

UNIVERSITATEA „AUREL VLAICU” DIN ARAD
FACULTATEA DE ȘTIINȚE EXACTE
DOMENIUL DE LICENȚĂ: INFORMATICĂ
PROGRAMUL DE STUDIU: INFORMATICĂ
FORMA DE ÎNVĂȚĂMÂNT: ID

LUCRARE DE LICENȚĂ

ÎNDRUMĂTOR ȘTIINȚIFIC
Prof. univ. dr. Ioan Dzitac

ABSOLVENT
Șucan M. Mihai

ARAD
2011

UNIVERSITATEA „AUREL VLAICU” DIN ARAD
FACULTATEA DE ȘTIINȚE EXACTE
DOMENIUL DE LICENȚĂ: INFORMATICĂ
PROGRAMUL DE STUDIU: INFORMATICĂ
FORMA DE ÎNVĂȚĂMÂNT: ID

**PROIECTAREA UNUI
WEBSITE UNIVERSITAR**
Aplicație practică: www.uav.ro

ÎNDRUMĂTOR ȘTIINȚIFIC
Prof. univ. dr. Ioan Dzitac

ABSOLVENT
Șucan M. Mihai

ARAD
2011

UNIVERSITATEA „AUREL VLAICU” DIN ARAD
FACULTATEA DE ȘTIINȚE EXACTE
DOMENIUL / PROGRAMUL DE STUDIU
Nr. _____ din _____

APROBAT
DECAN

VIZAT
Îndrumător științific

DATE PERSONALE ALE CANDIDATULUI:

1. Date privind identitatea persoanei:

Numele: Șucan

Numele anterior: _____

Prenumele: Mihai

2. Sexul: M / F

3. Data și locul nașterii:

Ziua / luna / anul: ... / ... /

Locul (localitate, județ): ...

4. Prenumele părinților:

Tata: ...

Mama: ...

5. Domiciliul permanent (str., nr., localitate, județ, cod poștal, telefon, email):

...

6. Sunt absolvent(ă) promoția: iulie / 2011.

7. Forma de învățământ pe care am absolvit-o este:

cu frecvență, cu frecvență redusă, **ID**;

cu taxă, **fără taxă**.

8. Locul de muncă (dacă e cazul):

9. Solicit înscrierea la examenul de licență (licență, diplomă, disertație):
Sesiunea iulie, anul 2011.
10. Lucrarea de licență (licență, diplomă, disertație) pe care o susțin are următorul titlu:
PROIECTAREA UNUI WEBSITE UNIVERSITAR.
Aplicație practică: www.uav.ro.
11. Îndrumător științific:
Prof. univ. dr. Ioan Dzițac.
12. Menționez că susțin examenul de licență (pentru prima oară, a doua oară – după caz) pentru prima oară și declar pe propria-mi răspundere că am luat la cunoștință de prevederile art. 143 din Legea 1/2011. Declar că prezenta lucrare nu este realizată prin mijloace frauduloase, fiind conștient de faptul că, dacă se dovedește contrariul, diploma obținută prin fraudă îmi poate fi anulată, conform art. 146 din Legea 1/2011.

SEMNĂTURA

REFERAT
PRIVIND LUCRAREA DE LICENȚĂ
A
ABSOLVENTULUI MIHAI ȘUCAN
DOMENIUL: INFORMATICĂ
PROGRAMUL DE STUDIU: INFORMATICĂ
PROMOȚIA 2011

1. Titlul lucrării:

Proiectarea unui website universitar. Aplicație practică: www.uav.ro.

2. Structura lucrării:

Lucrarea este structurată logic și corect, conține o introducere, trei capitole: Prezentarea siteului; Modul de funcționare a siteului; Extensiile aplicației web, concluzii, bibliografie, index și două anexe (lista de figuri și baza de date MySQL).

3. Precizări asupra conținutului lucrării de licență, organizare logică, mod de abordare, complexitate, actualitate, deficiențe:

Aplicația www.uav.ro reprezintă un site bine realizat la nivel vizual, cu o interfață potrivită cu conținutul academic specific. Interfața este accesibilă și dinamică, facilitând schimbările ulterioare la nivel de conținut și funcționalități.

Autorii aplicației www.uav.ro sunt Marius Șucan și Mihai Șucan, ambii fiind șefi de promoție la specializarea Informatică.

Autorul Mihai Șucan s-a ocupat cu preponderență de partea de programare.

Ca etapă premergătoare proiectării aplicației, cei doi autori au studiat mai multe website-uri ale unor universități renumite din lume și din România: Stanford University Stanford, California, SUA; Carnegie Mellon University Pittsburg, Pennsylvania, SUA, University of Oxford Oxford, Anglia, Marea Britanie; MIT, Rice University, Princeton; Universitatea Politehnică din București, Universitatea „Alexandru Ioan Cuza” din Iași etc. Studiul a avut ca scop cunoașterea site-urilor academice: ce tipuri de layout-uri se folosesc, cum sunt organizate și structurate paginile, și informațiile etc. Analiza s-a făcut din punct de vedere vizual și tehnic, prin prisma conținutului și a structurii.

Websiteul www.uav.ro cuprinde bloguri, galerii de imagini, modul specializat pentru citit sau-și descărcat fișiere un formular de contact flexibil, fluxuri de știri Atom, un motor de căutare precum și o hartă a

siteului. Pentru administratori siteul are un gestionar de fișiere și un gestionar de conturi și drepturi, precum și editoarele specifice fiecărui modul sau secțiuni din site.

În realizarea interfeței au fost parcurse următoarele etape: planificare, efectuarea designului, colorarea acestuia, popularea în final, implementarea interfeței la nivel tehnic, utilizând HTML, CSS și alte tehnologii web.

În final a rezultat o aplicație profesională, care a adus un spor de imagine Universității „Aurel Vlaicu” și a determinat senatul universitar să acorde autorilor câte o diplomă de excelență prin hotărârea din 29 septembrie 2010.

Nici lucrarea și nici aplicația practică nu au cusururi, totuși autorii, fiind niște perfecționiști, continuă să-i aducă optimizări aplicației și în prezent.

4. **Aprecieri asupra lucrării (se va menționa: numărul titlurilor bibliografice consultate, frecvența notelor de subsol, calitatea și actualitatea surselor consultate; modul în care absolventul a prelucrat informațiile din sursele bibliografice, contribuții originale):**

Lucrarea este foarte bine documentată, conține 27 de titluri bibliografice, de cea mai mare actualitate. Aplicația practică este originală în totalitate, excelent realizată, pagina web www.uav.ro fiind funcțională și activă.

Consider că www.uav.ro este unul din cele mai profesionale websiteuri universitare din România la ora actuală.

5. **Concluzii (valoarea lucrării elaborate de absolvent, relevanța studiului întreprins, competențele absolventului, consecvența și seriozitatea de care a dat dovadă absolventul pe parcursul documentării și elaborării lucrării):**

Munca depusă a fost perseverentă, consecventă și eficientă. Lucrarea are atât o valoare teoretică, cât și practică, care poate fi valorificată de proiectanții de websiteuri instituționale.

Absolventul dovedește reale calități de programator profesionist de aplicații web.

6. **Redactarea lucrării respectă normele de redactare.**

Metodologia în vigoare a fost respectată, în plus, se poate observa că autorul are un stil de exprimare și redactare de nivel profesionist.

7. Nu există suspiciuni de realizare prin fraudă a prezentei lucrări.

„Realizarea site-ului Universității «Aurel Vlaicu» din Arad se bazează pe mulți ani de studii pe cont propriu și de muncă în grafică și design. Acest proiect are la bază un motor de site la care s-a început lucrul încă din anii de liceu și este realizat integral de Marius și Mihai Șucan.”

Acesta este citat din concluziile lucrării lui Marius Șucan, care exprimă un adevăr frumos și o pildă de urmat. De menționat că Marius Șucan este fratele geamăn a lui Mihai Șucan, care s-a ocupat de partea de design a aplicației.

8. Consider că lucrarea îndeplinește condițiile pentru susținere în sesiunea de Examen de licență din iulie 2011.

Propun comisiei ca absolventul Mihai Șucan, autorul acestei lucrări și coautorul aplicației www.uav.ro, să fie notat cu 10 (zece) la examenul de licență.

Arad,
Data 20 mai 2011

Îndrumător științific
Prof. univ. dr. Ioan Dzițac

Cuprins

Introducere	3
1 Prezentarea siteului	5
1.1 Structura siteului pentru utilizatori	6
1.2 Funcționalități pentru utilizatori	8
1.3 Funcționalități pentru administratori	9
1.4 Compatibilitatea siteului	10
2 Modul de funcționare al siteului	11
2.1 Tehnologiile server-side și client-side folosite de aplicație	11
2.2 Cerințele tehnice pentru server	13
2.3 Privire pe ansamblu	13
2.3.1 Prezentarea modulelor din site	15
2.4 Structura siteului pe server, fișierele folosite	16
3 Extensiile aplicației web	26
3.1 Conturi pentru administratori	26
3.1.1 Autentificare	29
3.1.1.1 Formularul de autentificare	29
3.1.1.2 Formularul de resetare a parolei	38
3.1.2 Gestionarea conturilor de administrare site	40
3.1.2.1 Lista conturilor	41
3.1.2.2 Verificarea drepturilor unui cont	43
3.1.2.3 Adăugarea de utilizatori	46
3.1.2.4 Editarea drepturilor unui cont	50
3.1.2.5 Securitatea sistemului de autentificare	55
3.1.3 Editare profil	56

3.2	Gestionarea fişierelor de pe server	57
3.3	Gestionarea paginilor	61
3.3.1	Structura siteului în format XML	62
3.3.2	Configuraţia modulului	66
3.3.3	Adăugarea de pagini	71
3.3.4	Setări pagină	74
3.4	Editorul de pagini Awebitor	75
3.4.1	Curăţarea codului HTML cu ReTidy	79
3.5	Blogul	83
3.5.1	Funcţionalităţi pentru utilizatori	83
3.5.2	Funcţionalităţi pentru administratori	92
3.6	Galeria de imagini	96
3.7	Modulul „Fişiere de descărcat”	102
3.8	Prima pagină	106
3.9	Pagina Contact	107
3.10	Motorul de căutare	110
3.11	Harta siteului	113
	Concluzii	115
	Glosar	116
	Listă de figuri	119
	Bibliografie	121
A	Baza de date MySQL	124
A.1	blog	124
A.2	blog_tags	125
A.3	gallery	125
A.4	search_documents_en şi search_documents_ro	126
A.5	search_doc_index_en şi search_doc_index_ro	126
A.6	sitemap	127

Introducere

Lucrarea de față prezintă proiectul de licență uav.ro începând cu aspectele generale, pentru a continua mai apoi cu detaliile tehnice despre implementarea și dezvoltarea proiectului.

Scopul proiectului prezentat în această lucrare, stabilit în colaborare cu domnul profesor coordonator [Ioan Dzitac](http://dzitac.rdsor.ro)¹, a fost realizarea unui site nou pentru Universitatea „Aurel Vlaicu” din Arad. Cerințele noului site au fost definite astfel:

- un design nou, mai atrăgător și modern, potrivit cu profilul siteului,
- navigare ușoară și accesibilă pentru diferite tipuri de utilizatori,
- conținut bilingv, în limbile română și engleză,
- încărcarea rapidă a paginilor,
- tehnic compatibil cu majoritatea navigatoarelor web, pe majoritatea sistemelor de operare în uz,
- structură tehnică solidă, capabilă să evolueze în timp,
- bine structurat din perspectiva informațiilor oferite vizitatorilor, clar și concis,
- informativ și util pentru profesorii și studenții care vor să aplice la universitate, să profeseze sau să studieze,
- ușor de menținut și de actualizat pentru cei responsabili de conținut.

În plus față de cerințele menționate mai sus, a fost pusă problemă evoluției siteului în comparație cu versiunea precedentă. Practic s-a dorit ca noul site să crească, să fie mai bogat în conținut, mai informativ, mai bine structurat

¹<http://dzitac.rdsor.ro>

și mai atrăgător vizual. De asemenea s-a dorit ca în comparație cu celelalte siteuri de universități românești, noul site UAV să fie unul care face dreptate nivelului universității, un site care pune în valoare principiile, valorile și misiunea a universității.

Pentru realizarea acestui proiect ambițios am lucrat în colaborare cu [Marius Șucan](#)². Împreună cu el și cu domnul profesor Dzițac au fost stabilite criteriile de calitate și de performanță ale siteului, structura siteului și necesitățile tehnice și funcționale ale acestuia. Marius Șucan s-a ocupat de partea de design a proiectului, determinând cele mai potrivite moduri de exprimare vizuală, construind o identitate vizuală specifică UAV. Eu m-am ocupat de partea tehnică, implementând platforma aplicației web. Domnul profesor a participat oferindu-ne ghidare importantă în realizarea structurii siteului, în definirea tipului de conținut ce va fi introdus după finalizarea siteului. De asemenea, domnul profesor s-a ocupat de completarea siteului cu conținut.

²<http://www.robodesign.ro/marius>

Capitolul 1

Prezentarea siteului

Noul site al Universității „Aurel Vlaicu” din Arad este un site modern, cu un aspect plăcut, profesional, și avansat din punct de vedere tehnic.

Figura 1.1: Prima pagină a noului site uav.ro, în limba română.



Siteul include bloguri, galerii de imagini, modul specializat pentru fișiere de descărcat, un formular de contact flexibil, fluxuri de știri **Atom**, un motor de căutare și o hartă a siteului. Pentru administratori siteul are un gestionar de fișiere și un gestionar de conturi și drepturi, pe lângă editoarele specifice fiecărei secțiuni din site.

Nu în ultimul rând, platforma siteului este flexibilă permițând adăugarea de funcționalități noi în site, fără limitări tehnice.

1.1 Structura siteului pentru utilizatori

Siteul Universității „Aurel Vlaicu” din Arad are câteva secțiuni importante, pentru diferitele domenii de interes ale vizitatorilor. Aceste secțiuni sunt:

Academic secțiunea care conține paginile despre structura anului universitar în curs, despre treptele de studii universitare (licență, masterat și doctorat), și despre concursurile pentru posturile didactice vacante.

Cercetare secțiunea care conține informațiile necesare pentru profesorii și studenții interesați de cercetare în cadrul UAV.

Departamente secțiunea cu informații specifice pentru fiecare departament al UAV.

Facultăți locul unde fiecare facultate UAV are câte un set de sub-pagini care oferă informații pentru studenți: orare, datele de examinare, lista de profesori pe discipline, și altele.

Personal locul unde sunt publicate pagini de prezentare a profesorilor universității.

Resurse locul unde se găsesc informații despre editura UAV, biblioteci, platforma *e-learning*, și direcția resurse umane – salarizare.

Reviste locul unde sunt prezentate revistele și restul de publicații ale universității.

Studenți este secțiunea specifică pentru tinerii care doresc să studieze la UAV. Aici se găsesc informații despre taxe, burse, olimpiade, concursuri, regulamente, diplome, acte necesare, organizații studențești, cămine și altele.

Universitate este secțiunea unde se găsesc informații despre universitate, istoricul acesteia, o prezentare, documente de interes public (licitații, declarații de avere, hotărâri ale senatului), noutăți și evenimente.

Structura acestui site se bazează pe un studiu asupra câtorva siteuri de universități străine și românești. În colaborare cu domnul profesor coordonator am stabilit structura asigurându-ne că ea va face navigarea siteului mult mai ușoară pentru utilizatori. Amintim câteva siteuri studiate:

- [Universitatea de Vest din Timișoara¹](#),
- [Universitatea Politehnică București²](#),
- [Universitatea „Politehnică” din Timișoara³](#),
- [Universitatea „Alexandru Ioan Cuza” din Iași⁴](#),
- [Universitatea de Vest „Vasile Goldiș” din Arad⁵](#),
- [Rice University⁶](#),
- [Stanford University⁷](#),
- [Carnegie Mellon University⁸](#),
- [Harvard University⁹](#),
- [University of Oxford¹⁰](#).

¹<http://www.uvt.ro>

²<http://www.upb.ro>

³<http://www.upt.ro>

⁴<http://www.uaic.ro>

⁵<http://www.uvvg.ro>

⁶<http://www.rice.edu>

⁷<http://www.stanford.edu>

⁸<http://www.cmu.edu>

⁹<http://www.harvard.edu>

¹⁰<http://www.ox.ac.uk>

1.2 Funcționalități pentru utilizatori

Pentru utilizatori contează în primul rând ușurința de navigare și ușurința cu care ei găsesc informațiile dorite. Pentru a ajuta în această direcție, siteul este bilingv, în română și engleză, și oferă posibilitatea de căutare în conținut. În plus, o hartă completă a siteului este disponibilă.

Siteul dispune de următoarele module:

- Blog,
- Galerie de imagini,
- Fișiere de descărcat,
- Contact,
- Hartă site,
- Căutare.

Pe prima pagină este o diaporamă în care poze cu locațiile UAV sunt afișate. De asemenea sunt listate ultimele articole din blogurile Noutăți și Evenimente. Nu în ultimul rând sunt afișate și cele mai importante pagini din site.

Blogul permite adăugarea de articole sortate cronologic, pe categorii. Acesta este folosit pentru pagini de genul „noutăți” sau „evenimente”. Fiecare articol poate avea etichete atașate (*tags*), pentru a permite o organizare a articolelor mai bună, și pentru a face navigarea mai ușoară pentru utilizatori. În plus, fiecare blog include un *feed*/flux **Atom** la care se pot abona utilizatorii care doresc să țină pas cu noutățile din site. Fluxul Atom poate fi limitat și numai pe anumite etichete. Arhiva de articole din fiecare blog permite navigarea după data publicării.

Modulul de galerie de imagini include și el suport pentru fluxuri Atom, dacă utilizatorul dorește să urmărească un anumit album de poze.

Pagina de contact permite utilizatorilor să contacteze diferitele departamente ale universității. Formularul este protejat anti-spam cu serviciul online [recaptcha.net](https://www.recaptcha.net).

La fiecare pagină, utilizatorul poate alege să mărească sau să micșoreze dimensiunea textului afișat. Articolele pot fi imprimate excluzând elementele din site care nu sunt relevante – există un **CSS** specific pentru imprimare care aranjează paginile să arate bine pe foaie.

Siteul poate fi navigat pe telefoane mobile cu o interfață adecvată.

Legăturile permanente (de forma /pagina/sub-pagina/ în loc de `index.php?parametrii_care_nu_sunt_intelese_de_nimeni`), permit memorarea mai ușoară a adreselor, permit motoarelor de căutare să indexeze mai bine conținutul, și în general dau o notă importantă de profesionalism siteului.

Exemple de pagini:

- Blog Noutăți: <http://www.uav.ro/ro/universitate/noutati>,
- Galerie foto: <http://www.uav.ro/ro/universitate/galerie-foto>,
- Contact: <http://www.uav.ro/ro/contact>,
- Hartă site: <http://www.uav.ro/ro/sitemap>.

1.3 Funcționalități pentru administratori

Administratorii siteului pot intra pe:

- <http://www.uav.ro/ro/admin> pentru a edita paginile în limba română,
- <http://www.uav.ro/en/admin> pentru a edita paginile în limba engleză.

Siteul dispune de un sistem de autentificare utilizatori, care include posibilitatea de a defini drepturile fiecăruia în cel mai mic detaliu. Mai multe detalii tehnice se găsesc în secțiunea 3.1.

Dacă utilizatorul își uită parola, acesta poate să își reseteze parola. El/ea va primi parola nouă prin email la adresa ce o are configurată în profilul contului său.

După autentificare, administratorul poate accesa paginile și secțiunile de administrare ale siteului conform drepturilor ce i-au fost acordate. De asemenea, el își poate modifica profilul: numele, adresa de email și parola.

Pe lângă editarea paginilor, administratorii pot modifica bloguri, galerii de imagini și pot să încarce fișiere pe server. Un modul specializat pentru gestionarea fișierelor permite adăugarea de directoare, fișiere, redenumire, ștergere și alte operații.

Administratorii care au drepturile necesare pot să adauge sau să șteargă utilizatori și pot să le editeze drepturile.

Pentru a asigura calitatea conținutului adăugat de către administratori, am inclus și o pagină cu recomandări pentru redactarea conținutului.

1.4 Compatibilitatea siteului

Siteul este compatibil cu:

- [Mozilla Firefox](#)¹¹,
- [Opera](#)¹²,
- [Google Chrome](#)¹³,
- [Apple Safari](#)¹⁴,
- [Konqueror](#)¹⁵
- și [Microsoft Internet Explorer](#)¹⁶.

În afara celor mai moderne navigatoare web, se pot folosi și navigatoare care nu implementează **JavaScript** și **CSS**, sau navigatoare care afișează doar text. Siteul este accesibil, putând fi folosit și de cei care au sisteme Braille de navigare web sau sintetizatoare vocale.

Accesibilitatea siteului nu se limitează doar la cele de mai sus. Siteul include comenzi din tastatură către cele mai importante pagini, meniuri, submeniuri, articole blog, imagini de galerie și alte secțiuni. Întregul site poate fi navigat și numai din tastatură, fiind astfel accesibil și pentru cei cu handicap motor.

Mai mult decât atât, două butoane au fost incluse în site pentru cei care au probleme cu vederea. Acestea permit utilizatorilor să controleze dimensiunea textului în pagină – textul poate fi mărit sau micșorat.

Nu în ultimul rând, este inclus și un al doilea design specializat pentru telefoane mobile, cu CSS, și un al treilea design, pentru imprimare. Designul pentru telefoane mobile este optimizat pentru ecranele mici, cu lățime foarte redusă. Designul pentru imprimare are în vedere aspectul non-interactiv al siteului, atunci când acesta se găsește pe foaie – adică rămân vizibile doar elementele importante într-o pagină: titlul, articolul și alte elemente relevante, în funcție de pagină.

Toate aceste aspecte de compatibilitate și accesibilitate contribuie foarte mult la definirea profesionalismului acestui site de universitate.

¹¹<http://www.mozilla.com>

¹²<http://www.opera.com>

¹³<http://www.google.com/chrome>

¹⁴<http://www.apple.com/safari>

¹⁵<http://www.konqueror.org>

¹⁶<http://www.microsoft.com/windows/internet-explorer/>

Capitolul 2

Modul de funcționare al siteului

Pentru a explica modul de funcționare al siteului vom începe prin a enumera tehnologiile folosite de acesta și cerințele software pentru serverul web. În continuare vom descrie modul în care paginile siteului sunt generate dinamic de către scripturile PHP. Capitolul se încheie cu o secțiune în care sunt descrise toate fișierele sursă folosite de site.

2.1 Tehnologiile server-side și client-side folosite de aplicație

Siteul folosește următoarele tehnologii:

- **HTML** accesibil, structurat semantic, fără tabele pentru *layout*. Codul este optimizat pentru a fi ușor de indexat de către motoarele de căutare, pentru a obține rezultate mai bune la căutări.
- **CSS** pentru toată interfața siteului.
- **JavaScript** pentru elementele interactive din site, de exemplu pentru harta siteului.
- **PHP** pentru generarea dinamică a tuturor paginilor, pe server. Toate documentele HTML sunt generate dinamic atunci când utilizatorul accesează siteul uav.ro.

- **MySQL** pentru baze de date. Atunci când codul PHP generează paginile HTML, baza de date MySQL este folosită pentru popularea paginilor cu informații.
- **XML** este folosit pentru menținerea structurii siteului. Scripturile PHP citesc anumite fișiere XML pentru a determina ierarhia paginilor, titlul, tipul de pagini și alte informații.
- Fișierele de tip **JSON** sunt și ele folosite pentru stocarea de date despre utilizatorii modulului de administrare al siteului, pentru stocarea adreselor de email din [pagina Contact](http://www.uav.ro/ro/contact)¹ și pentru alte date.
- Fluxuri **Atom** pentru paginile de tip blog și galeriile de imagini. Aceste pagini sunt disponibile și în format Atom, format care poate fi folosit de utilizatori pentru a fi informați automat când sunt publicate articole/imagini noi în site.
- Sitemaps XML permite motoarelor de căutare să verifice ultimele modificări din site [1]. Toată harta siteului este generată dinamic în format HTML și în format Sitemaps XML.

Pe lângă tehnologiile menționate mai sus, siteul folosește în mod general **DOM** pe server cât și în navigatorul web. Pagina HTML este generată pe server într-un arbore DOM care la final este serializat în formatul dorit: HTML sau **XHTML**. Codul JavaScript care se execută în navigatorul web folosește și el structura DOM pentru a modifica pagina în funcție de interacțiunea cu utilizatorul.

Awebitor este editorul de articole folosit în modulul de administrare. Acesta utilizează **XMLHttpRequest** [2] pentru încărcarea dinamică de conținut și **designMode** [3] pentru capabilitatea de editare *What You See Is What You Get* (WYSIWYG).

Noul HTML5 introduce o multitudine de taguri și funcționalități noi [4]. Siteul UAV folosește capabilități HTML5 în navigatoarele web moderne. De exemplu, pentru câmpurile de tip email codul folosește noul atribut **type=email**.

¹<http://www.uav.ro/ro/contact>

2.2 Cerințele tehnice pentru server

Aplicația web [uav.ro](http://www.uav.ro) are următoarele cerințe tehnice pentru server:

- **PHP** 5.2. Pentru curățarea codului HTML la salvarea paginilor în modulul de administrare este necesar să fie instalat modulul `php5-tidy`.
- **Apache** 2. Pentru a permite utilizarea legăturilor permanente este necesară activarea fișierelor de tip `.htaccess` și a modulului Apache `mod_rewrite`.
- **MySQL** 4.1.

De obicei siteul este instalat pe sisteme de operare bazate pe Unix, unde aplicațiile menționate rulează cel mai bine.

2.3 Privire pe ansamblu

Fiecare accesare de pagină trece prin fișierul `index.php` al siteului. Acest script **PHP** face chemări la restul de fișiere în funcție de adresa aleasă de utilizator.

Vom începe cu un exemplu de accesare a paginii de [prezentare a Universității „Aurel Vlaicu” din Arad](http://www.uav.ro/ro/universitate/prezentarea-universitatii)². Cererea este procesată astfel:

1. Fișierul `.htaccess` dă instrucțiuni ca **Apache** să rescrie adresa sub forma următoare:
`index.php?path2=/ro/universitate/prezentarea-universitatii`.
2. Fișierul `index.php` este executat iar acesta preia calea dorită de utilizator din variabila `$_REQUEST['path2']`.
3. Fișierul `settings.php` este inclus. Acest fișier are doar variabile de configurare pentru întregul site.
4. Fișierul de traducere al siteului este inclus, fie `lang_en.php`, fie `lang_ro.php`, în funcție de setările lingvistice ale utilizatorului. Aceste fișiere conțin toate mesajele afișate de site în limba engleză și română. Setările lingvistice sunt salvate în navigatorul web al utilizatorului sub forma de **cookie**.

²<http://www.uav.ro/ro/universitate/prezentarea-universitatii>

5. Se caută în directorul `cache/pages/` dacă pagina este deja generată și salvată pe disc, pentru a se evita regenerarea acesteia. În acest caz, fișierul **HTML** este citit, trimis direct la navigatorul web și execuția se încheie.

Pentru fiecare cerere se generează un jeton unic pe baza parametrilor din `$_GET` – a se vedea funcția `gen_page_cache_id()` din `index.php`.
6. Dacă nu există pagina în `cache`, atunci este inclus `layout.php`. Acest fișier generează șablonul HTML, scheletul în care meniurile și conținutul paginii vor fi afișate. Din acest moment începe generarea paginii HTML.
7. Se creează un obiect `DOMDocument()` care să țină structura ierarhică a paginii, a interfeței. În acest **DOM** se încarcă interfața dată de `layout.php` în pasul anterior.
8. Se citește fișierul `xml/site_en.xml` sau `xml/site_ro.xml`. Acest fișier conține structura siteului, într-un format **XML** propriu (vezi secțiunea 3.3.1).
9. Se generează meniurile și restul interfeței, în funcție de calea accesată de utilizator. Datele sunt preluate din structura siteului, din fișierul XML citit în pasul anterior.
10. Se caută calea dorită de utilizator în structura siteului. Pentru exemplul dat mai sus, prezentarea universității este o pagină de tip articol.
11. Fișierul XML indică titlul paginii și fișierul HTML de pe disc care trebuie afișat: `pages/ro_universitate-prezentarea-universitatii.html`.
12. În final, toată pagina este serializată în format HTML. Aceasta conține interfața siteului împreună cu conținutul paginii accesate de utilizator. Serializarea se face folosindu-se obiectul `DOMDocument()`.
13. Rezultatul, pagina completă HTML, este scris pe disc în `cache/pages/` pentru a fi reutilizat pe viitor, și apoi este trimis către client (utilizator). Numele fișierului salvat pe disc este rezultatul funcției de fărâmițare `md5()` aplicată pe șirul de caractere dat de calea accesată de utilizator, împreună cu toate variabilele din cererea **HTTP** GET.

Dacă utilizatorul accesează o pagină de tip blog, sau alt modul special, atunci diferența în execuție este minoră. Practic, în loc să fie citit un fișier HTML, în structura XML este notat modulul care trebuie inclus. Acel fișier

PHP este inclus, iar din acel moment este responsabil să genereze conținutul siteului. Pentru mai multe detalii se poate vedea secțiunea 3.3.1.

În cazul paginilor de tip blog sau galerie de imagini, există noțiunea de sub-directoare în calea aleasă. De exemplu, blogul Noutăți are adresa `/ro/universitate/noutati/`, loc unde se găsesc articole. Codul care generează paginile blogului verifică întreaga cale accesată de utilizator pentru a determina articolul dorit, de exemplu `/ro/universitate/noutati/invitatie-la-sport`.

Secțiunea de administrare a siteului funcționează identic, dar în `admin/include.php` este schimbat fișierul XML care conține structura siteului. Adică, nu mai sunt încărcate fișierele `xml/site_en.xml` și `xml/site_ro.xml`. Sunt folosite XML-urile `admin_en.xml` și `admin_ro.xml`. În acestea se găsește structura siteului de administrare. Astfel, este refolosită interfața siteului principal, pentru partea de administrare – diferă doar meniurile și paginile care pot fi accesate.

2.3.1 Prezentarea modulelor din site

Paginile de tip blog sunt generate de `pages/blog.php`. Pe prima pagină a blogului sunt afișate cele mai recente articole. Acestea sunt preluate din tabelul **MySQL** `blog` (vezi anexa A.1). Articolele sunt asociate cu calea accesată de utilizator. Dacă utilizatorul alege să încarce un anume articol, atunci doar acela este afișat. Dacă se accesează fluxul **Atom** al paginii, atunci fișierul `pages/blog-atom.php` este inclus. Acest fișier **PHP** afișează în format Atom ultimele articole adăugate în pagină.

Blogurile mai au tabelul `blog.tags` (vezi anexa A.2) care asociază articole cu etichete (*tags*). Etichetele pot fi considerate drept categorii liber alese de administratorii siteului, atunci când aceștia adaugă articolele. Utilizatorii pot filtra articolele pe baza acestor etichete, putând să citească doar ce îi interesează.

Pentru a face paginile blog mai flexibile, există posibilitatea ca administratorul să editeze antetul / mesajul afișat pe prima pagină a blogului. De asemenea, se poate edita și panoul lateral – loc unde pot fi adăugate legături noi sau alte informații.

Stilarea paginilor de tip blog este în fișierul `css/blog.css`. Conținutul **HTML** al articolelor este salvat în directorul `pages/blog/`.

În secțiunea de administrare se pot adăuga/edita/șterge articolele din bloguri, folosindu-se scriptul `admin/blog.php`. Pentru mai multe detalii despre bloguri vezi secțiunea 3.5.

Modulul care permite adăugarea de galerii de imagini este aproape identic cu cel pentru bloguri. Diferențele sunt mici: imaginile nu au etichete, iar paginile de galerie nu au panou lateral. De asemenea, scripturile PHP se ocupă și de redimensionarea automată a imaginilor atunci când acestea sunt adăugate pe server. Pentru mai multe detalii vezi secțiunea 3.6.

Pagina Contact, scriptul `pages/contact.php`, include un formular HTML ce permite utilizatorilor să trimită emailuri către destinația dorită. Destinațiile oferite sunt citite dintr-un fișier **JSON**, `pages/en_contact.json` sau `pages/ro_contact.json`. Utilizatorul trebuie să treacă și testul reCaptcha, pentru a nu permite trimiterea de mesaje spam. În dreapta formularului este afișat documentul HTML `pages/en_contact.html` sau `pages/ro_contact.html`.

Pe partea de administrare, `admin/contact.php`, se pot edita datele de contact ce apar în dreapta formularului, și se pot edita destinațiile din JSON. În plus, datele de contact afișate pe prima pagină pot fi modificate din același loc. Pentru detalii vezi secțiunea 3.9.

Paginile cu fișiere de descărcat, `pages/downloads.php`, includ și ele un antet ce poate fi editat de administratorii siteului, urmat de lista de fișiere din directorul asociat paginii, în `files/downloads/`. Pentru flexibilitate, administratorii pot edita titlul și descrierea fiecărui fișier. În plus, pot fi adăugate legături către fișiere externe. Toate aceste date sunt salvate într-un fișier JSON în directorul asociat paginii. Pentru detalii vezi secțiunea 3.7.

Orice modificare efectuată este înregistrată automat cu modulele de Search și Sitemap. Paginile sunt indexate pentru a putea fi căutate și pentru a fi afișate în harta siteului.

2.4 Structura siteului pe server, fișierele folosite

Lista de directoare și fișiere din siteul UAV este prezentată mai jos pentru a face mai clar modul de funcționare al siteului. O trecere în revistă a acestei structuri permite formarea unei imagini mai clare asupra locurilor unde se poate găsi codul responsabil pentru diferitele funcții ale siteului.

- `admin/`

Tot codul care face posibilă administrarea siteului.

- `awebitor-interface.php`

Scriptul **PHP** care generează interfața editorului de articole, în format **XHTML**. Vezi secțiunea 3.4.

- `blog.php`
Modulul de administrare bloguri. Vezi secțiunea 3.5.2.
- `check.php`
Script care verifică dacă utilizatorul este autentificat și are dreptul de acces la admin.
- `contact.php`
Modulul de administrare a [paginii de contact](http://www.uav.ro/ro/contact)³. Vezi secțiunea 3.9.
- `downloads.php`
Modulul de administrare a paginilor cu fișiere de descărcat. Vezi secțiunea 3.7.
- `edit.php`
Editorul de articole. Vezi secțiunea 3.4.
- `files.php`
Gestionarul de fișiere pe server. Vezi secțiunea 3.2.
- `files-edithtml.php`
Modul de editare pagini **HTML** în gestionarul de fișiere.
- `files-retidy.php`
Modul de curățare pagini HTML în gestionarul de fișiere.
- `gallery.php`
Modulul de administrare a galeriilor de imagini. Vezi secțiunea 3.6.
- `help-ro.html`
Pagina cu recomandări editoriale pentru administratorii siteului. În acest document sunt explicate principalele criterii de calitate în redactarea conținutului.
- `iframe-layout.php`
Interfața paginilor din admin afișate în `<iframe>`-uri. În speță, acest fișier *layout* este folosit când gestionarul de fișiere este inclus dinamic într-un `<iframe>` – de exemplu atunci când utilizatorul alege să adauge o imagine în editor.
- `include.php`
Scriptul inclus de toate paginile din admin. Acesta configurează diferite aspecte generale ale paginilor de administrare. Mai precis acest fișier:
 - * încarcă structura siteului de administrare din `xml/admin_en.xml` sau `xml/admin_ro.xml`, în funcție de limba curentă.

³<http://www.uav.ro/ro/contact>

- * verifică dacă utilizatorul are drepturi de administrare, folosindu-se de `check.php`. Altfel, accesul nu este permis.
- * oferă câteva funcții generale folosite de paginile de administrare.
- `index.php`
Prima pagină a siteului de administrare.
- `pages.php`
Gestionarul de pagini. Aici se poate edita structura siteului. Acest script încarcă `xml/site_en.xml` sau `xml/site_ro.xml` și îl afișează. Administratorul poate adăuga/edita/șterge pagini din structura siteului. Vezi secțiunea 3.3.
- `pages-blog.php`
Modulul care permite adăugarea și ștergerea paginilor de tip blog. Vezi secțiunea 3.5.2.
- `pages-downloads.php`
Modulul care permite adăugarea și ștergerea paginilor cu fișiere de descărcat. Vezi secțiunea 3.7.
- `pages-gallery.php`
Modulul care permite adăugarea și ștergerea paginilor de tip „galerie de imagini”. Vezi secțiunea 3.6.
- `permissions.php`
Editorul de drepturi pentru utilizatori. Vezi secțiunea 3.1.2.4.
- `profile.php`
Acest script permite utilizatorilor să își editeze profilul. Vezi secțiunea 3.1.3.
- `users.php`
Gestionarul de utilizatori. Vezi secțiunea 3.1.2.
- `cache/`
Cacheul de pagini HTML generate de site. Aceste fișiere sunt refolosite pentru a asigura încărcarea rapidă a siteului când sunt mulți utilizatori care navighează siteul simultan.
 - `blog/`
 - `gallery/`
 - `pages/`
 - `search/`

- **css/**

Stilurile folosite în diferite secțiuni ale siteului.

- **admin.css**
Stilul pentru paginile de administrare.
- **admin-iframe.css**
Stilul folosit de paginile de administrare încărcate în `<iframe>`-uri. În speță, acesta este folosit când `iframe-layout.php` este încărcat.
- **awebitor.css**
Stilul folosit pentru interfața editorului de articole. Vezi secțiunea 3.4.
- **awebitor_iframe.css**
Editorul introduce conținutul articolului într-un `<iframe>`. Acest fișier de stil este folosit pentru afișarea paginilor care sunt în curs de editare.
- **blog.css**
Stilul pentru paginile de tip blog. Vezi secțiunea 3.5.
- **contact.css**
Stilul pentru pagina Contact. Vezi secțiunea 3.9.
- **gallery.css**
Stilul pentru galeriile de imagini. Vezi secțiunea 3.6.
- **handheld.css**
Stilul pentru celulare.
- **interface.css**
Stilul principal al siteului.
- **interface_ie6.css**
Stilul pentru Microsoft Internet Explorer 6.
- **interface_ie7.css**
Stilul pentru Microsoft Internet Explorer 7.
- **print.css**
Stilul folosit pentru imprimare.
- **search.css**
Stilul pentru pagina care afișează rezultatele unei căutări în site. Vezi secțiunea 3.10.

- `sitemap.css`
Stilul pentru pagina unde este afișată [harta siteului](#)⁴. Vezi secțiunea [3.11](#).
- `welcome.css`
Stilul pentru [prima pagină](#)⁵. Vezi secțiunea [3.8](#).
- `files/`
Locul unde administratorii siteului pot încărca fișiere cu gestionarul de fișiere.
 - `downloads/`
Aici sunt salvate fișierele încărcate de administratori în paginile cu fișiere de descărcat. Vezi secțiunea [3.7](#).
 - `gallery/`
Aici sunt salvate imaginile din galerii. Vezi secțiunea [3.6](#).
 - * `normal/`
 - * `original/`
 - * `thumbnails/`
 - `page_templates/`
Aici se pot adăuga pagini HTML, șabloane ce pot fi refolosite când administratorii doresc să adauge pagini noi. Vezi secțiunea [3.3.3](#).
- `images/`
Imaginile folosite în interfața siteului.
 - `awebitor/`
Iconițele editorului de articole. Vezi secțiunea [3.4](#).
 - `headers/`
Imaginile folosite ca antet în site.
 - `welcome-headers/`
Imaginile folosite ca antet pe prima pagină. Vezi secțiunea [3.8](#).
- `js/`
Fișierele JavaScript folosite în site.
 - `admin.js`
Codul JS folosit în paginile de administrare a siteului.

⁴<http://www.uav.ro/ro/sitemap>

⁵<http://www.uav.ro/>

- `awebitor.js`
Editorul de articole. Vezi secțiunea 3.4.
- `awebitor_iframe.js`
Acest fișier este folosit în `<iframe>`-ul ce permite editarea paginilor în modulul de administrare al siteului.
- `ie_fixes.js`
Cod JS pentru compatibilitatea interfeței siteului cu navigatorul web Microsoft Internet Explorer. Acesta implementează afișarea meniurilor și repară alte erori.
- `index.js`
Codul folosit în toate paginile siteului UAV. Acesta permite utilizatorilor să mărească/micșoreze dimensiunea textului afișat în site și restul de funcționalități interactive ale paginilor.
- `login.js`
Codul folosit pentru autentificarea utilizatorilor în modulul de administrare. Vezi secțiunea 3.1.1.1.
- `sha512.js`
Cod *open source* care criptează parola înainte de autentificare folosind algoritmul SHA-512 [5].
- `sitemap.js`
Codul folosit pentru harta siteului. Vezi secțiunea 3.11.
- `welcome.js`
Codul folosit pe prima pagină pentru animarea pozelor. Vezi secțiunea 3.8.
- `modules/`
Scripturi refolosite prin diferite părți ale siteului.
 - `json/`
Script *open source* de citire și scriere fișiere **JSON** pentru versiunile mai vechi de PHP [6]. Noile versiuni de PHP implementează nativ aceste funcții [7].
 - `recaptcha/`
Script *open source* de pe [recaptcha.net](https://www.recaptcha.net). Acesta poate fi folosit pentru protejarea formularelor împotriva spamului.
 - `pingomatic.php`
Script propriu de generare pinguri la motoarele de căutare care sunt specializate pentru bloguri. Acesta folosește serviciul web pingomatic.com.

- **search.php**
Script propriu care indexează conținutul paginilor într-o bază de date **MySQL**. Acesta permite altor scripturi PHP să indexeze pagini noi în baza de date, să șteargă și să modifice pagini existente. De asemenea, scriptul poate fi folosit pentru efectuarea de căutare în baza de date. Vezi secțiunea 3.10.
- **sitemap.php**
Script propriu care indexează harta siteului. Acesta permite altor scripturi PHP să indexeze pagini în baza de date, să șteargă și să modifice pagini existente. Sunt incluse funcții ce generează pagina HTML pentru harta siteului și fișierul Sitemaps XML [1] pentru motoarele de căutare. Vezi secțiunea 3.11.
- **pages/**
Aici sunt salvate paginile HTML adăugate de administratorii siteului. Pe lângă fișierele HTML sunt și scripturile PHP care aparțin diferitelor tipuri de pagini.
 - **blog/**
Aici sunt salvate fișierele HTML ale articolelor adăugate în pagini de tip blog. Vezi secțiunea 3.5.
 - **downloads/**
Aici sunt salvate fișierele HTML generate pentru paginile care conțin fișiere de descărcat. Vezi secțiunea 3.7.
 - **gallery/**
Aici sunt salvate fișierele HTML pentru fiecare imagine din orice galerie. Vezi secțiunea 3.6.
 - **blog.php**
Scriptul PHP care generează paginile de tip blog. Vezi secțiunea 3.5.
 - **blog-atom.php**
Scriptul care generează fluxul **Atom** pentru bloguri.
 - **contact.php**
Pagina cu formularul de contact. Acest script încarcă fișierul **en_contact.html** sau **ro_contact.html** pentru a afișa datele de contact ale universității, lângă formular. Fișierul **en_contact.json** sau **ro_contact.json** este încărcat pentru afișarea departamentelor ce pot fi contactate din formular. Utilizatorul nu poate să vadă adresele de email ale departamentelor, dorindu-se a fi o măsură de protecție împotriva spamului. Vezi secțiunea 3.9.

- `downloads.php`
Paginile cu fișiere de descărcat sunt generate de acest script. Vezi secțiunea 3.7.
 - `en_contact.html` și `ro_contact.html`
Datele de contact afișate pe pagina Contact.
 - `en_contact.json` și `ro_contact.json`
Destinațiile afișate în formularul de contact.
 - `en_welcome_contact.html` și `ro_welcome_contact.html`
Datele de contact afișate pe prima pagină.
 - `en_welcome_partners.html` și `ro_welcome_partners.html`
Secțiunea Parteneri afișată pe prima pagină.
 - `gallery.php`
Paginile de tip „galerie de imagini” sunt afișate de acest script. Vezi secțiunea 3.6.
 - `gallery-atom.php`
Fluxul Atom pentru galeriile de imagini.
 - `search.php`
Pagina de căutare în site. Aceasta afișează rezultatele căutării. Modulul `modules/search.php` este încărcat pentru a efectua căutările dorite în baza de date. Vezi secțiunea 3.10.
 - `sitemap.php`
Pagina care afișează harta siteului. Acest script încarcă fișierul `sitemap_en.html` sau `sitemap_ro.html`, în funcție de limba folosită de utilizator. Vezi secțiunea 3.11.
 - `sitemap_en.html` și `sitemap_ro.html`
Harta siteului, în engleză și română.
 - `welcome.php`
Scriptul care generează dinamic prima pagină. Vezi secțiunea 3.8.
- `xml/`
Fișierele XML folosite pentru structura siteului.
 - `admin_en.xml` și `admin_ro.xml`
Structura paginilor de administrare a siteului.
 - `site_en.xml` și `site_ro.xml`
Structura siteului. Acest XML listează toate paginile din site, cu titlul fiecăreia, cu structura meniurilor, și definește tipul fiecărei pagini. Vezi secțiunea 3.3.1.

- `sitemap.xml`

Harta siteului în format Sitemaps XML [1], folosită de motoarele de căutare. Vezi secțiunea 3.11.

- `404.php`

Pagina afișată atunci când utilizatorul încearcă să acceseze o pagină inexistentă.

- `admin.php`

Pagina de autentificare în secțiunea de administrare a siteului. Aici administratorul poate să își reseteze parola dacă a uitat-o. Vezi secțiunea 3.1.

- `awebitor_iframe.php`

Editorul de pagini încarcă acest fișier în `<iframe>`. În acest fișier aflat la rădăcina siteului este introdus codul HTML al paginilor care urmează să fie editate de utilizator. Vezi secțiunea 3.4.

- `index.php`

Punctul de intrare în site. Acest script generează toate paginile. Acesta citește setările siteului, se conectează la baza de date MySQL, citește structura siteului din XML și în funcție de calea dorită de utilizator, afișează pagina asociată cu calea dată.

- `lang_en.php` și `lang_ro.php`

Fișierele de traducere a siteului în limbile engleză și română.

- `layout.php`

Scriptul care generează interfața siteului în format XHTML.

- `settings.php`

Setările siteului.

- `.htaccess`

Fișierul de configurare pentru Apache. În acesta se găsesc regulile de rescriere a cererilor HTTP.

- `.private/`

Module private folosite de site

- `users/`

Modul de autentificare utilizatori și gestionare permisiuni. Vezi secțiunea 3.1.

2.4. STRUCTURA SITEULUI PE SERVER, FIȘIERELE FOLOSITE 25

- `retidy/`

Modul de curățare pagini HTML. Vezi secțiunea [3.4.1](#).

Capitolul 3

Extensiile aplicației web

În secțiunea 2.3 s-a putut vedea că siteul este construit într-un mod flexibil ce permite adăugarea de pagini de orice fel. Nu se impune nici o limită asupra tipului de pagini – trebuie doar să fie încărcat un fișier **PHP** din care autorul scriptului poate face orice dorește. Aceste scripturi le numim module sau extensii ale aplicației web.

Implicit aplicația web este cât se poate de „zveltă”, de robustă, minimalistă, fără să aibă funcționalități ce nu sunt dorite. În acest capitol vom prezenta extensiile realizate pentru siteul Universității „Aurel Vlaicu” din Arad.

3.1 Conturi pentru administratori

Secțiunea de administrare a siteului poate fi accesată doar de către personalul autorizat. În acest scop a fost dezvoltat un sistem de conturi pentru administratorii siteului, cu autentificare securizată.

Modulul privat `.private/users/users.php` este scriptul **PHP** care conține implementarea noțiunii de conturi. Clasa `RD_users_manager` permite autentificarea, resetarea de parole, schimbarea profilului la fiecare utilizator, adăugarea și ștergerea de utilizatori, modificarea drepturilor acestora.

Parolele și drepturile utilizatorilor se găsesc în directorul `.private/users/sites/`. Parolele sunt criptate cu funcția de fărâmițare SHA-512 – acestea nu pot fi decriptate [8]. Pentru drepturile și profilul fiecărui utilizator este folosit câte un fișier **JSON**. Formatul fișierului este următorul:

```
{
  "user": "nume-cont" ,
```

```

"lastLoginDate": 0, // UNIX timestamp
"lastLoginIP": "127.0.0.1",
"nrLogins": 1,
"profile": {
  "displayName": "Nume Utilizator",
  "email": "adresa@example.com"
},
"permissions": {
  "admin": {
    "view": true,
    "recursive_": true
  },
  "admin\\files\\director": {
    "default_": true,
    "recursive_": true
  },
  "admin\\pages\\ro\\pagina": {
    "edit": true,
    "recursive_": true
  }
}
}

```

După cum se poate vedea în codul de mai sus, fișierul JSON include numele contului, numele afișat, emailul, numărul de autentificări, data ultimei autentificări și IP-ul.

Drepturile unui cont sunt exprimate ierarhic, ca într-o structură de directoare, un arbore. Avem utilizatorul care poate efectua un număr nelimitat de acțiuni, în funcție de pagină, în funcție de secțiunea siteului. Avem secțiunile siteului structurate ca pagini și sub-pagini cu nume care au fiecare o cale. Pentru a permite editarea paginii contact trebuie acordat dreptul:

```

"admin\\pages\\ro\\contact": {
  "edit": true,
  "alta_actiune": false
  // orice altă acțiune urmată de true dacă este permisă
  // sau false dacă nu.
}

```

Dacă se doresc aceleași drepturi pentru sub-pagini, atunci se adaugă `recursive_`:

```
"admin\pages\ro\contact": {  
  "edit": true,  
  "recursive_": true  
}
```

Acum pentru `/ro/contact/orice-sub-pagina` se aplică aceleași drepturi.

Dacă nu se dorește specificarea fiecărei acțiuni permise/nepermise, atunci se poate folosi `default_`:

```
"admin\pages\ro\contact": {  
  // interzicem toate acțiunile posibile în afară de view  
  "default_": false,  
  "view": true  
}
```

Dacă profilul unui utilizator nu definește nici un fel de permisiuni, implicit orice acțiune este interzisă. Dreptul pentru acțiunile de vizualizare sau de editare în diferitele secțiuni ale siteului trebuie acordate explicit fiecărui utilizator. Pentru mai multe detalii despre interfața de editare a drepturilor vedeți secțiunea [3.1.2.4](#).

3.1.1 Autentificare

Pentru autentificare administratorul accesează una din paginile de mai jos:

- Română: <http://www.uav.ro/ro/admin>,
- Engleză: <http://www.uav.ro/en/admin>.

Figura 3.1: Formularul de autentificare.

3.1.1.1 Formularul de autentificare

Formularul de autentificare folosește **JavaScript** pentru a cripta parola înainte de a fi trimisă la server. Parola nu este nici o dată transmisă *plain text* – exact cum o scrie utilizatorul. Ea este criptată cu funcția de fărâmițare SHA-512 [8].

Scriptul `admin.php` generează codul **HTML** al formularului astfel:

```
<script src="/js/login.js"
      type="text/javascript"></script>
<script src="/js/sha512.js"
      type="text/javascript"></script>
<p><em>Autentificare</em></p>
<form method='post' action='?'
      onsubmit='return adminFormLogin(this) '>
<p><label>Cont:
      <input type='text' name='user' </label></p>
<p><label>Parola:
```

```

    <input type='password' name='pass'>/label</p>
</p>
<input type='hidden' name='l' value='ro'>
<input type='hidden' name='login' value='n'>
<input type='hidden' name='tok' value='$token'>
<input type='submit' value='Autentificare'>
<a href='?forgot=1'>Am uitat parola!</a>
</p>
</form>

```

După cum se poate vedea din cod:

- Sunt incluse două fișiere JavaScript: `login.js` și `sha512.js`. Primul conține funcțiile folosite de formularul de autentificare – de exemplu `adminFormLogin()`. Al doilea fișier este o implementare *open source* a algoritmului SHA-512 [5].
- Câmpul cont are numele `user`.
- Câmpul parolă are numele `pass`.
- Câmpul ascuns `l` are valoarea limbii curente: `ro` sau `en`.
- Câmpul ascuns `login` are valoarea `n`. Dacă serverul nu primește valoarea `y` nu continuă procesarea cererii de autentificare. Această măsură de protecție minimă este inclusă pentru a verifica dacă JavaScriptul a fost executat. Funcția `adminFormLogin()` schimbă valoarea acestui câmp să fie `y`.
- Câmpul ascuns `tok` este un jeton (*token*), un șir de caractere generate aleatoriu de server. Acest jeton este folosit de funcția `adminFormLogin()` pentru criptarea parolei. Serverul așteaptă ca parola primită să fie criptată cu acest șir, dacă nu autentificarea eșuează. Șirul se schimbă la fiecare încărcare de pagină și include în el câteva măsuri de siguranță ce vor fi discutate în această secțiune.

Pagina de autentificare este generată de `admin.php`. Acest script **PHP** include modulul privat de gestionare a conturilor de utilizatori: `.private/users/users.php`.

```

// Includerea modulului privat. Calea este dată de variabila
// globală $GLOBALS['users.php'] definită în settings.php.
require_once($GLOBALS['users.php']);

```

```
// Se crează o instanță a obiectului cu care putem gestiona
// utilizatorii.
$users_manager = new RD_users_manager();
```

```
if ($users_manager->did_auth()) {
    // Dacă utilizatorul este autentificat deja, facem
    // redirectionarea către prima pagină a siteului de
    // administrare.
    header('Location: /admin/index.php');
    exit;
}
```

Variabila `$token` pentru formular este generată astfel:

```
$token =
    htmlentities($users_manager->get_auth_token(),
        ENT_QUOTES, 'utf-8');
```

Funcția `get_auth_token()` generează un jeton aleatoriu ce este folosit pentru autentificare.

Atunci când utilizatorul alege să se autentifice se face următoarea verificare:

```
if (isset($_POST['login']) && $_POST['login'] == 'y') {
    $auth_failed =
        !$users_manager->login($_POST['user'],
                                $_POST['pass']);

    if (!$auth_failed) {
        header('Location: /admin/index.php');
        exit;
    }
}
```

Funcția `login()` face verificările necesare. Practic fișierul `admin.php` este doar interfața, iar `users.php` se ocupă de implementarea autentificării, de măsurile de securitate.

Un exemplu de `$token` generat: `BaoYC5>,8/<2=3A'1036<-61: ....`

În fișierul `users.php` implementarea funcției `get_auth_token()` arată în felul următor:

```
public function get_auth_token () {
    // Siteul are un jeton privat care a fost generat aleatoriu la
    // instalarea siteului. Acesta este folosit pentru criptarea
    // parolei pe server.
    $token_site = $this->get_site_token();
    if (!$token_site) {
        return null;
    }

    // Generăm un jeton aleator pentru sesiunea curentă.
    $this->gen_session_token();

    // Generăm două valori aleatoare unice pentru cererea
    // curentă de autentificare.
    $token_rnd1 = uniqid(' ', true) . time();
    $token_rnd2 = uniqid(' ', true);
    $token_auth = hash_hmac('sha512',
                            $this->session_token .
                            $token_rnd1, $token_rnd2);

    // Combinăm cele două jetoane într-un jeton.
    $prev = mt_rand(0, 15);
    $token_final = chr(ord('A') + $prev);
    $len = strlen($token_auth);
    for ($i = 0; $i < $len; $i++) {
        $a = $token_auth[$i];
        $b = $token_site[$i];
        $ai = intval($a, 16);
        $bi = intval($b, 16);
        $token_final .= chr(ord($a) - $prev) .
                        chr(ord($b) + $ai);
        $prev = $bi;
    }

    // Salvăm valorile aleatoare în sesiunea locală,
    // pe server.
    $_SESSION['user_auth_rnd'] =
        array($token_rnd1, $token_rnd2, time());
}
```

```

    return $token_final;
}

```

Implementarea funcției `gen_session_token()`:

```

protected function gen_session_token () {
    if (!$this->session_token) {
        $host = isset($_SERVER['HTTP_HOST']) ?
            $_SERVER['HTTP_HOST'] :
            (isset($_SERVER['SERVER_NAME']) ?
                $_SERVER['SERVER_NAME'] : '');

        // Folosim funcția de fărâmițare SHA-512 pentru a genera
        // un jeton unic dat de ID-ul sesiunii, IP-ul
        // utilizatorului, navigatorul web, server și port.
        $this->session_token = hash('sha512',
            $this->user_ip . $this->engine_id .
            @session_id() .
            @$_SERVER['HTTP_USER_AGENT'] .
            @$_SERVER['HTTP_ACCEPT_CHARSET'] .
            @$_SERVER['HTTP_ACCEPT_ENCODING'] .
            @$_SERVER['HTTPS'] .
            @$_SERVER['SERVER_PORT'] . $host);
    }
}

```

În `login.js` funcția `adminFormLogin()`:

```

function adminFormLogin(frm) {
    // Citim jetonul.
    var tok = adminFormParseToken(frm.tok.value);
    if (!tok) {
        return false;
    }

    // Folosim jetonul trimis de server pentru a cripta
    // parola.
    var sha512a = new jsSHA(frm.user.value +
                            frm.pass.value +
                            tok[1]),
        sha512b = new jsSHA(sha512a.getHash('SHA-512',
                                            'HEX')),

```

```

        res = sha512b.getHMAC(tok[0], 'ASCII', 'SHA-512',
                                'HEX');

    // Actualizăm variabilele din formular.
    frm.tok.value = '';
    frm.pass.value = res;
    frm.login.value = 'y';
    tok = sha512a = sha512b = res = null;

    return true;
}

```

Funcția `adminFormParseToken()` este:

```

function adminFormParseToken (tok) {
    if (tok.length != 257) {
        return false;
    }

    var toka = [], tokb = [];

    // Inversăm operația din users.php,
    // funcția gen_auth_token().
    var prev = parseInt(tok.charCodeAt(0) -
                        'A'.charCodeAt(0));
    for (var a1, b1, a2, b2, a3, i = 1; i < 256;
        i += 2) {
        a1 = tok.charCodeAt(i);
        b1 = tok.charCodeAt(i+1);
        a2 = String.fromCharCode(a1 + prev);
        toka.push(a2);
        a3 = parseInt(a2, 16);
        b2 = String.fromCharCode(b1 - a3);
        tokb.push(b2);
        prev = parseInt(b2, 16);
    }

    // Primul element din vector este jetonul sesiunii,
    // al doilea element este jetonul siteului.
    return [toka.join(''), tokb.join('')];
}

```

După execuția codului JavaScript se trimite cererea de autentificare către server, acolo unde se execută funcția `login()` din `users.php`:

```
public function login ($req-user , $req-pass) {  
    // Verificăm dacă numele contului este valid: conține numai  
    // caractere alfabetice și numere.  
    // Verificăm dacă avem variabila de sesiune user_auth_rnd  
    // unde sunt salvate valorile aleatoare pentru cererea de  
    // autentificare în derulare.  
    if (!$req-pass ||  
        !$this->is_username_valid($req-user) ||  
        !isset($_SESSION['user_auth_rnd'])) {  
        return false;  
    }  
  
    // Dacă utilizatorul este autentificat deja, nu permitem  
    // re-autentificarea. În plus, închidem sesiunea.  
    if ($this->did_auth()) {  
        $this->logout();  
        return false;  
    }  
  
    // Valorile aleatoare salvate pe server de  
    // gen_auth_token() la încărcarea paginii de autentificare.  
    $token_rnd = $_SESSION['user_auth_rnd'];  
    unset($_SESSION['user_auth_rnd']);  
  
    // Verificăm dacă nu a expirat cererea de autentificare.  
    // După încărcarea paginii de autentificare utilizatorul  
    // are timp maxim 10 minute până cererea expiră.  
    if ((time() - $token_rnd[2]) >  
        $this->tperiod_auth_form) {  
        return false;  
    }  
  
    // Verificăm dacă utilizatorul este blocat. Salvăm pe server  
    // numărul de eșecuri de autentificare. După 5 încercări  
    // eșuate IP-ul este blocat.  
    if ($this->user_blocked) {  
        return false;  
    }  
}
```

```
// Citim parola utilizatorului de pe server, în forma
// criptată.
$user_key = $this->get_user_key($req_user);
if (!$user_key) {
    // Dacă numele contului nu există considerăm că acesta
    // este un eșec de autentificare.
    $this->inc_ipfails();
    return false;
}

// Regenerăm jetonul sesiunii.
$this->gen_session_token();

// Regenerăm jetonul de autentificare, folosindu-ne de
// variabilele aleatoare salvate pe server de
// gen_auth_token().
$token_auth = hash_hmac('sha512',
                        $this->session_token .
                        $token_rnd[0],
                        $token_rnd[1]);

// Criptăm parola de pe server cu jetonul de autentificare.
$pass = hash_hmac('sha512', $user_key, $token_auth);

// Comparăm valoarea locală $pass cu cea trimisă de client.
if ($req_pass != $pass) {
    // Valoarea este greșită, înseamnă că utilizatorul nu a
    // introdus numele contului corect, parola sau jetoanele
    // folosite pentru criptare au fost diferite. Jetoanele
    // pot fi diferite doar dacă $token_rnd au fost diferite,
    // dacă IP-ul, navigatorul web sau alte informații folosite
    // de gen_auth_token() s-au schimbat.
    $this->inc_ipfails();

    return false;
}

// Utilizatorul este autentificat cu succes!
```

```
// Salvăm pe server în sesiunea PHP datele despre
// autentificare.
$date_auth = time();
$_SESSION['user_auth'] = array(
    // Salvăm un jeton cu datele despre sesiune. Dacă la
    // următoarele încărcări se schimbă ceva, sesiunea este
    // închisă și utilizatorul trebuie să se reautentifice.
    'tok' => hash_hmac('sha512', $user_key ,
                      $this->session_token .
                      $date_auth),
    'user' => $req_user ,
    'date_auth' => $date_auth ,
    'date_last' => $date_auth ,
);

return true;
}
```

La fiecare încărcare de pagină în secțiunea de administrare funcția `did_auth()` este chemată. Aceasta verifică dacă valorile din `$_SESSION['user_auth']` sunt încă valide.

La fiecare eșec de autentificare numărul de eșecuri este incrementat pentru IP-ul în cauză. Acest număr este salvat în directorul `.private/users/ipfails/`, într-un fișier care are drept nume IP-ul clientului. Un fișier `globalfails` este folosit pentru a contoriza eșecuri globale, indiferent de IP-uri. Dacă serverul este supus unui atac concertat, dispersat pe mai multe IP-uri, sistemul de autentificare se blochează la toți utilizatorii.

3.1.1.2 Formularul de resetare a parolei

Utilizatorii care uită parola pot opta să reseteze parola contului. Ei primesc prin email o parolă nouă, la adresa configurată în profilul lor.

Figura 3.2: Formularul de resetare a parolei.

Language: Română • English Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Universitate Facultăți Academic Studenți Cercetare Departamente Reviste Resurse Personal

Administrare

Introduceți numele contului dvs. pentru a vă trimite o parolă nouă la adresa de email asociată cu contul dat.

Cont:

finding Letheng

Type the two words:

RECAPTCHA

Resetează parola [Înapoi la autentificare](#)

© 2011 Universitatea „Aurel Vlaicu” din Arad. Toate drepturile rezervate. Site realizat de ROBO Design. Coordinator Prof. Dr. Ioan Dzitac.

Pentru formularul de resetare a parolei folosim librăria anti-spam [recap-
tcha.net](https://www.recaptcha.net/). Cod din `admin.php`:

```
$token = htmlentities($users_manager->get_auth_token(),
                      ENT_QUOTES, 'utf-8');

echo "<p><em>Resetare parola</em></p>
<form method='post' action='?'
      onsubmit='return adminFormForgotPassword(this)'>
<p><label>Nume cont:
      <input type='text' name='user'></label></p>
" . recaptcha_get_html($recaptcha['publickey']) . "
<p>
<input type='hidden' name='forgot' value='1'>
```

```

<input type='hidden' name='sendpass' value='n'>
<input type='hidden' name='l' value='ro'>
<input type='hidden' name='tok' value='$token'>
<input type='submit' value='Trimite'>
<a href='?'>Inapoi la autentificare</a>
</p>
</form>";

```

De această dată folosim funcția JavaScript `adminFormForgotPassword()` și chemăm `recaptcha_get_html()` pentru a afișa elementele necesare testului anti-spam.

Ca măsură de siguranță folosim jetonul de autentificare pentru a cripta numele contului, chiar dacă este transmis și fără criptare. Verificăm doar dacă la client funcționează toate măsurile de siguranță: dacă nu se schimbă sesiunea, dacă nu reușește să creeze numele contului cel mai probabil sunt probleme tehnice – probleme ce pot fi suspectate a face parte din încercările unui atacator.

La fel ca la formularul de autentificare avem câmpul `sendpass` cu valoarea inițială `n`. Ea trebuie să fie `y` pentru a permite procesarea cererii de resetare a parolei.

Funcția `adminFormForgotPassword()` arată astfel:

```

function adminFormForgotPassword (frm) {
    // Citire jeton.
    var tok = adminFormParseToken(frm.tok.value);
    if (!tok) {
        return false;
    }

    // Criptare nume cont.
    var sha512 = new jsSHA(frm.user.value + tok[1]),
        res = sha512.getHMAC(tok[0], 'ASCII',
                               'SHA-512', 'HEX');

    // Actualizare formular.
    frm.tok.value = res;
    frm.sendpass.value = 'y';
    sha512 = res = null;

    return true;
}

```

După ce utilizatorul introduce numele contului și răspunsul la „provocarea” anti-spam a serviciului reCaptcha, se execută următorul cod din `admin.php`:

```
$result = false;
if (isset($_POST['forgot'], $_POST['sendpass'],
          $_POST['user'], $_POST['tok'])) &&
    $_POST['sendpass'] == 'y') {
    // Folosim librăria reCaptcha pentru a verifica răspunsul
    // dat de utilizator.
    $result = recaptcha_check_answer(
        $recaptcha['privatekey'],
        $_SERVER["REMOTEADDR"],
        $_POST["recaptcha_challenge_field"],
        $_POST["recaptcha_response_field"]);
    if (!$result->is_valid) {
        unset($_POST['sendpass']);
        $pass_send_status = 'failed-recaptcha';
    }
}

if (isset($_POST['sendpass'])) {
    // Resetăm parola.
    $result =
        $users_manager->reset_user_password(
            $_POST['user'], $_POST['tok']);
    if ($result) {
        // Trimitem parola nouă prin email.
        // Avem $result['password'] și $result['email'].
    }
}
```

Scriptul `admin.php` trimite noua parolă generată, `$result['password']`, prin email la adresa dată de `$result['email']`. Funcția PHP `mail()` este folosită în acest scop.

3.1.2 Gestionarea conturilor de administrare site

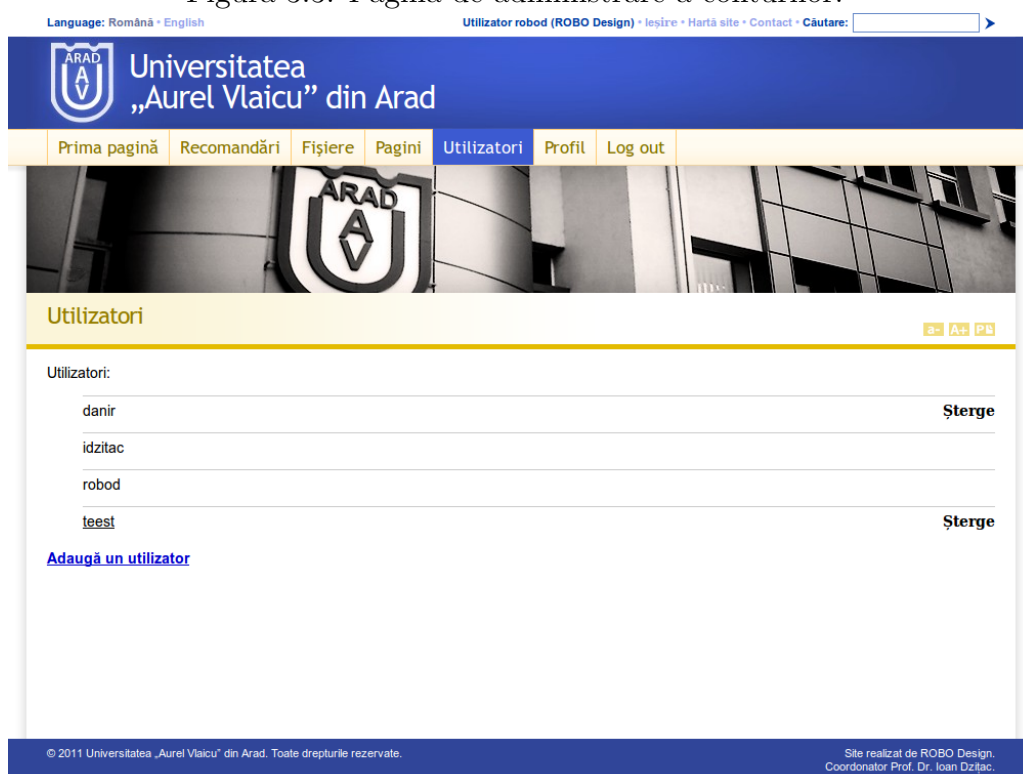
Gestionarea conturilor se face pe baza drepturilor primite de administratorul în cauză. Pentru adăugarea de conturi noi și pentru ștergerea conturilor existente se folosește interfața implementată de `admin/users.php`. Pentru editarea drepturilor se folosește interfața din `admin/permissions.php`. Toate

funcțiile care țin de conturi sunt implementate în clasa `RD_users_manager` din `.private/users/users.php` – celelalte fișiere **PHP** sunt doar interfețe grafice.

3.1.2.1 Lista conturilor

Un administrator cu drepturi suficiente poate să vizualizeze și să editeze conturile altor administratori de site.

Figura 3.3: Pagina de administrare a conturilor.



Lista de conturi este generată citindu-se lista de fișiere de pe disc din directorul `.private/users/sites/`.

Pagina este generată de scriptul `admin/users.php`. Acesta face chemare la scriptul `admin/include.php`.

În `include.php` se fac verificările dacă utilizatorul are dreptul de a accesa pagina de administrare dorită. În acest script se cheamă și funcția `did_auth()` din `.private/users/users.php`.

Pagina dată de `admin/users.php` este generată în felul următor:

```
// Modulul de gestionare utilizatori ne oferă posibilitatea de
// a afla lista de utilizatori. Această funcție returnează un
// array/vector de conturi.
$users = $users_manager->get_users();

echo "<p>{$lng['users']}</p>
<ul class='pageslist'>\n";

foreach ($users as $user) {
    $user2 = htmlspecialchars($user, ENT_QUOTES,
                                'utf-8');
    $user3 = htmlspecialchars(urlencode($user),
                                ENT_QUOTES,
                                'utf-8');

    echo "<li>";

    // Verificăm dacă avem dreptul să vizualizăm permisiunile
    // contului $user.
    if ($users_manager->have_perm("admin/users/$user",
                                'view_perms')) {
        echo "<a href='permissions.php?l=$l' .
            '&user=$user3'>$user2</a>";
    } else {
        echo $user2;
    }

    // Verificăm dacă avem drepturi să ștergem $user.
    if ($user != $user_self &&
        $users_manager->have_perm("admin/users/$user",
                                'delete')) {
        echo "<a class='user-del' href='#user-delete'
            onclick='return
            siteAdmin.userDelete(\" $user2\",
            this);'>{$lng['delete-user']}</a>";
    }
    echo "</li>";
}
echo "</ul>\n";
```

După cum se vede în codul de mai sus pagina folosește o funcție spe-

cifică a clasei `RD_users_manager`: `have_perm($sectiune, $actiune)`. Cu această funcție verificăm dacă utilizatorul autentificat în momentul chemării are dreptul să execute acțiunea dată în secțiunea dată. Această chemare simplă rezolvă problema granulară a drepturilor în cel mai mic detaliu.

3.1.2.2 Verificarea drepturilor unui cont

În implementarea funcției `have_perm()` se citește arborele de drepturi asociat utilizatorului pentru a se determina dacă acțiunea în secțiunea dată este permisă sau nu.

```
public function have_perm ($section , $action) {
    // Verificăm dacă utilizatorul este autentificat.
    if (!$this->did_auth()) {
        return false;
    } else if ($section == 'editable_' ||
               $section == 'default_') {
        // Nu permitem secțiuni cu nume „speciale”.
        return false;
    } else if ($action == 'recursive_' ||
               $action == 'default_') {
        // Nici acțiuni cu nume speciale.
        return false;
    }

    // Pornim de la răspunsul implicit: acțiunea nu este
    // permisă.
    $result = $this->perm_default;

    // Citim dreptul implicit al utilizatorului: este posibil ca
    // acesta să aibă toate drepturile sau nici unul, implicit.
    $perms = &$amp;this->user_data['permissions'];
    if (isset($perms['default_'])) {
        $result = $perms['default_'];
    }

    // Pornim iterația în „arborele” dat de secțiunea dorită.
    $arr = explode('/', $section);
    $path = '';
    foreach ($arr as $part) {
        if ($path == '') {
```

```

    $path = $part;
} else {
    $path .= '/' . $part;
}

// Dacă nu este nici un drept în calea curentă,
// nu avem ce face.
if (!isset($perms[$path])) {
    continue;
}

// Nu ne interesează drepturile acordate într-o pagină
// părinte dacă acele drepturi nu sunt aplicate recursiv.
if ($path != $section &&
    (!isset($perms[$path]['recursive_']) ||
     !$perms[$path]['recursive_'])) {
    continue;
}

// Vedem ce drept este dat acțiunii dorite.
if (isset($perms[$path][$action])) {
    $result = $perms[$path][$action];
} else if (isset($perms[$path]['default_'])) {
    // ... sau folosim dreptul acordat implicit pentru
    // calea curentă.
    $result = $perms[$path]['default_'];
}
}

// Verificări specifice pentru gestionarea de conturi.
if ($result && count($arr) == 3 &&
    $arr[0] == 'admin' &&
    $arr[1] == 'users' &&
    ($action == 'edit_perms' ||
     $action == 'delete')) {

    // Nu permitem utilizatorului curent să își șteargă contul.
    if ($action == 'delete' &&
        $arr[2] == $this->user_name) {
        return false;
    } else if ($action == 'edit_perms' &&

```

```

        $arr[2] == $this->user_name) {
    // Nu permitem utilizatorului curent să își editeze
    // drepturile.
    return false;
}

// Verificăm dacă numele de cont este valid.
if (!$this->is_username_valid($arr[2])) {
    return false;
}

// Verificăm dacă numele de cont există.
$user_data = $this->get_user_data($arr[2]);
if (!$user_data) {
    return false;
}

// Verificăm dacă drepturile utilizatorului „țintă” sunt
// editabile, când acțiunea dorită este edit_perms.
// Dacă acțiunea este delete verificăm dacă utilizatorul
// poate fi șters sau nu.
if (($action == 'edit_perms' &&
    isset($user_data['permissions']['editable_'])
    && !$user_data['permissions']['editable_'])
    || ($action == 'delete' &&
    isset($user_data['deletable']) &&
    !$user_data['deletable'])) {
    $result = false;
}
}

return $result;
}

```

Acest model de drepturi este scalabil, poate fi folosit în orice context al siteului, la cel mai mic detaliu.

3.1.2.3 Adăugarea de utilizatori

Pentru a adăuga un cont administratorul trebuie să introducă numele utilizatorului, numele contului, adresa de email a acestuia și parola inițială.

Figura 3.4: Opțiunea de adăugare cont.

La fel ca la orice formular de autentificare, parola nu este transmisă așa cum este scrisă de utilizator, ci aceasta este criptată cu algoritmul SHA-512 [8]. Parola este criptată împreună cu jetonul siteului, plus se transmite încă un câmp ascuns în care parola este criptată cu jetonul sesiunii pentru verificare dublă. Funcția `siteAdmin.userAddSubmit()` este chemată atunci când utilizatorul alege să trimită formularul la server. Această funcție se găsește în fișierul `js/admin.js`:

```
this.userAddSubmit = function (form) {
    // Ne asigurăm că numele de cont nu este prea scurt sau
    // prea lung.
    if (!form.username.value ||
        form.username.value.length < 3) {
```

```

    alert("<?php echo $lng['username-too-short']; ?>");
    return false;
}

if (form.username.value.length > 32) {
    alert("<?php echo $lng['username-too-long']; ?>");
    return false;
}

form.username.value =
    form.username.value.toLowerCase();

// Verificăm dacă există caractere nepermise în numele
// contului.
if (!/^[a-z0-9][a-z0-9._-]+$/.
    test(form.username.value)) {
    alert("<?php echo
        $lng['invalid-username']; ?>");
    return false;
}

// Verificăm dacă parolele introduse sunt aceleași.
if (form.pass1.value != form.pass2.value) {
    alert("<?php echo
        $lng['passwords-not-matching']; ?>");
    return false;
}

// Lungimea minimă a parolei...
if (!form.pass1.value ||
    form.pass1.value.length < 5) {
    alert("<?php echo
        $lng['password-too-short']; ?>");
    return false;
}

// Validăm adresa de email.
if (!form.email.value ||
    !this.checkEmail(form.email.value)) {
    alert("<?php echo $lng['invalid-email']; ?>");
    return false;
}

```

```

    }

    // Citim jetonul de autentificare.
    var tok = this.parseAuthToken(form.tok.value);
    if (!tok) {
        return false;
    }

    // Criptăm parola cu jetonul siteului.
    var sha512a = new jsSHA(form.username.value +
                            form.pass1.value + tok[1]);

    form.pass1.value = sha512a.getHash('SHA-512',
                                       'HEX');

    sha512a = null;

    // Criptăm parola încă o dată cu jetonul sesiunii.
    var sha512b = new jsSHA(form.pass1.value);
    form.pass2.value = sha512b.getHMAC(tok[0],
                                       'ASCII',
                                       'SHA-512',
                                       'HEX');

    tok = sha512b = null;

    form.tok.value = '';

    return true;
};

```

Pentru mai multe detalii despre jetoanele folosite și despre modelul de autentificare al utilizatorilor trebuie văzută secțiunea [3.1.1.1](#).

Fișierul `js/admin.js` este **JavaScript** generat dinamic cu **PHP**. **Apache** este configurat să trimită fișierul `admin.js` prin interpretatorul PHP – vezi fișierul `.htaccess` din același director. Această abordare este folosită pentru a putea face verificarea dacă utilizatorul este autentificat sau nu, deoarece nu dorim ca utilizatorii neautentificați să acceseze fișiere sensitive din punct de vedere al securității siteului. În plus, această abordare ne permite și utilizarea de mesaje traduse din `lang_en.php` și `lang_ro.php`.

Pe server, în `admin/users.php`, la adăugarea unui utilizator se fac aceleași verificări pentru validarea câmpurilor, pentru a nu permite păcălirea aplicației

web. După validare se execută:

```
// Drepturile implicite: permitem doar acțiunea view în
// paginile de administrare ale siteului.
$defaultUserPerms = array(
    'admin' => array(
        'view' => true,
        'recursive_' => true
    )
);

// Trimitem cererea de adăugare a unui nou utilizator la
// modulul de gestionare a conturilor.
$users_manager->add_user($_POST['username'],
    $_POST['pass1'],
    $_POST['pass2'],
    array('displayName' => $_POST['displayname'],
        'email' => $_POST['email']),
    $defaultUserPerms);
```

În implementarea funcției `add_user()` se verifică dacă utilizatorul are dreptul să adauge alți utilizatori, se verifică dacă `pass1` și `pass2` sunt valide și dacă numele noului cont este valid. La final, fișierele `.key` și `.json` pentru noul cont sunt salvate în `.private/users/sites/`.

Operația de ștergere a conturilor constă în verificarea drepturilor necesare pentru a face acest lucru și în ștergerea propriu-zisă a fișierelor de pe disc.

3.1.2.4 Editarea drepturilor unui cont

Drepturile unui cont pot fi afișate și modificate atunci când în lista de conturi administratorul face *click* pe un cont:

Figura 3.5: Lista de drepturi date unui cont.

Language: Română • English Utilizator robod (ROBO Design) • Iesire • Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Prima pagină | Recomandări | Fișiere | Pagini | Utilizatori | Profil | Log out

Drepturi utilizator: teeeeeest

Utilizatori a- A+ P

[Înapoi la lista de utilizatori.](#)

Drepturile utilizatorului **teeeeeest** (teest):

admin	Modifică Șterge
Orice acțiune este permisă implicit: Nu	
Drepturile sunt recursive: Da	
view: Nu	Modifică Șterge
Adaugă un drept aici	
admin/pages/omg	Modifică Șterge
Orice acțiune este permisă implicit: Nu	
Drepturile sunt recursive: Da	
edit: Da	Modifică Șterge
Adaugă un drept aici	
admin/users/wow	Modifică Șterge
Orice acțiune este permisă implicit: Da	
Drepturile sunt recursive: Da	
delete: Nu	Modifică Șterge
Adaugă un drept aici	
Adaugă o secțiune	

După cum se poate vedea în captura de ecran de mai sus, pagina afișează într-un mod ușor de înțeles secțiunea **permissions** din fișierul **JSON** al conținutului selectat (vezi detalii despre acest fișier în secțiunea 3.1). Drepturile pentru fiecare acțiune, din fiecare secțiune sunt listate lizibil, într-un limbaj accesibil.

Gestionarul de drepturi este scriptul `admin/permissions.php`. Acesta afișează drepturile, permite adăugarea, modificarea și ștergerea de secțiuni și

acțiuni permise pentru contul dorit. Pentru a citi toate drepturile unui cont se folosește funcția `$users_manager->get_permissions($user)` din clasa `RD_users_manager`.

Administratorul poate adăuga o secțiune nouă în care să modifice drepturile contului:

Figura 3.6: Editarea drepturilor: adăugare secțiune.

Language: Română • English Utilizator robod (ROBO Design) • Ieșire • Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Prima pagină | Recomandări | Fișiere | Pagini | Utilizatori | Profil | Log out

Drepturi utilizator: teeeeeest

Utilizatori

[Înapoi la lista de utilizatori.](#)

Drepturile utilizatorului **teeeeeest** (teest):

admin	Orice acțiune este permisă implicit: Da	Drepturile sunt recursive: Da	view: Nu	Adaugă un drept aici	Modifică Șterge
admin/pages/omg	Orice acțiune este permisă implicit: Da	Drepturile sunt recursive: Da	edit: Da	Adaugă un drept aici	Modifică Șterge
admin/users/wow	Orice acțiune este permisă implicit: Da	Drepturile sunt recursive: Da	delete: Nu	Adaugă un drept aici	Modifică Șterge

Adaugă/modifică o secțiune

Secțiunea din site:

O acțiune se aplică unei secțiuni din site. Trebuie să indicați secțiunea în care adăugați drepturile utilizatorului. De exemplu, utilizatorul poate avea permisiunea de a executa acțiunea edit în `admin/pages/pagina/sub-pagina-dorită`.

Dreptul implicit pentru orice acțiune:

☐ Drepturile date în această secțiune sunt recursive.

Adaugă o secțiune

Acest formular, precum și celelalte formulare, apare dinamic, fără a fi reîncărcată pagina. Acest lucru este făcut posibil cu codul **JavaScript** din fișierul `js/admin.js`. Se face doar manipulare **DOM** și se folosesc evenimente de interacționare cu formularele.

Formularul de adăugare secțiune în drepturile unui cont are următoarele câmpuri:

- **Secțiunea din site:** unde utilizatorul poate alege dintre câteva secțiuni predefinite sau poate să scrie calea dorită manual. Secțiunile predefinite sunt:
 - admin,
 - admin/files,
 - admin/pages,
 - admin/pages/en,
 - admin/pages/ro,
 - admin/users.

Un câmp adițional permite utilizatorului să scrie calea fișierului, utilizatorul sau a paginii dorite pentru care dorește să modifice drepturile contului. De exemplu, utilizatorul poate să aleagă secțiunea `admin/pages/ro` și adaugă `contact` pentru a modifica drepturile paginii Contact în limba română.

- **Dreptul implicit pentru orice acțiune:**
 - Drepturile sunt moștenite de la secțiunea părinte;
 - Acțiunea nu este permisă;
 - Acțiunea este permisă.
- **Drepturile date în această secțiune sunt recursive sau nu.**

Aceste opțiuni transpun într-o interfață plăcută opțiunile tehnice discutate în secțiunea 3.1 oferite de fișierul JSON al fiecărui cont.

În `js/admin.js` funcția `siteAdmin.permsAddEditSectionSubmit()` validează datele formularului înainte să fie trimise la server. Practic acest cod concatenează secțiunea predefinită aleasă de utilizator cu calea scrisă în câmpul de tip text. În plus, codul verifică dacă sunt folosite căi „speciale” precum `default_` sau `editable_` – aceste jetoane nu pot fi folosite ca secțiuni.

Pe server `permissions.php` doar preia valorile câmpurilor și construiește noul obiect `$perms`:

```
$perms = $users_manager->get_permissions($user);

$post_section =
    trim(trim(strtolower($_POST['section'])), '/');
$perms[$post_section] = array();

if ($_POST['defaultRight'] == 'allow') {
    $perms[$post_section]['default_'] = true;
} else if ($_POST['defaultRight'] == 'deny') {
    $perms[$post_section]['default_'] = false;
}

if (isset($_POST['recursive']) &&
    $_POST['recursive'] == 'y') {
    $perms[$post_section]['recursive_'] = true;
}

// Actualizăm drepturile utilizatorului.
$users_manager->set_permissions($user, $perms);
```

Modulul de gestionare conturi este responsabil cu serializarea obiectului `$perms` în format JSON cu funcția `json_encode()` și cu salvarea acestuia pe disc. Ambii pași sunt triviali, nefiind necesară exemplificarea acestora cu cod.

Adăugarea unei acțiuni funcționează foarte similar cu cele prezentate mai sus. Utilizatorul alege secțiunea și acțiunea dorită, pentru ca mai apoi să bifeze dacă aceasta este permisă sau nu.

Figura 3.7: Editarea drepturilor: adăugare acțiune.

Language: Română • English Utilizator robot (ROBO Design) • [Ieșire](#) • [Hartă site](#) • [Contact](#) • [Căutare:](#)

Universitatea „Aurel Vlaicu” din Arad

[Prima pagină](#) [Recomandări](#) [Fișiere](#) [Pagini](#) [Utilizatori](#) [Profil](#) [Log out](#)

Drepturi utilizator: teeeeeest

[Înapoi la lista de utilizatori.](#)

Drepturile utilizatorului **teeeeeest** (teest):

Secțiune	Orice acțiune este permisă implicit	Drepturile sunt recursive	view	edit	Acțiune	Modifică Șterge
admin	☐	☐	Nu	☐		
admin/pages/omg	☐	☐	☐	☐		
admin/users/wow	☐	☐	☐	☐		

Adăugă o acțiune

Secțiunea din site:
admin/users/wow

O acțiune se aplică unei secțiuni din site. Trebuie să indicați secțiunea în care adăugați drepturile utilizatorului. De exemplu, utilizatorul poate avea permisiunea de a executa acțiunea edit în admin/pages/pagina/sub-pagina-dorită.

Acțiunea:
add

Utilizatorul poate executa multiple acțiuni într-o singură secțiune a siteului. De exemplu, utilizatorul poate edita o pagină, sau poate adăuga/șterge sub-pagini. În acest câmp introduceți acțiunea pentru care doriți să acordați/revocați un drept utilizatorului selectat.

Acțiunea dată este permisă sau nu?
Nu

Adăugă o secțiune

Pentru fiecare secțiune predefinită și cunoscută de modulul de administrare câmpul „acțiune” oferă opțiuni cunoscute, specifice secțiunii în cauză. De exemplu pentru `admin/users` există acțiunea `edit_perms`, iar la `admin/files` există `upload`. Opțional, dacă utilizatorul dorește, acesta poate să scrie orice altă acțiune într-un câmp text.

Codul JavaScript din `js/admin.js`, cât și codul din `admin/permissions.php` validează valorile câmpurilor. Nu este permisă adăugarea de acțiuni cu

nume „speciale”, precum `default_` sau `recursive_`. Salvarea pe disc se face tot cu funcția `$users_manager->set_permissions()`.

3.1.2.5 Securitatea sistemului de autentificare

Sistemul este susceptibil la atacuri de tip *man in the middle*, în care un răufăcător spionează conexiunea clientului sau a serverului, alterând sau furând datele transmise de ambele părți. Această situație poate fi evitată ușor prin instalarea unui certificat digital pe server, în vederea folosirii unei conexiuni criptate cu protocolul **TLS**.

Dincolo de utilizarea protocolului TLS securitatea siteului depinde și de securitatea serverului și a softului folosit, începând de la distribuția Linux, kernel, Apache, PHP și multe alte aplicații.

Un alt aspect care trebuie luat în vedere este configurația serverului în rețeaua unde este instalat. De obicei, serverele sunt mult mai susceptibile atacurilor care vin din rețeaua locală.

Nu în ultimul rând nu trebuie uitat aspectul uman: administratorii siteului au responsabilitatea de a ține parolele în siguranță, de a folosi parole puternice – de lungime suficientă, cu o variație mare de caractere.

Codul prezentat mai sus nu include anumite verificări adiționale, pentru a fi ușor de citit. Pe lângă verificările de securitate, orice acțiune de autentificare, de ieșire, de adăugare/modificare/șterge cont este salvată într-un istoric care poate fi consultat ulterior. Operațiunile efectuate de utilizatori sunt înregistrate în orice moment.

3.1.3 Editare profil

Fiecare administrator poate să își modifice profilul contului:

Figura 3.8: Editare de profil utilizator.

The screenshot shows the 'Profil' (Profile) page of the University of Arad website. The page has a blue header with the university's logo and name. Below the header is a navigation bar with links: 'Prima pagină', 'Recomandări', 'Fișiere', 'Pagini', 'Utilizatori', 'Profil', and 'Log out'. The 'Profil' link is highlighted. The main content area is titled 'Profil' and contains two sections: 'Informații cont' (Account Information) and 'Schimbă parola' (Change Password). The 'Informații cont' section has input fields for 'Nume utilizator:' (User Name) with the value 'ROBO Design' and 'Email:' with the value 'user@example.com'. There is a 'Trimite' (Submit) button below these fields. The 'Schimbă parola' section has input fields for 'Parola veche:' (Old Password), 'Parola nouă:' (New Password), and 'Confirmare parolă nouă:' (Confirm New Password). There is a 'Trimite' (Submit) button below these fields. The footer of the page contains copyright information: '© 2011 Universitatea „Aurel Vlaicu” din Arad. Toate drepturile rezervate.' and 'Site realizat de ROBO Design. Coordonator Prof. Dr. Ioan Dăbâc.'

După cum se poate vedea în captura de ecran de mai sus, se poate modifica numele utilizatorului și adresa de email. De asemenea, se poate schimba parola de autentificare a contului.

Formularul prezentat se găsește în scriptul `admin/profile.php`. Codul JavaScript este dat de fișierul `js/admin.js`.

La modificarea profilului se folosește funcția JavaScript `siteAdmin.userProfileSubmit()`. Pe server se folosește funcția `update_profile()` din clasa `RD_users_manager` – vezi secțiunea 3.1.

Pentru modificarea parolei se face criptare cu funcția de fărâmițare SHA-512 așa cum am prezentat în secțiunea 3.1.1.1. Funcția JavaScript chemată este `siteAdmin.changePassword()`. Pe server se execută `set_user_key()` din `RD_users_manager`.

3.2 Gestionarea fișierelor de pe server

Administratorii siteului pot să încarce fișiere pe server, să le șteargă, să creeze directoare, să execute operații de redenumire, de curățare pagini **HTML** și de editare pagini.

Figura 3.9: Gestionarul de fișiere.

Language: Română • English Utilizator robod (ROBO Design) • Ieșire • Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Prima pagină Recomandări **Fișiere** Pagini Utilizatori Profil Log out

Fișiere

Directorul actual: / files / OM

☐ Nume fișier	Ultima modificare	Mărime
☐ 3753_2011_masuri_tranzitorii.pdf	2011-03-26 07:01:45	162.58 KB
☐ 3759_2011_CNATDCU.pdf	2011-03-26 07:27:10	129.69 KB
☐ 3760_2011_CNATDCU.pdf	2011-03-26 07:27:10	129.69 KB
☒ 3761_2011_CNATDCU.pdf	2011-03-26 07:01:45	129.69 KB
☒ 3794_2011_CNCS.pdf	2011-03-26 07:33:33	387.24 KB
☒ 3795_2011_CNCS.pdf	2011-03-26 07:02:25	54.87 KB
☐ 3931_2011_CNATDCU.pdf	2011-03-26 07:24:10	143.04 KB

[Inversare selecție](#)

Adaugă fișiere

+ "/home/robod/Images/c/ Choisir...

X "/home/robod/Images/e/ Choisir...

X "/home/robod/Documen Choisir...

Operații

- Șterge
- Redenumeste
- Adaugă director
- Curățare pagină
- Editare HTML

© 2011 Universitatea „Aurel Vlaicu” din Arad. Toate drepturile rezervate. Site realizat de ROBO Design. Coordonator Prof. Dr. Ioan Dzitjac.

Pagina din captura de ecran de mai sus este generată de scriptul `admin/files.php`. Interactivitatea paginii este dată de codul **JavaScript** din fișierul `js/admin.js`.

După cum se vede din captura de ecran de mai sus, `files.php` listează fișierele și directoarele din directorul `files/OM/`, din rădăcina siteului. Utilizatorul poate naviga prin ierarhia de directoare de pe server. Cu ajutorul funcțiilor **JavaScript** fișierele/directoarele pot fi selectate, ca mai apoi să fie aplicate orice fel de operații.

Încărcarea de fișiere noi pe server se face în directorul curent. Utilizatorul poate adăuga oricâte fișiere dorește simultan, într-un singur pas. Pentru a adăuga un nou fișier se face *click* pe semnul plus „+” – moment în care se execută funcția JavaScript `siteAdmin.fileAdd()`.

```
this.fileAdd = function ()
{
    // Formularul de încărcare fișiere.
    var frm = document.getElementById( 'faddf' );

    // Primul câmp de tip fișier.
    var def = document.getElementById( 'addfileinput' );
    if (!frm || !def)
        return false;

    // Adăugăm încă un paragraf cu un câmp de tip fișier.
    var p = document.createElement( 'p' ),
        inp = document.createElement( 'input' ),
        a = document.createElement( 'a' );

    inp.name = def.name;
    inp.type = def.type;

    // Opțiunea de ștergere a câmpului de tip fișier.
    a.href = '#';
    a.title = "<?php echo $lng[ 'admin-js ' ][2]; ?>";
    a.addEventListener( 'click', function () {
        this.parentNode.parentNode.
            removeChild( this.parentNode );
    }, false );

    a.appendChild( document.createTextNode( 'X' ) );
    p.appendChild( a );
    p.appendChild( inp );

    frm.insertBefore( p, def.parentNode.nextSibling );

    return false;
};
```

După trimiterea fișierelor pe server, codul **PHP** preia lista de fișiere și le pune în directorul curent. În configurația modului există o listă de extensii

de fișiere care nu sunt permise. De exemplu, dacă administratorul încearcă să încarce un fișier de tip PHP sau de tip executabil (binar), fișierul este respins. Această măsură a fost implementată din motive de securitate. Dacă parola administratorului este compromisă, un atacator are acces limitat la server în acest mod. El poate doar să strice siteul folosind paginile de administrare, dar nu permitem să execute cod arbitrar pe server.

Operația de ștergere a fișierelor este executată numai pe fișiere, nu și pe directoare, pentru a preveni ștergerea de directoare în masă sau posibile pierderi nedorite de date.

Operația de redenumire folosește JavaScript pentru a cere utilizatorului noul nume al fiecărui fișier selectat. Pe server este folosită funcția PHP `rename()`. Noile nume nu au voie să aibă extensii nepermise.

Pentru adăugarea unui director utilizatorul este întrebat cu funcția `prompt()` din **DOM**, în JavaScript, ce nume dorește. Pe server se folosește funcția `mkdir()` pentru a crea directorul în `files/`.

Restul de operații, curățarea de pagini HTML și editarea acestora, sunt operații modulare. Practic există un vector de configurare a modului `files.php` care permite adăugarea de noi operații ce pot fi aplicate la fișiere. Implementarea fiecărei operații se găsește într-un alt fișier PHP, într-o funcție bine definită. Configurația arată astfel:

```
$file_modules = array(
    // Curățarea de fișiere HTML se face folosindu-se modulul ReTidy.
    // Vezi secțiunea 3.4.1 pentru mai multe detalii.
    'retidy' => array(
        'title' => $lng['module-retidy'],
        'jsfunc' => 'fileRetidy',
        'file_required' => true,
        'callback_func' => 'fileRetidy',
        'callback_include' => 'files-retidy.php',
        'allow_in_iframe' => true,
    ),

    // Editarea paginilor HTML se face cu editorul Awebitor.
    // Vezi secțiunea 3.4 pentru mai multe detalii.
    'edithtml' => array(
        'title' => $lng['module-edithtml'],
        'jsfunc' => 'fileEditHTML',
        'file_required' => true,
        'callback_func' => 'fileEditHTML',
        'callback_include' => 'files-edithtml.php',
```

```

    'allow_in_iframe' => false ,
  )
);

```

Fiecare operație nouă are un ID. Cele două operații definite mai sus au ID-urile `retidy` și `edithtml`. Proprietățile de configurare sunt:

`title` definește titlul afișat în meniul „Operații”. În acest meniu utilizatorul poate executa operația definită.

`jsfunc` definește funcția JavaScript chemată atunci când utilizatorul face *click* pe această operație în meniul paginii cu fișiere.

`file_required` este o proprietate booleană care indică dacă operația se poate executa numai pe fișiere sau nu.

`callback_include` definește fișierul PHP care va fi inclus când se execută operația pe fișiere.

`callback_func` definește numele funcției PHP care se execută după includerea fișierului dat de `callback_include`.

`allow_in_iframe` este o proprietate booleană care indică dacă operația poate fi executată atunci când gestionarul de fișiere este afișat în `<iframe>`-uri.

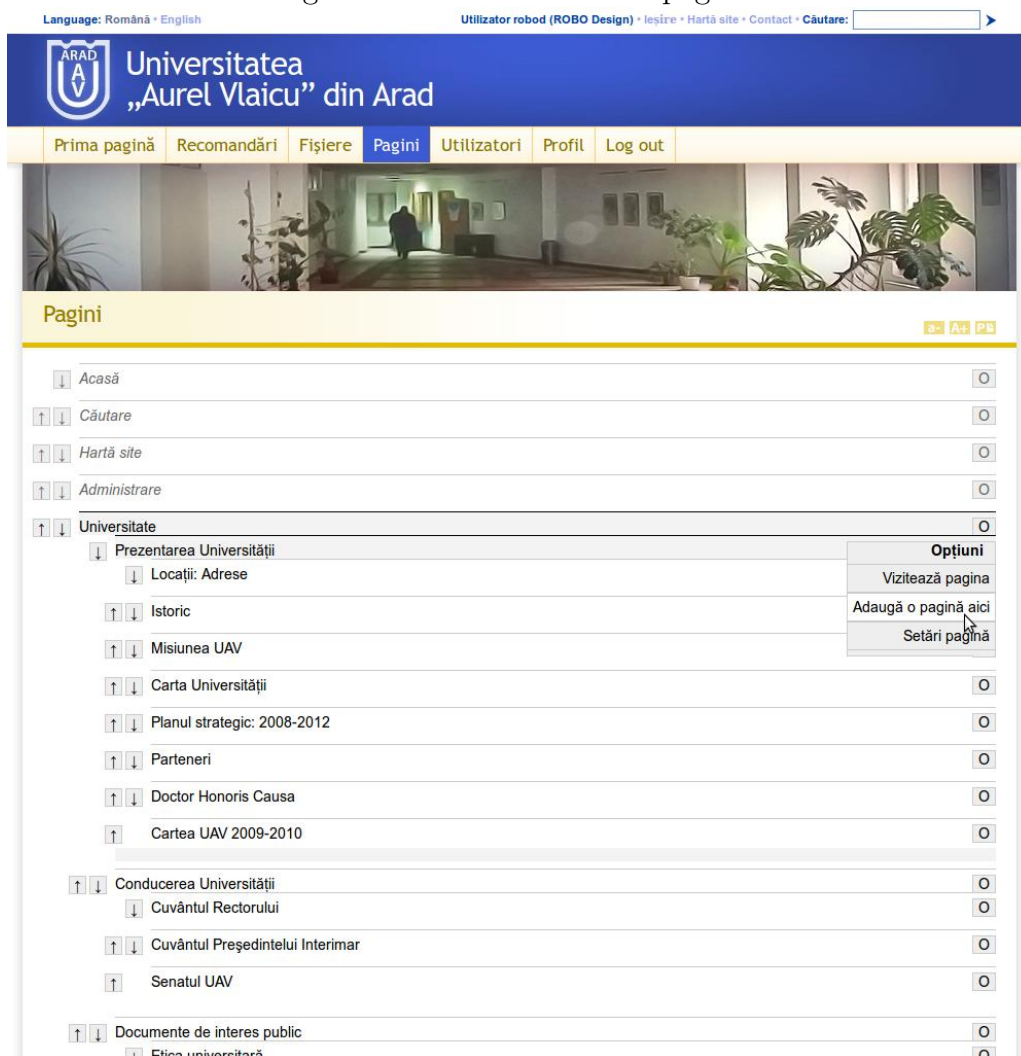
Editorul de pagini permite adăugarea de imagini în articole. Pentru adăugarea de imagini utilizatorul folosește acest gestionar de fișiere pentru a naviga pe server, pentru a selecta imaginea dorită. În acest mod de execuție unele operații nu se potrivesc – de exemplu editarea de pagini. Pentru detalii vezi secțiunea [3.4](#).

Orice acțiune de vizualizare sau de modificare a unui director sau fișier este trecută prin verificarea drepturilor necesare ale utilizatorului. Administratorul poate avea, de exemplu, dreptul de a încărca și edita fișiere numai într-un anumit set de directoare – a se vedea secțiunea [3.1.2.4](#).

3.3 Gestionarea paginilor

Paginile siteului uav.ro pot fi adăugate, modificate și șterse foarte ușor din gestionarul de pagini:

Figura 3.10: Gestionarul de pagini.



După cum se poate vedea din captura de ecran de mai sus, toate paginile din site sunt listate ierarhic. Administratorul poate să vadă din prima secundă harta siteului. El poate alege să editeze orice pagină dorește în orice moment. Săgețile din stânga permit utilizatorului să ordoneze paginile în meniurile siteului. Meniul „*Opțiuni*” din dreapta fiecărei pagini afișează:

Vizitează pagina este o legătură directă la pagină. Administratorul poate face *click* pe această opțiune pentru a vedea cum arată pagina la utilizatori.

Adaugă o pagină aici permite adăugarea unei sub-pagini.

Setări pagină permite administratorului să modifice titlul și calea paginii.

Șterge permite ștergerea paginii. Această opțiune apare doar dacă pagina nu are sub-pagini.

3.3.1 Structura siteului în format XML

Codul **PHP** care afișează și modifică structura siteului este în fișierul `admin/pages.php`. Acesta citește fișierul `xml/site_en.xml` sau `xml/site_ro.xml` în funcție de limba aleasă de administrator. Funcția recursivă `listPages()` este chemată pentru a afișa ierarhia paginilor.

Fișierele `xml/site_en.xml` și `xml/site_ro.xml` mențin lista de pagini din site. Acest format **XML** propriu arată astfel:

```
<?xml version="1.0" encoding="utf-8"?>
<nav xmlns="http://www.robodesign.ro/ns/nav-1.1">
  <menus xml:id="root" include="0" links="1">
    <menu xml:id="welcome" path="index" include="1"
      hide="1">
      <title>Acasa</title>
      <file>pages/welcome.php</file>
      <css media="screen , projection">
        css/welcome.css
      </css>
    </menu>
    <menu xml:id="sitemap" path="sitemap" include="1"
      hide="1">
      <title>Harta site</title>
      <file>pages/sitemap.php</file>
      <css media="screen , projection">
        css/sitemap.css
      </css>
    </menu>
    <menu xml:id="universitate" path="universitate"
      include="0" sub="sub-universitate">
```

```

    <title>Universitate</title>
    <file>pages/ro_universitate.html</file>
</menu>
<menu xml:id="contact" path="contact" include="1"
      hide="1">
    <title>Contact</title>
    <file>pages/contact.php</file>
    <css media="screen, projection">
      css/contact.css
    </css>
</menu>
<!-- ... -->
</menus>
<menus xml:id="sub-universitate" include="0"
      links="1">
  <menu xml:id="universitate-conducere"
    path="universitate/conducere"
    include="0">
    <title>Conducerea Universitatii</title>
    <file>
      pages/ro_universitate-conducere.html
    </file>
  </menu>
  <menu xml:id="universitate-prezentare"
    path="universitate/prezentare"
    include="0">
    <title>Prezentarea Universitatii</title>
    <file>
      pages/ro_universitate-prezentare.html
    </file>
  </menu>
  <!-- ... -->
</menus>
<!-- ... -->
</nav>

```

În codul sursă de mai sus pot fi văzute trei *taguri* importante:

<nav> este nodul rădăcină al documentului – fără a avea alt rol. În acest tag pot să apară doar taguri **<menus>**.

`<menus>` grupează împreună opțiunile dintr-un meniu. Fiecare opțiune este dată de un element `<menu>`.

`<menu>` reprezintă o opțiune dintr-un meniu, o pagină.

Pentru fiecare element `<menu>` se pot folosi următoarele atribute:

`xml:id` este jetonul unic de identificare a paginii. Acest jeton trebuie să fie unic în tot fișierul XML.

`sub` este un atribut opțional care conține ID-ul unei grupe de meniuri `<menus>`. În acest mod se construiește structura ierarhică a site-ului: meniuri cu submeniuri. Orice meniu poate avea un atribut `sub` care „cheamă”/include sub-pagini din alte grupe de meniuri (`<menus>`).

`path` definește calea completă a paginii.

`include` este un atribut boolean. Dacă valoarea este 1 atunci fișierul paginii dat de elementul `<file>` este inclus și executat de PHP. Dacă valoarea atributului este 0, atunci fișierul indicat de `<file>` este citit ca un document **HTML** și trimis exact cum este la navigatorul web, fără a fi interpretat.

`link` definește tipul de legătură pentru pagina definită. Trei valori sunt posibile:

- 0 pentru a folosi calea dată de `<file>` sau de `<link>` așa cum este ea, fără modificări.
- 1 pentru a genera o adresă „permanentă”: practic legătura paginii va fi valoarea atributului `path`. În unele cazuri, în unele configurații ale siteului calea are ca prefix limba aleasă de utilizator. Acest tip de legături sunt folosite pe siteul uav.ro.
- 2 pentru a folosi o legătură directă către `index.php`. Acest tip de legături sunt folosite atunci când serverul web **Apache** nu poate fi configurat să accepte rescrierea adreselor accesate de utilizatori. Un exemplu de legătură de acest tip: `index.php?path=/ro/universitate/prezentare`.

`hide` este un atribut, cu valoare booleană 0 sau 1, care permite ascunderea paginii din lista de meniuri afișată pe site.

`subpath` este un atribut, cu valoare booleană 0 sau 1, care spune că modulul indicat de `<file>` pentru pagina curentă se va ocupa cu procesarea căii, a adresei accesate de utilizator. Practic în acest moment se oprește căutarea paginii în structura XML a siteului. Se consideră că pagina găsită este cea dorită, chiar dacă adresa accesată are mai multe părți componente. De exemplu, dacă utilizatorul intră pe `/universitate/noutati/articol-1`, blogul Noutăți preia calea și caută să afișeze articolul cu jetonul de identificare `articol-1`.

Toate elementele `<menu>` trebuie să aibă atributul `xml:id`. Acestea pot să aibă și atributele `links` și `include`. Dacă un element `<menu>` nu are atributul `include`, valoarea implicită este preluată de la elementul părinte `<menu>` – la fel și cu atributul `link`.

Alte taguri care pot să apară în `<menu>`:

`<title>` titlul paginii.

`<file>` fișierul care se dorește a fi afișat sau interpretat de PHP.

`<link>` meniurile pot avea legături către pagini externe. În acest caz se folosește tagul `<link>` pentru a indica adresa.

`<css>` permite asocierea unui stil **CSS** cu pagina dată. Stilul este adăugat în antetul paginii, în `<head>`. Opțional, se poate folosi atributul `media` pentru a indica pe care media se aplică un anume stil.

În `admin/pages.php` fișierul XML este citit astfel

```
$doc = new DOMDocument( '1.0', 'utf8' );
$doc->preserveWhiteSpace = false;
$doc->formatOutput = true;
$doc->load( "xml/site_{$lang}.xml" );
```

Chemarea la funcția `listPages()` pornește de la elementul `<menu xml:id="root">`. Fiecare pagină dată de elementele `<menu>` este afișată. Pentru acele pagini cu atributul `sub`, funcția este re-chemată pentru a fi listate și sub-paginile.

```
function listPages( $menu_root )
{
    global $doc, $users_manager;
```

```

$root = $doc->getElementById($menu_root);

echo "\n<ul>\n";

$links = $root->getAttribute('links');
$include = $root->getAttribute('include');

$subs = $root->childNodes->length;
foreach($root->childNodes as $menu)
{
    $hide = $menu->getAttribute('hide');
    $path = array();
    $path['string'] = $menu->getAttribute('path');
    $path['array'] = explode('/', $path['string']);
    $id = $menu->getAttribute('xml:id');
    $submenu = $menu->getAttribute('sub');

    // ...

    if ($submenu)
        listPages($submenu);
}

echo "\n</ul>\n";
}

```

Drepturile administratorului de a adăuga pagini, de a le șterge sau modifica sunt verificate pentru fiecare pagină listată. Se face chemare la:

```

$users_manager->have_perm("admin/pages/$lng/$scale",
                           $actiune);

```

... unde *\$lng* este limba curentă, *\$scale* este calea paginii din listă, iar *\$actiune* este acțiunea dorită – de exemplu *add*, *edit* sau *delete*.

3.3.2 Configurația modului

Configurația modului de administrare a paginilor arată astfel:

```

// Lista de module cunoscute. În listPages() pentru fiecare element
// <menu include="1"> se verifică dacă modulul indicat de <file>
// apare în vectorul de mai jos.
$GLOBALS['known_modules'] = array(

```

```
// Prima pagină a siteului. Vezi secțiunea 3.8.
'pages/welcome.php' => array(
    // Nu permitem ștergerea paginii.
    'deletable' => false ,

    // Pagina este editabilă ca un articol simplu.
    'editable' => 'simple' ,

    // Fișierul HTML care poate fi editat.
    'editFile' => $pages_path . '/' .
                  $GLOBALS[ 'now' ][ 'language' ] .
                  '_welcome_partners.html' ,
    'add_module' => 'simple' ,

    // Nu permitem schimbarea căii pentru această pagină.
    'allowSetPath' => false ,
),

// Pagina Contact. Vezi secțiunea 3.9.
'pages/contact.php' => array(
    'deletable' => false ,

    // Această pagină este editabilă, dar are propriul modul de
    // administrare.
    'editable' => 'link' ,
    // Indicăm cu o legătură care este adresa editorului pentru
    // această pagină.
    'editLink' => 'contact.php?l=' .
                  $GLOBALS[ 'now' ][ 'language' ] ,

    // Permite schimbarea căii.
    'allowSetPath' => true ,
),

// Paginile de tip blog. Vezi secțiunea 3.5.
'pages/blog.php' => array(
    // Pot fi șterse.
    'deletable' => true ,

    // Când se execută cererea de ștergere includem fișierul PHP indicat
    // aici...
```

```

'deleteInclude' => 'pages-blog.php',

// ... și executăm funcția ...
'deleteFunction' => 'adminBlogDeletePage',

// Propriul modul de administrare a blogului.
// Când utilizatorul face click pe această pagină se face redirectionare
// la blog.php unde apare lista de articole din blog.
'editable' => 'link',
'editLink' => 'blog.php?l=' .
    $GLOBALS['now']['language'] .
    '&cat=',

// Aici indicăm dacă dorim să avem calea paginii transmisă la scriptul
// dat de editLink. Astfel blogurile pot să aibă articole diferite
// pentru fiecare secțiune din site.
'editParam' => true,

// Jeton care indică dacă se pot adăuga alte module de același tip.
// Vezi $GLOBALS['add_modules'].
'add_module' => 'blog',

// Permite schimbarea căii pentru bloguri, dar trebuie executată
// funcția dată de setPageFunction din fișierul indicat
// de setPageInclude.
'allowSetPath' => true,
'setPageFunction' => 'adminBlogSetPage',
'setPageInclude' => 'pages-blog.php',
),

// Paginile de tip galerie de imagini. Vezi secțiunea 3.6.
'pages/gallery.php' => array(
    // Aproape tot este identic cu modulul blog.
    'deletable' => true,
    'deleteInclude' => 'pages-gallery.php',
    'deleteFunction' => 'adminGalleryDeletePage',
    'editable' => 'link',
    'editLink' => 'gallery.php?l=' .
        $GLOBALS['now']['language'] .
        '&cat=',
    'editParam' => true,

```

```

    'add_module' => 'gallery',
    'allowSetPath' => true,
    'setPageFunction' => 'adminGallerySetPage',
    'setPageInclude' => 'pages-gallery.php',
),

// Paginile cu fișiere de descărcat. Vezi secțiunea 3.7.
'pages/downloads.php' => array(
    'deletable' => true,
    'deleteInclude' => 'pages-downloads.php',
    'deleteFunction' => 'adminDownloadsDeletePage',
    'editable' => 'link',
    'editLink' => 'downloads.php?l=' .
        $GLOBALS['now']['language'] .
        '&cat=',
    'editParam' => true,
    'add_module' => 'downloads',
    'allowSetPath' => true,
    'setPageFunction' => 'adminDownloadsSetPage',
    'setPageInclude' => 'pages-downloads.php',
),
);

// Dacă un element <menu include="1"> are un fișier necunoscut
// folosim setările de mai jos.
$GLOBALS['unknown_modules'] = array(
    'deletable' => false,
    'editable' => false,
    'allowSetPath' => false,
);

// La opțiunea de adăugare pagini utilizatorul poate selecta tipul de
// pagină. Lista de mai jos indică tipurile de pagini care pot fi adăugate.
$GLOBALS['add_modules'] = array(
    0 => 'simple',
    'simple' => true,
    'blog' => array(
        // Fișierul care apare în <file> când pagina de tip blog este adăugată.
        'file' => $pages_path . '/blog.php',
        // Funcția chemată când se adaugă o pagină de acest tip.
        'callFunc' => 'adminBlogAddPage',
    )
);

```

```

// Funcția este luată din fișierul...
'callFuncFile' => 'pages-blog.php',

// Indicăm dacă acest modul dorește să preia controlul căii.
// De exemplu: /pagina1/subpagina2/blog3/articol.
// Motorul siteului nu va încerca să caute articol în structura XML,
// ci va da control modulului executat.
'subpath' => true,
),

'gallery' => array(
// Paginile de tip galerie de imagini.
'file' => $pages_path . '/gallery.php',
'callFunc' => 'adminGalleryAddPage',
'callFuncFile' => 'pages-gallery.php',
'subpath' => true,
),

'downloads' => array(
// Paginile cu fișiere de descărcat.
'file' => $pages_path . '/downloads.php',
'callFunc' => 'adminDownloadsAddPage',
'callFuncFile' => 'pages-downloads.php',

// Nu avem „directoare virtuale” în acest tip de pagini.
'subpath' => false,
),
'link' => true,
'template' => true,
);

```

După cum se poate vedea din codul de mai sus și din comentariile explicative, lista paginilor este generată dinamic în funcție de mulți factori. Codul este extensibil, fiind permisă implementarea unei game largi de module, de funcționalități noi în site.

Adâncimea maximă de pagini în ierarhie este dată de `$max.depth`. Această valoare este aleasă în funcție de designul siteului. Unele interfețe de site permit o ierarhie mai adâncă, unele mai puțin. Pentru siteul uav.ro folosim adâncimea maximă de trei pagini, conform limitelor impuse de design.

3.3.3 Adăugarea de pagini

Administratorul cu drepturi suficiente are posibilitatea să adauge pagini noi în una sau mai multe secțiuni ale siteului. Formularul de adăugare arată astfel:

Figura 3.11: Adăugare pagină nouă.

The screenshot shows the 'Adăugare pagină' form within the University of Arad website administration interface. The form is titled 'Adăugare pagină' and contains the following fields and options:

- *Titlu:** A text input field for the page title.
- Calea:** A text input field for the page URL.
- Notă:** A note stating: 'Calea este folosită pentru generarea adresei acestei pagini. Exemplu: <http://www.uav.ro/ro/nume-pagina>. Lăsați acest câmp gol pentru generare automată în funcție de titlul paginii.'
- *Modul/tip de pagină:** A dropdown menu with the selected option 'Un singur articol'.
- ☐ **Pagină ascunsă**: A checkbox to mark the page as hidden.
- Buttons:** 'Adaugă' (Add) and 'Anulează' (Cancel).

The form is displayed on a page titled 'Pagini' (Pages), which includes a sidebar with a tree view of the site structure. The sidebar shows the following hierarchy:

- Acasă
- Căutare
- Hartă site
- Administrare
- Universitate
 - Prezentarea Universității
 - Locații: Adrese
 - Istoric
 - Misiunea UAV
 - Carta Universității
 - Planul strategic: 2008-2012
 - Parteneri
 - Doctor Honoris Causa
 - Cartea UAV 2009-2010
- Conducerea Universității
 - Cuvântul Rectorului
 - Cuvântul Președintelui Interimar
 - Senatul UAV
- Documente de interes public
 - Etica universitară

Câmpurile din formular sunt:

Titlul este titlul paginii.

Calea este calea paginii. Acest câmp este opțional. Valoarea trebuie să fie un jeton din caractere alfanumerice. Dacă nu se introduce

nici o valoare, atunci codul PHP generează calea paginii pe baza titlului.

Modul/tipul de pagină permite alegerea tipului de pagină adăugată.

Pagină ascunsă permite ascunderea paginii din lista de meniuri. Pagina poate fi accesată dacă adresa ei este știută.

În navigatorul web se folosesc funcții JavaScript din fișierul `js/admin.js`: `siteAdmin.pageAdd()` și `pageAddSubmit()`. Acestea afișează formularul și îl validează înainte de a fi trimis la server.

Pe server, în funcție de opțiunile alese de utilizator, se construiește un element `<menu>` care este adăugat în rădăcina dorită, într-un element `<menus>`. Structura **DOM** actualizată este salvată înapoi în fișierul XML.

Noțiunea de module pentru site este implementată cu meniuri, pagini care includ fișiere PHP. Atunci când utilizatorul alege un modul defapt el alege fișierul PHP care urmează să fie inclus. Modulele următoare pot fi alese:

Un singur articol se referă la o singură pagină HTML care poate fi editată de administrator, fără PHP. În XML aceasta este reprezentată cu un `<menu>` care are atributul `include="0"` și un `<file>` care are calea către un fișier HTML de pe disc, în directorul `pages/`.

Blog se referă la scriptul `pages/blog.php`. Acest script este pus în tagul `<file>`, iar `<menu>` primește atributul `include="1"`. Pentru detalii despre modulul Blog vezi secțiunea 3.5.

Galerie de imagini se referă la scriptul `pages/gallery.php`. Acesta este adăugat la fel ca blogul. Pentru detalii despre acest tip de pagini vezi secțiunea 3.6.

Fișiere de descărcat se referă la scriptul `pages/downloads.php`. Acest tip de pagini permit utilizatorilor să descarce fișiere dintr-un director asociat. Pentru detalii vezi secțiunea 3.7.

Legătură spre o pagină externă adaugă doar un element `<menu>` cu un `<link>`.

Șablon: `nume-șablon` adaugă un articol, o pagină HTML, dar preia un conținut predefinit. Lista de șabloane este preluată din directorul `files/page_templates/`. Fiecare fișier HTML din director este afișat în această listă. Când utilizatorul alege un șablon acel fișier este preluat în noua pagină.

Jetonul unic de identificare al paginii și calea paginii, dacă nu este specificată, sunt ambele generate pe baza titlului. În acest scop folosim funcția `AdminGenerateId()` din `admin/include.php`:

```
function AdminGenerateId($txt)
{
    $id = mb_strtolower($txt);
    // Transliterația caracterelor, pentru a elimina diacriticele.
    setlocale(LC_ALL, 'en');
    $id = @iconv('utf8', 'ascii//TRANSLIT', $id);
    setlocale(LC_ALL, $GLOBALS['now']['language']);

    $id = preg_replace('/[^a-z0-9\_-]/', '-', $id);
    $id = trim($id, '-');
    $id = preg_replace('/\--+/', '-', $id);
    $id = preg_replace('/^\d+/', '', $id);
    $id = trim($id, '-');

    // Dacă nu rămân suficiente caractere, folosim data curentă.
    if(!isset($id{1}))
        $id = 'id'.date('Y-m-d-H-i-s');

    return $id;
}
```

După ce calea și ID-ul paginii sunt generate, se crează un nou element `<menu>` în care se introduc atributele și restul de elemente în funcție de alegerile utilizatorului. Pentru articole simple se crează un fișier HTML nou în directorul `pages/`. Pentru celelalte module se fac chemări la funcțiile lor specifice, așa cum s-a putut vedea în secțiunea [3.3.2](#).

La final, structura XML a siteului este salvată înapoi pe disc în fișierul din directorul `xml/`.

3.3.4 Setări pagină

Setările paginii pot fi modificate ușor cu formularul prezentat mai jos:

Figura 3.12: Setări pagină.

Language: Română • English Utilizator robod (ROBO Design) • Ieșire • Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Prima pagină | Recomandări | Fișiere | **Pagini** | Utilizatori | Profil | Log out

Pagini

Acasă

Căutare

Hartă site

Administrare

Universitate

Prezentarea Universității

Locații: Adrese

Istoric

Misiunea UAV

Carta Universității

Planul strategic: 2008-2012

Parteneri

Doctor Honoris Causa

Cartea UAV 2009-2010

Conducerea Universității

Cuvântul Rectorului

Cuvântul Președintelui Interimar

Senatul UAV

Documente de interes public

Etica universitară

Setări pagină

*Titlu:

Calea:

Notă: Calea este folosită pentru generarea adresei acestei pagini. Exemplu: <http://www.uav.ro/ro/nume-pagina>. Lăsați acest câmp gol pentru generare automată în funcție de titlul paginii.

☐ **Pagină ascunsă**

Administratorul poate modifica titlul și calea paginii. Dacă pagina are sub-pagini atunci calea nu poate fi modificată. Pagina poate fi ascunsă sau afișată în lista de meniuri.

Toate modificările în structura XML sunt făcute cu operații **DOM** în scriptul `admin/pages.php`. De asemenea, pentru blog, galerie și alte tipuri de pagini se fac chemări la funcțiile lor specifice de actualizare titlu și cale.

3.4 Editorul de pagini Awebitor

Awebitor (*Advanced Web Editor*) este numele dat editorului de **HTML** realizat pentru uav.ro. Acesta permite editarea paginilor fără a fi necesare cunoștințe HTML:

Figura 3.13: Awebitor: editorul de pagini.



Bara de unelte a editorului oferă următoarele opțiuni:

- Înapoi și înainte în timp (*undo* și *redo*).
- Vizualizare cod sursă HTML.
- Adăugare tabel.
- Adăugare legătură către o adresă externă.
- Eliminare legătură – se șterge orice legătură în textul selectat.

- Adăugare imagine de pe server.
- Linie orizontală.
- Eliminare formatare text – textul selectat poate fi curățat de stil, de culori, etc.
- Alegere tip element: paragraf, titlu 1, titlu 2 și altele.
- Alegere font și mărime text.
- Text aldin (*bold*), italic și tăiat.
- Indice și exponent.
- Culoare text și fundal.
- Aliniere la stânga, dreapta și *justify*.
- Liste ordonate cu numerotare.
- Liste neordonate cu *bullets*.

Implementarea editorului se găsește în fișierul `js/awebitor.js`. Modul de funcționare:

1. Pagina cu editorul se încarcă. Formularul conține un `<textarea>` în care este pus tot conținutul HTML al paginii alese de administrator.
2. Se execută codul **JavaScript** al editorului Awebitor.
3. Elementul `<textarea>` este ascuns pentru a fi schimbat cu interfața editorului.
4. Interfața editorului din fișierul `admin/awebitor-interface.php` este încărcată folosindu-se un `XMLHttpRequest` [2]. Scriptul **PHP** generează codul **XHTML** care este adăugat în **DOM**-ul paginii unde rulează Awebitor.
5. Se adaugă un element `<link>` cu atributul `rel="stylesheet"` și calea către stilul `css/awebitor.css`. Acest stil **CSS** este folosit pentru interfața editorului.
6. Se adaugă un element `<iframe>` cu calea către `awebitor_iframe.php`. Acest script PHP generează o pagină goală cu puțin cod JavaScript care poate transmite mesaje documentului părinte, către pagina care include acest `<iframe>`.

7. La încărcarea `<iframe>`-ului se introduce codul HTML din `<textarea>` în `<iframe>`.
8. Se activează `document.designMode="on"` în `<iframe>`. Din acest moment utilizatorul poate edita pagina dorită. Încărcarea s-a încheiat.

Pentru editarea unei pagini simple, a unui articol, se folosește scriptul `admin/edit.php` care generează formularul cu elementul `<textarea>`. La încărcarea acestei pagini în `js/admin.js` există funcția `siteAdmin.awebitortaLoaded()` care se execută. Această funcție pornește inițializarea editorului.

În funcția `awebitor.init()` se fac verificări dacă navigatorul web are suport pentru tehnologiile web necesare, de exemplu pentru `designMode` [3]. În final funcția `awebitor.loadInterface()` este chemată. Această funcție inițializează un obiect `XMLHttpRequest` pentru a încărca interfața editorului. La încărcare este chemată funcția `awebitor.interfaceReady()`. Dacă fișierul XHTML s-a încărcat cu succes, atunci se face chemare la `awebitor.init_elems()` pentru a face adăugarea propriu-zisă a tuturor elementelor din fișierul luat în pagina curentă.

Funcția `awebitor.init_frame()` inițializează `<iframe>`-ul și face ca toate butoanele din interfață să funcționeze – sunt adăugate evenimentele dorite.

Pentru integrarea cu siteul de administrare uav.ro există funcții specifice introduse în Awebitor. Acestea se găsesc în `awebitor.customFunctions`:

`preInitElems` funcțiile din acest obiect se execută după ce se încarcă interfața editorului, înainte să fie elementele ei adăugate în pagină. Aici avem o funcție pentru opțiunea de adăugare imagini.

`postInitIframe` funcțiile din acest obiect se execută după ce este inițializat editorul cu tot cu `<iframe>`. Aici avem trei funcții: pentru opțiunea de alegere culori, pentru selectorul de imagini de pe server, și încă una pentru butoanele de salvare document și anulare editare.

Majoritatea opțiunilor din editor folosesc funcția DOM `document.execCommand()`. De exemplu, pentru a face textul selectat să fie aldin, se execută `document.execCommand("bold")`.

Adăugarea de imagini permite selectarea fișierului dorit de pe server:

Figura 3.14: Adăugarea de imagini în pagini.

Language: Română • English Utilizator robot (ROBO Design) • Ieșire • Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Prima pagină Recomandări Fișiere **Pagini** Utilizatori Profil Log out

Adaugă imagine

*Adresa imaginii: Aliniere:

Textul alternativ al imaginii:

Legătură externă:

Directorul actual: / files / imagini

☐ Nume fișier	Ultima modificare	Mărime
☐ anosr.png	2010-05-07 09:46:01	13.47 KB
☒ bologna.png	2010-05-07 10:11:00	56.87 KB
☐ pag1.jpg	2010-04-14 06:20:48	564.46 KB
☐ pag2.jpg	2010-04-14 06:20:51	472.34 KB

Inversare selecție

Operații

- Șterge
- Redenumeste
- Adaugă director
- Curățare pagină

Adaugă fișiere

+ Parcourir...

După cum se poate vedea din captura de ecran de mai sus, utilizatorul are un formular de adăugare imagine unde se poate introduce adresa imaginii, titlul care o descrie și, opțional, o legătură spre o pagină externă. Câmpul *Aliniere* permite administratorului să aleagă unde să fie poziționată poza: la stânga sau la dreapta.

Pentru a face mai ușoară adăugarea de imagini modulul de gestionare a fișierelor (vezi secțiunea 3.2) este inclus în formular – utilizatorul poate să aleagă fișierul dorit direct de pe server, sau poate să încarce o poză nouă în directorul dorit. Scriptul `admin/files.php` este inclus într-un `<iframe>`: aproape tot funcționează la fel, dar diferența importantă este că nu se mai încarcă toată interfața siteului în `<iframe>`, ci apare doar lista de fișiere.

La finalul editării documentului, când utilizatorul alege să salveze modificările, se transmite formularul la server. Funcția `awebitor.formSubmit()`

preia evenimentul `submit` al formularului pentru a actualiza conținutul elementului `<textarea>`. Pe server se salvează noul conținut HTML în fișierul paginii.

3.4.1 Curățarea codului HTML cu ReTidy

Oriunde este folosit Awebitor în siteul de administrare apare și opțiunea „profilul de curățare a paginii HTML”. Această opțiune permite utilizatorului să curețe codul **HTML** generat de navigatorul web – să elimine *tagurile* inutile.

Importanța curățării unei pagini crește atunci când conținutul este copiat dintr-un procesor de documente precum **OpenOffice**¹ Writer sau Microsoft Office Word. De obicei asemenea documente conțin foarte multe taguri inutile și chiar multe erori în codul HTML generat.

Pentru a curăța documentele de erori am scris un script **PHP** *open source* numit ReTidy.

Funcția `clean_awebitor()` este chemată la salvarea paginilor. Aceasta se găsește în `admin/include.php`:

```
// Includem scriptul PHP și pregătim o instanță a clasei ReTidy.
require_once($GLOBALS[ 'retidy.php' ] );
$GLOBALS[ 'retidy' ] = new ReTidy();
```

```
function clean_awebitor($html, $profile = false)
{
    // Profilul dorit pentru curățare.
    if(! $profile)
    {
        $profile = 'awebitor';
        if(isset($_REQUEST[ 'retidyp' ]))
            $profile = $_REQUEST[ 'retidyp' ];

        if($profile == 'n')
            return $html;
    }

    // Încărcare profil.
    if(! $GLOBALS[ 'retidy' ]->loadProfile($profile))
        return false;
```

¹<http://openoffice.org>

```

    $html = '<html>
<head>
<meta http-equiv="Content-Type"
        content="text/html; charset=utf-8" />
</head>
<body>' . $html . '</body>
</html>';

    // Curățare pagină.
    $GLOBALS['retidy']->setCode($html);
    if (!$GLOBALS['retidy']->cleanCode())
        return false;

    $html = $GLOBALS['retidy']->getCode();
    $GLOBALS['retidy']->free_mem();

    $html = preg_replace(array('/^(.+?)<body[^\>]*>/is ',
                                '/<\</body>(.+?)$/is '),
                        '', $html);

    return $html;
}

```

Profilurile de curățare a paginilor sunt doar fișiere diferite de configurare pentru ReTidy. Acesta are o mulțime de opțiuni de configurare a curățării: ce taguri să fie șterse, condițiile în care să fie șterse, attributele care sunt șterse, procesarea textului și multe altele.

Profilul „Awebitor” este selectat implicit, face o curățare minimală documentelor. Profilul „maxim” este făcut să curețe paginile care sunt preluate din documente Office. În asemenea cazuri trebuie să fie eliminate foarte multe taguri.

```

// Bucată de cod din profilul Awebitor.
return array(
    'encoding' => 'utf-8',

    // Lista de funcții de curățare ce sunt chemate în ordinea dată.
    // Acestea pot fi chemate de oricâte ori este necesar.
    'macro' => array('tidy_code', 'my_strip_tags',
                    'strip_br_dupes', 'trim_br_tags', 'replace_tags',
                    'tidy_code', 'dom_init', 'dom_strip_child_tags',
                    'dom_strip_only_child',

```

```

    'dom_strip_parent_only_child',
    'dom_merge_parent_attr',
    'dom_save', 'post_dom_stripme',
    'post_dom_renametag', 'strip_empty_tags',
    'combine_inline',
    'reorder_tags', 'combine_inline',
    'combine_br_tags',
    'trim_br_tags', 'fix_img_pos', 'tidy_code',
    'strip_lang', 'trim_br_tags'),

// Funcții importante care nu trebuie să se execute cu erori.
// Dacă apar erori (return false) în execuția acestora, se oprește
// curățarea paginii.
'important_methods' => array('tidy_code', 'dom_init',
                              'dom_save'),

// Elimină <br>-urile duplicate.
'strip_br_dupes' => true,

// Elimină orice <br> găsit la începutul sau la finalul tagurilor
// date în lista de mai jos.
'trim_br_tags' => array('p', 'div', 'table', 'th',
                        'td', 'h1', 'h2', 'h3', 'h4', 'h5', 'h6', 'ul',
                        'ol', 'dl', 'li', 'dt', 'dd', 'strong', 'em',
                        'i', 'u', 'b', 'tr', 'pre'),

// Elimină tagurile goale.
'strip_empty_tags' => array('p', 'title', 'style',
                            'h1', 'h2', 'h3', 'h4', 'h5', 'h6', 'b', 'i',
                            'em', 'strong', 'sup', 'sub', 'u', 'td', 'tr',
                            'table'),

// ... și multe alte opțiuni.
);

```

Pentru a curăța o pagină ReTidy preia codul și îl trece prin funcții de curățare, așa cum se cere de profil. Aceste funcții pot lucra pe codul sursă HTML sau pe structura DOM a paginii. Pe lângă aceste funcții există și funcția de curățare cu HTML Tidy [9] – un proiect *open source* care permite curățarea paginilor HTML, dar care menține prea mult din codul inițial al paginii. Rolul trecerii codului HTML prin HTML Tidy este de a face

curățarea inițială. După acest prim pas, restul de funcții de curățare au mai puțin de modificat.

3.5 Blogul

Siteul poate avea oricâte bloguri se dorește. Blogurile sunt tipuri de pagini cu articole sortate cronologic, de la cel mai nou la cel mai vechi, cu etichete alese de administratori, și cu fluxuri **Atom**. Pentru a vedea cum se pot adăuga pagini de tip blog vedeți secțiunea 3.3.3.

3.5.1 Funcționalități pentru utilizatori

Blogul **Evenimente**² arată în felul următor:

Figura 3.15: Blogul Evenimente.

Language: Română • English Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Universitate Facultăți Academic Studenți Cercetare Departamente Reviste Resurse Personal

Prezentarea Universității
Conducerea Universității
Documente de interes public
Legislație
Clipuri Video
Galerie foto
Noutăți
• **Evenimente**
Ranking
Links

Evenimente
Universitate a- A+ P

Lansare de carte, joi, 14.04.2011, ora 11
13 April 2011, 09:27
Eveniment cultural: Lansarea cărții „Aradul teatral 1752 – 2010”, autor Lizica Mihut.
Data și locația: Joi, 14.04.2011, ora 11. Clădirea Rectoratului, sala 13 (sala Senatului UAV).
Vor conferenția: Gelu Negrea, Ion Cocora, Vasile Dan, Florin Didilescu, Ovidiu Cornea, Diana Achim ș.a.
Detalii la <http://arq.ro/> și la <http://www.uav.ro/files/comunicate/art13.04.pdf>

ICRSMS 2011
8 April 2011, 06:32
2nd INTERNATIONAL CONFERENCE OF THE ROMANIAN SOCIETY FOR MASS SPECTROMETRY, ICRSMS 2011, May 1-5, 2011, Chemistry Institute of the Romanian Academy, Timisoara
Organized by:
• [Romanian Academy](#)
• [Aurel Vlaicu University of Arad](#)
• [Romanian Society for Mass Spectrometry](#)
Conference Chair:
[Alina D. Zamfir](#)
E-mail: rsms@uav.ro
URL: <http://www.uav.ro/en/conferences/icrsms-2011>

Pagini din istoria informaticii românești
29 March 2011, 08:20
Pagini din istoria informaticii românești

Arhivă
Lansare de carte, joi, 14.04.2011, ora 11
ICRSMS 2011
Pagini din istoria informaticii românești
Două evenimente importante (vineri -18 martie a.c.)
Concursul Internațional de Matematică "Caius Iacob"- ediția a 3-a
Constantin Karadja - Un diplomat european
Zilele Eminescu 2011
Etichete populare
academia, botosani, caius iacob, canepa, concurs, constantin karadja, concurs de matematica, critic de teatru, diplomat european, doctor honoris causa.

²<http://www.uav.ro/ro/universitate/evenimente>

Paginile de acest tip sunt generate de scriptul `pages/blog.php`. Prima pagină a oricărui blog listează ultimele articole, câteva articole din arhivă și cele mai populare etichete.

Administratorii pot asocia etichete arbitrar alese fiecărui articol, separate de virgulă. Acestea permit utilizatorilor să filtreze articolele de pe site doar pe etichetele/categoriile dorite. De exemplu, se pot vizualiza doar articolele despre examene.

Pentru designul grafic al paginilor de tip blog se încarcă fișierul `css/blog.css`.

Blogurile pot avea un antet **HTML** și un panou lateral editate de administratori. Codul din `blog.php`:

```
// Directorul unde se găsesc fișierele blogului.
$folder_blog = 'pages/blog';

// Jetonul de identificare al blogului.
$blog_id = str_replace('/', '-', $blog_path);

// Antetul blogului. De exemplu, pentru blogul Evenimente fișierul este
// pages/blog/universitate-blog-header.html.
$file_header = $GLOBALS['php_base'] . '/' .
               $folder_blog . '/' . $blog_id .
               '-header.html';

// Codul HTML din panoul lateral al blogului.
$file_panel = $GLOBALS['php_base'] . '/' .
              $folder_blog . '/' . $blog_id .
              '-panel.html';
```

Lista de articole și etichetele acestora sunt salvate în tabelele **MySQL** `blog` (vezi tabelul A.1) și `blog_tags` (vezi tabelul A.2).

Pentru a afișa lista de articole în mini arhiva din panoul lateral al paginii se execută:

```
echo "<ol>\n";

// Comanda SQL
$result = mysql_query('SELECT 'path', 'title' FROM ' .
                      $table_blog . ' WHERE 'category '=' ' .
                      mysql_real_escape_string($blog_path) . ' ' .
                      ORDER BY 'pubdate' DESC LIMIT 0, 10');
```

```

while($row = mysql_fetch_assoc($Qresult))
{
    $link = $GLOBALS['permanent_links'][$'prefix'] . '/' .
        $blog_path . '/' . $row['path'];
    echo '<li><a href="' . $link . '">' . $row['title'] .
        "</a></li>\n";
}

echo "</ol>\n";

```

Afișarea celor mai populare etichete are la bază următorul cod:

*// Comanda SQL care preia lista de etichete asociate cu orice articol din
 // blogul curent. Coloana **cnt** conține numărul de articole care au
 // eticheta dată de coloana **tag**.*

```

$Qresult = mysql_query('SELECT COUNT( 'bid' )
    AS 'cnt', 'tag' FROM ' . $table_blog_tags . ',
    ' . $table_blog . '
    WHERE ' . $table_blog . ' . 'id' = 'bid' AND
    ' . $table_blog . ' . 'category' = ' ' .
    mysql_real_escape_string($blog_path) . '
    GROUP BY 'tag' ORDER BY 'cnt' DESC, 'tag' ASC
    LIMIT 0, 10');

$tags = array();
if (mysql_num_rows($Qresult) > 0) {
    while ($row = mysql_fetch_assoc($Qresult)) {
        $tags[] = $row;
    }

    $res = '';
    foreach ($tags as $tag) {
        $size = max(2.5 - $tags[0]['cnt'] / $tag['cnt'],
            0.7);
        $res .= ', <a rel="tag" style="font-size:' .
            $size . 'em" href="' .
            $GLOBALS['permanent_links'][$'prefix'] . '/' .
            $blog_path . '/tags/' . $tag['tag'] . '">' .
            $tag['tag'] . '</a>';
    }
    $res{0} = ' ';
}

```

```

    echo '<div id="blog-popular-tags">
<h2>' . $lang['popular-tags'] . '</h2>
<p>' . $res . "</p>
</div> <!-- #blog-popular-tags -->\n";
}

```

Conținutul articolelor este dat de codul următor:

```

$Qresult = mysql_query( 'SELECT 'pubdate', 'path',
    'content', 'title', 'id'
    FROM ' . $table_blog . '
    WHERE 'category'=' ' .
    mysql_real_escape_string($blog_path) . '
    ORDER BY 'pubdate' DESC LIMIT 0, 5 ');

$i = 0;

// Comenzi din tastatură pentru primele patru articole.
$accesskeys = array('a', 'b', 'c', 'd');

while($row = mysql_fetch_assoc($Qresult)) {
    $link = $GLOBALS['permanent_links']['prefix'] .
        '/' . $category . '/' . $row['path'];
    $date = date('j F Y, H:i', $row['pubdate']);

    echo '<h2 class="blog-post-title"><a';

    if(isset($accesskeys[$i]))
        echo ' accesskey="' . $accesskeys[$i] . '"';

    echo ' href="' . $link . '">' . $row['title'] .
        '</a></h2>
<p class="blog-post-date">' . $date . '</p>
<div class="blog-post-content">' . "\n";

    readfile($GLOBALS['php_base'] . '/' .
        $folder_blog . '/' . $row['content']);

    echo "</div> <!-- .blog-post-content -->\n";

    $i++;
}

```

Când utilizatorul alege un articol, doar conținutul celui articol este afișat.

Figura 3.16: Vizualizarea unui articol în blogul Evenimente.

Language: Română • English Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Universitate Facultăți Academic Studenți Cercetare Departamente Reviste Resurse Personal



Prezentarea Universității
Conducerea Universității
Documente de interes public
Legislație
Clipuri Video
Galerie foto
Noutăți
• **Evenimente**
Ranking
Links

ICRSMS 2011
Evenimente • Universitate

2nd INTERNATIONAL CONFERENCE OF THE ROMANIAN SOCIETY FOR MASS SPECTROMETRY, ICRSMS 2011,
May 1-5, 2011, Chemistry Institute of the Romanian Academy, Timisoara

Organized by:

- [Romanian Academy](#)
- [Aurel Vlaicu University of Arad](#)
- [Romanian Society for Mass Spectrometry](#)

Conference Chair:
[Alina D. Zamfir](#)
E-mail: rsms@uav.ro

URL: <http://www.uav.ro/en/conferences/icrsms-2011>

Articolul a fost publicat în: 8 April 2011, 06:52.

[Înapoi la pagina principală](#)

[Articol mai vechi](#) [Articol mai nou](#)

Arhivă

Lansare de carte, joi, 14.04.2011, ora 11

ICRSMS 2011

Pagini din istoria informaticii românești

Două evenimente importante (vineri -18 martie a.c.)

Concursul Internațional de Matematică "Caius Iacob"- ediția a 3-a

Constantin Karadja - Un diplomat european

Zilele Eminescu 2011

Vezi arhiva

Etichete populare

academia, botosani, caius iacob, canepa, concurs, constantin karadja, concurs de matematica, critic de teatru, diplomat european, doctor honoris causa.

© 2011 Universitatea „Aurel Vlaicu” din Arad. Toate drepturile rezervate. Site realizat de ROBO Design. Coordonator Prof. Dr. Ioan Dzitac.

Pentru a vedea toate articolele din blog, inclusiv cele mai vechi articole care nu apar pe prima pagină, utilizatorul poate accesa arhiva de articole. Acestea sunt grupate vizual pe ani și pe luni.

Figura 3.17: Arhiva de articole într-un blog.

Language: Română • English Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Universitate Facultăți Academic Studenți Cercetare Departamente Reviste Resurse Personal



Prezentarea Universității

Conducerea Universității

Documente de interes public

Legislație

Clipuri Video

Galerie foto

Noutăți

• Evenimente

Ranking

Links

Arhivă

Evenimente • Universitate a- A+ P

[Înapoi la pagina principală](#)

2011

Mai

- [Lansare de carte, joi, 14.04.2011, ora 11](#)
- [ICRSMS 2011](#)

Aprilie

- [Pagini din istoria informaticii românești](#)
- [Două evenimente importante \(vineri -18 martie a.c.\)](#)
- [Concursul Internațional de Matematică "Caius Iacob"- ediția a 3-a](#)
- [Constantin Karadja - Un diplomat european](#)

Februarie

- [Zilele Eminescu 2011](#)

2010

Decembrie

- [Valorificarea costumului popular din Țara Zărandului în moda contemporană din in și cânepă](#)

August

- [Topul Integrității Universităților Publice \(C.U.C. 2010\)](#)
- [Un nou proiect lansat de UAV](#)
- [Premii pentru cercetare](#)
- [UAV - într-un proiect FP7](#)

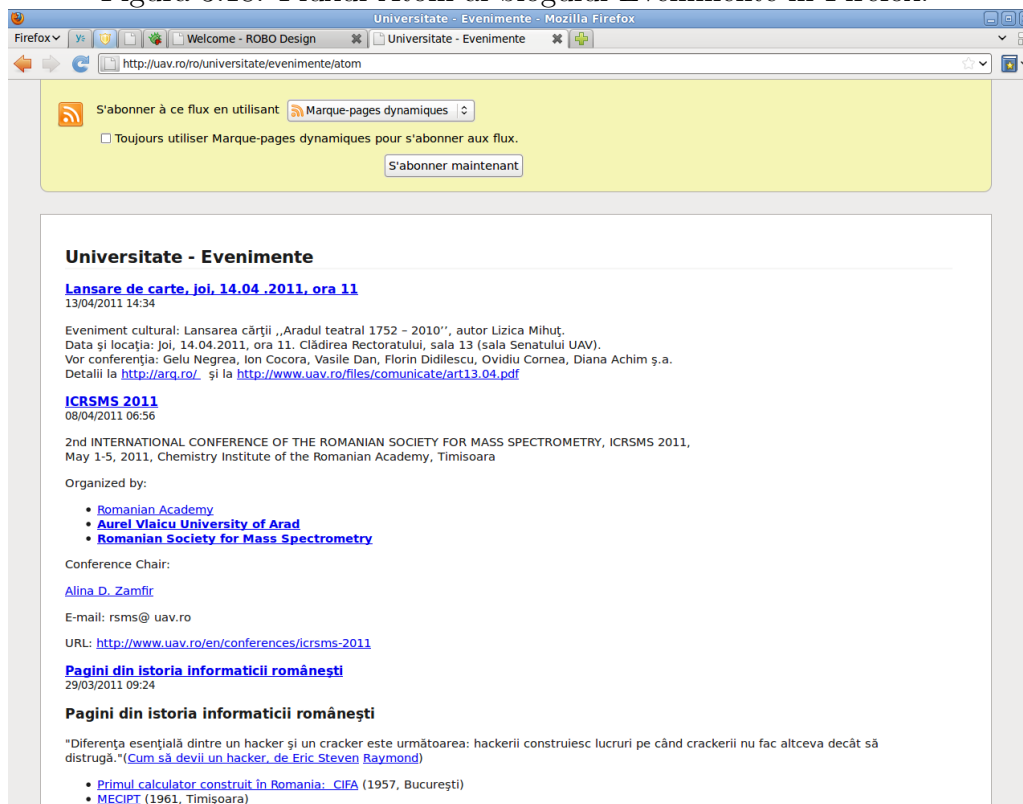
Etichete populare

academia, botosani, caius iacob, canepa, concurs, constantin karadja, conurs de matematica, critic de teatru, diplomat european, doctor honoris causa.

© 2011 Universitatea „Aurel Vlaicu” din Arad. Toate drepturile rezervate. Site realizat de ROBO Design.
Coordonator Prof. Dr. Ioan Dzitac.

Pentru utilizatorii care doresc să fie la curent cu ultimele articole publicate pe site, există fluxuri **Atom**. Aceste fluxuri în format **XML** pot fi citite de toate navigatoarele web moderne și de alte programe specializate. Când utilizatorul este abonat la fluxul unui blog, acesta este notificat automat dacă apar articole noi.

Figura 3.18: Fluxul Atom al blogului Evenimente în Firefox.



În captura de ecran de mai sus se pot vedea articolele blogului afișate de Firefox. Tot prin abonarea la fluxul Atom articolele pot fi afișate pe alte siteuri sau alte aplicații software. Codul **XML** este generat de `pages/blog-atom.php`. Exemplu de flux Atom:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xml:lang="ro"
      xmlns="http://www.w3.org/2005/Atom">
  <title>Universitate – Evenimente</title>
  <author>
    <name>Universitatea "Aurel Vlaicu" din Arad</name>
    <uri>http://www.uav.ro</uri>
```

```

</author>
<generator uri="http://www.robodesign.ro">
  ROBO Design v5
</generator>
<id>urn:www-uav-ro:universitate-evenimente</id>
<link href="/ro/universitate/evenimente"
      title="Universitate - Evenimente"
      type="text/html" />
<link rel="self"
      href="/ro/universitate/evenimente/atom"
      title="Universitate - Evenimente"
      type="application/atom+xml" />
<updated>2010-07-28T11:36:21+03:00</updated>

<entry>
  <title>Lansare de carte , joi , 14.04.2011 ,
    ora 11</title>
  <published>2011-04-13T09:27:44+03:00</published>
  <updated>2011-04-13T14:34:48+03:00</updated>
  <link href="/ro/universitate/evenimente/lansare-
de-carte-joi-14-04-2011-ora-11"
        title="Lansare de carte , joi , 14.04.2011 ,
ora 11" type="text/html" />
  <id>urn:www-uav-ro:universitate-
evenimente:448</id>
  <category term="academia" />
  <category term="critic de teatru" />
  <category term="lansare de carte" />
  <category term="teatru" />
  <content type="html">
    &lt;p&gt;
      &lt;span style="font-weight: bold;"&gt;
Eveniment cultural:&lt;/span&gt; Lansarea cartii
,,Aradul teatral 1752 - 2010'', autor Lizica Mihut.
      &lt;br /&gt;
      &lt;span style="font-weight: bold;"&gt;Data
si locatia:&lt;/span&gt; Joi , 14.04.2011 , ora 11.
Cladirea Rectoratului , sala 13 (sala Senatului UAV).
      &lt;br /&gt;
      &lt;span style="font-weight: bold;"&gt;Vor
conferentia:&lt;/span&gt; Gelu Negrea , Ion Cocora ,

```

Vasile Dan, Florin Didilescu , Ovidiu Cornea ,
Diana Achim s.a.

```

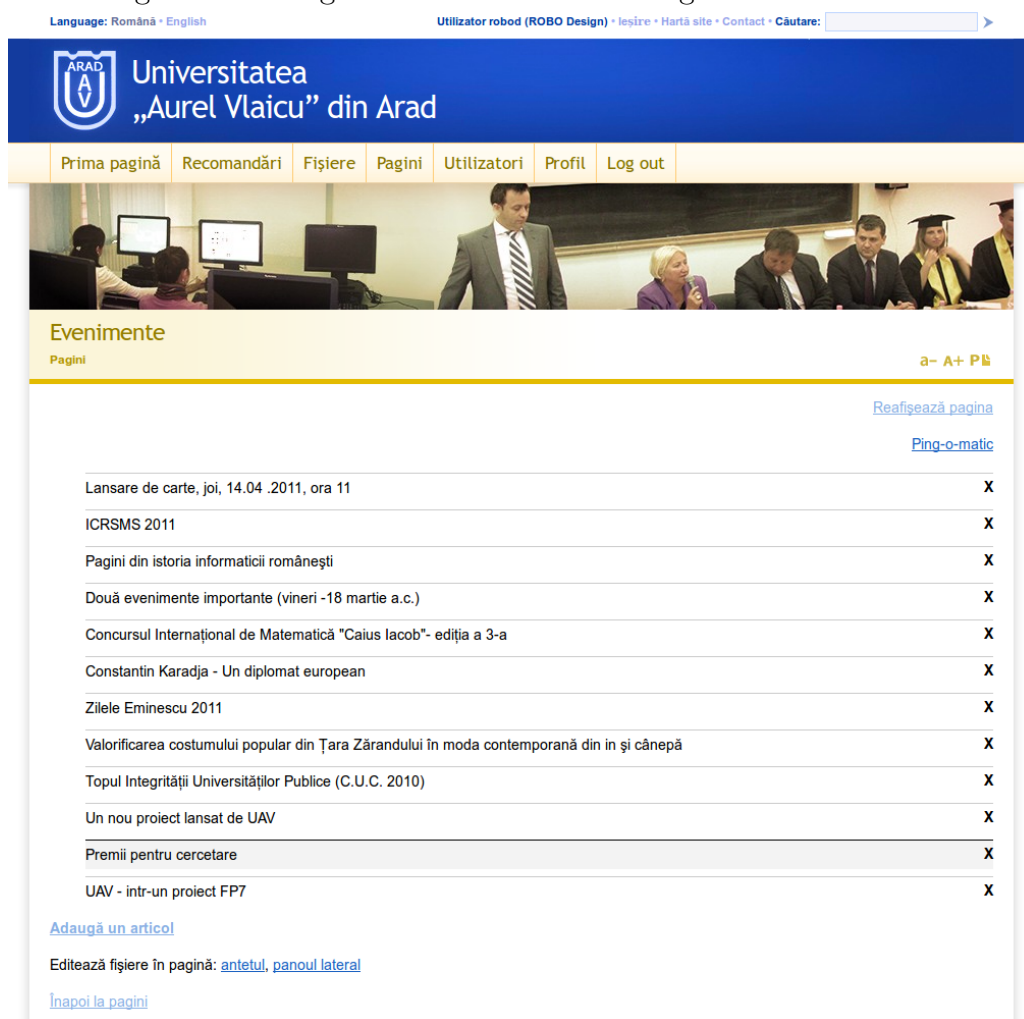
    <br />
    <span style="font-weight: bold;">
Detalii</span> la <a href="http://arq.ro/
lizica-mihut-aradul-teatral-1752-%E2%80%932010%E2%80
%99%E2%80%99/">http://arq.ro/&#160;<a>
&#160; si la <a href="files/comunicate/
art13.04.pdf">http://www.uav.ro/files/
comunicate/art13.04.pdf<a>
    </p>
    <p>
    <a href="http://www.uav.ro/files/
comunicate/art13.04.pdf"><a>
    </p>
</content>
</entry>
</feed>

```

3.5.2 Funcționalități pentru administratori

În gestionarul de pagini (vezi secțiunea 3.3) administratorul poate face *click* pe paginile de tip blog pentru editare. Editorul de bloguri este scriptul `admin/blog.php`. Articolele din blog sunt listate astfel:

Figura 3.19: Pagina de administrare a blogului Evenimente.



Listarea de articole se bazează pe o singură comandă **SELECT** din tabelul MySQL al blogului. Interacțiunea cu pagina web este implementată de codul JavaScript din `js/admin.js`.

Antetul blogului și panoul lateral pot fi editate cu Awebitor (vezi secțiunea 3.4), trebuie doar ca administratorul să facă un *click* pe opțiunea de editare acestor secțiuni ale blogului.

După ce se fac modificările dorite la articole, administratorul poate să acceseze opțiunea Ping-o-matic. Această opțiune este o legătură către serviciul [Ping-O-Matic](http://pingomatic.com)³. Adresa blogului este transmisă la Ping-O-Matic pentru a fi transmisă mai departe la alte motoare de căutare bloguri (inclusiv către Google). Motoarele de căutare indexează fluxul Atom și permit utilizatorilor să caute în conținutul acestuia.

Administratorul poate șterge articole printr-un singur *click* pe opțiunea „X” din dreapta fiecărui articol – moment în care se execută funcția `siteAdmin.blogDelete()`. Această funcție cere confirmarea acțiunii de ștergere. Pe server se execută:

```
// Scoatem articolul din tabelul blog.
mysql_query( 'DELETE FROM ' . $table_blog . ' '
    WHERE 'category=' . ' ' .
    mysql_real_escape_string($blog_path) . ' '
    AND 'id=' . ' ' . $REQUEST['id'] . ' ' );

// Ștergem etichetele asociate cu acest articol.
mysql_query( 'DELETE FROM ' . $table_blog_tags . ' '
    WHERE 'bid=' . ' ' . $REQUEST['id'] . ' ' );

// Ștergem fișierul HTML de pe disc.
if( @file_exists($base . '/' . $folder_blog . '/' .
    $row['content']))
    @unlink($base . '/' . $folder_blog . '/' .
    $row['content']);

// Ștergem pagina din indexul de căutare.
if( $use_search )
    searchDeleteDocument(
        $GLOBALS['permanent_links']['prefix2'] . '/' .
        $blog_path . '/' . $row['path']);

// Ștergem pagina din harta siteului.
if( $use_sitemap )
    $sitemap->delete(
        $GLOBALS['permanent_links']['prefix2'] . '/' .
        $blog_path . '/' . $row['path']);

// Ștergem pagina din cache.
```

³<http://pingomatic.com>

Formularul de adăugare arată astfel:

Figura 3.20: Adăugarea unui articol în blog.

Administratorul trebuie să introducă titlul articolului, opțional calea acestuia, data și etichetele. Implicit articolul are data curentă. Etichetele trebuie să fie separate cu virgule.

Funcția JavaScript care afișează formularul este `siteAdmin.blogAdd()`. Validarea este făcută de `siteAdmin.blogAddSubmit()`.

Pe server se generează calea articolului pe baza titlului dacă administratorul nu a introdus nici o valoare – în acest scop este chemată funcția

`AdminGenerateId()` (vezi pagina 73). În final se adaugă articolul în tabelul `blog` cu o comandă `INSERT`. În tabelul `blog_tags` este adăugată fiecare etichetă.

Editarea unui articol se face cu un *click* pe titlul acestuia:

Figura 3.21: Editarea unui articol din blog.

Language: Română • English Utilizator robod (ROBO Design) • Ieșire • Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Prima pagină | Recomandări | Fișiere | Pagini | Utilizatori | Profil | Log out

ICRSMS 2011
Evenimente • Pagini

a- A+ P

***Titlu:**
ICRSMS 2011

Calea:
icrsms-2011

Notă: Calea este folosită pentru generarea adresei acestui articol. Exemplu: <http://www.uav.ro/ro/titlu-articol/>. Lăsați acest câmp gol pentru generare automată în funcție de titlul articolului.

***Data și ora:**
2011-04-08 06:52:30

Notă: Formatul datei și orei trebuie să fie ANUL-LUNA-ZIUA ORA-MINUTUL-SECUNDA. Exemplu: 2011-04-15 13:50:41.

Etichete:

Scrieți etichetele ce doriți să le asociați cu acest articol. Etichetele trebuie să fie separate de virgulă.

Stil: Font: Mărime: B I U S a^b a_b a a

2nd INTERNATIONAL CONFERENCE OF THE ROMANIAN SOCIETY FOR MASS SPECTROMETRY, ICRSMS 2011,
May 1-5, 2011, Chemistry Institute of the Romanian Academy, Timisoara

Organized by:

- [Romanian Academy](#)
- [Aurel Vlaicu University of Arad](#)
- [Romanian Society for Mass Spectrometry](#)

Conference Chair:

[Alina D. Zamfir](#)

E-mail: rsms@uav.ro

URL: <http://www.uav.ro/en/conferences/icrsms-2011>

Anulare Salvare

Profilul de curățare a paginii:
Normal

Formularul de editare articol permite modificarea titlului, a căii de accesare a articolului, data și etichetele acestuia. Conținutul articolului poate fi editat fără a fi necesare cunoștințe **HTML** cu Awebitor (vezi secțiunea 3.4).

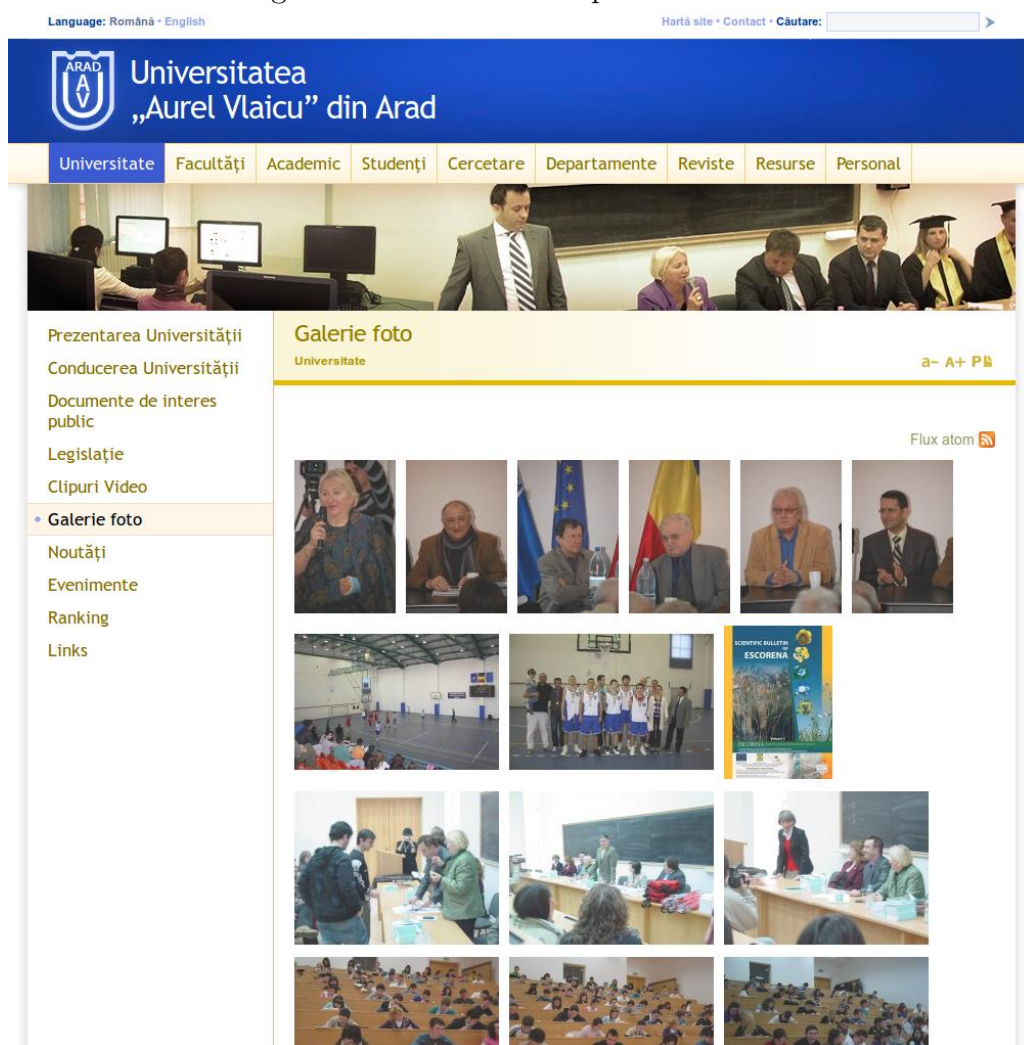
Operațiile de adăugare, ștergere și editare urmează permisiunile acordate administratorului. De exemplu, administratorul poate să nu aibă dreptul să

facă o anumită operație pe un anumit articol. Pentru detalii, vezi secțiunea 3.1.

3.6 Galeria de imagini

Modulul care permite adăugarea de pagini de tip „galerie de imagini” a pornit de la codul modulului de bloguri (vezi secțiunea 3.5). Administratorii pot adăuga pagini de tip „galerie de imagini” (vezi secțiunea 3.3.3).

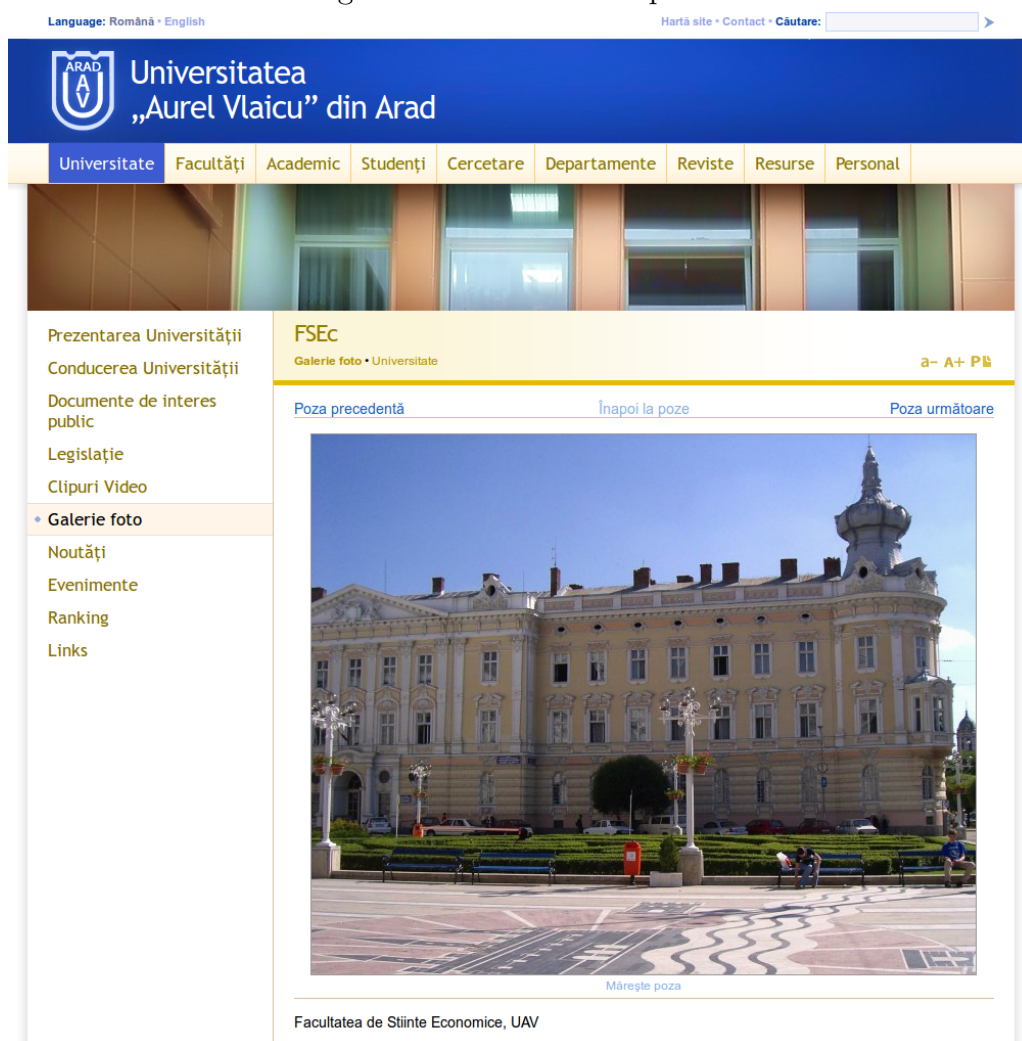
Figura 3.22: Galerie foto pe siteul UAV.



Precum blogurile, galeriile au și ele fluxuri Atom la care se pot abona utilizatorii care doresc să fie la curent cu ultimele poze adăugate.

După ce utilizatorul face *click* pe poza dorită se încarcă o pagină unde poza este afișată la o dimensiune medie. Sub poză este afișată descrierea acesteia:

Figura 3.23: Vizualizare poză.



Utilizatorul poate să facă încă un *click* pe poză pentru a vedea poza la mărimea originală.

Scriptul care generează lista de poze și pagina de vizualizare a unei poze este `pages/gallery.php`. Stilul acestor pagini este dat de `css/gallery.css`. Fluxul Atom este generat de `pages/gallery-atom.php`.

Pozele pentru fiecare galerie sunt salvate într-un tabel **MySQL**, vezi tabelul `gallery`, secțiunea **A.3**.

La adăugarea unei poze administratorul trebuie să încarce poza dorită. Aceasta este redimensionată pe server la o dimensiune mică pentru lista de poze și la dimensiunea medie pentru pagina în care este vizualizată poza.

Figura 3.24: Adăugare poză în galerie.

The screenshot displays a web application interface for managing a gallery. A modal window titled "Adaugă o imagine în galerie" is centered on the screen, overlaying a list of gallery pages. The modal contains the following fields and controls:

- *Titlu:** A text input field.
- Calea:** A text input field.
- Notă:** A note explaining that the path is used for generating the image address, with an example: `http://www.uav.ro/ro/nume-imagine`. It instructs to leave the field empty for automatic generation based on the title.
- *Data și ora:** A date and time input field showing "2011-04-19 16:09:34".
- Notă:** A note explaining the date and time format, with an example: "2011-04-19 16:09:33".
- *Imagine:** A text input field followed by a "Choisir..." button.
- Buttons:** "Salvare" (Save) and "Anulare" (Cancel) buttons at the bottom of the modal.

The background shows a list of gallery pages from "pag1" to "pag29", each with an "X" icon. At the bottom of the page, there are links for "Adaugă o imagine în galerie", "Editează fișiere din pagină: antet", and "Înapoi la pagini". The footer contains copyright information for 2011 and the site was realized by ROBO Design.

După cum se poate vedea din captura de ecran, administratorul trebuie să introducă titlul pozei și să aleagă fișierul de pe disc. Pozele nu pot avea etichete asociate. Opțional data pozei și calea acesteia pot fi schimbate.

Codul JavaScript folosit este în `js/admin.js`. Pentru afișarea formularului este chemată funcția `siteAdmin.galleryAdd()`. Pentru validarea formularului se execută funcția `siteAdmin.galleryAddSubmit()`. Pe server

se folosește funcția `AdminGenerateId()` pentru a genera calea pozei pe baza titlului, vezi pagina 73.

Pentru redimensionarea pozelor sunt folosite funcții din librăria *Graphics Draw* [10]. Funcția `handleImgUpload()` din `admin/gallery.php` este executată:

```
function handleImgUpload ($imgfile)
{
    // Verificăm dacă poza a fost încărcată cu succes.
    if ($imgfile['error'] !== UPLOAD_ERR_OK)
    {
        AdminMessage($lng['error-img-upload']);
        return false;
    }

    // Citim informații despre poză.
    $finfo = @getimagesize($imgfile['tmp_name']);
    if (!$finfo || ($finfo[2] !== 2 && $finfo[2] !== 3))
    {
        // Acceptăm doar poze în format JPEG sau PNG.
        AdminMessage($lng['error-img-format']);
        return false;
    }

    // În funcție de format este creată o imagine GD.
    if ($finfo[2] === 2)
    {
        $fext = '.jpg';
        $img = imagecreatefromjpeg($imgfile['tmp_name']);
    } else if ($finfo[2] === 3)
    {
        $fext = '.png';
        $img = imagecreatefrompng($imgfile['tmp_name']);
    } else
    {
        return false;
    }

    // Destinația pentru poza mică.
    $fname = basename(strtolower($imgfile['name']),
                        $fext);
    $fdest = $base . '/' . $folder_imgs . '/' .
             $folder_imgs_thumbnails . '/' .
```

```

    $fname . '.jpg';

    // Redimensionăm poza la mărimea mică dată de variabilele $img_tw
    // și $img_th.
    if (($finfo[0]/$finfo[1]) > ($img_tw/$img_th))
    {
        $new_w = $img_tw;
        $new_h = $img_tw/($finfo[0]/$finfo[1]);
    } else
    {
        $new_w = $img_th/($finfo[1]/$finfo[0]);
        $new_h = $img_th;
    }

    // Mai facem o imagine GD, acum pentru poza mică.
    $img2 = imagecreatetruecolor($new_w, $new_h);

    // Facem redimensionarea propriu-zisă.
    imagecopyresampled($img2, $img, 0, 0, 0, 0,
                        $new_w, $new_h, $finfo[0],
                        $finfo[1]);

    // Salvăm poza mică pe disc, cu calitatea dorită.
    $result = imagejpeg($img2, $fdest, $img_jpgq);

    if (!$result)
    {
        AdminMessage('/', $folder_imgs_thumbnails .
                        ': ' . $lng['error-img-save']);
        return false;
    }

    // ...

    return true;
}

```

După adăugarea unei poze administratorul poate să o editeze, să modifice titlul, calea, data și descrierea acesteia. Opțional, poza poate fi reîncărcată.

Figura 3.25: Modificarea unei poze.

Language: Română • English Utilizator robod (ROBO Design) • [Ieșire](#) • [Hartă site](#) • [Contact](#) • [Căutare:](#)

Universitatea „Aurel Vlaicu” din Arad

Prima pagină | Recomandări | Fișiere | Pagini | Utilizatori | Profil | Log out

FSEc
Galerie foto • Pagini

***Titlu:**
FSEc

Calea:
fsec

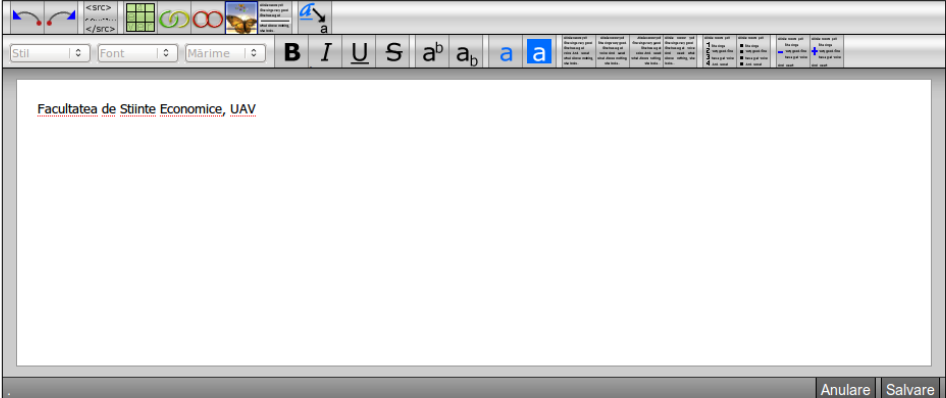
Notă: Calea este folosită pentru generarea adresei acestei imagini. Exemplu: <http://www.uav.ro/ro/nume-imagine>. Lăsați acest câmp gol pentru generare automată în funcție de titlul imaginii.

***Data și ora:**
2010-11-24 11:02:00

Notă: Formatul datei și orei trebuie să fie ANUL-LUNA-ZIUA ORA:MINUTUL:SECUNDA. Exemplu: 2011-04-19 16:10:39.

Imagine:





Facultatea de Stiinte Economice, UAV

Anulare | Salvare

Profilul de curățare a paginii:
Normal

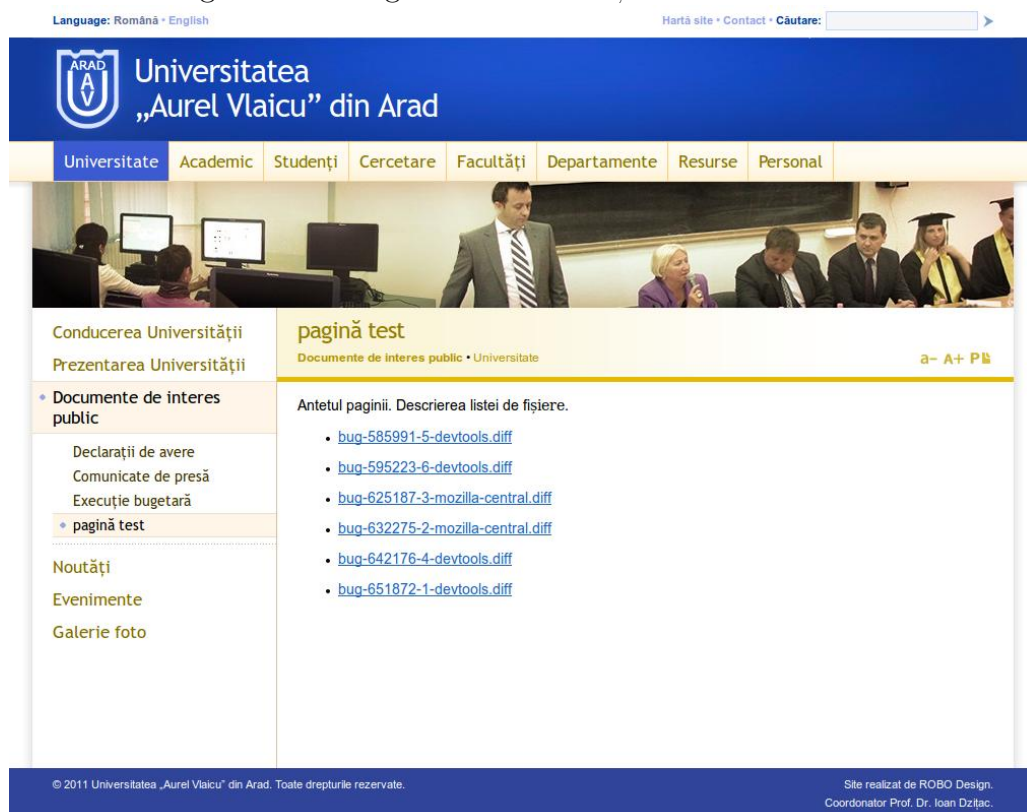
Pentru editarea descrierii este folosit editorul Awebitor, vezi secțiunea 3.4.

Paginile HTML cu descrierea fiecărei poze se găsesc în `pages/gallery`. Imaginile originale se găsesc în `files/gallery/original`. Imaginile de dimensiune medie sunt în `files/gallery/normal`, iar imaginile mici sunt în `files/gallery/thumbnails`.

3.7 Modulul „Fișiere de descărcat”

Unele pagini din site oferă utilizatorilor liste de fișiere care pot fi descărcate de pe server – de exemplu, pagina [Documente de interes public](http://www.uav.ro/ro/universitate/documente)⁴. Exemplu de pagină cu fișiere de descărcat:

Figura 3.26: Pagină de test cu fișiere de descărcat.



După cum se poate vedea din captura de ecran pagina are un antet care poate fi editat de administrator și lista de fișiere. Utilizatorul face *click* pe fișier și îl descarcă.

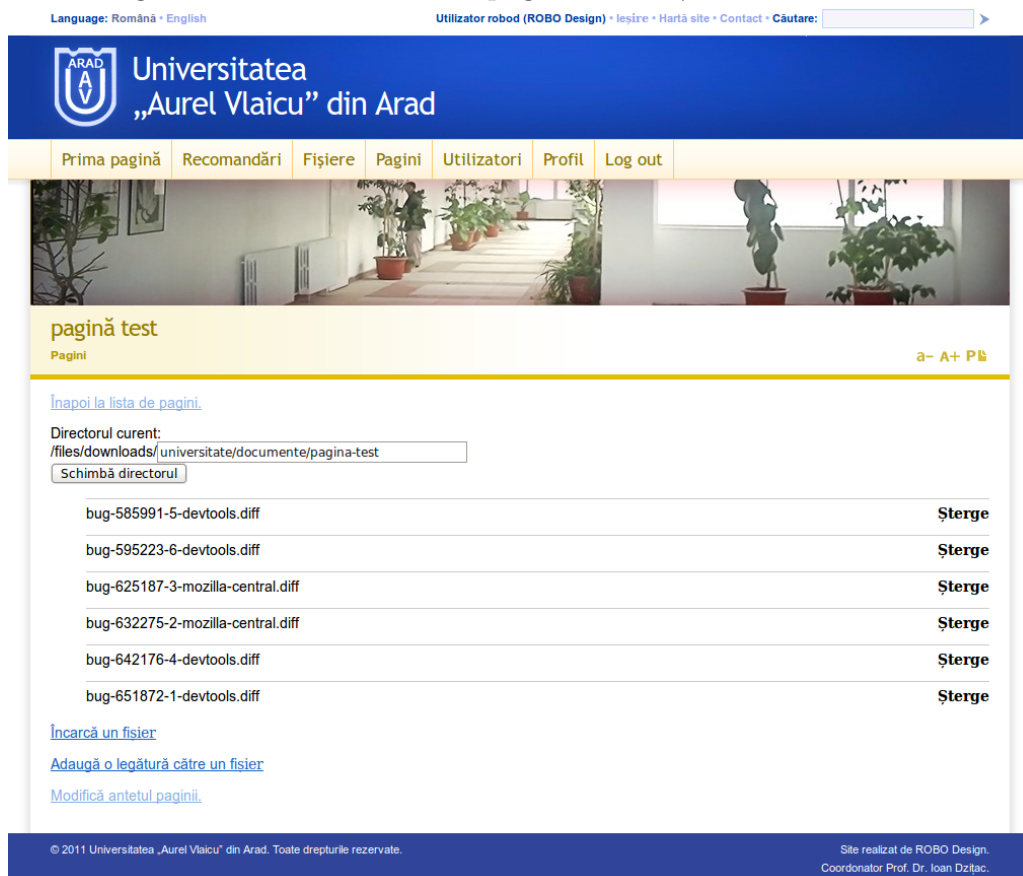
Paginile sunt generate de scriptul `pages/downloads.php`. În directorul `pages/downloads` se găsește antetul paginii, calea directorului de pe server asociat cu pagina accesată de utilizator și titlul paginii.

Atunci când administratorul adaugă o pagină nouă în site el poate să

⁴<http://www.uav.ro/ro/universitate/documente>

aleagă modulul „fișiere de descărcat” – detalii în secțiunea 3.3.3. Editarea acestui tip de pagini arată astfel:

Figura 3.27: Administrarea paginilor cu fișiere de descărcat.



Administratorul poate edita antetul paginii cu editorul Awebitor, vezi secțiunea 3.4. Lista de fișiere este generată automat din directorul asociat cu pagina în cauză. Administratorul poate să încarce fișierele prin **FTP**, sau poate să le încarce din pagina de administrare a siteului.

Fiecare fișier are un titlu și o descriere asociată. Aceste informații sunt salvate într-un fișier **JSON**. Conținutul fișierului arată astfel:

```
{
  "nume-fisier.extensie": {
    "type": "file",
    "file": "nume-fisier.extensie",
    "title": "titlul fișierului",
    "description": "descrierea fișierului",
  }
}
```

```
    },  
    "adresa-unui-fisier-extern": {  
        "type": "link",  
        "link": "http://example.com/fisier",  
        "title": "titlul legaturii externe",  
        "description": "descrierea legaturii",  
    }  
}
```

Acest fișier JSON este salvat în directorul asociat cu pagina editată. Administratorul poate schimba directorul asociat dacă dorește să afișeze fișiere dintr-un alt director decât cel implicit. Numele fișierului JSON este `_ro_metadata.json` pentru limba română și `_en_metadata.json` pentru limba engleză.

Din formatul fișierului JSON se poate vedea că pe lângă fișiere se pot adăuga și legături către fișiere externe. Unele documente se găsesc pe alte servere și atunci este mai simplă adăugarea unei legături directe către fișierele dorite, în loc să se facă o reîncărcare a acestora.

Încărcarea unui nou fișier arată astfel:

Figura 3.28: Încărcarea unui fișier.

The screenshot shows the web interface of the University of Arad. At the top, there is a blue header with the university's logo and name. Below the header is a navigation bar with links like 'Prima pagină', 'Recomandări', 'Fișiere', 'Pagini', 'Utilizatori', 'Profil', and 'Log out'. The main content area is divided into two columns. The left column displays a list of files with names like 'bug-585991-5-devtools.diff' and 'bug-595223-6-devtools.diff'. The right column contains a form titled 'Încarcă un fișier' (Upload a file). This form has fields for 'Fișierul:' (File), 'Titlu:' (Title), and 'Descrierea:' (Description). There are buttons for 'Chosir...', 'Trimite' (Submit), and 'Anulează' (Cancel). To the right of the form, there is a list of files with a 'Șterge' (Delete) button next to each one. The footer of the page contains copyright information and the site's creator details.

Formularul permite alegerea fișierului dorit de pe disc, și opțional se introduce titlul fișierului și descrierea acestuia. Afișarea formularului este implementată în `js/admin.js` de funcția `siteAdmin.downloadsAddFile()`. Validarea formularului este în funcția `siteAdmin.downloadsAddFileSubmit()`.

Adăugarea de legături externe arată la fel, dar în loc de câmpul pentru fișier este un câmp pentru adresa fișierului extern.

Dacă administratorul face *click* pe unul din fișierele listate, acesta poate să îi modifice titlul și descrierea.

Pe server adăugarea de fișiere trece printr-un filtru de extensii cunoscute, de exemplu nu este permisă încărcarea fișierelor de tip **PHP**. Directorul asociat cu o pagină este implicit ales pe baza căii paginii. Dacă pagina are calea `a/b/c`, atunci directorul pe server este `files/downloads/a/b/c`. Orice fișier încărcat este pus în acest loc.

3.8 Prima pagină

Prima pagină⁵ a siteului UAV este afișată de `pages/welcome.php`, vezi figura 1.1. Pe pagină se găsesc:

- o diaporamă aleatorie cu câteva antete care au design cu caracter promoțional,
- o mini-hartă a siteului, cu primele patru meniuri principale, fiecare având primele cinci sub pagini afișate în ordine aleatorie,
- ultimele articole din blogurile [Noutăți](#)⁶ și [Evenimente](#)⁷,
- datele de contact din fișierul `pages/en_welcome_contact.html` sau `pages/ro_welcome_contact.html`,
- partenerii universității din fișierul `pages/en_welcome_partners.html` sau `pages/ro_welcome_partners.html`,
- primul paragraf din trei pagini importante ale siteului: [Despre UAV](#)⁸, [Istoric](#)⁹ și [Cercetare](#)¹⁰.

În scriptul `welcome.php` funcția `mini_sitemap()` generează ierarhia de pagini a siteului: primele patru meniuri principale din care sunt luate aleatoriu cinci pagini. Ținând cont că în timpul execuției interfața siteului există într-un obiect **DOM** pe server, acest script preia lista de pagini direct din meniurile generate, fără să recitească structura **XML** a siteului (vezi secțiunea 3.3.1).

Funcția `blog_section()` afișează titlul ultimelor articole publicate din blogul dorit. Opțional această funcție poate afișa și primele cuvinte din articole. Funcția `page_section()` afișează numărul de cuvinte ales din pagina dorită.

Scriptul `js/welcome.js` implementează efectul de tranziție între pozele din antetul paginii. O poză este pusă cu **CSS** folosindu-se proprietatea `background-image`, iar cealaltă poză este suprapusă cu un element `<img>`. Cu funcția `setTimeout()` se modifică proprietatea `opacity` a elementului: valoarea crește sau scade, după cum este cazul.

⁵<http://www.uav.ro/>

⁶<http://www.uav.ro/ro/universitate/noutati>

⁷<http://www.uav.ro/ro/universitate/evenimente>

⁸<http://www.uav.ro/ro/universitate/prezentarea-universitatii>

⁹<http://www.uav.ro/ro/universitate/prezentarea-universitatii/istoric>

¹⁰<http://www.uav.ro/ro/cercetare>

3.9 Pagina Contact

Pagina de contact¹¹ a universității permite utilizatorilor să transmită mesaje departamentelor dorite. Formularul arată astfel:

Figura 3.29: Pagina de contact a Universității „Aurel Vlaicu” din Arad.

Language: Română • English Hartă site • Contact • Căutare:

Universitatea „Aurel Vlaicu” din Arad

Universitate Facultăți Academic Studenți Cercetare Departamente Reviste Resurse Personal

Contact a- A+ P

Destinație:
Admitere

Nume:

Email:

Subject:

Mesaj:

Adresă Rectorat:
B-dul Revoluției nr. 77,
310130 Arad, România,
P.O. BOX 2/158 AR

Telefon: +40-257-283010
Fax: +40-257-280070
Email: rectorat@uav.ro

Pentru prevenirea mesajelor de tip "spam-bot":
Vă rugăm să utilizați interfața de contact alăturată cu următoarele destinații:

- Admitere (Informații suplimentare despre Admiterea 2011)
- Alumni (Adresă de contact pentru foștii studenți UAV);
- Consiliere (Departamentul de consiliere în carieră);
- Comunicare studenți-rectorat (întrebări/reclamații/sesizări adresate conducerii UAV);
- Design (Facultatea de Design);
- DPPD (Departamentul pentru Pregătirea Personalului Didactic);
- DPPD studenți (Comunicare între studenți și DPPD);
- ECDL (Precizați obligatoriu: adresă de e-mail validă, nume, prenume, număr de telefon, domeniul de interes: cursuri, examinare EC DL, taxe, altele);
- EFS (Facultatea de Educație Fizică și Sport);
- FIATP (Facultatea de Inginerie Alimentară, Turism și Protecția Mediului);
- FSEPAS (Facultatea de Științe ale Educației, Psihologie și Asistență Socială);
- ID (Învățământ la Distanță);
- Inginerie (Facultatea de Inginerie);
- Info pub 1 (informații de interes public - date economice);
- Info pub 2 (informații de interes public - date de altă natură);
- Rectorat (Instituția Rectorului);
- Științe Economice (Facultatea de Științe Economice);
- Științe Exacte (Facultatea de Științe Exacte);

Vicksburg antrāt

Type the two words:

Trimite email

În acest formular utilizatorul poate să scrie subiectul emailului, conținutul acestuia și să își scrie adresa de email. Destinatarul este ales din lista de departamente adăugate de administratorii siteului. Formularul este protejat anti-spam cu serviciul [recaptcha.net](https://www.recaptcha.net). La trimiterea unui mesaj pe server se

¹¹<http://www.uav.ro/ro/contact>

face validarea adresei de email și a celorlalte câmpuri din formular. Mesajul este trimis la destinatarul ales cu funcția `mail()`.

Pagina este afișată de scriptul `pages/contact.php`. În dreapta formularului apar datele de contact din fișierul `pages/ro_contact.html` sau `pages/en_contact.html`. Aceste informații pot fi editate de administratorii siteului.

Administratorii siteului pot edita pagina de contact și lista destinatarilor.

Figura 3.30: Administrarea paginii de contact.

Language: Română • English Utilizator robod (ROBO Design) • [Ieșire](#) • [Hartă site](#) • [Contact](#) • [Căutare:](#)

Universitatea „Aurel Vlaicu” din Arad

Prima pagină | Recomandări | Fișiere | **Pagini** | Utilizatori | Profil | Log out

Contact

[Contact](#) a- A+ P

[Înapoi la pagini](#)

Editare pagini

Aci puteți alege să editați datele de contact afișate în diferite pagini din acest web site.

- [Pagina Contact: editează datele de contact.](#)
- [Prima pagină: editează datele de contact afișate pe prima pagină.](#)

Editează lista de contacte

Utilizatorii pot alege să trimită mesajul la diferite persoane sau departamente, după cum decideți dvs. Mai jos, aveți posibilitatea să adăugați persoane/departamente care să poată fi contactate de utilizatori.

Admitere	Modifică	Șterge
Design	Modifică	Șterge
Rectorat	Modifică	Șterge
Științe Economice	Modifică	Șterge
Științe Exacte	Modifică	Șterge
Științe Umaniste	Modifică	Șterge
Webmaster	Modifică	Șterge

Adaugă o nouă adresă de contact

© 2011 Universitatea „Aurel Vlaicu” din Arad. Toate drepturile rezervate. Site realizat de ROBO Design. Coordonator Prof. Dr. Ioan Dzitac.

Lista de destinatari este citită din fișierul `pages/ro_contact.json` sau `pages/en_contact.json`. Formatul fișierului **JSON** este foarte simplu:

```
{  
  "jeton-identificare": [ "nume" , "email@example.com" ] ,  
  // ... alte destinații  
}
```

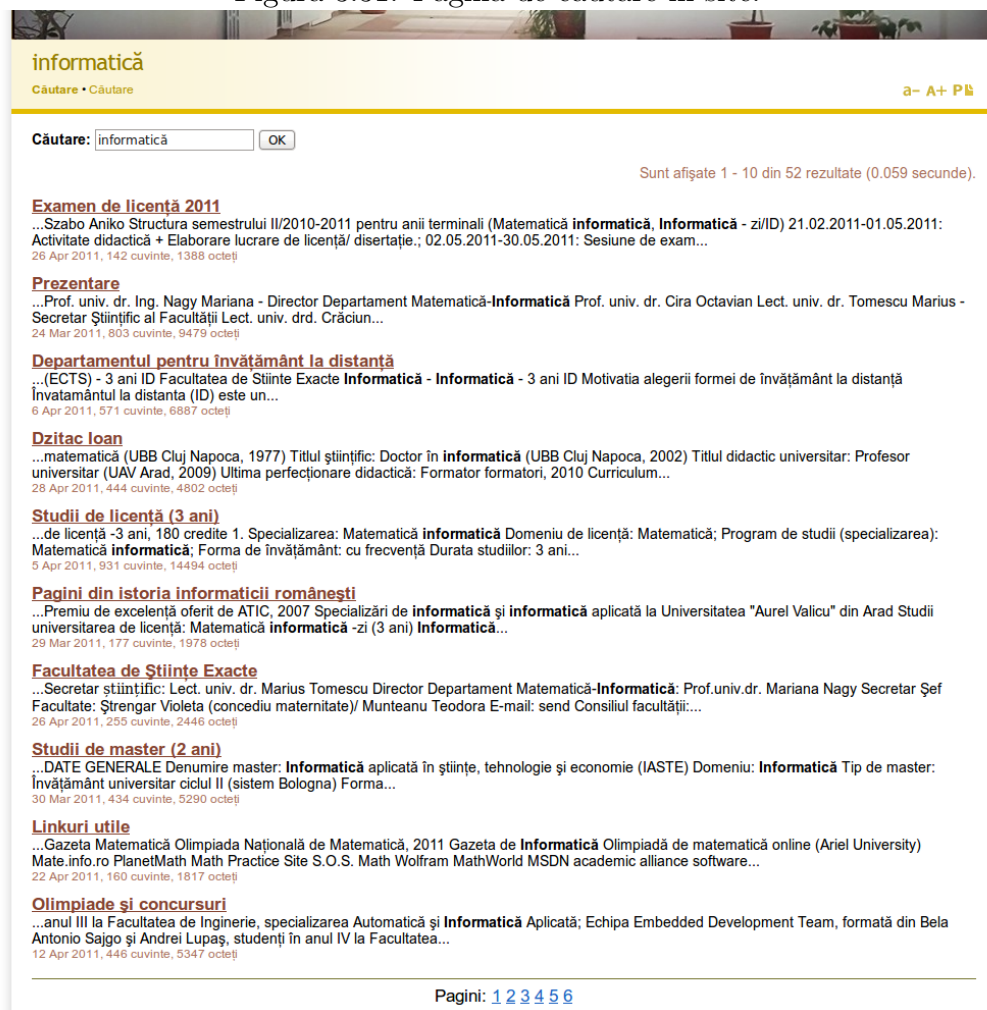
Pagina de administrare este generată de scriptul `admin/contact.php`. La adăugarea unui destinatar se execută funcția JavaScript `siteAdmin.contactAdd()`. Administratorul este întrebat de numele destinatarului și adresa de email a acestuia. Pe server jetonul de identificare al destinatarului este generat de funcția `AdminGenerateId()` (vezi pagina 73), ca apoi fișierul JSON să fie actualizat.

Editarea datelor de contact care apar în dreapta formularului și a datelor de contact care apar pe prima pagină se face folosindu-se Awebitor, vezi secțiunea 3.4.

3.10 Motorul de căutare

Pentru utilizatori motorul de căutare este principala metodă de a găsi conținutul dorit în siteul Universității „Aurel Vlaicu” din Arad. Exemplu de căutare:

Figura 3.31: Pagina de căutare în site.



În toate paginile interfața siteului afișează câmpul de căutare în antet. Când utilizatorul alege să caute ceva în site se accesează scriptul `pages/search.php`. Acesta preia cuvintele cheie și le transmite mai departe la scriptul `modules/search.php`. Rezultatele sunt date într-un vector (*array*) afișat în **HTML**.

Practic `pages/search.php` este doar interfața de căutare și afișare rezul-

tate a modulului de căutare `modules/search.php`. Stilul `css/search.css` este folosit pentru stilarea rezultatelor în navigatorul web.

Utilizatorii pot căuta în siteul UAV cu Google dacă adaugă prefixul „g”. În acest caz cuvintele cheie sunt redirectionate către motorul de căutare Google.

În scriptul din `pages` se execută:

```
$results = searchDocument( array(
    'keywords' => $_REQUEST[ 'q' ]
));
```

Funcția `searchDeleteDocument()` este dată de scriptul din `modules`. Aceasta permite căutarea paginilor din site. Parametrii funcției sunt următorii:

keywords Cuvintele cheie folosite pentru căutare.

sortby Sortarea rezultatelor după:

- updated** ultima actualizare a documentelor,
- words** numărul de cuvinte în documente,
- title** titlurile documentelor,
- url** adresele documentelor,
- score** scorul rezultatului.

Implicit rezultatele sunt sortate după scorul acestora. Scorul este calculat în funcție de câte ori apar cuvintele cheie căutate, după mărimea documentului și alți factori.

order Direcția de sortare: -1 pentru sortare descrescătoare, 1 pentru sortare crescătoare.

operator Operatorul folosit între cuvinte: **AND** sau **OR**. Dacă se folosește **AND** toate cuvintele cheie trebuie să apară în document. Dacă se folosește **OR** în lista de rezultate apar pagini care conțin oricare din cuvintele cheie date.

soundex Parametru boolean care spune că se dorește căutare de tip **Soundex** [11] în **MySQL**. Practic vor fi găsite și alte cuvinte similare cu cuvintele cheie date.

Fiecare rezultat este dat într-un vector:

id Numărul de identificare al paginii în tabelul **MySQL**.

title Titlul paginii.

url Adresa paginii.

snippet O bucată din document aleasă în așa fel încât să conțină unul sau mai multe cuvinte cheie. Aceasta este folosită în afișarea rezultatelor.

updated Ultima actualizare a paginii.

score Scorul rezultatului.

words Numărul de cuvinte în document.

size Mărimea documentului în octeți.

Cuvintele cheie sunt filtrate pentru a se elimina orice semn de punctuație, cuvintele mai lungi decât 255 de caractere și orice alt cuvânt care ar fi mal-format.

Motorul de căutare folosește două tabele MySQL pentru fiecare limbă: **search_documents** (vezi anexa A.4) și **search_doc_index** (vezi anexa A.5). În primul tabel sunt indexate documentele cu titlul lor, adresa și conținutul. Conținutul este filtrat să nu conțină taguri HTML. În al doilea tabel sunt indexate cuvintele din toate documentele, fiecare cuvânt fiind asociat cu **id**-ul documentului în care apare.

Comenzile de interogare MySQL folosesc funcții specifice de transliterare și colare a caracterelor. De exemplu, dacă utilizatorul caută după „științe” vor fi găsite și paginile care includ cuvântul „stiinte”. Similar, dacă utilizatorul nu folosește diacritice dar paginile au cuvintele scrise cu diacritice, rezultatele vor fi conform așteptărilor.

Pentru indexarea documentelor funcția **searchIndexDocument()** este chemată de toate modulele de administrare a siteului. De exemplu, blogurile și galeriile de imagini sunt indexate atunci când administratorul salvează o pagină, un articol sau o imagine. Funcția primește ca parametru adresa paginii, titlul și conținutul. Scriptul **search.php** se ocupă de filtrarea conținutului, de curățarea acestuia, de găsirea cuvintelor care trebuie indexate și tot ce mai este necesar.

Atunci când administratorul alege să șteargă o pagină funcția **searchDeleteDocument()** este chemată.

3.11 Harta siteului

Utilizatorii pot să caute informațiile dorite în harta siteului. Aceasta oferă o perspectivă completă asupra tuturor paginilor din site:

Figura 3.32: Harta siteului UAV.



Scriptul `pages/sitemap.php` citește fișierul `pages/sitemap_ro.html` sau `pages/sitemap_en.html`. Fișierele **HTML** sunt actualizate când se fac modificări la indexul de pagini.

Pagina cu harta siteului folosește stilul `css/sitemap.css` și scriptul `js/sitemap.js`. Fișierul **JavaScript** permite utilizatorilor să interacționeze cu harta siteului, să intre în structura de pagini după cum consideră necesar.

Scriptul `modules/sitemap.php` conține clasa `RD_sitemap`. Toate celelalte module de administrare creează o instanță a acestei clase pentru a putea adăuga/modifica/șterge pagini din harta siteului. Principalele metode ale clasei:

`index()` Permite indexarea unei pagini. Parametrii sunt titlul și adresa acesteia. Această metodă cheamă `add()` sau `update()`.

`add()` Adaugă o pagină în index.

`update()` Actualizează o pagină din index.

`delete()` Șterge o pagină din index.

`done()` Metodă chemată după ce se modifică indexul de pagini. Aceasta cheamă metodele `xml()` și `html()`.

`xml()` Generează harta siteului în format Sitemaps XML [1].

`html()` Generează harta siteului în format HTML.

Fișierul `xml/sitemap.xml` arată astfel:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
  xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <url>
    <dc:title>titlu pagina</dc:title>
    <loc>http://www.uav.ro/ro/...</loc>
    <lastmod>2011-04-22T13:08:42+03:00</lastmod>
  </url>
  <!-- ... -->
</urlset>
```

După cum se poate vedea din codul de mai sus formatul Sitemaps XML este simplu: trebuie doar adresa și data ultimei modificări a paginii în format **W3C** [12]. Pe lângă asta am inclus și titlul paginii într-un alt **XML namespace**, Dublin Core [13].

Harta siteului este indexată în tabelul **MySQL sitemap**, pentru detalii vezi anexa **A.6**.

Concluzii

Siteul Universității „Aurel Vlaicu” din Arad a evoluat mult de la versiunea precedentă. Îmbunătățirile tehnice și de design, împreună cu îmbunătățirile de conținut și structură, fac noul site UAV mult mai ușor de folosit, mai atractiv și mai informativ. Această evoluție pozitivă a siteului este confirmată de Societatea Academică din România [14], și ea se datorează muncii importante depuse în planuirea proiectului, implementarea și folosirea lui de la lansare.

Din acest proiect am învățat multe lucruri și am pus în valoare studiile de peste ani în domeniul informaticii.

Scopul proiectului a fost atins, dar acest lucru nu înseamnă că evoluția siteului trebuie să se oprească aici. Câteva direcții de evoluție a siteului ar putea fi:

- îmbunătățirea securității – trecerea pe **TLS** pentru modulul de administrare, vezi secțiunea **3.1.2.5**.
- reanalizarea conținutului și a oportunităților de îmbunătățire organizare / structură.
- optimizarea tehnică, încărcarea și mai rapidă a paginilor – optimizare la anumite poze.
- implicarea a mai multor administratori, pe diferite secțiuni din site.

Mulțumesc profesorilor UAV și mai ales domnului profesor coordonator Ioan Dzițac pentru că au făcut posibilă realizarea acestui proiect de licență, un proiect care nu este doar teoretic, ci are o aplicabilitate practică.

Glosar

Apache este cel mai folosit server web¹, distribuit ca software liber de Fundația Apache. 13, 14, 25, 47, 63

Atom este un format XML standard, RFC 4287², pentru fluxuri web. Acesta permite navigatoarelor web sau oricărui alt software să verifice dacă sunt actualizări la resurse de pe web. În unele implementări mai avansate, formatul Atom poate permite și publicarea și actualizarea resurselor/paginilor web. 7, 9, 13, 16, 23, 81, 88, 95

cookie este un antet folosit în protocolul HTTP. Acest antet permite salvarea de perechi *cheie – valoare* în navigatoarele web ale utilizatorilor, mai precis acestea sunt variabile cu valori asociate. Aplicațiile web pot modifica paginile generate în funcție de aceste variabile. 14

CSS (Cascading Style Sheets) este un limbaj de formatare a elementelor dintr-o pagină web³, standardizat de către W3C. 9, 11, 12, 64, 75, 105

DOM (Document Object Model) este o convenție pentru reprezentarea și interacționarea cu elementele și atributele din documentele de tip HTML, XHTML și XML. Această convenție este independentă de limbajul de programare folosit, iar elementele DOM sunt manipulate folosindu-se sintaxa și gramatica limbajului de programare în care API-ul⁴ DOM este implementat. Specificațiile DOM sunt recomandări publicate de W3C. 13, 15, 50, 58, 71, 73, 75, 105

ECMAScript este un limbaj de programare dinamic standardizat de Ecma International, în cadrul specificației ECMA-262⁵ și ISO/IEC

¹<http://httpd.apache.org>

²<http://tools.ietf.org/html/rfc4287>

³<http://www.w3.org/Style/CSS/>

⁴Application Programmable Interface

⁵<http://www.ecma-international.org/publications/standards/Ecma-262.htm>

16262. Limbajul este implementat sub forma unor dialecte bine cunoscute precum JavaScript, JScript și ActionScript. 116

FTP (File Transfer Protocol) este un protocol de rețea folosit pentru transferul de fișiere între clienți și server [18]. 102

HTML (HyperText Markup Language) este un limbaj de marcare folosit pentru crearea paginilor web⁶, standardizat de către W3C. Navigatoarele web interpretează codul HTML pentru a afișa conținutul. 12, 14, 16, 18, 29, 56, 63, 74, 78, 83, 94, 109, 112, 116

HTTP (HyperText Transfer Protocol) este un protocol de rețea distribuit pentru colaborare în sisteme informatice interactive [16]. Acest protocol este de tip cerere – răspuns, unde clientul, navigatorul web, cere o anumită adresă și primește răspunsul de la server. HTTP este fundamentul webului, fiind principalul protocol de transfer de date. 15, 25

JavaScript este o implementare a standardului ECMAScript, ce se regăsește cu precădere în navigatoarele web. Acest limbaj de programare permite aplicațiilor web să fie dinamice și interactive. 11, 12, 29, 47, 50, 56, 75, 112

JSON (JavaScript Object Notation) este un format de transfer de date structurate, simplu, ce poate fi citit de către oameni. Acesta este un standard deschis ce preia reprezentarea obiectelor din JavaScript. Pentru interpretarea JSON nu este necesară implementarea întregului limbaj JavaScript [15]. 13, 16, 22, 26, 49, 102, 107

MySQL este un sistem de gestiune a bazelor de date relațional⁷, software liber distribuit sub licența GPL⁸ v2. 12, 14, 16, 22, 25, 83, 96, 110, 113

PHP este un limbaj de programare dinamic⁹ ce nu are o specificație formală. Acesta este folosit cu precădere pe serverele web pentru generarea dinamică a paginilor HTML. Principala implementare a limbajului este un software liber cunoscut tot sub numele de PHP. 12–14, 16, 17, 26, 30, 40, 47, 57, 61, 75, 78, 104

⁶<http://www.w3.org/html/wg/>

⁷<http://www.mysql.com>

⁸GNU General Public License

⁹<http://www.php.net>

TLS (Transport Layer Security) este un protocol criptografic de rețea ce oferă securitate în comunicarea prin rețele precum Internetul [17]. Acest protocol este folosit pentru criptarea pe segmente a mesajelor transmise de alte protocoale. De exemplu, HTTP poate fi securizat cu TLS, ceea ce dă naștere protocolului cunoscut sub numele de HTTPS. 54, 114

W3C (World Wide Web Consortium) este principalul consorțiu internațional de standardizare a webului¹⁰. Membrii organizației dezvoltă și ratifică standarde web precum HTML, CSS și SVG. 113, 115–117

XHTML (Extensible HyperText Markup Language) este limbajul de marcare HTML redefinit în format XML. 13, 17, 75

XML (Extensible Markup Language) este un meta-limbaj de marcare, de descriere structurată a datelor¹¹, standardizat de către W3C. Cu ajutorul acestui meta-limbaj se pot defini alte limbaaje de marcare, precum XHTML, Atom și SVG. 13, 15, 24, 61, 88, 105, 113

¹⁰<http://www.w3.org>

¹¹<http://www.w3.org/XML/>

Listă de figuri

1.1	Prima pagină a noului site uav.ro , în limba română.	5
3.1	Formularul de autentificare.	29
3.2	Formularul de resetare a parolei.	38
3.3	Pagina de administrare a conturilor.	41
3.4	Opțiunea de adăugare cont.	46
3.5	Lista de drepturi date unui cont.	50
3.6	Editarea drepturilor: adăugare secțiune.	51
3.7	Editarea drepturilor: adăugare acțiune.	54
3.8	Editare de profil utilizator.	56
3.9	Gestionarul de fișiere.	57
3.10	Gestionarul de pagini.	61
3.11	Adăugare pagină nouă.	71
3.12	Setări pagină.	74
3.13	Awebitor: editorul de pagini.	75
3.14	Adăugarea de imagini în pagini.	78
3.15	Blogul Evenimente.	83
3.16	Vizualizarea unui articol în blogul Evenimente.	87
3.17	Arhiva de articole într-un blog.	88
3.18	Fluxul Atom al blogului Evenimente în Firefox.	89
3.19	Pagina de administrare a blogului Evenimente.	92
3.20	Adăugarea unui articol în blog.	94
3.21	Editarea unui articol din blog.	95
3.22	Galerie foto pe siteul UAV.	96
3.23	Vizualizare poză.	97
3.24	Adăugare poză în galerie.	98

3.25	Modificarea unei poze.	101
3.26	Pagină de test cu fișiere de descărcat.	102
3.27	Administrarea paginilor cu fișiere de descărcat.	103
3.28	Încărcarea unui fișier.	105
3.29	Pagina de contact a Universității „Aurel Vlaicu” din Arad. . .	107
3.30	Administrarea paginii de contact.	108
3.31	Pagina de căutare în site.	110
3.32	Harta siteului UAV.	113

Bibliografie

- [1] Documentația Sitemaps XML.
<http://www.sitemaps.org/>
- [2] Documentația pentru XMLHttpRequest.
<https://developer.mozilla.org/en/xmlhttprequest>
- [3] Documentația pentru designMode și contentEditable.
https://developer.mozilla.org/en/rich-text_editing_in_mozilla
- [4] Documentația HTML 5.
<http://www.whatwg.org/specs/web-apps/current-work/multipage/>
- [5] Implementarea *open source* a algoritmului criptografic SHA-512.
<http://jssha.sourceforge.net/>
- [6] Implementare *open source* a formatului JSON în PHP.
http://pear.php.net/package/Services_JSON
- [7] Implementare nativă a formatului JSON în PHP.
<http://www.php.net/manual/en/book.json.php>
- [8] Descrierea setului de funcții criptografice SHA-2.
<http://en.wikipedia.org/wiki/SHA-2>
- [9] Proiectul HTML Tidy.
<http://tidy.sourceforge.net/>
- [10] Librăria Graphics Draw.
<http://www.boutell.com/gd/>
- [11] Descrierea algoritmului fonetic de indexare cuvinte după sunet Soundex.
<http://en.wikipedia.org/wiki/Soundex>

- [12] Formatul de dată și oră W3C.
<http://www.w3.org/TR/NOTE-datetime>
- [13] Inițiativa de metadate Dublin Core.
<http://dublincore.org/>
- [14] Topul Integrității Universităților Publice.
<http://uav.ro/ro/universitate/evenimente/topul-integritatii-universitatilor-publice-c-u-c-2010>
- [15] Descrierea formatului JSON.
<http://en.wikipedia.org/wiki/JSON>
- [16] Descrierea protocolului HTTP.
http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- [17] Descriere a protocolului Transport Layer Security.
http://en.wikipedia.org/wiki/Transport_Layer_Security
- [18] Descrierea protocolului FTP.
http://en.wikipedia.org/wiki/File_Transfer_Protocol
- [19] Documentația Apache.
<http://httpd.apache.org/docs>
- [20] Documentația PHP.
<http://php.net/manual/en/>
- [21] Documentația MySQL.
<http://dev.mysql.com/doc/>
- [22] Documentația CSS 2.1.
<http://www.w3.org/TR/CSS2/>
- [23] Definiția formatului JSON.
<http://www.json.org/>
- [24] Specificația XML.
<http://www.w3.org/TR/REC-xml/>
- [25] Manual JavaScript.
<https://developer.mozilla.org/en/JavaScript/Reference>
- [26] Specificația DOM 3 Core.
<http://www.w3.org/TR/DOM-Level-3-Core/>

- [27] Specificația DOM 2 Events.
<http://www.w3.org/TR/DOM-Level-2-Events/>

Anexa A

Baza de date MySQL

A.1 blog

Tabelul conține toate articolele din toate blogurile siteului. Acest tabel are următoarele câmpuri:

id, *int(10)*. Acesta este numărul de identificare al articolului.

pubdate, *bigint(14)*. Data publicării articolului.

updated, *bigint(14)*. Data la care a fost actualizat ultima oară articolul.

category, *varchar(255)*. „Categoria” în care apare articolul. În acest caz „categoria” se referă la blogul/pagina în care apare articolul. Categoria este dată de calea paginii / blogului. De exemplu, categoria poate să fie `universitate/noutati`.

path, *varchar(255)*. Calea articolului. De exemplu, aceasta poate fi `noul-site-uav`. Utilizatorul poate accesa articolul pe adresa: `/ro/universitate/noutati/noul-site-uav`.

title, *varchar(255)*. Titlul articolului.

content, *varchar(255)*. Conținutul articolului. Acest câmp indică numai numele fișierului HTML care trebuie citit de pe disc pentru afișarea articolului.

Mai multe detalii despre acest modul în secțiunea [3.5](#).

A.2 blog_tags

Acest tabel asociază articole din bloguri cu etichete. Câmpurile tabelului sunt următoarele:

bid, *int(10)*. Numărul de identificare al articolului.

tag, *varchar(100)*. Eticheta asociată articolului.

A.3 gallery

Tabelul conține toate imaginile din toate galeriile de imagini ale siteului. Acest tabel are următoarele câmpuri:

id, *int(10)*. Acesta este numărul de identificare al imaginii.

pubdate, *bigint(14)*. Data publicării imaginii.

updated, *bigint(14)*. Data la care a fost actualizată ultima oară imaginea.

category, *varchar(255)*. „Categorie” în care apare imaginea. În acest caz „categoria” se referă la galeria/pagina în care apare imaginea. Categoria este dată de calea paginii / galeriei. De exemplu, categoria poate să fie `universitate/galerie-foto`.

path, *varchar(255)*. Calea imaginii. De exemplu, aceasta poate fi `pag26`. Utilizatorul poate accesa imaginea pe adresa: `/ro/universitate/galerie-foto/pag26`.

title, *varchar(255)*. Titlul imaginii.

content, *varchar(255)*. Descrierea imaginii este afișată sub imagine. Acest câmp indică numai numele fișierului HTML care trebuie citit de pe disc pentru afișarea descrierii.

img, *varchar(255)*. Imaginea care trebuie afișată. Valoarea acestui câmp este numele fișierului de pe disc.

Mai multe detalii despre acest modul în secțiunea [3.6](#).

A.4 search_documents_en și search_documents_ro

Acest tabel conține lista de pagini, indexate, care pot fi căutate de utilizatori folosind opțiunea de căutare a siteului. Tabelul are următoarele câmpuri:

id, *bigint(20)*. Numărul de identificare al paginii.

title, *varchar(255)*. Titlul paginii.

url, *varchar(255)*. Adresa paginii.

doc, *mediumtext*. Conținutul paginii.

keywords, *varchar(255)*. Cuvintele cheie/importante asociate cu pagina dată.

published, *bigint(14)*. Data la care pagina a fost publicată.

updated, *bigint(14)*. Data la care pagina a fost actualizată ultima oară.

words, *mediumint(8)*. Numărul de cuvinte în conținutul paginii.

size, *bigint(20)*. Mărimea paginii (în octeți).

Mai multe detalii despre acest modul în secțiunea 3.10.

A.5 search_doc_index_en și search_doc_index_ro

Acest tabel indexează cuvintele fiecărei pagini indexate de motorul de căutare al siteului. Câmpurile tabelului sunt următoarele:

ids, *text*. Lista de numere de identificare a paginilor asociate cu cuvântul dat.

word, *varchar(255)*. Cuvântul indexat.

total, *int(10)*. Numărul de menționări a cuvântului în toate paginile indexate.

counts, *text*. Numărul de menționări a cuvântului pentru fiecare pagină din coloana **ids**.

Mai multe detalii despre acest modul în secțiunea 3.10.

A.6 sitemap

Acest tabel conține lista de pagini indexate pentru harta siteului. Câmpurile tabelului sunt următoarele:

id, *bigint(20)*. Numărul de identificare al paginii.

title, *varchar(255)*. Titlul paginii.

url, *varchar(255)*. Adresa paginii.

published, *bigint(14)*. Data publicării paginii.

updated, *bigint(14)*. Data ultimei actualizări.

lang, *varchar(3)*. Limba în care este scris conținutul paginii.

Mai multe detalii despre acest modul în secțiunea [3.11](#).