

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
CATEDRA CALCULATOARE

SITE WEB DE SOCIALIZARE FOLOSIND TEHNOLOGII MICROSOFT

LUCRARE DE LICENȚĂ

Absolvent: **Nicoleta Delia MICAN**

Coordonator: **Șef lucr. ing. Cosmina IVAN**

2011

Commented [J.I.M.1]: anul depunerii proiectului



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
CATEDRA CALCULATOARE

DECAN,

Prof. dr. ing. Sergiu NEDEVSCI

VIZAT,

ȘEF CATEDRĂ,

Prof. dr. ing. Rodica
POTOLEA

Absolvent: **Nicoleta Delia MICAN**

SITE WEB DE SOCIALIZARE FOLOSIND TEHNOLOGII

MICROSOFT

1. **Enunțul temei :** *Proiectul își propune realizarea unui site web pentru socializare folosind cât mai multe tehnologii Microsoft care să ofere utilizatorilor posibilitatea de a interacționa cât mai ușor.*
2. **Conținutul proiectului:** *Pagina de prezentare, aprecierile coordonatorului de proiect, sinteza, Introducere, Obiectivele proiectului, Studiu bibliografic, Analiză și fundamentare teoretică, Proiectare de detaliu și implementare, Testare și validare, Concluzii și dezvoltări ulterioare, bibliografie, anexe.*
3. **Locul documentării:** Universitatea Tehnică din Cluj-Napoca, Catedra Calculatoare
4. **Consultanți:** Șef lucr. ing. Cosmina IVAN
5. **Data emiterii temei:** 1 noiembrie 2010
6. **Data predării:** 24 iunie 2011

Absolvent:

Coordonator științific:

Commented [J.I.M.2]: Exemplu: Universitatea Tehnică din Cluj-Napoca, Catedra Calculatoare

Commented [J.I.M.3]: ziua luna anul

Commented [J.I.M.4]: ziua luna anul

Commented [J.I.M.5]: Semnătura autorului

Commented [J.I.M.6]: Semnătura coordonatorului

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
CATEDRA CALCULATOARE

Declarație

Subsemnata, *Nicoleta Delia MICAN*, studentă a Facultății de Automatică și Calculatoare, Universitatea Tehnică din Cluj-Napoca, declar că ideile, analiza, proiectarea, implementarea, rezultatele și concluziile cuprinse în această lucrare de licență constituie efortul meu propriu, mai puțin acele elemente ce nu îmi aparțin, pe care le indic și recunosc ca atare.

Declar de asemenea că, după știința mea, lucrarea în această formă este originală și nu a mai fost niciodată prezentată sau depusă în alte locuri sau alte instituții decât cele indicate în mod expres de mine.

Data: 23 iunie 2011

Absolvent: **Nicoleta Delia MICAN**

Commented [J.I.M.7]: data predării proiectului

Număr matricol: 21010668

Semnătura:

Commented [J.I.M.8]: Semnătura autorului

Cuprins

1. Introducere – Contextul proiectului	6
1.1 Context	6
1.2 Tema și conținutul lucrării	6
2. Obiectivele proiectului	8
2.1 Obiective generale	8
2.2 Obiective specifice sistemului	8
2.2.1 Cerințe funcționale	8
2.2.2 Cerințele non-funcționale	9
3. Studiu bibliografic	11
3.1 Siteuri de socializare	11
3.2 Sisteme similare analizate	12
3.2.1 My space	12
3.2.2 LinkedIn	13
3.2.4 Ning	14
4. Analiză și fundamentare teoretică	16
4.1 Analiză	16
4.1.1 Arhitectura sistemului	16
4.2 Fundamentare teoretică	17
4.2.1 Framework-ul .Net	17
4.1.1.1 ASP.NET	19
4.1.1.2 Linq to SQL	22
4.1.1.2 AJAX	24
4.2.2 Cascading Style Sheets (CSS)	25
4.2.3 Microsoft SQL Server	26
5. Proiectare de detaliu și implementare	28
5.1 Modelul cazurilor de utilizare	28
5.1.1 Actorii sistemului	28
5.1.2 Cazurile de utilizare	29
5.2 Detalii de implementare	34
5.2.1 Componentele principale ale sistemului	35
5.2.2 Structura bazei de date	52
5.2.3 Modelul de date-LINQ	54
6. Testare și validare	56
7.1 Cerințele funcționale	56
7.2 Cerințe non-funcționale	56
7.2.1 Performanța	56
7.2.2 Securitatea	57
7.2.3 Scalabilitatea	57
7. Manual de instalare și utilizare	58
7.1 Manual de utilizare	58

Cuprins

7.2 Cerințe hardware și software	59
8. Concluzii și dezvoltări ulterioare	60
8.1 Concluzii	60
8.2 Dezvoltări ulterioare	60
BIBLIOGRAFIE	62
Acronime.....	63
Anexe	64
Anexa 1	64
Lista Figurilor.....	64

1. Introducere – Contextul proiectului

1.1 Context

Odată cu dezvoltarea internetului și creșterea accesului la internet rețele de socializare au devenit tot mai populare. Rețele de socializare pot fi văzute ca fiind cel mai nou tip de comunicare prin intermediul internetului după bloguri, forumuri și platformele de mesaje instant.

O rețea de socializare este o comunitate virtuală în care utilizatorii au o pagină web personală numită profil, care permite celorlalți să afle despre interesele și activitățile deținătorului profilului și să interacționeze cu acesta în multe alte moduri.

Scopul principal al rețelilor de socializare este de a crea noi prietenii sau a consolida legăturile între prieteni. Oamenii au încercat mereu să socializeze între ei, iar cu ajutorul rețelilor sociale acest lucru este mult mai ușor și se poate face chiar la nivel global. Unele din cele mai importante astfel de rețele sociale sunt Digg, LinkedIn, Facebook sau Twitter. Foarte multă lume a auzit de ele din care mulți le folosesc zilnic. Aceste comunități beneficiază de venituri prin reclame sau servicii suplimentare care trebuie plătite pentru utilizare.

Siteurile web de socializare au ajuns să fie mult mai populare decât e-mail-ul sau site-urile de știri. În momentul de față dacă sunt întrebat pe stradă câteva persoane despre internet majoritatea vor face legătura cu site-uri de socializare cum ar fi facebook, hi5 sau Myspace.

Popularitatea site-urilor de socializare introduce în procesul de dezvoltare al relațiilor, conceptul de comunicare mediata. Tinerii folosesc, în ziua de azi, siteuri de socializare pentru a cunoaște alți oameni de vârsta lor.

În ziua de azi există proceduri de organizare și software care controlează schimbul interpersonal de informații, în site-uri de socializare, mesagerie text, mesagerie instant, blog-uri, jocuri role-play online, educație online. Toate aceste aplicații pot fi grupate în termenul "social-media", un concept care descrie software-ul de interacțiune socială.

Multe persoane au cont pe o rețea de socializare și din ce în ce mai mulți fac parte din mai multe rețele de socializare. De aceea multe persoane de business încearcă să își creeze propria rețea de socializare și să o valorifice.

1.2 Tema și conținutul lucrării

Siteurile web de socializare sunt sisteme care permit utilizatorilor să își împartă informații personale, activități cu utilizatorii din lista de prieteni și alți membrii înregistrați ai site-ului. Acestea includ și alte aplicații integrate cum ar fi blogul, forumuri, galerii de fotografii sau camere de chat.

Utilizatorii siteurilor web de socializare își publică informațiile și conținuturi personale pentru ca ceilalți utilizatori să le vadă și să creeze oportunități de a interacționa cu prietenii lor și cu alți utilizatori cu care pot să se împrietenească.

Lucrarea de față urmărește implementarea unui site web care se încadrează în categoria siteurilor de socializare.

Siteul trebuie să permită socializarea între utilizatorii care s-au înregistrat pe site. Socializarea este reprezentată prin vizualizarea informațiilor altor utilizatori, adăugarea de conținut propriu care să poată fi vizualizat (blog, albume de fotografii), comentarea conținutului celorlalți utilizatori, trimiterea de mesaje private între utilizatori, comunicarea și într-o camera comună de chat. Deoarece siteul de socializare permite înregistrarea tuturor utilizatorilor care doresc, acesta are nevoie și de un administrator care să poată bloca accesul utilizatorilor cu intenții malițioase, să poată primi păreri și sugestii de la utilizatori și să

Capitolul 1

adauge administratori noi. Toate aceste funcționalități trebuie să fie reprezentate printr-o interfață cât mai prietenoasă ca să atragă cât mai mulți utilizatori deoarece cu cât sunt mai mulți membrii cu atât e mai atractivă comunitatea siteului și îi va face pe utilizatori să se întoarcă mereu la prietenii lor virtuali.

Toate aceste funcționalități au fost realizate folosind tehnologii furnizate de Microsoft, platforma de dezvoltare fiind .NET, iar baza de date este stocată în MS SQL Server.

2. Obiectivele proiectului

2.1 Obiective generale

Scopul principal al acestui proiect este implementarea unui site web eficient care să furnizeze cât mai multe funcționalități specifice sistemelor de socializare cu o interfață prietenoasă folosind tehnologii specifice platformei .NET și totodată specifice Visual Studio 2010: ASP, LINQ, AJAX ASP.NET.

Principalele funcționalități implementate sunt:

- Înregistrarea și autentificarea utilizatorilor;
- Crearea unei pagini web personale numită pagină de profil;
- Socializarea între utilizatori;
- Crearea unui pagini de blog;
- Trimitere de mesaje private între utilizatori;
- Conversații într-o cameră comună de chat;
- Administrarea sistemului.

2.2 Obiective specifice sistemului

2.2.1 Cerințe funcționale

Site-ul de socializare implementat trebuie să îndeplinească următoarele cerințe funcționale:

Pentru utilizatorii neautentificați:

- Înregistrarea în sistem pentru a se putea loga ulterior;
- Recuperarea parolei în cazul în care a fost uitată.

Pentru utilizatorii simpli:

- Logarea în sistem înainte ca utilizatorii să folosească site-ul;
- Crearea paginii de profil după înregistrare și modificarea ei după ce a fost creată;
- Căutarea prietenilor în funcție de numele de utilizator sau altă informație din pagina de profil;
- Vizualizarea profilurilor prietenilor sau a altor utilizatori înregistrați;
- Trimiterea de cereri de prietenie;
- Acceptarea sau respingerea cererilor de prietenie primite;
- Adăugarea albumelor de fotografii și a pozelor;
- Ștergerea de albume, fotografii sau poze;
- Comentarea fotografiilor prietenilor sau celor proprii.
- Trimiterea de mesaje între utilizatorii site-ului, vizualizarea mesajelor primite și marcarea mesajelor necitite, ștergerea mesajelor;
- Crearea blogului și primirea de comentarii referitoare la conținutul textului de la alți utilizatori, vizualizarea blogului altor utilizatori;
- Raportarea utilizatorilor;
- Trimiterea feedback administratorului.

- Schimbarea parolei;
- Dezactivarea contului.

Pentru administratori:

- Căutarea în lista de utilizatori înregistrați ai sistemului;
- Vizualizarea profilurilor utilizatorilor;
- Urmărirea activității utilizatorilor;
- Vizualizarea listei de utilizatori raportați;
- Posibilitatea de ștergere a utilizatorilor raportați;
- Crearea de conturi de administrator;
- Vizualizarea feedback-ului utilizatorilor;
- Delogarea.

2.2.2 Cerintele non-functionale

Cerintele non-functionale sunt cerintele care impun constrângeri cu privire la proiectare sau implementare, care nu sunt legate de funcții ale sistemului

2.2.2.1 Scalabilitatea

Scalabilitatea este o proprietate a unui sistem, care reprezintă capacitatea sistemului de a suporta corect un volum mai mare de încărcare, sau de a permite mărirea sau extinderea sa. Un sistem de prelucrare a datelor, este scalabil dacă el se comportă similar, fără defecțiuni, atunci când volumul de date pe care le prelucreză devine mai mare. De asemenea, un sistem poate fi considerat scalabil și dacă este capabil să ofere rezultate îmbunătățite în condițiile în care îi sunt adăugate resurse adiționale. Principalele sarcini la care sistemul trebuie să facă față pentru a fi scalabil sunt: numărul de utilizatori, cantitatea de resurse a sistemului, numărul de tranzacții și volumul de date stocat în baza de date.

În cazul de față scalabilitatea se referă la numărul de utilizatori care accesează web site-ul sau în cazul în care se vor adăuga noi cerințe funcționale modificările sau sistemul o să se extindă acesta să nu își piardă din performanță, corectitudine, fiabilitate, transparență și securitate.

Pentru ca un web site sau o aplicație să fie cât mai scalabilă este recomandată separarea diferitelor componente. Doar prin adăugarea unui server mai mare problema de performanță nu este rezolvată, acest server trebuie utilizat pentru componentele care au nevoie mai mare. Prin separarea componentelor se poate detecta mai ușor unde este nevoie de resurse suplimentare. Prin eliminarea dependențelor între componente scalabilitatea va fi mult îmbunătățită.

2.2.2.2 Toleranța la eșec

Principalul scop al tolerării la eșec este evitarea timpilor pierduți și asigurarea corectitudinii operațiilor chiar dacă se întâlnesc erori. Aceste sisteme care pot funcționa în condiții de eșec se numesc sisteme fiabile. Sistemele cu fiabilitate mare sunt construite în special dacă sistemul nu poate fi reparat sau dacă performanța sistemelor este critică. O disponibilitate mare este folosită dacă este foarte important pentru sistem să funcționeze non

stop. Scopul evitării erorilor este reducerea probabilității producerii de erori, dar și după aplicarea celor mai atente teste de detecția a erorilor acestea tot pot să apară. Toleranța la eșec este cea de-a doua metodă de îmbunătățire a fiabilității. Spre deosebire de evitarea erorilor, tehnica toleranței la eșec presune că eșecurile vor exista și de aceea este mult mai utilă. Prin această tehnică se asigură că sistemul va funcționa chiar dacă apar eșecuri.

2.2.2.3 Securitatea

Securitatea este abilitatea unui sistem responsabilă pentru prevenirea accesului neautorizat la informații și servicii. În mod tradițional problema securității este asociată cu bazele de date voluminoase. Trei forme fundamentale ale securității sunt confidențialitatea, integritatea și disponibilitatea. Autentificarea și controlul accesului sunt tehnicile utilizate pentru îndeplinirea confidențialității. Criptarea datelor este metoda folosită pentru păstrarea integrității datelor.

Când informația este citită sau copiată de o persoană neautorizată să facă aceste operații confidențialitatea este pierdută. Pentru anumite tipuri de informații confidențialitatea este un atribut foarte important (ex. informațiile de cercetare, specificații ale unui nou produs sau date de asigurare). În anumite condiții sunt reglementări care obligă protecția confidențialității utilizatorilor cum ar fi conturile bancare, afaceri online care cer date de pe card și conturi bancare, spitale, laboratoare medicale de testare, agenții care oferă servicii cum ar fi asistența psihologică, agenții care colectează taxele.

Dacă informațiile sunt modificate în moduri neașteptate atunci integritatea este pierdută. Aceasta înseamnă că modificările s-au făcut dintr-o eroare umană sau intenționat. Integritatea este foarte importantă pentru activitățile în care există informații care privesc siguranța și date bancare cum ar fi transferul bancar, controlul traficului aviatic și conturi bancare. Aceste informații pot fi șterse sau să devină indisponibile. Aceasta înseamnă că s-a pierdut disponibilitatea datelor, adică persoanele care sunt autorizate să acceseze informațiile nu reușesc[6].

2.2.2.4 Performanța

Se referă la timpul de răspuns al sistemului. Dintre caracteristicile unui sistem cele care pot să pericliteze performanța sunt numărul de utilizatori, tranzacțiile, resursele care sunt împărțite, complexitatea și dimensiunea bazei de date. Pentru a obține performanță mereu se face compromis între capacitățile sistemului și timpul de răspuns.

Caching-ul este una dintre cele mai simple și ieftine căi de a îmbunătăți performanța. Există o mare varietate de strategii la care se poate apela pentru a folosi caching-ul.

3. Studiu bibliografic

În cadrul acestui capitol vor fi analizate principiile componente și funcționalități ale siteurile de socializare, descrierea unor sisteme similare care se numără printre cele mai utilizare la nivel global și compararea acestora cu siteul de socializare descris în lucrare

3.1 Siteuri de socializare

Site-urile de socializare sunt aplicații web care permit indivizilor să construiască un profil public sau semi-public într-un sistem limitat, să conceapă o lista cu alți utilizatori cu care vor să fie conectați, să vadă și să traverseze lista acestora de prieteni, și operațiile făcute de alții în interiorul sistemului. Acestea sunt definite ca fiind un grup de interacții între diferiți membrii. Natura și nomenclatura acestor conexiuni poate să varieze de la un site la altul[13].

Site-urile de socializare, nu permit doar ca indivizii să cunoască noi persoane, dar permit utilizatorilor să alcătuiască și să facă publice informații în rețeaua lor de socializare(prieteni). Rezultatele pot fi legături între indivizi care în alte circumstanțe nu ar fi făcute, sau legături care sunt făcute între cunoștințe, care au conexiuni și în afara site-urilor. Pe multe site-uri de socializare, participanții nu caută neapărat să facă cunoștință cu alții, ci să relateze cu persoanele pe care deja le cunosc.

Rețelele sociale se bazează pe stabilirea și păstrarea conexiunilor între utilizatori facilitând mecanismele de comunicare pe internet: mesaje instant, textele chat, blogurile, comentele de tip wiki și comentariile [15].

În timp ce site-urile de socializare au implementat o multitudine de caracteristici tehnologice, componenta principală constă în profilurile vizibile care afișează listele de prieteni care sunt la rândul lor utilizatori ai sistemului. După înscrierea pe un site de socializare un individ este rugat să completeze un formular care conține o serie de întrebări, care în mod obișnuit includ descrieri precum vârsta, locația, interese și o secțiune ‘despre mine’. Majoritatea site-urilor încurajează utilizatorii să uploadeze cel puțin o poză de profil. Unele site-uri permit adăugarea de conținut multimedia, sau modificarea aparenței profilului. Alții, precum Facebook permite utilizatorilor să adauge module(‘aplicații’) prin care să dezvolte profilul lor.

Vizibilitatea unui profil depinde de site și de discreția utilizatorului. Din construcție profilele de pe Friendster și Tribe.net sunt parcurse de motoarele de căutare, făcându-le vizibile pentru oricine, indiferent dacă cel ce vizualizează are cont sau nu.

Site-uri precum Myspace permit utilizatorilor să aleagă dacă profilul lor este public sau este numai pentru prieteni. Facebook abordează acest subiect dintr-un alt punct de vedere. Utilizatorii care fac parte din același grup de prieteni pot să vadă profilul unuia altuia, cu excepția cazului în care proprietarul a decis să nu permită celor din grup să îi vadă profilul. Variații structurale precum vizibilitatea și accesul sunt criteriile de bază prin care site-urile de socializare se diferențiază între ele.

După înregistrarea pe un site de socializare, utilizatorii sunt rugați să îi identifice pe cei din sistem cu care au legătura. Etichetarea legăturilor depinde de site: ‘prieteni’, ‘contacte’, ‘fani’. Majoritatea site-urilor de socializare cer confirmarea bidirecțională a prieteniei, dar unele nu o cer. Aceste legături unidirecționale sunt uneori catalogate ca și ‘fani’ sau ‘urmasi’, dar majoritatea site-urilor îi numesc și pe acestia ‘prieteni’. Termenul de

‘prietenie’ poate duce în eroare, deoarece conexiunea nu înseamnă neapărat prietenie în adevăratul sens, iar motivul pentru care utilizatorii se interconectează sunt diverse.

Motoarele de căutare sociale sunt motoare de căutare care folosesc rețelele sociale pentru a filtra și priorita rezultatul căutării[15].

Afișarea publică a prietenilor este o componentă crucială a site-urilor de socializare. Listele de prieteni conțin legături către profilul fiecărui prieten permițând utilizatorilor să parcurgă graful rețelei dând click prin listele de prieteni. Pe majoritatea site-urilor listele de prieteni sunt vizibile pentru oricine îi este permis să viziteze profilul.

Majoritatea site-urilor de socializare oferă utilizatorilor un mecanism prin care aceștia pot lăsa mesaje pe profilurile prietenilor. Aceasta funcțiune înseamnă în general lăsaarea de ‘comentarii’. În plus site-urile de socializare au de obicei o caracteristică de private messaging similară cu cea de webmail. Deși atât comentariile, cât și mesajele private sunt populare, pe majoritatea site-urilor de socializare nu sunt disponibile pe toate site-urile[14].

În afară de profile, prieteni, comentarii și mesagerie privată, site-urile de socializare se deosebesc în general după caracteristici și utilizatori. Unele au capacitatea de video și foto-sharing, altele au incorporat tehnologie de blogging și mesaje instant. Există site-uri de socializare specific mobile(ex: Dodgeball), dar și altele bazate pe web suportă într-o măsură mai limitată interacțiuni prin interfața mobilă (ex:Facebook,Myspace și Cyworld). Multe site-uri de socializare au ca țintă oameni din regiuni geografice, lingvistice specifice, deși aceasta nu determină în totdeauna conținutul site-ului. De exemplu Orkut a fost lansat în SUA, cu o interfață numai în engleză, dar brazilienii, vorbitori de portugheză, au devenit repede grupul dominant de utilizatori. Unele site-uri sunt concepute special cu specific etnic, religios, orientare sexuală, politică sau alte categorii. Există site-uri de socializare chiar și pentru câini(Dogster) și pisici(Catster), deși proprietarii animalelor trebuie să se ocupe de profilele acestora.

3.2 Sisteme similare analizate

În acest subcapitol vor fi descrise trei dintre cele mai accesate siteurile de socializare existente care au fost studiate pentru a observa componentele și funcționalitățile principale ale acestui tip de siteuri web.

3.2.1 My space

Prima versiune de My Space a apărut în august 2003. În octombrie 2010 Myspace a introdus o versiune beta cu nou design al site-ului pe o scală limitată cu planuri de a modifica interfața tuturor utilizatorilor până în noiembrie. Cu introducerea acestei interfețe Myspace și-a schimbat orientarea dintr-un site de socializare general în unul orientat spre muzică pentru utilizatorii mai tineri.

Principalele componente ale siteului web de socializare Myspace sunt:

- Blog, multimedia și blurbs: profilurile conțin standardele ”blurbs”: cu secțiunile ”About Me” și ”Who I'd Like to Meet”, de asemenea și secțiunile ”Details” și ”Interests”. Profilul mai conține de asemenea și o parte de blog cu câmpuri standard pentru conținut și media.
- Galerii media, Myspace suportă și încărcări de imagini. Una dintre imagini poate fi aleasă ca imagine implicită, imagine care se va vedea în pagina principală a profilului, pagina de search și imaginea care va apărea lângă numele de utilizator la comentariile mesajelor. Este disponibil un editor de poze puternic creat de FotoFlexer. El poate să taie din imagini să le modifice

contrastul sau să le convertească în desene. De asemenea se pot adăuga și fișiere video. Toate aceste caracteristici pot fi ascunse în profil prin folosirea modului de customizare sau cod HTML și CSS. De asemenea pozele pot fi prezentate și în slide show.

- Secțiunea de comentarii, după "User's Friends Space" se găsește secțiunea de comentarii. În acest "spațiu" prietenii utilizatorului pot lăsa comentarii pentru toate postările lui. Utilizatorii Myspace au opțiunea de a șterge orice comentariu sau pot să ceară aprobare; comentariile pot apărea doar după ce utilizatorul le-a aprobat. Dacă unul din utilizatorii care au postat comentarii și-au șters contul comentariul lăsat de acel utilizator va fi înlocuit cu mesajul saying "This Profile No Longer Exists". De asemenea poate fi activată și opțiunea de utilizare a comentariilor în format HTML.
- Customizarea profilului cu HTML și CSS : Myspace permite utilizatorilor să își customizeze profilul cu cod HTML în secțiunile "About Me", "I'd Like to Meet" și "Interests", codul JavaScript nu este permis. În acest fel pot fi incluse Video-uri și conținutul bazat pe flash. Utilizatorii au și opțiunea de a adăuga muzică paginii de profil cu Myspace Music, un serviciu care permite formațiilor să posteze piese pentru a fi utilizate în Myspace.
- Buletinele sunt postări afișate în secțiunea de "bulletin board" pentru ca toți prietenii utilizatorului să le poată vedea. Buletinele pot fi utile pentru contactarea întregii liste de prieteni fără a fi nevoie să trimită mesaj fiecărui utilizator individual. Aceste au devenit primul punct de atac pentru publicatori. Buletinele sunt șterse în 10 zile de la data postării.
- Myspace Pools este o funcție prin care li se permite utilizatorilor să posteze voturi pe pagina de profil și să le împartă cu alți utilizatori. Pentru sistemul de forum Myspace utilizează o implementare a Telligent Community.

Myspace a avut componenta de Grup Care. Ea permitea unui grup de utilizatori să împartă o pagină comună și un bord de mesaje. Un grup poate fi creat de oricine, iar moderatorul grupului poate permite sau interzice accesul la grup. În noiembrie 2010 opțiunea de grup a fost retrasă, iar când un utilizator apasă pe link-ul pentru grupuri acesta este redirecționat spre o pagină care informează utilizatorul că opțiunea de grup a fost scoasă momentan și că se va anunța când va fi introdusă din nou. La fel s-a întâmplat și pentru pagina de help. Aceste opțiuni se vor relua dar nu a fost încă anunțată data în care o să apară[14].

Asemănările dintre Myspace și siteul web descris în această lucrare sunt pagina de profil, albumele de fotografii, comentariile, partea de socializare unde utilizatorii pot avea prieteni, contul utilizatorului unde informațiile profilului pot fi schimbare și partea de căutare. Siteul de socializare Friends implemntat are în plus partea de blog care a fost scoasă din Myspace

3.2.2 LinkedIn

LinkedIn este un site de socializare orientat spre business. A fost fondat în decembrie 2002 și lansat în mai 2003 fiind utilizat în general pentru socializare profesională. În 2011 LinkedIn are mai mult de 100 de milioane de utilizatori înregistrați din mai mult de 200 de țări și teritorii din întreaga lume. Site-ul este disponibil în engleză, franceză, germană, italiană, portugheză și spaniolă. În fiecare lună are 21.4 milioane de utilizatori unici din US și 47.6 global. Cu ajutorul linkedIn se poate face schimb de cunoștințe, idei și oportunități. Informațiile principale din profil sunt cele mai recente locuri de muncă ale utilizatorilor, proiecte realizate și informații de contact[13].

Principalele componente ale acestui site de socializare sunt:

- Pagina de start unde vor apărea noutățile și modificările făcute de prietenii utilizatorului sau care din vechii colegi ai utilizatorului și-au făcut cont nou pe site, cine a vizitat profilul utilizatorului și cine s-a alăturat grupului de prieteni;
- Pagina de profil este pagina care afișează informațiile utilizatorilor și conține referințe spre componentele care permit conectarea cu alți utilizatori, prezintă realizările profesionale ale utilizatorului și ajută la controlarea identității profesionale online.
- Componenta locuri de muncă permite căutarea locurilor de muncă în funcție de anumite cuvinte cheie, adăugarea unui nou loc de muncă și explorarea oportunităților în carieră.
- Secțiunea de grupuri permite crearea sau alăturarea unui grup. În grupuri utilizatorii pot vedea cele mai populare discuții din grupul profesional, pot adăuga comentarii la aceste discuții și să urmeze persoanele importante din grup după care să apară pe pagina de profil toate postările persoanei respective.
- Componenta numită "LinkedIn today and signal" permite vizualizarea articolelor cele mai citite postate din grupul de prieteni de cine au fost publicate articole, cine și ce comentează despre ele.
- Pagina de companie este o pagină specială care este creată pentru companii. Reprezintă profilul companiei. Cu ajutorul acestei pagini compania poate să ia legătura cu utilizatorii sistemului de care este interesată, iar utilizatorii pot găsi informații despre companii care au pagina setată.
- Răspunsuri este o secțiune care permite tuturor utilizatorilor să posteze întrebări și să primească răspuns de la oricare din utilizatorii rețelei [10].

Spre deosebire de siteul de socializare Friends descris în această lucrare LinkedIn are specific faptul că este orientat spre business pe când acesta este un site de socializare general. Asemănările între cele două platforme sunt paginile de profil, faptul că utilizatori pot posta noutăți și partea de transmitere mesaje între utilizatori. Deoarece site-ul web de socializare Friends este unul general are și funcționalități care permit adăugarea de albume de fotografii, blogging și comunicarea în camera de chat.

3.2.4 Ning

Ning este platforma online numărul 1 care permite organizatorilor, militanților și persoanelor de influență din lume să își creeze propria rețea socială. Acest sistem reprezintă o viziune diferită a siteurilor de socializare.

Proiectează o aplicație socială personalizată în mai puțin de 60 de secunde. Ning este o platforma online care permite utilizatorilor să își creeze propriul lor site de socializare și a fost lansat în octombrie 2005. Se estimează că în prezent are lunar 7.4 milioane de utilizatori unici.

Ning concurează cu site-uri de socializare cum ar fi Myspace, Facebook și Bebo apelând la utilizatori dispuși să își creeze propriul lor site de socializare în funcție de interesul fiecărui utilizator, folosindu-și propriul design și selectând doar opțiunile de care are nevoie.

Principala caracteristică a site-ului de socializare Ning este că oricine își poate crea propriul site de socializare cu un subiect particular și nevoi specifice. Când a fost lansat Ning a fost o platformă gratuită pentru dezvoltare și găzduire. Codul sursă al aplicațiilor Ning a fost disponibil utilizatorilor, iar serviciile care îl creau erau ușor de ramificat pentru ca utilizatorii să își creeze o aplicație Ning doar modificând codul PHP și rulând ca fiind propriul cod. Aplicațiile includ albume de fotografii, bloguri, și multe alte metode de împărtășit

conținut. În septembrie 2006 Ning își îndreaptă atenția spre a oferi un grup de website-uri: un website de fotografii și un website pentru filme pe care utilizatorii să le copieze și să le folosească. Aceste trei șabloane au fost înlocuite de unul singur modificat de aplicații menite să permită oricui să își creeze propriul site de socializare.

Ning oferă dezvoltatorilor să dețină un nivel de control aspra propriilor rețele de socializare prin permiterea modificărilor în funcționalități și în logica rețelei sociale.

Viitoare modificări vor înlocui întreaga sursă de cod cu abilitatea de a aduce noi funcționalități sau schimbări în partea de logică a funcționalităților existente prin OpenSocial și un set de noi API lansate după 2009.

Initial Ning a avut opțiuni gratuite și plătite dar le-a schimbat în trei nivele de servicii plătite. Fiecare nivel de plată oferă diferite grade de funcționalități. Rețelele sociale care rulează pe serviciul Ning sunt programate cu PHP, iar platforma este contruită în Java.

În 2008 Ning își anunță parteneriatul cu Scropts4Ning integrând produsele pentru programatori în Ning și oferindu-le gratuit. În 2009 Ning adaugă interfață pentru iPhone și face rețeaua mult mai compatibilă pentru mobile[15].

Pentru opțiunea platită pro sistemul permite siteului de socializare toate componentele necesare unei rețele de socializare: bloguri, forumuri, fotografii, videoclipuri, grupuri, evenimente, chat, interfață și adresă URL personalizate, integrări sociale, redări de fișiere media personalizate, încărcarea fișierelor de muzică și video, opțiuni de creare a aplicațiilor mobile și acces la API pentru personalizare avansată[11].

Carcateristicile comune ale platformei Ning și miniplatforma Friends descrisă în lucrare sunt bloguri, forumuri, fotografii, evenimente și chat.

Studierea și utilizarea funcționalităților acestor siteuri de socializare m-a determinat să doresc să implementez un astfel de sistem care să dețină funcționalități din toate sistemele studiat.

4. Analiză și fundamentare teoretică

Orice aplicație software presupune în faza de început a procesului de dezvoltare alegerea tehnologiilor folosite și alegerea unei soluții arhitecturale adecvate. În continuare vor fi analizate tehnologiile și arhitectura aleasă.

4.1 Analiză

Pentru implementarea sistemului am ales arhitectura pe trei nivele. La crearea paginilor web este una din cele mai avantajoase deoarece permite separarea nivelului de prezentare de cel de logică și de cel de acces la baza de date. Ca tehnologii am ales tehnologiile Microsoft deoarece acestea oferă un mediu consistent de programare orientat obiect.

4.1.1 Arhitectura sistemului

Unul din principalele avantaje ale arhitecturii pe trei nivele față de arhitectura client server îl reprezintă că nivelul de prezentare este separat de cel de logică. Introducerea unui nivel în plus face ca gestionarea controalelor utilizator să fie mult mai ușoară și reduce amestecul de cod pentru interfața și cel pentru partea de logică. În plus o arhitectură pe trei nivele este mult mai ușor de scalat sau modificat, iar puterea de procesare a calculatorului este mai mică pentru fiecare client ceea ce duce la un cost scăzut. Încă unul dintre motivele pentru care arhitectura pe trei nivele este mai utilă este faptul că aceasta permite caching, fapt ce reduce traficul adițional pe internet. În mod normal un server are putere de calcul mult mai mare decât calculatoarele obișnuite este mai rapid, avantajul fiind că partea de logică este pe server.

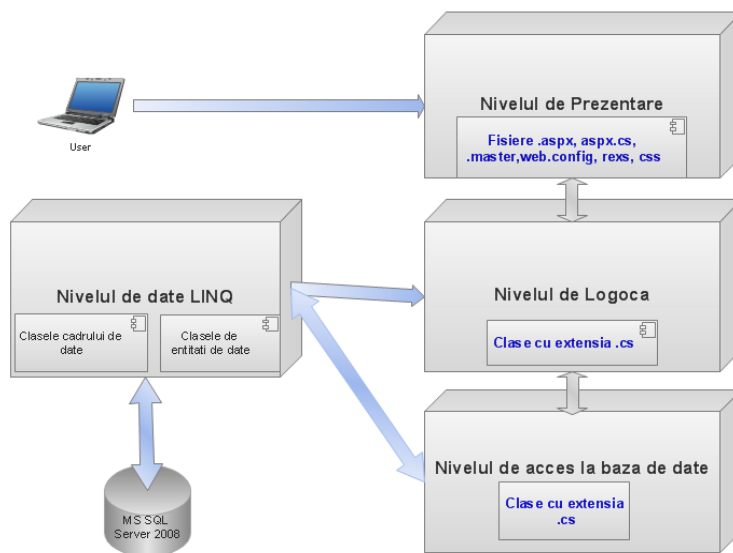


Figura 4.1 Arhitectura de ansamblu a sistemului.

Acest tip de aplicații fac funcțiile mult mai ușor de reutilizat și de găsit, codul este mult mai structurat, mai ușor de citit și înțeles, iar cazul în care ulterior alți programatori vor să schimbe din funcționalități vor putea să o facă cu ușurință.

- Nivelul de prezentare sau de aplicație: este componenta care furnizează interfața pentru utilizatorii care folosesc aplicația web. Acest nivel este folosit pentru setarea și primirea datelor de la baza de date la utilizator.
- Nivelul de logica este nivelul folosit pentru a scrie funcțiile care îndeplinesc rolul de mediator între nivelul de prezentare și cel al accesului la baza de date. Acest nivel conține și un subnivel numit nivel de proprietăți care are clase ce reprezintă informațiile (tabelele) din baza de date.
- Nivelul accesului la baza de date are rolul de a furniza datele necesare aplicației din baza de date. Acest nivel interacționează numai cu baza de date. În cazul de față nivelul accesului la baza de date este reprezentat de Linq to Sql.

Acest model de baza pe trei niveluri a constituit baza arhitecturala a aplicației de socializare *Friends*. În funcție de aceste niveluri contextul de date LINQ to SQL, entitățile claselor și logica de construcție se află în nivelul accesului la baza de date, iar entitățile non persistente, sunt implementate în nivelul de logică.

Nivelul de acces la baza de date aplează metode din nivelul intermediar, iar nivelul accesului la baza de date va executa interogările care sunt mapate în metodele clasei *Data Context* creată de Linq. Nivelul de mijloc returnează clienților conținut XML, reprezentări de entități sau un proxy al obiectelor.

În nivelul de mijloc entitățile sunt create de contextul de date care urmărește stările acestora și gestionează diferite forme de încărcare și modificări ale bazei de date. Aceste entități sunt atașate contextului de date, dar după ce aceste entități sunt transmise altui nivel pentru serializare acestea sunt detașate de contextul de date ceea ce înseamnă că stările entităților nu mai sunt monitorizate de contextul de date. Entitățile pe care clientul le trimite înapoi pentru a le modifica trebuie reatașate contextului de date înainte ca Linq să salveze modificările în baza de date.

Nivelul de prezentare este responsabil pentru furnizarea de valori inițiale sau marcaje de timp înapoi la nivel de logică în cazul în care acestea sunt necesare pentru controale de concurență optimistică.

4.2 Fundamentare teoretică

4.2.1 Framework-ul .Net

Framework-ul .Net este o componentă integrată Windows care suportă crearea și execuția generațiilor următoare de aplicații moderne și servicii Web. Componentele cheie a Framework-ului .Net sunt CLR (Common Language Runtime) și clasa de librării ale cadrului .Net care includ Ado.NET, ASP.NET, Windows Forms și WPF (Windows Presentation Foundation).

Framework-ul .Net furnizează un mediu de execuție simplificând developmentul și deployment-ul cu o mare varietate de limbaje de programare, aceasta ajutând la portabilitatea codului compilat între diferite calculatoare cu sistem Windows, cât și la reutilizarea codului în programe, indiferent de limbajul de programare utilizat.

Lansare primei versiuni a framework-ului .Net a fost începutul unei schimbări radicale a proiectării software. Rezultatul muncii celor de la Microsoft a fost o tehnologie matură de care dezvoltatorii se pot folosi pentru a realiza orice de la aplicații Windows până la executarea interogării bazelor de date și crearea de siteuri web folosind ASP.NET.

Vasta colecție de funcționalități pe care Framework-ul o furnizează este organizată astfel încât fiecare din miile de clase ale cadrului sunt grupate într-un container ierarhic și logic numit spațiu de nume (namespace).

Diferite spații de nume(namespace-uri) oferă diferite funcționalități. Luate împreună spațiile de nume furnizează funcționalități pentru aproape orice aspect al dezvoltării de aplicații distribuite. De la punerea în coadă a mesajelor la securitate. Această unitate este numită libraia de clase.

Componetele cheie ale framework-ului .NET sunt Common Language Runtime, librăria de clase .NET și unificarea componentelor.

Common Language Runtime (CLR) este un nivel între o aplicație și sistem de operare pe care aceasta rulează. CLR-ul simplifică design-ul și scurtează codul pe care programatorii trebuie să îl prin varietatea serviciilor de execuție cum ar fi managementul memoriei, al firelor de execuție, al timpului de viață al componentelor și tratarea implicită a erorilor.

CLR-ul este cel care verifică codul înainte ca acesta să fie executat. Când codul este executat prima dată CLR-ul invocă în compilator special numit Just In Time care transformă IL(Intermediate Language) în instrucțiuni executabile care sunt specifice tipului și modelului procesorului sistemului pe care rulează.

Una din cele mai importante caracteristici ASP.NET este că rulează în mediul de parcurgere CLR.

Printre principalele avantaje ale CLR-ului sunt:

- Managementul automat al memoriei și colectarea elementelor care nu mai sunt utilizate: de fiecare dată când aplicația este instanțiază o referință a unui obiect CLR-ul alocă spațiu în locul gestionat pentru acel obiect, dar niciodată dezvoltatorul nu trebuie să elibereze acest spațiu manual. După ce referințele spre acest obiect sunt șterse, obiectul devine disponibil pentru colectorul de deșeuri. Garbage collection rulează periodic în CLR recuperând memoria pentru obiectele inaccesibile.
- Când este compilată o aplicație .NET adaugă informație asamblului care indică detalii cum ar fi disponibilitatea clasei, membrii, tipurile de date etc. Ca rezultat, alte aplicații le pot folosi fără a cere fișiere suport adițional, iar compilatorul poate verifica dacă fiecare din apeluri este valid. Acest nivel suplimentar ajută la evitarea erorilor de nivel scăzut.
- Informațiile despre clase și membrii sunt doar un tip de metadată care este stocat de ansamblul de compilare .NET. Metadatele descriu codul și permit furnizarea de informații suplimentare către runtime sau altor servicii.
- Limbajul .NET oferă o tratare a erorilor structurată care permite organizarea codului de tratare a erorilor. Pot fi create blocuri separate care să gestioneze diferite tipuri de erori. De asemenea erorile se pot imbrica pentru a trata diferite nivele de excepții.

- CLR furnizează o multitudine de fire de execuție care poate fi utilizată de diferite clase. De exemplu se pot apela metode, citi fișiere sau comunica cu servicii web asincron fără a fi nevoie de crearea explicită a firelor de execuție.

4.1.1.1 ASP.NET

ASP reprezintă acronimul pentru Active Server Pages, una dintre metodele de construire dinamică a paginilor de Web. World Wide Web-ul este, fără îndoială, cea mai importantă utilizare a Internet -ului, este o lume virtuală în care se găsește totul, e o lume care schimbă viața noastră într-un mod fără precedent și care a pătruns din ce în ce mai mult în toate domeniile de activitate.

ASP.NET este o tehnologie gratuită pentru dezvoltarea de site-uri Web moderne, de la cele simple la servicii online și aplicații Web complexe și prezintă îmbunătățiri semnificative de la vechiul ASP, componenta inclusă în .NET Framework pentru procesarea codului din partea de server din paginile Web dinamice sau serviciile Web. Așadar ASP.NET este o tehnologie de tip server, noua generație ASP și nu o versiune îmbunătățită a acestuia. Față de vechiul ASP, ASP.NET oferă un model de programare OOP [6].

Principalele caracteristici ale lui ASP.Net sunt următoarele:

- este o tehnologie gratuită;
- este o tehnologie de tip server;
- Serverele Web pe care se pot găzdui aplicațiile și site-urile Web dezvoltate cu ASP.NET sunt Microsoft Internet Information Services – IIS și Apache, prin proiectul Mono;
- Paginile aspx și logica acestora, dezvoltate într-unul din limbajele platformei .NET Framework - C#, VB.NET și altele - sunt procesate, respectiv executate pe serverul Web;

Orice pagină ASP.NET - cunoscută și sub numele de *Web Form* - presupune două componente:

- Partea vizuală, definită într-un fișier cu extensia .aspx - controale, aspect, stiluri etc, într-un limbaj declarativ îmbogățit, ce poate conține HTML, XML și cod client (JavaScript, apeluri AJAX);
- Codul .NET, specificat într-un fișier cu extensia .aspx.cs - pentru cod C# - sau .aspx.vb - pentru cod VB.NET - care va fi executat pe server atunci când se face o cerere către respectiva pagină;

Începând cu versiunea ASP.NET 3.5. dezvoltatorii au la dispoziție tehnici moderne de dezvoltare cum sunt AJAX și MVC și pot integra aplicațiile Web cu alte tehnologii moderne cum sunt Silverlight și ADO.NET 3.5. [6]

Alte trăsături importante ale lui ASP.NET sunt următoarele:

- Permite includerea de mai multe funcționalități pe controalele ASP.NET iar managementul stării lor este mult îmbunătățit;
- Permite programarea bazată pe evenimente, cu diferențe minime față de programarea aplicațiilor windows;
- Este senzitiv la browserul pentru care se face trimiterea codului HTML, alegând automat codul optimizat pentru o colecție de browsere;
- Erorile grave apărute în aplicație ASP.NET nu vor duce la oprirea serverului Web (dar pot duce la imposibilitatea de a mai executa pagini ASP.NET până la repornirea serverului)

- Permite scrierea codului care se va executa pe server în orice limbaj .NET: C#, VB.NET, etc.

Spre deosebire de aplicațiile windows ce necesită ca .NET Runtime să fie instalat pe sistemul de calcul al clientului, la websiteurile realizate în ASP.NET nu este necesară existența acestuia numai pe partea de server. Este suficientă existența doar a unui browser modern pentru a putea rula acest gen de aplicații.

ASP.NET 4

Principalele îmbunătățiri ale ASP.NET4 față de celelalte versiuni sunt:

- Navighează cu fișiere de browsere updatate ceea ce înseamnă că partea de server poate recunoaște și furniza în mod corespunzător o gamă mai largă de browsere. Între acesta sunt incluse: Google Chrome, Internet Explorer 8, Firefox 3.5, Opera 10, Safari 4, dar și browserele pentru tehnologii mobile cum ar fi BlackBerry, iPhone, iPod, și sistemul de operare Windows Mobile.
- Are unelte de desfășurare mai bune: Visual Studio permite crearea de pachete web, fișiere comprimate care conțin aplicații și alte detalii cum ar fi SQL Server și setările IIS. Pachetul web lucrează cu un nou fișier web.config care permite separarea clară a setărilor care se aplică pentru build-ul test al aplicației și cel care se aplică instanțelor desfășurate.
- Caching-ul extensibil este una din noile funcționalități ale ASP.NET 4, acesta permite folosirea unor noi tipuri de stocare cache incluzând cache distribuit.
- Controlul graficelor

MODELUL CODULUI. Există două opțiuni de utilizare a codului pentru programarea paginilor web și modul în care evenimentele care apar în pagină sunt tratate în cod. Acestea sunt:

- Colul în linie (Inline code) acest model este mai apropiat de modul tradițional ASP. Atât codul cât și marcasele HTML sunt stocate într-un singur fișier cu extensia .aspx. Tot codul este încorporat în unul sau mai multe blocuri de script. Chiar dacă codul este într-un script bloc acesta nu își pierde suptul pentru debug și nu este nevoie să fie executat liniar. Acest mod este mai comod deoarece păstrează totul într-un singur loc, fiind popular pentru codificarea paginilor web simple.
- Code-behind acest mod separă fiecare pagina web în două fișiere diferite: un fișier .aspx fișierul de marcare cu cod HTML care va conține controale și un fișier de cod .cs (în cazul în care codul este scris în C#) care conține codul sursă pentru pagină. Acest model furnizează o organizare și separă interfața utilizator de logica de programare, lucru important în dezvoltarea paginilor web mai complexe. De asemenea modelul code-behind este mai curat și indică explicit clasa creată și spațiile de nume importate. Acest model introduce și posibilitatea ca partea de web design să fie rafinată fără a se modifica deloc codul de logică.

CONTROALELE UTILIZATORILOR

Setul de bază al controalelor ASP.NET este larg și impresionant. Include controale care încasulează taguri HTML elementare care oferă un model de nivel înalt cum ar fi calendarul, TreeView-urile și controalele de date. Dar nevoile de implementare nu pot fi acoperite nici cu

cel mai impresionant set de controale. În .NET există două moduri de creare a controalelor proprii:

- Controale utilizator – este o mică secțiune a unei pagini care poate include cod static HTML și servere de controale web. Avantajul controalelor utilizator este acela ca o dată ce au fost create acestea se pot folosi ori de câte ori în aceeași aplicație web. Pot fi adăugate până și proprietăți, evenimente și metode proprii.
- Controale *custom server* – sunt clase compilate care generează programatic propriul lor HTML. Spre deosebire de controalele utilizator controalele server sunt mereu precompilate în asambluri DDL. Depinzând de cum este creat controlul server conținutul se poate implementa de la 0, moșteni aparențe și comportament de la un control web și extinzându-i caracteristicile, sau construind interfața prin instalarea și configurarea unui grup de controale constituant.

MASTER PAGES

Paginile master sunt o inovație foarte importantă acestea oferind posibilitatea de a crea șabloane reutilizabile pentru paginile web. Folosind pagini master se poate defini schema pentru toate paginile website-ului cu toate detaliile obișnuite cum ar fi capătul, bara de meniu, banere etc. După ce paginile master au fost create acestea se pot folosi în tot website-ul asigurându-se astfel că toate paginile vor avea același design. Astfel vizitatorii pot naviga prin diferite secțiuni fără să își dea seama de schimbări.

Paginile master sunt o trasatură a ASP.NET care este utilă la standardizarea interfeței paginilor web din întregul proiect. Dacă se folosește aceeași pagină master trebuie să se țină cont de faptul că orice schimbare în pagina master va schimba automat orice altă pagină care o folosește.

Pentru a crea o pagină master practică și flexibilă trebuie să se țină cont de câteva recomandări:

- Abilitatea de a defini o porțiune din pagina separată astfel încât pagina master să se poată folosi la cât mai multe pagini din proiect.
- Abilitatea de a crea porțiuni din pagină editabile. Paginile care refolosesc acest master sunt constânse să modifice doar conținutul din aceste porțiuni editabile.
- Permitearea utilizatorilor să modifice elementele care se utilizează în toate paginile.
- Abilitatea de a construi cu ajutorul ContentPlaceHolders (regiuni care pot fi modificate) porțiuni în care se poate defini conținut specific pentru fiecare pagină care folosește pagina master în parte.

Paginile master sunt similare paginilor web form ASP.NET obișnuite. Paginile web form pot conține cod HTML, controale web și partea de code behind. Una din diferențe este că în timp ce paginile web form încep cu directiva Page, paginile master încep cu directiva Master. De asemenea în paginile master se poate folosi controlul ContentPlaceHolder, acesta este un container al paginii în care paginile care folosesc pagina master vor putea să îl editeze.

În timpul rulării paginile master pargurg pașii următori:

- Utilizatorul cere o pagină prin scrierea URL-ului paginii conținutului.
- Când pagina este preluată este citită directiva paginii. Dacă pagina conține o directivă de referință a unei pagini master este citită și aceasta. Dacă pagina este accesată prima dată atunci se compiează amândouă paginile.

- Pagina master și conținutul încărcat este îmbinat în arborele de controale al conținutului paginii.
- Conținutul containelelor individuale este pus în controalele ContentPlaceHolder din paginile master.
- Pagina rezultată este trimisă browserului.

Din perspectiva utilizatorilor combinația dintre pagina master și pagina cu conținut se vede că o pagină singură. URL-ul este cel al paginii cu conținut.

Din punct de vedere programatic cele două pagini acționează ca două containere separate. Pagina care folosește pagina master se comportă ca un container pentru pagina master.

Este de reținut faptul că pagina master devine o parte din pagina care o folosește. Deci pagina master se comportă ca și un user control: ca o componentă paginii și ca un container în acea pagină. Astfel pagina master este container pentru toate celelalte controale server afișate în browser.

4.1.1.2 Linq to Sql (Language Integrated Query)

LINQ to SQL este o componentă a Visual Studio care furnizează o infrastructură runtime pentru maparea datelor relaționale ca obiecte fără a pierde abilitatea de a le interoga. LINQ rezolvă această problemă prin traducerea interogărilor în limbaj pentru executarea lor în baza de date, după care se face traducerea lor înapoi la obiectele definite de dezvoltator după care obiectele pot fi folosite fără probleme pentru a manipula obiectele în timp ce LINQ to SQL monitorizează și salvează modificările asupra acestora.

LINQ este una din cele mai utile unelte adăugate la framework-ul .NET fiind un set de extensii care permit realizarea interogărilor în limbajul C#. Linq definește cuvinte cheie care se folosesc pentru realizare expresiilor query. Aceste query-uri pot realiza: selectare, filtrare, sortare, grupare și transformarea datelor stocate în baza de date. Diferite extensii LINQ to Objects permit cererea de colecții din diferite surse de date.

Linq to SQL mapează modelul relațional într-un model de obiect exprimat într-un limbaj de programare.

Atributul tabel are proprietatea *Nume* prin care se poate specifica numele exact al tabelului din baza de date. Dacă nici o proprietate nume nu este specificată Linq to Sql va lua numele tabelului la fel cu cel al clasei. Doar instanțele clasei declarate ca tablele vor fi stocate în baza de date. Instanțele acestor tipuri de clase sunt cunoscute sub numele de entități, iar clasele sunt cunoscute ca și clase de entități.

În plus pentru asocierea claselor la table este nevoie de specificarea fiecărui câmp sau proprietate care va fi asociată cu coloana bazei de date. Pentru aceasta Linq to Sql definește atributul coloană. Acest atribut are o varietate de proprietăți care sunt utilizate pentru customizarea exactă, pentru maparea între câmpurile din aplicație și coloanele bazei de date. Una din proprietățile cele mai importante este proprietatea id. Aceasta specifică cheia primară a tabelului.

Pentru accesarea atributului "tabel" este nevoie doar de furnizarea informațiilor din coloane, dacă se poate diferenția de ceea ce putem deduce din câmpul sau proprietatea declarată. Doar câmpurile sau proprietățile declarate ca și coloane pot fi persistate sau scoase din baza de date. Celelalte sunt considerate ca părți tranzitorii ale logicii aplicației.

DataContext este intermediarul principal prin care se extrag obiectele și prin care se realizează schimbările din baza de date. Este inițializat cu o conexiune sau o conexiune de stringuri care este furnizată de programator. Scopul *DataContext* este de a traduce cererile de obiecte din aplicație în interogări SQL făcute asupra bazei de date și apoi mapate în obiecte reprezentând rezultatul. *DataContext* permite limbajului integrat pentru interogări să implementeze aceiași operatori ca și operațiile standard de interogare cum ar fi clauzele "Where" și "Select".

Relațiile în bazele de date relaționale sunt modelate ca și chei străine a căror valori se referă la cheile primare ale altor tabele. Pentru navigarea între acestea cele două tabele trebuie să fie legate explicit folosind operația de *join*. Obiectele pe de altă parte se referă unul altuia folosind referințe de proprietăți sau colecții de referințe navigarea făcându-se utilizând operatorul "punct".

Linq to Sql definește atribute de asociere care se pot aplica pentru reprezentarea relațiilor. O relație de asociere este o relație de tip cheie-străină la cheie-primară care este făcută prin potrivirea valorilor coloanelor între tabele.

Deoarece puține aplicații sunt create doar cu interogări, Linq to Sql este proiectat pentru a oferi o flexibilitate mare în manipularea și salvarea schimbărilor făcute obiectelor. Imediat ce obiectele entități sunt disponibile (fie retrase printr-o interogare din baza de date, fie construite obiecte noi) acestea trebuie să poată fi manipulate ca obiecte normale ale aplicației, să li se poată schimba valorile, să se adauge sau să fie șterse din colecții etc. Linq to Sql monitorizează toate schimbările care sunt făcute asupra obiectelor și sunt transmise la baza de date imediat ce acestea sunt terminate.

Când metoda *SubmitChanges()* este apelată, Linq to Sql generează automat și execută comanda sql pentru a transmite schimbările la baza de date. Este de asemenea posibil să fie suprascrisă această metodă în funcție de nevoile programatorului.

Pentru a evita executarea multiplă a interogărilor rezultatul poate fi convertit în clase ale colecțiilor standard; rezultatul este ușor de convertit într-o listă sau array folosind metodele *ToList()* sau *ToArray()*.

Este de reținut faptul că dacă baza de date conține tabele fără cheie primară Linq To Sql permite să facă noi inserări în tabel dar nu permite modificarea rândurilor deja existente. Aceasta se întâmplă deoarece Framework-ul nu își dă seama care rând să îl modifice pentru că nu îl poate identifica din cauza lipsei cheii primare.

Având în vedere câte modificări se fac asupra obiectelor, acestea se execută doar în memorie asupra replicilor. Asupra datelor actuale din baza de date nu se produce nici o modificare. Transmiterea datelor modificate se va face doar după ce este explicit apelată metoda *SubmitChanges()* asupra *DataContext*. La apelarea metodei *SubmitChanges()*, *DataContext* va încerca să translateze schimbările în comenzi Sql echivalente: inserare, updatate, adăugare, schimbare sau ștergere în tabelele corespunzătoare. Primul lucru care se întâmplă la apelarea metodei *SubmitChanges()* este examinarea obiectelor pentru a determina dacă noile instanțe au fost atașate. Aceste noi instanțe sunt adăugate la setul de obiecte urmărite. În pasul următor toate obiectele cu schimbări în așteptare sunt ordonate într-o secvență de obiecte bazată pe dependențele între acestea. Acele obiecte ale căror schimbări depind alte obiecte sunt secvențiate în funcție de dependențe. Constăngerile de chei străine și

de unicitate în bazele de date au un rol important în determinarea ordinii corecte a schimbărilor. Înainte ca orice schimbare să se execute este începută o tranzacție pentru a încapsula seria de comenzi individuale. În final, una câte una, schimbările din obiecte sunt translate în comenzi SQL și trimise către server. În acest punct, orice erori detectate de baza de date vor genera întreruperea procesului și se declanșează o excepție. Toate modificările produse în baza de date până la apariția excepției se vor anula. DataContext va conține încă modificările făcute asupra obiectelor astfel se poate încerca remedierea problemei și reîncercarea modificării în baza de date prin replearea metodei.

4.1.1.3 AJAX

AJAX(Asynchronous JavaScript and XML) a devenit una din cele mai utilizate tehnologii moderne în mare parte aceasta datorându-se Google Suggest și Google Map. În ASP.NET Ajax permite procesărilor server-side să se execute fără a fi necesară reîncărcarea paginii.

Ajax este un grup de metode pentru dezvoltarea paginilor web utilizate pe partea de client pentru a crea aplicații web cât mai interactive. Cu ajutorul Ajax aplicațiile web pot să trimită și să primească informații spre și de la server în mod asincron (în background), fără a interfera cu interfața sau comportamentul paginii existente.

Începând de la ASP.NET 2.0 Microsoft a construit un modul de servicii bloc care fac personalizarea aplicațiilor web mult mai simplă. Unul din aceste seturi este controalele server AJAX ASP.NET. Componenta principală a acestui model server-side este ScriptManagerul. Acesta este un control web care nu are aparență vizuală pe pagina, dar face legătura spre linbraria ASP.NET Ajax.

Pentru adăugarea unui ScriptManager este suficient să fie tras din *ToolBox* de la tabul de Ajax Extensions se va genera urmatorul cod în pagina aspx :

```
<asp:ScriptManager ID="ScriptManager1" runat="server"> </asp:ScriptManager>.
```

Fiecare pagina care folosește ASP.NET Ajax necesită o instanță de ScriptManager. Se poate crea doar o singură instanță a ScriptManagerului.

Pe lângă crearea legăturilor pentru librăriile ASP.NET AJAX ScriptManager efectuează și alte task-uri importante. Acesta poate face referințe spre alte fișiere de script (de obicei cu alte controale și componente conținute de ASP.NET AJAX), să creeze proxy care permite aplearea serviciilor web în mod asincron; și să gestioneze modul în care UpdatePanel va face reîncărcarea propriului conținut.

4.1.1.4 Javascript

JavaScript este un limbaj de scripting interpretat, orientat spre obiecte, care poate fi folosit pentru a adăuga un comportament interactiv simplu unei pagini HTML prin intermediul unui script de cuvinte-cheie inserate într-o pagina web. Codul JavaScript este inserat direct într-un alt document (de obicei o pagină HTML). Codul este descărcat de computerul clientului și este executat de către browser.

Există două moduri de a adăuga cod JavaScript în paginile web:

- Se poate pune direct într-un atribut al evenimentului conținut de un element HTML. Acesta este modul cel mai direct de folosire a codului de JavaScript folosit doar pentru părți mici de cod.
- Se pot adăuga tag-uri `<script>` care conțin cod JavaScript. Se poate alege opțiunea de a executa acest cod automat când se încarcă pagina web, sau se pot implementa funcții care se vor rula la apelul unui eveniment din partea clientului.

Un fișier javascript are extensia .js. Pentru includerea unui script într-o pagina web se folosește construcția:

`<script type="text/javascript"> functii_javascript </script>`, iar pentru includerea unui fișier javascript în pagină se folosește construcția: `<script type="text/javascript" src="cale/fisier.js"></script>`, ambele fiind incluse în tagul `<head>` al paginii.

4.2.2 Cascading Style Sheets (CSS)

CSS este prescurtarea pentru Cascading Style Sheets și se referă la așa numitele foi de stiluri, după cum sunt recunoscute în limba română. Acestea servesc la stabilirea unui aspect unitar pentru elementele paginii web și facilitează modificarea acestora cu ușurință.

Stilurile se pot atașa elementelor din pagina .aspx corespunzătoare prin intermediul unor fișiere externe sau în cadrul documentului, prin elementul `<style>` și/sau atributul `style`.

4.2.2.1 Tipuri de stiluri

Stilurile interne

Acest tip presupune plasarea codului CSS în interiorul fiecărei pagini aspx pe care dorim să folosim stilurile respective, între tagurile `<head>` `</head>`.

Folosind această metodă (stilurile interne), fiecare fișier aspx va conține codul CSS folosit la stilizare. Asta înseamnă că atunci când dorim să realizăm o schimbare de stil (mărimea fontului, culoare, etc) va trebui să efectuăm modificarea în toate paginile ce conțin acel stil. Metoda descrisă este bună atunci când avem de stilizat un număr mic de pagini, însă când avem de a face cu websiteuri de zeci sau sute de pagini este destul de neplăcut să modificăm toate paginile.

Stilurile externe

Declararea stilurilor externe unui document aspx se face într-un fișier text care apoi se include în documentul care utilizează stilurile declarate în acel fișier. Un fișier CSS extern poate fi realizat cu orice editor simplu de text (Notepad, Wordpad, etc), cu editoare avansate sau în Visual Studio, care oferă facilități pentru realizarea rapidă și eficientă a acestor tipuri de fișiere. Fișierul CSS nu conține sub nici o formă cod html, ci doar cod CSS. Fișierul trebuie salvat cu extensia .css, în cazul în care este creat într-un editor text obisnuit și această operație nu este realizată în mod automat de către editorul respectiv[17].

Inserarea fișierului extern în paginile aspx se face prin plasarea unui link (legătură) în secțiunea `<head>` `</head>` a fiecărei pagini pe care dorim să folosim stilul respectiv.

Un exemplu de inserare a unui fișier extern .css într-o pagină aspx ar fi următorul:

```
<link rel="stylesheet" type="text/css" href=" fișierul.css" />
```

Pentru adăugarea foi de stil este necesară inserarea codului de deasupra în secțiunea `<head>` a paginii.

Folosind metoda fișierelor CSS externe, toate paginile aspx vor folosi același fișier de stil. Asta înseamnă că dacă dorim să facem o schimbare care să aibă efect în toate paginile, este de ajuns să modificăm un singur fișier, și anume cel de stil (.css), și efectul se va observa pe toate paginile aspx ce folosesc acel fișier.

Stilurile în linie

Stilurile în linie se definesc chiar în codul paginii aspx, în elementul pe care dorim să îl stilizăm.

Un exemplu relevant pentru acest tip este următorul:

```
<asp:Label ID="Label1" runat="server" Text="Label" style="color: red"></asp:Label>
```

Stilurile în linie nu permit schimbări rapide și facile pe mai multe fișiere în același timp, fiecare element necesitând atenția noastră pentru toate paginile.

Este posibilă folosirea simultană a tuturor celor 3 tipuri de stiluri, dar ordinea priorității este următoarea: prioritatea maximă o are stilul de linie, acesta fiind urmat de cel intern și doar la urmă este stilul extern. Însă cantitatea de cod necesă pentru aceste tipuri de stiluri este invers proporțională cu ordinea priorității lor .

4.2.3 Microsoft SQL Server

Caracteristica principală a aplicațiilor de baze de date constă în faptul că accentul este pus pe operațiile de memorare și retragere a informațiilor. Principala operație care apare în orice aplicație de baze de date este aceea de retragere a datelor, în scopul obținerii de informații din baza de date. Această este întrebuințarea și scopul tuturor bazelor de date. Alături de operațiile de retragere mai apar deasemenea și operații de memorare, folosite la introducerea de noi informații în baza de date, operații de ștergere și actualizare a unor date deja existente în baza de date. În cazul oricărei baze de date avem nevoie de un administrator al bazei de date (ABD) – o persoană, sau un grup de persoane cu atribuții bine stabilite relativ la organizarea și întreținerea bazei de date, care va realiza toate operațiile necesare asupra bazei de date, pentru menținerea unei stări optime a bazei de date în scopul funcționării ideale a sistemului ce folosește datele respective [3].

SQL Server 2008 este un server de baze de date implementat de Microsoft. Reprezintă un set de componente care lucrează împreună pentru nevoile de stocare a datelor și analiză ale site-urilor web sau ale sistemelor de procesare a datelor din companii.

Microsoft SQL Server 2008 este soluția Microsoft disponibilă pentru managementul și analiza datelor, care oferă securitate, scalabilitate și disponibilitate crescute pentru datele sistemele și aplicațiile analitice, făcând crearea, implementarea și managementul acestora mai facile.

Generat având la bază SQL Server 2000 și SQL Server 2005, SQL Server 2008 oferă o soluție integrată de management și analiză a datelor, permițând:

- Dezvoltarea, implementarea și administrarea de aplicații la nivel de companie mai sigure, scalabile și fiabile
- Maximizarea productivității IT prin reducerea complexității creării, implementării și administrării aplicațiilor pentru baze de date.
- Partajarea datelor pe mai multe platforme, aplicații și dispozitive pentru a facilita conectarea sistemelor interne și externe.
- Controlarea costurilor fără a sacrifica performanța, disponibilitatea, scalabilitatea sau securitatea.

4.2.3.1 Principale instrucțiuni SQL

Comenzile folosite pentru definirea datelor sunt : create, alter și drop, fiind utile pentru crearea, utilizarea și ștergerea oricărei structuri de date, îndeosebi tabele. O instrucțiune SQL este formată din cuvinte rezervate și cuvinte definite de utilizator. Cuvintele rezervate constituie partea fixă și se scriu exact cum este necesar. Cuvintele definite de utilizator reprezintă denumirile diverselor obiecte din BD.

Comenzile pentru manipularea datelor sunt: select, insert, update și delete fiind utilizate pentru selectarea, introducerea, actualizarea, respectiv ștergerea anumitor înregistrări din tabele.

5. Proiectare de detaliu și implementare

În acest capitol vor fi detaliate funcționalitățile sistemului reprezentate sub forma cazurilor de utilizare și sunt descrise detaliile de implementare ale componentelor specifice sistemului.

5.1 Modelul cazurilor de utilizare

Cazurile de utilizare descriu cum o categorie specifică numită actor va interacționa cu aplicația pentru a îndeplini o acțiune specifică. Acțiunea poate fi una simplă precum dobândirea unui conținut definit sau complexă care poate să necesite ghid de utilizare. Cazurile de utilizare descriu interacțiunea din punctul de vedere al utilizatorului.

Deși construirea și analiza lor necesită timp cazurile de utilizare:

- Ajută dezvoltatorii să înțeleagă cum percep utilizatorii interacțiunea lor cu aplicația web;
- Furnizează detaliile necesare pentru a crea o analiză efectivă a modelului;
- Sunt partea pe care se bazează crearea testelor pentru sistemul implementat;
- Furnizează un ghid important pentru cei care trebuie să testeze aplicația.

5.1.1 Actorii sistemului

Actorii sunt orice entitate care comunică cu sistemul sau produse ale sistemului care sunt externe față de aplicație. Fiecare actor are unul sau mai multe motive pentru care utilizează aplicația.

Actorii sunt diferite persoane sau sisteme care folosesc sistemul sau produse din sistem. Ei sunt necesari la descrierea comportamentului sistemului, reprezentând rolul pe care oamenii îl joacă în funcționarea sistemului.

Sistemul de față are trei actori principali:

- **utilizatorii autentificați**, cei care se loghează în sistem și folosesc funcționalitățile pentru care a fost creată aplicația web;
- **administratorul**, utilizatorul care are dreptul de a șterge din utilizatorii înregistrați în cazul în care este ceva în neregulă cu ei și are grijă ca sistemul să fie funcțional;
- **utilizatorii neînregistrați** care au posibilitatea de a deveni utilizatori înregistrați.

Utilizatorii autentificați site-ului sunt actorii principali, aceștia se bucură de cele mai multe funcționalități. Acestia își creează profilul și în funcție de acesta pot fi găsiți de alți utilizatori, pot căuta utilizatori și cel mai important lucru este că pot să socializeze cu oricare alt utilizator care este înregistrat pe acest site.

Utilizatorii neînregistrați nu vor putea utiliza nici una din funcționalitățile sistemului dar se pot înregistra pe site astfel ei se vor putea autentifica, după care să beneficieze de funcționalitățile sistemului.

Administratorul este cel responsabil pentru menținerea datelor din baza de date într-o stare consistentă astfel încât sistemul să fie perfect funcțional în orice moment. În acest sens el poate efectua diferite operații de ștergere de clienți nedorți, eliminarea unor comentarii, care nu respectă limitele unui limbaj minimal necesar în societate, sau vizualizarea feedback-

ului trimis de utilizatorii site-ului pentru a îmbunătăți aplicația și să facă astfel încât să fie mai plăcută de utilizatori.

5.1.2 Cazurile de utilizare

5.1.2.1 Utilizatori autentificați:

Cazurile de utilizare sunt inițiate de către actori. Aceștia au un scop bine definit, iar cazul de finalizare este cu succes în momentul în care scopul respectiv este satisfăcut.

Cazul principal de utilizare constă în efectuarea următoarelor acțiuni de către utilizator:

- Autentificare;
- Vizualizare profil propriu și al altor utilizatori;
- Modificarea informațiilor profilului propriu;
- Vizualizarea celor mai recente evenimente ale prietenilor;
- Publicare eveniment nou;
- Vizualizarea albumelor de fotografii ale altor utilizatori, adăugare comentarii;
- Schimbarea pozei de profil și ale copertelor de albume;
- Căutare utilizatori în sistem;
- Trimitere cerere de prietenie;
- Vizualizare/acceptare/ștergere cereri de prietenie;
- Folosirea camerei de chat;
- Adăugare/ștergere (la informațiile proprii) comentarii;
- Adăugare fotografii și albume;
- Adăugare/ștergere/modificare blog;
- Adăugare/ștergere prieteni;
- Trimitere mesaje altor utilizatori sau administratorului;
- Vizualizarea/ștergerea/răspunde mesajelor primite;
- Recuperare parolă uitată (primirea prin email);
- Contactare administrator;
- Dezactivare cont;
- Delogare.

O condiție importantă pentru ca utilizatorii să parcurgă aceste cazuri de utilizare este ca aceștia să se logheze la sistem. În cazul în care utilizatorii încă nu sunt înregistrați, aceștia au opțiunea de a se înregistra.

După ce utilizatorul își încheie activitatea de socializare pe site acesta poate să se delogheze pentru siguranța datelor din cont.

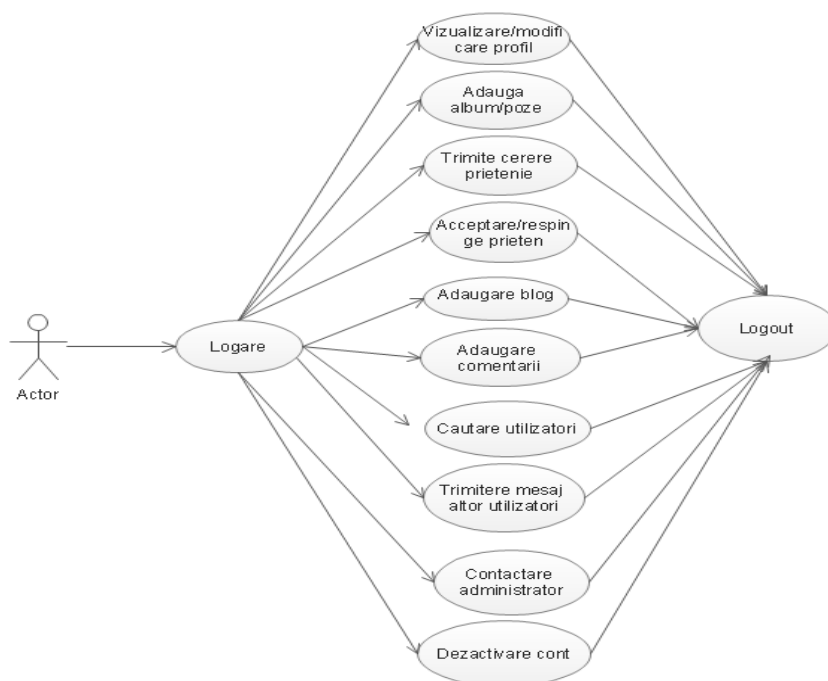


Figura 5.1 Diagrama caz de utilizare pentru utilizator înregistrat

În continuare vor fi detaliate câteva cazuri de utilizare ale utilizatorului înregistrat și administratorului

Logare utilizator

Caz de utilizare: logare utilizator

Actorul principal: utilizatorul autentificat

Obiectivul cazului de utilizare: autentificarea în sistem pentru ca utilizatorii să folosească funcționalitățile site-ului

Precondiții: utilizatorul trebuie să acceseze site-ul și să fie înregistrat în baza de date a sistemului

Declanșator: utilizatorul dorește să intre pe propria pagină.

Scenariu:

1. Utilizatorul intră pe pagina de început a site-ului;
2. Alege opțiunea de a se loga;
3. Introducerea username-ului și a parolei;
4. Intră în sistem.

Excepții:

1. Utilizatorul introduce un nume de utilizator sau parolă greșită. În cazul în care utilizatorul a uitat parola poate să o recupereze.
2. Profilul utilizatorului a fost șters de către administrator din cauza unor nereguli.

Prioritate: este obligatoriu, utilizatorii nu pot să beneficieze de nici o funcționalitate a site-ului fără să fie logat.

Disponibilitate: de fiecare dată când este accesată pagina de login.

Frecvență de utilizare: la fiecare accesare a sistemului.

Probleme existente: dacă un alt utilizator cu intenții malițioase cunoaște parola utilizatorului.

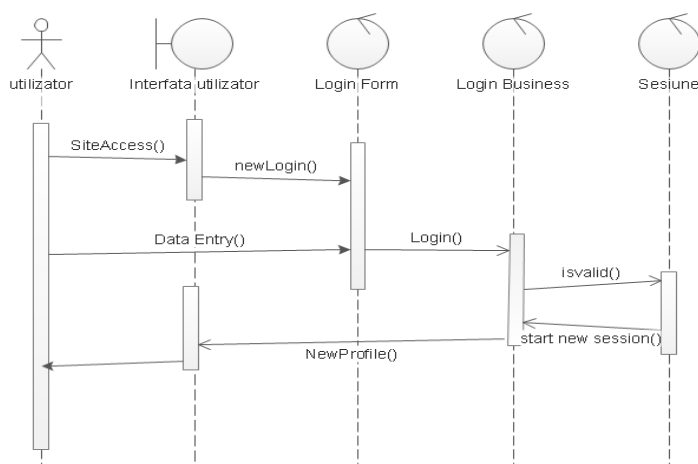


Figura 5.2 Diagrama de secvență pentru logare

Încărcare poză la album

Caz de utilizare: încărcare fotografie

Actorul principal: utilizatorul autentificat

Obiectivul cazului de utilizare: adăugarea unei noi poze la profilul utilizatorilor pentru a fi vizualizată de ceilalți utilizatori.

Precondiții: utilizatorul trebuie să fie autentificat

Declanșator: utilizatorul decide să adauge poză

Scenariu:

1. Utilizatorul intră pe pagina de fotografii;
2. Încarcă o poză din propriul calculator;
3. Pune un nume pozei care a ales-o;

4. Alege albumul în care utilizatorul dorește să adauge poza;
5. Salvează poza.

Excepții:

1. În cazul în care sesiunea utilizatorului a expirat utilizatorul este redirecționat la pagina de login;
2. Dacă este prima poză pe care utilizatorul dorește să o adauge profilului atunci acesta trebuie să creeze prima dată un album nou;
3. Dacă utilizatorul dorește să adauge poza într-un album nou, acesta trebuie să creeze prima dată albumul.

Prioritate: este facultativ, utilizatorii pot să socializeze și fără poze la profil.

Disponibilitate: de la prima logare în sistem utilizatorul poate să adauge poze.

Frecvență de utilizare: de fiecare dată când un utilizator dorește să adauge poză

Probleme existente:

1. Dacă un utilizator uită să se delogheze alți utilizatori care au acces la calculatorul utilizatorului pot să adauge poze la profilul lui fără ca el să știe.
2. Nu se poate adăuga o poză fără să se aleagă un album care să o conțină.

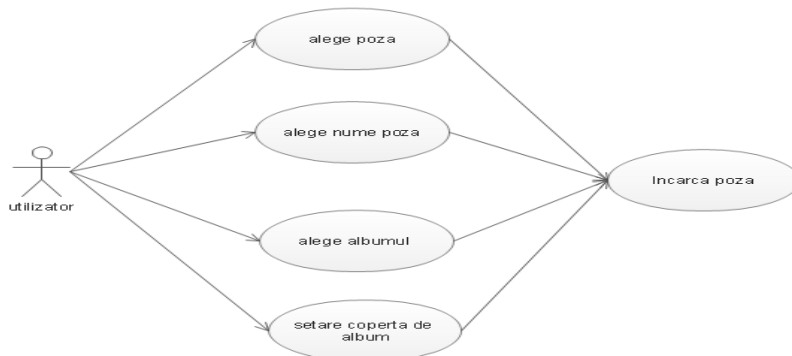


Figura5.3 Diagrama caz de utilizare pentru adăugare fotografie

Recuperare parolă uitată

Caz de utilizare: recuperare parolă

Actorul principal: utilizatorul neautentificat

Obiectivul cazului de utilizare: recuperarea parolei și autentificarea în site

Precondiții: utilizatorul trebuie să acceseze site-ul

Declanșator: utilizatorul dorește să intre pe site dar nu își mai știe parola.

Scenariu:

1. Utilizatorul intră pe pagina de start a site-ului;

2. Accesează link-ul de recuperare a parolei;
3. Introduce username-ul alege opțiunea de trimitere parolă pe email;
4. Verifică emailul unde a primit o parolă nouă aleatoare;
5. Intra în cont cu parola primită pe email și își schimbă parola.

Prioritate: este obligatoriu, utilizatorii își uită parolele.

Disponibilitate: tot timpul.

Frecvență de utilizare: de fiecare dată când un utilizator își uită parola.

Probleme existente:

1. Dacă utilizatorul și-a uitat și adresa de email și username-ul, acesta nu mai poate intra pe site.

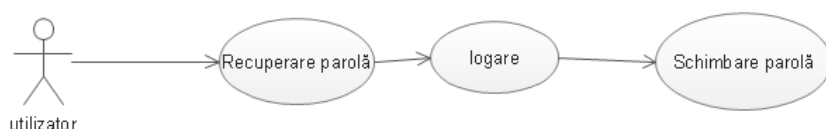


Figura 5.4 Diagrama caz de utilizare schimbare parolă

5.1.2.2 Administratorii siteului

Administratorul este actorul care are grijă ca datele din baza de date să se afle într-o stare optimă, de a monitoriza activitatea utilizatorilor și de a primi feedback de la aceștia.

Astfel administratorul realizează următoarele operații autentificându-se la rândul lui și el, în sistem:

- Autentificare
- Căutare/vizualizare profil utilizatori;
- Să vizualizeze evenimentele utilizatorilor din baza de date;
- Să trimită și să primească mesaje;
- Ștergere utilizatorii care au fost raportați de alți utilizatori din motive întemeiate;
- Adăugarea de noi administratori;
- Citirea mesajelor trimise de către utilizatori către administrator.
- Delogare.

Mai în jos este descris cazul de utilizare pentru ștergerea unui utilizator din sistem.

Ștergere utilizator vizat

Caz de utilizare: ștergere utilizator vizat;

Actorul principal: administrator sistem;

Obiectivul cazului de utilizare: înlăturarea utilizatorilor cu intenții malițioase.

Precondiții: administratorul trebuie să fie logat și să existe utilizatori raportați;

Declanșator: administratorul dorește să elimine utilizatorii care crează probleme;

Scenariu:

1. Administratorul se autentifică;
2. Vizualizează tabelul cu utilizatorii raportați și motivele;
3. Șterge utilizatorul vizat;

Prioritate: este obligatoriu, există utilizatori care pot crea probleme;

Disponibilitate: tot timpul;

Frecvență de utilizare: de fiecare dată când un administrator intră în site;

Probleme existente:

1. Dacă utilizatorii care au un comportament neadecvat pe site nu sunt raportați.

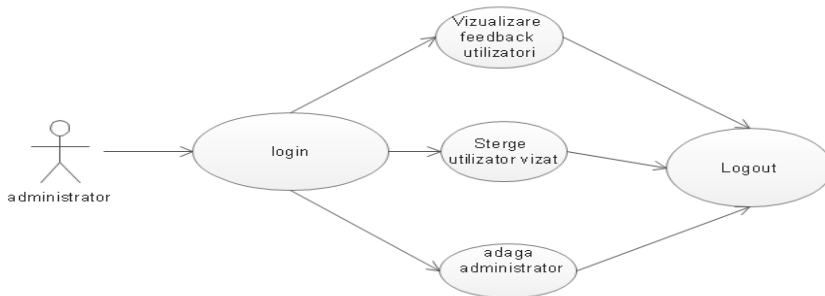


Figura 5.5 Diagrama caz de utilizare administrator

5.2 Detalii de implementare

Pe lângă structurarea proiectului pe nivele l-am structurat și în funcție de componentele sistemului. Această structurare oferă o modularizare mai bună și permite scalarea sistemului. Modulele în care este împărțit proiectul sunt în funcție de tablele din baza de date cu care trebuie să interacționeze sistemul și în funcție de funcționalități. Astfel,

pentru operațiile administratorului se folosește un anumit pachet din proiect: pentru utilizatori (înregistrare, creare/modificare profil, autentificare, schimbare parolă etc.), pentru partea de socializare (obținere listă prieteni, cerere prietenie, acceptare/respingere prietenie, ștergere prieten etc.) pentru partea de fotografii (adaugă/șterge poză, afișare poze, schimbare poză), pentru partea care se ocupă de trimiterea mesaje între prieteni și administrator, pentru partea de blog, chat și noutăți.

Un modul este un grup de caracteristici și funcționalități ale aplicației. Modulele sunt un mod de organizare a unei aplicații complexe pentru a o împărți în părți mai mici și mai ușor de înțeles. În cazul de față, modulele vor fi conturi și profile, galerii foto, blog, socializare și mesaje . Aceste module, pe lângă faptul că țin funcționalitățile decuplate, sunt utile și în cazul în care se dorește scalarea sistemului.

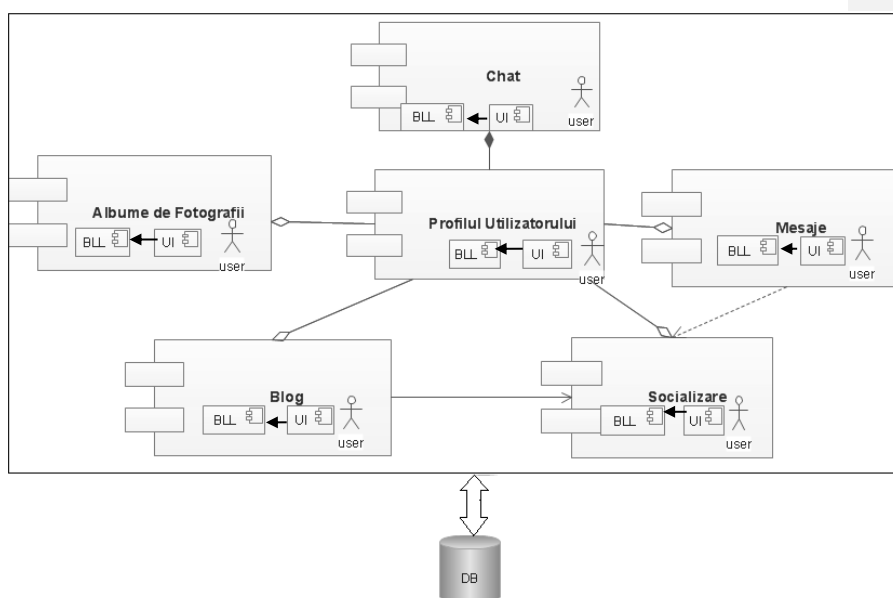


Figura 5.6 Arhitectura generală sistemului

În secțiunea următoare vor fi descrise modulele sistemului, modul în care au fost implementate și interacțiunile între ele.

5.2.1 Componentele principale ale sistemului

5.2.1.1 Profilul și contul utilizatorilor

Profilele și conturile se diferențiază prin faptul că profilul este partea în care utilizatorul își vizualizează informațiile care le-a adăugat și poza de profil sau pot să vizualizeze profilul altor utilizatori ai sistemului dacă sunt prieteni. Dacă un utilizator dorește să vadă un profil al unui utilizator care nu îi este prieten poate să îl vadă numai dacă acesta îi

permite. Contul unui utilizator reprezintă informațiile lui pe care acesta și le poate modifica dacă dorește. De asemenea, în secțiunea *My Account* utilizatorul poate să își schimbe parola sau să își dezactiveze contul. Profilul utilizatorului va permite utilizatorilor să publice detalii despre ei. Profilul trebuie să fie un modul flexibil pentru a putea fi extins ușor.

Înregistrarea

Operația de înregistrare este singura operație pe care un utilizator neînregistrat poate să o efectueze.

Pentru înregistrare utilizatorul trebuie să își introducă datele personale în TextBox-urile din pagina web. Informațiile ca sex-ul utilizatorului și data nașterii nu sunt obligatorii, în schimb numele, prenumele, username-ul, parola, adresa de email și TextBox-ul pentru caracterele captcha sunt obligatorii. Pentru realizarea înregistrării din clasa de code Behind a web form-ului Login se apelează metoda `register()` din clasa de logică `personsBusiness.cs` care va crea un obiect de tipul utilizator și va transmite mai departe spre `DataContext` care, cu ajutorul metodelor `InsertOnSubmit(newlog)` și `dbcontext.SubmitChanges()`, va realiza înregistrarea în tabelele `logare_utilizator` și `utilizator` a noului utilizator;

Captcha este un standard pentru teste automate care diferențiază computerele de oameni folosind un test generat ce poate fi ușor trecut de persoane, dar nu poate fi trecut de programe de calculator. Acest tip de test este des întâlnit la înscrierea pe diverse site-uri. Test-ul include o imagine a unui text distorsionat pe care utilizatorul trebuie să îl introducă corect pentru a putea să realizeze înregistrarea.

Scopul principal este de a preveni utilizatorii care folosesc site-ul pentru publicitate de la utilizare cu scopul de a completa automat datele cerute la înregistrare și a utiliza conturile cu acest motiv. Textul din imagine este distorsionat pentru a preveni utilizarea softului OCR (optical character reader) să citească aceste caractere automat.

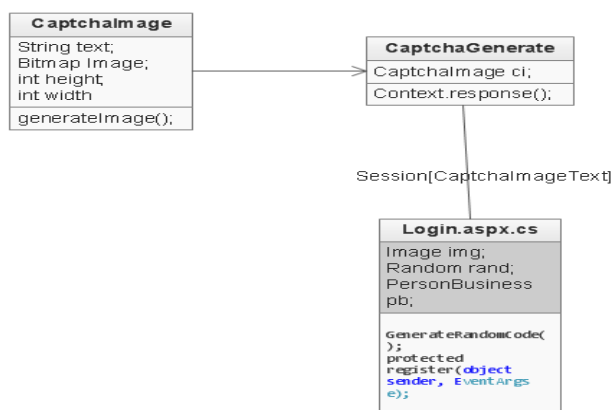


Figura 5.7 Diagrama de clase pentru generarea imaginii Captcha

Clasele care participă la generarea caracterelor captcha sunt:

- *CaptchaImage.cs* – definește obiectele *CaptchaImage* care crează imaginile.
- *CaptchaGenerate.ashx* – un fișier web handler (fișier cu un pas în spatele browserului unde codul de html și C# pot fi amestecate) proiectat pentru a returna imaginea JPEG.
- *Login.aspx*, *Login.aspx.cs* - web form unde este afișată imaginea.

CaptchaImage.cs

Această clasă crează o imagine cu anumiți parametri care va reprezenta textul care urmează să fie afișat. Metoda principală a acestei clase este metoda *generateImage()* care crează o imagine bitmap cu înălțimea și lățimea specificate. Această metodă este apelată în constructorul *CaptchaImage*, ceea ce permite ca obiectul să fie utilizabil imediat ce a fost creată o instanță a acestei clase.

Pentru a crea imaginea, inițial am umplut background-ul folosind un fundal hașurat pentru a crea impresia de imagine "murdară". La afișarea textului începem cu inițializarea mărimii fontului în funcție de înălțimea imaginii folosind metoda *Graphics.MeasureString()*. În cazul în care textul este mai mare decât înălțimea imaginii atunci înălțimea fontului va fi micșorată.

```
Do
{
    fontSize--;
    font = new Font(this.familyName, fontSize, FontStyle.Bold);
    size = g.MeasureString(this.text, font);
}
while (size.Width > rect.Width);
```

După ce font-ul textului a fost setat cu ajutorul metodei *GraphicsPath()* se transformă textul într-un set de linii și curbe, care ulterior sunt distorsionate cu ajutorul metodei *GraphicsPath.Wrap()*. Cu câteva valori generate aleator textul va fi distorsionat. Pentru ca imaginea să fie și mai modificată vor fi printate aleator mici pete peste imagine, astfel adăugându-se zgomot.

```

int m = Math.Max(rect.Width, rect.Height);
for (int i = 0; i < (int) (rect.Width * rect.Height / 30F); i++)
{
    int x = this.random.Next(rect.Width);
    int y = this.random.Next(rect.Height);
    int w = this.random.Next(m / 50);
    int h = this.random.Next(m / 50);
    g.FillEllipse(hatchBrush, x, y, w, h);
}

```

Login.aspx

Este pagina care efectuează autentificarea utilizatorului, redirecționează spre o pagină pentru recuperarea parolei și efectuează operația de înregistrare, unde este folosită imaginea captcha. Această pagină conține în fișierul .aspx un element pentru a putea afișa imaginea. Aceasta va conține un atribut SRC care va pointa spre fișierul *CaptchaGenerate.ashx*.

În fișierul login.aspx există metoda generateRandomCode() care generează un text string care va fi folosit în crearea imaginii. Validarea imaginii se face când utilizatorul apasă butonul register caz în care este comparat textul introdus de utilizator cu obiectul sesiune unde a fost setată sesiunea la textul generat. Motivul pentru care textul este stocat în sesiune este că trebuie să fie accesat de fișierul *GenerateCaptchaImage.ashx*.

GenerateCaptchaImage.ashx

Pentru această clasă am folosit un fișier web de manipulare, deoarece nu este nevoie de cod HTML, doar trebuie generată imaginea. Se crează un obiect CaprchaImage folosind textul din Session salvat în Login.aspx, acest obiect este necesar pentru crearea imaginii bitmap. Se va seta tipul de răspuns HTTP la "image/jpeg" pentru că clientul (browserul) să știe că va primi o imagine. Ultimul pas al afișării imaginii este extragerea imaginii bitmap din CaptchaImage. Image și introducerea ei în răspunsul HTTP în format JPEG.

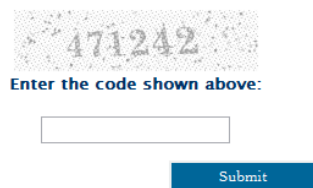


Figura 5.8 Caractere captcha

Dacă utilizatorul a completat corect câmpurile cu o adresă de email unică, cu username-ul unic și cu textbox-ul de caractere captcha completat corect, utilizatorul va fi

întregirat în baza de date și va primi un email care îl anunța că a fost întregirat în care va primi username-ul cu care se va putea loga.

Autentificare

Îmediat după ce utilizatorul s-a înregistrat este redirecționat spre pagina de login. Unde utilizatorul trebuie să își introducă username-ul și parola. Username-ul și parola sunt verificate cu metoda `public string autentificare(string username, string parola)` din clasa `personBusiness`. Această metodă returnează un string reprezentând tipul utilizatorului sau null, dacă datele nu se găsesc în baza de date. În funcție de rezultat, utilizatorul este redirecționat spre pagina corespunzătoare. La autentificare este construit un obiect session care va reține username-ul utilizatorului. Acest obiect este necesar deoarece HTTP este un protocol stateless și nu menține starea. Cu ajutorul session username-ul utilizatorului va fi disponibil în fiecare pagină a web site-ului. Obiectul session expiră după 20 de minute în care utilizatorul nu a avut activitate.

Modificarea informațiilor din profil

Pentru a modifica informațiile din profil, utilizatorul trebuie să acceseze pagina *My account.aspx*. În această pagină utilizatorul va putea modifica orice informație pe care o are în cont. De asemenea, tot din această pagină poate fi redirecționat astfel încât să își poată schimba parola sau să își dezactiveze contul. Clasele care sunt folosite în această parte sunt *MyAccount.aspx* dacă utilizatorul dorește să își închidă contul sau să își schimbe parola este redirecționat la pagina *Desactivate.aspx*, clasa din nivelul de logică folosită este `personBusiness.cs`, iar pentru interacțiunea cu baza de date este folosită clasa *DataContext.dbml*.

Dacă utilizatorul dorește să își schimbe parola, acesta trebuie să introducă prima dată parola curentă. Verificarea acestei parole se face în clasa `personBusiness` cu metoda `chackpassword (string password, string id_ut)`. Dacă parola este introdusă corect, utilizatorului îi va fi permis să schimbe parola. Când utilizatorul își introduce noua parolă, va putea vedea nivelul de securitate pe care i-l oferă parola introdusă. Aceasta se realizează cu controlul `PasswordStrength` din librăria `AjaxControlToolkit.dll` a tehnologiei ASP.NET

Pentru a folosi oricare dintre aceste controale ale AJAX este nevoie de introducerea unui `ScriptManager` în partea de markup.

`ScriptManager` asigură că fișierele ASP.NET sunt incluse și că suportul AJAX este adăugat. Pentru securitate utilizatorul este obligat să aibă o parolă de cel puțin 5 caractere. Pentru parolele mai mari de 5 caractere acest control are rolul doar de informare a securității va fi oferită de parolă. Pentru schimbarea parolei este apelată metoda `changepassword` din `personBusiness` care are ca parametri id-ul utilizatorului și noua parolă a utilizatorului.

În cazul care utilizatorul va dori să își dezactiveze contul trebuie să își introducă parola astfel verificându-se că utilizatorul care execută această operație este deținătorul contului. Înainte să-i fie dezactivat contul, utilizatorului îi va apărea un pop-un prin care trebuie să confirme că aceasta este operația pe care dorește să o facă, deoarece, o dată cu dezactivarea contului, utilizatorul își va pierde toate informațiile de la lista de prieteni până la

poze, blog comentarii ș.a. Pentru ca acest pop-up să apară, butonul de dezactivare va avea atașat un control AJAX numit ConfirmButton:

```
<asp:ConfirmButtonExtender ID="ConfirmButtonExtender1"
ConfirmOnFormSubmit="true" ConfirmText="Are you sure you want to
deactivate your account?" TargetControlID= "Button1"
runat="server" />
```

După ce utilizatorul va confirma că dorește să își dezactiveze contul, în clasa de `businessPerson` se va apela metoda `dbcontext.utilizatoris.DeleteOnSubmit(utiliz)`; care va realiza ștergerea în cascadă din baza de date a tuturor câmpurilor care au fost introduse sau au legătură cu acest utilizator.

Recuperarea parolei uitate

Una dintre funcționalitățile care pot fi realizate de către utilizatorii neautentificați este recuperarea parolei. Recuperarea parolei se realizează cu ajutorul numelui de utilizator sau a adresei de email introduse la înregistrare. Clasele care interacționează pentru realizarea acestei funcționalități sunt *ForgotPassword.aspx*, *personBusiness.cs*, *sendEmail.cs* și *DataContext.dbml*. Pentru aș-i recupera parola, utilizatorul va primi pe adresa de email noua parolă. Trimiterea de email se face în clasa *sendEmail.cs* cu metoda `public string sendmail (string username, string email)`. În această metodă sunt apelate metodele din *personBusiness*, care extrage adresa de email din baza de date și care generează o parolă nouă aleatoare. Pentru trimiterea emailului se adaugă la clasa *sendEmail* directiva *System.Net.Mail*. Se crează câte un obiect *SmtpClient* și *MailMessage*.

Emailul va fi trimis după ce sunt verificate `username`-ul sau adresa de email introduse de utilizator și după ce va fi generată parola. Parolele random cresc securitatea site-ului deoarece generează o parolă care nu se va putea ghici. Utilizatorul are opțiunea să își păstreze această parolă, sau să o schimbe ulterior.

Metoda de generare a parolei este implementată ajutorul `System.Guid.NewGuid()`. Cu această metodă se poate genera un set de caractere aleator, prin combinarea cifrelor, caracterelor speciale și a literelor din alfabet. Un obiect "Guid" este un set de 16 caractere, care poate fi folosit la toate calculatoarele și rețelele dacă este cerut un identificator unic. Un asemenea identificator este foarte greu de generat identic cu altele anterioare.

Deoarece acest obiect are 16 caractere, este prea mult pentru a fi generat ca și parolă și se vor păstra doar 8 dintre caractere. Pe lângă faptul că șirul va fi scurtat, îi vor fi șterse și caracterele "-" care sunt generate inițial pentru separarea grupurilor de câte patru caractere.

```
public string GetRandomPasswordUsingGUID(int length)
{
    string guidResult = System.Guid.NewGuid().ToString();
    guidResult = guidResult.Replace("-", string.Empty);
    if (length <= 0 || length > guidResult.Length)
        return guidResult.Substring(0, length);
}
```


Tablele din baza de date care interacționează cu această componentă sunt: utilizatori, logare_utilizatori, detalii_utilizatori și utilizatori_raportați. În tabela detalii_utilizatori sunt incluse datele utilizatorilor care sunt de tipul varchar(max), deoarece de aceste câmpuri nu este nevoie tot timpul și sunt extrase numai dacă sunt folosite.

Câmpurile din tabela **utilizatori** sunt:

- id_utilizator care este cheie primară de tipul int, iar prin specificarea opțiunii identity aceasta se generează automat;
- username de tipul varchar(70) trebuie să fie unic și este necesar la logarea în sistem și trebuie să fie diferită de null;
- Nume și prenume sunt de tipul varchar(50);
- Adresa de email este de tipul varchar(50), este unică și diferită de null;
- Sex-ul este de tipul varchar(8) și trebuie să ia una din valorile: feminin sau masculin;
- Data_nașterii este de tipul DateTime;
- Orașul actual și orașul natal este de tipul varchar(50);
- Univeristatea și facultatea sunt de tipul varchar(50) și sunt permise și valori null;
- Photo este de tipul Image și conține poza de profil a utilizatorului;
- Anul admiterii, anul terminării și data_inregistrării sunt de tip DateTime;
- Județul și țara sunt de tipul varchar(50) și nu este obligatorie completarea lor.

Tabelul **detalii_utilizator** conține informații suplimentare din profilul fiecărui user, iar câmpurile tabelului sunt:

- Id_detalii_utilizatori reprezintă cheia primară a tabelului este de tipul int și este completată automat;
- Id_utilizator este cheia străină care are referință la tabelul utilizatori;
- Despre_mine este de tipul varchar(max) și poate fi detalii suplimentare care utilizatorul dorește să le introducă la profil;
- Hobby și sporturi este de tipul varchar(max);

Tabelul **logare_utilizator** conține informații care sunt necesare la logare:

- Id_logare_utilizator este cheia primară a tabelului;
- Id_utilizator este cheia străină și este referință spre tabelul utilizatori;
- Tip_utilizator este de tipul varchar(10) poate fi "simple user" sau "admin" și semnifică rolul utilizatorului;
- Data_inregistrare este de tipul DateTime și reprezintă data la care utilizatorul și-a făcut cont pe site;
- Câmpurile întrebare și răspuns de tipul varchar(70) ajută la recuperarea parolei în cazul în care aceasta a fost uitată;
- Câmpul parolă este de tipul varchar(50), fiind parola necesară la logare.

Tabelul **utilizator_vizat** conține lista utilizatorilor care au fost raportați și are câmpurile:

- Id_utilizator_vizat este cheia primară a tabelului și este de tipul int identity;

- Id_utilizator este cheia străină care este referință spre utilizatorul care a raportat;
- Username reprezintă numele de utilizator al utilizatorului raportat;
- Data_raportare este de tipul DateTime.

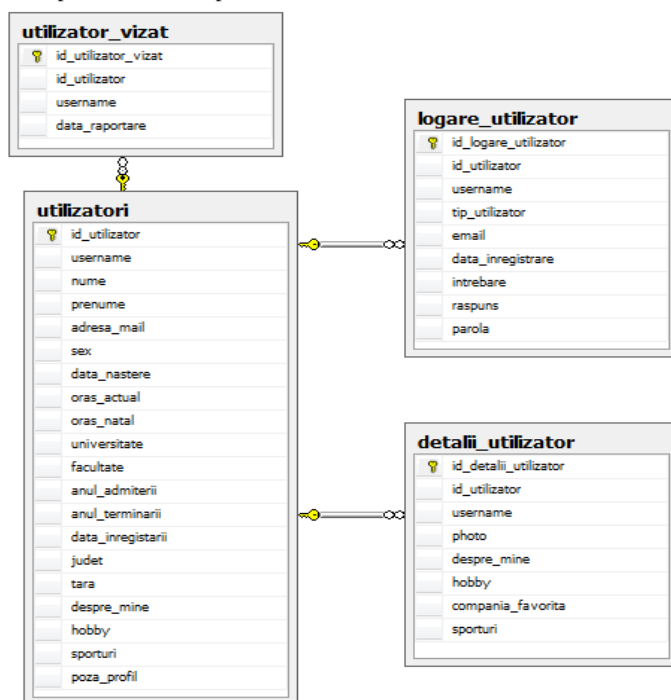


Figura 5.9 Diagrama din baza de date a utilizatorilor

5.2.1.2 Socializare

Socializarea se referă la posibilitatea utilizatorilor sistemului de a interacționa și comunica între ei. Conceptul de a avea prieteni într-o comunitate este modul prin care site-ul face ca utilizatorii să se întoarcă. Un prieten al utilizatorului este un alt utilizator al sistemului căruia utilizatorul îi va permite să vadă toate informațiile profilului și de care este interesat să îi vadă informațiile și modificările pe care le face în profil. Clasele principale care formează această componentă sunt: *socializationDataAccess.dbml*, *socializationBusiness.cs*, *NewsBusiness.cs*, *viewFriends.aspx*, *viewFriendsRequest.aspx*, *SearchEngine.aspx* și *RecentUpdate.aspx*.

Căutarea utilizatorilor

Pentru ca utilizatorii să poată interacționa între ei, aceștia trebuie să se găsească, folosind pagina de căutare a site-ului. Căutarea se poate face cu metoda *searchUsers* (string

firstname, string lastname, string username, string email, string university, string college, string country, string city), care returnează o listă de utilizatori în funcție de principale informații din profilul utilizatorului, cum ar fi nume sau prenume, numele de utilizator, anul nașterii, universitatea sau facultatea la care studiază sau au studiat, adresa de email, sau orașul în care trăiesc care au fost introduse la căutare. Căutarea se va face doar în funcție de câmpurile care au fost completate de către utilizatorul care dorește să realizeze căutarea. Această căutare a fost realizată folosind metoda: contains (string secvența) din clasa Enumerable a spațiului de nume System.Linq. Această metodă determină dacă o secvența din baza de date conține un element specific transmis ca parametru.

Lista de prieteni va fi afișată într-un repetor care va conține o mică poză a utilizatorului, numele și prenumele acestuia.

Poza va fi o legătura spre pagina de profil a utilizatorilor găsiți la căutare, unde, în cazul în care cei doi utilizatori nu sunt prieteni, sau unul dintre ei așteaptă răspuns la cererea de prietenie, există opțiunea ca acesta să îi ceară să fie prieteni.

Acceptare/ respingere prietenii

Pentru ca doi utilizatori să fie prieteni, unul dintre ei trebuie să îl invite pe celălalt să fie prieteni, iar celălalt să accepte. Când un utilizator îi cere altuia să fie prieteni, se va introduce în baza de date un rând cu cele două numele de utilizatori și data trimiterii cererii de prietenie. Utilizatorul care a primit cererea de prietenie poate să vadă lista de cereri de prietenie în pagina viewFriendsRequest, iar de aici, acesta poate să respingă cererea de prietenie sau să o accepte. După ce utilizatorul acceptată prietenia, cererea va fi ștearsă din baza de date și utilizatorul va fi introdus în lista de prieteni. Dacă cererea a fost respinsă cererea va fi ștearsă din tabel. Clasa de logică care se ocupă de aceste operații este SocializationBusiness.cs.

După ce o cerere de prietenie a fost acceptată, cei doi utilizatori vor fi adăugați în lista de prieteni, după care acești utilizatori vor avea acces reciproc la informațiile din profil. Prin partea de noutăți cei doi utilizatori vor avea o pagină recentUpdates care le va permite să își vizioneze ultimile modificări în profil și mesaje. În această pagină utilizatorii vor putea să introducă și mesaje care să le apară celorlalți prieteni. Lista de noutăți va fi afișată într-un repetor care este imbricat într-un UpdatePanel <asp:UpdatePanel ID="UpdatePanel1" runat="server">. Acesta este un element din biblioteca AjaxControlToolkit.dll care permite ca secțiuni din pagină să fie schimbate la modificările în baza de date fără, să se execute reîncărcarea, oferind impresia unei pagini mai rapide și mai dinamice.

Tabelele fac parte din această componentă sunt: utilizatori, cerere_prietenie, prieteni, și updates.

Tabela cerere prietenie are următoarele câmpuri:

- Id_cerere_prietenie este cheia primară fiind de tipul int identity și este diferită de null;
- Delauserid este id-ul userului care a trimis cererea de prietenie și face referința la câmpul id_utilizator din toabelul utilizatori;
- Pentruuserid este id-ul utilizatorului care a primit cererea de prietenie;

- Acceptat este de tipul varchar(2) poate fi da sau nu și indică dacă s-a răspuns cererii de prietenie;
- Data_timiterii este de tipul DateTime.

Tabelul prieteni are cheia primară id_prieteni, cheia străină id_utilizator care face referință spre id-ul prietenului care a trimis cererea de prietenie și numele de utilizator al prietenului.

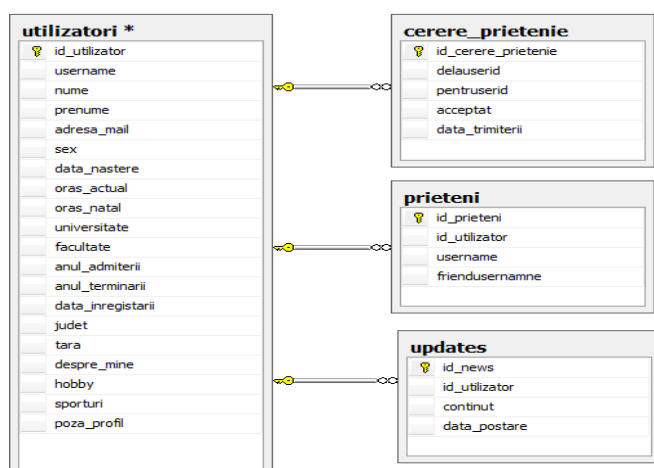


Figura 5.10 Diagrama de bază de date a componentei socializare

5.2.1.3 Albume de fotografii

Pentru implementarea albumelor de fotografii clasele utilizate în ordinea nivelului arhitectural din care fac parte sunt *photoAlbumDataAccess.dbml*, *photoBusiness.cs*, *GenerateImage.ashx*, *Albums.aspx*, *viewPhoto.aspx*.

În pagina de album utilizatorii autentificați au posibilitatea de a adăuga fotografii noi la albumele existente sau în albume nou create tot în această pagină. Adăugarea fotografiilor este realizată cu ajutorul unui control `<asp:FileUpload ID="FileUpload1" runat="server" />`. După ce utilizatorul va alege poza, aceasta va fi transformată în tipul byte ca să poată fi stocată în baza de date. La adăugarea fotografiei, utilizatorul trebuie să aleagă albumul în care va fi adăugată poza, numele pozei, care poate fi și null, și dacă utilizatorul dorește ca poza să fie fotografia de copertă a albumului în care a fost adăugată. În funcție de acestea se va insera un nou rând cu aceste câmpuri.

Albumele de fotografii sunt afișate într-un repetor care va conține numele albumului și coperta lui. Coperta albumului este micșorată în imagine cu înălțimea și lățimea de 150 px. Pozele albumelor sunt generate de fișierul Thumbnail.ashx care are ca parametru id-ul pozei și este URL-ul imaginii din repetor.

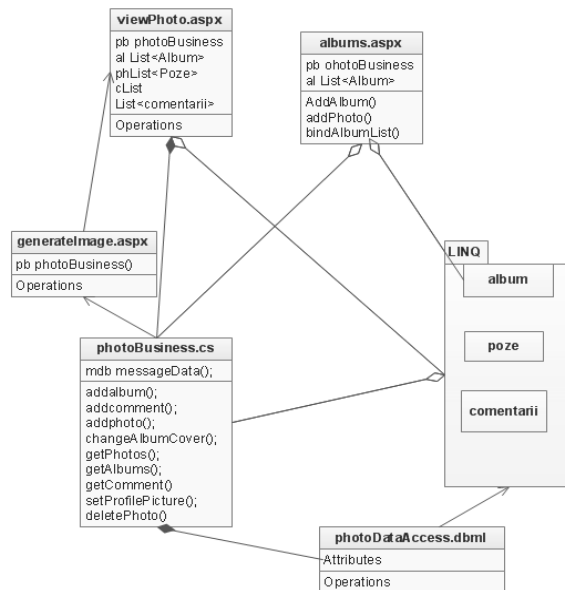


Figura 5.11 Diagrama de clase pentru componenta albume de fotografii

Tabelele din baza de date care alcătuiesc această componentă sunt: poze, album, comentarii și utilizatori.

Tabelul **poze** conține următoarele câmpuri:

- Id care este cheia primară este de tip int identity;
- Nume_poza, varchar(50) reprezintă numele pozei pe care utilizatorul l-a introdus;
- Poza este câmpul care conține poza, fiind de tipul Image;
- Id_album este albumul din care face parte poza fiind cheie străină;
- Câmpul copertă este de tipul varchar(2) poate fi "DA" sau "NU" și indică dacă poza este copertă a albumului.

Tabelul **album** are câmpurile:

- Id, cheia primară a tabelului;
- Id_utilizator este cheia străină, fiind referința spre id-ul utilizatorului;
- Nume_album de tipul varchar(50) este denumirea albumului dată de utilizator.

Tabelul **comentarii** conține câmpurile:

- Id care este cheia primară a tabelului;
- Id_poza este id-ul pozei la care s-a făcut comentariu, fiind cheie straină cu referință la tabelul poze;
- id_utilizator reprezintă id-ul utilizatorului care a lăsat comentariul;
- comment este de tipul varchar(Max) și reprezintă comentariul propriu zis, trebuie să fie diferit de null.

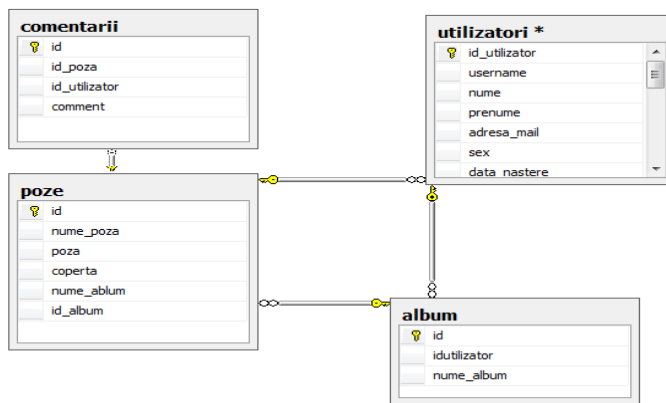


Figura 5.12 Tabelele pentru componenta albume de fotografii

Pentru vizualizarea albumelor de fotografii în nivelul de logică, este nevoie de o clasă specială *generateImages.ashx*. Acest fișier este un fișier de manipulare care generează imaginile în funcție de informația byte care este primită din baza de date.

La vizualizarea albumelor, fiecare album este afișat de către un repetor cu imaginea care este coperta de album și numele albumului, reprezentând o legătură spre vizualizarea pozelor. Vizualizarea albumelor se face în forma web viewPhoto.aspx care primește ca parametru id-ul primei poze din album. Imaginea unde va fi afișată fotografia are atributul ImageUrl egal cu clasa generateImages.ashx. Această clasă returnează poza după ce o primește de la clasa de logica photoBusiness.cs.

```

public void ProcessRequest(HttpContext context)
{
    string id = context.Request.QueryString["id"];
    byte[] buffer = pb.getbyteimage(id);
    context.Response.ContentType = "image/jpeg";
    context.Response.OutputStream.Write(buffer, 0, buffer.Length);
}

```

În clasa de logică este apelată metoda de extragere a imaginii în tipul byte din baza de date. Imaginea din prezentare este inclusă într-un UpdatePanel pentru a nu se reîncărca întreaga pagină la schimbarea pozei. La vizualizarea fiecărei poze în parte, utilizatorul va avea opțiunile de a șterge poza, să o facă copertă a albumului sau să o aleagă ca și poză de profil.

5.2.1.4 Blog

Utilizatorii accesează site-ul pentru a primii informații sau pentru a le furniza cu un anume subiect. De multe ori informația poate fii furnizată sub formă de blog. Blogul permite utilizatorilor să își spună întâmplările, să își scrie articolele sau să genereze subiecte de discuții cu un anumit subiect.

Claselor folosite pentru implementarea blogului sunt friendsBlog.aspx, friendsBlog.aspx.cs, blog.aspx, blog.aspx.cs și blogBusiness.cs. Introducerea de noi informații în blog de către utilizatorii autentificați se face cu clasa din nivelul de prezentare blog.aspx. Utilizatorul își va introduce în TextBox-urile corespunzătoare subiectul care îl ca avea blogul și conținutul. Cu informațiile transmise de interfață și id-ul utilizatorului clasa Blog.aspx.cs apelează metoda addBlog din clasa din nivelul de logică blogBusiness.cs.

Toate obiectele de tip blog se vor afișa într-un repetor după câmpurile unde utilizatorul poate adăuga conținut nou, în acest repetor fiind butoane care vor permite vizualizarea și ștergerea vechilor intrari. Dacă se selectează vizualizare blogului utilizatorul va putea să își modifice intrările. Pentru a identifica care din intrari va fi ștersă sau vizualizată se adăugă butonului un argument care va conține id-ul unic.

Modificarea blogului se face cu metoda updateBlog tot din blogBusiness care va avea ca parametrii id-ul blogului care va fi modificat, noul subiect și noul conținut.

```
<asp:Button ID="deleteblog" runat="server" align="right" Width="50px"
Text="delete" BackColor="RED" Font-Names="Bell MT" Font-Bold="true"
CommandArgument='<## DataBinder.Eval(Container.DataItem,"id_feedback") %>'
ForeColor="White" OnClick="deleteFeedback" Height="20px" />
```

Repetorul în care sunt afișate conținutul blogului este inclus într-un UpdatePanel pentru a de reîncărca intrările la fiecare modificare în baza de date.

Forma web friendsBlog.aspx va permite utilizatorilor din lista de prieteni să vadă intrările blogului. Deoarece HTTP este un protocol stateless este nevoie de o variabilă ascunsă care să rețină id-ul blogului care este vizualizat pentru a se putea adăuga comentariile blogului corespunzător. Lista de blog-uri a utilizatorului este returnată de metoda de business getBlog(string user).

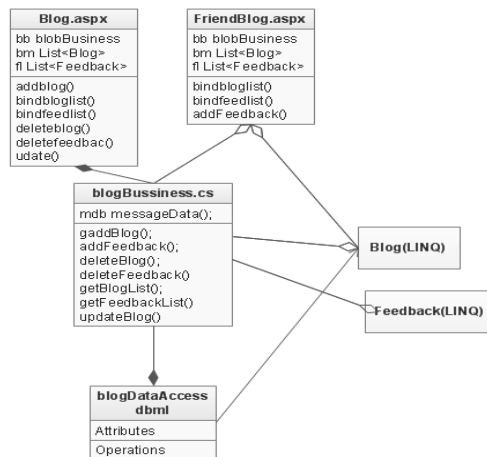


Figura 5.13 Diagrama de clase a componentei Blog

Tabele din baza de date de care are nevoie aceasta componentă sunt **blog**, **feedback**, și utilizatori. Tabelul **blog** conține următoarele câmpuri:

- **id_blog** care este cheie primară de tipul **int**, iar prin specificarea opțiunii **identity** aceasta se generează automat;
- **id_utilizator** este cheia străină care leagă tabelul de tabelul utilizatori;
- **Subiect_blog** este de tip **varchar(Max)** este permis să fie și **null**;
- **Text_blog** este conținutul introdus de utilizatori, de tipul **varchar(Max)**, nu este permis să fie **null**;
- **Data_postarii** este un câmp de tipul **varchar** reprezentând data la care a fost adăugat blogul de către utilizator.

Tabelul **feedback** sunt comentariile pe care ceilalți utilizatori le adaugă. Câmpurile tabelului sunt:

- **id_feedback** **blog** care este cheie primară de tipul **int**, care este generată automat;
- **id_utilizator** reprezintă **id-ul** utilizatorului care a introdus comentariul;
- **id_blog** este cheia străină cu care reprezintă legătura între **blog** și **feedback**;
- **username-ul** este numele de utilizator al celui care a introdus comentariul;
- **data_postarii** reprezintă data la care a fost introdus comentariul fiind de tipul **DateTime**;
- **mesaj** este conținutul comentariului și este de tipul **varchar(max)**;

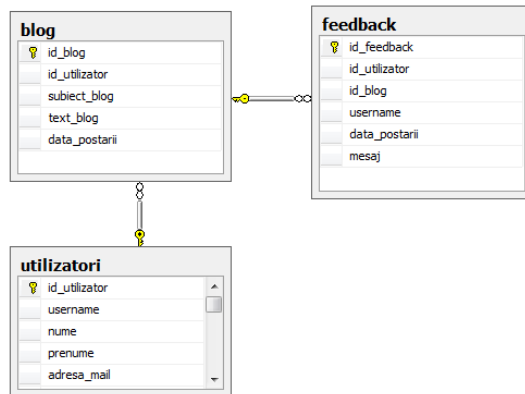


Figura 5.14 Tabelele din baza de date pentru componenta blog

5.2.1.5 Mesaje

Mesajele între utilizatori sunt importante deoarece astfel pot să trimită informații între ei astfel încât acestea nu se șteg oricât timp ar trece de la postarea lor și pot fi private, sunt trimise doar între doi utilizatori. Există și mesaje pe care utilizatorii le pot trimite administratorului pentru ca acesta să stie părerea utilizatorilor depre site.

Clasele care ajută la implementarea aceste componente sunt *sendMessage.aspx*, *sendMessage.aspx.cs*, *viewMessage.aspx*, *messageBusiness* și *messageDataAccess.dbml*. În clasa *sendMessage.aspx* va compune mesaje pentru a le trimite. Ca parametrii spre nivelul de logică vor fi transmiși sunt subiectul mesajului, mesajul și destinatarul. Dacă se va transmite un mesaj spre un alt utilizator simplu mesaje vor fi adăugate în tabela mesaje, iar dacă va fi transmis către admin se va insera în tabela mesaje_admin.

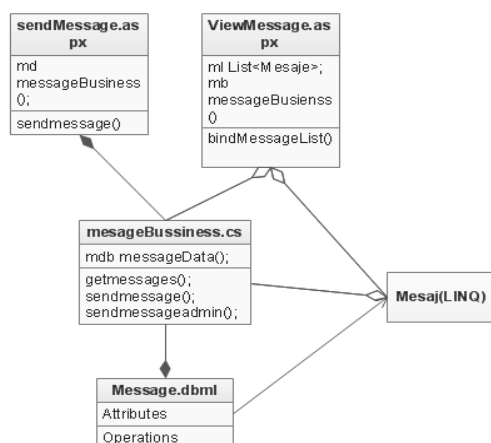


Figura 5.15 Diagrama de clase a componentei Mesaje

Tabelele din baza de date a aceste componente sunt mesajAdmin, mesaje si utilizatori.

Tabelul **mesajAdmin** conține următoarele câmpuri:

- id_mesaj este de tip int identity și este cheia primară a acestui tabel;
- username este de tip varchar(50) și reprezintă numele de utilizatorului care a transmis mesajul;
- Mesaj este de tipul varchar(MAX) fiind mesajul transmis, trebuie să fie diferit de null;
- Data_postarii este de tip DateTime este data în care a fost trimis mesajul;
- Id_utilizator este cheia straină cu care face legătura cu tabelul utilizatori.

Tabelul **mesaje** contine următoarele câmpuri:

- id_mesaj este de tip int identity și este cheia primară a acestui tabel
- delusername este de tipul varchar(50), trebuie să fie obligatoriu diferit de null, reprezentând numele de utilizator al celui care a trimis mesajul;
- id_utilizator este cheia straină cu care face legătura cu tabelul utilizatori fiind cheia utilizatorului care a primit mesajul;
- câmpul mesaj reprezinta mesajul propriu zis;
- data_expedierii este de tip DateTime este data în care a fost trimis mesajul;
- subiect este subiectul mesajului care a fost trimis.

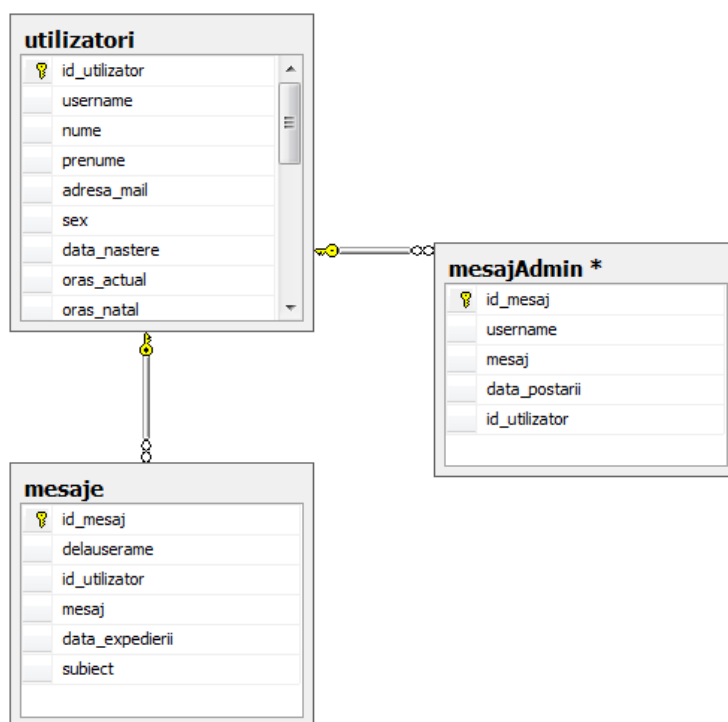


Figura 5.16 Tabelele din baza de date pentru trimitere mesaje

5.2.1.6 Conversații (Chat)

Componenta de chat implementată furnizează o cameră de conversații în care toți utilizatorii autentificați ai sistemului pot să adauge câte mesaje doresc, iar ceilalți pot să le vadă în timp real.

Clasele care le utilizează această componentă sunt Chat.aspx, Chat.aspx.cs, ChatPage.aspx, ChatPage.aspx.cs și Chat.cs. Chat-ul funcționează cu toate browserele care suport <iFrame>.

La reținerea mesajelor trimise de către utilizatori am folosit un array static. Array-ul static este declarat în clasa Chat.cs. Aici sunt citite și scrise toate mesajele publicate de utilizatori. Dacă numărul de mesaje este mai mare decât 500 atunci mesajele publicate cel mai devreme vor fi șterse pentru a fi stocate altele noi. Când utilizatorul adaugă un mesaj nou este apelată metoda addMessage care va adăuga un rând nou în array. Pagina conținută de tag-urile <iFrame> se va reîncărca la fiecare patru secunde fata ca utilizatorul să îi de reîncărcare.

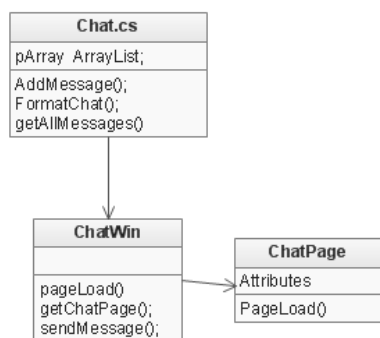


Figura 5.17 Diagrama de clase pentru camera de chat

5.2.1.7 Administrator

Pentru ca sistemul să fie protejat de utilizatorii care postează conținut interzis și pentru a menține baza de date este nevoie de partea de administrator. Administratorul are dreptul să vizualizeze toate profilurile utilizatorilor indiferent dacă este sau nu prieten cu ei. Clasele din nivelul de prezentare care fac parte din această componentă sunt *adminFirstPage.aspx*, *adminFirstPage.aspx.cs*, *deleteUsers.aspx*, *deleteUsers.aspx.cs*, *feedbackUsers.aspx*, *feedbackUsers.aspx.cs*, *newAdmin.aspx*, *newAdmin.aspx.cs*, *search.aspx*, *userProfile.aspx*, *userProfile.aspx.cs*, *Admin.master*. Toate paginile administratorului au aceeași pagină master. Acestea sunt puse într-un director special în care pot avea acces numai utilizatorii care au rolul de administrator. Clasele de logica și de acces la baza de date sunt *adminBusiness.cs* și *admin.dbml*.

Pentru operația de ștergere a unui utilizator al site-ului trebuie să existe utilizatori care au fost raportați. Pagina *deleteUsers* conține un gridview în care este afișată lista de utilizatori care au fost raportați. Fiecare rând al tabelului va conține coloana cu numele utilizatorului raportat care este legătură la pagina *userAdmin* care pagina lui de profil cu toate informațiile care le-a introdus.

5.2.2 Structura bazei de date

În această secțiune este prezentată diagrama bazei de date care conține toate tabelele folosite și modul în care acestea interacționează între ele.

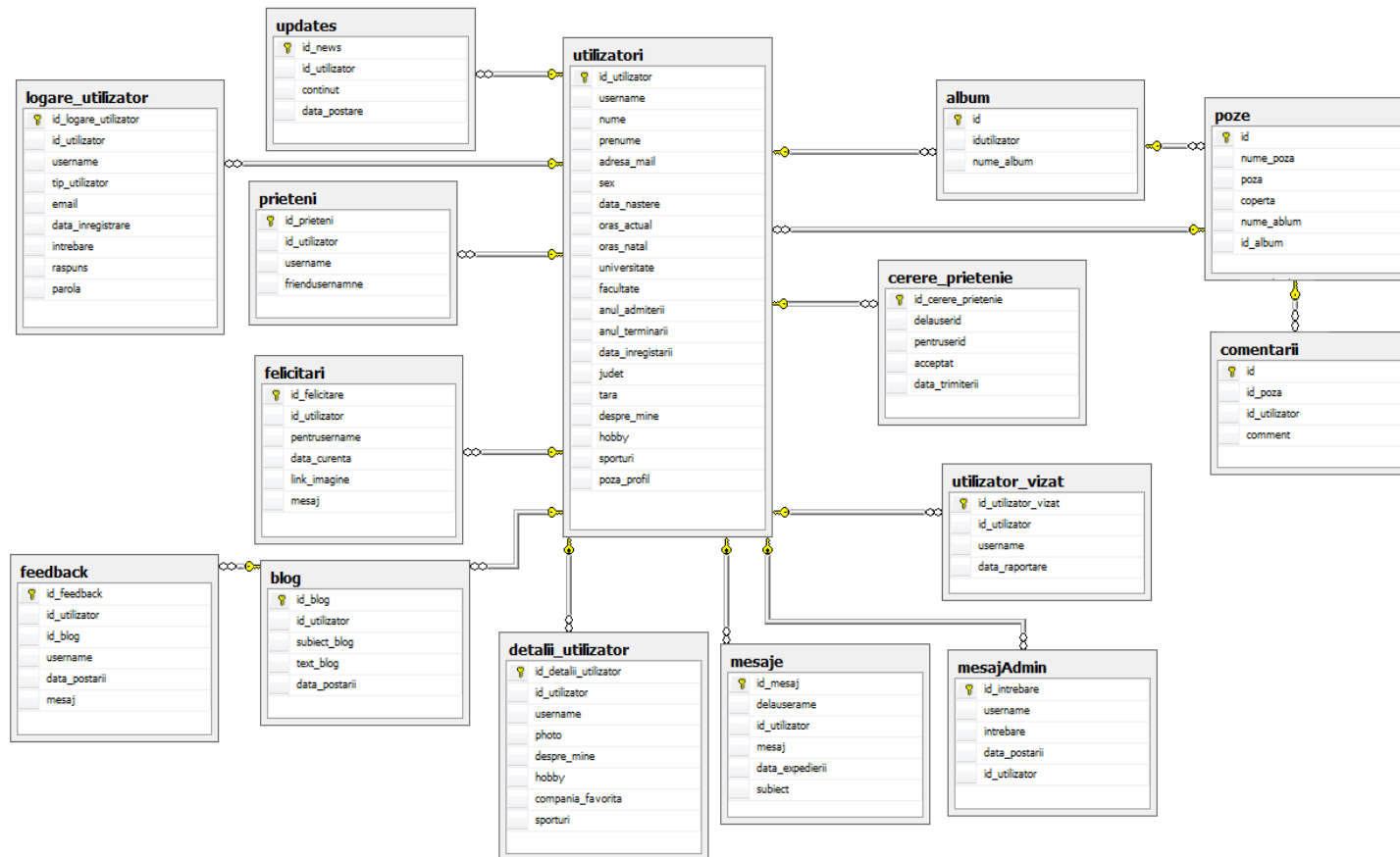


Figura 5.6 Structura tabelor din baza de date

5.2.3 Modelul de date-LINQ

Motivele pentru care am ales tehnologia Linq pentru accesul la baza de date sunt că nu are nevoie la fiecare modificare în baza de date de căutarea procedurilor stocate asociate, după care să se facă modificările în metodele relaționate din accesul la baza de date și pe urmă în metodele din nivelul de business. Cu ajutorul Linq la fiecare modificare se face o precompilare, în care se schimbă datele din memoria aplicației, după care se face modificarea și în baza de date. De asemenea codul Linq este ușor de citit, mai scurt și este din ce în ce mai portabil.

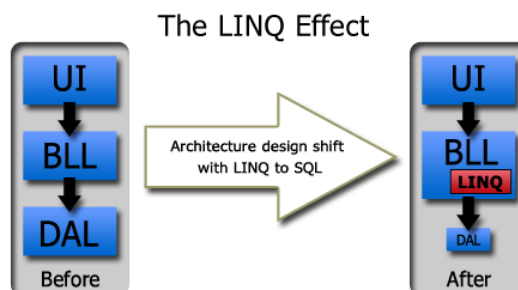


Figura 5.6 Efectul Linq asupra arhitecturii pe nivele.

Imaginea a fost preluat din [18]

UI (User Interface)-interfața utilizatorului;

BLL (Business Logic Layer)-nivelul de logică;

DAL (Data Access Layer) - Nivelul de acces la date;

Pentru fiecare tabel din baza de date este creată automat o clasă separată care are atributele câmpurile tabelului, iar metodele sunt aceleași pentru fiecare clasă generată de LINQ dar au parametrii specifici fiecărui tabel. Dacă există clase spre care tabelul are referință acestea vor fi incluse între atributele clasei, astfel se va putea accesa orice câmp al din tabele cu care are legătură.

Metodele din clasa generată de linq pentru tabelul albume sunt reprezentate în figura de mai jos.

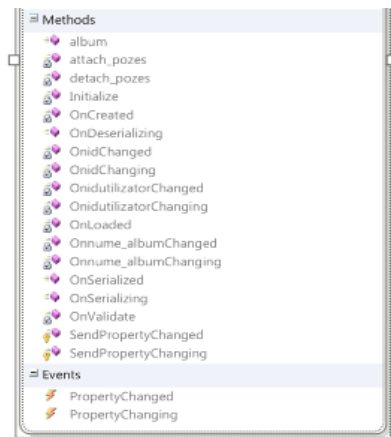


Figura 5.16 Metodele din clasa album generată de LINQ

Pentru ca modulele să fie mai evidente și cu scalabilitate îmbunătățită fiecare din componente câte o clasă Linq To SQL. Prin folosirea componentelor Linq To SQL operațiile care realizează conexiunea cu baza de date au fost facilitate, iar codul este mult mai ușor de înțeles.

6. Testare și validare

Scopul principal al procesului de testare este identificarea erorilor, izolarea și fixarea lor. Testarea nu este un proces care garantează funcționarea corectă a sistemului în orice condiții doar identifică unele cazuri în care sistemul nu funcționează conform așteptărilor.

Deseori în aplicațiile software există eșecuri din cauza problemelor de compatibilitate cauzate atât de interacțiunea lor cu alte aplicații sau sisteme de operare sau în cazul de față cu browserele. Siteul web de socializare poate rula cu ajutorul tuturor browserelor existente momentan pe piață: Google Chrom, Internet Explorer, Mozilla Firefox, Opera ș.a. Deși există încă unele diferențe de interpretare a codului scripturilor între Internet Explorer și Mozilla Firefox aplicația a fost realizată astfel încât să prezinte aceeași structură indiferent de browserul web.

Validarea are scopul de a confirma dacă un site îndeplinește cerințele funcționale și respectă cazurile de utilizare realizate înainte de implementării și cerințele nonfuncționale.

7.1 Cerințele funcționale

Pentru reducerea timpului de implementare verificarea respectării cazurilor de utilizare s-a făcut pe parcursul implementării și pentru că există componente care sunt dependente, iar dacă una dintre ele are o eroare o propagă mai departe.

Testele făcute la finalul implementării sunt testele care implică mai mult partea de interfață:

- Testarea legăturilor între paginile web ale sistemului;
- Afișarea corectă a conținutului în pagină;
- Modul în care componentele din interfață interacționează;
- Apariția mesajelor de confirmare la operațiilor mai radicale (dezactivarea, contului, ștergerea utilizatorului).

7.2 Cerințe non-funcționale

Cerințele non-funcționale sunt cerințele care impun constrângeri cu privire la proiectare sau implementare, care nu sunt legate de funcții ale sistemului. Cerințele non funcționale au fost testate atât în timpul implementării cât și la final.

7.2.1 Performanța

Performanța este un ansamblu de cerințe și restricții impuse de o funcționare optimă a sistemului, în condițiile satisfacerii utilizatorilor.

Framework-ul .Net permite păstrarea în memoria tampon a informației din baza de date. Deoarece utilizatorii lucrează cu aceste informații face ca websiteul să aibă un timp de răspuns mult mai bun. În plus cu ajutorul panoului de modificare din biblioteca AJAX

ASP.NET modificările realizate de utilizator se vor modifica automat fără sa fie nevoie de reîncărcarea paginii web.

7.2.2 Securitatea

Pentru îmbunătățirea securității am folosit autentificarea cu ajutorul formei de autentificare a ASP.NET. Acest mod de autentificare permite autentificarea în funcție de rolul (administrator și utilizator). În meniul de configuration tool în partea Securitate sunt setate rolurile utilizatorilor și accesul acestora asupra paginilor web ale siteului web. Paginile administratorului nu vor putea fi vizualizate de utilizatorii simplii deoarece au accesul restricționat asupra acestui director. Rolul utilizatorilor care accesează sistemul este cunoscut în momentul în care aceștia se autentifică cu ajutorul numelui de utilizator și al parolei. În cazul în care utilizatorul își uită parola aceasta poate fi recuperată numai printr-un email trimis la adresa introdusă la înregistrare care va conține o parolă generată automat de sistem. La modificarea informațiilor personale ale utilizatorului acesta trebuie să își introducă din nou parola de asemenea și la dezactivarea contului.

După autentificare este numele utilizatorului va fi reținut într-o sesiune care va păstra utilizatorul logat, dacă trec 20 de minute de neactivitate sesiunea va expira. Tot pentru îmbunătățirea securității utilizatorilor nu le este permis să dețină o parolă mai scurtă de cinci caractere, iar la adăugarea sau modificarea parolei acesta este informat de nivelul de securitate care îl furnizează.

7.2.3 Scalabilitatea

Structurarea proiectului pe componente permite adăugarea fără efort considerabil a altor module noi. De asemenea împărțirea sistemului pe trei niveluri arhitecturale face posibilă o creșterea complexității componentelor deja existente și adăugarea unor funcționalități noi.

7. Manual de instalare și utilizare

În acest capitol vor fi prezentate condițiile de funcționare și modul în care funcționează siteul de socializare

7.1 Manual de utilizare

Pentru accesarea siteului de socializare Friends se introduce în browser adresa *http://localhost:50853/WebSite1/login.aspx* . Pentru ca aceasta să funcționeze serverul IIS trebuie să fie pornit.

Deoarece unul din obiectivele principale a fost ca siteul web să aibă o interfață cât mai prietenoasă legăturile între pagini și bara de meniuri sunt intuitive. După autentificare utilizatorul va fi redirecționat spre pagina de profil unde sunt legături spre toate componentele principale. După accesarea fiecărei componente există legături spre toate funcționalitățile acestora.

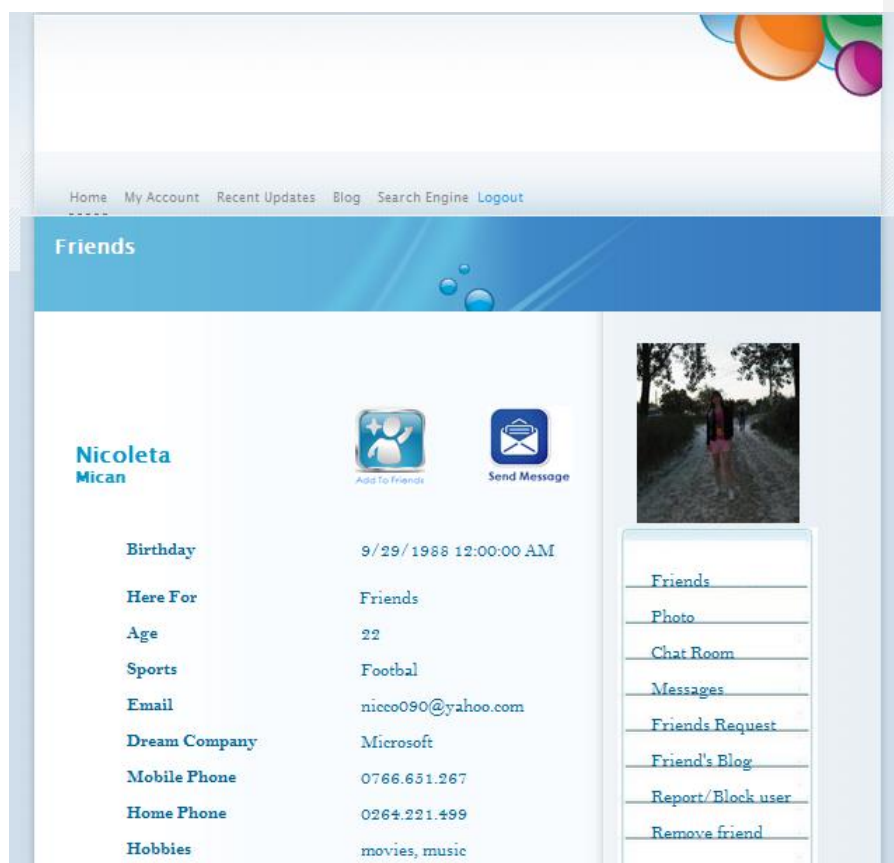


Figura 7.1 Pagina de profil a unui utilizator care nu e prieten cu utilizatorul autentificat

Bara de meniul din pagina de profil conține:

- *Home*- pagina de profil;
- *My Account* – pagina unde se fac modificările informațiilor de profil;
- *Recent Updates* - pagina cu ultimile modificările făcute de utilizatorii prieteni;
- *Blog* –secțiunea de blog a siteului de socializare;
- *Search Engine* – pagina unde se poate face căutarea utilizatorilor înregistrați;
- *Logout* – realizează delogarea utilizatorului.

În partea dreaptă există următoarele legături în pagina de profil:

- *Friends* – reprezintă legătura spre pagina cu prietenii utilizatorului;
- *Photo* –pagina de galerii de fotografii;
- *Chat Room*- legătura spre camera de chat;
- *Messages* – pagina unde sunt afișate mesajele de la utilizatori;
- *Friends Request* –pagina cu cererile de prietenie care nu au fost acceptate sau respinse.

Pentru vizualizarea profilului altor utilizatori apar în plus în această secțiune:

- *Friend's Blog* –pagina spre blogul utilizatorului al cui e profilul;
- *Report/Block User*- raportarea utilizatorului;
- *Remove Friend* – stergerea din lista de prieteni.

Pe lângă opțiunile din cele două meniuri în stânga paginii de profil există două butoane unul pentru trimitere cerere de prietenie și un buton care redirecționează la pagina din care se poate trimite mesaj utilizatorului al cui e profilul.

Paginile de administrator au doar meniul în partea dreaptă a siteului, funcționalitățile acestuia fiind doar de gestionare. Acestea sunt:

- *Home* – pagina în care apar toate modificările recente făcute de utilizatori;
- *Target users*- pagina listei de utilizatori raportați care vor putea fi șterși;
- *Add admin* –pagina care permite adăugarea unui administrator;
- *Messages*- pagina unde sunt afișate mesajele de la utilizatori;
- *Search* – pagina în care se poate face căutarea utilizatorilor;
- *Feedback* –pagina cu mesajele trimise către administratori de utilizatori;
- *Logout* –delogarea administratorului.

7.2 Cerințe hardware și software

Din punct de vedere hardware, pentru accesarea siteului de socializare Friends caracteristicile minime recomandate sunt: o stație de lucru Pentium 3 2GBHz , 512 MB RAM, 20 GB Hdd, placă de rețea 10/100Mbps și o conexiune la internet, cu o viteză de transfer de 128KB/sec. Din punct de vedere software, este nevoie de un sistem de operare, recomandabil Windows cu una din versiunile Microsoft Windows XP , Microsoft Windows Vista sau Windows 7 pentru un sistem de calcul mai performant. Ca software preinstalat, este nevoie de orice browser de internet modern, recomandabil ultimele versiuni apărute.

8. Concluzii și dezvoltări ulterioare

8.1 Concluzii

Obiectivul principal a acestei lucrări a fost implementarea unui web site de socializare cu ajutorul tehnologiilor Microsoft care să prezinte cât mai multe din componentele specifice ale rețelelor sociale. De asemenea s-a realizat consolidarea și dobândirea de cunoștințe noi referitoare la tehnologiile Microsoft utilizate și nu numai.

Prin lucrarea de față am reușit realizarea unui site web de socializare care este util persoanelor sociabile, dar care totodată sunt dependente de internet. Siteul de socializare *Friends* permite păstrarea legăturilor între prieteni cu ajutorul internetului fără ca utilizatorii să piardă mult timp cu distanța pentru a fi împreună cu prietenii lor, astfel relațiile de prietenie pot fi create și întreținute la nivel global.

Pentru ca siteul realizat să fie cât mai atractiv pentru utilizatori acesta are o interfață prietenoasă care permite utilizatorilor navigarea cu ușurință între componente. Utilizatorii sistemului de socializare *Friends* vor reveni mereu pe site pentru că au acces la ultimile activități importante ale prietenilor, pot să le vadă pozele, să citească comentariile lăsate la secțiunile de blog și de albume de fotografii de alți utilizatori, să primească mesaje, să interacționeze cu persoane noi în camera de chat, sau să își publice informațiile și conținutul propriu pentru ca prietenii lor virtuali să poată avea acces la el.

Relațiile între componente sistemului sunt astfel încât realizate modificarea uneia dintre acestea să nu afecteze pe cât posibil funcționalitatea celorlalte, iar schimbările în cadrul uneia sa poată fi realizate rapid și fără erori propagate în celelalte componentă.

8.2 Dezvoltări ulterioare

Deoarece în zilele noastre dispozitivele mobile sunt tot mai performante și pot sustine aplicații tot mai complexe ca dezvoltare ulterioară poate fi implementată o miniaplicație în platforma Windows Phone Application. Cu ajutorul celor două aplicații utilizatorii putea interacționa indiferent dacă folosesc un computer sau un telefon mobil.

Deoarece sistemul de față permite doar comunicarea instanță doar în camera de chat ulterior pentru conversații private se poate realiza un chat care să se facă doar între doi utilizatori care sunt de acord să înceapă o discuție. Canalul de chat privat între două persoane se poate implementa sub formă de videochat astfel utilizatorii care cominica să se poată vedea cu ajutorul camerei web. Pentru o îmbunătățire a interfeței grafice se poate utiliza tehnologia nouă pentru partea de client furnizată de Microsoft : Silverlight, tehnologie care permite aplicații mai complexe și cu interfață mai dinamică. De asemenea un site de socializare ar putea permite împărțirea și altor tipuri de fișiere înafara de fotografii cum ar fi muzica, video sau formate care contin text.

Pentru ca socializarea să fie mai ușor de realizat sistemul ar putea conține o componentă care să permită importarea prietenilor de pe alte sisteme similare sau invitarea prietenilor din lista de contacte de la sistemele de email. Astfel utilizatori noi nu trebuie să își

Capitolul 8

caute prietenii cu care interacționează online cu alte sisteme în motorul de căutare, ci poate să apase câteva butoane și contactele să fie realizate.

În multe din siteurile de socializare există o componentă de grup. Aceasta ar putea fi adăugată și la siteul de socializare Friends realizat. Componenta grup va permite utilizatorilor să creeze sau să se alăture unui grup de utilizatori care au aceleași interese, iar conținutul propriu va fi împărțit cu întregul grup.

Pentru persoanele publice se poate implementa o pagină specială de profil care să permită fanilor să urmărească pagina.

BIBLIOGRAFIE

- [1] **Andrew Siemer** “*ASP.NET 3.5 Social Networking*”, Packt Publishing, 2008
- [2] **D. Chappell**, “*Understanding .NET - A Tutorial and Analysis*”, Addison-Wesley Pearson Education, 2002.
- [3] **I. Salomie, T. Cioara, I. Anghel, T. Salomie** – “*Distributed Computing and Systems A Practical Approach*”, Ed. Grupul Microinformatica, Cluj Napoca, 2008
- [4] **Matthew MacDonald, Adam Freeman, Mario Szpuszta** “*Pro ASP.NET 4 in Csharp 2010*”, (4th Edition) Apress 2010
- [5] **R. Dollinger, L. Andron**, “*Baze de date si Gestiunea Tranzactiilor*”, Editura Albastra, 2004
- [6] **Mridula Parihar „ASP.NET Bible”, Editura Hungry Minds, Inc., 2008**
- [7] Allen, Christopher http://www.lifewithalacrity.com/2004/10/tracing_the_evo.html (accesat la 7.12.2010)
- [8] C# Help, <http://www.csharpshelp.com> (accesat la 4.2.2011)
- [9] Kanaka Juvva Dependable Embedded System (accesat la 10.4.2011) http://www.ece.cmu.edu/~koopman/des_s99/security/index.html
- [10] LinkedIn Learning Center <http://learn.linkedin.com/> (accesat la 5.12.2010)
- [11] Ning <http://www.ning.com/chooseplan> (accesat la 5.12.2010)
- [12] Stowe Boyd <http://www.stoweboyd.com/post/2325281845/are-you-ready-for-social-software> (21.05.2011)
- [13] Wikipedia, the free encyclopedia <http://en.wikipedia.org/wiki/LinkedIn> (accesat la 5.12.2010)
- [14] Wikipedia, the free encyclopedia, <http://en.wikipedia.org/wiki/Myspace> (accesat la 5.12.2010)
- [15] Wikipedia, the free encyclopedia [http://en.wikipedia.org/wiki/Ning_\(website\)](http://en.wikipedia.org/wiki/Ning_(website)) (accesat la 5.12.2010)
- [16] Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Social_software (accesat la 5.12.2010)
- [17] Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Cascading_Style_Sheets (accesat la 15.11.2010)
- [18] Blog <http://www.adamtibi.net/08-2008/linq-to-sql-the-data-access-layer-dal-shrinker> (accesat la 12.03.2011)

Acronime

AJAX- Asynchronous JavaScript and XML

ASP – Active Server Pages

BLL-Business Logic Layer

CLR – Common Language Runtime

CSS – Cascading Style Sheets

CTS – Common Type System

DAL- Data Access Layer

FCL – Framework Class Library

HTML – Hyper Text Markup language

HTTP – Hyper Text Transfer Protocol

IIS- Internet Information Server

IL -Intermediate Language

JPEG- Joint Photographic Experts Group

LINQ- Language Integrated Query

MVC- Model–view–controller

OOP – Object-oriented programming

UI - User Interface

UML – Unified Modelling Language

URL – Uniform Resource Locator

XML – eXtensible Markup Language

Anexe

Anexa 1

Lista Figurilor

Figura 4.1 Arhitectura de ansamblu a sistemului	16
Figura 5.1 Diagrama caz de utilizare pentru utilizator înregistrat.....	30
Figura 5.2 Diagrama de secvență pentru logare.....	31
Figura 5.3 Diagrama caz de utilizare pentru adăugare fotografie.....	43
Figura 5.4 Diagrama caz de utilizare schimbare parolă.....	33
Figura 5.5 Diagrama caz de utilizare administrator.....	34
Figura 5.6 Arhitectura generală sistemului.....	35
Figura 5.7 Diagrama de clase pentru generarea imaginii Captcha.....	37
Figura 5.8 Caractere captcha.....	39
Figura 5.9 Diagrama din baza de date a utilizatorilor.....	42
Figura 5.10 Diagrama de bază de date a componentei socializare.....	44
Figura 5.11 Diagrama de clase pentru componenta albume de fotografii.....	45
Figura 5.12 Tabelele pentru componenta albume de fotografii.....	46
Figura 5.13 Diagrama de clase a componentei Blog.....	48
Figura 5.14 Tabelele din baza de date pentru componenta blog.....	50
Figura 5.15 Diagrama de clase a componentei Mesaje.....	50
Figura 5.16 Tabelele din baza de date pentru trimitere mesaje.....	51
Figura 5.17 Diagrama de clase pentru camera de chat	52
Figura 5.18 Structura tabelor din baza de date	53
Figura 5.19 Efectul Linq asupra arhitecturii pe nivele.....	54
Figura 5.20 Metodele din clasa album generată de LINQ	55
Figura 7.1 Pagina de profil a unui utilizator care nu e prieten cu utilizatorul autentificat	58