***„Get your paper published”***

**Aplicație web pentru evaluarea lucrărilor științifice**

|  |  |
| --- | --- |
| **CHIRA ALEXANDRA-DIANA** | **D:\New folder (2)\FSEGA-Informatica Economica\Poze Diploma\Chira Alexandra-Diana.jpg** |

Cuprins

[Partea I – Raport de Analiză 4](#_Toc516754517)

[Introducere 4](#_Toc516754518)

[1. Identificarea și Descrierea Problemei 6](#_Toc516754519)

[1.1 Context 6](#_Toc516754520)

[1.1.1 Fațeta subiect 6](#_Toc516754521)

[1.1.2 Fațeta utilizare 7](#_Toc516754522)

[1.1.3 Fațeta IT 8](#_Toc516754523)

[1.1.4 Fațeta dezvoltare 8](#_Toc516754524)

[1.2 Motivație 8](#_Toc516754525)

[1.2.1 Diagrama Fishbone 9](#_Toc516754526)

[1.2.2 Diagrama Pareto 11](#_Toc516754527)

[1.2.4 Diagrama de descompunere a obiectivelor 13](#_Toc516754528)

[2. Cerințe de sistem 22](#_Toc516754529)

[2.1 Surse de cerințe 22](#_Toc516754530)

[2.2 Elicitația cerințelor 23](#_Toc516754531)

[2.3 Documentarea cerințelor 23](#_Toc516754532)

[2.3.1 Procese 25](#_Toc516754533)

[2.3.2 Cazuri de utilizare 28](#_Toc516754534)

[2.3.3 Brainstorming 29](#_Toc516754535)

[2.3.4 Interviu 30](#_Toc516754536)

[3.Model de dezvoltare 31](#_Toc516754537)

[Partea II – Proiectarea Sistemului Informatic 33](#_Toc516754538)

[4. Proiectarea Logică 34](#_Toc516754539)

[4.1 Arhitectura Sistemului 36](#_Toc516754541)

[4.2 Baza Informaţională 40](#_Toc516754542)

[5. Proiectarea Tehnică 42](#_Toc516754543)

[5.1 Proiectarea interfeţei cu utilizatorul 42](#_Toc516754544)

[5.2 Procese şi Algoritmi 45](#_Toc516754545)

[5.3 Structura fizică a datelor 48](#_Toc516754546)

[5.4 Tehnologii specifice 52](#_Toc516754547)

[Glosar 53](#_Toc516754548)

[Bibliografie 56](#_Toc516754549)

# Partea I – Raport de Analiză

# Introducere

Tehnologia a avansat atât de mult încât este prezentă peste tot in viața noastră, neexistând niciun loc pe glob unde să nu fi pătruns acest trend important al ultimelor două secole. În prezent, aceasta ocupă o parte tot mai mare din viața noastră. Societatea actuală, prin intermediul globalizării, se schimbă într-o manieră incredibil de rapidă. Evoluția tehnologică continuă. În aceeași manieră, noi, oamenii evoluăm odată cu ea.

Necesitatea dezvoltării profesionale, dar și personale a omului se bazează într-o anumită etapă pe crearea unor lucrări știintifice în direcția domeniului și nivelului spre care ne axăm. Așadar, aici intervine momentul în care am luat decizia de a concepe o aplicație web de tip *peer review* (evaluarea sau recenzia profesională).

*Peer review* sugerează recenzarea unei lucrări de către unul sau mai multe persoane care sunt capabile si similare din punct de vedere profesional pentru a verifica și înainta publicarea acestei lucrări în procedura unei conferințe sau într-o revistă științifică.

Procesul de tip *peer review* este benefic în condițiile în care lucrarea revizuită de grupul de persoane mai sus menționat este îmbunătățită și sunt aduse indicații autorului pentru a putea fi publicată și/sau susținută. Dupa efectuarea evaluării de către *peer review*-uri, editorul va putea determina aprobarea sau respingerea lucrării spre publicare. În final, scopul este de a se publica dacă sunt îndeplinite cerințele, alături de informația aferentă în concordanță cu tema aleasă si la standardele stabilite.

*“Get your paper published”* iese în evidență datorită ușurinței cu care este folosit, reprezentând un concurent serios pentru cele mai cunoscute *site*-uri, asemenea *easychair.org* sau *edas.info*.

Aplicația web este realizată in asp.NET C# folosind programarea orientată pe obiecte, alături de *framework*-uri de specialitate. Cu ajutorul acestora, reiese un *website* ușor de folosit cu o interfață facilă, dotat cu comenzile specifice pentru a face folosirea acestuia cât mai “prietenoasă” pentru utilizatori.

Aplicația este adresată persoanelor doritoare de a se afirma într-un anumit domeniu profesional, prin publicarea sau/și susținerea unor lucrări științifice a persoanelor respective. De regulă, ținta propusă de lucrare este domeniul didactic, grupurile profesorale.

# 1. Identificarea și Descrierea Problemei

Datorită evoluării continue și rapide a tehnologie sunt redactate și susținute din ce în ce mai multe lucrări științifice. Cu ajutorul acestora, ne perfecționăm, excelăm și dobândim cunoștințe.

Organizația țintă poate fi o universitate, o facultate sau un grup profesoral mai restrâns. În concluzie, aplicația este pentru cadrul didactic, în principiu pentru Departamentul de Informatică Economică din Facultății de Științe Economice și Gestiunea Afacerilor, pentru gestiunea conferințelor și a lucrărilor științifice. Departamentul mai sus menționat nu dispune de un astfel de sistem de gestiune, iar *site*-urile concurente nu satisfac pe deplin nevoile cadrului profesoral. Principalul obiectiv al procesului de *peer review* este de a ridica standardele calitative și de a asigura informație cât mai corectă, precisă și de actualitate.

## Context

### 1.1.1 Fațeta subiect

Modul de funcționare a aplicației este următorul :

1. Se creează un cont de utilizator
2. După introducerea datelor în formularul de înregistrare, utilizatorul primește un *e-mail* cu un *link* (în interioul căruia este scrisă o cheie de verificare unică și adresa de *e-mail*). În urma apăsării *link*-ului menționat anterior, utilizatorul este redirecționat către o nouă pagină unde își creează o parolă.
3. Odată salvată parola în baza de date, utilizatorul este trimis către pagina de *login*, unde se poate autentifica cu noul cont.

O persoană poate avea mai multe roluri, cum ar fi:

1. *Chair*

*Chair*-ul deleagă ce recenzori vor evalua lucrarea autorului, prin intermediul unui *e-mail* primit de la acesta, sub forma unei invitații.

1. Autor

Autorul este acela care dorește să își publice/susțină lucrarea științifică, iar în urma deciziei chair-ului, lucrarea sa va fi evaluată de recenzori.

În urma evaluării, lucrarea poate fi aprobată, refuzată sau completată pentru a fi îmbunătățită.

1. Evaluator

Evaluatorul sau recenzorul, va evalua lucrarea științifică a autorului, după primirea invitației din partea *chair*-ului. Cum a fost menționat la partea de obiective, este necesară aducerea de recomandări din partea recenzorului, pentru o lucrare îmbunătățită din punct de vedere al calității.

Pe partea de administrare, administratorul nu va avea nevoie de cont de utilizator, deoarece va opera direct din baza de date. Acesta va alege *chair*-ul conferinței.

În cadrul conferințelor, autorii își vor putea susține lucrările științifice.

### 1.1.2 Fațeta utilizare

În timpul utilizării acestei aplicații, părțile interesate urmăresc o bună desfășurare a întregului proces, opinii și sugestii cât mai adecvate și folositoare, iar în consecință, publicarea/susținerea lucrărilor științifice. Principala menire a sistemului de gestiune a lucrărilor științifice este de a menține standarde calitative, astfel încat autorul să se afirme, iar informația să fie în rezonanță cu tema abordată. Se asigura unicitate, îmbunătățirea calității și lizibilității articolului.

Utilizatorul poate avea trei roluri:

* *Chair* – stabilește cine va evalua o anumită lucrare științifică
* Autor – încarcă lucrarea spre a fi evaluată conform criteriilor aferente
* Evaluator – evaluează lucrările autorilor din toate punctele de vedere, venind cu propuneri și îmbunătățiri conform criteriilor impuse

În concluzie, fluxul de date ilustrat este denumit ca fiind un proces de tip *peer review*, care necesită o anumită pregătire din partea tuturor persoanelor incluse. Denumită în română, recenzarea va fi practicată de evaluatori, după invitația *chair*-ului, experți în domeniul abordat sau tema abordată, pentru o lucrare științifică cât mai competentă din toate punctele de vedere, dar mai ales din punct de vedere calitativ.

### 1.1.3 Fațeta IT

Aplicația se bazează pe o bază de date bine structurată, unde sunt salvate informațiile despre utilizatori, conferințe, recenzo și așa mai departe. Este folosit un server local, SQL Server, inclus în programul *software* Visual Studio 2015. Datorită acestuia, dezvoltarea aplicației este mai ușoară. Datele sunt stocate, de aceea pot fi manipulate cu ușurință și mereu la îndemână, în timp real.

Aplicația poate fi folosită atât pe telefon, dar și de pe laptop, fiind nevoie de o conexiune la internet. Aplicația a fost testată pe fiecare *browser*, în special: Google Chrome, Mozilla Firefox.

### 1.1.4 Fațeta de**z**voltare

În privința procesului se folosește modelul de tip spirală, ca și un model generator pentru proiectul *software*. Se bazează pe riscurile unice ale fiecărui proiect. În plus, acesta trece în mod repetat prin faze la fiecare incrementare. În acest model, partea interesată este conștientă întotdeauna de toate evenimentele din dezvoltarea *software*-ului.

Pas cu pas, începând cu analiza cerințelor părților interesate și rezumarea acestora în obiectivele, și până la aplicația în sine.

Deși pare un model complex, modelul spiralat se mapează perfect pe tema abordată.

## 1.2 Motivație

Principalul scop al aplicației “*Get your paper published*” este de a veni în ajutorul grupului țintă care dorește să beneficieze de sugestiile și recomandările recenzorilor în scopul unei lucrări științifice adecvate din punct de vedere al calității și eficacității.

Grupul țintă mai sus menționat este reprezentat de caderele didactice din instituțiile de învățământ în speranța evaluării lucrărilor academice pentru o abordare cât mai bună.

### 1.2.1 Diagrama Fishbone

Acest tip de sistem de gestionare a lucrărilor științifice trebuie să fie cât mai atractiv, cu o interfață ușor de înțeles, iar *flow*-ul funcționalităților să poată fi folosit cu ușurință. Ideea principală a acestui sistem este ca autorii să fie evaluați și publicați, cu lucrări de calitate, fără costuri adiționale.

În principiu, sistemele de gestiune cele mai cunoscute care fac parte din concurență sunt *easychair.org* sau *edas.info,* care nu sunt întocmai preferate de majoritate, cel din urmă având chiar taxe pentru folosirea acestuia.

Diagrama “*Fishbone*” redă informații cât mai amănunțite și ilustrează relațiile de tip cauză-efect.

Cauzele aflate pe “șira spinării” reprezintă cauzele principale, având o denumire mai generală, care integrează alte cauze, cele secundare.

După cum s-a menționat și mai sus, *target-*ul sugerat este unul destul de restrâns, grupuri mici, iar aceste activități de evaluare a lucrărilor științifice sunt periodice la un anumit grup țintă. De aceea, acest tip de sistem de gestiune nu este foarte cunoscut, decât în eventuala posibilitate a folosirii.

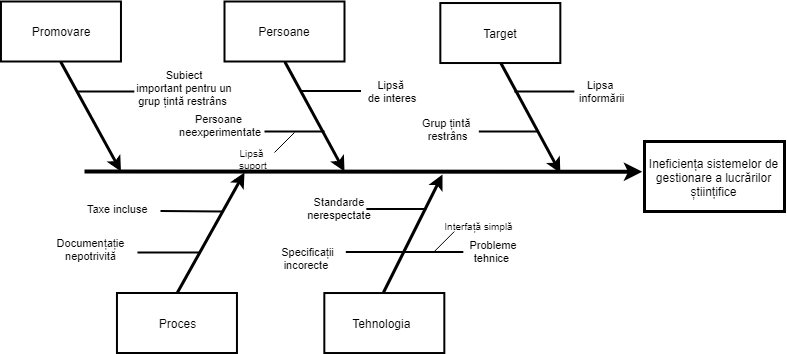


Figura 1 – Diagrama “Fishbone

### 1.2.2 Diagrama Pareto

Diagrama “*Pareto*” este un instrument grafic prin care se reprezintă prioritățile în cazul mai multor factori, pe baza distribuției efectelor a anumitor cauze, ierarhizate de la cele mai frecvente la cele mai puțin frecvente. Practic, diagrama redă cu ușurință problemele existente și gradul importanței lor.

Figura 2 – Diagrama Pareto

### 1.2.4 Diagrama de descompunere a obiectivelor

În prim plan se urmărește implementarea unui sistem de gestionare a lucrărilor științifice pentru evaluarea, îmbunătățirea și publicarea/susținerea acestora.

Din obiectivul principal se pot desprinde alte obiective secundare, și anume: implementarea unor funcționalități noi, creșterea calității lucrărilor academice pe baza recenziilor și aprofundarea informației ca fiind corectă și în raport cu stadiul dezvoltării științifice, încurajarea originalității și exprimarea autorului, implicarea recenzorilor.

Prima parte se poate ilustra ca și un plus față de aplicațiile *web* asigurate de competiție, și anume, aplicația va fi gratuită, fără taxe, partea de autentificare și înregistrare va fi ușor de folosit și sigură, cu ajutorul adresei de *e-mail* și al parolei.

Pentru partea de creștere a calității se dorește o abordare conform normelor impuse, ideea ca autorul să lanseze o idee teoretică cât mai concretă și să intemeieze pe baza acesteia, un șir de argumente pentru a dobândi interesul receptorului, astfel încat finalul să corespundă celor mai exigente standarde. Se urmărește prin urmare: rigurozitate, acuratețe și onestitate.

Cea de a treia parte ilustrează autorul, acesta folosindu-se de resursele intelectuale, capacitatea de analiză și sinteză, coerenţa gândirii şi a rostirii, elocvenţă. Autorul trebuie să-și susțină ipoteza principală, folosindu-se de o exprimare elevate și elegantă, și bineînțeles, de corectitudinea verbală. În mod normal, originalitatea nu presupune neapărat tratarea temei care nu a mai fost abordată vreodată, ar fi absurd. De accea, autorul va trebui să nuanțeze și să interpreze totul într-o manieră cât mai personală, formală.

Ultima parte implică abordarea obiectivă a recenzorilor, de obicei voluntari sau invitați, care este obligatorie, pentru a îmbunătăți și a aduce în plus recomandări. Astfel, munca autorului poate fi și mai bună.

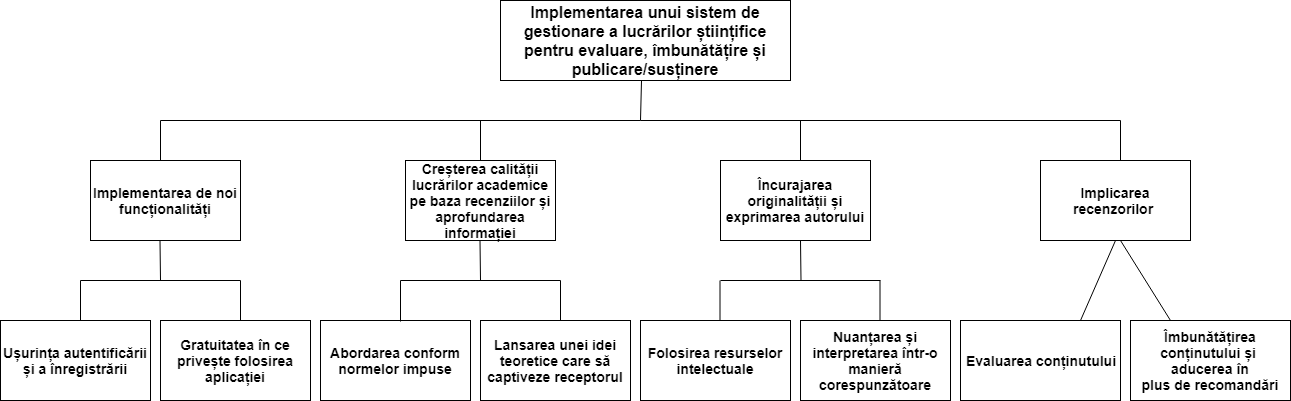


Figura 3 - Diagrama de descompunere a obiectivelor

Pentru o înțelegere mai aprofundată a obiectivelor, se va detalia fiecare scop în parte:

1. **Implementarea unui sistem de gestionare a lucrărilor științifice pentru evaluare, îmbunătățire și publicare/susținere**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | ISG |
| Nume | Implementarea unui sistem de gestionare a lucrărilor științifice pentru evaluare, îmbunătățire și publicare/susținere |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Beneficiarii (Cadrul didactic) |
| Nivel | Sistem |
| Descrierea scopului | Sistemul este benefic pentru cadrele didactive în scopul avansării pe plan profesional, prin scrierea și susținerea unei lucrări științifice |
| Scop părinte | Nu există |
| Sub-scopuri | Implementarea de noi funcționalități, creșterea calității lucrărilor academice pe baza recenziilor și aprofundarea informației, încurajarea originalității și exprimarea autorului, implicarea recenzorilor. |
| Dependențe de alte scopuri | Nu există |

1. **Implementarea de noi funcționalități**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | INF |
| Nume | Implementarea de noi funcționalități |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Beneficiarii (Cadrul didactic) |
| Nivel | Sistem |
| Descrierea scopului | Sunt implementate aceste funcționalități pentru o bună desfășurare a activității pe *site* |
| Scop părinte | Implementarea unui sistem de gestionare a lucrărilor științifice pentru evaluare, îmbunătățire și publicare/susținere |
| Sub-scopuri | Ușurința autentificării și a înregistrării și gratuitate pentru folosirea aplicației *web* |
| Dependențe de alte scopuri | Nu există |

1. **Ușurința autentificării și/sau înregistrării**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | INF |
| Nume | Ușurința autentificării și/sau înregistrării |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Beneficiarii (Cadrul didactic) |
| Nivel | Sistem |
| Descrierea scopului | Înregistrarea și autenficarea reprezintă primul pas spre folosirea aplicației *web,* de aceea trebuie să fie cât mai ușor de folosit |
| Scop părinte | Implementarea de noi funcționalități |
| Sub-scopuri | Nu există |
| Dependențe de alte scopuri | Nu există |

1. **Gratuitate în folosirea aplicației *web***

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | G |
| Nume | Gratuitate în folosirea aplicației *web* |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Beneficiarii (Cadrul didactic) |
| Nivel | Sistem |
| Descrierea scopului | Un competitor acordă folosirea aplicației contra-cost, de aceea, pentru a aduce un plus aplicației, folosirea acesteia va fi absolut gratuită |
| Scop părinte | Implementarea de noi funcționalități |
| Sub-scopuri | Nu există |
| Dependențe de alte scopuri | Nu există |

1. **Creșterea calității lucrărilor academice pe baza recenziilor și aprofundarea informației**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | CCL |
| Nume | Creșterea calității lucrărilor academice pe baza recenziilor și aprofundarea informației |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Beneficiarii (Cadrul didactic) |
| Nivel | Sistem |
| Descrierea scopului | Este necesară și obligatorie evaluarea cât mai corectă pentru a atinge cele mai exigente standard |
| Scop părinte | Implementarea unui sistem de gestionare a lucrărilor științifice pentru evaluare, îmbunătățire și publicare/susținere |
| Sub-scopuri | Abordarea conform normelor impuse și lansarea unei idei teoretice care să captiveze receptorul |
| Dependențe de alte scopuri | Nu există |

1. **Abordarea conform normelor impuse**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | ACNI |
| Nume | Abordarea conform normelor impuse |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Autorul lucrării științifice |
| Nivel | Sistem |
| Descrierea scopului | Este necesară și obligatorie evaluarea cât mai corectă pentru a atinge cele mai exigente standard |
| Scop părinte | Creșterea calității lucrărilor academice pe baza recenziilor și aprofundarea informației |
| Sub-scopuri | Nu există |
| Dependențe de alte scopuri | Nu există |

1. **Lansarea unei idei teoretice**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | LIT |
| Nume | Lansarea unei idei teoretice |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Autorul lucrării științifice |
| Nivel | Sistem |
| Descrierea scopului | Ideea principală, alături de ideile secundare, concluzie trebuie să aibă un singur fir narativ, să fie corect ilustrate în lucrare, astfel încât să captiveze atenția receptorului |
| Scop părinte | Creșterea calității lucrărilor academice pe baza recenziilor și aprofundarea informației |
| Sub-scopuri | Nu există |
| Dependențe de alte scopuri | Nu există |
|  |  |

1. **Încurajarea originalității și exprimarea autorului**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | IOEA |
| Nume | Încurajarea originalității și exprimarea autorului |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Autorul lucrării științifice |
| Nivel | Sistem |
| Descrierea scopului | Se încurajează originalitatea și libera exprimare, fără plagiat, pentru creații subiective la nivel formal |
| Scop părinte | Implementarea unui sistem de gestionare a lucrărilor științifice pentru evaluare, îmbunătățire și publicare/susținere |
| Sub-scopuri | Folosirea resurselor intelectuale și nuanțarea, și interpretarea într-o manieră corespunzătoare |
| Dependențe de alte scopuri | Nu există |

1. **Folosirea resurselor intelectuale**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | FRI |
| Nume | Folosirea resurselor intelectuale |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Autorul lucrării științifice |
| Nivel | Sistem |
| Descrierea scopului | Este încurajată folosirea tuturor cunoștințelor spre scrierea și/sau susținerea lucrării științifice, pe tema abordată |
| Scop părinte | Încurajarea originalității și exprimarea autorului |
| Sub-scopuri | Nu există |
| Dependențe de alte scopuri | Nu există |

1. **Interpretarea într-o manieră corespunzătoare**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | IMC |
| Nume | Interpretarea într-o manieră corespunzătoare |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Mediu spre ridicată |
| Criticitate | Mediu spre ridicată |
| Sursă | Autorul lucrării științifice |
| Nivel | Sistem |
| Descrierea scopului |  |
| Scop părinte | Încurajarea originalității și exprimarea autorului |
| Sub-scopuri | Nu există |
| Dependențe de alte scopuri | Nu există |

1. **Implicarea recenzorilor**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | IR |
| Nume | Implicarea recenzorilor |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Ridicată |
| Criticitate | Ridicată |
| Sursă | Autorul lucrării științifice |
| Nivel | Sistem |
| Descrierea scopului | Este foarte importantă implicarea rezenzorilor pentru o modificare/îmbunătățire a lucrării științifice pentru un rezultat perfecționat |
| Scop părinte | Implementarea unui sistem de gestionare a lucrărilor științifice pentru evaluare, îmbunătățire și publicare/susținere |
| Sub-scopuri | Îmbunătățirea conținutului și aducerea în plus de recomandări |
| Dependențe de alte scopuri | Nu există |

**Îmbunătățirea conținutului și aducerea în plus de recomandări**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | ICAR |
| Nume | Îmbunătățirea conținutului și aducerea în plus de recomandări |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Mediu spre ridicată |
| Criticitate | Mediu spre ridicată |
| Sursă | Autorul lucrării științifice |
| Nivel | Sistem |
| Descrierea scopului | Cu ajutorul recenzorilor se poate îmbunătăți lucrarea științifică acolo unde este nevoie, pentru un rezultat apreciabil, valabil |
| Scop părinte | Implicarea recenzorilor |
| Sub-scopuri | Nu există |
| Dependențe de alte scopuri | Nu există |

1. **Evaluarea conținutului**

|  |  |
| --- | --- |
| **Secțiune** | **Conținut** |
| Identificator | EC |
| Nume | Evaluarea conținutului |
| Autor |  |
| Versiune | V 1.0 |
| Prioritate | Mediu spre ridicată |
| Criticitate | Mediu spre ridicată |
| Sursă | Autorul lucrării științifice |
| Nivel | Sistem |
| Descrierea scopului | Recenzorii evaluează lucrările științifice, oferind calificative |
| Scop părinte | Implicarea recenzorilor |
| Sub-scopuri | Nu există |
| Dependențe de alte scopuri | Nu există |

# 2. Cerințe de **sistem**

## 2.1 Surse de cerințe

**Cerințe funcționale**

Cerințele care sunt legate de aspectul funcțional apar în această categorie. Acestea definesc funcțiile și funcționalitatea, în și din sistemul *software*.

Utilizatorii sunt grupați pe trei categorii

* Chair
* Evaluator
* Autor

Chair-ul va decide cine va evalua o anumită lucrare a autorului, pe baza unei liste a utilizatorilor pe urma căreia va trimite o invitație prin *e-mail*.

Editorul va putea recenza o lucrare științifică și va decide dacă aceasta este potrivită sau nu. Decizia sa va putea fi monitorizată în baza de date.

Autorul este cel care va scrie lucrarea în ideea susținerii/publicării.

Pe lângă utilizatori, este si administratorul care se ocupă de partea de conținut și tot ce ține de baza de date, incluzând funcțiile CRUD (create, reade, update, delete).

**Cerințe non-funcționale**

Cerințele care nu legate de aspectul funcțional al software-ului intră în această categorie. Acestea sunt caracteristicile implicite sau așteptate, pe care utilizatorii le presupun.

Cerințe obligatorii

* Securitatea datelor
* Înregistrare/autentificare
* Accesibilitate

**Cerințe pentru interfața utilizatorului**

* Ușor de folosit
* Răspuns rapid
* Interfață simplă dar consistentă
* Gestionare eficientă a erorilor operaționale

## 2.2 Elicitația cerințelor

Părțile implicate în procesul de gestiune a lucrărilor științifice sunt cadrele didactice, în vederea evaluării.

Un utilizator poate fi:

* *Chair*
* Evaluator
* Autor

Principalul beneficiu adus de această aplicație are rolul de a îmbunătăți lucrările științifice în urma evaluărilor conferite din partea recenzorilor. Prin prisma utilizării acestui sistem de gestiune, autorii se pot dezvolta din punct de vedere personal și profesional. Prin dezvoltare se face referire la exprimarea și originalitatea autorilor, care vor fi dobândite în urma utilizării acestui sistem.

## 2.3 Documentarea cerințelor

Înaintea formulării cerințelor este nevoie de a analiza toate datele din “Elicitația cerințelor”. Analiza datelor presupune determinarea și atingerea scopului final. Se identifică problema clientului și se reprezintă cât mai accesibil cu putință de către dezvoltator.

Se stabilesc următoarele niveluri ale cerințelor:

|  |  |  |
| --- | --- | --- |
| **Cerințe de business** | **Cerințe utilizator** | **Cerințe funcționale** |
| Se relatează obiectele de nivel înalt ale clientului. În cazul de față, clienții sunt reprezentați de către cadrul didactic, iar obiectivul principal este implementarea unui sistem de gestiune al lucrărilor științifice. | Acest nivel dezvăluie funcționalitățile sistemului software, pe care utilizatorul le va putea folosi pentru a-și îndeplini obiectivele. | Sunt reprezentate cerințele esențiale care trebuie implementate astfel încât activitățile utilizatorul să decurgă conform așteptărilor acestuia.  S-au dedus următoarele:   * Chair-ul este cel care va putea concepe o conferință, prin urmare va putea alege evaluatorii și de asemenea, cel care va avea puterea de decizie finală în privința unei lucrări științifice. * Editorul este utilizatorul care poate fi ales în speranța evaluării unei lucrări științifice. * Autorul este persoana care va putea scrie o lucrare științifică, în ideea avansării profesionale, prin publicarea/susținere. |

### 

### 2.3.1 Procese

Pe partea de administrator, acesta se ocupă de prelucrarea *site*-ului, și anume:

* Accesarea bazei de date
* Manipularea bazei de date ( CRUD )



Figura 4 - Diagrama de activitate (partea de administrator)

În cadrul aplicației, utilizatorii (exclus poziția de chair) pot desfășura mai multe activități cum ar fi:

* Înregistrare și/sau autentificare
* Vizualizare site
* Publicare lucrare științifică
* Evaluare lucrare științifică
* Comunicare decizie

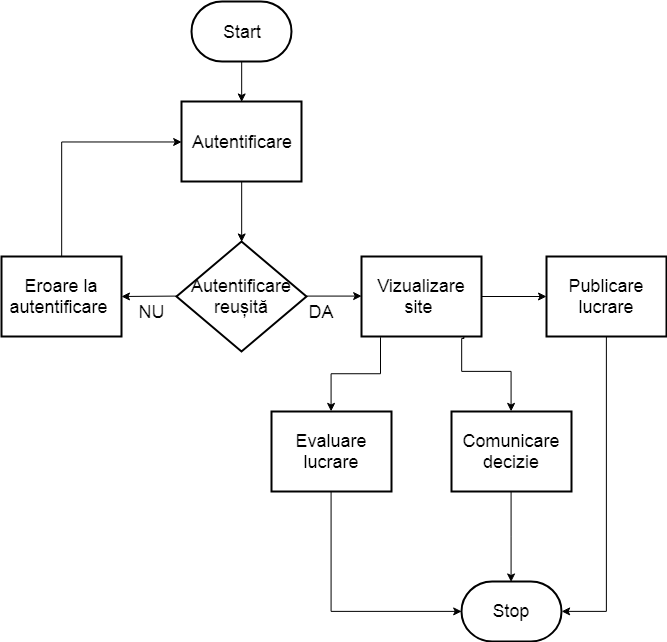


Figura 5 - Diagrama de activitate (partea de utilizator)

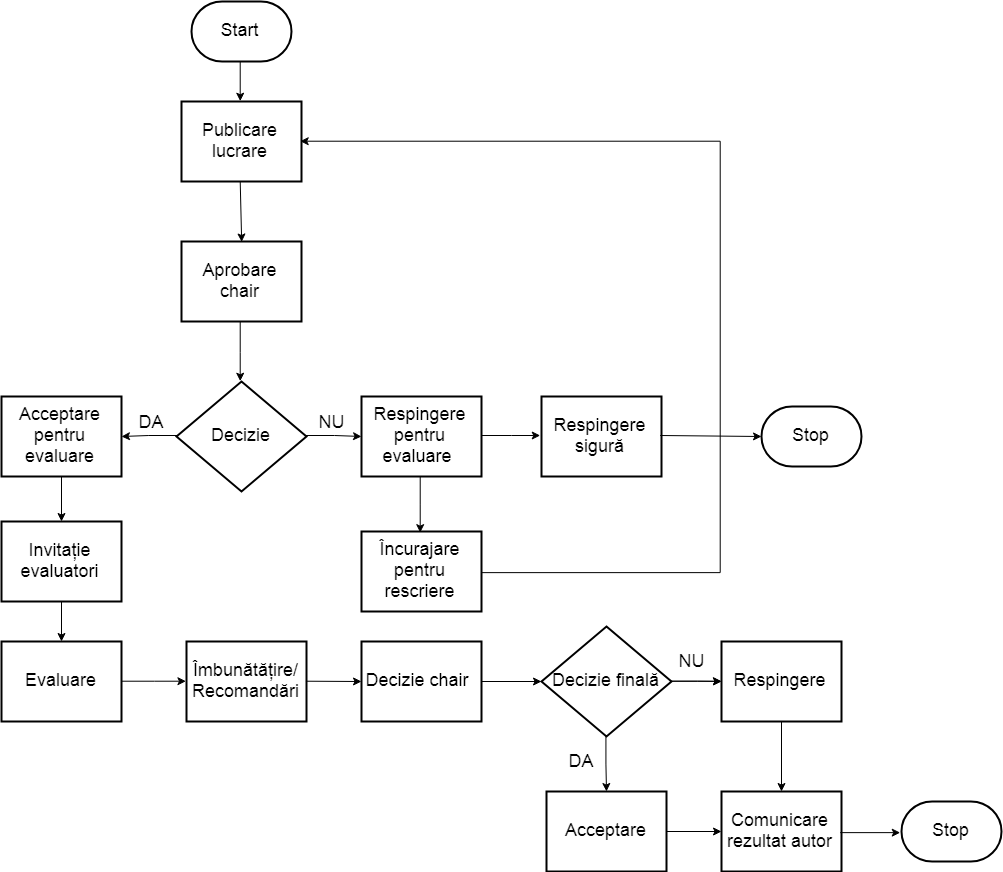


Figura 6 - Diagrama de activitate (proces peer review)

### 2.3.2 Cazuri de utilizare

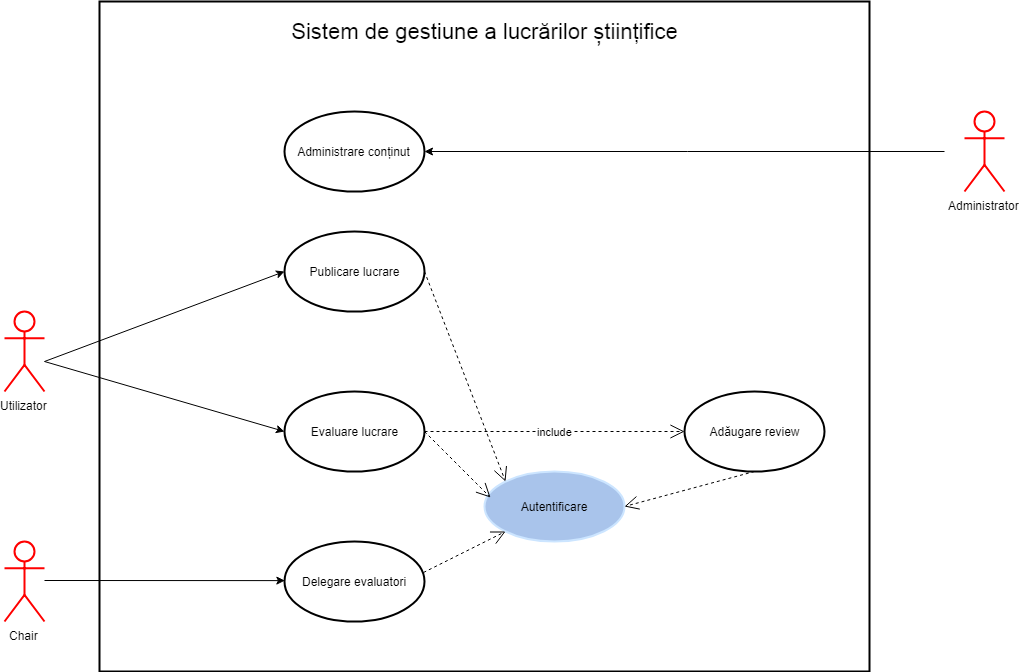


Figura 7 - Diagrama Use-Case

**Autentificare**

Diagrama “*Use-Case*” descrie felul în care utilizatorul se înregistrează în cadrul procesului de *peer review.* Toate funcționalitățile pot fi folosite doar dacă utilizatorii au cont (autentificarea se face prin adresă de *e-mail* și parolă).

**Actori**

**Administratorul** este persoana care efectuează schimbări de conținut cum ar fi adăugare, ștergere, modificare și tot ce reprezintă administrarea conținutului *site*-ului.

**Utilizatorul** poate desfășura următoarele acțiuni : să publice lucrări știițifice sau să evalueze lucrările altor autori.

***Chair*-ul** se ocupă de alegerea evaluatorilor pentru a recenza lucrările autorilor.

**“*Used*” Use Cases (Cazuri de utilizare “folosite”)**

* Publicare lucrare
* Delegare evaluatori
* Adăugare *review*/recenzie/recomandări

Toate funcționalitățile pot fi folosite doar în urma autentificării.

### 2.3.3 Brainstorming

Ca și metoda de elicitație, a fost potrivit *brainstorming*-ul, ca și o discuție cu părțile interesate, pentru a reda necesitățile acestora, prin combinarea de idei și dezvoltarea lor. Astfel a fost creată baza de date, cu un număr de nouă tabele.

1. *Users* – ilustrează toți utilizatorii
2. *Conferences* – această tabelă reprezintă lista cu toate conferințele disponibile

Un utilizator poate vizualiza conferințele la care este *chair*, *PCmember* sau invitat.

Doar *chair*-ul poate crea o conferință, să o șteargă ( doar dacă aceasta este fără membrii deja invitați) sau să modifice datele. Un *PCmember* poate doar vizualiza detalii despre o anumită conferință.

1. *PCmembers* – tabela indică toti utilizatorii care iau parte la procesul care ține de evaluarea lucrărilor științifice.

Aceștia primesc invitație de la *chair*, pentru a evalua lucrările autorilor.

1. *Papers* – tabela “*Papers*” indică toate informațiile despre lucrările științifice
2. *PaperAssignments* - acest tabel este un tabel intermediar între tabelele *PCmembers* și *Papers*
3. *Subreviewers* - tabelul cuprinde informația propriu-zisă a procesului de selectare și delegare a evaluatorilor, inclusive termenele limită
4. S*ubreviews* - acest tabel cuprinde evaluările secundare – adică acele evaluări realizate de evaluator – cu mențiunea că evaluarea finală îi revine *chair*-ului. Acest tabel se referă și la domeniul de specializare al fiecărui autor, deoarece există și posibilitatea ca în lipsa unor evaluatori adecvați să fie selectați și alții care poate nu sunt atât de specializați în domeniul pentru care se cere evaluarea. Acest lucru influențează în mare măsură evaluarea finală care va fi făcută de *chair*
5. *Reviews* - ilustrează recenziile realizate de evaluatorii desemnați de către *chair* caresunt principale, deoarece reprezintă baza în îmbunătățirea lucrării, dar în același timp secundare, deoarece decizia finală ii revine *chair*-ului care formulează o decizie pe baza a ceea ce s-a evaluat.
6. *Authors* – în această tabelă sunt stocate datele despre autorii lucrărilor științifice

### Interviu

A fost nevoie de detalierea și de înțelegerea cât mai bine a necesităților grupului țintă, prin urmare, s-a adoptat metoda interviului. S-au exprimat subiectiv punctele de vedere în urma interogării, fiind adresate întrebari mai generale, urmate de cele riguroase și rafinate.

În urma interviului, s-au stabilit următoarele :

1. *Chair*-ul creează conferințele, este singurul care le poate șterge sau modifica, alege evaluatorii, decide verdictul final de acceptare sau refuzare a lucrării științifice.
2. *PCmembers* primesc invitație prin *e-mail* de la *chair*, verifică și evaluează lucrarea științifică. Pot fi aleși/invitați oricâți, fără a se ține cont de numărul acestora. Se va evalua lucrarea conform standardelor impuse în momentul de față, prin adăugarea recomandărilor, sugestiilor acolo unde este nevoie.

Va fi nevoie de părerile sincere ale evaluatorilor astfel încât lucrarea științifică să fie mai mult decât potrivită pentru nivelul dorit.

1. Autorii vor preda lucrarea științifică în vederea publicării și a susținerii, după etapa de evaluare de către recenzori. Vor trebui să conceapă o lucrare la standardele propuse, între termenii subiectului ai lucrării științifice.

Conceptul în sine a sistemului de peer review este apreciat de grupul țintă, în special de autorii care doresc să fie remarcați și percepuți de persoanele mai specializate. Tot în urma interviului s-a decis utilizarea unei aplicații cu o interfață cât mai simplă, cu bun gust și ușor de folosit.

# 3.Model de dezvoltare

Pentru sistemul de gestiune al lucrărilor științifice, “*Get your paper published*”, s-a adoptat modelul de dezvoltare Agile de tip spirală.

Modelul spiralat este similar cu dezvoltarea incrementală a unui sistem, cu un accent mai mare pus pe analiza riscurilor. Pe baza modelelor de risc unice ale unui anumit proiect, modelul spiralat ghidează o echipă pentru a adopta elemente ale unuia sau mai multor modele de proces, cum ar fi prototipuri incrementale.

Astfel, a fost creată următoarea diagramă, cu următoarele menționări:

Partea de început subliniază colectarea cerințelor cadrului didactic și transpunerea acestora în obiective prin analiză. Se menționează cea mai importantă, și anume, implementarea unui sistem de gestiune al lucrărilor științifice.

Faza de identificare a riscurilor este exprimată de estimarea și respectarea fezabilității tehnice, mai precis, ce probleme s-au întampinat de-a lungul dezvoltării.

Al treila cadran dezvăluie implementarea și validarea la următorul nivel al produsului după rezolvarea riscurilor identificate.

Ultima fază ilustrează analiza rezultatelor cu clientul până în prezent și se planifică următoarea iterație în jurul spiralei.

Modelul în spirală dezvăluie o abordare sistematică, integrată într-un cadru iterativ, lucru care duce la siguranța că nu există probleme la produsul *soft*.

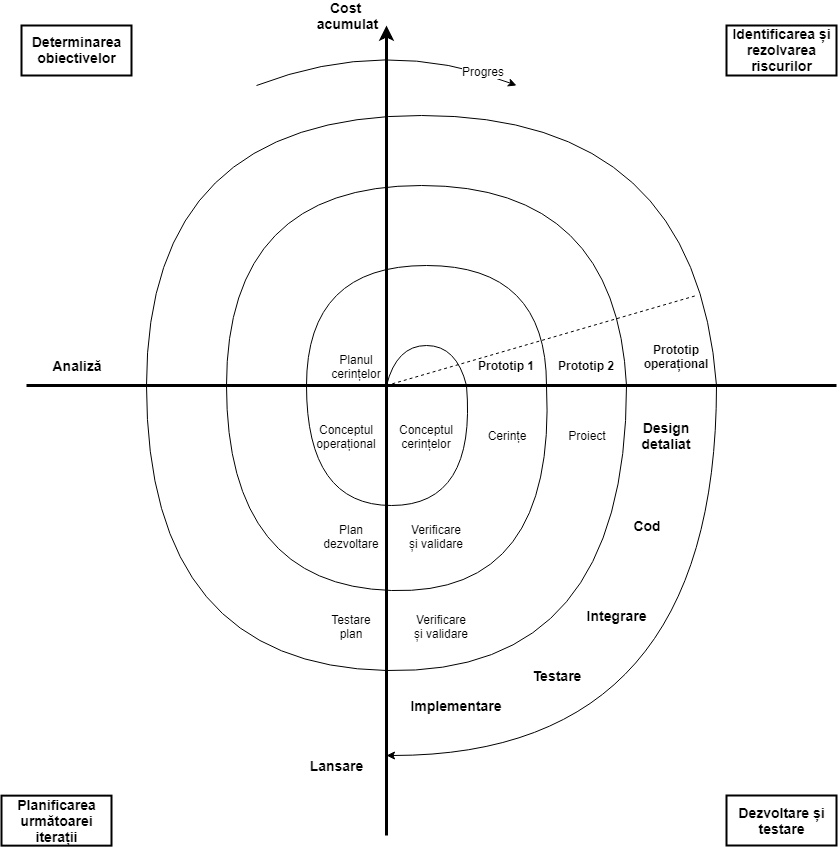


Figura 8 – Modelul de dezvoltare Agile în spirală

# Partea II – Proiectarea Sistemului Informatic

# 4. Proiectarea Logică

“*Get your paper published*” este reprezentat printr-un sistem de baze centralizat, datele fiind stocate pe o singură stație. Într-un asemenea sistem, există un server și elementele care solicită servicii, clienții. Cu ajutorul unui sistem de baze de date centralizat, se realizează micșorarea redundanței, se evită inconsistența datelor, fapte care elimină neplăcerile cauzate de multiplicări. Similar, este folosită partajarea datelor, fiind posibil ca mai mulți utilizatori să acceseze baza de date în același timp. Integritatea va fi respectată cu ajutorul corectitudinii datelor încărcate și manipulate astfel încât să se respecte restricțiile. Independența datelor va permite o organizare a acestora care să fie transparentă pentru utilizatori, sistemul bazei de date va putea fi modificat la orice nivel fără a afecta nivelurile superioare.

**Fluxul evenimentelor**

1. Autentificarea utilizatorului pe *website*

2. Aplicația va afișa pagina cu formularul de autentificare

3. Sistemul va verifica informația introdusă de utilizator

4. Sistemul va permite accesul utilizatorului pe site

5. Se va afișa pagina principală

6. Utilizatorul va decide acțiunile pe care dorește să le realizeze

## 

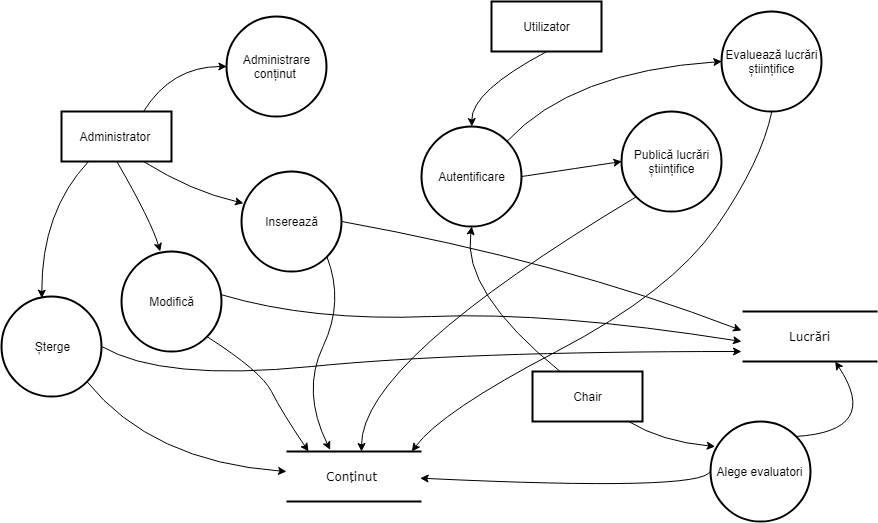


Figura 9 – Diagrama flux de date

## 4.1 Arhitectura Sistemului

Sistemul de gestiune a lucrărilor științifice, *„Get your paper published”*, se bazează pe arhitectura *server-side*. Aceasta este o tehnică folosită în dezvoltarea *web* care implică utilizarea de *script*-uri pe un *server web,* care produc un răspuns personalizat pentru solicitarea fiecărui utilizator (client) pe *site*. *Server-side* este adesea folosit pentru a oferi o interfață personalizată pentru utilizator. Aceste *script*-uri pot asambla caracteristicile clientului pentru a putea personaliza răspunsul pe baza acelor caracteristici, a cerințelor utilizatorului, a drepturilor de acces și așa mai departe.

Arhitectura se folosește în aplicațiile *web*, indiferent de limbajul de programare. Programarea *server-side* este mapată pe ASP.NET în C#, concepută pentru dezvoltarea *web* pentru a produce pagini dinamice. Procesează *input*-ul care vine de la utilizator, afișează pagini, structurează paginile *web*, dar și interacționează cu spațiul de stocare permanent ( SQL, *server* local).

Sunt create pagini *web* interactive, cu conținut specific dinamic. Acest lucru se întămplă datorită limbajului JavaScript și al sintaxelor HTML cu Razor (este o caracteristică a *framework*-ului MVC), CSS.

*Framework*-ul MVC (*Model, View, Controller*) este destinat să returneze pagini .NET, făcând o diferențiere/separare între straturi:

* *Models* – nivelul care este responsabil de ținerea datelor
* *Views* – nivelul care ilustrează utilizatorului datele regăsite în model
* *Controllers* – nivelul unde se regăsește codul propriu-zis, care implementează interacțiunile dintre *Model* si *View*

MVC este foarte popular deoarece se izolează logica aplicației, făcând abstracție pentru a reda exact părțile esențiale. Pe scurt, *controller*-ul primește toate cererile, mai apoi lucrează cu modelul pentru a pregăti datele necesare care trebuie să fie trimise către *view*, pentru a fi vizualizate de utilizator.

S-a folosit ultima versiune de MVC, și anume 5.2.3.

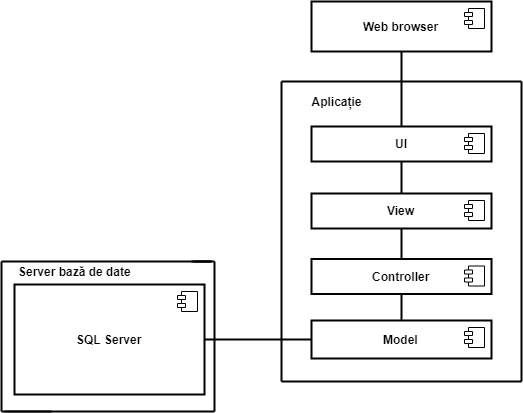


Figura 9 – Diagrama deployment

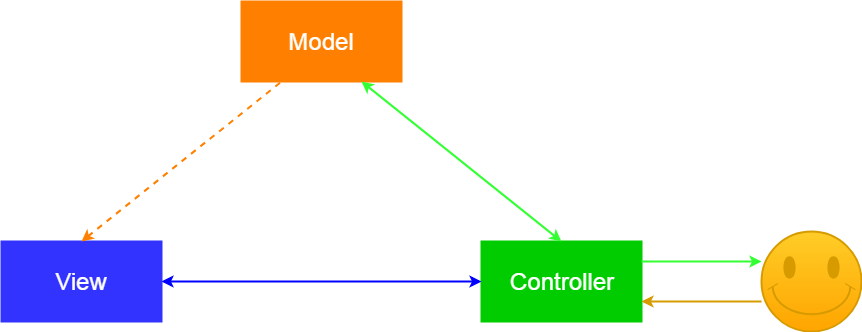


Figura 10 – Triunghiul MVC

Cum s-a mai menționat, arhitectura MVC separă logica aplicației de restul interfeței, prin cele trei părți. *Controller*-ul primește *input* de la *user* și preia din model datele necesare, care mai apoi sunt transmise către *view*, pentru a fi afișate utilizatorului.

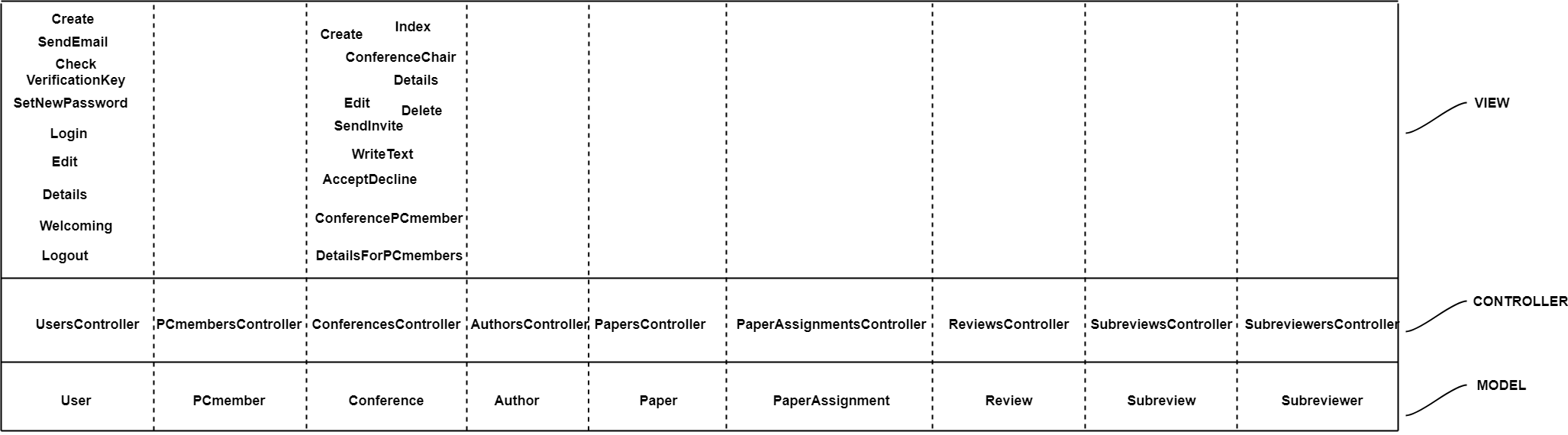
**

Figura 11 – Diagrama MVC detaliată

## 4.2 Baza Informaţională

Datele de *input* și *output* sunt redate în baza de date, având ca și sursă locală *server*-ul SQL încorporat în Visual Studio. Baza de date a aplicației este în forma normal a treia (3NF). Pentru a trece la forma normală a treia, trebuie eliminate dependenţele tranzitive. Eliminarea este realizată prin ştergerea câmpurilor care sunt dependente tranzitiv de cheia primară din relaţia iniţială şi crearea unei noi relaţii cu aceste atribute şi determinantul lor.

Pentru folosirea bazei de date, s-a adăugat *Entity Framework (EF)*, tehnologia de acces a datelor recomandată de Microsoft pentru aplicații noi. Framework-ul este un *mapper* de tip entitate-relație care permite programatorilor să lucreze cu date relaționale folosind obiecte specifice. Ușurează munca dezvoltatorilor prin cele trei moduri de creare ale unei bazei de date : *Model First, Code First, Database First*.

S-a folosit *Model First*, deoarece oferă un mod ușor de a crea baza de date. Se desenează în fișierul cu suspenia *.edmx* baza de date, iar mai apoi se generează clasele, fiind incluse atributele și relațiile.

Figura 10, ilustrată mai jos, reprezintă baza de date creată cu ajutorul *Entity Framework*, cu *Model First*.

La *Entity Framework* s-a folosit ultima versiune, și anume 6.1.3.

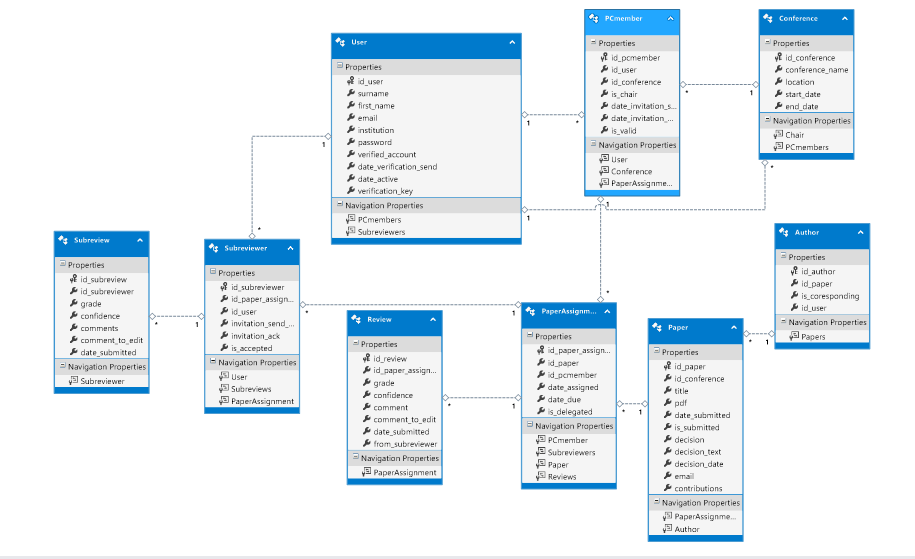


Figura 10 – Diagrama entitate-relație ASP.NET

# 5. Proiectarea Tehnică

## 5.1 Proiectarea interfeţei cu utilizatorul

În prima parte, accesarea *site*-ului, se deschide pagina de “Home”, împărțită în *navigation-bar* și partea de *body-content*, care include un *plugin* de tip *carousel*. Dedesubt se află o porțiune de tip *grid view*. Pagina este simplă, cu bun gust și ilustrează strict esențialul despre temă, dar și punctul de început pentru a se putea continua folosirea *website*-ului, invitând utilizatorul spre utilizare.

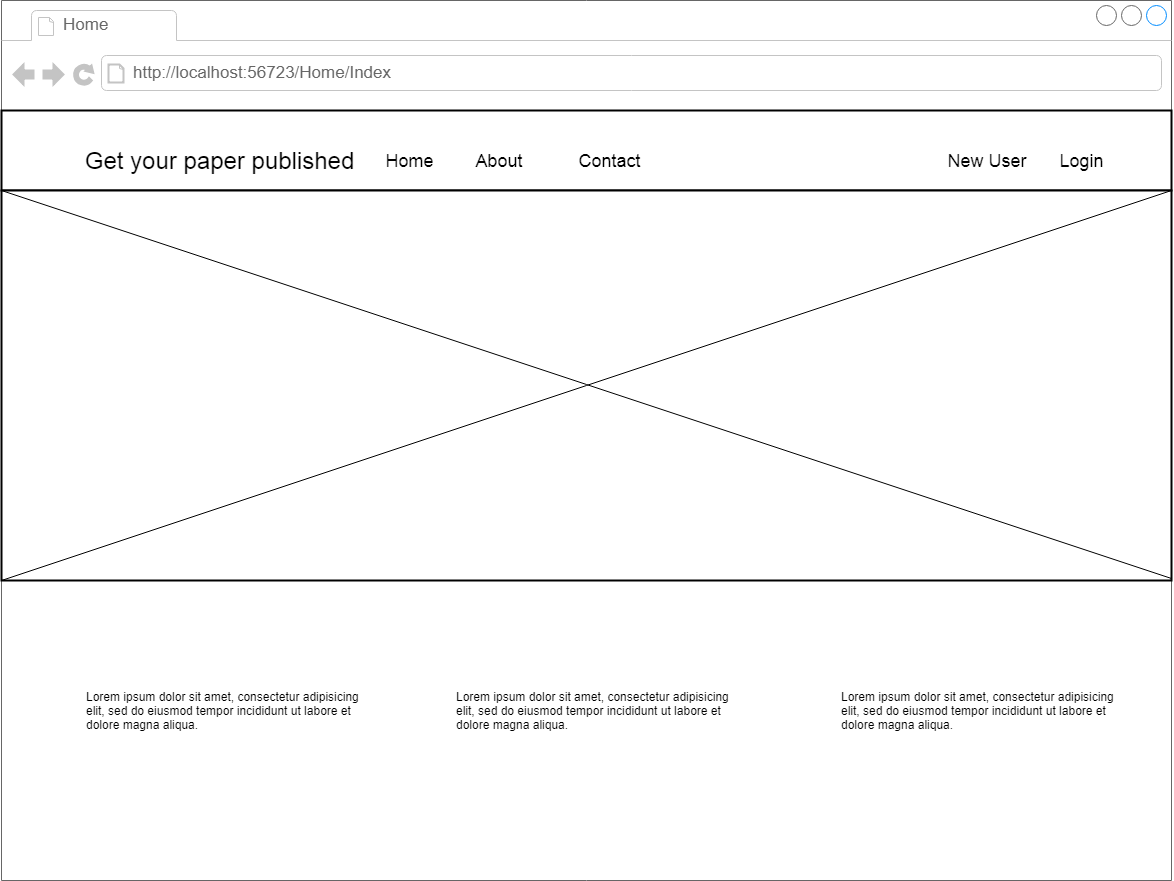


Figura 11 – Pagina “Home”

Calea *“~/NewUser/Create*” accesează pagina unde utilizatorul își creează cont, prin introducerea datelor în căsuțele de *input*. În urma apăsării *click* pe butonul “*Create*”, utilizatorul va primi *e-mail,* cu un *link* unde va putea să-și introducă parola.

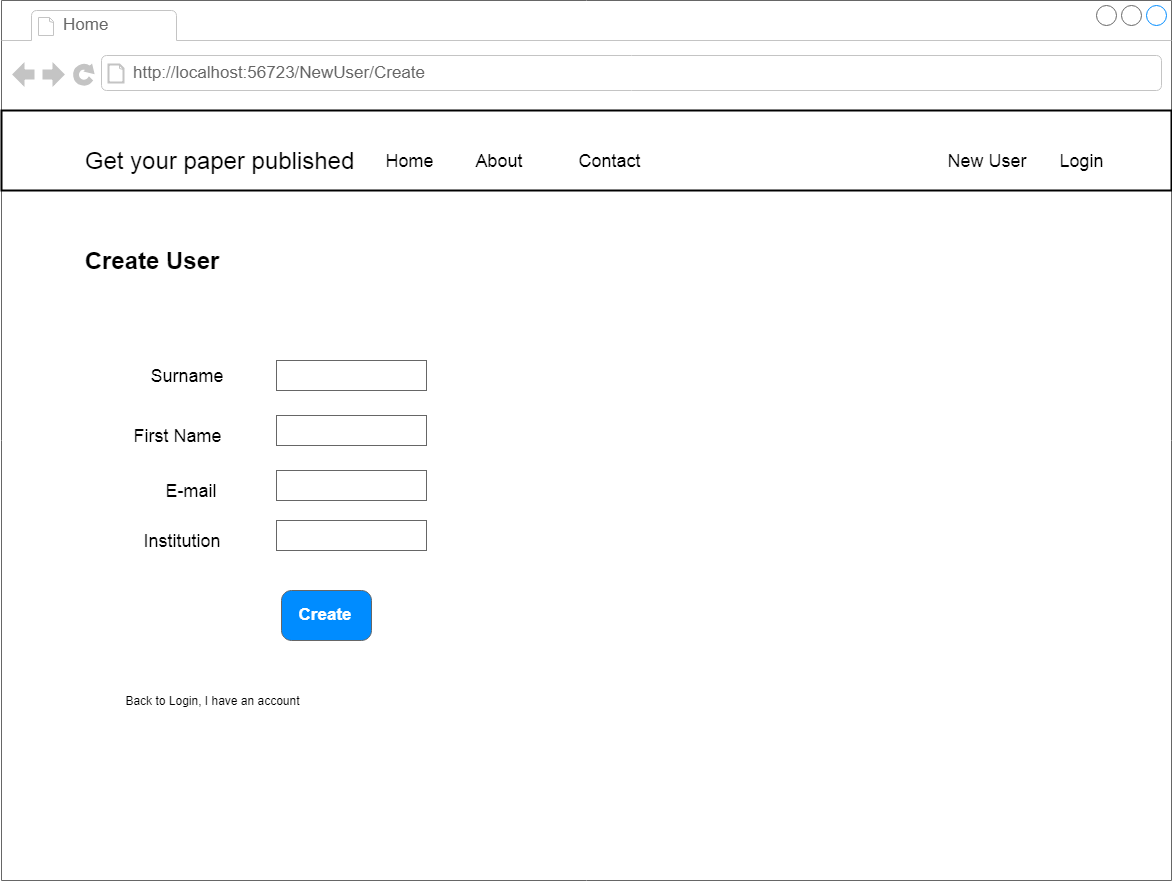


Figura 12 – Formular de înregistrare

Pagina “*SetNewPassword*” este reprezentată de două căsuțe de *input*, cu etichetele aferente, alături de butonul de tip *submit*, denumit „*Confirm*”. Pe aceasta pagină, utilizatorul își creează o parolă pentru contul său, pentru a se putea autentifica.

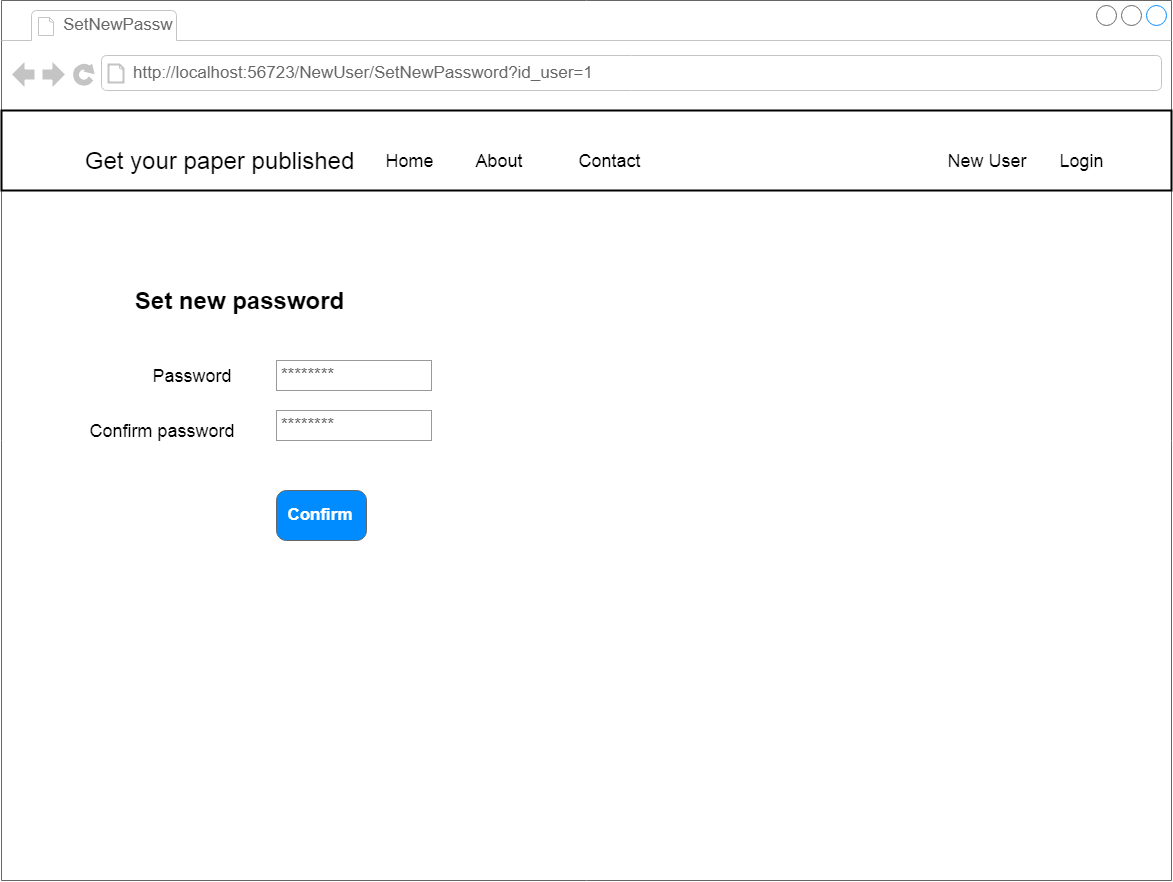


Figura 13 – Formular pentru introducerea parolei

După confirmarea parolei, utilizatorul este redirecționat către pagina de autentificare. Aici introduce adresa de *e-mail* și parola proaspăt creată, fiind mai apoi trimis, din nou, la pagina actualizată de “*Home*”.

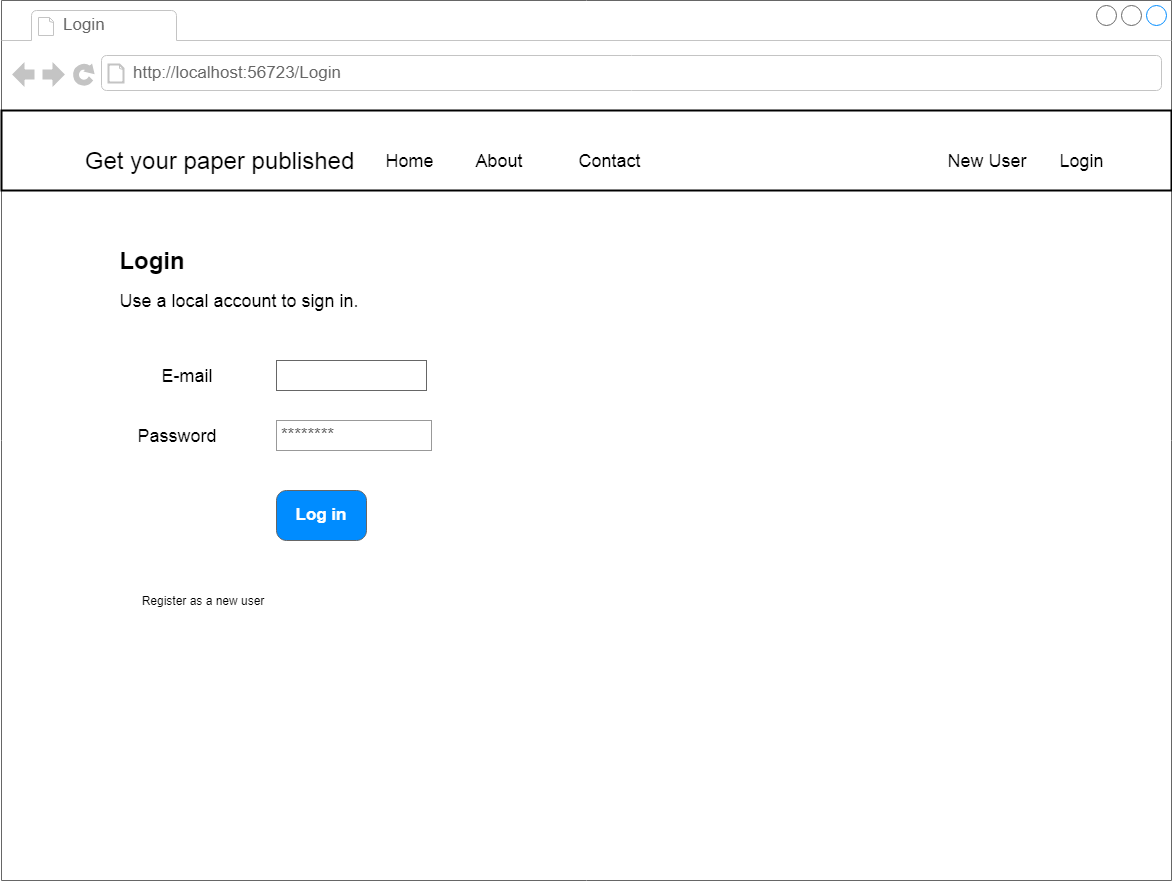


Figura 14 – Formular de autentificare

## 5.2 Procese şi Algoritmi

**Înregistrare utilizator**

Introducere date (

nume,

prenume,

adresă de email,

instituție)

1. Dacă câmpurile au fost completate

2. Trimite date

3. Salvează date în baza de date

4. Altfel

5. Afișează eroare datorită necompletării

6. Realizează aceste operațiuni

7. Cât timp datele sunt corecte și complete

8. Dacă datele sunt corecte și complete

9. Trimite e-mail pentru adăugare parolă

10. Altfel

11. Afișează eroare

12. Dacă a fost primit e-mail

13. Click pe link

14. Adăugare parolă

15. Adăugare confirmare parolă

16. Dacă parolele sunt identice

17. Mesaj pentru cont creat

18. Redirecționare pagină pentru autentificare

19. Altfel

20. Mesaj eroare datorită parole diferite

21. Sfârșit

**Autentificare utilizator**

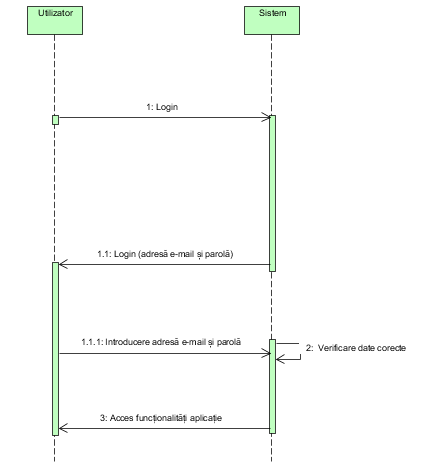


Figura 15 – Diagrama de secvență

**Etape (**după confirmarea noii parole):

1. Utilizatorul este redirecționat spre pagina de login
2. Își introduce datele
3. Dacă acestea nu sunt corecte, le poate reintroduce
4. Odată introduse datele corecte, utilizatorul are acces la funcționalitățile aplicației

## 5.3 Structura fizică a datelor

Este ilustrată baza de date a aplicației prin intermediul tabelelor și relațiilor dintre acestea, bazate pe chei primare și chei străine, fiind necesar ca tipul de date a cheii secundare să fie acelaşi cu cel al cheii primare. Datele vor fi introduse conform tipului de date definit.

**Tipuri de date :**

* *Integer* (valori numerice întregi)
* *Varchar* (șiruri de caractere)
* *DateTime* (dată+oră)
* *Boolean* (valoare de tip *true/false*)

În diagrama de mai jos se poate observa următorul lucru:

Există tabela *User* și tabela *Paper*, unde se poate deduce o relație de m:m, adică, mai mulți utilizatori pot avea mai multe lucrări științifice. Datorita regulilor de proiectare a bazelor de date, ar mai fi fost nevoie de un tabel de intersecție, ceea ar fi îngreunat vizualizarea. În urma sesiunii de *brainstorming*, s-a hotărât că cea mai bună soluție este adăugarea tabelului *Author*, având ca și câmp, atributul din tabela *User*, și anume, *id\_user*. Astfel, un utilizator poate fi autor, dar nu este obligatoriu.

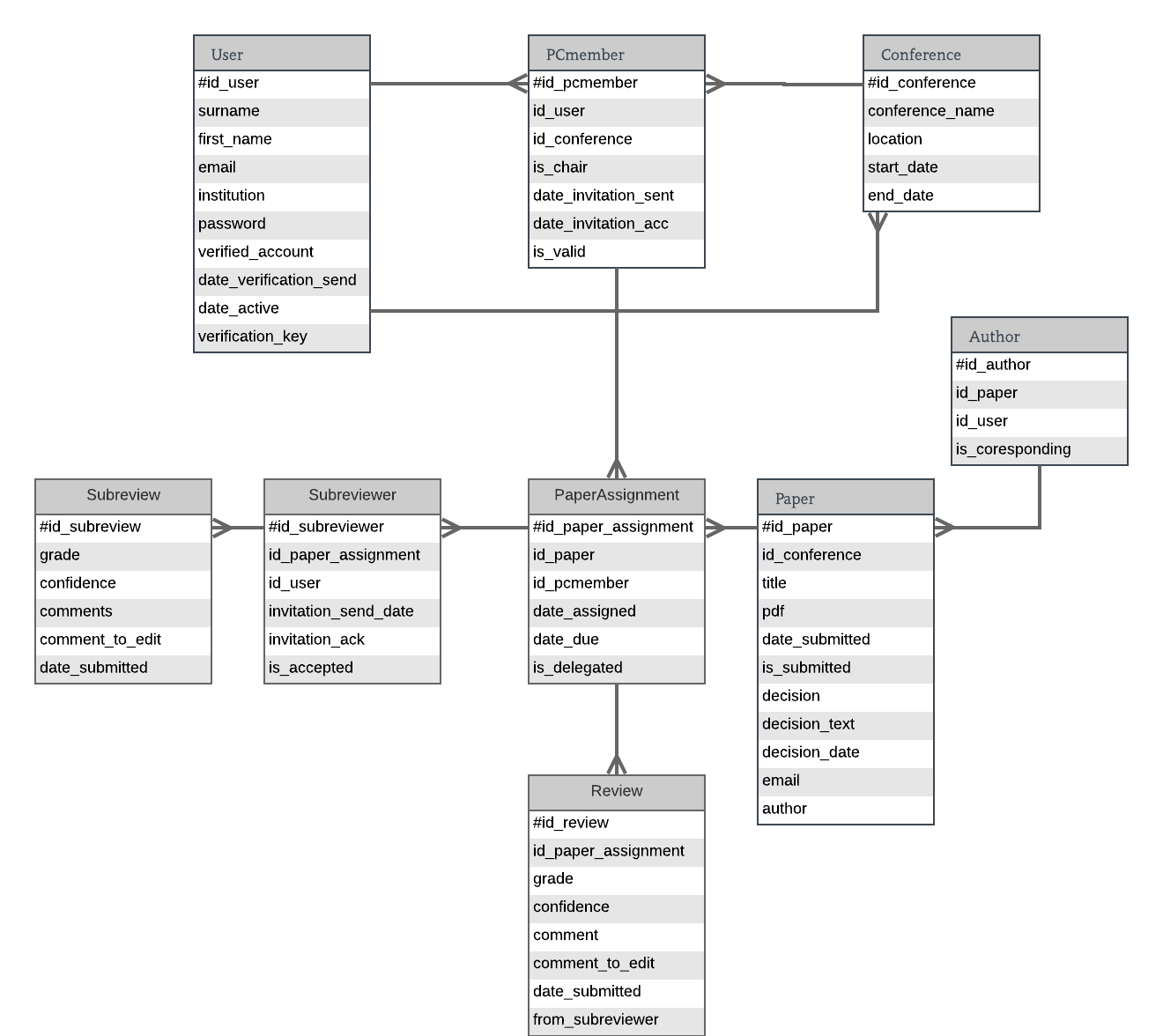


Figura 16 – Diagrama Entitate-Relație

Modelarea ERD a surprins cele mai importante tabele din baza de date, iar mai apoi s-au creat relațiile între acestea. S-au introdus doar câteva atribute, pentru a nu încărca diagrama de modelare, evidențiindu-se doar cele mai importante, inclusiv cheile primare.

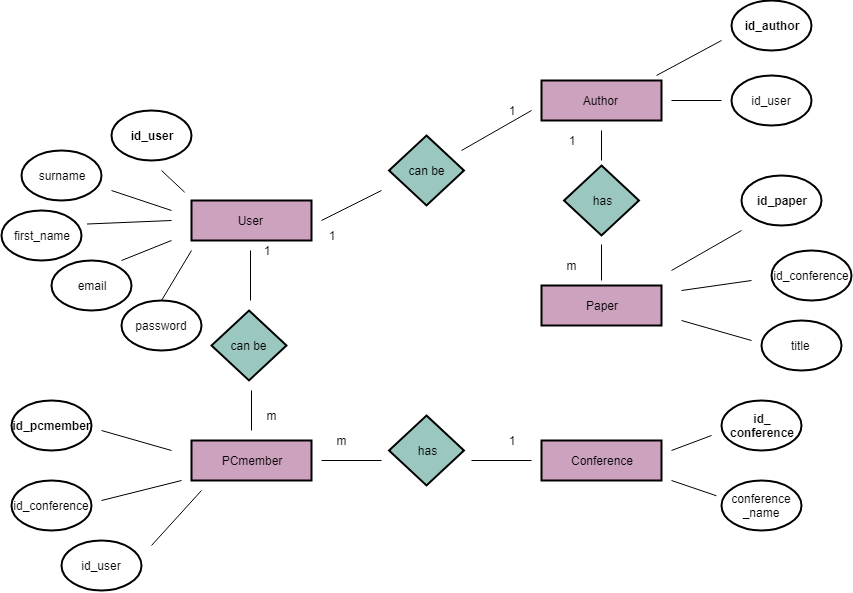
**

Figura 16 – Modelarea ERD (notația Chen)

## 5.4 Tehnologii specifice

Pentru crearea aplicației web s-au folosit câteva dintre cele mai cunoscute tehnologii bazate pe .NET, atât *back-end*, cât și pentru *front-end*.

Pe partea de *back-end*, codul rulat pe *server*, s-a folosit limbajul C#, cu ajutorul limbajului de programare ASP.NET. Acesta facilitează crearea de *site*-uri și aplicații dinamice prin utilizarea unor limbi compilate precum C#. Prin rularea pe *server*, codul poate efectua sarcini complexe, cum ar fi accesarea bazelor de date. SQL Server ajută la stocarea locală a datelor, acestea putând fi manipulate prin intermediul comenzilor de tip LINQ. Comenzile cuprind o serie de operatori care sunt utilizați pentru interogarea, filtrarea și proiectarea datelor în tabele, clase enumerabile, baze de date relaționale și XML. Conceptul LINQ tratează sursa de date ca obiect, mai degrabă decât o bază de date.

Toate conceptele enumerate mai sus, se regăsesc în substratul oferit de MVC, *Controller*. Aici sunt definite metodele pentru manipularea bazei de date, făcând legătura cu modelele și partea de *view*, unde sunt afișate rezultatele sau altfel spus, “ceea ce va vedea utilizatorul”.

Pe partea de *front-end*, paginile dinamice, ilustrate sub forma de *View*, parte a MVC, a fost folosit HTML alături de sintaxa Razor, în fișiere sub forma *.cshtml*. Se bazează în principiu pe ASP.NET și este conceput pentru crearea de aplicații *web*. Alături de HTML și Razor, s-a folosit și CSS, altă sintaxă, pentru stilizarea paginilor web dinamice, dar și limbajul JavaScript.

Aplicațiile ASP.NET, cum s-a precizat, sunt găzduite de un *server web* și sunt accesate utilizând protocolul HTTP. În urma cererii de tip GET, urmează răspunsul cu datele cerute din partea *server*-ului în paginile *web* dinamice. La cererile de tip POST, sunt preluate datele de utilizator, prin intermediul formularelor.

# Glosar

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Termen** | **Explicație** |
| 1 | peer review | evaluarea activității științifice, academice sau profesionale a unei persoane sau a unor lucrări științifice înainte de publicare de către alte persoane care lucrează în același domeniu. |
| 2 | ASP.NET(Active Server Pages) | cadru web open-source pentru construirea de aplicații și servicii web moderne cu .NET; creează site-uri web |
| 3 | Framework | o structură conceptuală ce reprezintă o arhitectură de software care modelează relațiile generale ale entităților site-ului |
| 4 | Chair | o persoană care prezidează o întâlnire, o comisie, o consiliu, etc. |
| 5 | SQL Server | sistem de gestionare de baze de date relaționale |
| 6 | software | gamă de produse de programare, uzual formată din sistem de operare, drivere și programe de aplicație |
| 7 | CRUD | Create, Read, Update, Delete  patru funcții de bază ale stocării persistente |
| 8 | diagrama “Fishbone” | diagrama “schelet de pește”, care ilustrează relațiile dintre un anumit efect identificat și cauzele potențiale ale acestuia |
| 9 | diagrama “Use-Case” | reprezentare a interacțiunii unui utilizator cu sistemul care arată relația dintre utilizator și diferitele cazuri de utilizare în care este implicat utilizatorul |
| 10 | Agile | abordare de dezvoltare software în care se susține planificare adaptabilă, dezvoltare evolutivă, livrare timpurie, îmbunătățire continuă și încurajează răspunsul flexibil și rapid |
| 11 | MVC | Model, View, Controller – model arhitectural software care împarte o aplicație în trei părți interconectate |
| 12 | hardware | partea fizică a calculatorului sau a sistemului informatic |
| 13 | arhitectura server-side | serverul sau limbajul de back-end rulează scripturile sale înainte ca pagina să fie încărcată; server-side se folosește pentru a interacționa cu datele stocate. Server-ul va putea să trimită pagini la client și să proceseze input-ul de la utilizator. Server-side se folosește atunci când se face cerere către o pagină și când pagini sunt postate înapoi datorită server-ului. |
| 14 | front-end | partea aplicației web cu care interacționează utilizatorii |
| 15 | back-end | partea aplicației web care permite administrarea conținutului |
| 16 | CSS(Cascading Style Sheets ) | limbaj de stilizare pentru formatarea conțintului |
| 17 | Razor | sintaxă care permine să încorporați codul bazat pe server în pagini web |
| 18 | JavaScript | limbaj de programare pentru aplicații web, partea de front-end |
| 19 | model first, code first,  database first | 3 abordări diferite de reprezentare a bazei de date, folosind Entity Framework |
| 20 | navigation-bar | secțiune a unei interfețe grafice de utilizator destinată să ajute utilizatorii să acceseze informații |
| 21 | body-content | conținut principal |
| 22 | plugin carousel | componentă ca un slideshow, cu elemente ciclice |
| 23 | grid view | element grafic de control care prezintă o vizualizare tabulară a elementelor |
| 24 | submit | caracteristică pentru un buton, pentru a înainta informații dintr-un formular |
| 25 | LINQ (Language-Integrated Query) | sintaxă de interogare în C # utilizată pentru a salva și a prelua date din diferite surse. Este integrat în C#, eliminând astfel neconcordanța între limbile de programare și bazele de date, precum și oferind o singură interfață de interogare pentru diferite tipuri de surse de date |
| 26 | Script | un program sau o secvență de instrucțiuni care este interpretată sau efectuată de un alt program și nu de un procesor de calculator |
|  | relație m:m |  |
| 27 | integer |  |
| 28 | varchar |  |
| 29 | datetime |  |
| 30 | boolean |  |
| 31 | HTTP(Hypertext Transfer Protocol) | metoda cea mai des utilizată pentru accesarea informațiilor în Internet care sunt păstrate pe servere World Wide Web (WWW) |
| 32 | POST | trimiterea datelor de intrare către server (input) |
| 33 | GET | metodă când serverului i se cere o resursă |

# Bibliografie

1. **Cristian Darie, Zak Ruvalcaba, “**Build Your Own ASP.NET 2.0 Web Site Using C# & VB”
2. **David Chappell,** Understanding .NET
3. Xin Chen, **Developing application frameworks in .NET**
4. **Harold Davis, Visual C# .NET Programming**
5. Chris Hart, John Kauffman, David Sussman, Chris Ullman , **Beginning ASP.NET 2.0**
6. Steven Sanderson, **Pro ASP.NET MVC Framework**
7. **Jeffrey Palermo, Ben Scheirman, Jimmy Bogard, ASP.Net MVC in Action**
8. **Jon Galloway, Brad Wilson, K. Scott Allen, David Matson, Professional ASP.NET MVC 5**
9. **Joshi N., Programming ASP.NET MVC 5**
10. Julia Lerman, **Programming Entity Framework**
11. Tomson Brett, ASP.NET
12. Lee Naylor, ASP.Net MVC with Entity Framework and CSS
13. **Brian Driscoll, Nitin Gupta, Robert Vettor, Zeeshan Hirani, Larry Tenny,** [**Entity Framework 6 Recipes**](http://libgen.io/ads.php?md5=795F99A462E8B31568FFEAE2CEFF5C41)
14. , **Paolo Pialorsi, Marco Russo**, Introducing Microsoft LINQ
15. **Emily A. Vander Veer,** JavaScript for Dummies
16. Imar Spaanjaars, Beginning ASP.NET 4.5: in C# and VB
17. Pro ASP.NET 4 in C# 2010 (Expert's Voice in .NET), [Matthew MacDonald](https://www.amazon.com/s/ref=dp_byline_sr_book_1?ie=UTF8&text=Matthew+MacDonald&search-alias=books&field-author=Matthew+MacDonald&sort=relevancerank), [Adam Freeman](https://www.amazon.com/Adam-Freeman/e/B005DOMYNE/ref=dp_byline_cont_book_2)