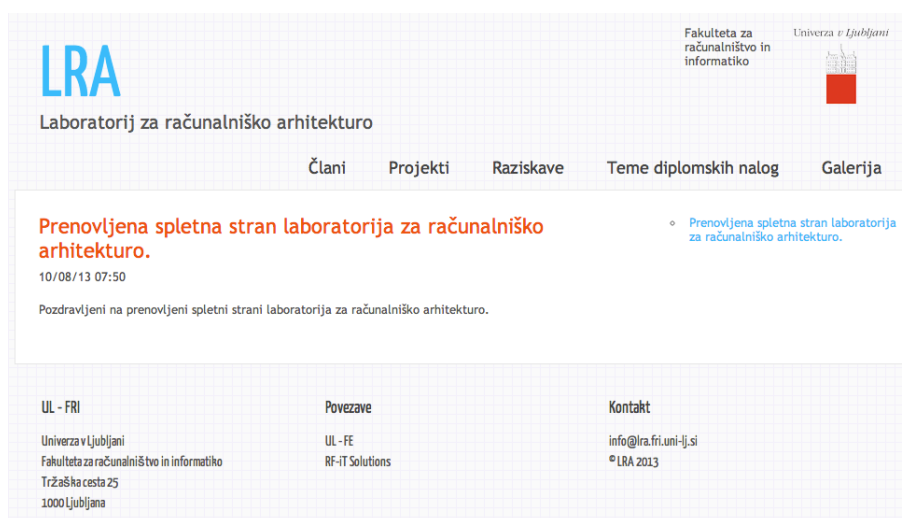
Slika 4.25: Upravitelj datotek pri dodajanju slik v galerijo.



Slika 4.26: Drevesna struktura galerij.

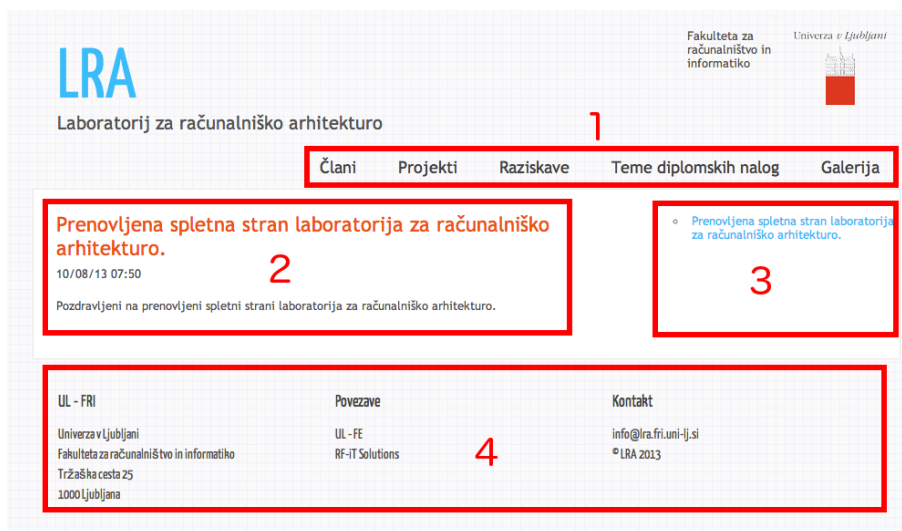
## 4.4 Predstavitev spletne strani LRA

Pred izdelavo dizajna za spletno stran laboratorija za računalniško arhitekturo smo preverili že obstoječo spletno stran, koliko ima vsebin in kako je vsebina strukturirana [21]. Nato smo na podlagi že obstoječe strani naredili nov dizajn, ki je prikazan na sliki 4.27.



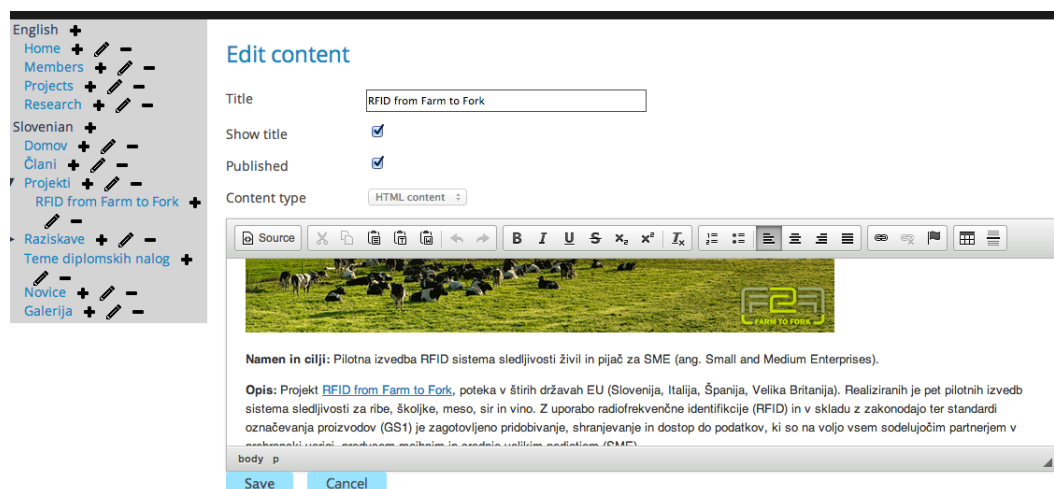
Slika 4.27: Vstopna stran in osnovni dizajn spletne strani.

Spletno stran smo razdelili na štiri območja, kot so prikazana na sliki 4.28. Območje številka 1 je namenjeno prikazu glavnega menija spletne strani. Poimenovali smo ga "menu", saj bo glavni meni na vseh straneh isti, ker moramo vedno imeti dostop do glavnih strani. Območje številka 2 je namenjeno vsebini spletne strani in smo ga poimenovali "content". V njem se bo prikazovala vsebina, galerije, objave itd. Območje številka 3 "sidebar", je namenjeno podmeniju na posamezni strani. V njem lahko prikažemo tudi navadno vsebino. Kontaktne podatke in ostale povezave smo dali v nogo "footer" spletne strani ki je v območju številka 4. Ko smo razdelili spletno stran na območja, smo napisali kodo HTML (A) in jo dodali v sistem kot nov opis spletne strani. Območjem smo v sistemu dodali slovenska imena za lažjo prepoznavnost le-teh.

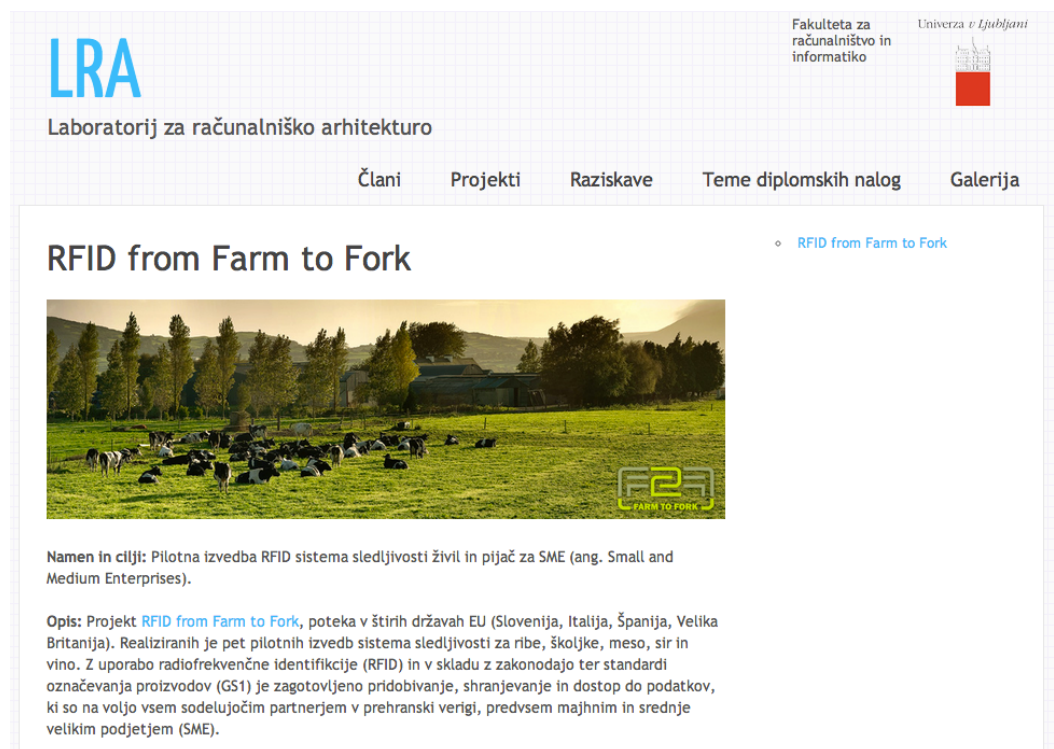


Slika 4.28: Razdelitev vstopne strani na segmente.

Ko smo uspešno dodali opis spletne strani, smo izdelali in v sistem dodali datoteke s kodo CSS in Javascript, ki smo jih potrebovali. Nato smo se lotili izdelave strukture spletne strani. Najprej smo dodali slovensko in angleško jezikovno različico, saj jo želimo imeti v dveh jezikih. Nato smo se lotili izdelave strukture strani. Najprej smo naredili vstopno spletno stran in nato dodali še vse ostale podstrani. Slika 4.29 prikazuje vnosno masko z vnešenimi podatki, ki jih prikazujemo na spletni strani. Pretvorjene podatke nato vidimo na spletni strani tako kot prikazuje slika 4.30



Slika 4.29: Vnosna maska za urejanje vsebine z vnešeno vsebino.



Slika 4.30: Primer prikaza vsebine na spletni strani.

Spletni strani smo dodali tudi novice, ki so vidne na vstopni strani. Obiskovalec spletne strani lahko takoj vidi, če ima laboratorij za računalniško arhitekturo kakšno obvestilo. Novice smo dodali s pomočjo objav v DerpCMS. Naredili smo novo kategorijo z imenom "Novice" in v njo dodali novo novico. Nato smo vstopni strani dodali novo vsebino, ki skrbi za prikaz novic. Tako smo povezali novice z območjem na vstopni strani, kjer so prikazane.

Stran vsebuje tudi galerijo, katere namen je prikazati delo laboratorija in njihove dosežke tudi v sliki. Naredili smo galerijo z imenom "Galerija", ki vsebuje vse slike o laboratoriju. Nato smo naredili novo stran v strukturi in ji dodali vsebino, ki skrbi za prikaz galerij. Slike se nam na strani prikažejo v obliki tabele (slika 4.31). S klikom na zeleno sliko se prikaže v večjem oknu.



Slika 4.31: Stran z galerijo.

# Poglavje 5

## Sklep in ugotovitve

Diplomska naloga je vključevala zasnovo in izdelavo sistema za upravljanje z vsebinami, ki omogoča izdelavo in urejanje celotne spletne strani. Sistem uvaja novosti v svet sistemov za upravljanje z vsebinami, saj nikjer še nismo zasledili sistema, ki bi omogočal dodajanje funkcionalnosti spletne strani kar preko brskalnika.

Največja prednost sistema je, da lahko dodajamo in izdelujemo nove funkcionalnosti sistema preko brskalnika. Prednost je tudi ločevanje strani glede na jezikovno različico, ker lahko za različne jezike naredimo popolnoma drugačne spletne strani.

Slabost sistema je uporabljena podatkovna baza SQLite, saj ima zelo šibko varovanje in je varnost podatkov lahko hitro ogrožena.

Ena izmed največjih težav je bila prikaz celotnih objav na spletni strani. Kot vemo so objave najprej prikazane samo kot spisek le teh, nato pa ko izberemo željeno objavo, se nam odpre v novem oknu. To smo rešili tako, da pri kategoriji objav izberemo na kateri strani želimo, da se nam prikaže celotna vsebina objave. Naslednja težava je bila izdelava komponent. Komponente morajo imeti v ozadju programsko kodo, ki se mora izvršiti na strežniku. V večini programskih jezikov, bi bila ta naloga zelo težavna, saj jih večina deluje tako, da najprej kodo prevede v računalniku berljivo kodo in jo šele nato izvaja. Pri programskem jeziku Python to deluje tako, da računalnik sproti interpretira programsko kodo in nam omogoča,

da lahko dodajamo funkcionalnosti kar preko brskalnika. Poleg tega smo morali razdeliti in poimenovati metode, ki se bodo klicale, ko bodo komponente na spletni strani. Kot vemo spletna stran deluje na dva načina. Prvi način je takrat ko želimo samo prejeti podatke iz strežnika, drugi pa je ko želimo na strežnik poslati podatke. Zato smo naredili dve metodi, ki se pokličeta v pravem trenutku.

Sistem bi lahko nadgradili še z izdelavo kontaktnih obrazcev, ki so zelo pogosti na spletnih straneh. Potrebno bi bilo še vložiti precej dela v razvoj komponent, saj so trenutno precej omejene kakšne funkcionalnosti lahko preko njih dodamo v sistem. Funkcionalnosti, ki bi jih lahko dodali še v sistem za opravljanje z vsebinami je zelo veliko. Odvisne so predvsem od tega, kaj naročnik oz. uporabnik želi na svoji spletni strani (npr.: meni, kalkulator, poljubni izračuni ...).

# Literatura

- [1] (2013) Django. Dostopno na: <https://www.djangoproject.com/>
- [2] (2013) The Django Book. Dostopno na:  
<http://www.djangobook.com/en/2.0/introduction.html>
- [3] (2013) SQLite. Dostopno na: <http://www.sqlite.org/about.html>
- [4] (2013) Kaj pomeni CMS. Dostopno na:  
<http://team.slojoomla.si/pzv/kaj-je-cms.html>
- [5] (2013) Razvojno okolje Sublime Text 2. Dostopno na:  
<http://www.sublimetext.com/2>
- [6] (2013) Sistem za upravljanje z vsebinami WordPress. Dostopno na:  
<http://wordpress.org>
- [7] (2013) Orodje za izdelavo relacijskega modela podatkovne baze. Dostopno na: <http://ondras.zarovi.cz/sql/demo/>
- [8] (2013) Urejevalnik besedil Ace. Dostopno na:  
<http://ajaxorg.github.io/ace/>
- [9] (2013) Joomla CMS. Dostopno na:  
<http://www.joomla.org/>
- [10] (2013) Chrome Developer Tools. Dostopno na:  
<https://developers.google.com/chrome-developer-tools/>
- [11] (2013) Knjižnica jQuery. Dostopno na: <http://jquery.com/>

- [12] (2013) Sistem za upravljanje z vsebinami Django CMS. Dostopno na:  
<https://www.django-cms.org/en/>
- [13] (2013) Javascript. Dostopno na: <http://www.w3schools.com/js/>
- [14] (2013) Google Chrome. Dostopno na:  
<https://www.google.com/intl/en/chrome/browser/>
- [15] (2013) CSS3. Dostopno na: <http://www.css3.info/>
- [16] (2013) HTML5. Dostopno na:  
[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)
- [17] (2013) Python. Dostopno na: <http://www.python.org/>
- [18] (2013) AJAX. Dostopno na: <http://www.w3schools.com/ajax/>
- [19] (2013) Knjižnica django-dajaxice. Dostopno na:  
<http://www.dajaxproject.com/>
- [20] (2013) Strežnik Apache HTTP. Dostopno na: <http://www.apache.org/>
- [21] (2013) Spletna stran laboratorija za računalniško arhitekturo.  
Dostopno na: <http://lra.fri.uni-lj.si/>

# Dodatek A

## Koda HTML opisa spletne strani

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Laboratorij za računalniško arhitekturo</title>
  <meta name="description" content="website description" />
  <meta name="keywords" content="website keywords, website keywords" />
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <link rel="stylesheet" type="text/css"
href="http://fonts.googleapis.com/css?family=Tangerine&v1" />
  <link rel="stylesheet" type="text/css"
href="http://fonts.googleapis.com/css?family=Yanone+Kaffeesatz" />
  <link rel="stylesheet" type="text/css"
href="/static/page/css/style.css">
  <link rel="stylesheet" type="text/css"
href="/static/page/css/colorbox.css">
  <script type="text/javascript"
src="/static/page/js/jquery-2.0.2.min.js"></script>
  <script type="text/javascript"
src="/static/page/js/jquery.colorbox-min.js"></script>
</head>
```

```
<body>
  <div id="main">
    <div id="header">
      <div id="logo">
        <div class='main-logo'>
          <h1><a href="/">LRA</a></h1>
          <div class='clear'></div>
          <div class='name'>Laboratorij za računalniško
            arhitekturo</div>
        </div>
        <div class="uni"><span class="logo"></span>
          <span class='school'>Fakulteta za računalništvo in
            informatiko</span></div>
      </div>
      <div class='clear'></div>
      <div id="menubar">
        [:menu:]
      </div>
    </div>
    <div id="site_content">
      <div id="sidebar_container">
        [:sidebar:]
      </div>
      <div id="content">
        [:content:]
      </div>
    </div>
    <div id="footer">
      [:footer:]
    </div>
  </div>
```

```
<script>  
    $('a.galgroup').colorbox({rel:'gal'});  
</script>  
</body>  
</html>
```