

UNIVERZA V LJUBLJANI  
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika – praktična matematika (VSŠ)

Ernest Šlosar

**RAZVOJ SPLETNIH STRANI Z ORODJEM TYPO3**

Diplomska naloga

Ljubljana, 2006

# Diplomska naloga

## ZAHVALA

Zahvalil bi se rad vsem, ki ste mi tako ali drugače pomagali pri snovanju diplomske naloge in mi brez vaše pomoči ne bi uspelo uresničiti zastavljenih ciljev. Za pomoč, vodenje, usmerjanje in posredovane ideje pri izdelavi diplomske naloge, se zahvaljujem mentorju mag. Matiji Lokarju. Hvala podjetju ISA.IT d.o.o. v okviru katerega sem bil vključen v projekt EUP, ki je bil osnova moje diplomske naloge. Prav posebna zahvala gre vsem mojim domačim, posebej staršem, ki so mi ob nastajanju diplomske naloge ves čas stali ob strani in mi po svojih močeh pomagali.

Diplomsko nalogo posvečam svojim staršem.

# KAZALO

<b>1. SISTEMI ZA UPRAVLJANJE Z VSEBINAMI</b>	<b>7</b>
1.1 KAJ SO SISTEMI ZA UPRAVLJANJE Z VSEBINAMI?	7
1.1.1 Potek procesa od ustvarjanja strani pa do njene objave na internetu	10
1.2 PREDNOSTI IN SLABOSTI SUV	11
1.3 IZBIRA PRAVEGA SUV	12
1.4 UPORABA SUV V SPLOŠNEM	13
<b>2. SISTEM ZA UPRAVLJANJE VSEBIN TYPO3</b>	<b>15</b>
2.1 KAJ JE IN ZAKAJ VSE JE DOBER TYPO3	15
2.2 DELITEV NA ČELNI SISTEM IN ZALEDNI SISTEM	15
2.2.1 Zaledni sistem	16
2.2.2 Čelni sistem	16
2.3 DOKUMENTACIJA ZA TYPO3 NA INTERNETU	16
2.4 RAZŠIRITVE SISTEMA TYPO3	16
2.4.1 Seznam razširitev sistema na internetu	17
2.4.2 Sestavine razširitve sistema	17
2.5 TYPOSCRIPT	18
2.5.1 Kaj je TypoScript?	18
2.5.2 Sintaksa TypoScript-a	18
2.5.3 Constants in Setup	21
2.5.4 Predloge	22
<b>3. SISTEM ZA UPRAVLJANJE VSEBIN TYPO3 (DELO S PROGRAMOM)</b>	<b>24</b>
3.1 NAMESTITEV TYPO3 IN ZAGON PROGRAMA	24
3.2 POVRŠINA IN NJENA OBMOČJA	25
3.2.1 Pregled delovne površine	25
3.2.2 Izbirni del	26
3.2.3 Navigacijski del	26
3.2.4 Podrobni prikaz	26
3.3 GLAVNE IZBIRE TYPA3	26
3.3.1 Izbira »Web«	26
3.3.1.1 Page	26
3.3.1.2 List	28
3.3.1.3 Info	29
3.3.1.4 Functions	29
3.3.1.5 Access	30
3.3.1.6 Template	31
3.3.2 Izbira »File«	32
3.3.2.1 Filelist	33
3.3.2.2 Images	33
3.3.4 Izbira »User«	34
3.3.4.1 Setup	34
3.3.5 Izbira »Tools«	35
3.3.5.1 Extension Manager	36
3.3.5.2 Extension Kickstarter	38
3.3.5.3 phpMyAdmin	38
<b>4. PROJEKT EUP</b>	<b>39</b>
4.1 OPIS PROJEKTA EUP	39
4.2 POSTOPEK IZDELAVE PROJEKTA	39
4.2.1 Analiza	39
4.2.1.1 Cilji in usmeritve	39
4.2.1.2 Funkcionalne zahteve oz. zahteve, ki jih je podal naročnik projekta	39
4.2.1.3 Identifikacija in opis uporabnikov	41
4.2.1.4 Identifikacija in opis primerov uporabe	41
4.2.1.5 Diagram primerov uporabe	41
4.2.2 Načrtovanje	42
4.2.2.1 Postavitev spletnega mesta	42
4.2.2.2 Tabele v podatkovni bazi	43
4.2.3 Uporabljeni orodja za izdelavo	43

4.3 IZDELAVA PROJEKTA EUP .....	43
4.4 OPIS IN IZVEDBA NEKATERIH FUNKCIONALNOSTI PORTALA .....	48
4.4.1 <i>Prijava v portal</i> .....	48
4.4.2 <i>Priprava koledarja dogodkov</i> .....	50
4.4.3 <i>Razširitev Podatkovna baza EUP</i> .....	56
4.4.4 <i>Razširitvi Address list in Address Directory</i> .....	68
<b>5. ZAKLJUČEK</b> .....	<b>73</b>
<b>LITERATURA</b> .....	<b>74</b>

in fiziko  
matematiko za  
Fakulteta

## Program dela

V diplomski nalogi prikažite uporabo sistema za upravljanje z vsebinami Typo3. Opišite tudi razvoj portala za potrebe informacijske podpore priprav na predsedovanje Slovenije Evropski uniji.

Mentor:  
mag. Matija Lokar

## Povzetek

V diplomski nalogi sem opisal, kaj so sistemi za upravljanje z vsebinami (s kratico SUV). Opisane so prednosti in slabosti SUV in kriteriji za izbiro ustreznega sistema. Posebej je opisan sistem Typo3. Prikazane in opisane so tiste najpomembnejše izbire tega sistema, ki so bile uporabljene pri izdelavi portala EUP. V zadnjem delu je opisan projekt izdelave portala EUP. V diplomski sem se osredotočil na opis izdelave posameznih funkcionalnosti portala. Tako sem opisal, kako smo izvedli samo prijavo v portal. Opisana je tudi razširitev sistema Typo3 s koledarjem dogodkov. Poleg koledarja dogodkov sem opisal še razširitev, ki je bila uporabljena za vodenje podatkov o kontaktnih osebah. Opisana je tudi najpomembnejša razširitev (Podatkovna baza EUP), ki je bila v celoti izdelana za potrebe tega projekta.

Math. Subj. Class. (2000): 68M10, 68N15, 68N19, 68P20, 68U35

Comp. Class. Sys. (1998): C.2.2, C.2.3, D.1.7, D.3.1, D.3.2, D.3.3, E.1, H.2.3, I.7.2

Ključne besede: Typo3, portal, spletni koledar, sistemi za upravljanje z vsebinami (SUV), vtičnik

Key words: Typo3, portal, web calendar, content management system (CMS), plug-in

# 1. Sistemi za upravljanje z vsebinami

## 1.1 Kaj so sistemi za upravljanje z vsebinami?

V današnjem času naraščajo težave pri upravljanju spletnih strani. Količina informacij, ki jih želi določena ustanova, podjetje, organizacija, .... nuditi preko svetovnega spleta, se povečuje. Tako nenehno narašča število spletnih strani, ki jih je potrebno postavljati in vzdrževati. Z naraščanjem števila spletnih strani se povečuje čas, ki ga moramo žrtvovati za oblikovanje in urejanje spletnih strani.

Problem, ki se pojavlja, je naslednji:

Imamo veliko spletnih strani, ki jih želimo postaviti na svetovni splet. Pri tem želimo, da se strani gradijo dinamično, glede na podatke, ki so največkrat v določeni bazi podatkov. Prav tako hočemo, da imajo strani poenoten izgled in da so čim enostavnejše za urejanje.

Če želimo zgornji problem rešiti, potrebujemo

- spletni strežnik,
- bazo in
- ustrezeni programski jezik.

Spletni strežnik bo poskrbel, da bodo uporabniki dobili ustrezno oblikovane strani. Da bo to mogoče, se bodo na strežniku izvajali določeni programi, ki bodo generirali kodo v jeziku HTML. Pri tem bodo potrebovali podatke, ki jih bodo črpali iz baze podatkov. Če bi želeli vse potrebne programe v ta namen napisati sami in nato še vse ustrezno vzdrževati, bi bilo potrebno vložiti zelo veliko časa in truda. Zato so se razvili sistemi za upravljanje z vsebinami, ki jih označujemo tudi s kratico SUV. Angleško je ustrezna kratica CMS – Content Management Systems. Ti vsa prej naštetja opravila združujejo v enem paketu. S pomočjo SUV je urejanje, oblikovanje in vzdrževanje spletnih strani običajno lažje in cenejše. Ti sistemi so v večini primerov zelo enostavni za upravljanje.

Trg SUV je velik premik doživel z večjo razširjenostjo odprtokodnih rešitev. Tako so postale prosto dostopne vse poglavitne sestavine sistema, ki nam omogoča dinamično generiranje spletnih strani v povezavi z bazo podatkov: spletni strežnik, sistemi za delo z bazami podatkov in ustrezeni programski jeziki z bogatimi knjižnicami in moduli. Posebej pri spletnih strežnikih se je zelo uveljavil odprtokodni spletni strežnik Apache in celo vodi po razširjenosti vseh spletnih strežnikov.

Na področju priprave kompleksnejših spletnih strani pogosto srečamo dve kratici. To sta LAMP in WAMP. Prva označuje kombinacijo operacijskega sistema Linux, spletnega strežnika Apache, podatkovnega strežnika MySQL in enega od tako imenovanih P-skriptnih programskih jezikov (PHP, Perl ali Python). Druga kratica (WAMP) označuje podobno kombinacijo, le vse skupaj je zasnovano nad operacijskim sistemom Windows.

Značilnost sistemov za upravljanje z vsebinami je hierarhična razdelitev poteka ustvarjanja internetnih vsebin. Običajno razlikujemo med glavnimi uredniki (vsesplošno odgovorni za celotni sistem), souredniki (odgovorni za določena področja) in avtorji (pišejo članke oz. besedila, vendar brez odobritve urednikov nimajo pravice za objavo gradiva na spletnih mestih). V administrativnem smislu poleg oseb, ki ustvarjajo vsebino spletnih strani, obstajata še administrativni in grafični urednik. Prvi je odgovoren za nadzor pravic, ki jih imajo avtorji, uredniki in uporabniki ter določa kategorije in konfiguracijo skript SUV. Grafični urednik oblikuje predloge in grafične elemente, ki določajo izgled spletnih strani.

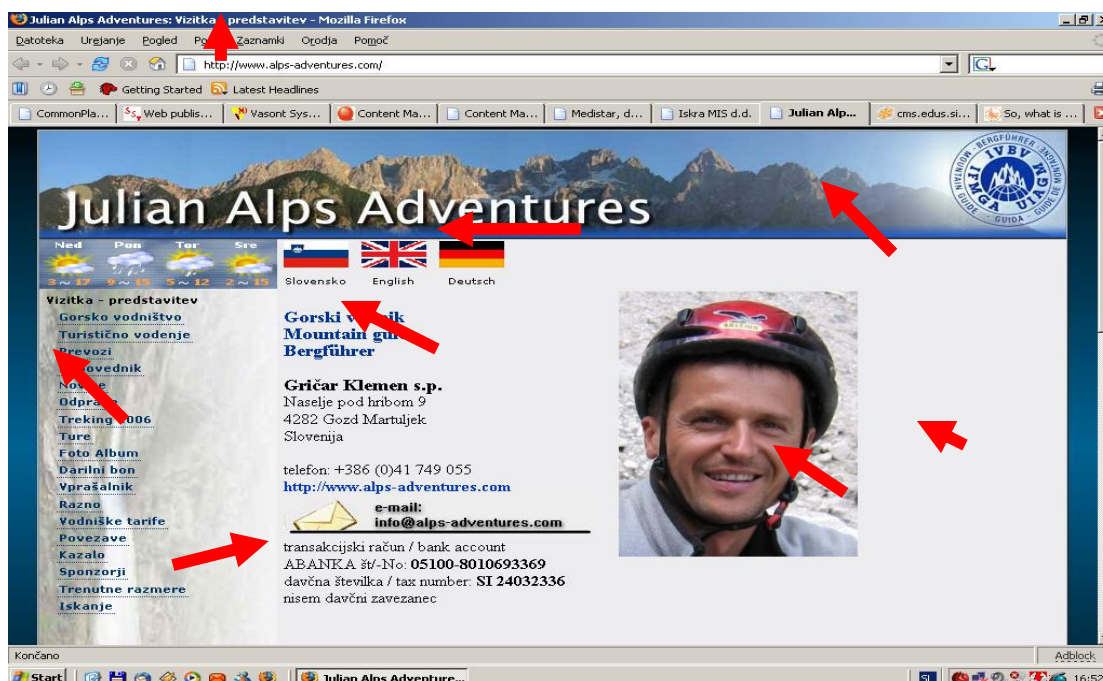
Sistemi za upravljanje z vsebinami nam nudijo možnosti na različnih področjih:

- integracija vsebin  
Gre za nudenje dostopa do informacij več uporabnikom hkrati. Ti lahko dostopajo do različnih ali istih dokumentov. Omogočen je nadzor nad različnimi verzijami dokumentov. Prav tako nam SUV lahko omogočajo avtomatsko pretvorbo iz večine važnejših datotečnih formatov v

format HTML. Na ta način preko spleta na poenoten način prikažemo vsebino dokumentov, ne glede na to, v katerem formatu so nastali.

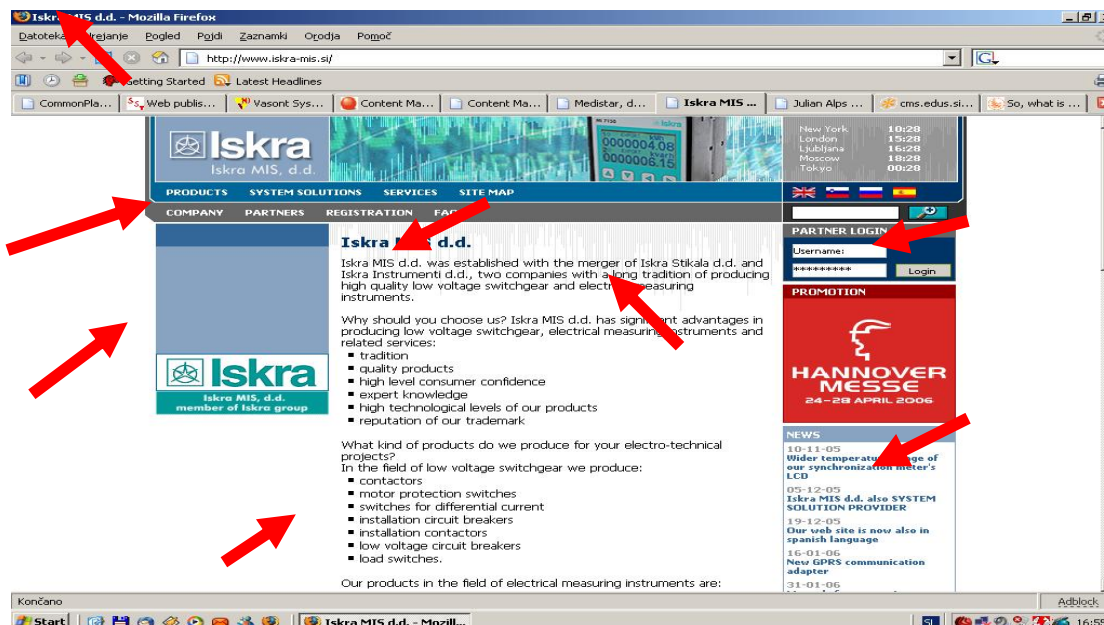
- tok dokumentov (angleško workflow)  
Sistemi za upravljanje vsebin nam omogočajo definiranje toka dokumentov pred dokončno objavo. Tako povemo, kdo je lahko avtor določenega dokumenta, kdo ga mora odobriti pred objavo, kdo ga lahko briše in podobno.
- podatkovni repozitorij  
Obstoječe podatke iz podatkovnih baz lahko integriramo v sistem. Prav tako imamo možnost shranjevanja podatkov v formatu XML.
- uporabniški vmesnik  
Omogoča lahko upravljanje z uporabniki in skupinami preko interneta. Imamo vizualni urejevalnik vsebin, s katerim natančno predvidimo izgled vsebine.
- objava vsebin

Na spodnjih dveh slikah sta prikazani dve spletni strani, ki sta (ali bi lahko bili) nadzorovani s sistemom za upravljanje vsebin. Deli, ki jih lahko kontroliramo in urejamo s SUV, so označeni s rdečimi puščicami.



Slika 1: Prikaz kontrole internetne strani s Sistemi UV





Slika 2: Prikaz kontrole internetne strani s Sistemi UV

### Nadzor zgornjih dveh spletnih strani s SUV

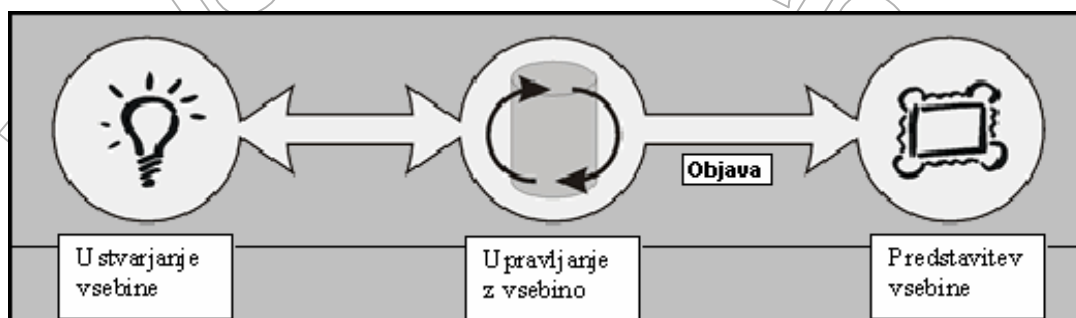
Sistemi nam omogočajo, da pri ustvarjanju strani najprej določimo ozadje in strukturo strani. Takrat povemo, kako in koliko stolpcev vsebine bomo imeli na strani. Pri strani na sliki 1 imamo dva stolpca. Levi stolpec je namenjen seznamu strani, do katerih lahko dostopamo, desni stolpec pa je namenjen objavi vsebine, kot so slike, sezname, ... Podatki, ki so prikazani na strani (denimo e-poštni naslov in podobno) niso »zapečeni« v samo stran, ampak so shranjeni v bazi. Tako ni težav, če se naslov spremeni, saj se vedno črpa iz enega mesta. SUV nam lahko omogoča, da se bo slika gorskega vodnika menjala in tako pogostejši obiskovalci te strani ne bodo vedno gledali iste fotografije. Tudi sliko SUV lahko črpa iz baze podatkov. Pri prikazani strani na sliki 1 imamo tudi nakazano možnost prikaza strani v slovenskem, angleškem in nemškem jeziku. Sistemi SUV nam omogočajo, da le pripravimo prevedene vsebine, sam sistem pa potem določi, katere elemente mora sestaviti v stran v določenem jeziku.

Pri strani, ki jo prikazuje Slika 2, je bila uporabljena drugačna predloga strani. Imamo tri stolpce. Levi stolpec je namenjen prikazu logotipa podjetja. Sredinski stolpec je, kot pri prejšnji strani, namenjen objavi vsebine strani, desni stolpec pa je namenjen prikazu reklam, aktualnih dogodkov in prijavi v portal. Novice SUV črpa iz baze in sam skrbi za to, da so vedno predstavljene tiste najnovejše. Naslov strani, barvo podlage menujskega sistema, izbiro oblike črk, ... prav tako lahko nadzira sistem SUV. Skratka praktično vsako malenkost, ki jo vidimo na zgornjih dveh slikah, lahko nadziramo s sistemi za upravljanje vsebin. Posebej je pomembno, da so vse strani določenega mesta strukturirane na enak način, z isto grafično podobo, zato bodo uporabniki te strani lažje uporabljali. Prav tako z vidika vzdrževanja ne bo težav, če se bomo odločili na primer spremeniti sliko gorske skupine na straneh, ki so predstavljene na Sliki 1. Le v bazi bomo zamenjali sliko in vse spletne strani, ki to predlogo uporabljajo, bodo prikazane z novo sliko.

SUV pokrivajo celoten razvoj spletnih strani: od ustvarjanja, oblikovanja, objavljanja, pa vse do arhiviranja strani. V sistemih imamo možnost nastavitve pravic dostopa do spletnih strani. Tako lahko za določene obiskovalce oziroma uporabnike nastavimo, katere strani lahko vidijo in katere ne. Prav tako imamo možnost, da določimo različnim avtorjem in urednikom različne pravice do urejanja posameznih delov sistema in strani.

### 1.1.1 Potek procesa od ustvarjanja strani pa do njene objave na internetu

Od ugotovitve, da je potrebno določeno informacijo v obliki spletne strani objaviti, do njene dejanske objave, gremo pri uporabi SUV preko treh faz.



Slika 3: Potek procesa ustvarjanja strani

#### Ustvarjanje vsebine:

Tu oblikujemo vsebino na način, ki je zelo podoben urejanju v običajnem urejevalniku besedil (kot je npr. program Microsoft Word). Čeprav se strani oblikujejo v skladu z jezikom HTML, nam ni potrebno skrbeti npr. za pravilno gnezdenje značk jezika HTML, saj za to skrbi sam SUV. V tej fazi lahko tudi oblikujemo strukturo strani. Tu določimo predvsem to, kako naj bodo strani med seboj povezane.

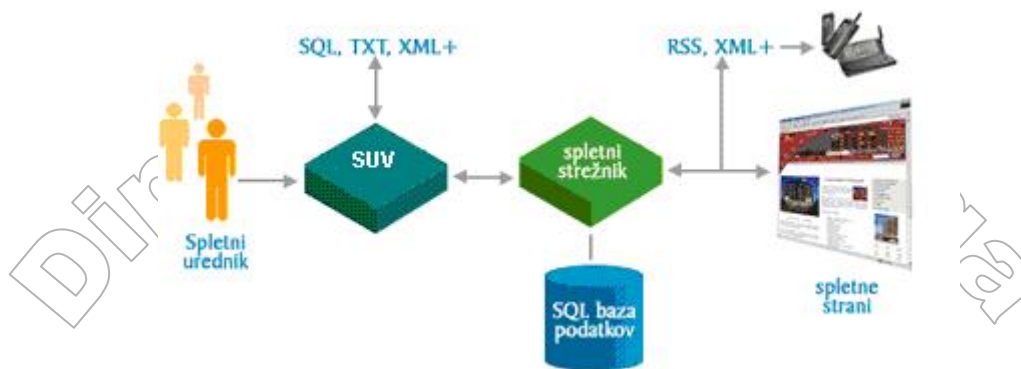
#### Upravljanje z vsebino:

Ko je stran narejena, se shrani v repozitorij SUV. Tukaj so shranjene vse vsebine, ki bodo prikazovane na strani. Repozitorij hrani vse verzije, ki smo jih naredili, od nastanka strani do zadnje spremembe. Hrani vse spremembe, ki so bile narejene in podatke o tem, kdo jih je naredil. V tej fazi se dejansko zgodi pretok podatkov od avtorja do odgovornega urednika. Ko avtor napiše besedilo (vsebino), ki ga želi dodati na spletno stran in to besedilo shrani, se besedilo doda na stran, ki je predvidena za shranjevanje še ne objavljenih vsebin. Ta stran je za ostale avtorje skrita. Prav tako je skrita pred obiskovalci strani. Ta stran je vidna samo odgovornemu uredniku. Šele, ko se odgovorni urednik odloči, da se vsebina lahko prikaže na strani, predvideni za objavo te vsebine, se ta dejansko prikaže na strani.

#### Predstavitev vsebine:

Ko uporabnik zahteva prikaz določenih spletnih strani, sistem za upravljanje vsebin poskrbi, da se bodo na spletnem strežniku generirali ustrezni opisi strani v HTML in posredovali uporabnikovem brskalniku. Pri tem SUV upošteva, katere strani imajo dovoljenje za objavo in na podlagi kategorije uporabnika določi, kaj bo uporabnik videl. Za ustrezen izgled strani poskrbi z izbiro primernih vzorcev in podlag ter z izbiro ustreznih datotek s stili (CSS). S pomočjo podatkov v bazi podatkov in kategorije uporabnika se ustrezno zgradi vsebina, vključno z vsemi povezavami, sestavnimi deli strani, izbirniki in podobno.

Spodnja slika prikazuje tok podatkov v celotnem postopku od priprave določene informacije do prikaza te informacije uporabnikom.



Slika 4: Pregled toka prenosa podatkov

Na Sliki 4 se pojavijo določene kratice. Opis podrobnosti o teh pojmihi bi presegal okvirje te diplomske naloge, zato se zadovoljimo s kratko razlago:

- SQL je jezik, ki je namenjen delu s podatkovnimi zbirkami
- TXT je datotečna končnica za besedilne datoteke. To so datoteke, shranjene brez oblikovnih podatkov.
- XML je opisni jezik, podoben jeziku HTML in nam omogoča standardiziran način opisovanja strukture podatkov.
- RSS je tehnologija na podlagi XML, s katero smo lahko samodejno obveščeni o temi, ki nas zanima.

## 1.2 Prednosti in slabosti SUV

Glavna prednost sistemov za upravljanje z vsebinami je v tem, da omogočajo enostavno urejanje in oblikovanje.

Tako lahko preko spletnega brskalnika kadarkoli in od koderkoli urejamo vsebino spletnega mesta. Ker nam SUV omogočajo identifikacijo uporabnikov, lahko določimo, da ima dostop do uredniškega dela s SUV več oseb znotraj podjetja oziroma ustanove. Tako lahko opravilo vzdrževanja spletnih strani porazdelimo med več osebami. Na ta način lažje dosežemo ažurnost spletnih strani.

Sistemi UV pa ne omogočajo le urejanja vsebine, temveč tudi celovito upravljanje spletnega mesta, vključno z njegovo strukturo, oblikovno podobo, pravicami uporabnikov in z vsemi moduli, ki so vključeni v spletno mesto. Tako lahko, če želimo spremeniti strukturo strani, na enem in edinem mestu, kjer imamo definirano, kakšna naj bo struktura strani, zamenjamo trenutno strukturo z novo. Pri tej spremembi ne uničimo nobene vsebine, ki je bila prej prikazana na straneh. Zdaj je le prikazana malo drugače, glede na to pač kako je zasnovana nova struktura strani. Pravice uporabnikov lahko spreminjamo od kjerkoli in kadarkoli. Pogoja za to sta le dva: dostop do interneta in uporabniško ime in geslo za prijavo v sistem SUV. Na podoben način lahko tudi razširjamo sistem z novimi dodatki. Tako lahko od koderkoli v sistem dodamo npr. koledar dogodkov, knjigo gostov in še ostale dodatke, ki bi jih želeli imeti na strani.

Ker pogosto avtorji vsebin (prispevkov, besedil, slik, ...) niso najbolj veščji sintakse jezika HTML, je pomembna prednost sistemov UV tudi ta, da se avtorjem pri oblikovanju in urejanju vsebine ni potrebno ukvarjati s sintakso HTML, saj to opravi sam SUV. Za tista spletna mesta, kjer je pomembno, da je informacija na voljo v več jezikih, je pomembno, da je s pomočjo uporabe SUV možno lažje vzpostaviti in vzdrževati večjezično podporo.

Kljub vsem prednostim, ki jih imajo sistemi UV, imajo tudi svoje slabosti. Ena izmed teh je, da v splošnem SUV niso primerni za vsak tip spletne strani. Nekateri sistemi pri oblikovanju spletnih strani puščajo več, drugi manj kreativne svobode. To je posledica uporabe že vnaprej pripravljenih predlog. Pogost očitke spletnim stranem, ki nastajajo s pomočjo sistemov za upravljanje vsebine, je ta, da so si celotni skupki spletnih strani preveč podobni in zato podjetje, oziroma organizacija, v spletu ni dovolj prepoznavna.

Pri odprtokodnih sistemih UV pogosto tudi pogrešamo preprost in jasen postopek namestitve. Veliko sistemov zahteva namestitve celega skupka programov najrazličnejših ponudnikov, kar lahko povzroča določene probleme glede združljivosti programskih knjižnic. Pri odprtokodnih SUV pogosto naletimo tudi na take, kjer je preveč poudarka na sami tehnologiji. Številni sistemi so namreč delo programerjev, ki želijo narediti sistem tehnološko čim boljši in pri tem pozabljajo na uporabnika. Čeprav so njihovi izdelki tehnološko izpiljeni, jim manjka uporabniške prijaznosti.

### 1.3 Izbira pravega SUV

Slabost sistemov za upravljanje z vsebinami je tudi zelo široka izbira le teh. Zato se je težko odločiti, kateri sistem bo za nas pravi. Če vemo, da obstaja več kot 250 SUV, je izbrati pravega in pri tem ne porabiti preveč časa, zelo težko. Pri izbiranju je največja napaka zaupati priporočilom drugih. Namreč naše zahteve in potrebe se pogosto zelo razlikujejo od zahtev in potreb tistih, ki nam določen SUV priporočajo. Kar je za nas prednost, je morda za druge slabost in obratno. Preden sploh začnemo izbirati ustrezen SUV, pa naj bo to komercialni ali odprtokodni, moramo najprej ugotoviti, kakšne so naše zahteve in želje. Ko dobro premislimo o le-teh, se lahko posvetimo izbiri sistema.

Obstajajo določena merila, ki nam pomagajo oziroma olajšajo izbiro sistema. Najprej se moramo odločiti, ali smo pripravljeni za sistem plačati ali ne. Ker so komercialni SUV pogosto dragi in zahtevni izdelki, pogledjmo, kaj nam ponujajo odprtokodni sistemi. Pri teh je pomembno dovoljenje za uporabo, oziroma licenčna določila. Tukaj imamo pri SUV najpogosteje opraviti s tremi možnostmi. Večina avtorjev odprtokodnih sistemov dovoljuje njihovo uporabo v skladu z dovoljenji GPL, LGPL in BSD. Dovoljenje GPL zahteva, da morebitne dopolnitve in razširitve programa javno objavimo v obliki izvorne kode. Če bomo torej vgrajevali v SUV določene razširitve, ki predstavljajo našo poslovno skrivnost, verjetno tovrstno dovoljenje ne bo najprimernejše. Dovoljenji LGPL in BSD pa nimata tovrstnih zahtev. Natančnejša razlaga razlik med dovoljenji pa spet presega okvir tega diplomskega dela.

Zelo je pomembna tudi podlaga – torej operacijski sistem, nad katerim sistem deluje. Če bi slučajno izbrali sistem, ki ima za podlago drug operacijski sistem, kot tisti, ki ga uporabljamo za svoje strežnike, nam to utegne povzročiti velike stroške, bodisi s samim učenjem in vzdrževanjem novega operacijskega sistema, bodisi z menjavo sistema.

Pomemben je tudi programski jezik, v katerem združujemo in nadziramo različne gradnike sistema. Čeprav je osnovna funkcionalnost vseh teh jezikov enaka, pa za določena opravila in okolja bolj ustreza jezik Python, za druga jezik Perl, v tretji okoljih bi se spet najbolj znašli v jezikih .NET. Dodatki, oziroma razširitve sistema, s strani skupnosti razvijalcev so spet ena od zahtev, ki jo mora imeti dober SUV. Če SUV ni preveč priljubljen, oziroma je le malo ljudi, ki ga razširja z novimi moduli, bomo morali vse narediti sami.

Med pomembnejšimi zahtevami so tudi varnostne zmožnosti sistema. Tako je pomembno, da v sistem ne more vstopiti vsak, ki nima pravice dostopa. V sistem naj dostopajo le osebe, ki imajo ustrezno uporabniško ime in geslo. Dobro je, če sistem uporablja protokol SSL. SSL je kratica za Secure Sockets Layer. Protokol SSL je izdelal Netscape in ga podpirajo vsi popularni brskalniki, kot so na primer Internet Explorer, Netscape, Firefox in Opera. Podatki, ki se pošiljajo preko protokola SSL, so zaščiteni z mehanizmom, ki preprečuje prisluškovanje in vdor. SSL nam omogoča, da uporabniki vedo, da so podatki, ki jih izmenjujejo s strežnikom (na primer številke kreditnih kartic) zavarovani in nedostopni tretjim osebam. Upabniki lahko takoj opazijo, kdaj se začne njihova seja SSL, saj se v

brskalniku pojavi ikona (ključavnica). Ob tem se v naslovni vrstici začetek spletnega naslova strani spremeni iz http v https, ki ponazarja, da je komuniciranje med uporabnikovim brskalnikom in strežnikom varno in zaščiteno.

Sistem mora biti narejen tako, da se lahko prilagaja glede na jezikovno področje oziroma večjezičnost. To je predysem uporabno, če želimo narediti strani, tako da jih je mogoče prikazati v več jezikih. Ta zmožnost je zelo uporabna, če imamo na straneh vsebine, ki bi utegnile zanimati tudi obiskovalce, ki razumejo le druge jezike. Take so na primer strani s predstavitvami izdelkov, ki jih ponujamo, izobraževalne vsebine, različne informacije o Sloveniji ali kaj podobnega.

Zelo zaželeno je, da sistem podpira enostavno urejanje. Pri urejanju se moramo počutiti domače. Vmesnik za oblikovanje mora biti čim enostavneje narejen. Pri sistemu je pomembno, kako hitro se ga lahko naučimo uporabljati. Če sistem ni enostaven za uporabo in vsebuje veliko funkcij za delo z njim, nam lahko povzroči veliko stroškov z izobraževanjem zaposlenih za delo s sistemom. Zato je pri izbiri zelo pomembna enostavnost urejanja in oblikovanja.

Zadnja, a ne najmanj pomembna lastnost, je možnost prenosa vsebine na drug sistem. Zaželeno je čim lažje prenašanje vsebine iz sistema na sistem. Tako bo manj težav, če bomo čez čas pač ugotovili, da smo se pri izbiri sistema odločili za napačnega in bomo želeli izvesti zamenjavo.

Kot vidimo, je kriterijev, ki so nam v pomoč pri izbiri sistema, veliko. Odveč je pripomniti, da so to le glavni kriteriji in ne vsi.

## 1.4 Uporaba SUV v splošnem

Oglejmo si poenostavljen zgled uporabe SUV. Denimo, da želimo sestaviti spletno mesto, ki ga sestavlja 6 strani. Na prvi strani bo opis in slika vasi Dolenje. Na drugi strani bodo podatki prijateljev in prijateljic s slikami. Na tretji strani bo opis Fakultete za matematiko in fiziko. Spodaj na tretji strani bodo še razna obvestila v povezavi z kolegi in kolegicami iz FMF (izleti, srečanja, ...). Na četrti strani bo knjiga gostov, v katero se bo lahko vpisal vsak obiskovalec spletnih strani. Peta stran bo namenjena forumu za prijatelje. Šesta stran bo namenjena uporabnim povezavam na spletu (trgovine z računalniško opremo, trgovine z športno opremo, ...).

Predlogo za strani želimo imeti sestavljeno iz štirih delov. To so levi rob in osrednji del, zgoraj in spodaj pa imamo še pasico. Seznam strani bo v levem robu. Želimo, da imajo vse strani enako ozadje in enako zgornjo in spodnjo pasico. V zgornji pasici je naslov "Ustvarjam v Typo3" podložen s sliko, ki je na datoteki zgornja\_pasica.jpg. V spodnji pasici pa naj bo besedilo "Ernest Šlosar" poravnano desno, krepko in poševno.

Pri ustvarjanju zgoraj navedenih strani uporabimo enega od sistemov UV. V sistem se prijavimo z uporabniškim imenom in geslom, ki nam pripada. Pokaže se delovna površina, na kateri delamo z vsebinami. Najprej izberemo ustrezno predlogo, ki bo ustrezala našim željam. Pogledamo, če izmed že v sistemu vgrajenih predlog obstaja katera, ki nam ustreza. Če take predloge ne najdemo, jo poizkusimo poiskati še na internetu. Če jo še tam ni, jo pač naredimo sami. Predlogo naredimo z orodji za predloge. V zgornjo pasico damo besedilo "Ustvarjam s Typo3" in jo podložimo s sliko zgornja\_pasica.jpg. V spodnjo pasico napišemo "Ernest Šlosar" poravnano desno, krepko in poševno. Ko imamo enkrat predlogo definirano, naredimo šest strani, ki smo jih opisali zgoraj. Strani naredimo s pomočjo funkcij za ustvarjanje spletnih strani. Za vse strani uporabimo prej pripravljeno predlogo. Vsem stranem nato dodamo vsebino.

Začnemo dodajati vsebino na prvo stran. V sistemu izberemo tip vsebine, ki nam omogoča dodajanje besedila in slike na stran. Vnesemo besedilo o vasi Dolenje in priprimo sliko. Ker pa uporabljamo sistem UV, lahko uporabimo možnost, ki nam omogoča dinamično menjavanje slik na strani. Tako lahko namesto, da bi veskozi prikazovali eno in isto sliko Dolenj, to lahko zamenjujemo s pomočjo SUV. Na strežnik dodamo več slik vasi Dolenje, ki se bodo zamenjale v določenem obdobju. S

pomočjo ustreznih skript v programskem jeziku poskrbimo, da se bodo ustrezno izmenjevale. Verjetno pa pisanje skript sploh ne bo potrebno, saj ima večina SUV že na voljo ustrezne gradnike, kjer le v njihove lastnosti vpišemo seznam slik in intervale, kako naj se izmenično prikazujejo.

Gremo na drugo stran. Na drugi strani želimo imeti seznam prijateljev in prijateljic. Ker uporabljamo sistem UV, si najprej ogledamo ali obstaja na internetu razširitev za sistem v obliki imenika oseb. Poiščemo ustrežno razširitev in jo uvedemo v sistem. Sama razširitev omogoča uporabnikom enostavno prilagajanje pogleda na ta seznam, iskanje po seznamu, urejanje in morda še kaj. Mi moramo le poskrbeti, da razširitvi povemo, kje v bazi podatkov se nahaja seznam prijateljev, ki ga želimo prikazati. Za vse ostalo bo poskrbljeno avtomatično.

Na tretji strani naredimo isti postopek kot na prvi. Na stran dodamo vsebino s sliko. Ponovno damo na strežnik več slik FMF, ki se bodo zamenjale po določenem obdobju. Rekli smo, da bomo na tej strani imeli tudi obvestila o različnih dogodkih. Pogosto vidimo spletne strani, ki prikazujejo obvestila, ki so že zastarela, druga spet imajo vsa obvestila zbrana v dolgem in nepreglednem seznamu, tretje strani nam spet ne omogočajo vpogleda v minula obvestila, ... Mi bi obvestila na tej strani radi naredili čim preglednejša, hkrati pa ne bi radi zgubljali dela s programiranjem vseh funkcionalnosti, ki bi odpravljali prej naštetih pomanjkljivosti. Zato na internetu poiščemo razširitev SUV v obliki koledarja dogodkov, ki nam bo služil za obvestila o raznih dogodkih. Koledar dogodkov uvozimo v sistem UV in ga vstavimo na tretjo stran. Koledar dogodkov deluje tako, da ob kliku na določen dan odpre dodaten prikaz, ki pove, kakšen dogodek se dogaja na ta dan. S tem je prikaz dogodkov pregleden, omogočen je arhiv, ... Edina naloga skrbnika te strani bo le, da bo dogodke dodajal v ustrezno bazo, za vse ostalo pa bo poskrbel SUV sam.

Na četrti strani bo knjiga gostov. Da bi sami naredili knjigo gostov, bi potrebovali kar nekaj časa. A tudi tu bomo izkoristili dejstvo, da gre za funkcionalnost, ki je na spletnih straneh pogosto uporabljena. Zato bo potrebno le poiskati ustrežno razširitev in jo vključiti na stran.

Na peto stran dodamo razširitev v obliki foruma, ki jo spet najdemo na internetu.

Na šesto stran dodamo vsebino v obliki seznama. Vnesemo ustrezne internetne naslove za povezave in vsebino dodamo na stran. Za primerno oblikovanje bo poskrbel SUV.

Želimo tudi, da do druge in pete strani dostopajo samo registrirani uporabniki. Zato na prvo stran dodamo prijavní obrazec. V sistemu s pomočjo izpolnjevanja enostavnih obrazcev z lastnostmi strani nastavimo, da do druge in pete strani dostopajo, in njihovo vsebino vidijo, le registrirani uporabniki.

Ustavimo se tukaj za trenutek in premislimo, kaj vse smo zgoraj naredili. Naredili smo šest strani z različno vsebino. Če bi možnosti, ki smo jih dodali na strani kot npr: koledar dogodkov, knjiga gostov, forum, hoteli sprogramirati sami, bi to trajalo lahko veliko časa. Tako pa smo imeli koledar dogodkov v nekaj minutah že pripravljen za uporabo. Druga prednost, ki smo jo dosegli z uporabo sistema UV, je ta, da smo naredili naše strani dinamične in ne pasivne. Tako se pri vzdrževanju lahko osredotočimo na samo vsebino, za ustrezen prikaz in črpanje podatkov pa bo poskrbel SUV. Tako na primer za prikaz novega dogodka ne bo potrebno popravljanje spletnih strani, ampak le dodajanje tega dogodka v bazo.

Velika prednost, ki smo jo z uporabo sistema UV dosegli, je tudi enostavnost bodočega urejanja. Za prikaz nove pasice ne bodo potrebni popravki vsake od šestih strani, ampak bomo zadevo spremenili le na enem mestu, saj se vsaka stran dinamično zgenerira iz podatkov. Ko bomo želeli, da se na strani prikazuje nekakšna nova slika Dolenj oz. FMF ali pa katerega od prijateljev, bomo sliko le dodali na strežnik v ustrezno mapo s slikami, za vse ostalo bo poskrbel sam sistem UV. Podobno je pri tekstu na straneh. Če bomo želeli dodati nov tekst, bo to opravljeno zelo hitro s pomočjo sistema UV. Če pa bomo želeli preoblikovati oz. urediti že obstoječe besedilo na straneh, bomo to naredili z lahkoto ob pomoči SUV, ne da bi se poglobljali v podrobnosti jezika HTML.



## 2. Sistem za upravljanje vsebin Typo3

### 2.1 Kaj je in zakaj vse je dober Typo3

Typo3 je odprtokodni sistem za upravljanje s spletnimi vsebinami in velja za eno od najzmogljivejših odprtokodnih rešitev na področju urejanja spletnih vsebin. Omogoča večnivojsko urejanje, enostavno izdelovanje grafičnih menijev, dinamično oblikovanje slik, ipd. Zlasti pomembna je možnost večnivojskega urejanja. Večnivojsko urejanje pomeni, da avtor najprej pripravi spletno stran in jo umesti v sistem. Uporabniki strani še ne morejo videti. Preden je stran objavljena na internetu, jo pregledajo uredniki, ki odobrijo objavo strani.

Typo3 ima svoj deklaracijski jezik, imenovan TypoScript. TypoScript je sestavljen iz kombinacije jezikov PHP in HTML. Z njim lahko nastavimo vse parametre sistema, kot so prikazovanje predloge na strani, definicija barv v povezavi s stili CSS, nastavljanje jezika v zalednem sistemu, ... Za nadzor in črpanje podatkov Typo3 uporablja kombinacijo jezika PHP in podatkovne baze MySQL. Novejše različice omogočajo tudi uporabo drugih baz podatkov, kot so na primer PostgreSQL, Oracle, MicrosoftSQL in druge. Podlaga (operacijski sistem Windows ali Linux), na katero ga namestimo, mora biti ustrezno razširjena z raznimi knjižnicami, saj Typo3 zahteva, da so nameščeni številni dodatki za PHP, kot je na primer knjižnica za delo s slikami ImageMagic.

Preden se naučimo uporabljati sistem Typo3, porabimo kar nekaj časa. Če želimo sistem popolnoma obvladati, moramo spoznati mehanizem predlog in deklaracijski jezik TypoScript. Vsebino spletnih strani sestavljamo iz gradnikov, ki jih izberemo iz seznama. Typo3 ima tudi vizualni urejevalnik vsebine, kar nam omogoča enostaven pregled nad obliko. Na voljo je tudi odložišče, preko katerega lahko vsebino med stranmi poljubno prenašamo. V sistemu imamo tudi urejevalnik slogov CSS, kar nam spet olajša delo s slogi. Typo3 omogoča popolno ločitev vsebine in oblike. To pomeni, da se avtorji ukvarjajo le z vsebino spletnih strani, za izgled pa skrbi na primer grafični urednik. Typo3 je preveden v več kot 40 jezikov, kar je dokaz, da je med uporabniki SUV zelo priljubljen. Med drugim je preveden tudi v slovenščino. Žal pa je prevod narejen samo za verzijo 3.8.0 in v njem je kar nekaj pomanjkljivosti. Zato bomo rajši opisovali (in uporabljali) angleško različico.

Typo3 nam omogoča izdelavo najrazličnejših spletnih strani: od kompleksnih portalov, e-trgovin, do strani v sklopu interneta, extraneta in intraneta. Omogoča tudi večjezičnost, kontrolo verzij, časovno specificirane objave, indeksirani iskalnik, urejanje preko čelnega sistema ind. Poleg obsežnega nabora funkcionalnosti in možnosti ga odlikujejo tudi izjemna stabilnost in nezahtevnost glede strojne opreme.

Typo3 ima vse svoje strani in vsebine na straneh shranjene v bazi. Vse kar vidimo na spletnih straneh se generira s pomočjo zapisov, ki so shranjeni v bazi podatkov. S tem je poskrbljeno, da se ne more nič izgubiti oz. spremeniti na strani. Povedati je potrebno še, da ima vsaka stran, vsebina, uporabnik, uporabniška skupina, besedilo ind. svojo enolično identifikacijsko oznako (ID). ID je številka, ki je prirejena vsem elementom v Typo3 in se nikoli ne podvoji. Tako se v internetnem brskalniku pokaže v vnosni vrstici id vsake strani, na kateri se nahajamo. Na ta način tudi lažje najdemo napake, odpravimo pomanjkljivosti in podobno, ko opazujemo obnašanje spletnih strani s stališča uporabnika, saj preko ID strani vemo, kje so.

V nadaljevanju bomo uporabnikom, ki vidijo samo čelni sistem, rekli uporabniki ali bralci in uporabnikom, ki imajo dostop poleg čelnega sistema še v zaledni sistem, bomo rekli uredniki. Kaj pa čelni in zaledni sistem sta, si bomo ogledali v naslednjem razdelku.

### 2.2 Delitev na čelni sistem in zaledni sistem

Odprtokodni sistem Typo3 se načeloma deli na dva dela: **čelni sistem** (angleško frontend) in **zaledni sistem** (angleško: backend).

### 2.2.1 Zaledni sistem

Glavno vlogo v Typo3 ima zaledni sistem, saj kot avtorji vsebin in oblik večino časa preživimo v njem. V njem ustvarjamo nove strani, dodajamo vsebine na strani, ustvarjamo in dodajamo nove gradnike oziroma razširitve sistema ind.

Ta del služi tudi za administracijo sistema. V zalednem sistemu urejamo pravice in dostop do naših internetnih strani posameznim uporabnikom/urednikom ali uporabniškim/uredniškim skupinam. Omeniti velja, da lahko videz zalednega sistema prilagajamo glede na velikost ekrana. Tako lahko po potrebi izklopimo sočasen pogled na navigacijski del in podrobni del sistema. To nam pri manjšem ekranu močno poveča pregled nad površino, na kateri delamo.

### 2.2.2 Čelni sistem

Čelni sistem je prikaz vsebin spletnega mesta uporabnikom. S pomočjo čelnega sistema lahko uporabnik pregleduje vsebine, ki sestavljajo spletne strani. Uporabnik vidi v brskalniku strani, ki jih posreduje spletni strežnik. Strani, ki jih posreduje spletni strežnik so se generirale s pomočjo podatkov v bazi in s pomočjo raznih nastavitev, ki smo jih naredili v zalednem sistemu. Čelni sistem je uporabniški vmesnik, ki omogoča uporabnikom strani, da vidijo podatke, ki so bili vneseni in spremenjeni v zalednem sistemu.

V čelni sistem je vgrajena tudi možnost okrnjenega urejanja strani. Na voljo imamo namreč tako imenovano »Administracijsko ploščo« (angleško: Admin panel). Z njo lahko določamo, ali so določene strani na internetu vidne ali ne. Tu lahko spreminjamo vsebino na straneh. Preko "Administracijske plošče" lahko dobimo informacije o določeni spletni strani. Administracijska plošča se vklopi le, če se najprej prijavimo v zaledni sistem in nato iz zalednega še v čelni sistem. Prav nam pride torej takrat, ko kot uredniki želimo imeti pogled na spletno mesto, kot ga bodo videli bralci, hkrati pa obdržati možnost manjšega urejanja strani.

## 2.3 Dokumentacija za Typo3 na internetu

Na domači strani projekta Typo3 je na voljo tudi veliko dokumentacije. Tako lahko najdemo opise različnih začetnih korakov, dokumentacijo o tem, kako sistem Typo3 namestimo, dokumentacijo za razvijalce gradnikov in podobno. Na voljo je tudi veliko podatkov o posameznih gradnikih sistema. Tistim, ki bi želeli podrobneje spoznati Typo3 in sestavljati razširitve, bo prav prišla dokumentacija o celotnem razvojnem okolju (API). Učenje določenih postopkov je še lažje, ker na internetu najdemo veliko video posnetkov, ki nadzorno pokažejo, kako se v Typo3 naredi določeno nalogo. Dokumentacija je večinoma v formatu, kot ga podpira starejša verzija paketa OpenOffice.org (swx). Poleg dokumentacije v tem formatu, imamo na voljo vedno več tudi dokumentacije v formatu PDF in dokumentacije v spletni obliki.

## 2.4 Razširitve sistema Typo3

Razširitve sistema so dodatki oz. gradniki z določenimi metodami, ki jih vključimo v sistem z namenom, da postane sistem bolj uporaben in funkcionalen. Razširitve, ki jih lahko dodamo v sistem, so lahko v obliki koledarja dogodkov, seznama kontaktnih oseb, ozadja za sistem, predvajalnika datotek mp3, klepetalnice, prijavnega obrazca, itd. Tako dodamo tiste razširitve, ki nam pridejo prav pri izgradnji spletnih mest, ki jih na strežniku upravljamo s SUV Typo3.

Razširitve v obliki, kot jih Typo3 uporablja sedaj, so bile uvedene z različico 3.5. Tudi pred različico 3.5 je bilo sistem mogoče razširjati, a je bilo razširitve zelo zapleteno vgraditi v sistem. Problem, ki se je velikokrat pojavil, je bil ta, da je neka nova razširitev, ki smo jo dodali v sistem, onemogočala delovanje neki razširitvi, ki smo jo prej dodali v sistem. Od različice 3.5 pa se je sistem razširitev sistema zelo izpopolnil. Uvoz oz. dodajanje razširitev poteka preko izbire Ext Manager. Dodajanje razširitev je tako postalo enostavno, njihova integracija oz. namestitvev pa lažja.



Vsako razširitev lahko prilagajamo svojim željam in potrebam. Tako je na primer standardno obnašanje razširitve Koledar dogodkov tako, da se ob kliku na določen datum prikažejo dogodki za izbrani dan. Če želimo, da se nam ob kliku na določen datum prikažejo dogodki ne le za izbrani dan, ampak še za naslednji dan, lahko to popravimo. Ker je na voljo ustrezna koda koledarja, napisana s skriptnim jezikom PHP, jo ustrezno spremenimo.

### 2.4.1 Seznam razširitev sistema na internetu

Seznam razširitev za Typo3 je zelo velik. Ponuja nam pestro izbiro vseh mogočih razširitev. Tako najdemo različne razširitve za koledar, števec obiskov, ozadja za administracijski vmesnik, pregledovalnike slik. Najboljša možnost iskanja razširitev je, da gremo na stran <http://typo3.org>. Tam se premaknemo na kartico Extensions (razširitve) in vpišemo ključno besedo, s katero bomo zožili izbor za nas zanimivih gradnikov. Če na primer potrebujemo razširitev za sistem v obliki koledarja dogodkov, vpišemo v polje za iskanje »calendar«.

Vsaka razširitev ima svoj unikaten ključ (angleško: extension key). S pomočjo tega ključa razširitve tudi uvozimo v sistem Typo3. Pred uvozom oz. integracijo razširitve v sistem je namreč ta ključ potrebno vpisati v Ext Manager, s katerim upravljamo z razširitvami. Podrobneje bo Ext Manager opisan v nadaljevanju diplomske naloge.

Če ustrezne razširitve, ki bi sistemu dodala željeno funkcionalnost, ne najdemo (na primer ne najdemo razširitve, ki nam bi na podlagi izbire dveh krajev v Sloveniji izpisala zračno razdaljo med njima), jo lahko pripravimo (napišemo) sami. Razširitve najlažje ustvarjamo s pomočjo razširitve z imenom Extension Kickstarter, ki jo bomo razložili v nadaljevanju diplomske naloge.

### 2.4.2 Sestavine razširitve sistema

Razširitev sistema je sestavljena iz večjega števila datotek. V teh datotekah so definirane tabele in imena polj v tabelah, ki jih razširitev potrebuje za pravilno delovanje. Vsaka od razširitev ima tudi določeno število vtičnikov. Za vtičnik v splošnem lahko rečemo, da je program, ki deluje v povezavi z glavno aplikacijo (npr. s spletnim brskalnikom ali s programom za elektronsko pošto) in skrbi za določeno funkcijo glavne aplikacije. Glavna aplikacija poskrbi, da vtičnik dobi podatke, ki jih lahko uporabi. Vtičniki so v večini primerov odvisni od podatkov, ki jih pridobijo od glavne aplikacije in ne morejo delovati samostojno. Vtičniki v čelnem sistemu Typo3 so internetne aplikacije, ki delujejo na internetni strani samostojno. Za njihovo delovanje torej ni potrebno, da bi se izvajali določeni procesi v zalednem sistemu.

Omenili smo že, da Typo3 podpira tudi večjezičnost. Tako med datotekami najdemo tudi datoteki `locallang.php` in `locallang_db.php`, ki sta pripravljeni v ta namen. Tu imamo prevode izrazov, ki nastopajo v razširitvi, v obliki parov »originalno ime polja« : »prevod«. Denimo, da imamo razširitev, ki nam na strani omogoča prikaz telefonskega imenika. V razširitvi so vnosna polja z naslovi Name, Surname, Address, Telephone, ... . Zaradi datoteke, v kateri so napisani prevodi (Name : Ime, Surname : Priimek, ...), pa so na spletni strani in vnosni formi sedaj namesto angleških izrazov izrazi Ime, Priimek, Naslov, Telefon,...

Razširitve sistema Typo3 razvrstimo v devet kategorij. Katere so te kategorije, lahko vidimo v izbiri Ext Manager. Tu so razširitve, ki jih imamo nameščene v naš sistem, urejene po teh kategorijah. Tako lahko najdemo razširitve, ki se nanašajo na zaledni del sistema, razširitve za čelni del sistema, itd. Čeprav se razširitve glede na kategorijo in uporabnost med seboj razlikujejo, pa imajo zelo dobro skupno lastnost. Vse imajo podobno strukturo map. Zato se med novimi razširitvami lažje znajdemo, saj vemo, kje iskati posamezne datoteke. Potrebno je še omeniti, da določena razširitev lahko zahteva, da so pred njo nameščene še druge razširitve. Takrat je potrebno najprej namestiti manjkajoče razširitve in šele na koncu namestiti razširitev, ki jo potrebujemo.

## 2.5 TypoScript

### 2.5.1 Kaj je TypoScript?

TypoScript je deklaracijski jezik, s pomočjo katerega določimo določene lastnosti spletnega mesta. Te navedemo tako, da jih umestimo v predlogo, na osnovi katere zgradimo določene spletne strani. TypoScript torej uporabljamo, da z njim ustvarjamo predloge in da nastavimo določene parametre razširitvam, ki to zahtevajo. Na predlogo lahko dodamo tako stalne, nespremenljive dele (na primer sliko za ozadje, določeno nespreminjajoče se besedilo in podobno), kot tudi različne dinamične (torej spreminjajoče se) vsebine, kot so dinamični meniji, menjajoče se slike in podobno. Vse stalne, nespremenljive dele, kot tudi dinamične vsebine ind. pišemo v polje Setup, ki ga najdemo v izbiri Web → Template.

Omeniti je potrebno, da TypoScript ni niti programski jezik niti skriptni jezik. Ne moremo ga primerjati z jeziki kot so Java, Delphi, PHP ali JavaScript. Tako TypoScript ne pozna zank, vejitev, metod in podobno. Pozna pa določene funkcije. Ena izmed funkcij je npr. pogojni stavek if. Gre predvsem za to, da s pomočjo deklaracij nadziramo izgled določenih elementov spletnih strani in celih skupin spletnih strani. S sintakso TypoScripta na hierarhičen način definiramo določene lastnosti. Oglejmo si primer zapisa nekaj stavkov v tem jeziku.

```
page.10 = HTML
page.10.value = <table border="0" cellpadding="0" cellspacing="0" width="1">
page.10.value {
    addParams._offset = -1
    addParams.bgColor = {$page.bgColors.wholeTableBgColor}
    addParams.background = {$page.file.wholeTableBgImg}
}

stdWrap.if {
    equals.field = header_layout
    value = 100
    negate = 1
}
```

Koda, napisana zgoraj, predstavlja delček kode predloge za spletno stran.

### 2.5.2 Sintaksa TypoScript-a

Vse vrednosti v TypoScript-u nastavljam na sledeč način:

```
[objektna pot].[lastnost] [operator] [vrednost]
```

Spremenljivke v TypoScript se imenujejo objektne poti. Te objektne poti lahko potem še ločimo na objekte in lastnosti. Kot bomo videli v nadaljevanju, ima to lahko določene prednosti. Pri navajanju vrednosti večinoma uporabljamo le operator »=«.

Za imena objektov in lastnosti lahko uporabimo samo znake A-Z, a-z, 0-9, pomišljaj »\_« in podčrtaj »\_«. Pri uporabi imen moramo paziti, ker TypoScript razlikuje med velikimi in malimi črkami. Imena `mojeBesedilo`, `MojeBesedilo`, `MOJEBESEDILO` so torej različna imena.

Pri zapisu objektne poti ne smemo delati presledkov pred in/ali za piko. Pri proučevanju številnih strani, napisanih s pomočjo TypoScripta lahko opazimo, da kot imena lastnosti pogosto uporabljajo kar števila (10, 20, ...). Žal v literaturi nisem našel razloga za to. Verjetno pa bo razlog povezan s tem, da se te objektne poti in lastnosti hranijo kot tabele v jeziku PHP.

TypoScript pozna tudi konstante. Konstante imajo imena oblike `$ime`.

Vrednosti objektne poti ali konstante pri prirejanju ne navajamo z narekovaji ampak brez njih. Vrednost zapišemo po operatorju in samo v eni vrstici. Vse, kar je za znakom »=« do konca vrstice se upošteva kot vrednost. Le vodilni in končni presledki se ne upoštevajo. Če pa presledke naredimo vmes, so upoštevani. Poglejmo nekaj zgledov

```
10 = TEXT
10.value = Beseda
```

Objektna pot »10« vsebuje vrednost »TEXT« in objektna pot »10.value« vsebuje vrednost »Beseda«.

```
10 = FILE
10.value = fileadmin/images/logo.gif
```

Objektna pot »10« vsebuje vrednost »FILE«, objektna pot »10.value« vsebuje vrednost »fileadmin/images/logo.gif«.

```
10 = HTML
10.value = <b>HTML</b> besedilo
```

Objektna pot »10« vsebuje vrednost »HTML« in objektna pot »10.value« vsebuje vrednost »<b>HTML</b> besedilo«.

Oklepaje { } uporabimo za bolj pregledno naštevanje in zapis lastnosti nekega objekta in pri zapisu več lastnosti naenkrat.

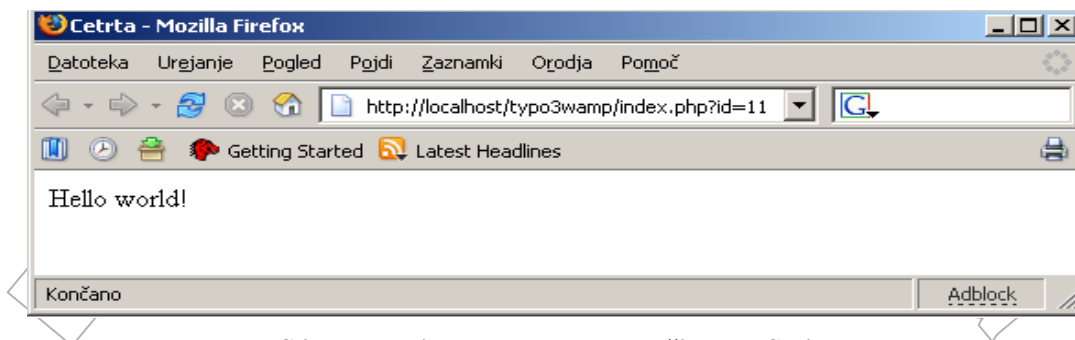
```
page = PAGE
page {
    typeNum = 0
    10 = TEXT
    10 {
        value = Hello world!
    }
}
```

Page je objektna pot z vrednostjo »PAGE«. page je tudi objekt, ki ima lastnosti typeNum in 10. TypeNum nam definira položaj objekta na strani. Če napišemo typeNum = 0 pomeni, da bo izpisan niz »Hello world!« v levem zgornjem kotu strani (Slika 5). 10 je spet objekt/lastnost, ki ima lastnost value. Vrednost, ki jo priredimo objektu/lastnosti 10 pa je niz »Hello world!«.

Ekvivalentni zapis zgornje kode brez oklepajev { } bi bil naslednji:

```
page = PAGE
page.typeNum = 0
page.10 = TEXT
page.10.value = Hello world!
```

Če bi stran, ki je "podprta" s temi ukazi (torej je nastala tako, da smo za predlogo uporabili predlogo, ki vsebuje te ukaze) prikazali, bi izgledala:



Slika 5: Napis Hello world s pomočjo TypoScript

Če je vrednost, ki jo želimo shraniti, predolga za eno vrstico ali če želimo vnesti večvrstično vrednost, si pomagamo z oklepaji ( ).

Zgled:

```
page {
    typeNum = 0
    10 = HTML
    10.value (
        <table width = »100%« border = »0«>
            <tr>
                <td><span><b>Lastnosti, ki so napisane v več kot eni
                    vrstici!!!!</b></span>
                </td>
            </tr>
        </table>
    )
}
```

V zgornjem zapisu smo poleg uporabe večvrstične vrednosti za "10.value" objektne poti razbili s pomočjo oklepajev { }. V zgornjem zapisu bi bili »typeNum« in »10« lastnosti objekta »page«. »value« pa lastnost objekta/lastnosti 10 (oz. page.10).

Komentarje lahko pišemo na več načinov. Prvi so enovrstični komentarji, ki se začno z znakom # ali / in se končajo s koncem vrstice. Za komentar, ki se razteza čez več vrstic, pa uporabimo oznaki /\* in \*/.

```
# komentar vrstice
/ komentar, ki je ravno tako komentar ene vrstice
/* prva vrstica komentarja
   druga vrstica komentarja
   tretja vrstica komentarja*/
```

Na koncu lahko pokažemo kako bi zapis v TypoScriptu izgledal v zapisu tabele v PHP. Uporabimo lahko kar zgled od prej in sicer:

```
page = PAGE
page {
    typeNum = 0
    10 = TEXT
    10 {
        value = Hello world!
    }
}
```

Ta zapis bi v PHP tabeli izgledal takole:

```
$TS['page'] = 'PAGE';
$TS['page.']['typeNum'] = '0';
$TS['page.']['10'] = 'TEXT';
$TS['page.']['10.']['value'] = 'Hello world!';
```

oziroma ekvivalentno

```
$TS = array(
    'page' => 'PAGE',
    'page.' => array(
        'typeNum' => '0',
        '10' => 'TEXT',
        '10.' => array(
            'value' => 'Hello world!'
        )
    )
)
```

### 2.5.3 Constants in Setup

Kam pa te ukaze v TypoScriptu sploh napišemo? Vsak program v TypoScriptu je vezan na predlogo strani. Kot smo omenili, vsako spletno stran naredimo tako, da si najprej izberemo predlogo. Pri tem v izbiri Templates najdemo polji Constants in Setup. Vse ukaze v TypoScript-u navedemo v ti dve polji.

V polju Constants poimenujemo konstante, ki jih bomo uporabljali in jim določimo vrednosti. Vrednosti konstant so npr.: velikost pisave, pisava, barva ozadja, ... pa tudi znakovni nizi, ki predstavljajo besedilo kot tako. V polju Setup konstante, ki smo jih definirali v polju Constants, lahko uporabimo. S pomočjo polja Setup, kot smo že omenili v razdelku 2.5.1, tudi definiramo, kako naj se vsebina, ki je shranjena v bazi, gradi dinamično na straneh.

Oglejmo si, kako bi zgled iz razdelka 2.6.2 naredili s pomočjo uporabe konstant.

V polje Constants vnesemo

```
mojeBesedilo.Content = Hello world!
```

S tem smo definirali konstanto z imenom mojeBesedilo (nanjo se bomo sklicevali z \$mojeBesedilo) in ji kot vsebino (Content) določili niz Hello world!

V polju Setup napišemo

```
page = PAGE
page.typeNum = 0
page.10 = TEXT
page.10.value = {$mojeBesedilo.Content}
```

Če torej želimo uporabiti konstanto kot vrednost, jo »ovijemo« v {}.

Kot rezultat bi dobili enako stran, kot jo prikazuje Slika 5. Če bi hoteli, da je napis napisan s krepko pisavo (**Hello world!**), bi polje Setup spremenili v:

```
page = PAGE
```

```
page{
  typeNum = 0
  10 = TEXT
  10.value = {$mojeBesedilo.Content}
  10.wrap = <strong>|</strong>
}
```

Edina prava razlika s prejšnjim zgledom je dodana objektna pot

```
page.10.wrap = <strong>|</strong>.
```

Z `wrap` smo vrednost objektna poti, ki je v našem primeru »Hello world!«, vstavili med znački `<strong>` in `</strong>`. `<strong>` je seveda ustrezeni ukaz iz HTML, | pa označuje objekt/lastnost 10 z lastnostjo `value` in njeno vrednostjo.

Uporabili smo tudi možnost, opisano prej, da kadar ima veliko lastnosti isto predpono (v našem primeru `page`), to napišemo le enkrat, nato pa v zavutih oklepajih navedemo lastnosti. To lahko tudi ponavljamo, torej bi lahko napisali

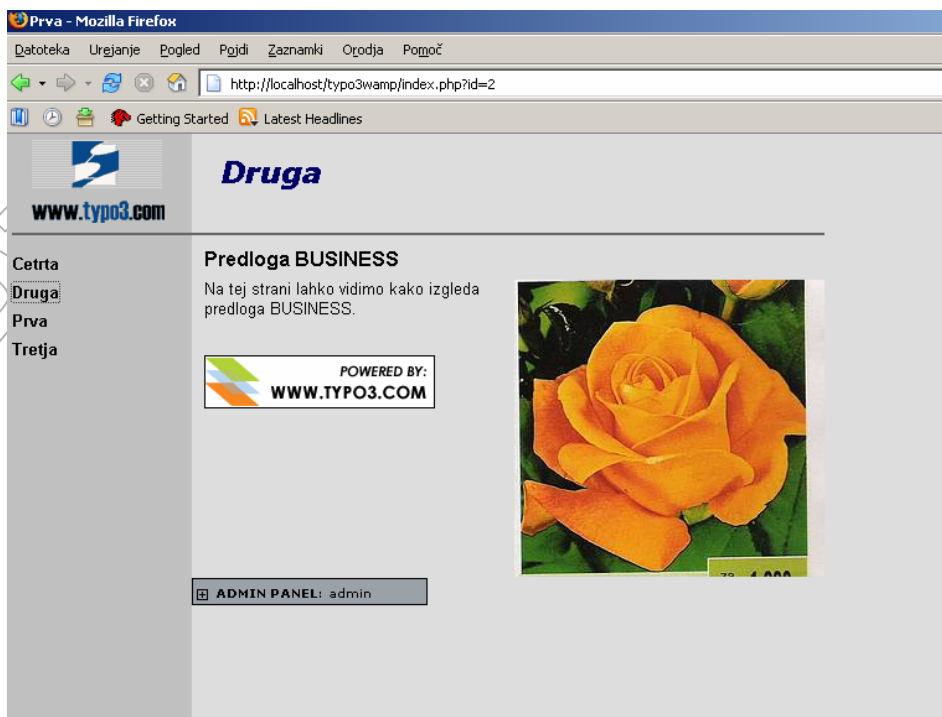
```
page = PAGE
page{
  typeNum = 0
  10 = TEXT
  10 {
    value = {$mojeBesedilo.Content}
    wrap = <strong>|</strong>
  }
}
```

## 2.5.4 Predloge

Predloga je struktura strani, kjer definiramo, kje na strani bomo imeli seznam povezav, kje bomo imeli prostor za slike, kje se bo prikazovala vsebina, ki jo bomo vnašali na stran in podobno.

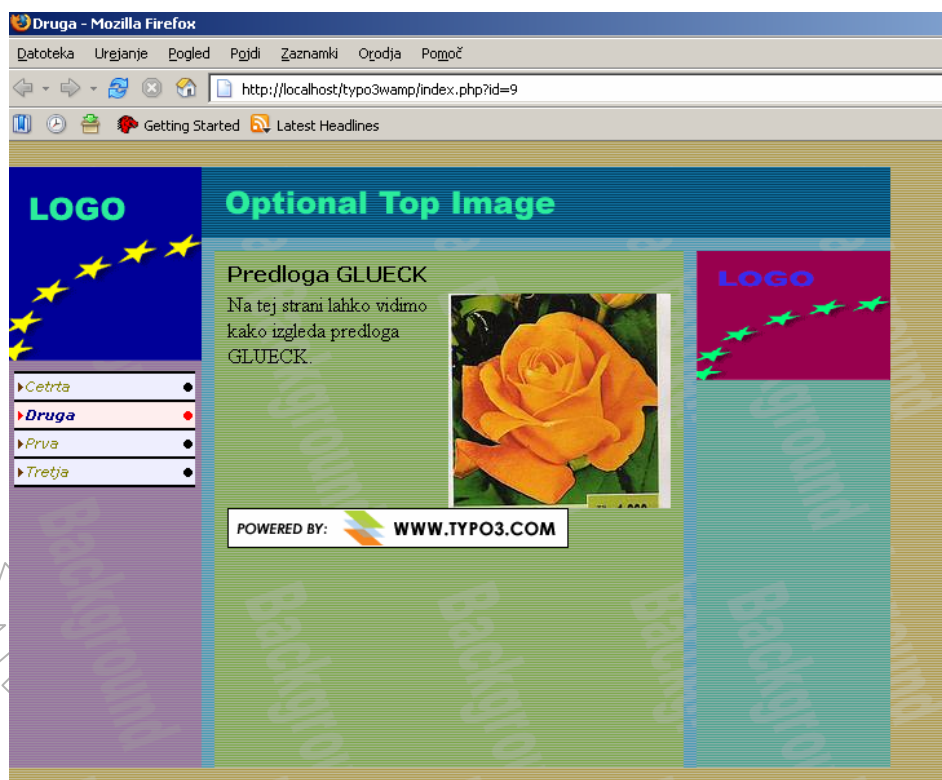
Vsaka stran ima lahko samo eno predlogo. Pri straneh, ki so v hierarhiji spletnega mesta pod neko stranjo, se predloga avtomatično podeduje od nadrejene strani. Ker bomo verjetno za večino strani, ki sestavljajo isto spletno mesto, uporabljali isto predlogo, nam ni potrebno za prav vsako stran posebej določiti predlogo, ampak samo za stran, ki je najvišje v drevesni strukturi strani. Lahko pa se zgodi, da želimo imeti za posamezno stran drugačno predlogo od ostalih. Takrat se premaknemo na to stran in ji določimo novo predlogo. Kako to naredimo, si bomo ogledali v razdelku 3.3.1.6.

V standardni distribuciji SUV Typo3 imamo na voljo 12 predlog, ki jih lahko uporabimo pri oblikovanju naših strani. Imena teh predlog so TU, RE, NEWSLETTER, HYPER, GREEN, GLUECK, FIRST, CrCPH, CANDIDATE, BUSINESS, MM in BUG. Vsaka izmed teh predlog ima svoje značilnosti. Oglejmo si le dve. Predloga BUSINESS ima dve območji. V levem so imena strani, ki jih vsebuje naš portal. Desno območje je namenjeno vsebini trenutno pregledovane strani.



Slika 6: Predloga BUSINESS z administracijsko ploščo

Pri predlogi GLUECK je stran razdeljena na tri stolpce oziroma tri območja. Levo je namenjeno seznamu imen strani, sredinsko območje je namenjeno vsebini strani, tretje oz. desno pa je namenjeno prikazu slike.



Slika 7: Predloga GLUECK

Poleg teh standardnih predlog lahko na internetu najdemo še veliko dodatnih. Stran, na kateri dobimo največ razširitev in dodatkov za Typo3, je že omenjena spletna stran <http://typo3.org>.

### 3. Sistem za upravljanje vsebin Typo3 (delo s programom)

#### 3.1 Namestitev Typo3 in zagon programa

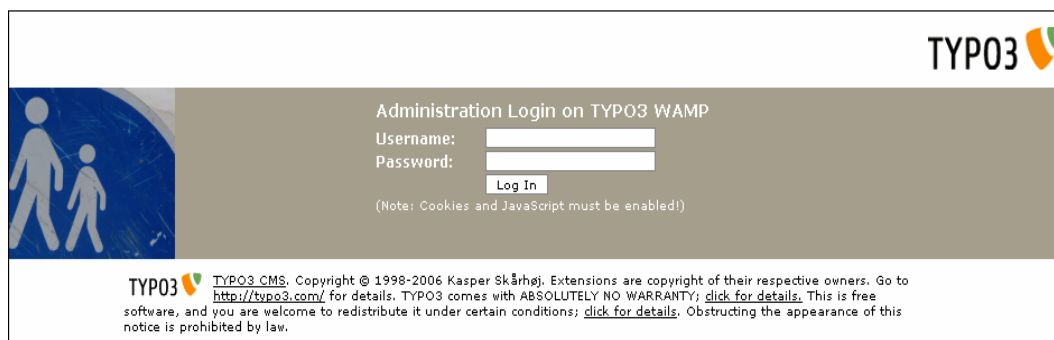
Ko želimo na naš računalnik namestiti program Typo3, imamo dve možnosti. Namestimo lahko celotni paket, ki vsebuje spletni strežnik Apache, strežnik za delo z bazami podatkov MySQL, programski paket za delo z jezikom PHP in sistem Typo3 ali pa namestimo samo sistem Typo3. Katero namestitev bomo izbrali, je predvsem odvisno od tega, kaj imamo na računalniku že nameščeno. Če na računalniku nimamo nameščenih vseh treh programov (Apache, MySQL in PHP) moramo vsekakor namestiti paket, ki vse to vsebuje.

Po uspešni namestitvi lahko začnemo delati s programom Typo3. Ker pa za zagon programa Typo3 ni povsem vseeno, katero izmed dveh možnosti namestitve izvedemo, bomo opisali obe.

Denimo, da smo namestili paket z vsemi potrebnimi programi. Takrat v meniju Start izberemo najprej Typo3 WAMP (oziroma Typo3 LAMP, če smo v operacijskem sistemu Linux). S tem zaženemo strežnike Apache, MySQL in PHP. Ko je enkrat to narejeno, v meniju Start izberemo še Typo3 Backend. S tem zaženemo zaledni sistem Typo3.

Če pa imamo omenjene tri strežnike že nameščene in delujoče in smo namestili samo sistem Typo3, v brskalnik vnesemo spletni naslov našega računalnika, ki mu sledi /typo3 (npr.: <http://193.235.176.45/typo3>). S tem zaženemo zaledni sistem. Pri tej možnosti moramo nato še konfigurirati podatkovno bazo za portal. Pri tej konfiguraciji določimo kakšno bo uporabniško ime in geslo za dostop do baze, kakšen bo naslov podatkovne baze in na koncu še uvozimo datoteko database.sql, ki jo najdemo v izbirnem seznamu pri konfiguraciji.

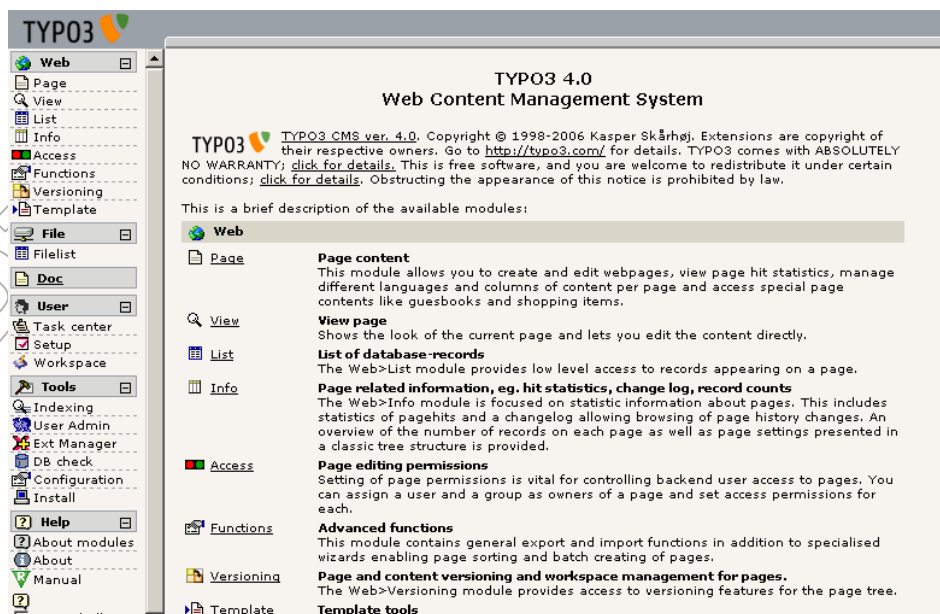
Preden v zalednem sistemu lahko sploh kaj počnemo, se moramo v sistem prijaviti. Odpre se pogovorno okno za vnos uporabniškega imena in gesla:



Slika 8: Prijavni obrazec

Če smo administrator sistema, potem je ob prvi prijavi naše uporabniško ime admin in geslo password. Ob prvi prijavi je nujno zamenjati geslo. V primeru, da nismo administrator sistema, vnesemo svoje uporabniško ime in geslo, kot nam ju je dodelil administrator spletnih strani. Po prijavi se odpre začetni zaslon. Začetni zaslon, kot ga prikazuje Slika 9, se prikaže, če imamo administracijske pravice. Če administracijskih pravic nimamo, zagledamo podoben zaslon, le z manj funkcijami in možnostmi.





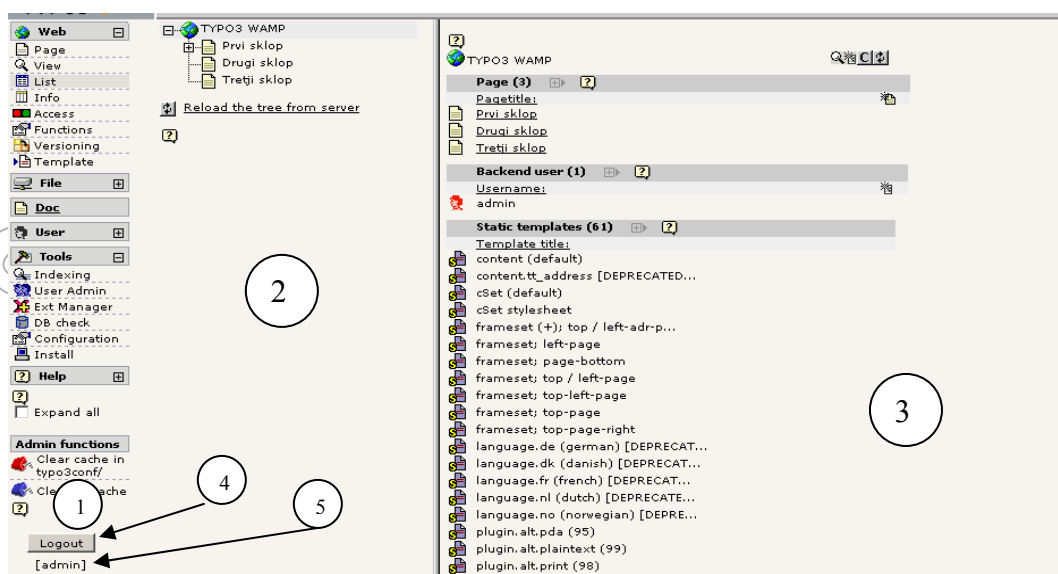
Slika 9: Začetni zaslon – administrativni pogled

## 3.2 Površina in njena območja

### 3.2.1 Pregled delovne površine

Delovna površina Typo3 je razdeljena na tri glavna področja:

- 1) Izbirni del – v izbirnem delu najdemo izbire, ki jih lahko uporabimo pri delu s programom. Vsaka izbira nam služi za določeno opravilo v Typo3. Nekatere izmed izbir bodo opisane v nadaljevanju diplomske naloge.
- 2) Navigacijski del – v navigacijskem delu se nam na podlagi izbire, ki smo jo izbrali v izbirnem delu, prikaže ali drevesna struktura spletnih strani ali pa drevesna struktura map za shranjevanje slik in ostalih datotek.
- 3) Podrobni prikaz – predstavlja področje v katerem administrator ali urednik zalednega sistema ustvarja spletne strani, dodaja vsebine na strani, razširitve, ind.



Slika 10: Pregled delovne površine

Na površini sta vedno prikazana:

- 4) Gumb Odjava (Logout) – ko želimo končati delo, se odjavimo s tem gumbom. Typo3 shrani naš zadnji delovni položaj in nas pri naslednji prijavi vrne na ta položaj.
- 5) Uporabnik – trenutno prijavljeni uporabnik

### 3.2.2 Izbirni del

V izbirnem delu lahko izberemo določeno izbiro, s katero želimo nekaj narediti v programu Typo3. Pri vsaki izbiri imamo še razne podizbire. Tako recimo izbira Web vsebuje podizbire Page, View, List, Info, Access, Functions in Template. Vsaka izmed teh podizbir ima neko funkcionalnost v Typo3. Z določenimi izbirami ustvarjamo nove strani, z drugimi dodajamo vsebino na stran, itd. Opis nekaterih izbir bo opisan v nadaljevanju.

### 3.2.3 Navigacijski del

V navigacijskem delu vidimo meni izbrane izbire iz izbirnega dela. Ko s klikom na levo tipko na miški izberemo določeno izbiro (npr. »Web> Page«), je v navigacijskem delu na razpolago tej izbiri pripadajoči meni. Tako se z izbiro »Web>Page« v navigacijskem delu prikaže drevo spletnih strani, z izbiro »File>Filelist« pa drevo map. Drevo map je prikazano na način, kot ga poznamo iz orodja Raziskovalec v operacijskem sistemu Windows.

Če je prikazano drevo spletnih strani, se ob kliku z levo tipko miške na eno izmed teh strani odpre kontekstni meni in ponudi naslednje možnosti: pokaži stran v čelnem delu, uredi stran, nova spletna stran oz. vsebina na strani, kopiraj stran, izreži stran, skrij stran, izbriši stran. Kontekstni meni največkrat uporabljamo pri dodajanju novih vsebin na stran oz. pri ustvarjanju novih strani.

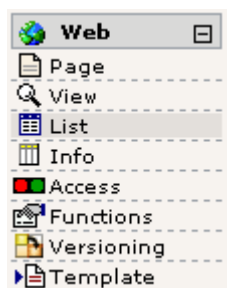
### 3.2.4 Podrobni prikaz

Na področju Podrobni prikaz leži osrednje delovno področje Typa3. V podrobnem prikazu dodajamo vsebine na stran, ustvarjamo nove strani, dodajamo razširitve na stran. Vse, kar počnemo v Typo3 in ima odraz na spletni strani, torej počnemo na podrobnem prikazu. Podrobni prikaz se vedno prilagodi glede na izbrano izbiro v izbirnem delu.

## 3.3 Glavne izbire Typa3

### 3.3.1 Izbira »Web«










Tukaj najdemo najpomembnejše izbire, ki definirajo tako vsebino kot tudi izgled spletne strani. Vsaka podizbira iz izbire Web nam nudi neko funkcionalnost v povezavi s stranjo. Tako lahko z določeno podizbiro določimo strani predlogo oz. strukturo. Z eno izmed podizbir urejamo pravice uporabnikov.





Slika 11: Izbira "WEB"

#### 3.3.1.1 Page

Ta izbira je namenjena urejanju posameznih strani. Ko v drevesni strukturi izberemo določeno stran (kliknemo nanjo), v podrobnem prikazu zagledamo prikaz te strani (Pagecontent), po strani lahko

iščemo (Search) in imamo na voljo določene druge funkcije (Advanced functions). Prav na vrhu podrobnega prikaza pa imamo dostop do kontekstnega menija za stran (New, Edit,...) - ikona . Druga ikona  nam prikaže vse vsebine, ki se nahajajo na strani (podobno izbiri List). Tretja ikona  pa nam prikaže stran v čelnem sistemu. Poleg teh treh ikon imamo še šest drugih ikon, ki nam omogočajo različne možnosti oz. opravila. Prva izmed njih je . Ob kliku na to ikono dobimo seznam sprememb or. arhiv sprememb vsebin na strani. Tako lahko iz arhiva sprememb izvemo katere vsebine so bile izbrisane, katere popravljene itd. Druga ikona je . Ob kliku nanjo se v podrobnem prikazu odprejo tipi vsebin, ki jih lahko dodamo na stran. Tretja ikona je . Ob kliku na ikono nam je omogočeno premikanje strani po drevesu strani. Četrta ikona je . S pomočjo te ikone lahko stran kopiramo in to kopijo strani dodamo v drevo strani. Peta ikona je . Z njo popravljamo nastavitve strani. Ob kliku nanjo dobimo formo z nastavitvami za stran. Šesta ikona je . Z njeno pomočjo dobimo opis, kaj lahko počnemo z izbiro Page v načinu Columns.

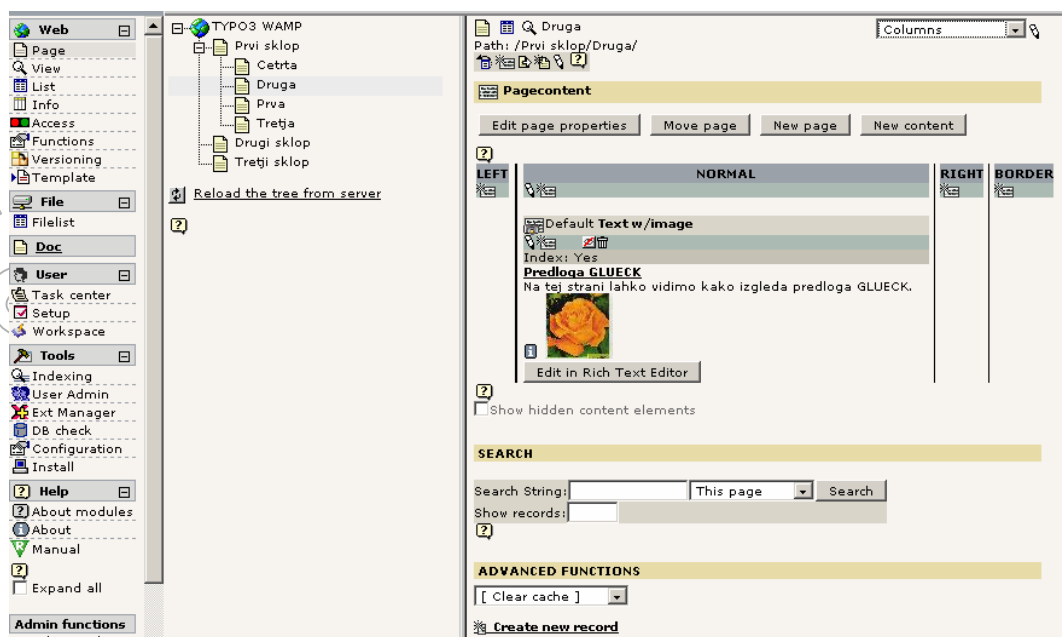
S klikom na gumb »New page« ustvarimo novo stran, ki je hierarhično pod to izbrano stranjo ali pa je na istem nivoju. Ko novo stran naredimo, se preselimo nanjo in jo začnemo urejati. Gumb »Move page« nam omogoča premikanje strani. Premikanje strani pomeni, da stran v drevesu strani izrežemo in premaknemo na drugo mesto v drevesu strani. Če premaknemo stran, ki ima še druge podstrani, se ob premiku tudi te premaknejo. Tako se hierarhija strani, ki so pod stranjo, ki jo premikamo, ohranja.

S pomočjo gumba »Edit in Rich Text Editor« lahko urejamo besedni del vsebine v urejevalniku. Ta je zelo podoben urejevalniku v MS Word. S klikom na  vsebino skrijemo. Tako ni vidna v čelnem sistemu. Z ikono  izberemo vsebino na strani ob kateri se nahaja ikona.

Vse zgoraj opisane možnosti so na voljo, če imamo izbran način »Columns« v izbirnem seznamu desno zgoraj v podrobnem prikazu.









Preko načina »Page information« v desnem zgornjem kotu imamo pregled nad informacijami o strani. Informacije, ki jih lahko izvemo, so datum nastanka strani, zadnja sprememba strani, zadnja sprememba vsebin na strani ind.


Glavni namen izbire Page je urejanje strani na kateri se nahajamo oz. strani, ki jo izberemo. Pod urejanjem strani mislimo na dodajanje vsebin, popravljanje vsebin in drugih nastavitvev strani, premikanje strani, ...
















Slika 12: Izbira Page

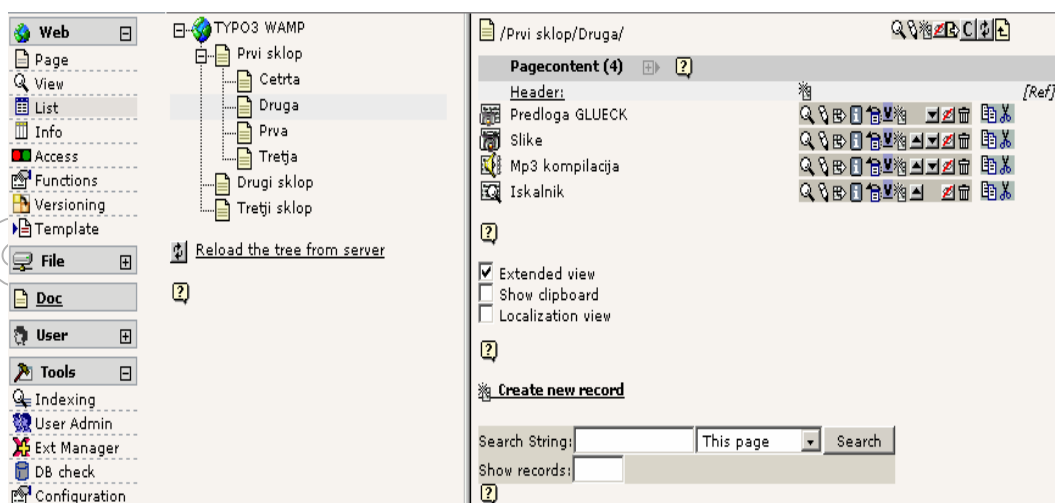
### 3.3.1.2 List

Ta izbira je namenjena pregledu in urejanju posameznih vsebin na strani. Ko v drevesni strukturi izberemo določeno stran (kliknemo nanjo), v podrobnem prikazu zagledamo prikaz te strani (Pagecontent). Po strani lahko iščemo (Search). V desnem zgornjem kotu podrobnega pogleda imamo veliko ikon, ki nam omogočajo razne funkcionalnosti. Ikona  nam prikaže stran v čelnem sistemu. Druga ikona je . S pomočjo te ikone imamo možnost popraviljanja nastavitvev za stran. Ob kliku na ikono se odpre forma z nastavitvami za stran. Z ikono  imamo v podrobnem prikazu na voljo razne možnosti. Tam je čarovnik za ustvarjanje nove strani, čarovnik za ustvarjanje nove vsebine strani itd. Z  stran, na kateri se nahajamo, v čelnem sistemu skrijemo. Naslednja ikona je . Ob kliku nanjo strani lahko premikamo po drevesu strani. Z ikono  izpraznimo predpomnilnik za to stran. Ikona  nam osveži stran, na kateri se nahajamo. Na koncu nam ostane ikona . Ob kliku na to ikono se pomaknemo po drevesu strani za en nivo više.

Tudi pri izbiri List imamo na dnu podrobnega prikaza še  **Create new record**. Ob kliku se v podrobnem prikazu odprejo razne možnosti. Tako je na voljo možnost uporabe čarovnika za ustvarjanje nove strani oz. nove vsebine za stran itd. Vsebine, ki so lepo pregledno našteje, lahko tudi na zelo enostaven način urejamo. Na ikono pred imenom vsebine kliknemo z levim gumbom miške in odpre se kontekstni meni. V kontekstnem meniju izberemo, kar želimo početi z izbrano vsebino (Edit, New, Copy, Cut, Hide,...).

Z vklopom polja  **Extended view** se nam pri naštetih vsebinah pojavijo še ikone, ki nam ponujajo razne funkcionalnosti (Slika 13). Opišimo, kaj lahko počnemo s temi ikonami. Prva ikona je . Ikona nam prikaže stran v čelnem sistemu. Z  urejamo vsebino, pri kateri smo kliknili to ikono. Z  lahko prestavljamo vsebino po strani na kateri se nahajamo oz. lahko vsebino premaknemo na katerokoli drugo stran. Ikona  nam poda osnovne informacije o vsebini (tip vsebine, naslov vsebine, tekst vsebine, na kateri strani se vsebina nahaja, ...). Ikona  nam prikaže arhiv sprememb za izbrano vsebino. Z  izvemo id in pid strani, na kateri je vsebina. Z  naredimo novo vsebino takoj pod izbrano vsebino. Ikona  nam omogoča, da izbrano vsebino pomaknemo po eno mesto navzgor v seznamu vsebin na strani. Ikona  nam omogoča, da izbrano vsebino pomaknemo po eno mesto navzdol v seznamu vsebin na strani. Z  lahko vsebino na strani skrijemo. Z  lahko vsebino na strani izbrišemo. Z  lahko vsebino na strani kopiramo in z  lahko vsebino na strani izrežemo.

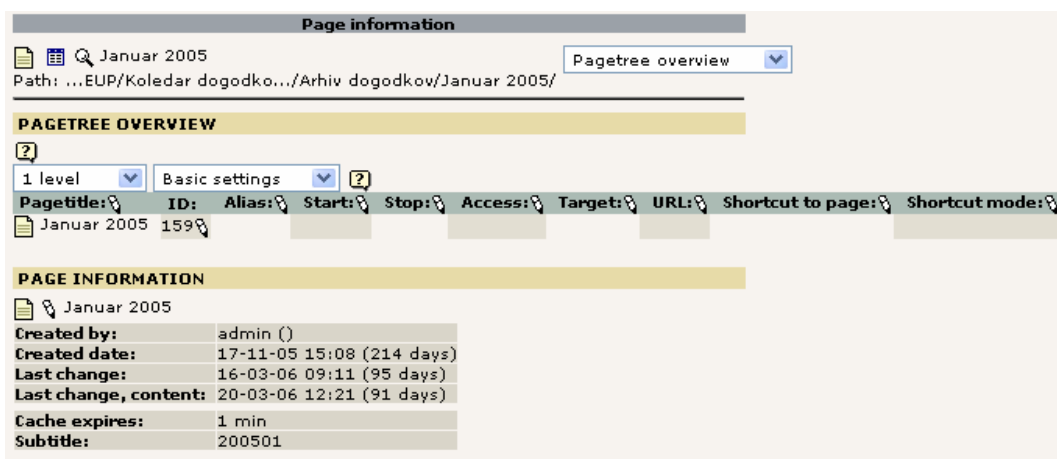
Glavni namen izbire List je urejanje vsebin na strani na kateri se nahajamo oz. strani, ki jo izberemo.



Slika 13: Izbira List

### 3.3.1.3 Info

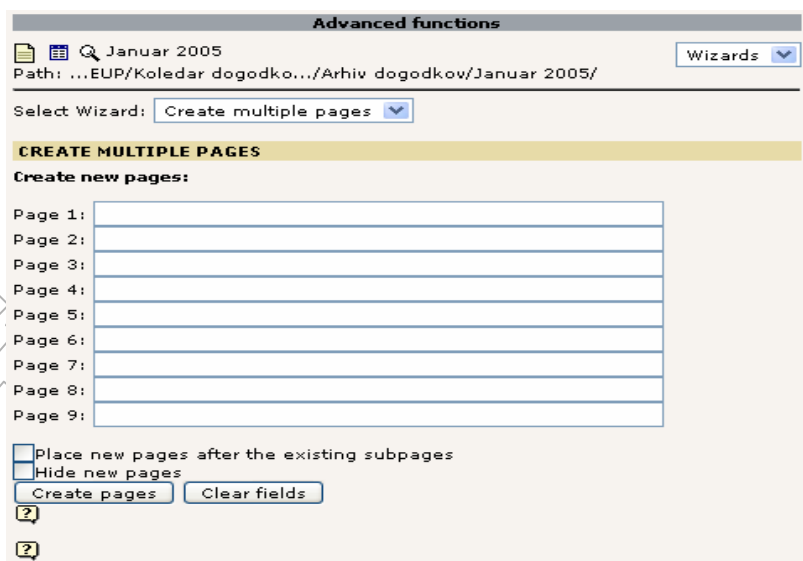
Ker se ta izbira redkokdaj uporablja, spregovorimo o njej le par besed. Izbira »Info« zdi združuje skoraj vse pomembne statistične informacije o določeni strani. V načinu »Pagetree overview« so prikazane tako informacije o strani (kdaj in kdo jo je ustvaril, čas zadnje spremembe, ključne besede, navedba vsebine, status, itd). Te informacije so nam pogosto zelo koristne, saj imamo na enem mestu podatke o celotni strani. Na podlagi teh informacij takoj vidimo, kdo je avtor strani in kar je najpomembnejše, kdaj je bila nazadnje stran urejena. Poleg načina »Page overview« imamo tukaj še več načinov, ki nam ponujajo določene informacije o strani. Izbiro »Info« večinoma uporabljamo le za splošne informacije o strani, zato opis ostalih načinov izpustimo.



Slika 14: Izbira Info

### 3.3.1.4 Functions

V izbiri Functions so zbrani različni postopki, ki nam pomagajo pri učinkovitem delu s sistemom Typo3. Ena od bolj uporabnih stvari v tej izbiri je možnost, da naenkrat naredimo celo vrsto podobnih strani. V ta namen poskrbimo, da je v izbirnem seznamu desno zgoraj izbran način "Wizards". Kot ustreznega čarovnika si izberemo "Create mutiple pages" (Slika 15). V seznam vpišemo naslove novih strani, ki jih želimo narediti. Po kliku na gumb "Create pages" dobimo nove strani brez vsebine. Strani bi se v našem primeru naredile pod stranjo Januar 2005. Informacijo, katera stran je trenutno aktivna (torej pod katero se bodo ustvarile te podstrani), nam pove vrstica z začetno besedo Path. V njej je zapisana celotna pot do te strani.






Slika 15: Izbira Functions

Drugi uporabni čarovnik je "Sort pages". Čarovnik nam lahko sortira spletne strani po imenu, datumu nastanka strani, ... Ker pa to počnemo večinoma le, ko smo že izkušeni uporabniki SUV Typo3, bomo podroben opis te in ostalih možnosti tu izpustili.

### 3.3.1.5 Access

V izbiri »Access« urejamo pravice dostopa urednikom do posameznih strani. Izbiro uporabimo, ko želimo določenim stranem določiti, kdo od urednikov in uredniških skupin jih lahko vidi in na njih kaj počne. Uredniška skupina je skupina urednikov, ki imajo enake pravice.

Vsaka stran ima svojega lastnika. To je bodisi posamezni urednik ali pa uredniška skupina. Ko stran ustvarimo, je lastnik te strani prijavljeni uporabnik, kasneje pa lahko lastništvo spremenimo. To naredimo tako, da kliknemo izbiro Access. V navigacijskem delu izberemo stran, ki ji želimo spremeniti lastništvo. Poskrbimo, da je v izbirnem seznamu desno zgoraj izbran način "Permissions". Ko najdemo v podrobnem pogledu stran, ki ji želimo spremeniti lastnika ali druge pravice, kliknemo na ikono – svinčnik , ki se nahaja za imenom strani npr.  Januar 2005 . Primer si lahko ogledamo na slikah 16 in 17.

Uredniki imajo lahko naslednje pravice:

- Show page (prikaži stran) – če ima urednik to dovoljenje, lahko določi, ali se ta stran prikaže ali ne v čelnem sistemu (torej običajnim uporabnikom).
- Edit content (uredi vsebino) – uredniki s tem dovoljenjem bodo lahko spreminjali vsebino strani. Tako na primer bodo lahko popravili besedilo, spremenili sliko, Ne bodo pa mogli spremeniti same strukture strani. Za to potrebujejo dovoljenje edit page.
- Edit page (uredi stran) – s tem dovoljenjem lahko spreminjamo strukturo strani. Lahko dodamo določene elemente, določene izbrišemo, zamenjamo njihov položaj, predloge pa ne moremo zamenjati. Zamenja jo lahko le administrator, saj ima le on dostop do izbire Template s katero urejamo predloge.
- Delete page – s tem dovoljenjem lahko strani in vsebine na straneh brišemo
- New pages – če nimamo tega dovoljenja, potem ne moremo ustvariti strani, ki so tej strani podrejene

Na sliki 16 vidimo, da avtor (lastnik) strani lahko izvede vse zgoraj omenjene akcije. Prav tako lahko pripadniki skupine povsem spreminjajo stran, le izbrisati je ne morejo. Običajni uredniki zalednega sistema pa ne morejo početi ničesar v zvezi s stranjo.

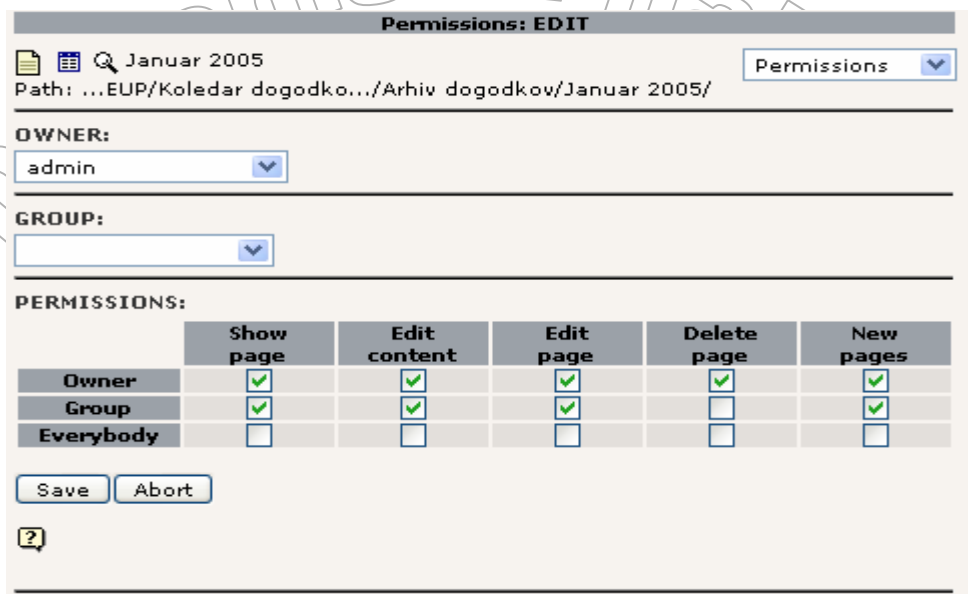


Slika 16: Izbira Access - splošen pogled

Če želimo pravice spremeniti oz. določiti novega lastnika strani ali pa zamenjati skupino, ki ima dostop do te strani, kliknemo na "svinčnik" ob imenu strani (na sliki 16 se svinčnik nahaja ob naslovu



strani Januar 2005) in zagledamo nastavitve, kot so na sliki 17. Po spremembi nastavitvev kliknemo gumb "Save" in nove nastavitve začnejo veljati.



Slika 17: Izbira Access - podroben pogled

### 3.3.1.6 Template

S pomočjo te izbire določimo vrhajo predlogo, stranem spreminjamo predlogo, kot tudi sestavimo novo predlogo spletnih strani ali pa popravimo eno izmed že obstoječih predlog. O sestavljanju novih predlog, njihovem urejanju in shranjevanju ne bomo govorili. Omenili bomo le, kako določimo vrhajo predlogo in kako popravimo stranem že prirejene predloge.


Kot smo omenili že prej, vsaka stran avtomatično podeduje predlogo nadrejene strani. Vrhajo, torej glavno predlogo, določimo tako, da si iz izbirnega seznama nad gumbom »Create template for a new site« (Slika 18) izberemo ustrezno predlogo (med tistimi standardnimi 12 in tistimi, ki smo jih uvozili kot razširitve) in kliknemo na ta gumb. Da sploh vidimo ta gumb, smo morali prej iz spustnega seznama desno zgoraj izbrati način »Constant Editor«.

Če v izbirnem seznamu nismo izbrali nobene izmed predlog in smo vseeno kliknili gumb, se je zgenerirala predloga, ki nam da na stran napis »HELLO WORLD!«.

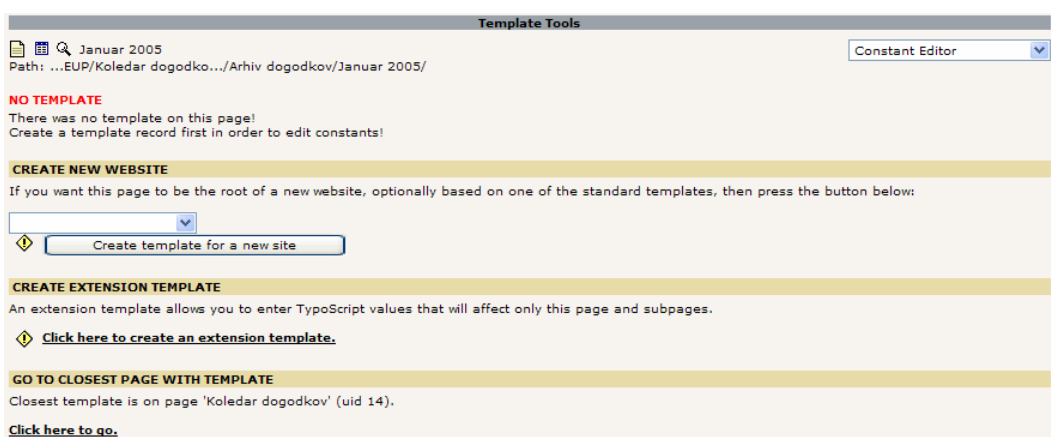
Pri določitvi zgornje predloge so vse strani privzele to predlogo. Če želimo neki od podstrani spremeniti predlogo, v izbiri Template preprosto kliknemo na stran in poskrbimo, da je v spustnem seznamu desno zgoraj izbran način »Constant Editor«. Kot vidimo na sliki 18 nam sistem pravi, da stran nima predloge. To sporočilo moramo razumeti tako, da je predloga te strani pač tista privzeta od nadrejene strani. Nato, kot smo storili prej za vrhajo predlogo, izberemo ustrezno predlogo in kliknemo na gumb »Create template for a new site«. Napis na gumbu je sicer zavajajoč, saj ne ustvarjamo predloge za novo spletno mesto, a naredi želeno. S tem smo tej strani spremenili predlogo.

Ko želimo predlogo urejati, se v navigacijskem delu premaknemo na stran, ki ji želimo urediti predlogo. Poskrbimo, da imamo v Podrobnem prikazu izbran način »Constant Editor«. Sedaj imamo možnost urejanja predloge (slika 19). To storimo lahko le na straneh na katero je bila predloga dodana. Na straneh, ki so predlogo podedovale, predloge ne moremo urejati. Premakniti se moramo na ustrezno nadrejeno stran.

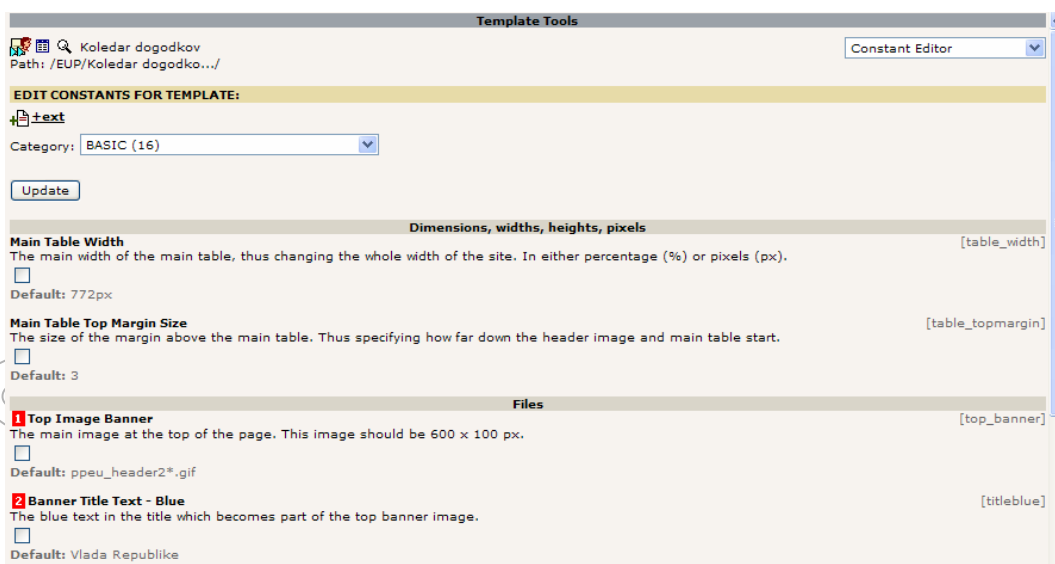
Urejamo lahko kakšna bo barva povezav v seznamu strani, velikost pisave v čelnem sistemu za vsebine, velikost logotipa v zgornji pasici strani, itd. Če želimo predlogi na strani kaj dodati, lahko to naredimo tako, da izberemo v spustnem seznamu desno zgoraj izbran način »Info/Modify«. Nato

kliknemo na »Click here to edit whole template record«. S tem smo si odprli obrazec z nastavitvami za predlogo na strani. Nato lahko v polji Constants in Setup vpišemo ukaze, ki bodo omogočali dodatne funkcije na predlogi. Lahko pa tudi spremenimo predlogo za stran. Tako v polju Include static: s pomočjo gumba  pobrišemo trenutno predlogo na strani in s klikom na eno izmed naštetih v seznamu, izberemo drugo predlogo. S tem, ko smo strani spremenili predlogo, se nam ni treba bati, da smo ob spremembi karkoli izgubili. Vsebine na strani so ostale enake, le prikazane bodo drugače. Če smo prej imeli vsebino v enem stolpcu in nam nova predloga prikazuje vsebino v dveh stolpcih, bo prejšnja vsebina prikazana v dveh stolpcih. Razlika bo le, da bodo zaradi dveh stolpcev vrstice vsebin krajše. Pri spremembi vrhnje predloge se bo sprememba poznala tudi na podstraneh, razen seveda na tistih katerih smo jim posebej določili drugačno predlogo.

Poleg načina »Constant Editor« obstaja še pet načinov, ki nam omogočajo na predlogi izvesti različne stvari. Eden izmed bolj uporabnih je način »CSS Styler«. Če ga izberemo, vidimo kaskadne stile, ki so uporabljeni na strani. Z uporabo enostavnega urejevalnika lahko te stile ustrezno spremenimo. Mi načina »CSS Styler« ne bomo opisovali podrobneje, ker ga pri našem projektu nismo uporabili.



Slika 18: Izbira Template – nova predloga



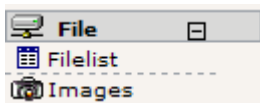
Slika 19: Izbira Template 2 – urejanje predloge

### 3.3.2 Izbira »File«

Izbiri »File« uporabimo, ko želimo dodati na strežnik določene datoteke (.doc, .xls, .txt, .ppt, .sxw, .mp3, ...) in slike, ki jih bomo potrebovali pri spletnih straneh. Slike in datoteke lahko dodamo bodisi z izbiro Filelist bodisi z izbiro Images. Slike in ostale datoteke, ki jih na spletni strani lahko večkrat



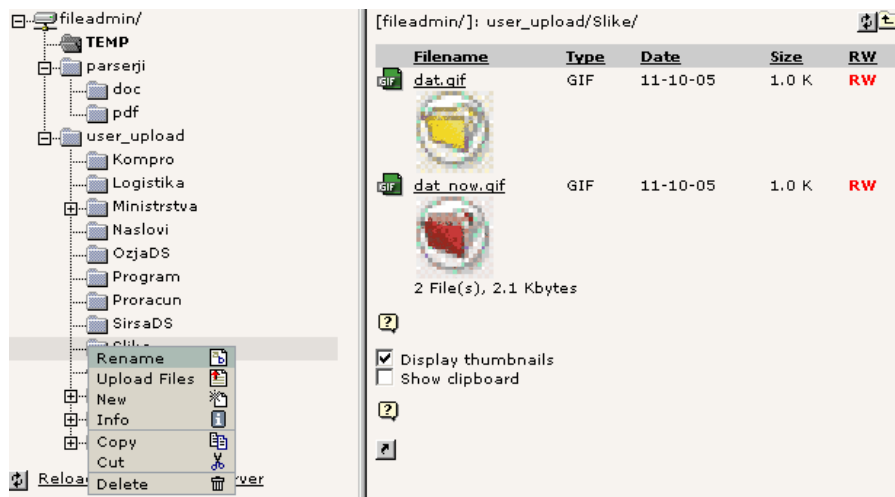
uporabimo, so shranjene v pregledni strukturi map. Struktura je zelo podobna Raziskovalcu v operacijskem sistemu Windows. V tej izbiri lahko pregledujemo katere slike in datoteke imamo na strežniku in jih lahko uporabimo.



Slika 20: Izbira "File"

### 3.3.2.1 Filelist

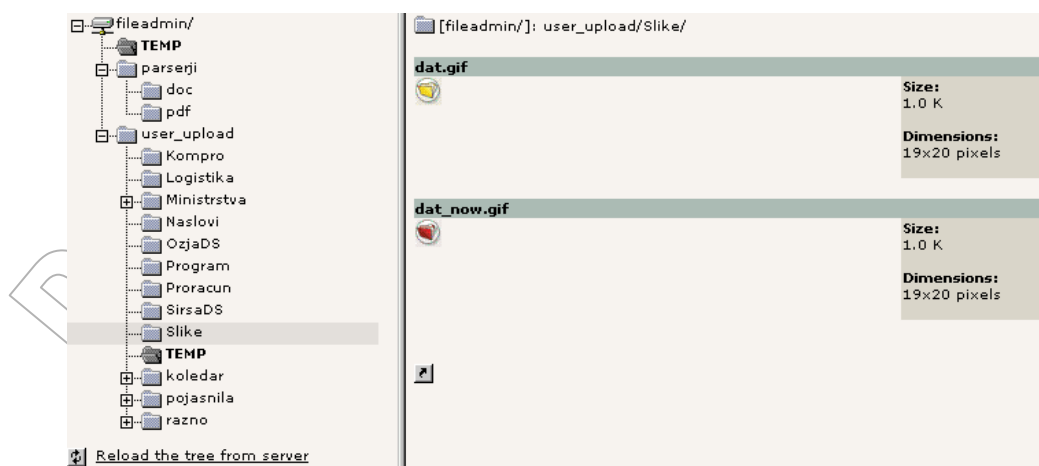
Izbira »Filelist« predstavlja sistem za upravljanje z datotekami, potrebnimi za spletno mesto, ki ga nadziramo s SUV Typo3. Uporabimo jo, ko želimo na strežnik dodati datoteke ali slike. Pri izbiri »Filelist« nam je glede na naše uporabniške pravice omogočen dostop do map datotek. S klikom z levo tipko na določeno mapo v stranskem meniju/drevesu dobimo na razpolago meni (Slika 21), s katerim lahko mape preimenujemo, ustvarimo nove in naložimo datoteke ali slike na strežnik, kopiramo, izrežemo ali izbrisemo datoteke.



Slika 21: Izbira Filelist

### 3.3.2.2 Images

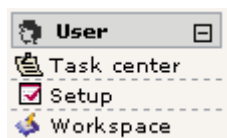
Izbira »Images« nam služi za dodajanje slik na strežnik. Postopek dodajanja je enak kot pri izbiri "Filelist". V tej izbiri slik ni mogoče kopirati in prestavljati. To moramo narediti z izbiro "Filelist". Izbire se največkrat poslužujemo samo, ko si želimo ogledati slike, ki jih imamo na strežniku. Dobra stvar te izbire je, da prikaže velikost in dimenzije slik.



Slika 22: Izbira Images

### 3.3.4 Izbira »User«

Tu so v podizbirah zbrane tiste izbire, ki služijo da si priredimo delovno okolje v zalednem sistemu. Izmed vseh je najpomembnejša izbira »Setup».

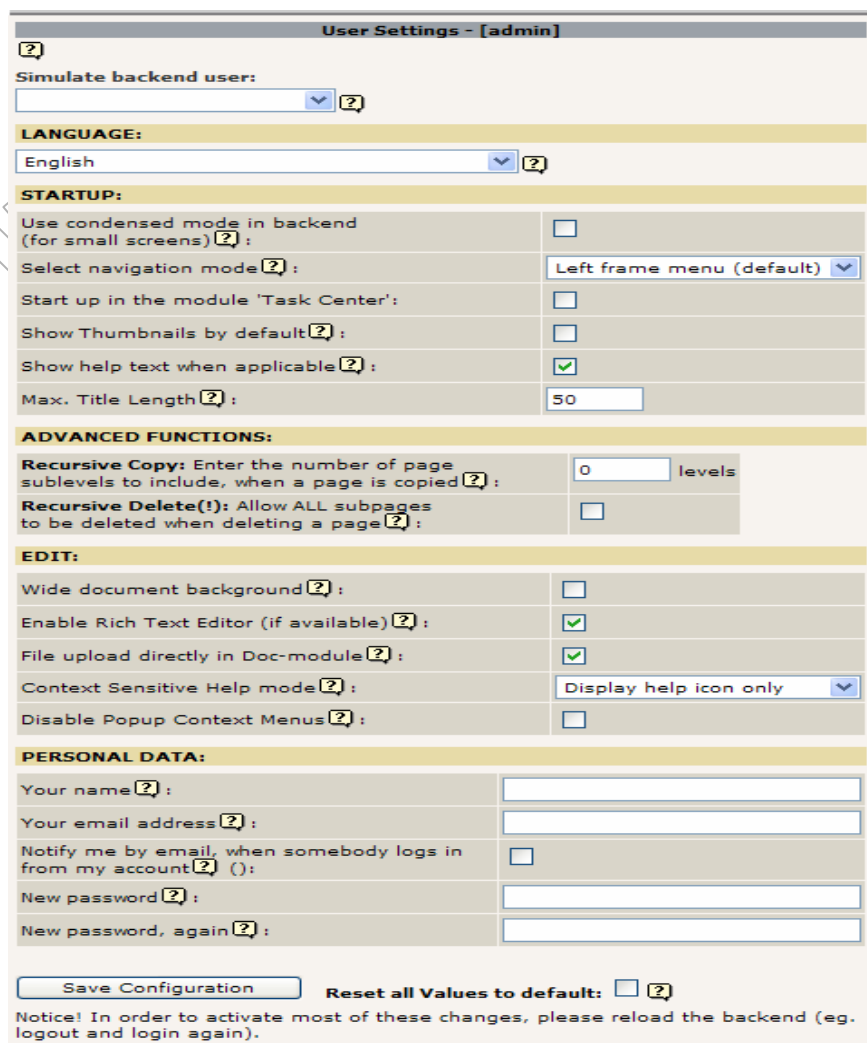


Slika 23: Izbira "User"

#### 3.3.4.1 Setup

Z izbiro Setup si lahko prilagodimo svoje delovno okolje po svojih željah. Typo3 tako ponuja uporabnikom zalednega sistema možnost, da vsak posameznik izbere jezik, ki ga želi uporabljati v zalednem sistemu. Pogoj je seveda ta, da za izbrani jezik v sistemu obstaja razširitev s prevodom za ta jezik. Nastavimo lahko tudi ali naj bo dovoljeno rekurzivno kopiranje oziroma brisanje. Pri rekurzivnem kopiranju se poleg strani, ki jo kopiramo, kopirajo tudi vse podstrani do določenega nivoja v drevesni strukturi. Če si označimo možnost rekurzivnega brisanja dosežemo, da bo ob brisanju strani sistem pobrisal tudi vse postrani, ki jih stran vsebuje. Če te možnosti ne označimo, strani, ki vsebuje podstrani, sploh ne bomo mogli izbrisati. Najprej bomo morali pobrisati podrejene strani in šele potem zeleno stran.

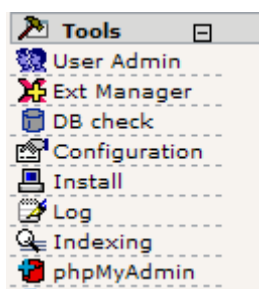
Da povečamo nadzor nad uporabo sistema lahko nastavimo možnost, da vedno, ko se kdo z našim uporabniškim imenom prijavi v zaledni sistem, dobimo na svoj e-poštni naslov sporočilo. Prav tako tu spremenimo geslo za prijavo v zaledni sistem. Po spremembi nastavitve kliknemo gumb »Save Configuration«. V brskalniku stran še osvežimo, da začnejo nastavitve veljati.



Slika 24: Izbira Setup

### 3.3.5 Izbira »Tools«

Izbire v »Tools« uporabljamo, pri administraciji uporabnikov, ko želimo uvažati nove razširitve v sistem (izbira Ext manager), urejati bazo podatkov (phpMyAdmin), nastavljeni določene konfiguracijske parametre SUV Typo3 in podobno. Največ uporabljamo izbiri »Ext Manager« in »phpMyAdmin«, ki ju bomo v nadaljevanju tudi bolj podrobno opisali. Izbirno področje »Tools« je na voljo le administratorju sistema. Iz tega sledi, da tudi z razširitvami lahko upravlja le administrator sistema. Enako velja tudi za administracijo baze podatkov.

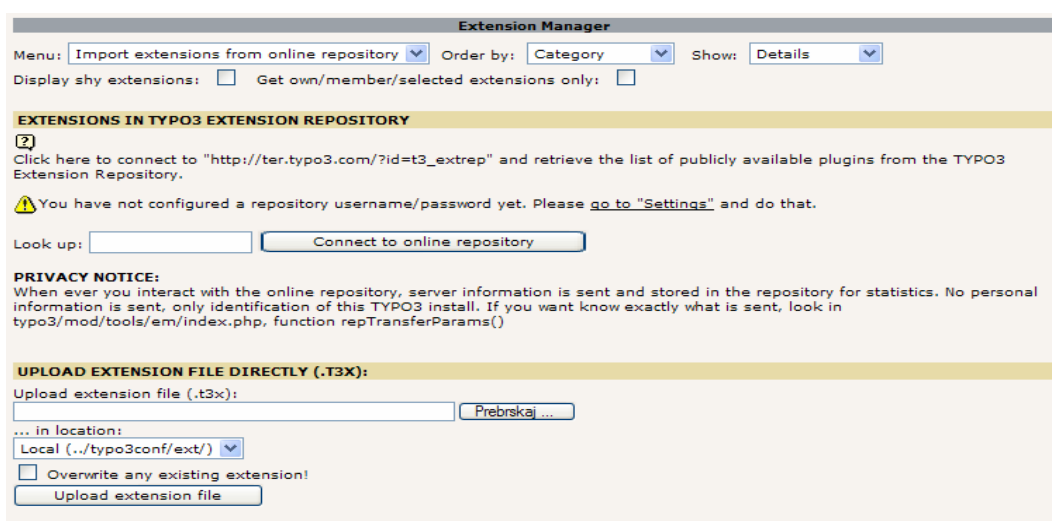


Slika 25: Izbira "Tools"

### 3.3.5.1 Extension Manager

Upravljalca razširitev, Extension Manager, s katerim upravljamo razširitve, je v bistvu tudi razširitev Typo3. Vendar je v sistem dodan takoj, ko sistem Typo3 namestimo na računalnik. Omogoča nam, da z njim v sistem uvažamo nove razširitve in da jih namestimo. Typo3 namreč razlikuje med postopkom uvoza razširitve, ko na ustrezno mesto na strežniku dodamo datoteke, ki sestavljajo razširitev in samo namestitev razširitve, ko med uvoženimi razširitvami izberemo tiste, ki postanejo del sistema.

Uvoz razširitev opravimo tako, da najprej kliknemo na izbiro Ext Manager in v polju Menu izberemo način »Import extensions from online repository« (Slika 26). V polje »Look up:« vneseemo ključ razširitve (extension key) in kliknemo gumb »Connect to online repository«. Če obstaja razširitev z navedenim ključem, se nam ta prikaže v novem oknu. V sistem jo uvozimo tako, da kliknemo na beli kvadrček z rdečo puščico (Slika 27). Razširitev pa lahko uvozimo v sistem tudi na drugačen način. Namreč razširitve lahko dobimo tudi v obliki arhiva. Ta je v obliki datoteke s končnico .t3x. V polje »Upload extension file (.t3x)« (Slika 26) s pomočjo gumba »Prebrskaj« vpišemo pot do te datoteke. Nato kliknemo na gumb »Upload extension file« in razširitev je uvožena v sistem.

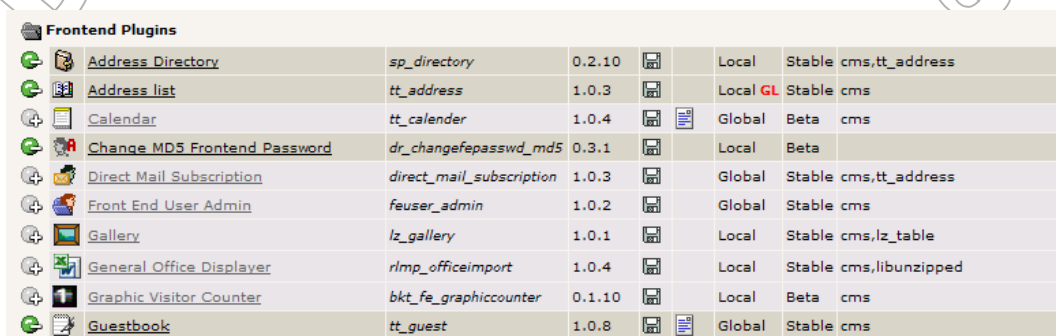


Slika 26: Razširitev Extension Manager - način Import extensions from online repository



Slika 27: Razširitev Extension Manager - način Import extensions from online repository

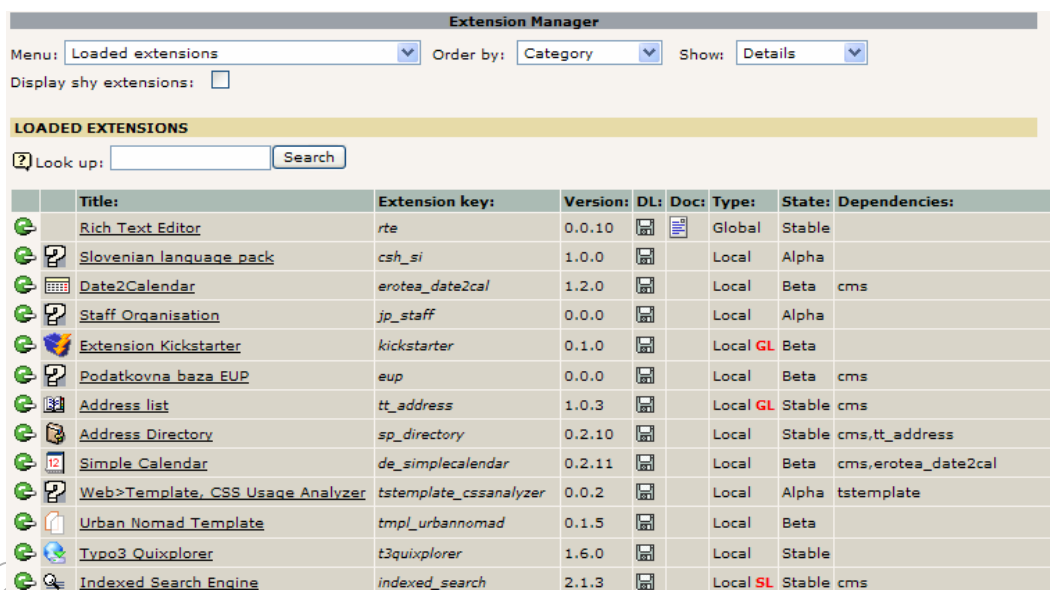
Če želimo razširitev namestiti v sistem, v polju Menu izberemo način »Available extensions to install«. Zagledamo seznam razširitev, ki so bodisi že nameščene bodisi pripravljene za namestitev. Prve so označene z zelenim krogcem z minusom, druge pa s sivim krogcem s plusom. Pomaknemo se do razširitve, ki jo želimo namestiti in kliknemo na ikono, siv krogec s plusom, pred imenom razširitve. S tem smo razširitev namestili v sistem in jo lahko od sedaj naprej uporabljamo. Na sliki 28 ne vidimo vseh razširitev, ki so na voljo za namestitev, saj jih je preveč, da bi vse uspeli prikazati na eni sliki. Na sliki 28 torej vidimo zgolj nekatere razširitve, ki pripadajo skupini Frontend Plugins.



Frontend Plugins							
	Address Directory	sp_directory	0.2.10		Local	Stable	cms,tt_address
	Address list	tt_address	1.0.3		Local	GL Stable	cms
	Calendar	tt_calender	1.0.4		Global	Beta	cms
	Change MD5 Frontend Password	dr_changefefpasswd_md5	0.3.1		Local	Beta	
	Direct Mail Subscription	direct_mail_subscription	1.0.3		Global	Stable	cms,tt_address
	Front End User Admin	feuser_admin	1.0.2		Global	Stable	cms
	Gallery	lz_gallery	1.0.1		Local	Stable	cms,lz_table
	General Office Displayer	rlmp_officeimport	1.0.4		Local	Stable	cms,libunzipped
	Graphic Visitor Counter	bkt_fe_graphiccounter	0.1.10		Local	Beta	cms
	Guestbook	tt_guest	1.0.8		Global	Stable	cms

Slika 28: Razširitev Extension Manager - način Available extensions to install

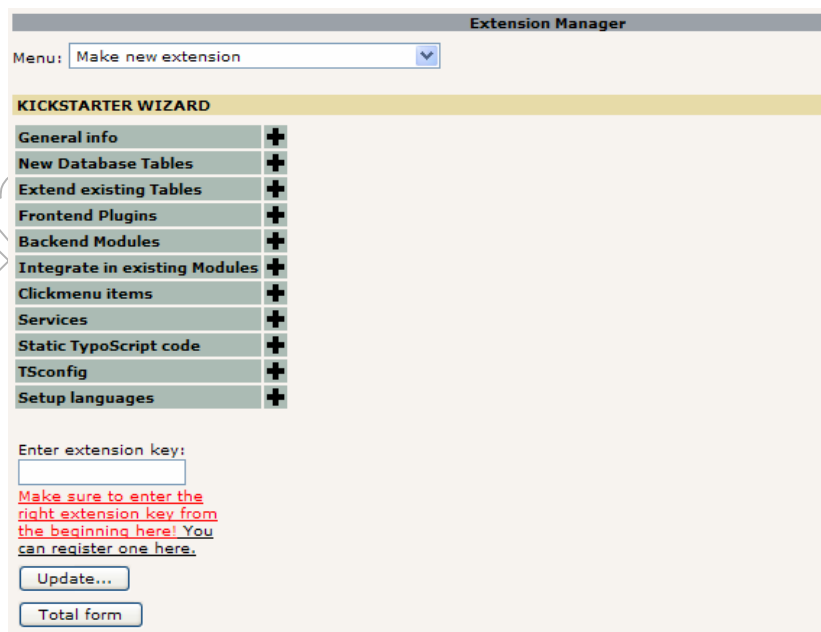
Če si želimo ogledati katere razširitve imamo na voljo za uporabo v sistemu, si v polju Menu izberemo način »Loaded extensions«. V seznamu, ki ga prikazuje Slika 29, vidimo del razširitev, ki so nameščene in na voljo za uporabo.



Extension Manager								
Menu: Loaded extensions		Order by: Category		Show: Details				
Display shy extensions: <input type="checkbox"/>								
LOADED EXTENSIONS								
Look up: <input type="text"/> Search								
	Title:	Extension key:	Version:	DL:	Doc:	Type:	State:	Dependencies:
	Rich Text Editor	rte	0.0.10			Global	Stable	
	Slovenian language pack	csh_si	1.0.0			Local	Alpha	
	Date2Calendar	erotea_date2cal	1.2.0			Local	Beta	cms
	Staff Organisation	jp_staff	0.0.0			Local	Alpha	
	Extension Kickstarter	kickstarter	0.1.0			Local	GL Beta	
	Podatkovna baza EUP	eup	0.0.0			Local	Beta	cms
	Address list	tt_address	1.0.3			Local	GL Stable	cms
	Address Directory	sp_directory	0.2.10			Local	Stable	cms,tt_address
	Simple Calendar	de_simplecalendar	0.2.11			Local	Beta	cms,erotea_date2cal
	Web>Template, CSS Usage Analyzer	tstemplate_cssanalyzer	0.0.2			Local	Alpha	tstemplate
	Urban Nomad Template	tmpl_urbannomad	0.1.5			Local	Beta	
	Typo3 Quixplorer	t3quixplorer	1.6.0			Local	Stable	
	Indexed Search Engine	indexed_search	2.1.3			Local	SL Stable	cms

Slika 29: Razširitev Extension Manager - način Loaded extensions

Če želimo tudi sami ustvarjati nove razširitve, moramo v sistem Typo3 uvoziti in namestiti razširitev Extension Kickstarter. Ko je razširitev Extension Kickstarter nameščena v sistem, imamo v Extension Managerju v polju Menu še en način in sicer »Make new extension«. Ta nam omogoča zagon razširitve Extension Kickstarter (Slika 30).



Slika 30: Razširitev Extension Manager - način Make new extension

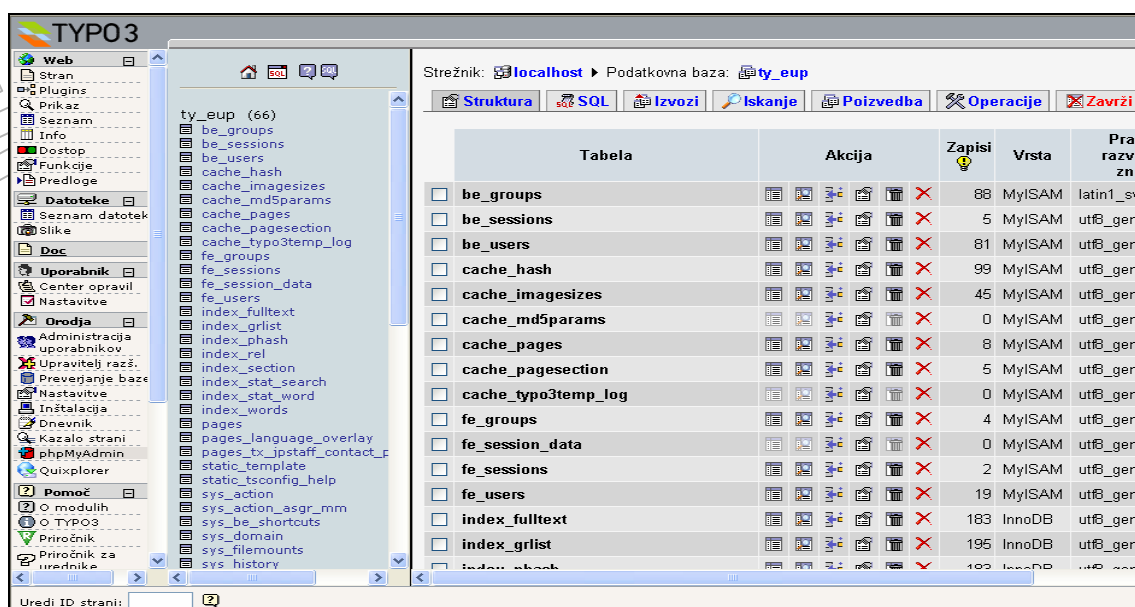
### 3.3.5.2 Extension Kickstarter

Extension Kickstarter je orodje za izdelavo novih razširitev. Kot smo že omenili, ta razširitev ob namestitvi sistema Typo3 ni privzeto nameščena, ampak jo moramo posebej namestiti. Izdelave novih razširitev se lotimo, ko že malo obvladamo Typo3.

Opis posameznih načinov oz. izbir v razširitvi Extension Kickstarter bo opisan v razdelku 4.4.3, kjer je opisano kako smo naredili razširitev Podatkovna baza EUP.

### 3.3.5.3 phpMyAdmin

Izbira phpMyAdmin predstavlja orodje za administracijo strežnika MySQL. Dostop do baze poteka preko Typo3. S pomočjo phpMyAdmin lahko pregledamo vsebino posameznih tabel. Vidimo imena in tipe polj. PhpMyAdmin uporabljamo tudi, ko nas zanimajo določeni podatki iz tabel oz. če želimo opraviti poizvedbo direktno na bazi podatkov.



Slika 31: Izbira phpMyAdmin

PhpMyAdmin uporabimo tudi takrat, ko ugotovimo, da določene tabele ne potrebujemo več. S pomočjo tega orodja jo izbrisemo. Na podlagi phpMyAdmina pa lahko tudi tabele izvozimo v datoteke, v formatih oziroma primerne za uporabo v programih SQL, LaTeX, Microsoft Excel, Microsoft Word, XML, ... Ker nam omogoča popoln nadzor nad podatkovno bazo, zahteva uporaba nekaj previdnosti in znanja. PhpMyAdmin je na voljo le administratorju sistema.

## 4. Projekt EUP

### 4.1 Opis projekta EUP

Pri projektu EUP je bilo potrebno razviti informacijski portal za potrebe informacijske podpore pripravljen na predsedovanje Slovenije Evropski uniji. Portal že deluje in sicer od 1. decembra 2005 dalje.

Intranetni portal v podporo predsedovanju Slovenije Evropski uniji je namenjen vsem sodelujočim v projektu priprav Slovenije na predsedovanje EU v prvi polovici leta 2008. Služi za interno informiranje in komuniciranje ter hranjenje ključnih informacij in vsebin na enem mestu. Uporabljajo ga interni uporabniki, ki so vključeni v projekt predsedovanja na ravni vlade in državne uprave.

Intranetni portal je postavljen kot samostojno delujoče spletno mesto, ki deluje znotraj državne uprave. Je zaprto področje, do katerega imajo uporabniki dostop le z uporabniškimi gesli. Ime portala je Vlada Republike Slovenije – Intranetni portal za podporo Predsedovanju Slovenije EU. Glede na interno naravo informiranja in komuniciranja spletno mesto deluje samo v slovenščini

Za pripravo projekta vzpostavitve intranetnega portala je bila zadolžena Delovna skupina intranet/internet, ki deluje v okviru Podskupine za komunikacijo z javnostmi in promocijo. Člani te delovne skupine so Urad vlade za informiranje, Služba vlade za evropske zadeve in Direktorat za e-upravo in upravne postopke na Ministrstvu za javno upravo. V delovni skupini so sodelovali tudi predstavniki podjetja ISA.IT, ki je bilo izbrano za izdelavo portala. V okviru prakse, ki sem jo opravljal v okviru študija na VSS – Praktična matematika, pri podjetju ISA.IT, sem pri izdelavi portala sodeloval tudi jaz. Na začetku je bila moja naloga predelati koledar dogodkov v portalu, da bo ustrezal željam naročnika. Sčasoma pa sem prevzel še ostale naloge, ki jih je bilo potrebno opraviti v povezavi s portalom EUP. Tako sem iz administracijskega vidika skrbel za uporabnike sistema. Predelava ostalih razširitev sistema je pripadla meni. Med prakso sem kot asistent sodeloval tudi na tečajih za usposabljanje urednikov.

### 4.2 Postopek izdelave projekta

#### 4.2.1 Analiza

Glavna naloga faze analize je priti do podrobnih zahtev glede portala in ugotoviti, kateri akterji in primeri uporabe nastopajo v sistemu.

##### 4.2.1.1 Cilji in usmeritve

Cilje in usmeritve projekta lahko povzamemo v nekaj točkah:

- V celoti je potrebno izdelati portal, ki bo omogočal hranjenje informacij in vsebin na enem mestu
- Čas zaključka je bil 1. december 2005.
- Pomembni sta preglednost, zanesljivost in lahka dostopnost sistema.
- Omogočen naj bo vnos razprav preko spletnih strani.

##### 4.2.1.2 Funkcionalne zahteve oz. zahteve, ki jih je podal naročnik projekta

Radi bi naredili portal, v katerem želimo imeti naslednje:



## a) Uporabniki

Uporabniki spletnih strani se ločijo v štiri skupine: glavni urednik, področni uredniki, bralci in bralci s posebnimi pravicami. Spletno mesto je zasnovano tako, da ima eden ali več glavnih urednikov pravico dodajati ali brisati programske sklope ali dokumente (vsebine na strani). Bralci imajo samo možnost branja dokumentov. Bralci lahko razlikujejo med prebranimi in neprebranimi dokumenti. Bralci s posebnimi pravicami imajo poleg pravic, ki jih imajo bralci, še možnost vnosa razprav na posamezne dokumente.

## b) Jezik

Spletno mesto je izdelano samo v slovenskem jeziku. Naložena vsebina je lahko tudi v tujih jezikih.

## c) Sklopi

Intranetni portal je urejen po sklopih, ki ustrezajo vsebinskim področjem. Vsak sklop ima svojega urednika, odgovorno osebo, ki skrbi za celovitost in ažurnost vsebin v posameznem sklopu. Sklopi so označeni s šifro sklopa (ODS, RAZ, ...), ki je namenjena oblikovanju šifre dokumenta. Vseh sklopov je devet. To so KADRI, PROGRAM, LOGISTIKA, KOMPRO, PRORAČUN, MINISTRSTVA, OŽJA DS, ŠIRŠA DS in RAZNO.

## d) Področja

Vsak sklop je razdeljen na dve področji. Na prvem so objavljene razprave in na drugem dokumenti.

## e) Dokumenti

Dokumenti so temeljni elementi v sistemu. Te bodo uporabniki največkrat iskali in potrebovali. Vsi, ki imajo geslo za dostop do Intranetnega portala v podporo predsedovanju EU, lahko pregledujejo vse objavljene dokumente. Dokument na intranetnem portalu lahko poleg osnovnih podatkov kot so: naslov, avtor, besedilo dokumenta, ... vsebuje še priponko, ki je lahko Wordov dokument, Excelov dokument, dokument v formatu pdf, ....

Vsi dokumenti imajo svojo šifro (prikazuje se jo kot ID dokumenta), ki se vzpostavlja samodejno. Sestavljena je iz »šifre sklopa« (do 5 mest), »šifre zadeve« (do 5 mest) in »sistemске številke dokumenta«.

Primer oblikovanja ID dokumenta:

ŠIFRA SKLOPA-ŠIFRA ZADEVE/ZAPOREDNA ŠTEVILKA DOKUMENTA = 'ŠDS-PS/23'

Dokumenti, ki so objavljeni, morajo imeti ob vnosu izpolnjene sledeče atribute:

- ID dokumenta (avtomatični prikaz, ni potrebno vnašati)
- Naslov dokumenta (potrebno vpisati)
- Datum dokumenta (potrebno vpisati)
- Besedilo dokumenta (potrebno vpisati)
- Avtor dokumenta (potrebno vpisati)
- Tip dokumenta (izbere se iz nabora oz. šifranta)
- Status dokumenta (izbere se iz nabora oz. šifranta)
- Sklop ali podsklop (izbere se iz nabora oz. šifranta)
- Naziv zadeve (izbere se iz nabora oz. šifranta)

K dokumentu lahko priprnemo še eno ali več priponk (poljubnih datotek). Uredniki imajo možnost ustvarjati nove dokumente in popravljati stare.

## f) Zadeve

Znotraj posameznega sklopa so dokumenti razvrščeni po zadevah. Z zadevo označujemo kategorijo nalog, opravi, dokumentov ali dogodkov. Zaradi lažje in boljše preglednosti je centralno uredništvo predvidelo enotna poimenovanja za določene zadeve znotraj posameznih sklopov. Ta so na primer: Centralne naloge, Tuje izkušnje, Projektna naloga, Projektna skupina, Sestanki vodij podskupin, itd.



Nove kategorije zadev lahko dodaja samo glavni urednik. Vsaka zadeva vsebuje enega ali več dokumentov.

Glavni urednik pri kreiranju "Zadeve" vpisuje:

- šifro zadeve (potrebno vpisati),
- naziv zadeve (potrebno vpisati),
- sklop ali podsklop (izbere se iz nabora oz. šifranta).

g) Razprave

Intranetni portal omogoča tudi neposredno odzivanje sodelujočih v projektu na dokumente, ki bodo na intranetni portal uvrščeni kot delovni dokumenti. Razprave lahko vnašajo uredniki in bralci s posebnimi pravicami. Ostali bralci nimajo možnosti vnosa razprav, lahko pa jih pregledujejo.

h) Iskanje

Na spletnem mestu je vključenih več iskalnikov. Za vsak sklop imamo na straneh iskalnik, ki omogoča iskanje po zadevah in po dokumentih. Na posebni strani se nahaja iskalnik, ki omogoča iskanje po celotnem portalu. Deli se na splošno iskanje in razširjeno iskanje. Pri splošnem iskanju to poteka po vsebini vseh objavljenih dokumentov in po naslovih priponk. Pri razširjenem iskanju pa se pri iskanju lahko omejimo na posamezen sklop.

i) Šifranti

Obstaja šifrant Tipi dokumentov, iz katerega vsakemu dokumentu izberemo njegov tip. Ta je enolično določen. Šifrant uredniki lahko poljubno dopolnjujejo, ne morejo pa brisati že vnesenih šifer.

#### 4.2.1.3 Identifikacija in opis uporabnikov

Uporabniki so opisani z vidika vloge, ki jo imajo v informacijski domeni. Osredotočimo se na opis pomena vloge za sistem. Uporabniki v sistemu so:

- BRALEC – vsebine samo pregleduje.
- BRALEC S POSEBNIMI PRAVICAMI – vsebine pregleduje in ima možnost omejenega odzivanja na določen dokument (vnos razprav).
- UREDNIK – ustvarja in spreminja dokumente v svojem sklopu, kreira tudi uporabniška imena in gesla za bralce in bralce s posebnimi pravicami, vnaša dogodke v koledar dogodkov
- GLAVNI UREDNIK – ustvarja in spreminja dokumente znotraj vseh sklopov, vnaša nove zadeve, vnaša kontaktne osebe, vnaša dogodke v koledar dogodkov
- ADMINISTRATOR – vzdržuje uporabnike v sistemu in ustvarja nove, vzdržuje šifrante, ki se uporabljajo v sistemu

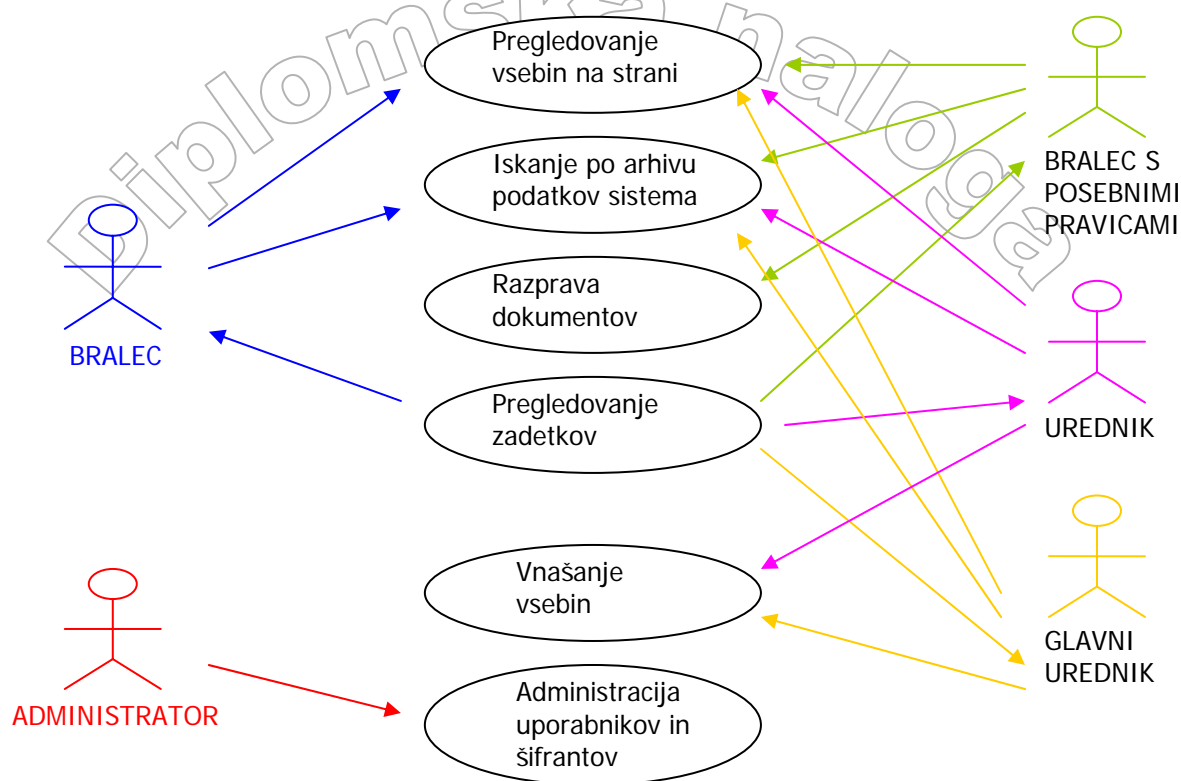
#### 4.2.1.4 Identifikacija in opis primerov uporabe

V tem razdelku si oglejmo, katere skupine uporabnikov bodo izvajale določene najbolj pogoste primere uporabe portala:

- pregledovanje vsebin na strani – to izvajajo prav vse v prejšnjem razdelku našete skupine uporabnikov, z izjemo administratorja: bralec, bralec s posebnimi pravicami, urednik in glavni urednik
- iskanje po arhivu podatkov sistema – tudi to opravilo izvajajo iste skupine, kot prejšnje opravilo
- pregledovanje zadetkov – ustrezne skupine so bralec, bralec s posebnimi pravicami, urednik in glavni urednik
- razprava dokumentov – to izvaja le bralec s posebnimi pravicami
- vnašanje vsebin – to izvajata urednik in glavni urednik
- administracija uporabnikov in šifrantov – to izvaja administrator

#### 4.2.1.5 Diagram primerov uporabe

Diagram na sliki 33 predstavlja komunikacijo med vrstami uporabnikov in sistemom.



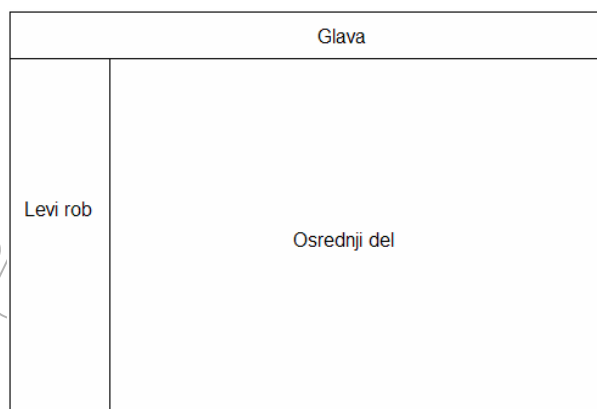
Slika 32: Diagram primerov uporabe za projekt EUP

## 4.2.2 Načrtovanje

Fazi analize sledi faza načrtovanja, kjer na podlagi specifikacije zahtev izdelamo načrt projekta.

### 4.2.2.1 Postavitev spletnega mesta

Pri oblikovanju koncepta izgleda spletnega mesta je ključno vlogo igrala enostavnost in preglednost portala. Zato je bila odločitev, da je predloga sestavljena iz treh delov: glave, levega roba in osrednjega dela. V »Glavi« je logotip organizacije, vodoravni meni in splošni podatki (tekoči datum). V »Levem robu« je glavni meni s povezavami do vseh devetih vsebinskih sklopov in še z ostalimi povezavami kot so: koledar, iskanje, administrativni dostop, obvestila za uporabnike, kontaktne osebe. Po izbiri iz levega menija se po potrebi odpre podmeni. Vsebina se odpira in prikazuje v »Osrednjem delu«.



Slika 33: Postavitev spletnega mesta

#### 4.2.2.2 Tabele v podatkovni bazi

Za izvedbo projekta smo uporabili relacijsko podatkovno bazo MySQL. Vsa interakcija z bazo poteka s pomočjo jezika SQL preko sistema Typo3. Ker je tabel, ki sestavljajo bazo veliko, bomo opisali samo najpomembnejše oz. tabele, ki jih potrebujemo za najpomembnejšo razširitev. Tabele, ki jih uporabljamo za razširitev Podatkovna baza EUP, bodo opisane pri opisu razširitve.

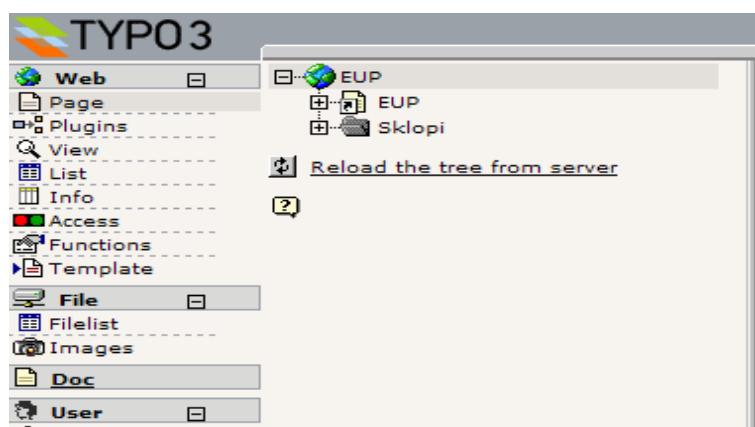
#### 4.2.3 Uporabljeni orodja za izdelavo

Celotno spletno mesto temelji na odprti kodi. Osrednje orodje, ki se ga je uporabljalo za izdelavo spletnega mesta, je TYPO3, verzija 3.8.0. Uporabljeni so bili še spletni strežnik Apache (ver. 2.0.53), skriptni jezik PHP (ver. 4.3.4 s podporo MySQL) in podatkovna zbirka MySQL (ver. 4.0.16). Za urejevanje in spreminjanje kode skriptnega jezika PHP se je uporabil program Notepad++.

### 4.3 Izdelava projekta EUP

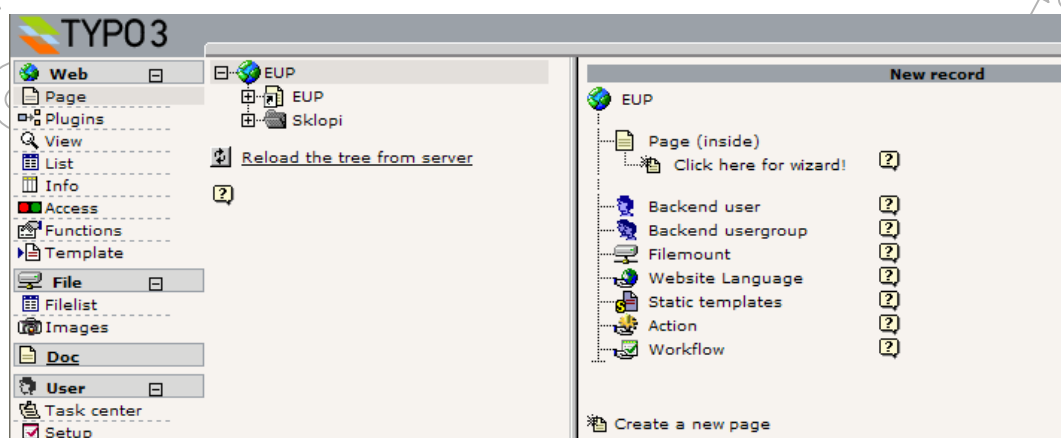
Prvo opravilo, ki ga naredimo, je namestitev sistema Typo3 na strežnik. Namestimo paket, v katerem je sam sistem Typo3. Po zagonu programa Typo3 najprej konfiguriramo podatkovno bazo, ki jo bomo uporabili za projekt. Podatkovni bazi damo ime EUP.

Najprej želimo narediti navadno stran z imenom EUP in sistemsko mapo z imenom Sklopi kot prikazuje slika 34.



Slika 34: Strani EUP in Sklopi


Z levim gumbom miške kliknemo na ikono v obliki globusa z imenom EUP. Iz kontekstnega menija izberemo New. V podrobnem prikazu se odprejo možnosti, kot jih kaže slika 35.



Slika 35: Ustvarjanje novih strani

Izberemo možnost »Click here for wizard!«. V podrobnem prikazu označimo, kje v drevesni strukturi strani želimo, da se nahaja naša stran. Ker je to prva stran, ki jo ustvarjamo, nimamo veliko izbire, saj je trenutno to še edina stran. Ko pozicijo strani izberemo, se odpre forma za konfiguracijo strani.

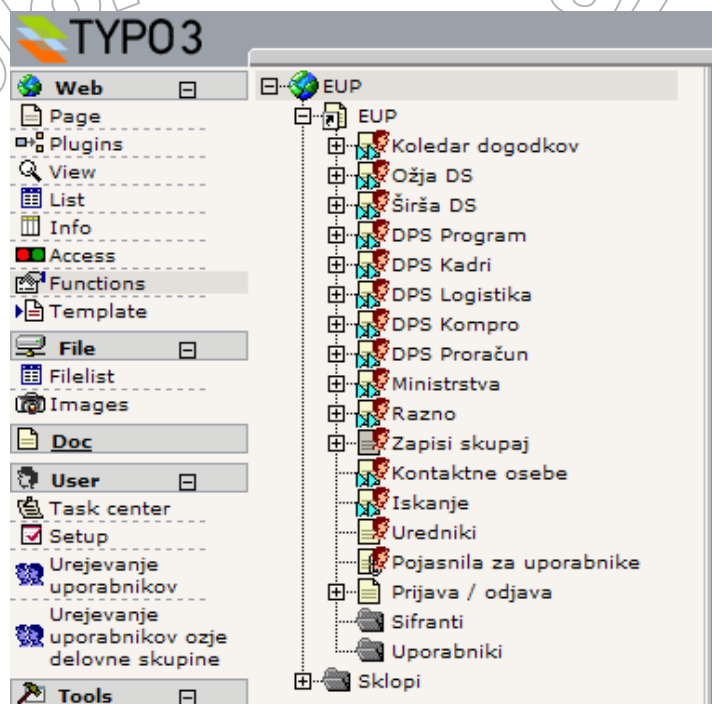
Slika 36: Forma za vnos podatkov za stran

Najbolj zgoraj je potrditveno polje, kjer imamo možnost skriti stran v čelnem sistemu (»Hide page:«). Mi jo ne bomo skrili in zato pustimo kvadrata prazen. V izbirnem seznamu za »Type« pustimo izbrano možnost Standard. Pri projektu bomo za tipe strani uporabljali samo možnosti Standard, SysFolder in Not in menu. Postavimo se v polje »Page title:« in vpišemo naslov strani. V našem primeru je to EUP. Ostala polja pustimo prazna. Da bomo zadevo shranili, izmed treh gumbov, ki so levo spodaj in zgoraj na sliki 36, kliknemo sredinski gumb (  ). Prvi gumb shrani nastavitve strani in forme ne zapre, drugi gumb shrani nastavitve strani in zapre formo, tretji gumb pa formo zapre brez shranitve nastavitve strani. Za sistemsko mapo ponovimo enak postopek kot smo ga izvedli za stran EUP, le pri tipu strani izberemo tip SysFolder in za ime mape napišemo Sklopi.

Na strani EUP določimo kakšno predlogo strani hočemo. Kot smo že omenili, smo se odločili za predlogo sestavljeno iz treh delov: glava, levi del in osrednji del. Podobno strukturo sicer ima predloga BUSINESS, a hoteli smo malo drugačno. Zato smo na strani <http://typo3.org> našli predlogo, ki povsem ustreza našim potrebam. Ime predloge je Urban Nomad Template. To predlogo smo uvozili v sistem in dodali na stran EUP.

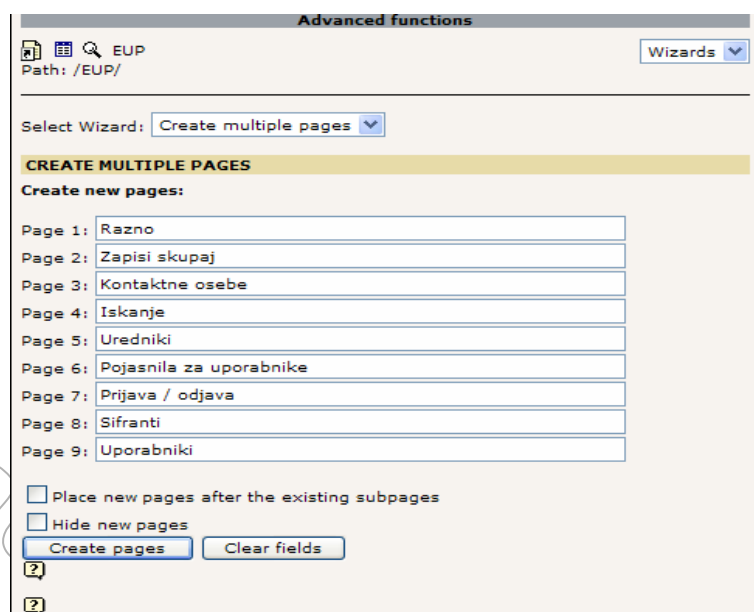
Nato želimo ustvariti še strani za posamezne sklope, stran s koledarjem dogodkov, stran s podatki o kontaktnih osebah, stran, kjer bo dostopen iskalnik, stran, namenjeno urednikom, stran s pojasnili za uporabnike in stran prijava/odjava. Želimo torej narediti, kar prikazuje slika 37. Na sliki 37 vidimo, da

so na sliki pri določenih straneh poleg samih imen strani še plusi (+), ki običajno označujejo, da se pod stranmi skrivajo še podstrani, nekakšne glave na ikonah strani in modri trikotniki. Zamislimo si, da na zgornji sliki teh dodatkov ni. Podrobneje bomo te oznake razložili v nadaljevanju.



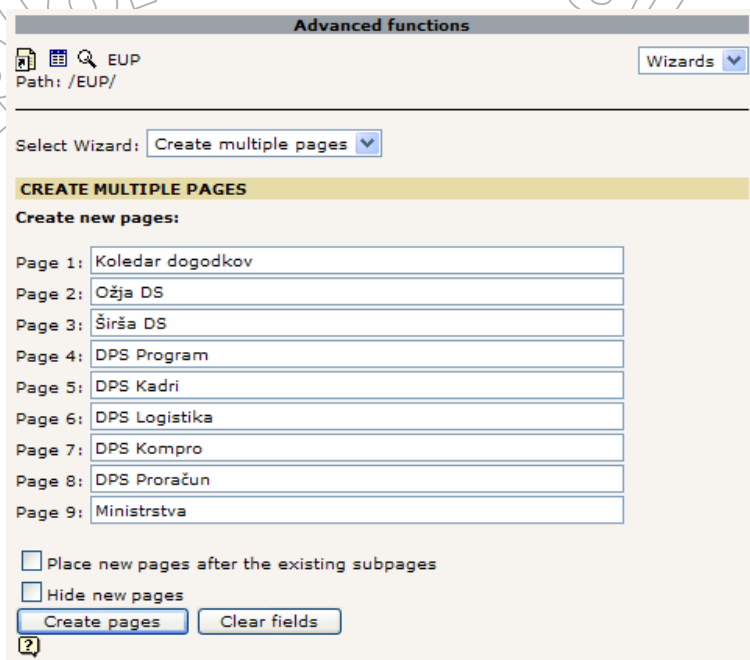
Slika 37: Strani sklopov in sistemskih map

Kot vemo, imamo na voljo izbiro, ki nam pomaga pri ustvarjanju več strani naenkrat. Ta izbira je izbira Functions. Kliknemo nanjo in nato še na stran EUP. V izbiri Functions poskrbimo, da je v izbirnem seznamu desno zgoraj izbran način "Wizards". Kot ustreznega čarovnika si izberemo "Create mutiple pages". V polja za imena strani vpišemo najprej spodnjih devet strani kot kaže Slika 38 in kliknemo gumb »Create pages«.



Slika 38: Uporaba izbire Functions

Zgornji postopek ponovimo še za preostalih devet strani. Ponovno kliknemo na izbiro "Functions", nato na stran EUP. Poskrbimo za čarovnika "Create multiple pages". Vpišemo preostalih devet strani (Slika 39) in pritisnemo gumb "Create pages".





Slika 39: Uporaba izbire Functions

Iz slike 37 je razvidno, da je ikona strani "Zapisi skupaj" obarvana sivo. To pomeni, da je tip strani "Not in menu" oz. da stran ni v seznamu spletnih strani. Da smo za stran "Zapisi skupaj" dosegli, da ni v seznamu strani, smo v drevesu strani z levim miškinim gumbom kliknili na njeno ikono in iz kontekstnega menija izbrali "Edit page header". Odpre se forma z lastnostmi za to stran. V polju "Type" (tip strani) spremenimo "Standard" v "Not in menu". Za strani "Sifranti" in "Uporabniki" želimo, da sta to imenika in ne običajni strani. Postopamo, kot smo opisali pri strani "Zapisi skupaj", le da za tip strani izberemo "SysFolder".

Vsaka stran posameznega sklopa ima še svojo podstran z imenom Vsi dokumenti sklopa. Stran Prijava/odjava ima tudi svojo podstran z imenom Sprememba gesla. Stran Koledar dogodkov ima tri podstrani in sicer Arhiv dogodkov, En dogodek in En dan. Arhiv dogodkov ima štiriindvajset podstrani in sicer Januar 2005, Februar 2005, ..., Januar 2006, ..., December 2006. Vsak izmed sklopov ima svojo sistemsko mapo za shranjevanje.

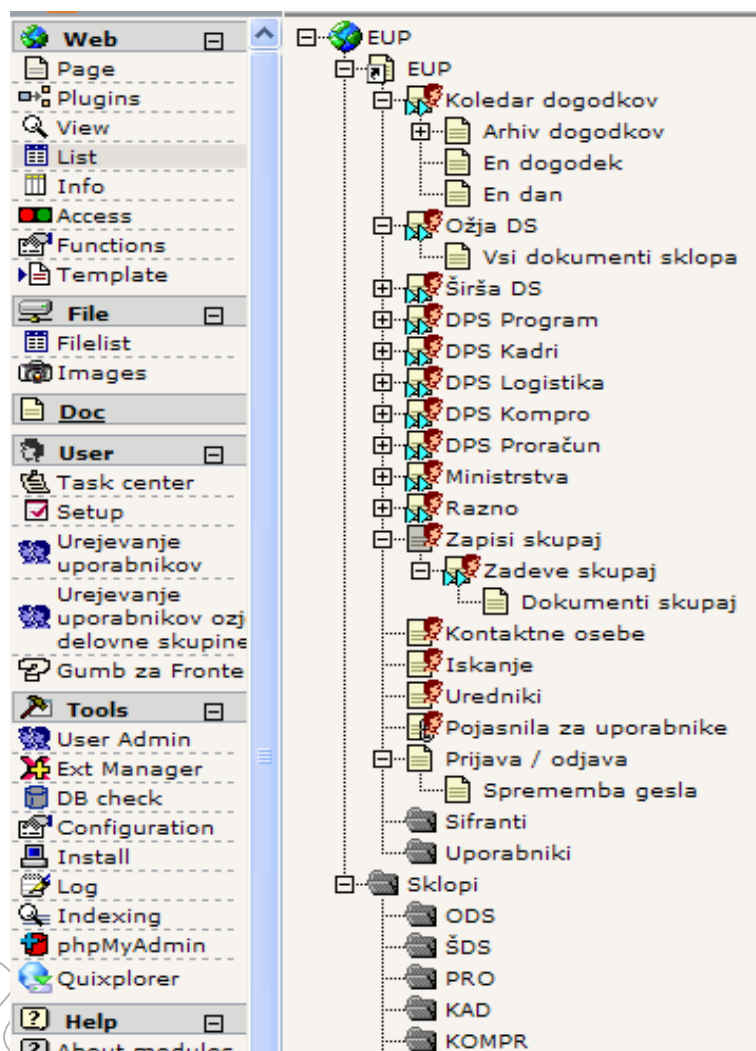
Vsako podstran posameznega sklopa moramo narediti posebej. Npr. za stran Ožja DS naredimo podstran Vsi dokumenti sklopa na sledeč način: z levim gumbom miške kliknemo na ikono strani Ožja DS. Iz kontekstnega menija izberemo New. Izberemo možnost »Click here for wizard!«. V podrobnem prikazu označimo, kje v strukturi drevesa strani želimo, da je naša stran. Ker naj bi bila stran »Vsi dokumenti sklopa« podstran strani Ožja DS, to označimo v strukturi drevesa strani. Na enak način tudi ostalim stranem naredimo podstrani »Vsi dokumenti sklopa«. Žal v tem primeru nismo mogli uporabiti izbire za ustvarjanje več strani naenkrat. Izbira Functions lahko res naredi večje število strani naenkrat, vendar so vse te nove strani podstrani iste strani. Izbira Functions pa bi lahko uporabili na strani Koledar dogodkov, saj tam potrebujemo tri podstrani. Ko posamezni strani naredimo podstrani, se pred ikono nadstrani pojavi plus(+) ali minus(-). + pomeni da so podstrani skrite v drevesni strukturi, - pa, da so odprte. Zadeva deluje enako kot Raziskovalec v operacijskem sistemu Windows.

Na strani EUP (na osnovni, vhodni strani) naredimo preusmeritev na stran Prijava/odjava. Torej nas vstop v portal avtomatsko preusmeri na stran Prijava/odjava. Da gre pri strani za preusmeritev, prikazuje ikona v obliki majhnega kvadratka s puščico. Preusmeritev naredimo tako, da na stan EUP

kliknemo z levim gumbom miške in iz kontekstnega menija izberemo Edit. Nato kliknemo na gumb "Edit page properties". V polju Type spremenimo iz Standard v Shortcut. V polju »Shortcut to page« s pomočjo gumba  v drevesni strukturi poiščemo stran Prijava/odjava in kliknemo nanjo. Nato kliknemo še gumb  in preusmeritev je narejena.

Potrebno je tudi, da strani naredimo dostopne samo določenim uporabnikom. Na vsaki izmed glavnih strani (Koledar dogodkov, Ožja DS, Širša DS, ..., Pojasnila za uporabnike) kliknemo z levim gumbom miške in iz kontekstnega menija izberemo »Visibility settings«. Odpre se nova forma z nastavitvami. V polju Access izberemo iz seznama skupino Bralci. V polju »Include subpages« pa označimo kvadrček s kljukico. S tem, ko smo v polju Access izbrali neko skupino, se je na straneh pojavila glava, kar pomeni, da dostopajo do te strani le določene osebe. S tem, ko smo označili s kljukico kvadrček v polju »Include subpages« pa smo dosegli, da velja enako omejen dostop tudi za podstrani. Pri tem se pojavita na ikoni strani dva trikotnika modre barve. Tako so zdaj vse strani in podstrani, razen strani Prijava/odjava in Sprememba gesla, dostopne le uporabniški skupini Bralci.

Na koncu želimo, da je struktura strani v drevesu naslednja:



Slika 40: Končna drevesna struktura strani



## 4.4 Opis in izvedba nekaterih funkcionalnosti portala

V tem razdelku bomo opisali, kako izvedemo določene funkcionalnosti portala. Ni namreč smiselno, da bi opisovali vse podrobnosti pri razvoju. Zato si bomo ogledali le način, kako razvijemo zanimivejše sestavine in zmožnosti.

### 4.4.1 Prijava v portal

#### OPIS

V portal EUP ne more vstopiti vsak. Ker so podatki v portalu zaupne narave, je dostop omogočen le določenim osebam. Prijava v aplikacijo je mogoča samo preko uporabniškega imena in gesla. Poleg uporabe uporabniškega imena in gesla je potrebno, da ima uporabnik dostop do omrežja H-KOM. To je zaprto omrežje, namenjeno zaposlenim znotraj ministrstev in vlade. Že za dostop do tega omrežja je potrebna prijava s posebno kartico, na kateri se vsako minuto spreminja geslo. Do omrežja namreč lahko dostopamo le preko določenih računalnikov, ki imajo nameščen ustrezen program za prijavo v omrežje H-KOM. S tem, ko se uporabnik prijavi v omrežje H-KOM, se točno ve, kdo je oseba, prijavljena v omrežju, saj je kartica, ki jo dobimo za prijavo v omrežje, izdana za točno določeno osebo. Tako se pri vsaki prijavi ve, kdo je vstopil v omrežje in kdaj. Ker pa vsi uporabniki omrežja H-KOM niso pooblaščen, da bi uporabljali portal EUP, je potrebna še prijava v sam portal. Ob prijavi v portal mora uporabnik vnesti še pravilno uporabniško ime in geslo. Ob uspešnem vnosu je uporabnik prijavljen v portal in vstopi v čelni sistem. Zato je bilo potrebno portal opremiti z ustreznim mehanizmom za prijavo.

#### IZVEDBA



Sistem Typo3 že ob namestitvi vsebuje razširitev v obliki prijavnega obrazca za uporabnike čelnega sistema. Med samim testiranjem obrazca za prijavo smo ugotovili, da nam ne nudi tistega, kar smo hoteli. Zato smo se odpravili na stran <http://typo3.org>, kjer smo iskali ustrezen nadomestek za prijavni obrazec. Na strani smo poiskali zavihek Extensions in v iskalno polje vpisali "login". Med vsemi zadetki smo izbrali »New front end login box«. Testiranje je pokazalo, da je bila odločitev za to razširitev prava.

Najprej smo komponento »New front end login box« uvozili v sistem in jo namestili. Na spletni strani Prijava/odjava smo dodali vtičnik prijavnega obrazca. Vtičnik nam omogoča prikaz obrazca za vnos uporabniškega imena in gesla. Vtičnik dodamo tako, da na stran Prijava/odjava kliknemo z levo tipko miške. Iz kontekstnega menija izberemo New. V podrobnem prikazu se nam prikažejo razne možnosti. Izberemo »Click here for wizard« (Pagecontent → Click here for wizard). V podrobnem prikazu se odprejo nove možnosti. Pod kategorijo Plugins izberemo Better login-box. Ob prejšnjem kliku se je odprl tudi seznam vsebin na strani, kjer moramo nato še določiti položaj vtičnika. Ker je to prva vsebina na strani Prijava/odjava, nam ni potrebno preveč razmišljati o položaju. Kliknemo v območje seznama vsebin in vtičnik je dodan na stran.

Kot smo povedali že v začetku diplomske naloge, ima vsak zapis v Typo3 svoj id. Tako imajo tudi vsi uporabniki čelnega in zalednega sistema svoj id. Vse strani Typo3 imajo svoj id in so shranjene v bazi podatkov v tabeli pages. Ker je sistemska mapa ena od vrst spletnih strani, je tudi ona zapisana v bazi. Typo3 dinamično izbira te zapise iz tabele PAGES in strukturira drevesno strukturo strani. V tabeli FE\_USERS imamo polje PID in ta nam pove, na katero sistemsko mapo ustreza uporabnik. Torej nastavitve uporabnikov so zapisane v bazi, pregledujemo in spreminjamo pa jih lahko v sistemski mapi Uporabniki.

S tem, ko smo na prijavno stran namestili vtičnik za prijavo, smo že poskrbeli za celoten mehanizem prijavljanja. Vsa potrebna preverjanja, reakcije ob napačnem geslu in podobno so že vgrajena v sami logiki komponente in nam je preostalo le še v datoteki s prevodi (locallang.php) napisati prevode iz angleščine v slovenščino. V datoteki locallang.php je bilo potrebno narediti prevode za slovenščino. To je bilo precej enostavno delo saj je bilo potrebno le napisati pare »originalno ime polja« : »prevod«.

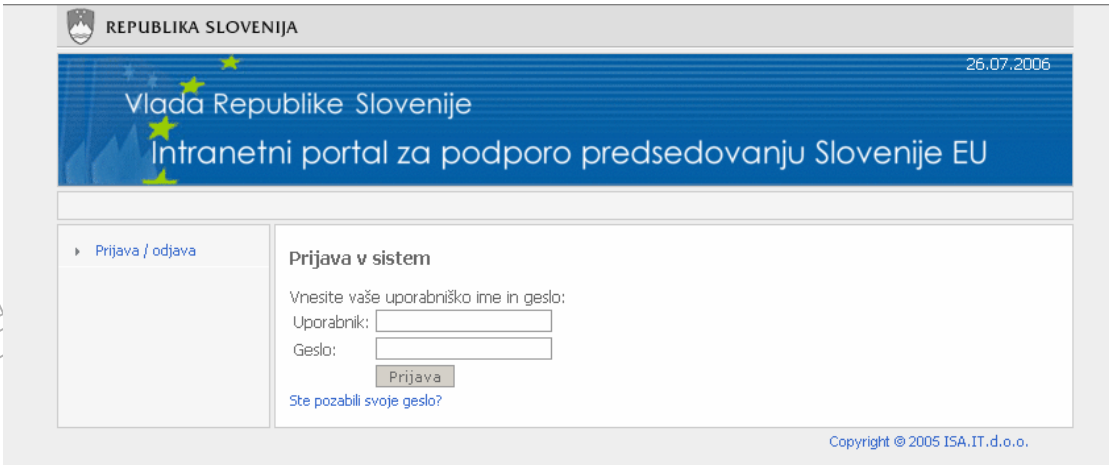


Torej je vsa naša naloga pri zagotavljanju prijave v sistem bila le namestitev ustrezne razširitve, zapis prevodov v slovenščini, vključitev vtičnika na stran in določitev, da se kot mapa za hranjenje podatkov te spletne strani uporabi mapa, kjer hranimo podatke o uporabnikih sistema. Mapo, kjer hranimo podatke o uporabnikih sistema, smo določili tako, da smo v nastavitvah strani Prijava/odjava v polje »General Record Storage page:« dodali sistemsko mapo Uporabniki. Do nastavitve strani pridemo tako, da kliknemo na izbiro Page. Nato kliknemo na stran Prijava/odjava. V podrobnem prikazu kliknemo na gumb »Edit page properties«. Odpre se obrazec z nastavitvami za stran. V polju »General Record Storage page:« kliknemo na gumb  in v drevesni strukturi izberemo sistemsko mapo Uporabniki. Na koncu kliknemo še na gumb , ki pomeni »Shrani in zapri«.

Hoteli smo, da gesla niso shranjena v navadni tekstovni obliki, ampak zakodirana s kodo md5. S tem smo hoteli onemogočiti, da bi administrator portala oz. nekdo, ki bi imel vpogled v bazo podatkov, lahko videl kakšna gesla imajo uporabniki čelnega sistema. Zato smo poizkusili najti še ustrezno razširitev, ki bi nam to omogočala. Tako smo v polje za iskanje razširitev vpisali "md5" in dobili seznam razširitev v povezavi z md5. Ustrezna razširitev, ki nam je ustrezala, se imenuje MD5 FE Password. Ta nam omogoča shranitev gesel tako, da jih pred shranitvijo zakodira v md5 kodo in nato shrani. S pomočjo kode md5 so gesla uporabnikov čelnega sistema dobro zakodirana in varna pred zlorabo. Dodatnih nastavitvev pri razširitvi MD5 FE Password ni potrebno konfigurirati.

Za zaledni sistem je postopek prijave že avtomatično na voljo s samo namestitvijo sistema na strežnik. Zato ni potrebno v sistem namestiti nobene dodatne razširitve. Geslo za zaledni sistem se že samo shranjuje v kodi md5.

V obeh primerih prijav (v čelni ali zaledni sistem) se zakodirano geslo in vneseno uporabniško ime s pomočjo poizvedbe, ki jo izvede razširitev »New front end login box« (za čelni sistem) ali sam sistem Typo3 (za zaledni sistem), preveri v bazi podatkov. Če uporabnik obstaja in je geslo pravilno, je s tem prijavljen v portal. Glede na njegove pravice se odpre ustrezna struktura map, strani ter izbir v zalednem sistemu oz. strani v čelnem sistemu.



Slika 41: Prijavni obrazec v portal za čelni sistem



Slika 42: Prijavni obrazec v portal za zaledni sistem

#### 4.4.2 Priprava koledarja dogodkov

##### OPIS

V portalu želimo imeti tudi koledar dogodkov. Ta naj bi omogočal, da uporabnik na hiter in pregleden način vidi dogodke, ki se bodo zgodili v določenem tednu, hkrati pa služi tudi za kronološki pregled dogajanja na portalu. Slika 43 prikazuje, kako je videti mesečni in dnevni koledar. Mesečni koledar je prikazan na sliki 43 zgoraj. Pod njim je dnevni koledar za več dni. V mesečnem koledarju je tekoči dan obrobljen z rdečim kvadratom in ima rdečo številko dneva. Za vsak dan v mesecu, ki vsebuje dogodek, se številka na koledarju odebeli in je modre barve (na primer dan 28 na sliki 43). Prazniki se na mesečnem koledarju obarvajo z vijoličastim kvadratom. Ob kliku na dan v koledarju, ki vsebuje dogodek, se v dnevnem koledarju prikažejo dogodki za ta izbrani dan in še za šest dni naprej. Dnevni koledar prikazuje dogodke za trenutni ali izbrani dan in še za šest dni naprej. V splošnem pogledu prikazuje datum in uro dogodka, naslov dogodka in njegov kratek opis ter povezavo za ogled podrobnega opisa dogodka (podrobno>). Ob kliku na naslov dogodka oz. na povezavo podrobno> se odpre podroben opis dogodka. Arhivski koledar deluje po istem principu kot dnevni koledar, le da imamo pri arhivskem koledarju na strani izpisane vse dogodke za določen mesec.

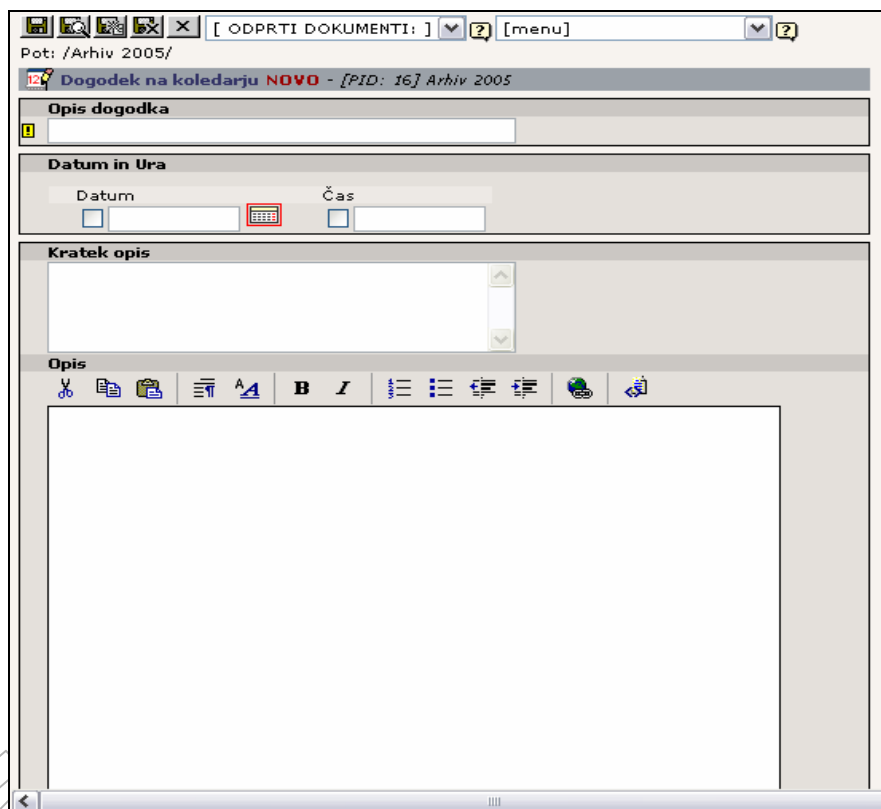
Dogodke vnašamo v zalednem sistemu. Premaknemo se na stran Arhiv dogodkov (Slika 46). Pod kategorijo Dogodek na koledarju kliknemo na ikono . Odpre se obrazec za vnos podatkov o dogodku (Slika 45). Ko polja napolnimo s podatki, kliknemo še gumb . Obrazec se zapre in dogodek je shranjen.



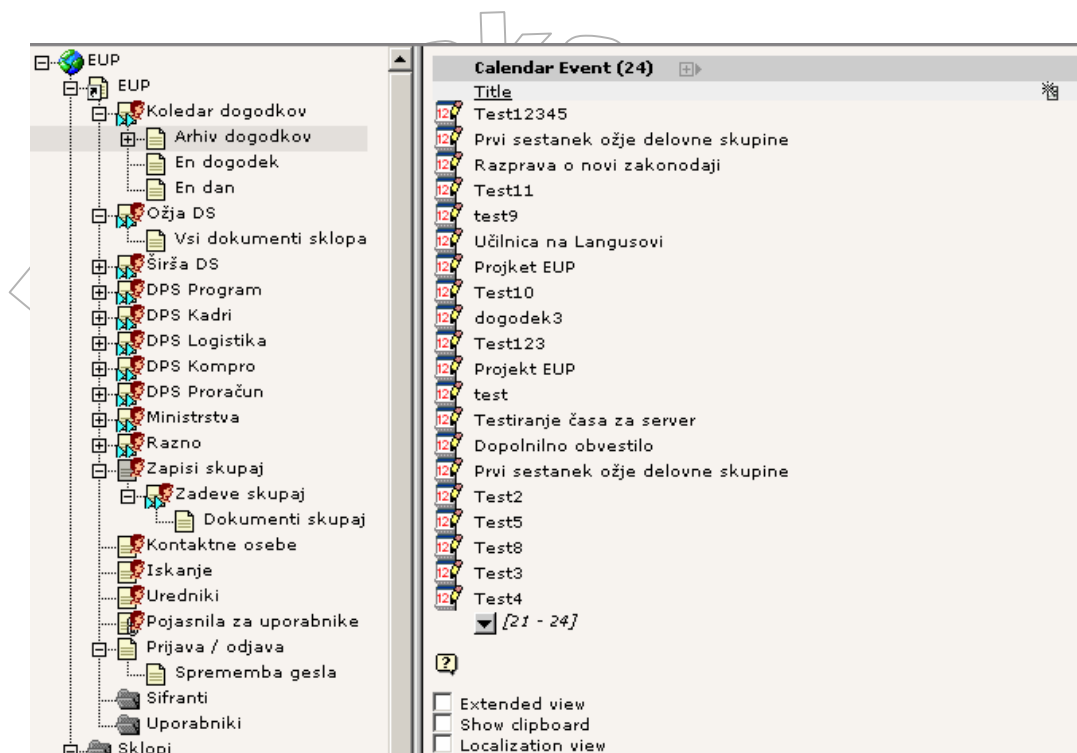
Slika 43: Koledar dogodkov - mesečni in dnevni + 6 dni pogled



Slika 44: Koledar dogodkov - arhiv pogled



Slika 45: Koledar dogodkov - vnos dogodka v zalednem sistemu



Slika 46: Seznam dogodkov na strani Arhiv dogodkov

## IZVEDBA

Sam sistem Typo3 ne vsebuje razširitev, ki bi delovala kot koledar dogodkov. Omenili smo že, da je za sistem na voljo cel kup razširitev. Ker je koledar zadeva, ki jo pogosto uporabljamo na portalih, smo predpostavljali, da zagotovo obstaja ustrezna razširitev, ki nam da na voljo tak vtičnik. Zato smo spet pogledali na osnovno stran sistema, <http://typo3.org>. Na zavihku Extensions smo v polje za iskanje vpisali »calendar« in kliknili gumb Search. Dobili smo seznam razširitev v obliki koledarjev. Za vsako rešitev smo v opisu (Description) prebrali kakšen je koledar in kaj ta koledar nudi. Med razširitvami, ki so bile na voljo, je bila našim potrebam najbližja razširitev Simple Calendar. Zato smo jo z Ext Manager uvozili in namestili v sistem.

Na strani Arhiv dogodkov smo hoteli, da bi se nahajala mesečni in dnevni + šest dni koledarja (Slika 43). Opišimo postopek, kako to dosežemo. Na stran Arhiv dogodkov dodamo vtičnik koledarja. Z levo tipko miške kliknemo na to stran in iz kontekstnega menija izberemo New. V podrobnem prikazu se prikažejo razne možnosti. Izberemo »Click here for wizard« (Pagecontent → Click here for wizard). V podrobnem prikazu v kategoriji Plugins izberemo Simple Calendar in vtičnik dodamo na vrh seznama vsebin. Nato opravimo nastavitve za razširitev. V polje »Header:« vpišemo »V naslednjih sedmih dneh«. V polje »Code:« vpišemo »upcoming:fixed« in v polju »Startingpoint« dodamo stran »Arhiv dogodkov«. Z nastavitvijo v polju Code smo povedali, katerega od metod za prikaz koledarjev želimo uporabiti na strani kjer je vtičnik koledarja. V primeru »upcoming:fixed« uporabimo metodo, ki na strani prikaže kombinacijo mesečnega koledarja in dnevnega koledarja + šest dni. Z nastavitvijo v polju Startingpoint smo povedali, na kateri strani bomo hranili dogodke za koledar. Ker bomo hranili dogodke na strani Arhiv dogodkov je bilo potrebno to stran dodati v nastavitvah.

Na isti način koledar dodamo še na straneh »En dogodek« in »En dan«. Na strani »En dan« moramo v nastavitvah razširitve v polje »Header:« vpisati »Pregled dogodkov za en dan«. V polje »Code:« vpišemo »list« in v polju »Startingpoint« dodamo stran »Arhiv dogodkov«. Na strani »En dogodek« pa v polje »Header:« vpišemo »Izbrani dogodek«, v polje »Code:« pa »single«. V polju »Startingpoint« dodamo stran »Arhiv dogodkov«. Klic metode list v polju Code tokrat pomeni, da želimo, da se na strani prikaže dnevni koledar + šest dni naprej, medtem, ko single omogoča prikaz enega dogodka, ki ga izberemo iz dnevnega koledarja.

Da je koledar dogodkov ustrezal željam naročnika, je bilo potrebno popraviti kodo in opraviti še določene druge spremembe v razširitvi. Najprej se je bilo potrebno spoznati s strukturo razširitve, saj je bila to prva razširitev, ki smo jo bolj podrobno predelovali. Po pregledu vseh datotek smo prišli do ugotovitve, da je v našem primeru potrebno predelati datoteki `tx_desimplecalendar_pi1.php` in `locallang.php`. V datoteki `locallang.php` je bilo potrebno narediti prevode za slovenščino.

Kompleksnejše je bilo popravljanje kode v datoteki `tx_desimplecalendar_pi1.php`. Tukaj smo spremenili kodo metode `listView()`, ki ima nalogo prikazovanja dogodkov za določen dan. Popraviti jo je bilo potrebno tako, da se poleg podatkov za en dan prikažejo še podatki za preostalih šest dni naprej.

Metoda `listView` za prikaz dogodkov kliče metodo `de_getRecords`. Ta metoda nam vrne zapise o dogodkih, ki so se zgodili v določenem obdobju. Del kode je prikazan spodaj

```
function de_getRecords($command,$count=0,$day='', $month,$year,
                      $onlyUpcoming='') {

..... → manjkajoca koda

case 'list':

if($this->selectParams['daterange']) {
    //we parse the daterange, which has to comply with this format:
    //1-1-2004,31-1-2004, or special keyword. It takes precedence
    //over upcoming config daterange with special keywords
    if(ereg('[a-z]', substr($this->selectParams['daterange'],0,1))
        switch ($this->selectParams['daterange']) {

                case 'day': //calendar month given day
                    $bottom = mktime(0,0,0,$month,$day,$year);
                    $top = mktime(23,59,59,$month,$day,$year);
                    break;
                case 'day6': //tekoči dan in še 6 dni vnaprej
                    $bottom = mktime(0,0,0,$month,$day,$year);
                    $top = mktime(23,59,59,$month,$day+6,$year);
                    break;

..... → manjkajoca koda
}
```

Tisti del kode, ki smo ga dopisali, je označen polkrepko. Originalno kodo smo pustili v angleščini, komentarje pri naši dodani kodi pa smo namerno pisali v slovenščini, da smo s tem še bolj poudarili spremembe, ki smo jih opravili. Zato je koda dvojezična.

V tem delu kode izračunamo začetni in končni čas obdobja, za katerega nas zanimajo dogodki. Originalna koda je v primeru, da je bil določen parameter nastavljen na "day", začetni čas nastavila na polnoč tekočega dne in kot končni čas določila čas 23h 59min 59sec tega istega dne. Po tem vzorcu smo dopolnili kodo tako, da smo predvideli novo možnost parametra. Če je ustrezen parameter namesto na "day" nastavljen na "day6", je ta interval od trenutnega dneva do šest dni naprej. Torej začetni dan je trenutni dan ob polnoči in končni datum pa šest dni naprej do 23h 59min 59sec.

Del kode metode `listView`:

```
/* Metoda, ki nam prikaze dogodke za trenuten dan + sest dni naprej
   oz. dogodke za dan,
   ki ga izberemo iz mesecnega koledarja + sest dni naprej */
function listView() {
    $this->pi_setPiVarDefaults();
```

```

$this->pi_loadLL(); // Loading the LOCAL_LANG values
/* Configuring so caching is not expected.
   This value means that no cHash params are ever set.
   We do this, because it's a USER_INT object! */
$this->pi_USER_INT_obj=1;
// Local settings for the listView function
$lConf = $this->conf["listView."];
// here we set the orderBy clause for lists
$this->selectParams['orderBy'] = $lConf['orderBy'];
$onlyUpcoming = $lConf['onlyUpcoming'];
// DATE MANIPULATIONS
if($this->piVars['date']) {
    $date= explode('-', $this->piVars['date']);
    $day = $date[0];
    $month = $date[1];
    $year = $date[2];
    $this->selectParams['daterange'] = 'day6';
}
else {
    if($this->selectParams['month']) {
        $month = $this->selectParams['month'];
    }
    else $month = date("m");
    if($this->selectParams['year']) {
        $year = $this->selectParams['year'];
    }
    else $year = date("Y");
}
.... → manjkajoca koda
}

```

Edina potrebna sprememba kode v tej metodi je bila ta, da smo spremenili parameter v daterange. Tu smo namesto prej uporabljenega day, uporabili day6. Tako bo metoda `de_getRecords` sedaj vrnila zapise za zeleno obdobje.

Za pregled arhiva dogodkov po mesecih je bila potrebna dodatna metoda `arhivView()`, ki je nadgradnja metode `listView`. Pregled dogodkov za posamezni mesec smo uporabili na straneh Januar 2005, Februar 2005, ..... do strani December 2006. Te strani so podstrani strani Arhiv dogodkov. Na vsako od teh podstrani smo dodali vtičnik, ki je omogočal pregled dogodkov za posamezni mesec. Torej na vsaki izmed teh 24 strani smo naredili naslednje:

Na enak način kot smo dodali vtičnik na stran En dogodek, dodamo na vsako stran en vtičnik koledarja. Nato v nastavitvah vtičnika v polje »Header« vpišemo »Arhiv dogodkov za #mesec# #leto#«, pri čemer za #mesec# vpišemo januar, ..., ali pa december in #leto# 2005 oz. 2006, odvisno seveda od tega, na kateri strani se nahajamo. Če smo na strani Januar 2005 bomo vpisali »Arhiv dogodkov za januar 2005« in na strani September 2006 bomo vpisali »Arhiv dogodkov za september 2006«. Nadalje moramo v nastavitvah vtičnika v polje »Code« vpisati »arhiv« in v polju »Startingpoint« dodati stran »Arhiv dogodkov«. Arhiv v polju Code tokrat pomeni, da želimo uporabiti to metodo, ki na strani prikaže dnevni koledar za obdobje celega meseca.

Da metoda deluje prav za posamezni mesec, smo uporabili trik. Vsem stranem, ki imajo imena npr. Januar 2005, Februar 2005, Marec 2005, ... smo vpisali tudi podnaslov strani. Tako smo za Januar 2005 vpisali 200501, za Februar 2005 smo vpisali 200502, ... V metodi `arhivView()` smo potem pogledali podnaslov strani. Glede na podlagi imena podnaslova se je potem izvedel pregled po bazi dogodkov za določen mesec in nam na strani prikazal vse dogodke tega meseca.

Del kode metode `arhivView()` je prikazan spodaj.

```

/* Metoda, ki prikaže arhiv dogodkov za posamezen mesec,
   Mesec in leto določimo iz podnaslova strani
*/
function arhivView() {

```



```

.... → manjkajoca koda

//Arhiv za leto 2005
/* V if stavku preverimo kaksen je podnaslov vsake podstrani
na kateri se nahaja vtičnik za prikaz arhiva dogodkov.
Glede na podnaslov, podamo funkciji de_getRecords ustrezne parametre */
if ($GLOBALS["TSFE"]->page['subtitle']){
    $leto = substr($GLOBALS["TSFE"]->page['subtitle'],0,4);
    $mesec = substr($GLOBALS["TSFE"]->page['subtitle'],4);
    $res = $this->de_getRecords($this->command,1,'',$mesec,$leto,'');
}
.... → manjkajoca koda

}

```

Na vsaki strani (od tistih 24), kjer se nahaja vtičnik koledarja, preverimo kakšen podnaslov ima stran. Če ima stran podnaslov oblike »YYYYMM« potem s funkcijo substr vzamemo YYYY kot oznako za leto (npr. 2005) in MM kot oznako z mesec (01 za januar, ...). Potem le še pokličemo metodo `de_getRecords` z ustreznimi parametri in ta nam vrne dogodke za želeno obdobje. Opazimo lahko, da je v zgornji kodi metode `de_getRecords` kot drugi parameter podana vrednost 1. Privzeto ima drugi parameter z imenom `$count` prirejeno vrednost 0. Če kličemo metodo `de_getRecords` in ji kot vrednost drugega parametra podamo 0, pomeni, da želimo zapise, ki jih metoda vrne, prikazati, medtem, ko z vrednostjo 1 drugega parametra zapise le preštujemo.

Prikaz praznikov smo naredili enostavno s pomočjo pogojnih stavkov `if`, kjer smo preverjali, če je dan enak dnevu iz fiksnega nabora datumov praznikov. Take smo v mesečnem koledarju označili z ustrežno barvo, da so se ločili od ostalih dni.

Težava je bila le z Velikonočnim ponedeljkom, saj ta ni na fiksen datum. Zato smo napisali kodo, ki za vsako leto sama izračuna in vrne datum na katerega pade Velikonočni ponedeljek.

```

//funkcija, ki izracuna datum za velikonocni ponedeljek za poljubno leto
function izracunajVelPonedeljek($leto) {
    $a = $leto % 19;
    $b = floor($leto / 100 );
    $c = $leto % 100;
    $d = floor($b / 4 );
    $e = $b % 4;
    $f = floor(($b + 8) / 25 );
    $g = floor(($b - $f + 1) / 3 );
    $h = (19 * $a + $b - $d - $g + 15) % 30;
    $i = floor($c / 4 );
    $k = $c % 4;
    $l = (32 + 2 * $e + 2 * $i - $h - $k) % 7;
    $m = floor(($a + 11 * $h + 22 * $l) / 451 );
    $dan = ($h + $l - 7 * $m + 114);
    $mesec = floor($dan / 31 );
    $dan = ($dan % 31) + 2;
    //kot rezultat dobimo tabelo z dnevom in mesecem
    return array($dan,$mesec);
}

```

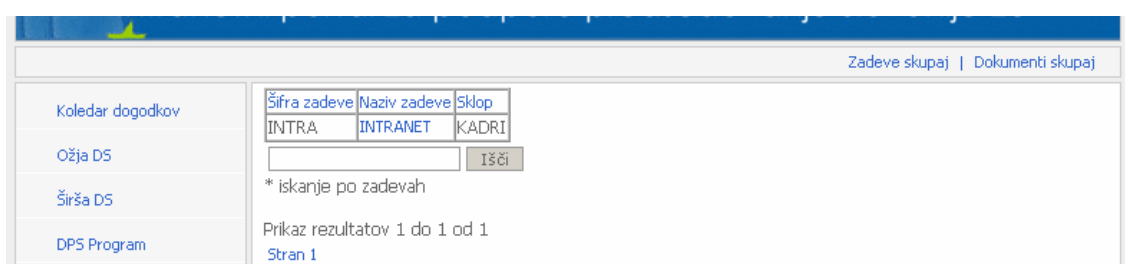
Koda ni nič drugega kot zapis enostavne formule za izračun datuma Velikonočnega ponedeljka, ki smo jo našli na spletni strani <http://en.wikipedia.org/wiki/Computus>.

Razširitev Simple Calendar nam ob namestitvi v sistem dopolni tudi bazo. Naredi nam štiri nove tabele. V teh tabelah hranimo podatke, ki jih lahko opredelimo za dogodek, ki ga dodamo dnevu na koledarju. Podatkov, ki jih lahko pripišemo posameznemu dogodku, je zelo veliko. Ker nismo potrebovali vseh teh polj, smo določena polja odstranili. To smo naredili tako, da smo v datoteki `tca.php` polja, ki jih ne potrebujemo, zakomentirali.

### 4.4.3 Razširitev Podatkovna baza EUP

Glavni namen portala je bil, da se v njem vodijo različni dokumenti, razprave o dokumentih in podobno. Kot smo omenili, so dokumenti razporejeni v več sklopov. Ena glavnih nalog je bila, da se dokumenti ustrezno organizirajo. Organizacijo dela s temi dokumenti smo izvedli s pomočjo razširitve, ki smo jo nekoliko neprimerno poimenovali kar Podatkovna baza EUP.

Kako so dokumenti organizirani in kaj lahko uporabniki z njimi počnejo, si lahko ogledamo na samih posnetkih uporabe. V nadaljevanju je prikazanih nekaj zaslonskih slik uporabe posameznih spletnih strani in njihov opis. Prva slika (Slika 47) prikazuje pregled zadev. Iz slike lahko vidimo, da je trenutno prikazana le ena zadeva. Prikazana je le zadeva z imenom INTRANET, saj ima le ta prirejene dokumente. Tako zlahka vemo, če iščemo dokumente z določeno zadevo ali obstajajo. Namreč, če dokument z določeno zadevo obstaja, moramo to zadevo videti v seznamu zadev. Če iskane zadeve v seznamu ni, tudi dokumentov s to zadevo ni. Pri prikazu zadev se nam vedno prikažejo šifra zadeve, naziv zadeve in sklop. Poleg samega seznama zadev imamo tudi iskalnik, ki išče po nazivih zadev. Ob kliku na naziv zadeve, v našem primeru INTRANET, se preselimo v pogled pregled dokumentov.



Slika 47: Sklop Kadri - pregled zadev

Na sliki 48 vidimo pregled dokumentov. V tem primeru je to pregled dokumentov za zadevo INTRANET. Pri prikazu dokumentov lahko vidimo, da se nam poleg prejšnjih podatkov (šifra zadeve, naziv zadeve in sklop) pojavita še dva nova podatka za in sicer datum zadeve in datum spremembe. Pri pregledu dokumentov vidimo še ikono, ki nam prikaže ali je dokument prebran ali ne (rumena mapa pomeni prebran dokument, rdeča neprebran). S pomočjo te ikone lahko ob naslednji prijavi vidimo katere dokumente smo prebrali in katere ne. Drugi podatek je ID dokumenta. ID dokumenta je sestavljen iz šifre sklopa, šifre zadeve in systemske številke dokumenta. Prikazan je tudi naslov dokumenta, status dokumenta, avtor dokumenta, tip dokumenta, datum dokumenta in datum spremembe.

Poleg samih podatkov v tabeli je vključen tudi iskalnik, ki išče po dokumentih. Ob kliku na naslov dokumenta, npr. Test\_delovni, se preselimo v pogled vsebina dokumenta.



Zadeve skupaj | Dokumenti skupaj

Koledar dogodkov  
Ožja DS  
Širša DS  
DPS Program  
DPS Kadri  
Vsi dokumenti sklopa  
DPS Logistika  
DPS Kompro  
DPS Proračun  
Ministrstva  
Razno  
Kontaktne osebe

Šifra zadeve INTRA  
Naziv zadeve INTRANET  
Sklop KADRI  
Datum zadeve 16.11.2005  
Datum spremembe 16.11.2005  
[Nazaj](#)

**Dokumenti**

ID dokumenta	Naslov dokumenta	Status	Avtor	Tip dok.	Datum dokumenta	Datum spremembe
KADRI-INTRA-107	Test _ delovni	delovni	Ernest	Ostalo	14.03.2006	14.03.2006
KADRI-INTRA-106	Test - končni	koncni	Ernest	Ostalo	14.03.2006	14.03.2006

\* Iskanje po dokumentih

Prikaz rezultatov 1 do 2 od 2  
Stran 1

Slika 48: Pregled dokumentov za zadevo INTRANET - sklop Kadri

Slika 49 prikazuje vsebino določenega dokumenta in obrazec za vnos razprave. Podatki, ki jih lahko vidimo, so ID dokumenta, naslov dokumenta, opis dokumenta. Vidimo tudi, katere priponke vsebuje dokument. Med podatki so še status dokumenta, avtor dokumenta, datum nastanka dokumenta, datum spremembe, zadeva, tip dokumenta in sklop. Iz naštetih podatkov lahko tako takoj razberemo, kdo je objavil določen dokument. Dokler je status dokumenta delovni, se odpre tudi obrazec za vnašanje razprav. Status delovni pomeni, da imajo bralci s posebnimi pravicami še možnost razprave na ta objavljeni dokument. Ko pa dokument pridobi status končni, vnos razprav ni več možen in obrazec za vnos razprav se ne prikaže več.

Koledar dogodkov  
Ožja DS  
Širša DS  
DPS Program  
DPS Kadri  
Vsi dokumenti sklopa  
DPS Logistika  
DPS Kompro  
DPS Proračun  
Ministrstva  
Razno  
Kontaktne osebe  
Iskanje  
Uredniki  
Pojasnila za uporabnike  
Prijava / odjava

ID dokumenta KADRI-INTRA-107  
Naslov dokumenta Test \_ delovni  
Opis dokumenta Test dokumenta delovni!!!  
Pripete datoteke  
Status delovni  
Avtor Ernest  
Datum dokumenta 14.03.2006  
Datum spremembe 14.03.2006  
Zadeva INTRA  
Tip dok. Ostalo  
Sklop KADRI

**Nova razprava**

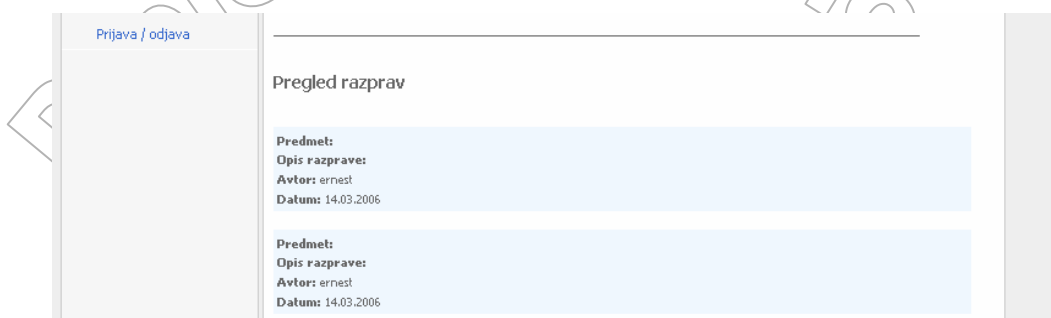
Predmet:

Opis:

Slika 49: Vsebina dokumenta in vnos razprave

Slika 50 prikazuje pregled razprav. Pregled razprav je narejen tako, da se za izbrani dokument po vsakem vnosu razprave le ta prikaže v seznamu vnesenih razprav. Da je pregled boljši, je vsaka posamezna razprava v modrem pravokotniku. Ob vnosu razprave vnesemo le predmet in opis

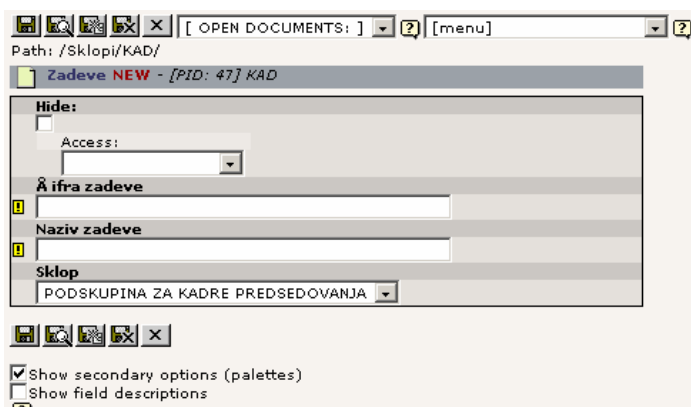
razprave, medtem, ko se pri pregledu razprav poleg predmeta in opisa razprave prikaže še avtor razprave in datum razprave. Avtor razprave je oseba, ki je prijavljena v čelni sistem in ki je vnesla razpravo. Za razliko od obrazca za vnos razprav, ki se ne prikaže več, ko dokument dobi status končni, se razprave o dokumentu prikazujejo ne glede na status dokumenta.



Slika 50: Pregled razprav

Zgornje slike so torej prikazovale pogled na dokumente s stališča bralca. Same dokumente pa seveda vnašamo v zalednem sistemu. Še tu si oglejmo nekaj tipičnih zaslonskih slik in jih opišimo:



Prva slika (Slika 51) prikazuje obrazec za vnos zadeve. Pri vnosu zadeve sta najpomembnejši polji šifra in naziv zadeve. Ti dve polji sta tudi obvezni za vnos. To nam prikazuje klicaj v rumenem kvadratu, ki je izpisan pred tema dvema poljema. Pri vnosu zadeve imamo možnost zadevo začasno skriti. To naredimo tako, da obkljukamo kvadratek z imenom Hide. Dostop do dokumenta posameznim skupinam uporabnikov čelnega sistema lahko omejimo z uporabo spustnega seznama v polju Access. Tu izberemo skupino, ki ji želimo omogočiti, da v čelnem sistemu vidi zadevo v seznamu zadev.

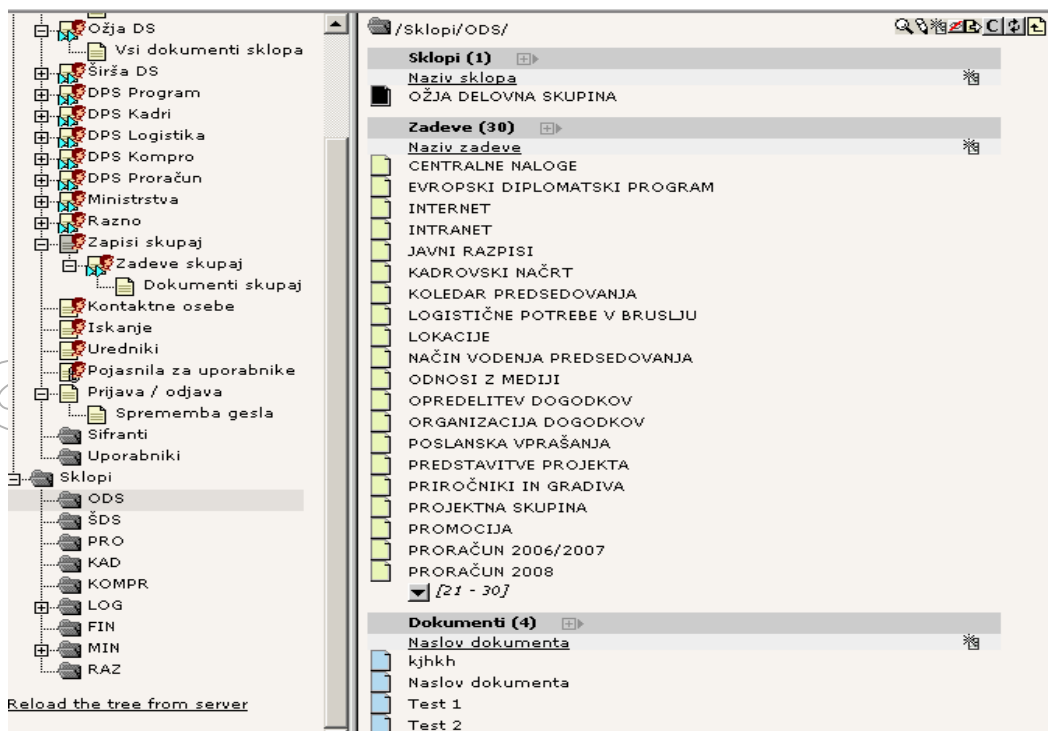


Slika 51: Vnos nove zadeve

Slika 52 prikazuje obrazec za vnos dokumenta. Vnosno polje Naslov dokumenta je obvezno za vnos. Tudi tukaj imamo možnost določiti, katera skupina uporabnikov čelnega sistema lahko dokument vidi. V opis dokumenta napišemo vsebino dokumenta. Pripete datoteke so lahko datoteke vseh vrst le datotek s končnico php ne smemo pripenjati. Dokument ima lahko le dve vrsti statusa: delovni ali pa končni, kar označimo v polju Status dokumenta. V polje avtor dokumenta napišemo ime in priimek avtorja dokumenta. Datum dokumenta lahko vpišemo sami ali pa izberemo s pomočjo izbire iz koledarčka. Naziv zadeve lahko izberemo iz spustnega seznama. Velja, da ima vsak dokument lahko le eno zadevo. Tip dokumenta prav tako izberemo iz spustnega seznama.

Slika 52: Vnos novega dokumenta

Zadeve in dokumente vnašamo v zalednem sistemu. Slika 53 prikazuje kako izgleda seznam zadev, dokumentov in sklopa v sistemski mapi. Na naši sliki lahko vidimo vsebino systemske mape za sklop Ožja delovna skupina oziroma s kratico ODS. Če želimo vnesti novo zadevo, se premaknemo pod kategorijo Zadeve in kliknemo na ikono . Odpre se obrazec, kot ga vidimo na sliki 51. Če želimo vnesti nov dokument, se premaknemo pod kategorijo Dokumenti in kliknemo na ikono . Ustrezen obrazec je prikazan na sliki 52.



Slika 53: Pregled sklopa, zadev in dokumentov za sklop ODS

**OPIS**

Ker gre za lastnosti, ki so tipične za naš portal, seveda nismo pričakovali, da bi lahko našli razširitev, ki bi že vsebovala želene funkcionalnosti. Zato je bila ta razširitev izdelana na novo. Služi za vnašanje in prikazovanje zadev, dokumentov in razprav. V sklopu te razširitve so bili razviti štirje vtičniki:

- a.) Prikaz zadev
- b.) Prikaz dokumentov
- c.) Prikaz razprav
- d.) Vnos razprav

Opišimo vsakega od teh vtičnikov.

**a.) Prikaz zadev**

Služi za vnašanje zadev v zalednem sistemu in prikazovanje le teh v čelnem sistemu. Ob izbiri posameznega sklopa v čelnem sistemu se prikažejo zadeve, urejene po abecednem vrstnem redu nazivov. Vtičnik prikaže samo zadeve, ki imajo vnesene dokumente. Pri vnosu zadev v zalednem sistemu nam vtičnik poskrbi, da se vnesene zadeve shranijo v bazo in skrbi za pravilnost vnosa. Če bi recimo v polje za datum vnesli besedo namesto datuma v obliki npr. 12.10.2005 nam vtičnik ne pusti podatkov shraniti v tej obliki ampak vnosno polje počisti.

**b.) Prikaz dokumentov**

Služi za vnos dokumentov v zalednem sistemu in prikazovanje dokumentov v čelnem sistemu. Dokumenti so v čelnem sistemu sortirani padajoče po datumu spremembe. Pri vnosu dokumentov v zaledni sistem vtičnik poskrbi, da se vneseni dokument shrani v bazo. Ob shranitvi dokumenta v bazo poskrbi, da se id dokumenta sestavi iz treh podatkov in sicer šifraSklopa (do 5 mest) - šifraZadeve (do 5 mest) - sistemskaŠtevilkaDokumenta. Vtičnik skrbi, da se pravilno prikazuje ikona prebrano oz. neprebrano.

**c.) Prikaz razprav**

Služi za prikazovanje razprav v čelnem sistemu. Vtičnik poskrbi, da se prikažejo le razprave, ki se nanašajo na trenutno prikazani dokument.

**d.) Vnos razprav**

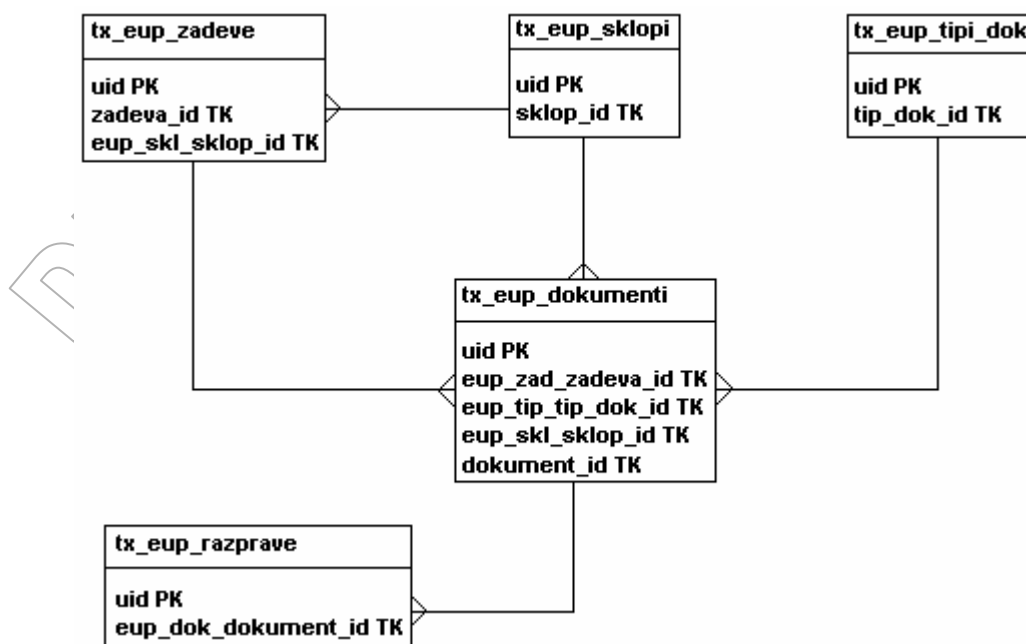
Služi za vnos razprav v čelnem sistemu.. Vtičnik poskrbi, da se ob vnosu razprave poleg predmeta in opisa zadeve, avtomatično shrani še avtor in datum razprave v bazo. Vtičnik prikazuje obrazec za vnos le v čelnem sistemu.

**IZVEDBA**

Ker smo razširitev naredili povsem na novo, smo morali uporabiti orodje Extension Kickstarter. Kot smo omenili že prej, je to orodje razširitev, ki je namenjena temu, da na ustrezen način razvijemo nove razširitve za Typo3.

V razširitvi bomo v bazo vpeljali tudi določene nove tabele. Razširitev Podatkovna baza EUP je sestavljena iz petih tabel: tx\_eup\_zadeve, tx\_eup\_sklopi, tx\_eup\_dokumenti, tx\_eup\_tipi\_dok in tx\_eup\_razprave. Tabela tx\_eup\_tipi\_zad se ne uporablja, zato jo v shemi nismo vključili.

Tabele so med seboj povezane na sledeč način:



Slika 54: Shema tabel za razširitev Podatkovna baza EUP

Opišimo malo zgornjo shemo!

V shemi smo označili katera polja v tabelah so tuji ključi (TK) in katera polja v tabelah so primarni ključi (PK). Vsak sklop ima lahko več zadev. Zato sta tabeli tx\_eup\_sklopi in tx\_eup\_zadeve povezani z relacijo tipa ena proti več. Prav tako velja za sklope in dokumente. Vsak sklop ima lahko več dokumentov. Tabeli tx\_eup\_sklopi in tx\_eup\_dokumenti sta spet povezani ena proti več. Vsaka zadeva ima lahko tudi več dokumentov. Vsak dokument ima lahko več razprav in en tip dokumentov ima lahko več dokumentov.

Poglejmo potek izdelave razširitve z razširitvijo Extension Kickstarter. Ogleдали si bomo le nekatere važnejše dele postopka.

Na začetku vnesemo osnovne podatke o sami razširitvi. Zato takoj po zagonu Extension Kickstarter v vrstici kjer piše »General info« kliknemo na plus. Odpre se novo okno, kamor bomo vpisali podatke o razširitvi. V polja, ki se nam prikažejo, vpišemo informacije, kot so npr. na sliki 55 in kliknemo na gumb »Update...«.

Extension: **Podatkovna baza EUP** (eup) Edit in Kickstarter Go back

**KICKSTARTER WIZARD**

General info	General info
<b>Podatkovna baza EUP</b>	Enter general information about the extension here: Title, description, category, author...
<b>New Database Tables</b>	<b>Title:</b> Podatkovna baza EUP
Tipi dokumenta	<b>Description:</b> extension za EUP sistem
Sklopi	<b>Category:</b> Frontend Plugins
Zadeve	<b>State</b> Beta (Under current development, should work partly)
Dokumenti	<b>Dependencies (comma list of extkeys):</b>
Razprave	<b>Author Name:</b> Tomaž Bobnar, Ernest Šlosar
Tipi zadeve	<b>Author email:</b> tomaz.bobnar@isa-it.si, ernest.slosar
<b>Extend existing Tables</b>	<input type="button" value="Update..."/>
<b>Frontend Plugins</b>	
Pregled zadev	
Pregled dokumentov	
Pregled razprav	
Vnos razprav	
<b>Backend Modules</b>	
<b>Integrate in existing Modules</b>	
Clickmenu items	
Services	
Static TypoScript code	
TSconfig	
Setup languages	
<input type="button" value="Click to Edit"/>	

Enter extension key:  
eup

Slika 55: Razvoj razširitve v Ext. Kickstarter - General info

Posebej pomembni sta polji Category in Author email. Prvo je pomembno, saj v tem polju povemo, kakšno razširitev pravzaprav ustvarjamo. Ker smo mi ustvarjali razširitev, ki vsebuje vtičnike za čelni del sistema, smo izbrali Frontend Plugins. Če bi našo razširitev pozneje objavili tudi na internetni strani <http://typo3.org>, bi bila razširitev na podlagi polja Category ustrezno shranjena in prikazana. Polje Author email je pomembno, saj ob problemih z razširitvijo nudi možnost, da stopimo v stik z avtorjem razširitve.

Nato v vrstici Setup language označimo, da bomo naredili razširitev tudi v slovenskem jeziku.

Extension: **Podatkovna baza EUP** (eup) Edit in Kickstarter Go back

**KICKSTARTER WIZARD**

General info	Setup languages
<b>Podatkovna baza EUP</b>	Start here by entering the number of system languages you want to use in your extension.
<b>New Database Tables</b>	<b>Enter which languages to setup:</b>
Tipi dokumenta	<input type="checkbox"/> Danish
Sklopi	<input type="checkbox"/> German
Zadeve	<input type="checkbox"/> Norwegian
Dokumenti	<input type="checkbox"/> Italian
Razprave	<input type="checkbox"/> French
Tipi zadeve	<input type="checkbox"/> Spanish
<b>Extend existing Tables</b>	<input type="checkbox"/> Dutch
<b>Frontend Plugins</b>	<input type="checkbox"/> Czech
Pregled zadev	<input type="checkbox"/> Polish
Pregled dokumentov	<input checked="" type="checkbox"/> Slovenian
Pregled razprav	<input type="checkbox"/> Finnish
Vnos razprav	<input type="checkbox"/> Turkish
<b>Backend Modules</b>	<input type="checkbox"/> Swedish
<b>Integrate in existing Modules</b>	<input type="checkbox"/> Portuguese
Clickmenu items	<input type="checkbox"/> Russian
Services	<input type="checkbox"/> Romanian
Static TypoScript code	<input type="checkbox"/> Chinese
TSconfig	<input type="checkbox"/> Slovak
Setup languages	<input type="checkbox"/> Lithuanian
<input type="button" value="Click to Edit"/>	<input type="checkbox"/> Icelandic
	<input type="checkbox"/> Croatian
	<input type="checkbox"/> Hungarian
	<input type="checkbox"/> Greenlandic
	<input type="checkbox"/> Thai
	<input type="checkbox"/> Greek
	<input type="checkbox"/> Chinese (Trad)
	<input type="checkbox"/> Basque
	<input type="checkbox"/> Bulgarian
	<input type="checkbox"/> Brazilian Portuguese
	<input type="checkbox"/> Estonian
	<input type="checkbox"/> Arabic
	<input type="checkbox"/> Hebrew

Enter extension key:  
eup

Slika 56: Razvoj razširitve v Ext. Kickstarter - Setup Languages

Ker bomo v razširitvi potrebovali podatke, ki jih bomo hranili v bazi podatkov, se kot naslednji korak premaknemo v vrstico z imenom »New Database Tables«. Glede na to, da je glavni namen te razširitve pravzajem in prikaz ključnih podatkov, ki jih bomo hranili na portalu, gre za enega najpomembnejših postopkov pri tvorjenju portala. Tu bomo ustvarili ustrezne tabele podatkovne baze in opisali imena vseh polj, njihove tipe in podobno. Kliknemo na plus na koncu vrstice. Odpre se novo okno z različnimi polji. V načinu »New Database Tables« definiramo, kakšna bodo polja v tabeli (tip in ime polja) in kakšen bo naslov tabele. Hkrati se avtomatsko generira tudi forma za vnos podatkov v zalednem sistemu. Ko bomo razširitev namestili, bomo torej že imeli vnosno formo, s pomočjo katere bomo vnašali podatke v bazo. Če nam ta vnosna forma ni povsem všeč, jo lahko spremenimo in dopolnimo. Mi smo bili z vnosno formo zadovoljni in jo nismo spreminjali. Po izbiri "New Database Tables" vpišemo ime tabele in imena polj ter tipe polj (Slika 57). Ko smo vnesli vsa polja, ki jih potrebujemo za vnos podatkov, smo pripravili vse potrebno za ustvarjanje nove tabele z imenom tx\_eup\_sklopi, v kateri bomo hranili podatke o sklopih. Nato spodaj kliknemo na gumb »Update...«.

Poleg tabele tx\_eup\_sklopi smo potrebovali še tabele v katerih bomo hranili podatke o zadevah, podatke o razpravah, podatke o dokumentih, podatke o tipih dokumentov in podatke o tipih zadev. Za vsako od zgoraj naštetih stvari smo morali narediti novo tabelo z ustreznimi polji. Nove tabele smo naredili na enak način kot tabelo tx\_eup\_sklopi. Pri ustvarjanju novih polj je zlasti potrebno paziti na ime polja. Velja, da ime polja ne sme vsebovati presledka. Pri izbiri tipa polja pa imamo na voljo številne možnosti npr. string input (vnosno polje za niz znakov), text area (polje za večvrstični vnos znakov), checkbox (potrditveno polje – logična vrednost), link (povezava), date (datum), date and time (datum in čas), integer (celo število), ... Če obkljukamo kvadrček v polju Exclude-field lahko kasneje v nastavitvah nastavimo, kateri uporabniki imajo na voljo to polje prikazano za vnos in kateri ne. To je dobro, saj včasih želimo, da bi lahko določena polja za vnos imeli le izbrani uporabniki in ne vsi.

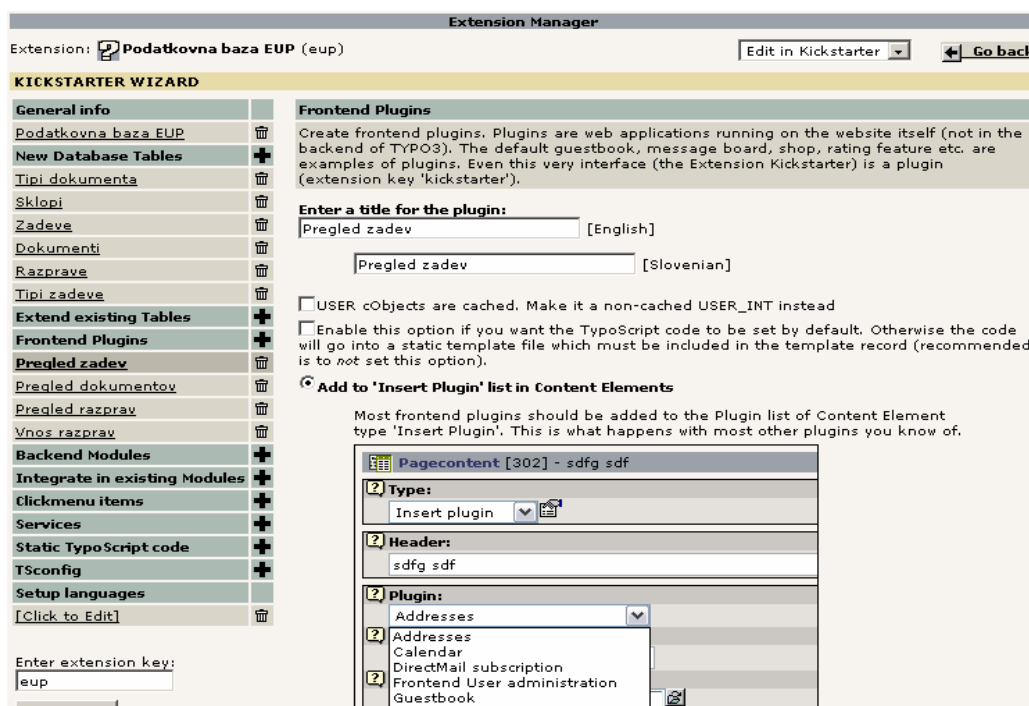
Slika 57: Razvoj razširitve v Ext. Kickstarter - New Database Tables

Ker ustvarjamo vtičnike za prikaz v čelnem sistemu, se premaknemo v vrstico »Frontend Plugins«. Tukaj ustvarimo vtičnike za čelni sistem. Vtičnike potrebujemo za prikaz zadev, dokumentov, razprav v čelnem sistemu in vtičnik za vnos razprav v čelnem sistemu. Vtičnik v vrstici »Frontend Plugins«

naredimo na določen način. Vsakemu vtičniku moramo določiti neko ime. Za naš prvi vtičnik smo izbrali ime Pregled zadev. Pri nastavitvah imamo še možnost izbire, ali želimo dodati vtičnik v seznam vtičnikov, ki so na voljo za vstavljanje. To možnost običajno vedno izberemo.

Pred nastavitve shranimo, moramo iz seznama v katerem imamo tabele, ki jih uporabljamo v celi razširitvi, izbrati tisto, na katero se nanaša vtičnik. V našem primeru izberemo tabelo z imenom tx\_eup\_zadeve. Na koncu kliknemo na gumb »Update...«, da te nastavitve shranimo.

Na isti način kot za vtičnik Pregled zadev naredimo še vtičnike za Pregled dokumentov (tabela tx\_eup\_dokumenti), Pregled razprav (tabela tx\_eup\_razprave) in Vnos razprav (tabela tx\_eup\_razprave). Pri ustvarjanju vtičnikov nam ni potrebno pisati nobene kode. Za kodo in pravilen klic metod poskrbi sam Extension Kickstarter. Kodo vtičnika dopolnjujemo oz. popravljamo šele takrat, ko moramo delovanje vtičnika prilagoditi našim željam.



Slika 58: Razvoj razširitve v Ext. Kickstarter - Frontend Plugins

Nadaljujemo s pripravo celotne razširitve. Tu moramo opraviti še zadnji vnos, ki pa ga nikakor ne smemo pozabiti. To je vnos v polje Enter extension key. Kot smo že povedali, je vsaki razširitvi prirejen ključ razširitve. Na podlagi tega ključa razširitve uvozimo v sistem. Da lahko razširitev sploh namestimo v sistem, ji moramo torej določiti ključ. Če gre za razširitev, ki bi bila dana v javni repozitorij razširitev, bi moral biti ta ključ unikaten. Ker pa bo ta razširitev uporabljena le v našem sistemu, to lahko zanemarimo. Zato smo se odločili kar za ključ eup. V polje Enter extension key vtikamo torej eup in kliknemo na gumb »Update...«. Kliknemo še na gumb View result. S tem dobimo seznam vseh datotek, ki jih bo vsebovala naša razširitev.

Zadnji korak je klik na gumb Write s pomočjo katerega razširitev prenesemo v mapo, kjer so tudi ostale razširitve sistema. Razširitev nato še namestimo v sistem na prej opisan način.

Nato je potrebno razširitev testirati. Namreč Extension Kickstarter naredi ogromno kode, ki vsa ni potrebna ali pa ne deluje ravno tako, kot potrebujemo. Zato moramo na Extension Kickstarter gledati kot na orodje, ki nam naredi ogrodje za razširitev. Da je razširitev taka, kot smo želeli, pa je potrebno kodo v datotekah razširitve ustrezno prilagoditi.



Da smo razširitev Podatkovna baza EUP testirali, smo na straneh Ožja DS, Širša DS, DPS Program, DPS Kompro, DPS Logistika, DPS Kadri, DPS Proračun, Ministrstva in Razno dodali vtičnika za pregled zadev in pregled dokumentov. Vtičnike smo dodali na že prej opisani način: v navigacijskem delu kliknemo na strani, iz kontekstnega menija izberemo New in po izbiri »Click here for wizard« (Pagecontent → Click here for wizard) v podrobnem prikazu v kategoriji Plugins izberemo Pregled zadev in določimo položaj za vtičnik. Nato opravimo še ustrezne nastavitve. Tako za stran Ožja DS pri nastavljanju vtičnika za pregled zadev v polje »Header:« vpišemo »Pregled zadev«. V polje »Startingpoint« dodamo sistemsko mapo »ODS«. Vtičnik za pregled dokumentov dodamo na enak način le v polje »Header« vpišemo »Pregled dokumentov«.

Ko smo razširitev testirali, smo ugotovili, da bo potrebno razširitev predelati, če hočemo, da bo ustrezala željam naročnika. Popravki, ki smo jih naredili so bili narejeni v datotekah tx\_eup\_pi1.php, tx\_eup\_pi2.php, tx\_eup\_pi3.php in tx\_eup\_pi4.php.

Najprej je bilo potrebno popraviti kodo, ki prikazuje zadeve v čelnem sistemu. Vtičnik je namreč prikazal vse zadeve, ki so bile vnesene v zalednem sistemu. Mi pa smo želeli, da se prikažejo le zadeve, ki imajo prirejene dokumente. V datoteki tx\_eup\_pi1.php, ki je namenjena vnosu in pregledu zadev, smo popravili del kode kjer se prikazujejo zadeve. Pogoji, da se v čelnem sistemu zadeva lahko sploh prikaže je, da mora obstajati vsaj en dokument, ki je prirejen tej zadevi. Del kode iz metode listView() v datoteki tx\_eup\_pi1.php kjer preštejemo in pokažemo zadeve, ki imajo prirejene dokumente je napisan spodaj.

```
/* v bazi pogledamo katere zadeve imajo prirejene dokumente
   in te zadeve prestejemo */
$query = ' AND EXISTS (SELECT * FROM tx_eup_dokumenti';
$query .= ' WHERE tx_eup_dokumenti.eup_zad_zadeva_id = tx_eup_zadeve.uid ';
$query .= $this->cObj->enableFields('tx_eup_dokumenti').' )';
$res = $this->pi_exec_query("tx_eup_zadeve",1,$query);
list($this->internal["res_count"])=$GLOBALS['TYPO3_DB']->sql_fetch_row($res);
```

V zgornji kodi metodi pi\_exec\_query podamo ime tabele, v kateri želimo narediti poizvedbo. Kot drugi parameter ji podamo 1. To pomeni, da želimo zapise v bazi prešteti. Kot tretji parameter pa podamo ukaz v jeziku SQL s katerim pregledamo, ali ima zadeva prirejen kakšen dokument.

```
/* v bazi pogledamo katere zadeve imajo prirejene dokumente
   in te zadeve prikazemo */
$query = ' AND EXISTS (SELECT * FROM tx_eup_dokumenti';
$query .= ' WHERE tx_eup_dokumenti.eup_zad_zadeva_id = tx_eup_zadeve.uid ';
$query .= $this->cObj->enableFields('tx_eup_dokumenti').' )';
$res = $this->pi_exec_query("tx_eup_zadeve",0,$query);
$this->internal["currentTable"] = "tx_eup_zadeve";
```

Tukaj je zgornja koda zelo podobna prejšnji. Razlika je le, da metodi pi\_exec\_query podamo kot drugi parameter število 0. S tem smo metodi povedali, da želimo zapise, če obstajajo, prikazati.

Naslednja sprememba, ki smo jo želeli, se je nanašala na lep prikaz podatkov. Privzeto so se prikazovali v tabeli brez roba. Želeli smo, da ima prikazana tabela tudi rob, saj se podatki potem bolj ločijo eden od drugega. Zato smo tabeli dodali rob.

```
// Adds the whole list table
$fullTable.= $this->pi_list_makelist($res);
// tabeli naredimo rob
// znacko <table zamenjamo z <table border=1 frame=box cellspacing=0
$stari_niz = "<table";
$novi_niz = "<table border=1 frame=box cellspacing=0 ";
$fullTable = str_replace($stari_niz,$novi_niz,$fullTable);
```

V ta namen smo del značke »<table« nadomestili z »<table border=1 frame = box cellspacing=0«. To zamenjavo nam je omogočila metoda iz PHP in sicer str\_replace.

Prav tako smo morali poskrbeti, za to, da se je ob vsakem dokumentu prikazala ikona, ki je označevala, če je bil dokument že prebran. Sledenje branosti smo izvedli s pomočjo piškotkov. Piškotki (angl. cookies) so mehanizem, ki omogoča spletnemu brskalniku, da si zapomni, kaj je uporabnik počel na spletni strani (kdaj je odprl neko stran, kaj je vpisal v obrazec, kateri gumb ali povezavo je pritisnil ...). Najlažje si jih predstavljamo kot posebno vrsto spremenljivk, ki jih hrani spletni brskalnik (kje in v kakšni obliki je odvisno od posameznega brskalnika). V metodi `pi_list_row` v datoteki `tx_eup_pi2.php`, ki je namenjena vnosu in pregledu dokumentov, smo dodali del kode, kjer se glede na stanje piškotkov prikaže ustrezna ikona, ki označuje prebranost dokumenta.

```
/* vsakemu dokumentu v celnem sistemu prikazemo
   ikono prebrano/neprebrano */
if ($editPanel)      $editPanel("<TD>".$editPanel."</TD>");
    $spot = '/EUP/fileadmin/_temp/';
    $prebrano = '';
// preverimo, ce je dokument prebran
if (isset($_COOKIE['cookie'].$this->getFieldValue("uid")))
    // ce je dokument prebran, prikazemo novo sliko
    $prebrano = '';
```

Vsakemu dokumentu smo hoteli pripeti tudi določeno število priponk. Da bi bilo to možno, smo morali v metodo `getFieldContent` dodati kodo, ki bo omogočala, da bomo dokumentu lahko pripeli priponke.

```
// izpisemo vsebino tabele
function getFieldContent($fN)    {
    switch($fN) {

        ..... <-- manjkajoca koda

        /* prebere iz tabele imena datotek, ki pripadajo dokumentu
           in naredi povezave do njih */
        case "priponka":
            // v spremenljivko narekovaj shranimo dvojni narekovaj (")
            $narekovaj = chr(34);
            //pot, kjer so na strezniku datoteke
            $spot = '/EUP/uploads/tx_eup/';
            //sprejememo niz iz tabele v katerem so zapisane pripete datoteke
            $vhodni_niz = $this->internal["currentRow"][$fN];
            $filelink = '';
            $delim = ','; //v tabeli so imena datotek locena z vejico
            //zanka gre skozi seznam z imeni datotek in tvori povezave
            while (is_string($vhodni_niz)) {
                // v spremenljivko $vhodni_niz_file shranimo ime datoteke
                // ki ji zelimo narediti povezavo
                $vhodni_niz_file = substr($vhodni_niz,0,strpos($vhodni_niz','));
                if ($vhodni_niz_file)
                    // vsaki datoteki dodamo povezavo do nje
                    $filelink = $filelink.'<a href ='.$narekovaj.$spot;
                    $filelink = $filelink.$vhodni_niz_file.$narekovaj.'>';
                    $filelink = $filelink.$vhodni_niz_file.'</a><br>';
                    /* ta if stavek potrebujemo, da lahko naredimo povezavo
                       zadnji datoteki v spremenljivki $vhodni_niz */
                    if (!strpos($vhodni_niz',')) {
                        $filelink=$filelink.'<a href ='.$narekovaj.$spot;
                        $filelink = $filelink.$vhodni_niz.$narekovaj.'>'.$vhodni_niz.'</a>';
                        break;
                    }
                /* iz prvotnega niza v spremenljivki $vhodni_niz odrezemo najprej
                   ime prve datoteke v nizu, nato drugo in tako vse do
                   zadnjega imena datoteke */
                $vhodni_niz = substr($vhodni_niz,strpos($vhodni_niz',')+1);
            }
            //metoda vrne niz, v katerem so povezave do datotek, locene z vejico
```

```

    return $filelink;
    break;
}
}

```

V datoteki `tx_eup_pi2.php` je bila prav tako spremenjena koda pri prikazu dokumentov v tabeli. Tudi tukaj smo tabeli dokumentov dodal rob.

```

$fullTable.='<h3>Dokumenti</h3>';
$fullTable.=$this->pi_list_makelist($res);
// tabeli naredimo rob
// del znacke <table zamenjamo s <table border=1 frame=box width=500 cellspacing=0
$stari_niz = "<table";
$novi_niz = "<table border=1 frame=box width=500 cellspacing=0 ";
$fullTable = str_replace($stari_niz,$novi_niz,$fullTable);

```

V zgornji kodi smo ravno tako kot pri prikazu zadev v tabeli, uporabili metodo `str_replace` za prikaz roba na tabeli, ki prikazuje dokumente.

V datoteki `tx_eup_pi2.php`, ki je namenjena pregledu razprav, je bila predelana metoda `makeListItem()`, saj smo hoteli prikazovanje razprav narediti čim bolj enostavno in pregledno. Hoteli smo prikazati le predmet zadeve, opis zadeve, avtorja in datum zadeve. Poleg tega smo tabeli določili, da ima nebesno modro ozadje, da se zapisi razprav še bolj ločijo eden od drugega.

```

//metoda naredi seznam vseh razprav glede na posamezen dokument
function makeListItem() {
    $out='<table style="background-color:#eff7ff; width:570px"><tr>
    $out .= <td><p'. $this->pi_classParam("listrowField-naslov");
    $out .= '><b><font size="1">Predmet: '. $this->getFieldContent("naslov");
    $out .= '</font></b><br';
    $out .= $this->pi_classParam("listrowField-opis");
    $out.='><font size="1"><b>Opis razprave: </b>'. $this->getFieldContent("opis");
    $out .= '</font><br'. $this->pi_classParam("listrowField-avtor");
    $out .= '><font size="1"><b>Avtor: </b>'. $this->getFieldContent("avtor");
    $out .= '</font><br'. $this->pi_classParam("listrowField-avtor");
    $out .= '><font size="1"><b>Datum: </b>'. $this->getFieldContent("crdate");
    $out .= '</font><br'. $this->pi_classParam("listrowField-eup-dok-dokument-id");
    $out .= '><font size="1">'. $this->getFieldContent("eup_dok_dokument_id");
    $out .= '</font></p></td></tr><br></table>';
    return $out;
}

```

V sami kodi smo med ustrezne oblikovne značke jezika HTML s katerimi smo oblikovali ustrezen prikaz tabele, vsebino samih celic tabele dodajali s pomočjo ukazov v jeziku PHP. Tako nam npr. ukaz `$this->getFieldContent("naslov")` vrne naslov oz. predmet razprave.

Datoteka `tx_eup_pi4.php`, ki omogoča vnos razprav v čelnem sistemu, je bila v celoti predelana. Potrebno je bilo namreč narediti formo za vnos v čelnem sistemu. Spodaj je prikazan del kode metode `main`, ki je edina metoda v `tx_eup_pi4.php`. Koda prikazuje, kako je narejena vnosna forma za razprave.

```

/* Koda se nahaja v metodi main in naredi formo
oz. obrazec za vnos razprav */
// ce je bil izveden klik na gumb Vpisi
if (isset($this->piVars['submit_button'])) {
    /* poiščemo sistemsko mapo, kjer je shranjen trenutni dokument
    in tja shranimo se razpravo */
    $query = 'SELECT pid,status FROM tx_eup_dokumenti WHERE uid = ';
    $query .= addslashes($this->piVars['DATA']['eup_dok_dokument_id']);
    $res = mysql(TYPO3_db,$query);
    $row = mysql_fetch_assoc($res);
    $result = '';

```

```

$submit_razprava = 1;
t3lib_div::_POST($submit_razprava);
do {
    $result = $row['pid'];
} while ($row = mysql_fetch_assoc($res));

$insertArray = array(
    'naslov' => addslashes($this->piVars['DATA']['naslov']),
    'opis' => addslashes($this->piVars['DATA']['opis']),
    'avtor' => $GLOBALS['TSFE']->fe_user->user['username'],
    'crdate' => date("U"),
    'eup_dok_dokument_id'=>addslashes($this->piVars['DATA']['eup_dok_dokument_id']),
    'pid' => $result);
//v tabelo tx_eup_razprave vstavimo podatke, ki so v spremenljivki $insertArray
$query= $GLOBALS['TYPO3_DB']->exec_INSERTquery('tx_eup_razprave',$insertArray);
/* po vstavljanju podatkov v bazo prikazemo novo formo oz. obrazec za vnos
   razprave */
$content = '<h5>Razprava je bila shranjena!</h5><br></br>';
$content .= '<h3>Nova razprava</h3><form action=""';
$global_id = $GLOBALS["TSFE"]->id;
$array_raz = array("razprava_submit">1);
$content .= $this->pi_getPageLink($global_id,'',$array_raz);
$content .= '" method="POST"><input type="hidden" name="no_cache" value="1">';
$content .= <table><tr><td>Predmet:</td><td><input type="text" name=""';
$content .= $this->prefixId.' [DATA][naslov]" value=""</td></tr>';
$content .= '<tr><td>Opis:</td><td><textarea rows="7" cols="50" name=""';
$content .= $this->prefixId.' [DATA][opis]" value="">';
$content .= '</textarea></td></tr><tr></tr>';
$content .= '<tr><td></td><td><input type="hidden" name=".'.$this->prefixId;
$content .= '[DATA][eup_dok_dokument_id]" value=""';
$content .= $ _SESSION['trenutni_dokument'];
$content .= '"></td></tr><tr><td><input type="submit" name=".'.$this->prefixId;
$content .= '[submit_button]" value="Vnesi"></td></tr></table></form>';
}

```

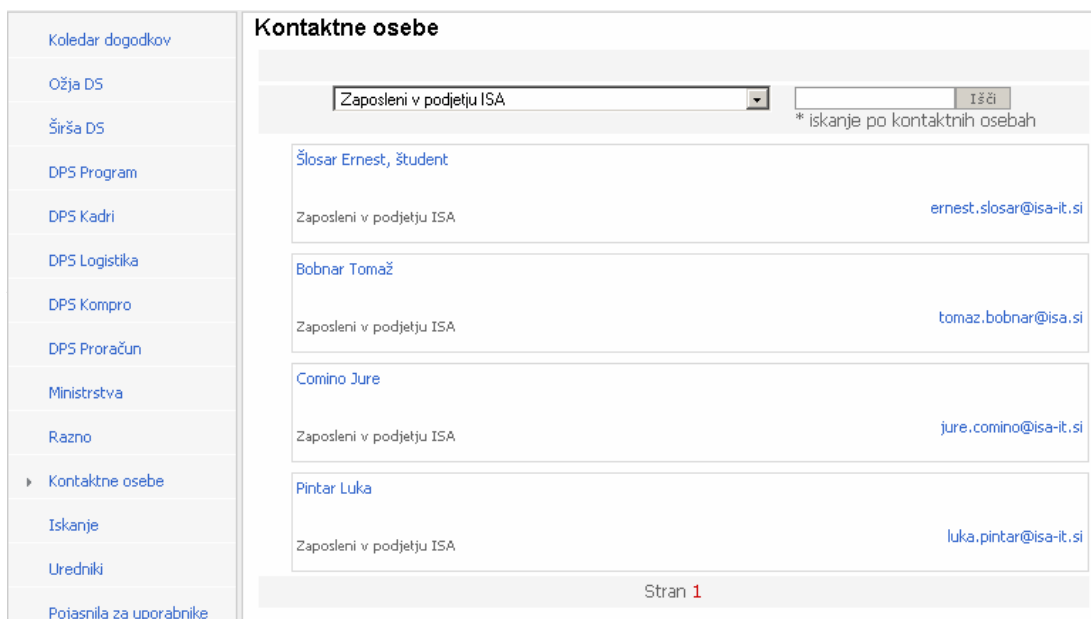
Ob vnosu nove razprave se le ta shrani v tabelo tx\_eup\_razprave, kjer hranimo podatke o razpravah. Ob shranitvi razprave se ponovno zgenerira obrazec za vnos nove razprave.

#### 4.4.4 Razširitvi Address list in Address Directory

Za portal smo potrebovali tudi razširitev, ki bi omogočala vnos in prikaz kontaktnih oseb. Ob namestitvi sistema Typo3 imamo na voljo razširitev, ki omogoča vnos podatkov o osebah. Pomanjkljivost razširitve je ta, da ne omogoča prikaz teh oseb v čelnem sistemu. Zato smo sklenili, da pogledamo na stran z razširitvami (<http://typo3.org>) ali obstaja razširitev, ki omogoča vnos in prikazovanje podatkov o osebah. Po kratkem iskanju smo našli razširitev. Razširitev se imenuje Address Directory in deluje v povezavi z razširitvijo Address list. Address Directory prikaže vnesene podatke o osebah v čelnem sistemu. Ustrezala nam je tudi možnost, da določene podatke o osebi lahko vnesemo le posebej preko obrazca, ki je ločen od ostalih vnosov podatkov. To smo izrabili za to, da smo v zalednem sistemu dodali možnost vnosa (državnega) organa, kateremu oseba pripada, saj smo hoteli, da ta podatek spreminja le pooblaščen oseba.

Kako je videti prikaz in vnos kontaktnih oseb, si lahko ogledamo na spodnjih dveh slikah.

Na sliki 59 lahko vidimo prikaz kontaktnih oseb. Osebe lahko prikažemo na podlagi organa kateremu pripadajo (so zaposleni, so člani skupine, oddelka, ...). Tako so na sliki prikazane tiste osebe, za katere v opisu piše, da so v skupini Zaposleni v podjetju ISA. Na strani za kontaktne osebe imamo tudi iskalnik, ki omogoča iskanje po kontaktnih osebah. Ob kliku na priimek in ime osebe ali pa ob kliku na e-poštni naslov osebe se preselimo v podrobni pogled za izbrano osebo. Tam vidimo vse podatke o osebi.

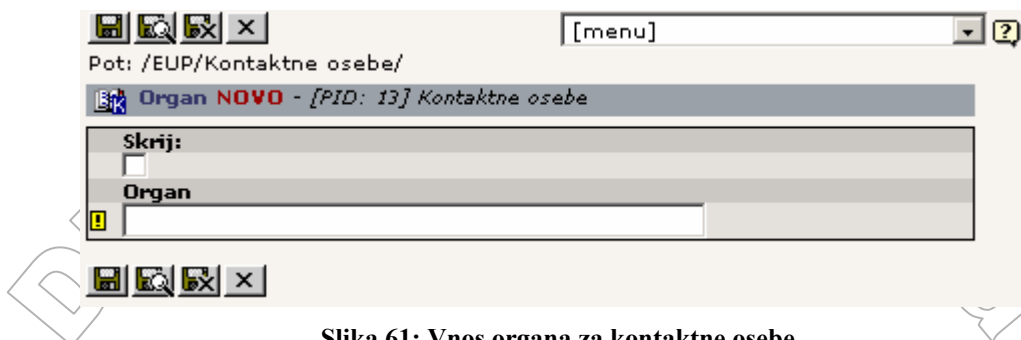


Slika 59 : Pregled kontaktnih oseb

Slika 60 prikazuje obrazec za vnos osebe. Nobeno izmed polj ni obvezno za vnos.

Slika 60: Vnos kontaktne osebe

Kot smo omenili, je organ, ki mu oseba pripada, mogoče vnesti le posebej. Slika 61 prikazuje ustrezeni obrazec. Kot lahko vidimo, je polje z imenom Organ obvezno za vnos.



Slika 61: Vnos organa za kontaktne osebe

**OPIS**

Na strani »Kontaktne osebe« so navedeni vsi udeleženci v projektu predsedovanja. Seznam je urejen po abecednem redu. Seznam poleg imena vsebuje ime (državnega) organa, iz katerega prihaja sodelujoči, kontaktne podatke osebe in opis njene vloge v projektu priprav na predsedovanje Slovenije EU.

Kontaktne osebe vnašamo v zalednem sistemu na strani Kontaktne osebe. Pri prikazovanju kontaktnih oseb v čelnem sistemu imamo na voljo tri različne prikaze.

Ob kliku na stran Kontaktne osebe v čelnem sistemu lahko vidimo naslednje podatke o osebi:

- vrsta organa, kateri kontaktna oseba pripada
- priimek in ime, akademski naziv
- telefon
- faks
- e-poštni naslov

Sistem prikaže le tiste podatke med naštetimi, ki so dejansko vneseni.

Če iščemo osebe glede na organ, kateremu pripadajo, dobimo pogled z naslednjimi podatki:

- priimek in ime, akademski naziv
- organ
- telefon
- faks
- mobitel
- e-poštni naslov
- vloga na projektu

Seveda spet vidimo le tiste podatke med naštetimi, ki so dejansko vneseni.

Detail view oz. podroben pogled pa lahko prikaže naslednje podatke o osebi:

- priimek in ime, akademski naziv
- vloga na projektu
- organ
- opis osebe
- telefon
- faks
- mobitel
- e-poštni naslov
- domača stran
- slika

**IZVEDBA**

Z Ext Manager smo razširitev Address Directory najprej uvozili in nato namestili v sistem. Na strani Kontaktne osebe smo dodali vtičnik razširitve Address Directory.

V razširitvi Address Directory je bilo potrebno v glavni datoteki `tx_spdirectory_pi1.php` odstraniti metodo `htmlentities()`, ki je narobe pretvarjala šumnike. Ob namestitvi je vtičnik na čelnem sistemu omogočal prikaz abecednega seznam oseb. S pomočjo tega seznama smo zlahka poiskali osebo z določeno začetnico. Ker pa je naročnik želel, da se iskanje kontaktnih oseb izvaja samo na podlagi iskalnika na strani kontaktnih oseb in glede na organ osebe, smo abecedni seznam oseb iz datoteke `tx_spdirectory_pi1.php` odstranili. Tudi pri tej razširitvi je bilo potrebno v datotekah `locallang.php` in `locallang_db.php` narediti prevode.

V datoteki `tx_spdirectory_pi1.php` smo naredili popravke v metodah `makeDetail` in `makelist`. Metoda `makeDetail` je omogočala prikaz vseh podatkov, ki smo jih vnesli v zalednem sistemu, prikazala ni le iz katerega organa je določena oseba. Razlog je bil v tem, ker sta tabeli oseb in organov povezani preko vmesne tabele, v kateri so podatki zapisani v dveh stolpcih. Prvi stolpec vsebuje id oseb in drugi stolpec id organa. Zato smo morali, če smo želeli prikazati organ vsaki osebi v čelnem sistemu, dodati kodo, ki bo za vsako osebo poiskala, kateremu organu pripada. To kodo smo dodali v metodo `makeDetail`.

Del kode, ki je bil dodan, je tale:

```
/* s pomocjo tega dela kode najprej pogledamo v tabelo
   tt_address podatke o osebi na podlagi id osebe */
$pidList=$this->pi_getPidList($this->conf["pidList"],$this->conf["recursive"]);
$query = "SELECT * FROM tt_address WHERE pid IN (".$pidList.")";
$query = "." ".$this->cObj->enableFields(tt_address)." AND uid = ".$id;
/* poizvedba v spodnji vrstici nam iz tabele tx_spdirectory_mm na
   podlagi id osebe poišče ustrezen uid_foreign,
   ki je hkrati id nekega drzavnega organa, ce seveda obstaja */
$query2 = "SELECT uid_foreign FROM tx_spdirectory_cat_mm WHERE uid_local='";
$query2 .= $id."'";
$res2 = mysql(TYPO3_db,$query2);
$row2 = mysql_fetch_assoc($res2);
$res = mysql(TYPO3_db,$query);
$row = mysql_fetch_assoc($res);
/* ce smo nasli v tabeli uid_foreign za doloceno osebo,
   naredimo poizvedbo s katero v tabeli tx_spdirectory_cat
   poiscemo vse podatke o organu glede na uid_foreign */
if($row2[uid_foreign] != ''){
    $query3 = "SELECT * FROM tx_spdirectory_cat WHERE pid IN (".$pidList.")";
    $query3 .= " ".$this->cObj->enableFields(tx_spdirectory_cat);
    $query3 .= " AND uid = ".$row2[uid_foreign];
    $res3 = mysql(TYPO3_db,$query3);
    $row3 = mysql_fetch_assoc($res3);
}
}
```

V zgornji kodi najprej pogledamo v vmesno tabelo, kjer imamo povezave med osebami in organi, ali je oseba povezana s kakšnim organom. Če je oseba povezana, smo v tej vmesni tabeli našli, kateremu organu pripada. Potem smo v tabeli, kjer hranimo organe, na podlagi id organa le še pogledali, kakšen naziv ima organ.

V metodi `makelist` je bila dodana koda, ki je zelo podobna prej opisani, zato je ne bomo podrobneje razlagali.

```
/* del kode, ki je napisan tukaj uporabimo za poizvedbo po tabeli
   s katero poiscemo vsaki osebi organ, ki ji pripada */
$stevec = 1;
$odg = '';
while($row = mysql_fetch_assoc($res)) {
    if($stevec == 1){
        $query2 = "SELECT uid_foreign FROM tx_spdirectory_cat_mm";
        $query2 .= " WHERE uid_local='".$row[uid]."'";
        $res2 = mysql(TYPO3_db,$query2);
        $row2 = mysql_fetch_assoc($res2);
    }
}
```

```
$query3 = "SELECT * FROM tx_spdirectory_cat WHERE uid = ";  
$query3 .= $row2[uid_foreign];  
$res3 = mysql(TYPO3_db, $query3);  
$row3 = mysql_fetch_assoc($res3);  
$odg = $row3[catname];  
$stevec = $stevec - 1;  
}  
..... → manjkajoca koda  
}
```

Koda deluje na podoben način kot tista v metodi `makeDetail`. Razlika je le v tem, da smo tukaj uporabili `stevec`, ki nam je omogočil pravilno poizvedovanje. Pri metodi `makeDetail` naredimo poizvedbo le za eno osebo, v metodi `makeList` pa to naredimo za vse osebe naenkrat.



## 5. Zaključek

### Prihodnost portala EUP

S postavitvijo portala EUP je Vlada Republike Slovenije pridobila pomembno aplikacijo, ki ji bo omogočala komuniciranje med vsemi ministrstvi in drugimi kadri, povezani s predsedovanjem Slovenije EU. Portal bo najbolj zaživel po 1.januarju 2008, ko bo Slovenija začela s predsedovanjem EU. Portal je narejen tako, da se ga lahko nadgrajuje in popravlja. Že sedaj se portal nenehno nadgrajuje z različnimi novimi funkcionalnostmi.

### Vsebina aplikacije, njena izvedba in uporabljena tehnologija

Pred začetkom dela na projektu je bila izdelava portala EUP videti zelo enostavna naloga. Gre za bazo podatkov in portal na spletu, s pomočjo katerega v bazo vpisujemo podatke in jih na ustrezen način pregledujemo. Vse se je zdelo zelo logično in enostavno. Največjo oviro nam je predstavljalo orodje Typo3. Tehnologija SUV Typo3 je zelo dobra in omogoča zelo stabilno delovanje portalov. Vendar je bil to prvi projekt v podjetju ISA.IT, kjer je bil sistem Typo3 uporabljen. Potrebno je bilo kar nekaj časa, da se je skupina, ki je delovala na projektu, seznanila z vsemi prednostmi in pomanjkljivostmi SUV Typo3.

### Moja izkušnja

Ob delu na projektu EUP sem pridobil veliko izkušenj na področju internetne tehnologije. Pred tem projektom sem le malo poznal internetno tehnologijo. Poznal sem le programska jezika Java in Delphi, jezik HTML in vedel nekaj o oblikovanju spletnih strani s CSS. Skozi projekt EUP sem se naučil še dela z bazami podatkov in skriptnega jezika PHP. SUV Typo3 sem se naučil uporabljati s pomočjo internetnega gradiva, ki sem ga našel na strani <http://typo3.org>. Pri tem so mi bili v veliko pomoč video posnetki, ki so nadzorno pokazali kaj naredi kakšna izbira v sistemu. V začetku mi je bilo malo težje delati s sistemom Typo3, saj sem z učenjem in delom na projektu začel malo pozneje kot ostali. V tem času so mi zaposleni v podjetju ISA.IT nudili veliko pomoč. Kasneje je večji del opravil na projektu pripadel ravno meni. Rečem lahko, da je bilo zame delo na projektu zanimivo in mi je nudilo velik izziv. Naučil sem se jezikov kot sta PHP in SQL. Najbolj pri srcu med vsemi pa mi je ostal Typo3.

## Literatura

1. Werner Altmann, Rene Fritz, Daniel Hinderink, TYPO3 – Enterprise Content Managment, Packt Publishing Ltd, Birmingham, July 2005
2. Interno gradivo podjetja ISA.IT d.o.o.
3. Typo3.org Home (online). Dostopno na naslovu: <http://typo3.org>
4. TYPO3 CMS (online). Dostopno na naslovu: <http://www.typo3.com>
5. Vasont Systems. CRASH COURSE - content management (online). Dostopno na naslovu: <http://www.vasont.com/vasont/crashcourse/index.asp>
6. Typo3 slovenia (online). Dostopno na naslovu: <http://www.typo3-slovenia.org>