**UNIVERSITATEA BABEŞ-BOLYAI**

**Facultatea de Ştiinţe Economice şi Gestiunea Afacerilor**

**Denumirea completă a programului de studii**

**Lucrare de licență**

Absolvent,

Sorana-Andrada MARIŞ

Coordonator ştiinţific,

Lect. Univ. dr. Cristian BOLOGA

**2024**

**UNIVERSITATEA BABEŞ-BOLYAI**

**Facultatea de Ştiinţe Economice şi Gestiunea Afacerilor**

**Denumirea completă a programului de studii**

**Lucrare de licență**

Analiza și proiectarea unei aplicații online pentru cabinet stomatologic

Absolvent,

Sorana-Andrada MARIŞ

Coordonator ştiinţific,

Lect. Univ. dr. Cristian BOLOGA

**2024**

CUPRINS

[I. Raport de analiză 1](#_Toc453780183)

[Introducere 2](#_Toc453780184)

[1. Identificarea și descrierea problemei 3](#_Toc453780185)

[1.1 Motivație 3](#_Toc453780186)

[1.2 Context 6](#_Toc453780187)

[2. Cerințe de sistem 10](#_Toc453780188)

[2.1 Surse de cerințe 10](#_Toc453780189)

[2.2 Elicitația cerințelor 11](#_Toc453780190)

[2.3 Formularea cerințelor 26](#_Toc453780191)

[2.3.1 Procese și activitați 26](#_Toc453780192)

[3. Model de dezvoltare 30](#_Toc453780193)

[Glosar 32](#_Toc453780194)

[II. Proiectarea sistemului informatic 33](#_Toc453780195)

[4. Proiectarea logică 34](#_Toc453780196)

[4.1 Arhitectura sistemului 35](#_Toc453780197)

[4.2 Baza informațională 37](#_Toc453780198)

[5.Proiectarea tehnică 38](#_Toc453780199)

[5.1 Procese și algoritmi 38](#_Toc453780200)

[5.2 Structura fizica a datelor 40](#_Toc453780201)

[5.3 Tehnologii specifice 41](#_Toc453780202)

[III. Testarea 42](#_Toc453780203)

[6. Testarea 43](#_Toc453780204)

[6.1 Testarea manuală 43](#_Toc453780205)

[IV. Mentenanță 49](#_Toc453780206)

[8.Operații de întreținere 50](#_Toc453780207)

[Concluzie 51](#_Toc453780208)

[Bibliografie 52](#_Toc453780209)

**Lista diagramelor**

[1. Diagrama “Fish-Bone” 5](file:///F:\PSI\Vargan_Ioana.docx#_Toc453774927)

[2. Diagrama de descompunere a obiectivelor 6](#_Toc453774928)

[3. Diagrama cazurilor de utilizare 11](#_Toc453774929)

[4. Diagrama cazurilor de utilizare pentru cazul “Administrează” 20](#_Toc453774930)

[5. Diagrama de secvențe 21](#_Toc453774931)

[6. Modelul de business 23](file:///F:\PSI\Vargan_Ioana.docx#_Toc453774932)

[7. Diagrama de activități “Vizitator” 26](file:///F:\PSI\Vargan_Ioana.docx#_Toc453774933)

[8. Diagrama de activități “Administrator” 27](file:///F:\PSI\Vargan_Ioana.docx#_Toc453774934)

[9. Modelul cascasdă 28](file:///F:\PSI\Vargan_Ioana.docx#_Toc453774935)

[10. Diagrama de flux de date 32](#_Toc453774936)

[11. Diagrama client-server 33](#_Toc453774937)

[12. Diagrama de componente 34](#_Toc453774938)

[13. Structura bazei de date 39](file:///F:\PSI\Vargan_Ioana.docx#_Toc453774939)

# **I .Raport de analiză**

# **Introducere**

În demersul de a se integra cu ușurință pe piața stomatologică, o nouă inițiativă vizează crearea unui site web care să permită programările online și să prezinte în detaliu serviciile și promoțiile oferite.

Scopul acestui proiect este de a atrage și influența deciziile potențialilor clienți, prin furnizarea unei platforme online accesibile și informative. Este crucial ca orice cabinet dentar să adopte o prezență online robustă, având în vedere că majoritatea persoanelor în căutarea serviciilor medicale se orientează către internet pentru a lua decizii informate. Un site web bine dezvoltat servește acestui scop, facilitând accesul la informații relevante despre serviciile disponibile și consolidând încrederea în cabinet.

Pentru a asigura funcționalitatea și eficiența aplicației, am propus dezvoltarea a două interfețe distincte: una pentru clienți și una pentru administrare. Interfața destinată clienților va fi concepută pentru a fi intuitivă și ușor de navigat, permițând accesul rapid la informațiile necesare și evidențiind datele de contact pentru o comunicare eficientă cu personalul cabinetului.

Pe de altă parte, interfața administrativă va oferi funcționalități complexe de tipul creare, citire, actualizare și ștergere (CRUD) asupra datelor, fiind proiectată pentru a facilita gestionarea eficientă a conținutului și a programărilor.

Aspectele tehnice ale implementării includ utilizarea limbajului C# în cadrul framework-ului ASP.NET, pentru o programare orientată obiect, și integrarea cu Entity Framework pentru conectarea la baza de date.

Aceste tehnologii vor susține crearea unei aplicații robuste, capabilă să satisfacă nevoile specifice ale unui cabinet stomatologic și să ofere o experiență de utilizare de înaltă calitate. Astfel, investiția în dezvoltarea unei astfel de platforme online devine esențială pentru orice cabinet stomatologic care aspiră să se distingă în peisajul digital actual, atrăgând mai mulți clienți și oferindu-le acestora un serviciu superior.

# **Identificarea și descrierea problemei**

"Stoma Dent" este un cabinet de medicină dentară aflat la început de drum, cu un colectiv de specialiști stomatologi dedicați și cu planuri ambițioase de extindere pe viitor. Promovarea afacerii s-a realizat până acum prin Instagram și grație aprecierilor pozitive ale clienților. Însă, evidențiază necesitatea urgentă de a adopta soluții digitale pentru o gestionare mai eficientă a pacienților și pentru a prezenta serviciile într-un mod mai atractiv. De aceea, intenționăm să dezvoltăm o aplicație specifică, menită să simplifice procesul de programare online și să faciliteze accesul la informații comprehensive despre oferta cabinetului. Aplicația va avea, de asemenea, rolul de a îmbunătăți administrarea internă a cabinetului, marcând un pas important în modernizarea modului în care interacționăm cu pacienții noștri.

## **Motivație**

În cadrul secțiunii de identificare a provocărilor, am constatat că dificultățile majore cu care se confruntă cabinetul stomatologic vizează în principal administrarea și modalitățile de prezentare ale organizației. La ora actuală, gestiunea informațiilor despre pacienți se face prin utilizarea unei agende fizice, unde sunt înregistrate detaliile clientului, serviciile utilizate și data vizitei. Acest sistem face dificilă și timp-consuming accesarea istoricului medical al pacienților, bazându-se pe data consultării pentru a găsi informațiile necesare.

Referitor la marketing, acest domeniu esențial este tratat superficial. Cabinetul are doar o pagină de Facebook pentru prezentare, care include informații de contact dar omite detalii importante precum serviciile disponibile și tarifele aferente. Astfel, medicul întâmpină dificultăți în comunicarea modificărilor de servicii, iar pacienții sunt nevoiți să apeleze la telefon pentru a obține informații despre serviciile cabinetului.

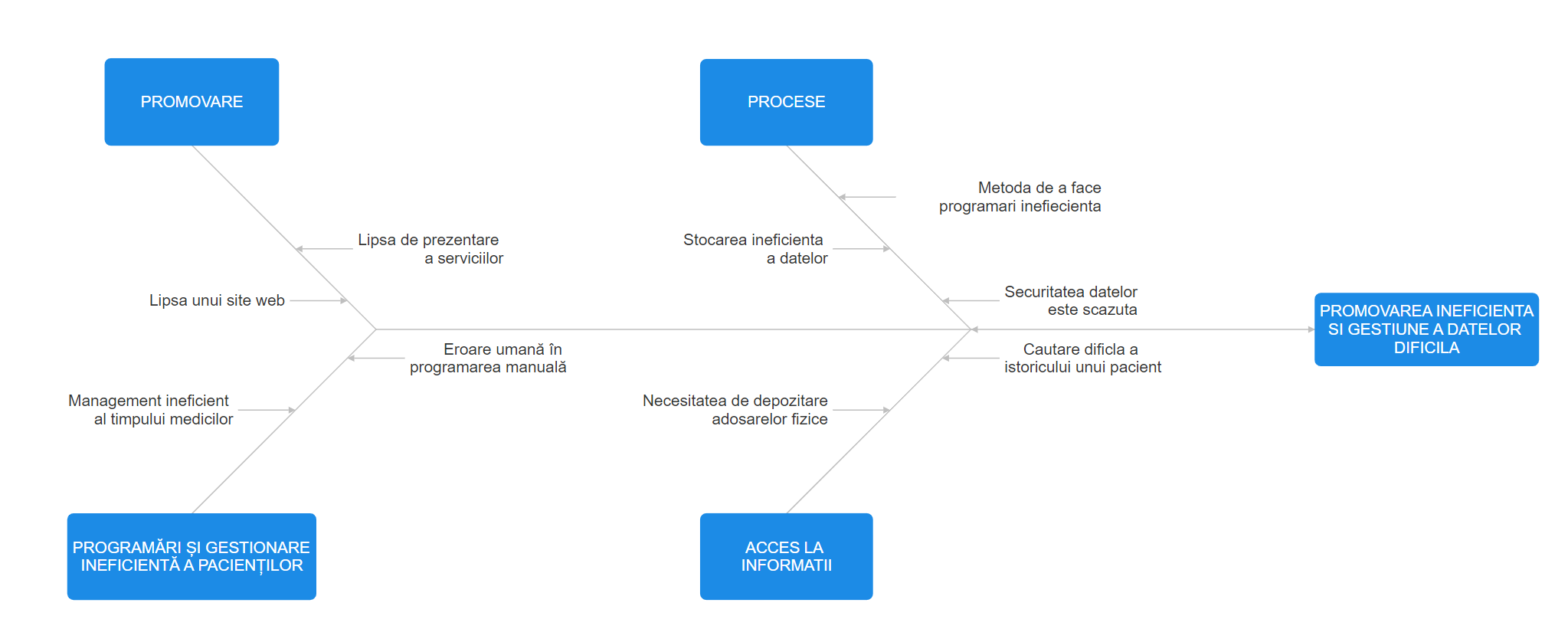
Solționarea acestei probleme presupune dezvoltarea unui site web cu o interfață amicală pentru clienți, menită să faciliteze interacțiunea dintre pacienți și medic, și să permită programări online. Având în vedere ritmul de viață accelerat al pacienților și timpul semnificativ petrecut pe internet, un site web oferă avantaje multiple atât pentru medic, care economisește timp prin automatizarea răspunsurilor la întrebări, cât și pentru pacienți, care accesează mai repede informațiile dorite.

O prezență online este crucială nu doar pentru îmbunătățirea gestionării relației cu pacienții, dar și pentru promovarea și extinderea afacerii. Un site web asigură că informațiile despre cabinet sunt disponibile non-stop, mărind vizibilitatea afacerii și încrederea în serviciile oferite. În plus, un portal online permite colectarea feedback-ului de la pacienți, esențial pentru îmbunătățirea continuă a serviciilor.

Pe lângă îmbunătățirea comunicării și promovării, obiectivul principal al introducerii soluției digitale vizează optimizarea administrativă. Se urmărește trecerea de la documentația fizică la stocarea digitală a datelor, eliminând necesitatea păstrării fișelor tipărite și asigurând un acces rapid la informații. Aceasta va facilita un flux de lucru mai eficient în cabinet și va crește securitatea datelor pacienților.

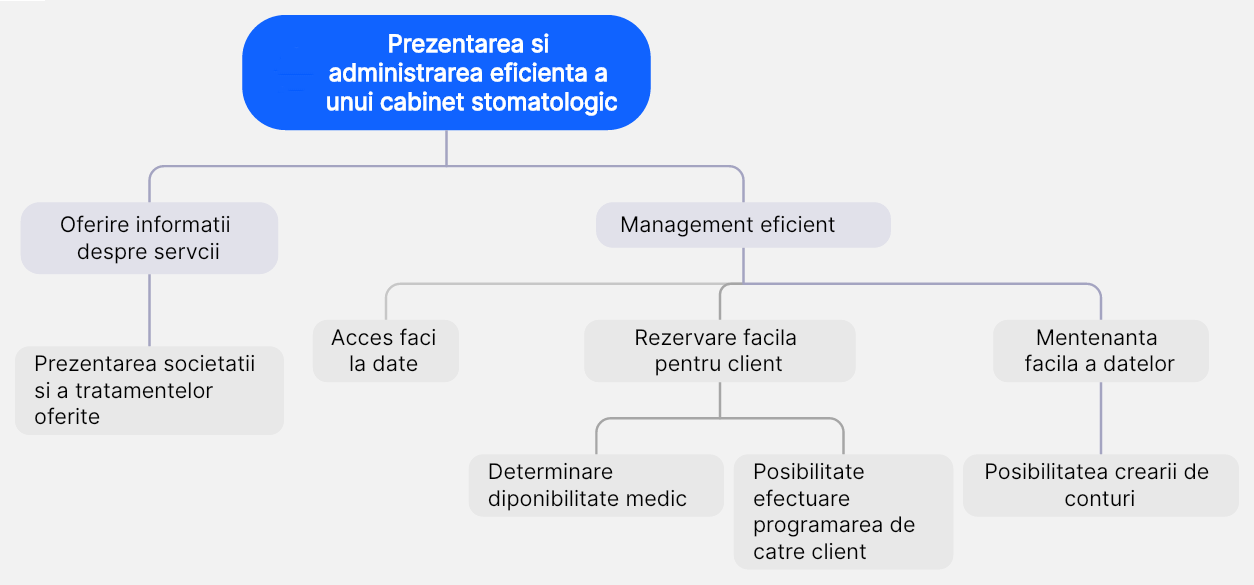
Implementarea unei astfel de soluții digitale răspunde nevoii de informații actualizate despre pacienți pentru personalul medical și de asigurare a confidențialității și securității datelor acestora.

Am utilizat diagrama Fish-bone, cunoscută și sub denumirea de diagrama cauze și efect, pentru a reprezenta vizual corelația dintre rezultat și multiplele cauze care contribuie la situația actuală. Această metodă oferă o structură clară și detaliată pentru identificarea și adăugarea cauzelor secundare care influențează problema principală, facilitând astfel procesul de analiză și înțelegere a factorilor determinanți. Prin utilizarea acestui instrument, putem descompune problemele complexe în componente mai mici și mai gestionabile, analizând fiecare cauză în parte. Astfel, putem observa legăturile dintre diferitele elemente și cum acestea contribuie la rezultatul final. În plus, diagrama Fish-bone permite echipei să colaboreze mai eficient, deoarece oferă o platformă vizuală comună pentru discuții și brainstorming. Aceasta nu doar că ajută la identificarea cauzelor evidente, dar și la descoperirea celor ascunse, care ar putea fi neglijate într-o analiză mai puțin sistematică. Prin urmare, utilizarea diagramei Fish-bone este esențială pentru a avea o perspectivă completă și detaliată asupra problemei, contribuind semnificativ la dezvoltarea soluțiilor eficiente și bine fundamentate.



*Figură 1. Diagrama “Fish-Bone”*

Pentru îndeplinirea obiectivului fundamental și abordarea eficientă a situației prezentate, este esențial să stabilim o serie de scopuri secundare, care sunt evidențiate într-o diagramă ce detaliază descompunerea obiectivelor majore. La baza acestei inițiative stă intenția de a optimiza prezentarea și gestionarea operațională a cabinetului medical. Aceasta presupune concentrarea eforturilor asupra unor direcții principale: îmbunătățirea sistemului de programari online prin crearea unui profil de utilizator, detalierea amănunțită a serviciilor oferite de cabinet și dezvoltarea capacității de a modifica, adăuga sau elimina informații din sistemul de gestionare a datelor.



*Figură 2. Diagrama de descompunere a obiectivelor*

## **Context**

Conform detaliilor menționate în prima secțiune, cabinetul dentar a început să funcționeze acum un an, evidențiind astfel necesitatea urgentă de a se face cunoscut și de a-și asigura un loc stabil pe piață. Având această prioritate, clinica este gata să investească în dezvoltarea unei pagini web ce va îmbunătăți modul de comunicare cu pacienții, permițând o interacțiune mai eficientă cu aceștia. În plus, clinica dorește să adopte o abordare modernă în gestionarea informațiilor, evitând acumularea de hârtii și documente fizice. Așadar, ne propunem să proiectăm un sistem IT personalizat care să răspundă direct acestor cerințe specificate.

Această secțiune își propune să definească și să expună atât elementele interne ale sistemului, cât și pe cele externe care interacționează cu acesta. Vom folosi metodologia fațetelor pentru a realiza această delimitare.

1. **Componenta (Fațeta) subiect**

Aplicația online va colecta și procesa informații despre medicul proprietar al cabinetului dentar, despre serviciile medicale oferite, precum și date personale ale pacienților. Conform prevederilor legale pentru protejarea datelor cu caracter personal, cabinetul trebuie să gestioneze aceste informații în condiții de securitate și exclusiv pentru scopurile stabilite. Informații despre tratamentele efectuate sunt partajate doar cu acordul pacientului.

Principalii beneficiari ai sistemului informatic sunt medicul dentist, acesta este și managerul/medicul cabinetului, precum și utilizatorii și potențialii clienți ai aplicației. Următoarele descrieri evidențiază modul în care fiecare categorie de stakeholderi este influențată de implementarea aplicației.

Manager/medic:

* Medicul folosește aplicația pentru a oferi informații despre serviciile disponibile și pentru a simplifica procesul de programare. Aplicația îi facilitează administrarea datelor despre pacienți, tratamente și programările efectuate, ajutându-l să identifice cele mai solicitate tratamente și perioadele cu cea mai mare disponibilitate a pacienților.

Utilizatorul site-ului web:

* Prin accesarea site-ului, utilizatorii pot afla detalii despre serviciile oferite și, dacă sunt interesați, pot efectua programări online, specificând data și tipul tratamentului dorit. De asemenea, aplicația le oferă oportunitatea de a oferi feedback despre experiența avută, facilitând astfel exprimarea opiniei lor.

1. **Perspectiva (Fațeta) utilizării**

Aplicația online, în curs de dezvoltare, este destinată atât navigatorilor pe internet, dar și medicului stomatolog care gestionează activităţile cabinetul.

Vizitatorii website-ului sunt interesați în principal să afle detalii despre tratamentele disponibile, despre cabinet și despre echipamentele utilizate. După ce se documentează și sunt convinși de calitatea serviciilor, își manifestă intenția de a programa o vizită.

Medicul, în rolul său de manager, folosește aplicația pentru a supraveghea și administra informațiile despre pacienți, programările și tratamentele efectuate. Prin intermediul aplicației, poate efectua operațiuni precum vizualizarea, adăugarea, modificarea sau ștergerea datelor. Utilizarea digitală elimina necesitatea documentelor fizice sau a unei persoane dedicate pentru gestionarea datelor. Medicul poate, de asemenea, să genereze rapoarte detaliate despre activitatea cabinetului, cum ar fi frecvența tratamentelor sau numărul clienților pe o anumită perioadă.

Traseul unui vizitator de pe site-ul nostru începe când acesta dorește informații despre un tratament specific, cum ar fi albirea dinților. După ce accesează website-ul și este mulțumit de informațiile și prețurile prezentate, alege să facă o programare. Procesul include crearea unui cont, furnizarea unor detalii personale necesare, și stabilirea detaliilor tratamentului și a orei dorite. Confirmarea programării se va face printr-un e-mail trimis de către medic.

Există trei tipuri principale de utilizatori ai aplicației:

• Utilizatori noi, la prima vizită

• Utilizatori existenți, cu conturi create

• Medic/Manager

• Administrator

Politica de acces variază în funcție de tipul de utilizator:

• Utilizatorii noi au acces complet la informațiile din meniul site-ului, scopul fiind acela de a-i încuraja să creeze un cont.

• Utilizatorii cu cont pot efectua programări online și pot lăsa feedback despre serviciile primite.

• Medicul/Managerul are acces complet la detalii despre pacienți și programări, putând actualiza baza de date și crea rapoarte.

• Administratorul gestionează conturile utilizatorilor.

1. **Aspectele (Fațeta) IT**

Din perspectiva tehnologică, obiectivul principal al dezvoltării aplicației este de a optimiza interacțiunea cu utilizatorii, asigurând o prezentare clară a informațiilor și un transfer eficient de date. Accesibilitatea aplicației este garantată prin asocierea acesteia cu un nume de domeniu și o adresă IP dedicată.

Pentru realizarea aplicației, am ales limbajul de programare C#, dezvoltat de Microsoft, datorită similarităților sale cu C++ și Java, precum și pentru simplitatea și eficiența sa în codificare. C# este un limbaj orientat obiect, ideal pentru crearea de componente software complexe compatibile cu diverse sisteme de operare.

Platforma .NET, fundamentul pentru C#, facilitează rularea și dezvoltarea aplicațiilor web și a serviciilor web, operând indiferent de platforma hardware folosită. De asemenea, .NET permite integrarea altor limbaje de programare, cum ar fi Visual Basic, Java și C++.

Mediul de dezvoltare utilizat va fi Microsoft Visual Studio 2015, care incorporează un arsenal de instrumente pentru construirea de aplicații ASP.NET, servicii Web XML, precum și aplicații pentru desktop și dispozitive mobile. Aceste instrumente, disponibile pentru Visual C++,Visual Basic,Visual J# și Visual C#, susțin dezvoltarea de soluții poliglotice și exploatează la maximum facilitățile oferite de .NET Framework, simplificând astfel dezvoltarea aplicațiilr web ASP și servicii Web XML cu Visual Web Developer.

În ceea ce privește gestionarea datelor, aplicația va utiliza Microsoft SQL Server. Conexiunea și accesul la baza de date vor fi realizate prin intermediul Entity Framework, un cadru ORM (Object/Relational Mapping) care convertește datele relaționale în obiecte specifice domeniului. Acesta reduce semnificativ necesitatea scrierii de cod extensiv pentru accesul la date, permitând dezvoltatorilor să interogheze baza de date prin LINQ și să manipuleze datele ca obiecte tipizate. Entity Framework oferă, de asemenea, funcționalități precum urmărirea modificărilor, rezoluția identității, încărcarea leneșă și traducerea interogărilor, permitând dezvoltatorilor să se focalizeze mai mult pe aspectele de logică de afaceri decât pe tehnicile de gestionare a datelor.

1. **Procesul (Fațeta) de dezvoltare**

Construirea sistemelor informatice traversează mai multe faze cheie, constituind ciclul de viață al software-ului. Aceste etape includ analizarea nevoilor utilizatorilor, conceperea arhitecturii sistemului, realizarea efectivă a software-ului, testarea funcționalităților și menținerea continuă. Fiecare dintre aceste segmente trebuie abordat cu metode și tehnici specifice.

Pentru acestă aplicaţie, am selectat modelul de dezvoltare în cascadă, o metodă tradițională care impune finalizarea completă a unei faze înainte de a începe următoarea. Acesta este un model structurat pe nivele, unde progresia seamănă cu etapele unei cascade naturale. Modelul este potrivit pentru proiecte unde cerințele sunt clar definite de la început și unde nu se anticipază nevoia de modificări majore pe parcursul dezvoltării. Adoptarea acestui model asigură o abordare disciplinată și previzibilă, ideală pentru specificațiile noastre bine stabilite.

# **2.** **Cerințe de sistem**

## **2.1 Surse de cerințe**

Sursele principale ale cerințelor pentru aplicație provin de la stakeholderi, deoarece ei sunt direct implicați în utilizarea și beneficiile oferite de aceasta.

În urma unor discuții aprofundate cu medicul stomatolog, care ocupă și rolul de manager al cabinetului, am identificat aspectele ce necesită îmbunătățiri în prezentarea afacerii și gestionarea clienților. Am luat în considerare și necesitățile potențialilor pacienți care ar putea folosi aplicația, încercând să aliniem aplicația la așteptările lor pentru a asigura succesul acesteia.

Chiar dacă au existat situații în care stakeholderii nu au fost siguri de ceea ce doreau sau nu au putut articula clar nevoile lor, am lucrat pentru a ajunge la un consens și a stabili clar cerințele necesare dezvoltării aplicației.

**2.2 Elicitația cerințelor**

Când discutăm despre părțile implicate în acest proiect, trebuie să includem atât stakeholderii, cât și echipa noastră care lucrează la dezvoltarea aplicației.

Un beneficiu major pentru cabinetul stomatologic este capacitatea aplicației de a ajuta la promovarea serviciilor și la creșterea bazei de clienți. Utilizatorii, pe de altă parte, vor găsi mai ușor să programeze vizite și să obțină informații despre serviciile oferite.

Pentru noi, dezvoltatorii, avantajele includ atât profiturile financiare câștigate, cât și experiența pe care o acumulăm. După finalizarea acestui proiect, intenționăm să utilizăm aplicația ca prototip pentru dezvoltarea altor platforme similare, ceea ce ne-ar putea ajuta să atragem mai mulți clienți din aceeași industrie.

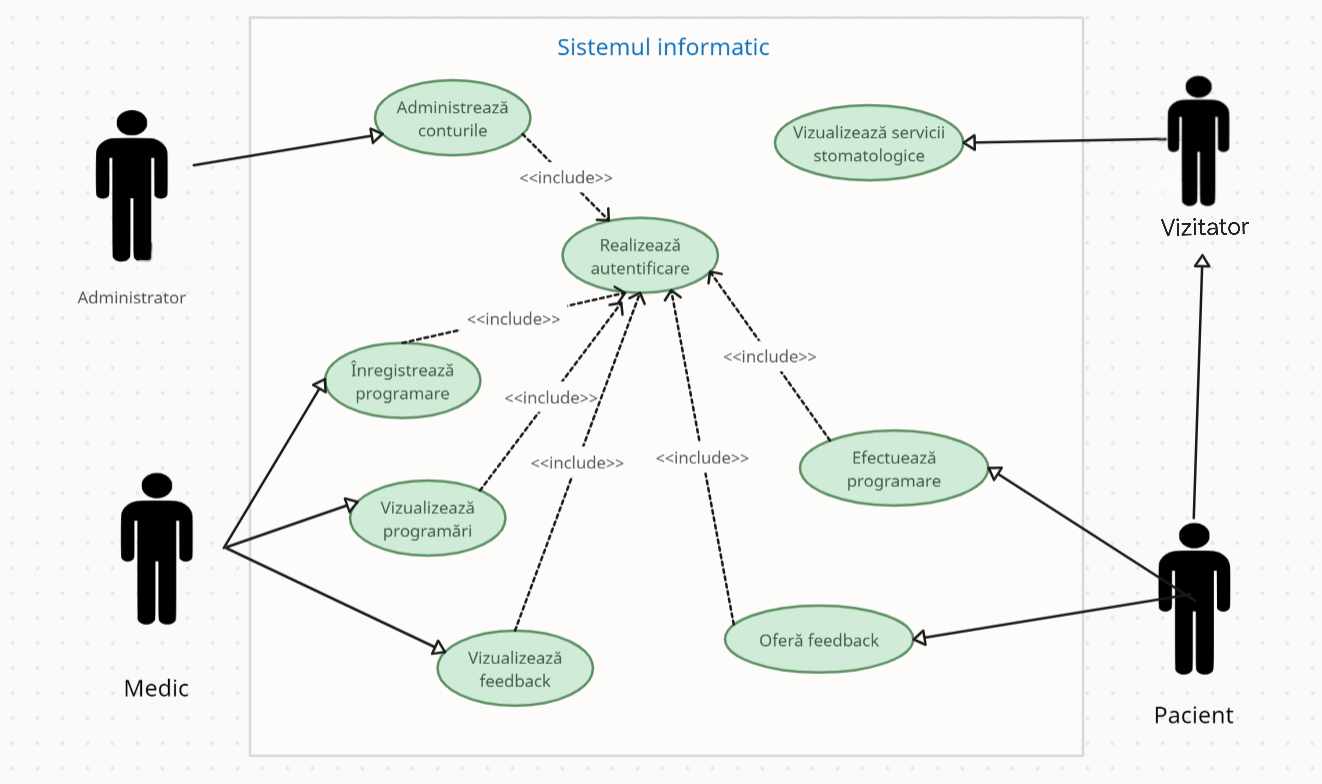
Există, însă, și câteva dezavantaje. Beneficiarul nu poate ajusta sau extinde aplicația fără asistența mea, ceea ce înseamnă că orice modificare necesită intervenția echipei noastre.

Din perspectiva mea, deși scopul meu este să dezvolt site-ul conform cerințelor și să o mențin activă post-lansare, sarcina de întreținere poate deveni un dezavantaj. Resursele și timpul alocat pentru mentenanță ar putea fi folosite pentru dezvoltarea de noi proiecte și extinderea portofoliului de clienți.

Am folosit diverse metode, precum cazurile de utilizare, interviurile și modelarea de business, pentru a asigura că înțelegem în profunzime necesitățile stakeholderilor și pentru a identifica cerințele precise ale aplicației.

**Metoda Cazurilor de Utilizare (Use Cases)**

Această metodă are ca obiectiv identificarea interacțiunilor dintre utilizatori și sistem, precum și definirea interfețelor necesare. Este esențială pentru asigurarea că toate cerințele sunt luate în considerare și că dezvoltatorii înțeleg nevoile beneficiarului.



*Figură 3. Diagrama Use Cases*

Descrierea cazurilor de utilizare a fost realizată utilizând limbajul de modelare unificat (UML) prin Creately. Din diagrama prezentată, se observă următorii actori principali ai sistemului:

* Vizitatorul: are capacitatea de a naviga prin paginile site-ului și de a iniția crearea unui cont.
* Pacientul: similar cu vizitatorul, acesta trebuie să aibă un cont preexistent. Accesează complet interfața pentru a vedea informații despre serviciile și procedurile disponibile, pentru a compara tarife, a oferi feedback și a efectua programări online.
* Medicul: responsabil cu gestionarea internă a cabinetului, acesta accesează detalii despre programările actuale, pacienții înregistrați, și are autoritatea de a modifica sau actualiza serviciile, procedurile și tarifele. De asemenea, validează feedback-ul și consultă diverse rapoarte.
* Administratorul: acesta administrează conturile utilizatorilor, având facilitatea de a modifica, șterge sau crea conturi noi.

Pe baza discuțiilor despre funcționalitățile dorite de beneficiar, s-au definit următoarele cazuri de utilizare:

* Administrează conturi: implică gestionarea detaliilor conturilor utilizatorilor.
* Realizează autentificare: permite introducerea datelor de acces, cum ar fi numele de utilizator și parola, necesare pentru efectuarea unei programări.
* Vizualizează servicii: oferă posibilitatea de a explora meniul site-ului.
* Efectuează programare: permite pacienților să completeze detaliile necesare într-un formular de programare.
* Înregistrează programare: permite medicului să programeze un pacient pentru serviciul dorit.
* Vizualizează programări/rapoarte: permite medicului să revadă lista de programări și să consulte rapoarte detaliate bazate pe nevoile sale.
* Oferă feedback: permite pacienților să transmită feedback despre serviciile primite.
* Vizualizează feedback: medicul poate vedea și evalua feedback-ul primit de la pacienți.

Documentația textuală a acestor scenarii va fi detaliată în continuare pentru a clarifica fiecare caz de utilizare în parte.

* 1. **Proces de Programare Online**

Descrierea procedurii prin care un pacient poate realiza o programare online.

**Actori** :Pacient

**Prioritate** :Ridicată

**Pre-condiții** :Utilizatorul trebuie să fie autentificat.

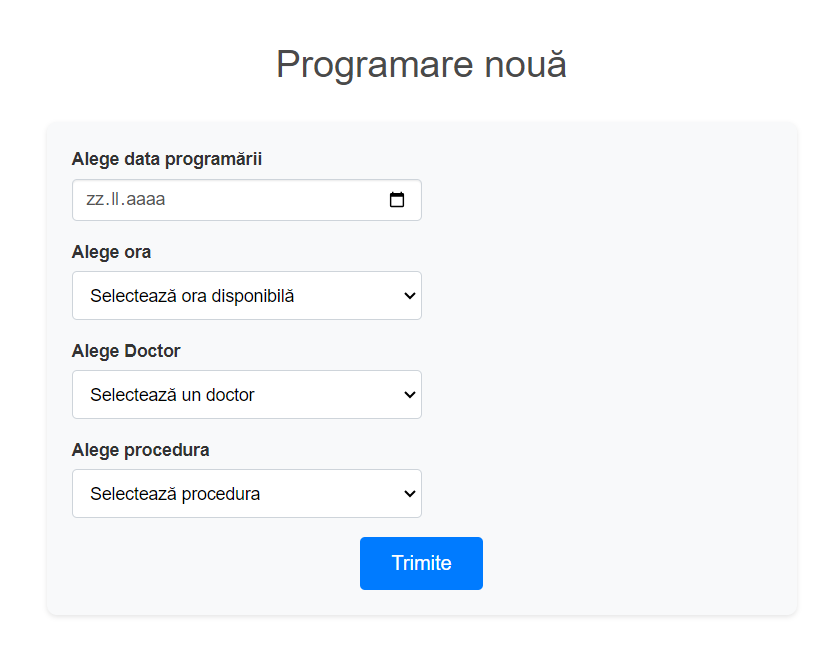
**Post-condiții**

* Programarea este realizată și stocată în sistem.
* Pacientul primește un e-mail de confirmare cu detaliile programării.
* Pacientul rămâne pe pagina principală.

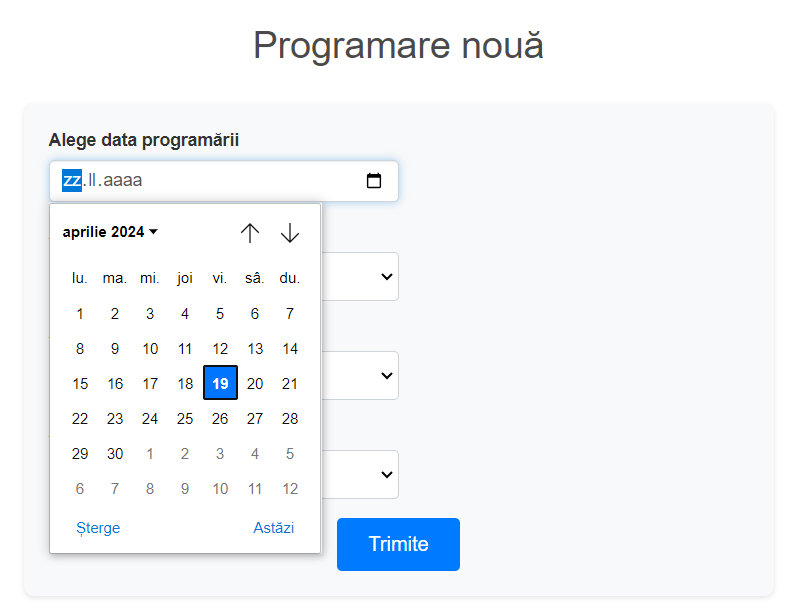
**Fluxul principal de acțiuni**

1. Autentificare
2. Pacientul selectează opțiunea de a programa din meniu.
3. Se încarcă pagina de programare.
4. Pacientul apasă pe "Alege data programării" pentru a deschide calendarul.
5. Un calendar este afișat.
6. O dată este selectată din calendar.
7. Pacientul apasă pe "Ora" pentru a vedea orele disponibile.
8. Lista cu ore disponibile este afișată.
9. Pacientul alege ora dorită.
10. Pentru alegerea procedurii, pacientul apasă "Alege procedură".
    * Aplicația prezintă o listă arborescentă a serviciilor disponibile.
11. Pacientul finalizează prin a apăsa "Trimite".
12. Datele programării sunt salvate în sistem.
13. Pacientul este îndrumat către pagina principală.

**Interfața pentru utilizator**

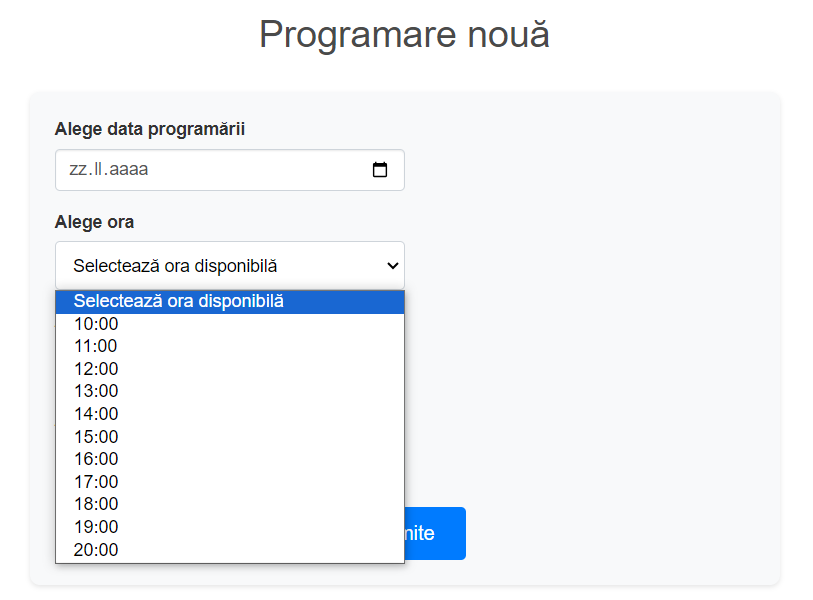


**Selectorul de dată**

****

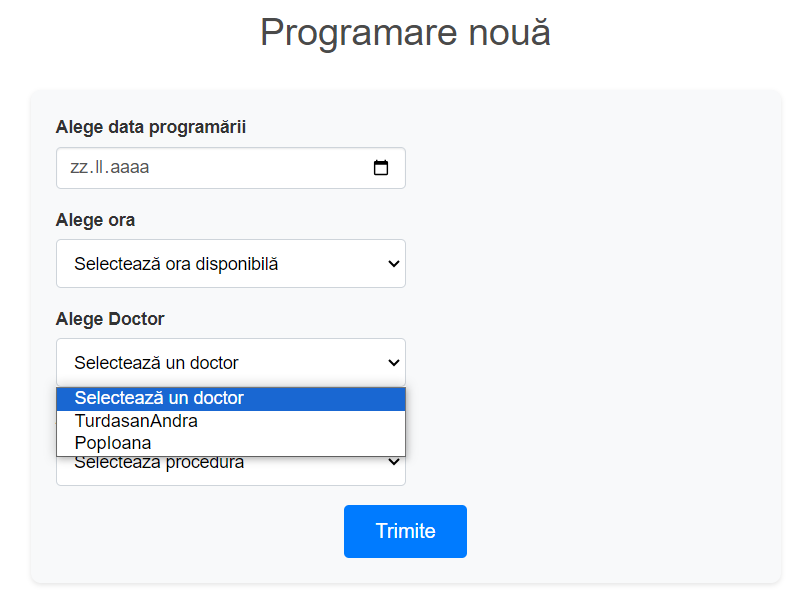
Sistemul permite atât completarea câmpului de dată de mână, cât și utilizarea unui calendar interactiv. Pentru comoditate, data zilei de azi este preselectată automat. În plus, la completarea manuală, nu se acceptă date care sunt înainte de ziua actuală. De asemenea, câmpul prezintă o valoare prestabilită ce îi indică utilizatorului formatul corect pentru introducerea datei.

**Meniu derulant de selecție a orei**

****

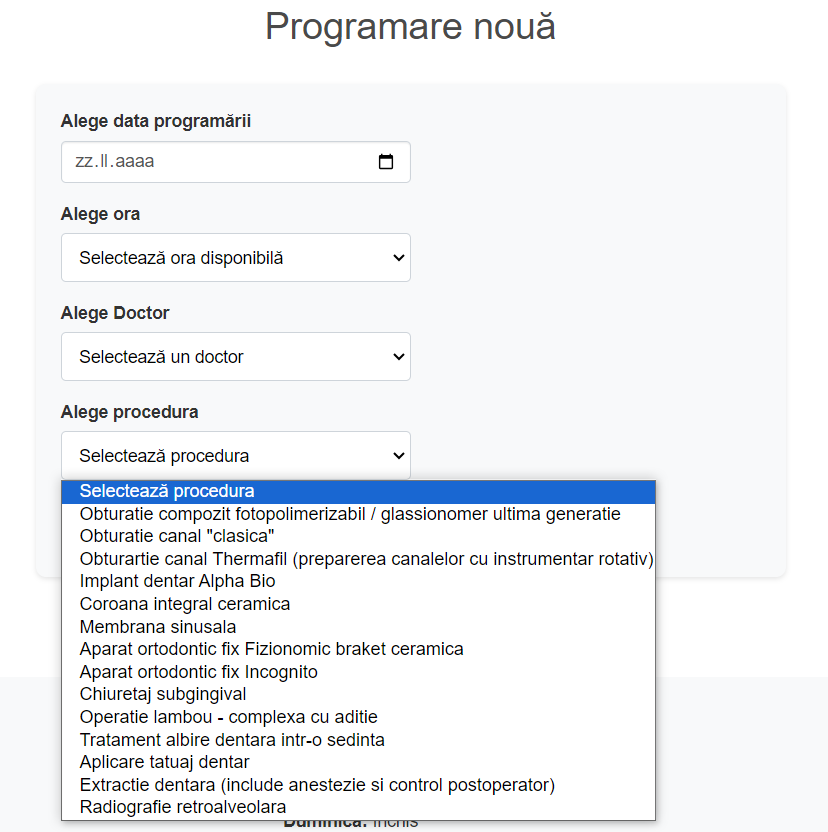
Sistemul prezintă un meniu derulant ce conține intervalul orar disponibil al medicului pentru stabilirea programărilor. În plus, se furnizează o opțiune implicită, „Selectează ora disponibilă”, care îndrumă utilizatorul cu privire la tipul de informație ce trebuie să o introducă.

**Meniu de selecție a medicului**

****

Acesta este un meniu derulant care afișează numele tuturor medicilor disponibili pentru programare. Utilizatorii pot derula prin lista respectivă și pot selecta numele dorit. Pot să înceapă să tasteze numele medicului, iar sistemul va sugera automat numele care se potrivesc cu caracterele introduse, facilitând astfel găsirea medicului preferat într-un mod rapid și eficient.

**Meniu de selecție a procedurii**

****

Când vine vorba de selectarea unui serviciu, sistemul pune la dispoziție o listă din care utilizatorul poate alege procedura dorită. Pentru a simplifica procesul de selecție, pacientul are opțiunea de a apăsa litera inițială a serviciului căutat, caz în care sistemul filtrează și afișează automat toate procedurile care încep cu acea literă.

* 1. **Procedura de Programare**

**Contextul Procedurii:** Acest set de instrucțiuni explică secvențial acțiunile necesare pentru ca un medic să programeze o consultație.

**Roluri Implicate:** Medic

**Natura Scenariului:** Scenariu de interacțiune

**Scop:** Efectuarea unei programări

**Precondiții:** Autentificarea medicului este esențială.

**Post-condiții:**

* Finalizarea programării pentru un pacient specific și înregistrarea detaliilor.
* Menținerea sesiunii medicului pe pagina de bază.

**Proces Principal:**

1. Logare
2. Accesarea funcției de programări din interfața principală.
3. Afișarea ecranului de programare.
4. Introducerea identității pacientului.
5. Pentru pacienți preexistenți, sistemul confirmă și arhivează numele complet.
6. Pentru pacienți noi:
   * Indicație afișată: "Acest pacient nu exista.Înregistrați pacientul nou."
   * Prezentarea unui chestionar de înregistrare.
   * Completarea detaliilor de către medic.
   * Păstrarea noilor date în baza de date.
7. Inițierea alegerii datei pentru întâlnire.
8. Calendarul devine vizibil prin intermediul aplicației.
9. Determinarea unei date anume.
10. Apăsarea pentru deschiderea intervalului orar.
11. Fixarea orei întâlnirii.
12. Desfășurarea listei de proceduri medicale disponibile.
13. Selecția de către medic a serviciului necesar pacientului.
14. Finalizarea prin butonul de confirmare "Trimite".
15. Consignarea în sistem a datelor programării.

**Criterii de Eficiență:** Realizarea activităților este optimizată pentru rapiditate și simplitate, amplificând astfel experiența de utilizare a aplicației.

* 1. **Consultare Programări**

Aceasta secțiune detaliază procesul prin care un medic poate consulta lista programărilor realizate.

**Participanți:** Medic

**Caracterul Scenariului:** Interactiv

**Scopuri:** Examinarea programărilor pentru gestionarea eficientă a timpului.

**Precondiții:** Autentificarea pe baza unui cont de medic este o condiție obligatorie pentru accesul la interfața dedicată.

**Post-condiții** Listarea de către sistem a tuturor programărilor solicitate.

**Secvența de Acțiuni Principală:**

1. Logarea medicului.
2. Interfața specifică medicului este afișată de aplicație.
3. Selectarea de către medic a secțiunii "Listă de programări".
4. Stabilirea datei specifice pentru consultarea programărilor.
5. Sistemul listează detaliile programărilor pentru data aleasă, inclusiv ora, numele pacientului și serviciul medical.

**Indicatori de Performanță:** Simplitatea și rapiditatea activităților contribuie la creșterea eficacității aplicației.

* 1. **Procesul de Oferire a Feedback-ului**

Acest scenariu explică etapele pe care un pacient le urmează pentru a oferi feedback.

**Participanți:** Utilizator cu cont de pacient

**Tipul Scenariului:** Interactiv

**Obiective:** Transmiterea feedback-ului de către utilizator

**Condiții Inițiale:** Utilizatorul trebuie să fie înregistrat pentru a se putea autentifica și a furniza feedback.

**Condiții Finale:** Feedback-ul oferit va fi stocat în baza de date pentru afișare ulterioară.

**Rezultate:** Feedback înregistrat și așteptat pentru moderare

**Flux Principal de Acțiuni:**

1. Logare a pacientului.
2. Prezentarea paginii principale de către sistem.
3. Selectarea secțiunii „Testimoniale” din meniul superior.
4. Apăsarea butonului „Adaugă testimonial”.
5. Afișarea formularului de introducere a feedback-ului.
6. Completarea secțiunii „Descriere” de către pacient.
7. Confirmarea trimiterii prin apăsarea butonului „Trimite”.
8. Înregistrarea datelor de feedback în baza de date.

**Standarde de Calitate:**Procesul este conceput să fie rapid și simplu, crescând utilizabilitatea aplicației.

* 1. **Procesul de Autentificare**

Acest scenariu ilustrează procedura de autentificare a utilizatorilor în aplicație și detaliază permisiunile acordate diferitelor grupe de utilizatori

**Participanți:** Utilizator (pacient, administrator sau medic)

**Natura Scenariului:** Interactiv

**Obiective:** Logarea pentru acces la funcționalități specifice, precum vizualizarea informațiilor sau programarea de servicii.

**Precondiții:** Utilizatorul trebuie să fie înregistrat și să dispună de o conexiune stabilă la internet pentru a se autentifica.

**Post-condiții:** Utilizatorul este autentificat și capabil să efectueze acțiunile necesare.

**Rezultatul Final:** Succesul autentificării în sistem.

**Flux Principal de Acțiuni:**

1. Afișarea paginii de logare de către sistem.
2. Introducerea de către utilizator a numelui de utilizator și a parolei.
3. Validarea credențialelor de către aplicație.
4. Redirecționarea către pagina specifică tipului de cont al utilizatorului.

**Scenariu Excepțional:**

Dacă datele de logare sunt introduse incorect:

1. Mesaj de eroare afișat de sistem: „Nume de utilizator sau parolă incorecte”.
2. Revenirea la pasul 2 din fluxul principal.

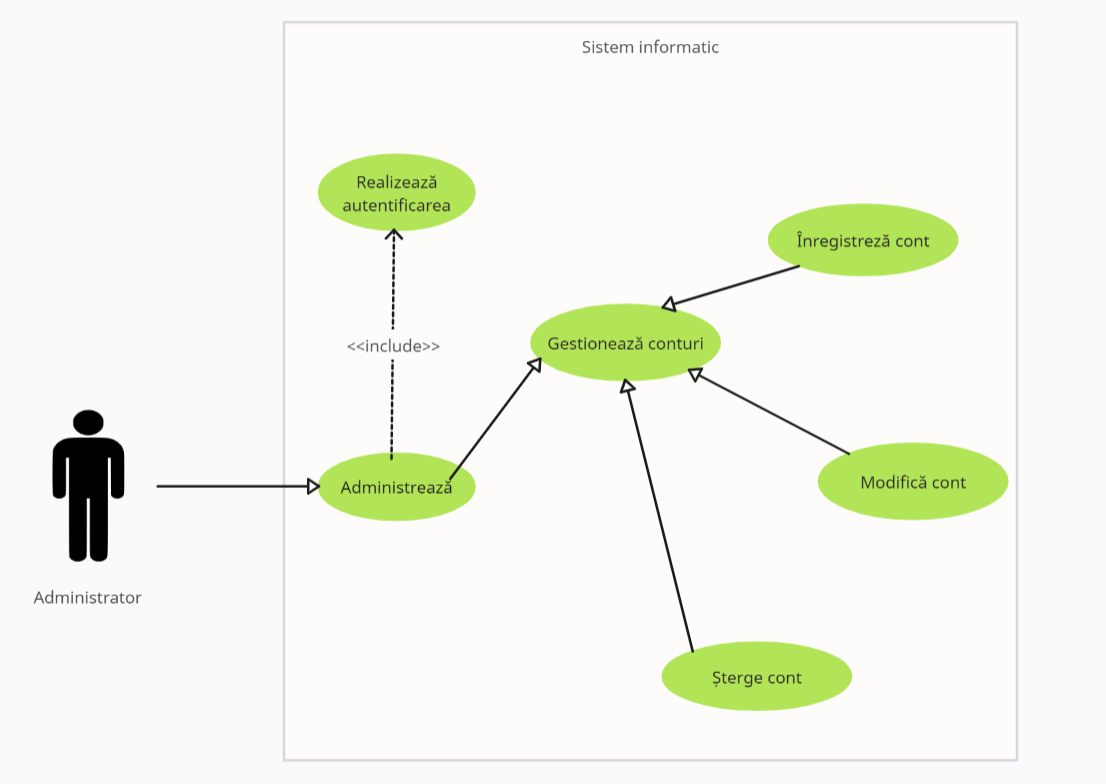
**Prezentare pe scurt a altor cazuri de utilizare:**

**Vizualizare Servicii Stomatologice:**

* **Participanți:** vizitator
* **Flux Principal:**
  1. Vizitatorul alege din meniu opțiuni precum „Despre noi”, „Servicii”, „Tarife”, „Testimoniale”, „Contact”.
  2. Sistemul redă pagina corespunzătoare selecției.

**Administrare Conturi:**

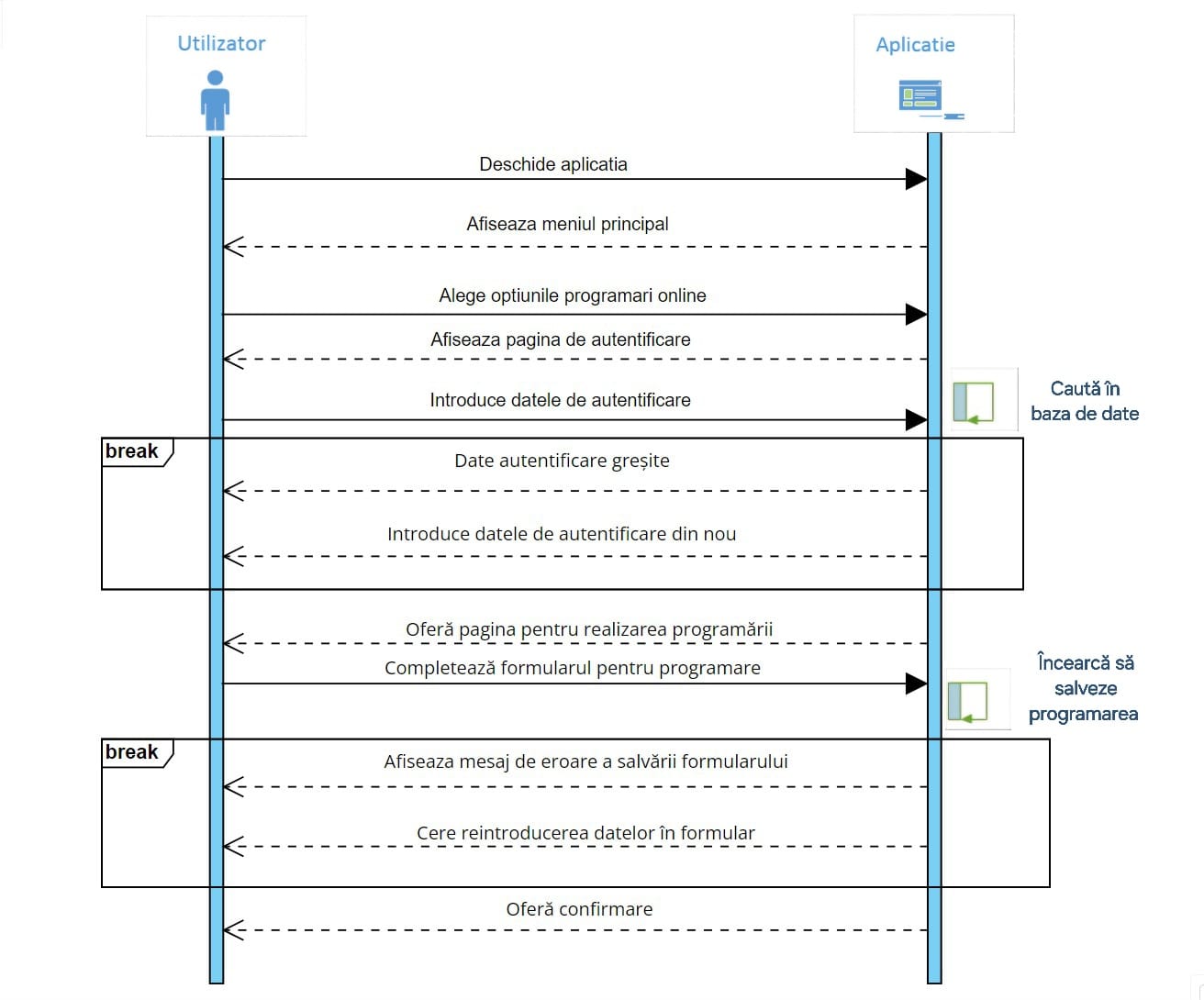
* **Participanți:** administrator
* **Descriere:** Acțiuni efectuate de administrator pentru adăugarea sau modificarea unui cont.
* **Flux Principal:**
  1. Logarea administratorului cu drepturi complete.
  2. Alegerea de către administrator a opțiunii dorite: Înregistrează cont, Șterge cont, Modifică cont.
  3. Salvarea modificărilor efectuate de sistem.



Figură 4. Diagrama cazurilor de utilizare pentru Administrator

Diagrama prezentată este o reprezentare UML a cazurilor de utilizare pentru funcțiile de administrare efectuate de un administrator în cadrul unui sistem informatic. În centrul interacțiunilor stă capacitatea administratorului de a gestiona conturile, o funcție ce include activități precum autentificarea (indicată ca fiind o cerință obligatorie), înregistrarea de noi conturi, modificarea detaliilor conturilor existente, și ștergerea conturilor. Relațiile dintre aceste cazuri de utilizare subliniază fluxul logic al proceselor și dependențele dintre ele în gestionarea sistemului

Diagrama de secvență descrisă mai jos joacă un rol crucial în procesul de documentare a scenariilor de utilizare. Aceasta captează și ilustrează în detaliu interacțiunile dintre actori și sistem pe parcursul timpului, oferind astfel o perspectivă clară și structurată asupra fluxurilor de proces și a diferitelor scenarii posibile. Utilizarea acestei diagrame este deosebit de valoroasă în clarificarea secvențelor de acțiuni și în identificarea punctelor critice ale interacțiunii, contribuind semnificativ la optimizarea designului și funcționalității sistemului.



Figură 5. Diagrama de secvențe

## *2.2.1 Metoda interviului*

Interviul este o tehnică eficientă pentru elucidarea dificultăților întâmpinate de o organizație și identificarea soluțiilor adecvate din perspectiva tuturor părților implicate. Acest proces este valoros atât în faza de analiză a problemei cât și în etapa de colectare a cerințelor.

Întrebări generale și specifice dintr-un dialog cu un client pot include:

1. În ce mod este prezentată afacerea în mediul online în prezent? Răspuns: Utilizăm o pagină de Facebook pentru a afișa detaliile de contact.
2. Ce sistem este implementat pentru organizarea programărilor și administrarea clienților? Răspuns: Înregistrez informații despre clienți și detaliile programărilor într-o agendă personală.
3. Ce îmbunătățiri credeți că ar putea fi aduse modului actual de prezentare? Răspuns: Considerând importanța tot mai mare a internetului, mi-ar plăcea să dezvolt un site web care să includă informații despre serviciile oferite, detaliile de contact și o opțiune pentru realizarea online a programărilor.

3.1 Care informații sunt esențiale pentru stabilirea unei programări? Răspuns: Ar fi necesare numele, prenumele, numărul de telefon, adresa de email, tipul serviciului dorit și ora la care pacientul este disponibil.

3.2 Este dorită trimiterea automată a confirmărilor pentru programări? Răspuns: Da, ar fi util ca fiecare client să primească automat un e-mail de confirmare.

1. Doriți includerea gestionării activităților interne în aplicație, pe lângă interfața destinată clienților? Răspuns: Mi-ar fi de ajutor o soluție software ce mă scutește de necesitatea consultării agendei, permițându-mi să văd direct pacienții, tratamentele aplicate și programările.

4.1 Cum se vor introduce datele? Răspuns: Atât eu, cât și pacienții vom introduce datele necesare prin intermediul unor formulare online.

1. Care sunt așteptările dumneavoastră privind simplitatea în utilizarea aplicației? Răspuns: Doresc ca software-ul să faciliteze realizarea programărilor rapid și intuitiv, permițând utilizatorilor să găsească informațiile dorite cu eficiență.

5.1 Cine va opera aplicația? Răspuns: Aplicația va fi folosită de mine, în rolurile de manager și medic, dar și de clienții potențiali, vizitatori ai site-ului web. Un administrator se va ocupa de gestionarea bazei de date, dată fiind lipsa mea de competențe în acest domeniu.

5.2 Care vor fi restricțiile de acces pentru diferitele categorii de utilizatori? Răspuns: Vizitatorii noi vor avea acces limitat la descrierile serviciilor și la crearea unui cont, clienții înregistrați vor putea face programări online, în timp ce eu voi avea acces complet la baza de date, inclusiv la informațiile despre pacienți, programări și servicii.

## *2.2.2 Modelul de business*

Modelul de business este esențial pentru înțelegerea și reprezentarea abstractă a proceselor și structurii organizaționale. Acesta clarifică interacțiunile cu partenerii externi și organizarea spațială a elementelor organizației. Contribuind indirect la optimizarea acesteia, modelul poate facilita, de asemenea, reingineria proceselor organizaționale.

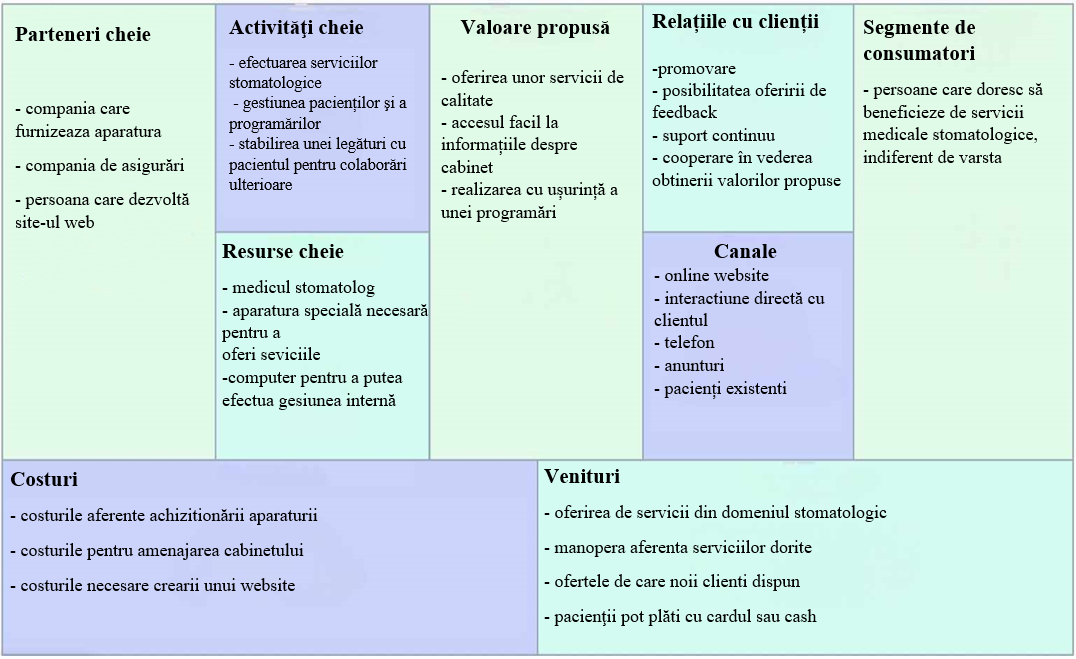


Figura 6. Modelul de business

## *2.2.3 Workshop de cerințe*

Această tehnică de colectare a informațiilor implică adunarea tuturor părților interesate pentru o întâlnire concisă și bine definită, cu scopul de a înțelege opiniile și necesitățile fiecărui participant. Beneficiile acestei abordări includ obținerea rapidă a feedback-ului, percepția de transparență a procesului și oportunitatea de a asculta fiecare voce.

În situația descrisă, noi, dezvoltatorii, am convocat o întâlnire cu medicul proprietar al clinicii dentare, doi dintre pacienții săi și o persoană care nu cunoștea clinica până la acel moment. În timpul întâlnirii, am organizat sesiuni de generare de idei și, la final, am distribuit câteva chestionare. Obiectivul acestui atelier a fost de a identifica nevoile și preferințele utilizatorilor în ceea ce privește designul și conținutul site-ului web.

Feedback-ul obținut a indicat următoarele preferințe ale utilizatorilor:

* O interfață ușor de utilizat
* Prezentarea detaliată a serviciilor oferite
* Includerea testimonialelor
* Un design simplu și intuitiv
* Un sistem eficient pentru programările online
* Informații de contact exacte și ușor accesibile
* Necesitatea primirii confirmărilor prin e-mail.

## **2.3** **Formularea cerințelor**

Cerințele sunt clasificate în două grupe principale: funcționale și calitative. Prin cerințele funcționale explică ce acțiuni trebuie să efectueze sistemul, cum să reacționeze la diverse intrări sau ieșiri, sau în situații specifice. Cerințele calitative, pe de altă parte, se ocupă de calitatea și condițiile în care sistemul își realizează funcțiile.

Detaliile pentru cerințele funcționale includ:

* Sistemul ar trebui să furnizeze utilizatorilor date complete despre serviciile disponibile.
* Este necesar să includă un meniu ușor de utilizat pentru acces simplu la informații.
* Utilizatorii trebuie să poată crea conturi personale.
* Sistemul ar trebui să verifice exactitatea datelor introduse în formulare.
* Trebuie să ofere detalii curente despre disponibilitatea medicilor.
* Este esențial să poată genera diverse rapoarte necesare medicului.
* Ar trebui să confirme automat programările efectuate.
* Trebuie să faciliteze adăugarea simplă a datelor și utilizatorilor noi în baza de date.
* Nu ar trebui să permită introducerea de informații greșite în formulare.
* Trebuie să asigure protecție ridicată a datelor.

Pentru cerințele calitative:

* Aplicația ar trebui să fie intuitivă, ușor de întreținut și testat, oferind performanță superioară în aceste domenii.

## *2.3.1 Procese și activități*

Scopul principal al aplicației este abordat din două perspective: prezentarea cabinetului stomatologic și gestionarea internă.

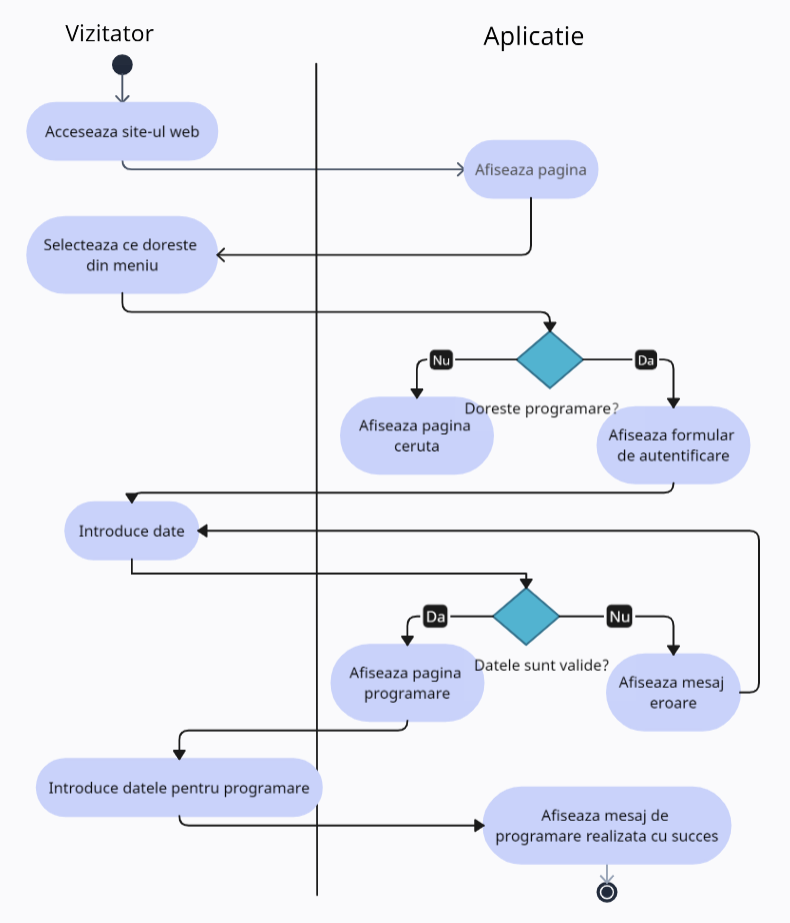
Pentru prezentare, afacerea utilizează un profil de Instagram care oferă detalii despre localizare și contact. În plus, promovarea se face printr-un sistem de recomandări directe între pacienți, aceștia sugerând cabinetul altor persoane în cazul în care sunt satisfăcuți de tratamentul primit.

Din perspectiva administrativă, managementul pacienților este realizat cu ajutorul unei agende tradiționale. Când un pacient contactează cabinetul pentru o programare, medicul îi notează informațiile personale, tratamentul dorit și detaliile programării manual.

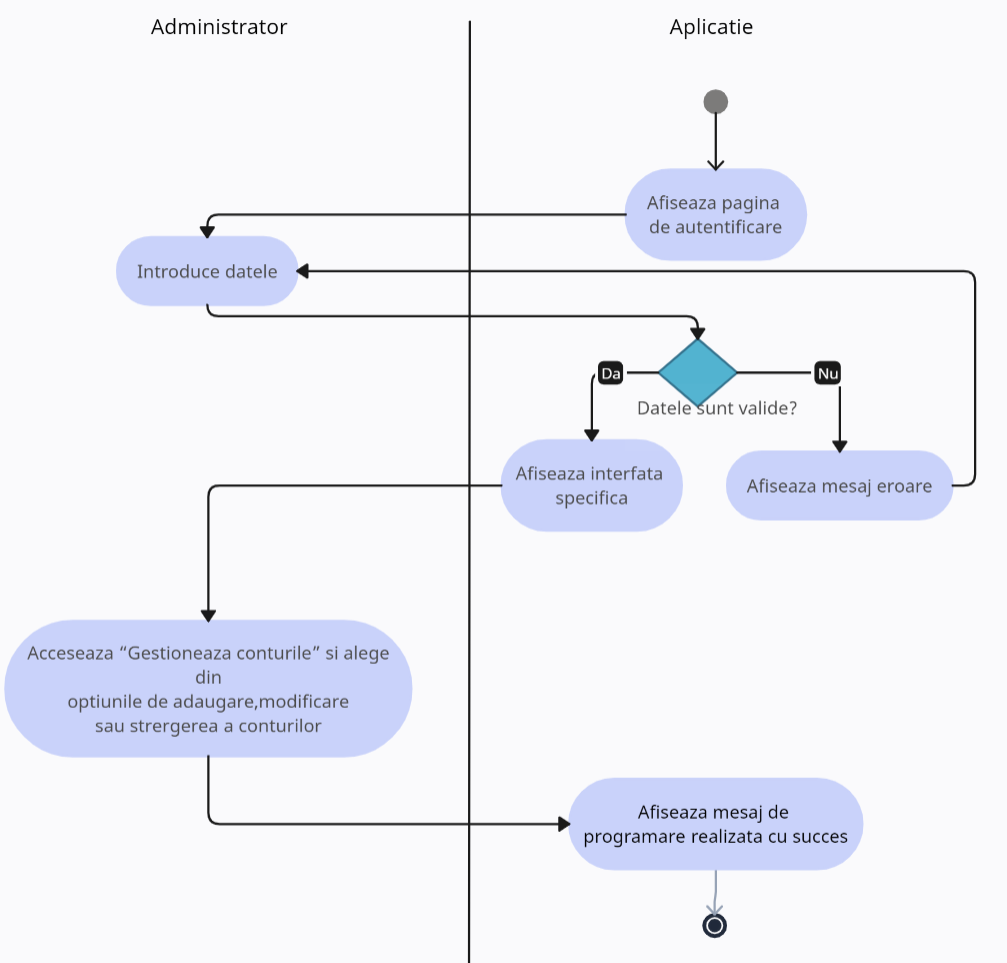
Pentru optimizarea acestui sistem, intenționăm să dezvoltăm un site web care să integreze atât funcții de administrate, cât și de prezentare. Interfața utilizatorului va afișa un meniu cu serviciile disponibile și va permite efectuarea de programări online. Pacienții vor crea un cont personal în care să își introducă datele necesare pentru programare, care vor fi apoi stocate într-o bază de date. După finalizarea programării, utilizatorul va primi un email de confirmare, prin care medicul poate confirma sau reprograma vizita.

Interfața de administrare va permite managerilor accesul la baza de date pentru adăugarea sau modificarea informațiilor și gestionarea conturilor de utilizator.

Privind din perspectiva vizitatorului, acesta accesează website-ul, fiind întâmpinat de pagina principală de unde poate explora serviciile oferite. După informarea despre servicii și tarife, dacă dorește să programeze o vizită, va apăsa pe butonul "Programare". Acesta va fi ghidat către pagina de înregistrare pentru crearea unui cont nou sau spre un formular de programare dacă este deja înregistrat. La finalizarea acestui proces, va fi redirecționat la pagina de start a aplicației.



*Figură 7. Diagrama de activități “Vizitator”*



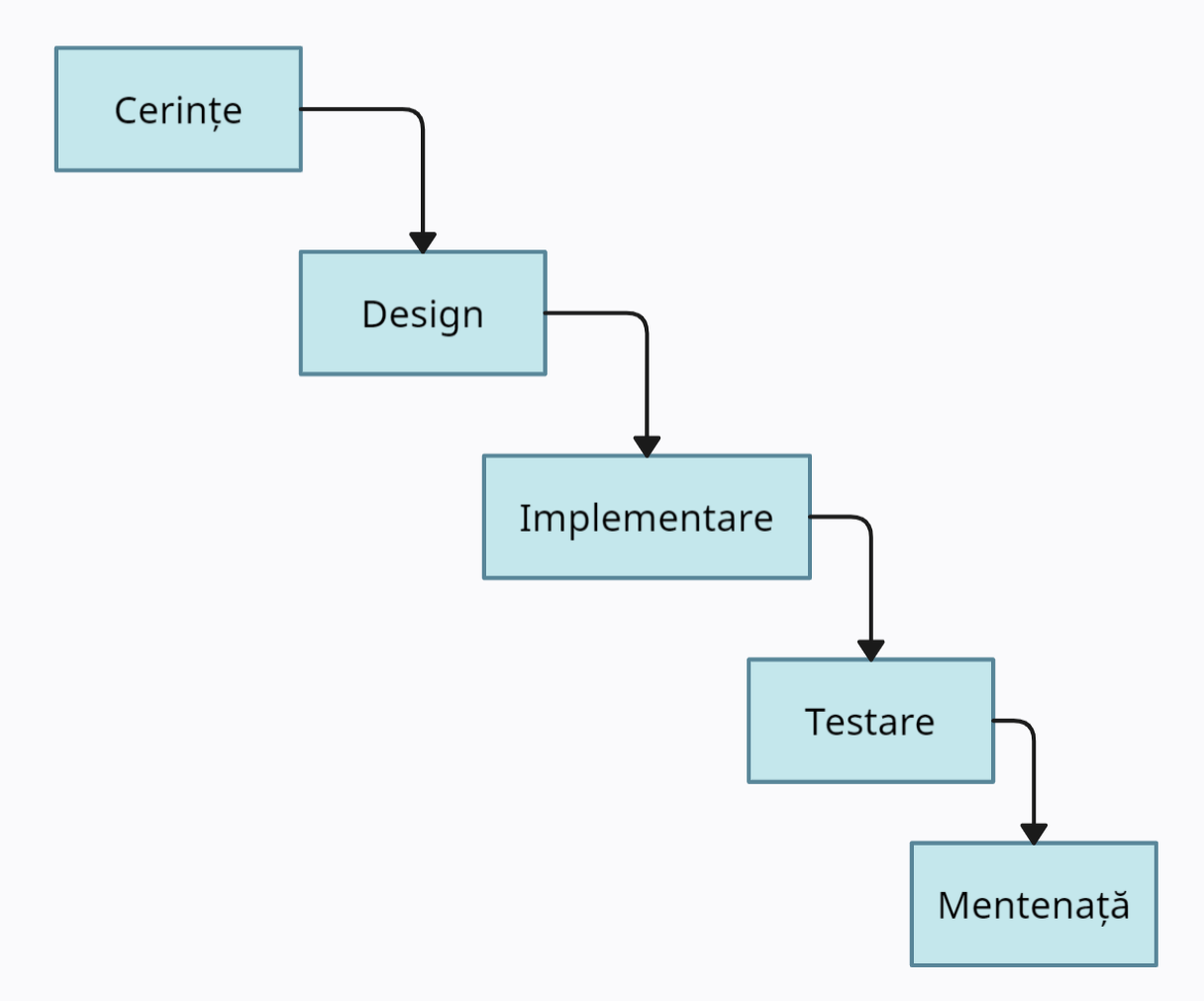
*Figură 8. Diagrama de activități “Administrator”*

# **3**. **Model de dezvoltare**

Așa cum este detaliat în secțiunea 1.2, dedicată aspectelor de dezvoltare, cel mai adecvat model pentru acest proiect este modelul secvențial liniar, sau modelul cascadă. Examinând caracteristicile acestui model, observăm că procesul de dezvoltare a software-ului este structurat în etape distincte, fiecare etapă generând rezultate ce servesc drept intrare pentru următoarea.

Voi prezenta acum succint etapele principale ale modelului cascadă. În primul stadiu, cel de analiză a cerințelor, se colectează toate specificațiile necesare de la părțile interesate pentru a defini modul în care acestea vor fi integrate în aplicație. Cerințele includ un pachet de funcționalități și limitări așteptate de către client, fiind formulate de către medicul dentist titular al cabinetului, dar și de utilizatorii potențiali ai site-ului. La finalul acestei faze, se elaborează un document descriptiv al cerințelor care va fundamenta etapa următoare.

Etapa de proiectare implică mai multe sub-etape. Inițial, ne concentrăm pe diverse atribute ale sistemului, cum ar fi structura datelor, arhitectura, interfața utilizator și algoritmi specifici. În acest stadiu, cerințele sunt transpuse în reprezentări ale soluției software, care sunt evaluate pentru calitate înainte de a începe codificarea propriu-zisă. Scopul este de a verifica dacă proiectul îndeplinește nevoile utilizatorilor și este de înaltă calitate, evitând interpretările greșite ale cerințelor.



Figură 9. Modelul cascasdă

Faza de implementare utilizează designul finalizat ca intrare și transformă descrierile tehnice în cod sursă, care ulterior poate fi compilat și executat. Aici, codul este elaborat astfel încât să fie clar și modificabil, fiecare segment fiind testat pentru a identifica posibile erori și a asigura coerența cu specificațiile de proiectare.

Testarea joacă un rol crucial în validarea calității software-ului. Testele cuprinzătoare asigură că atât codul, cât și funcționalitățile și interfața sistemului îndeplinesc cerințele stabilite.

Ultima fază, mentenanța, presupune utilizarea efectivă a sistemului de către client și este continuă. Problemele care apar post-lansare, care nu au fost identificate în etapele anterioare, necesită intervenții rapide pentru corectare. De asemenea, pot exista solicitări pentru funcționalități noi sau ajustări ale celor existente, ce vor fi gestionate în această etapă.

Modelul cascadă are avantaje evidente, cum ar fi controlul riguros asupra procesului și o documentare amănunțită, ceea ce îl face potrivit pentru proiecte de dimensiuni reduse cu cerințe clar definite. Testarea este facilitată de existența unor scenarii bine stabilite bazate pe cerințe, iar riscurile legate de design sunt minimizate prin identificarea lor timpurie. De asemenea, costurile totale ale proiectului pot fi estimate încă din faza de analiză.

Principalul dezavantaj este rigiditatea: odată stabilite, cerințele nu pot fi modificate ușor pe parcursul dezvoltării, ceea ce poate fi problematic dacă beneficiarii își exprimă dificil nevoile inițiale. Acest aspect poate extinde durata dezvoltării, deoarece fiecare etapă trebuie completată înainte de a trece la următoarea.

**Glosar**

1. Ajax = o metodă de dezvoltare web ce folosește JavaScript asincron și XML pentru a permite aplicațiilor web să interacționeze cu serverul fără a reîncărca pagina.
2. Bootstrap = un framework creat inițial de Twitter și acum la versiunea sa a treia, facilitează designul site-urilor adaptabile la diferite dimensiuni de ecran, cum ar fi cele ale desktopurilor, tabletelor și telefoanelor mobile.
3. Feedback-ul = descrie procesul de primire a reacțiilor la un mesaj inițial, permițând emitentului să înțeleagă cum a fost perceput și interpretat mesajul său, și oferă informații despre efectele comportamentului său, sugerând modificări posibile.
4. JQuery = este o bibliotecă JavaScript destinată simplificării traversării documentelor HTML, gestionării evenimentelor și animațiilor, precum și realizării de cereri AJAX, concepută pentru a maximiza compatibilitatea între diferite browsere.
5. MVC = este un cadru de lucru în cadrul ASP.NET, care folosește arhitectura Model-View-Controller pentru dezvoltarea aplicațiilor web. Acesta permite separarea datelor (Model), interfeței utilizator (View) și logicii de interacțiune (Controller), facilitând gestionarea aplicațiilor complexe.
6. Mentenabilitatea = se referă la capacitatea unui sistem de a fi întreținut sau restaurat rapid la funcționalitatea optimă, implicând activități de mentenanță necesare pentru a păstra sau restabili starea de funcționare.
7. Procedură = denotă diverse metode utilizate în tratamente medicale, exemplificând cu serviciul de implantologie care include proceduri precum implantul dentar Nobel Biocare, adiția de os Bio-Oss sau coroanele integrale ceramice.
8. Programarea online = procesul prin care pacienții stabilesc o întâlnire la cabinetul medical folosind un formular web pentru a specifica data, ora și serviciul medical solicitat, eliminând necesitatea contactului telefonic direct.
9. Utilizabilitatea = ușurința cu care un produs software poate fi înțeles, învățat, folosit și apreciat de utilizatori în condiții specifice de utilizare.
10. Serviciul stomatologic = diverse tratamente medicale dentare disponibile la un cabinet stomatologic, incluzând proceduri precum endodonția, implantologia, tratamentele estetice sau ortodonția.

# **II . Proiectarea sistemului informatic**

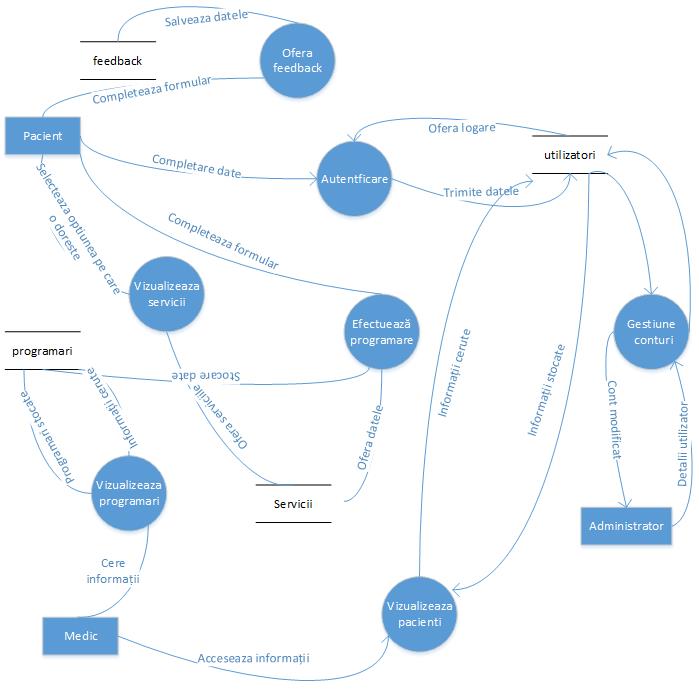
# **4.** **Proiectarea logică**

Am decis să adoptăm un sistem centralizat de gestionare a datelor, unde atât informațiile cât și sistemul de administrare sunt găzduite pe un singur computer. Acest aranjament asigură o integritate ridicată a datelor, minimizând redundanțele (prin evitarea duplicării informațiilor în multiple locuri). De asemenea, facilitează consistența în prelucrarea datelor și gestiunea eficientă a tranzacțiilor. Aceste aspecte constituie punctul de plecare în proiectarea logică a sistemului nostru.

**Diagrama flux-ului de date**

Diagrama fluxului de date (DFD) ilustrează grafic modul în care informațiile se deplasează și sunt procesate într-un sistem. Aceasta diagramă este utilizată pentru a arăta clar procesarea datelor. Elementele incluse în această reprezentare sunt procesele, care descriu activități specifice necesare sistemului pentru a funcționa, transformând intrările în rezultate. De asemenea, diagrama include surse de date, ce reprezintă entitățile externe care interacționează cu sistemul, și depozitele de date, care păstrează informațiile esențiale pentru procesare.

Diagrama prezentată ilustrează fluxul de date într-un sistem medical. Pacienții completează formulare, oferă feedback și se autentifică pentru a accesa servicii, cum ar fi vizualizarea și programarea consultațiilor. Medicii pot accesa și cere informații legate de programări, în timp ce administratorii gestionează conturile utilizatorilor și vizualizează detalii despre pacienți, asigurându-se că datele sunt transmise corespunzător între diferitele entități ale sistemului.



*Figură 10. Diagrama de flux de date*

# **4.1 Arhitectura sistemului**

Sistemul adoptă o arhitectură client/server structurată pe trei nivele: nivelul client sau de prezentare, nivelul de logică a aplicației sau serverul de aplicații, și nivelul de gestionare a datelor. Alegerea acestei arhitecturi este motivată de avantajele semnificative pe care le oferă în termeni de performanță, scalabilitate, flexibilitate, ușurință de întreținere și capacitatea de reutilizare a componentelor față de alte modele.



*Figură 11. Diagrama client-server*

Arhitectura pe trei niveluri a unui sistem client/server este structurată astfel: primul strat este dedicat interfeței cu utilizatorul, gestionând ferestrele de dialog și introducerea datelor, cunoscut sub numele de logica de prezentare. Acest nivel funcționează direct pe dispozitivul utilizatorului, adesea printr-un browser web.

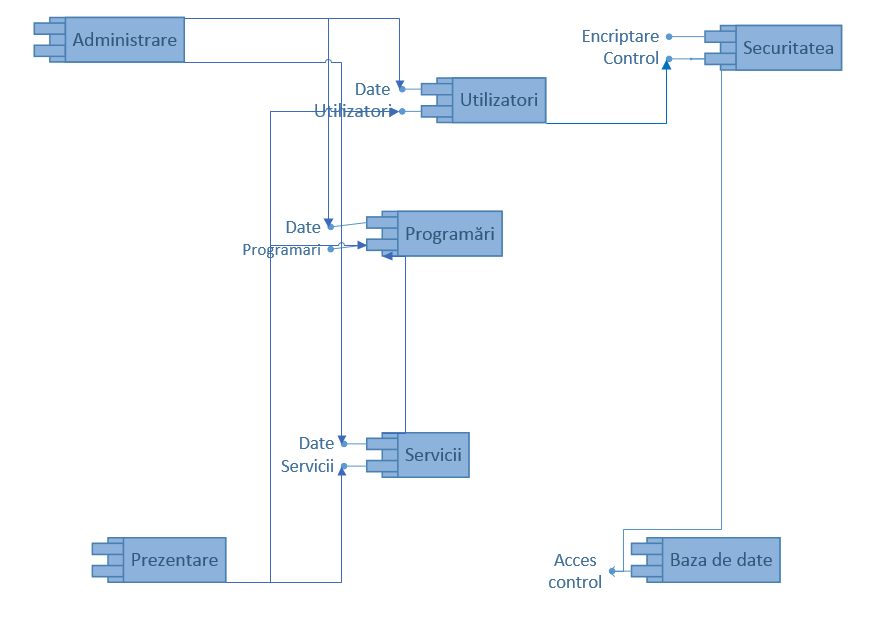
Cel de-al doilea strat, numit serverul de aplicații, centralizează procesarea datelor. Acest nivel intermediar facilitează administrarea eficientă a modificărilor sistemului, permițând actualizări centralizate care sunt apoi diseminate în întregul sistem.

Ultimul nivel este cel al gestionării bazelor de date, unde datele sunt stocate și prelucrate. Acest nivel se ocupă de consistența datelor, replicarea și securitatea acestora, prin utilizarea unor tehnici precum blocarea și validarea datelor.

Diagrama componentelor ilustrează interacțiunile dintre diferitele părți ale sistemului. Fiecare componentă este autonomă și oferă funcționalități specifice altor componente prin diverse interfețe. În cadrul acestui sistem, am identificat următoarele componente principale:

* Utilizatori – responsabilă de interacțiunea cu utilizatorii și activitățile pe care aceștia le pot efectua.
* Servicii – include detalii despre serviciile disponibile în aplicație și modul în care sunt utilizate.
* Programări – administrează tot ce implică planificarea și organizarea programărilor.
* Administrare – această componentă gestionează aspectele administrative și are acces la toate datele furnizate de celelalte componente.
* Prezentare – se ocupă exclusiv de interfata utilizatorului, accesând doar elementele de prezentare ale altor componente din motive de securitate.
* Baza de date – stochează și gestionează toate datele sistemului.
* Securitate – o componentă esențială care include interfețe pentru criptare și controlul accesului, legată de componenta „Utilizatori” pentru a reglementa accesul la date.

Această structură permite o mai bună organizare și o gestionare eficientă a resurselor și datelor în cadrul sistemului.



Figură 12. Diagrama de componente

## **4.2 Baza informațională**

În urma evaluării efectuate, am decis să organizăm informațiile în sistem astfel încât să sprijinim eficient funcțiile principale dorite de beneficiar. Pentru modulul de autentificare, vom include detalii despre rolurile atribuite fiecărui utilizator, nivelurile de acces corespunzătoare și informații personale esențiale pentru personalul medical.

Pentru descrierea serviciilor oferite de cabinet, vom diferenția între varietatea de servicii disponibile și procedurile specifice asociate, furnizând detalii vitale pe care pacienții trebuie să le înțeleagă. Segmentul de programări va reuni elemente din toate sectoarele discutate, oferind utilizatorilor posibilitatea de a introduce informații referitoare la data și ora dorite pentru programare.

De asemenea, vom implementa un mecanism de feedback, permițând pacienților să evalueze serviciile prin atribuirea de stele bazate pe diverse criterii definite și salvate în baza de date. Această abordare va contribui la ameliorarea continuă a calității serviciilor prestate.

Organizarea strategică a acestor date va îmbunătăți interacțiunea între pacienți și cabinet, asigurând accesibilitate și gestionare eficientă a informațiilor, ceea ce va facilita o experiență utilizator superioară și o administrare mai simplă a datelor din cadrul sistemului.

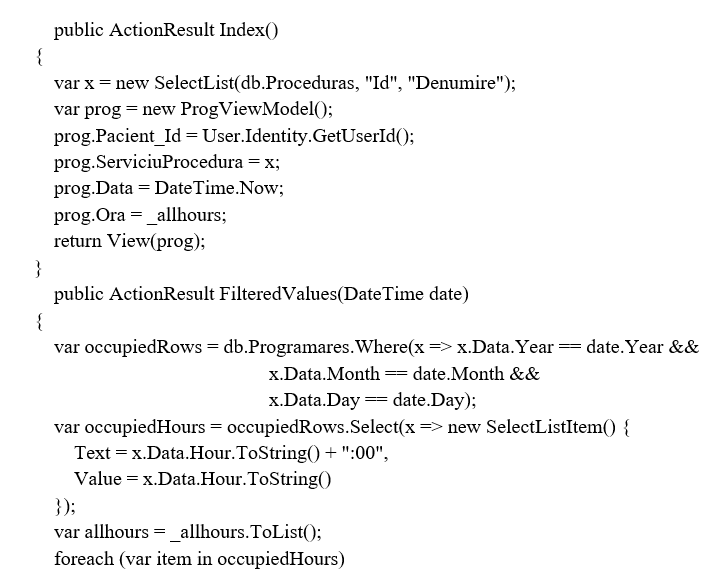
## **5. Proiectarea tehnică**

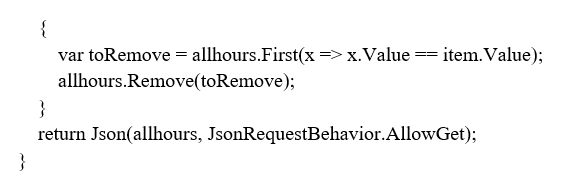
Acest capitol, intitulat "Proiectarea Tehnică", descrie arhitectura detaliată a aplicației mele, punând accent pe funcționalitatea principală de programare online. Detaliem modul în care utilizatorii pot selecta și programa servicii medicale folosind un formular interactiv, și cum aceste date sunt gestionate și stocate. De asemenea, explicăm structura informațiilor legate de servicii, proceduri și prețuri, oferind o perspectivă clară asupra procesului tehnic din spatele interfeței utilizator.

## **5.1 Procese și algoritmi**

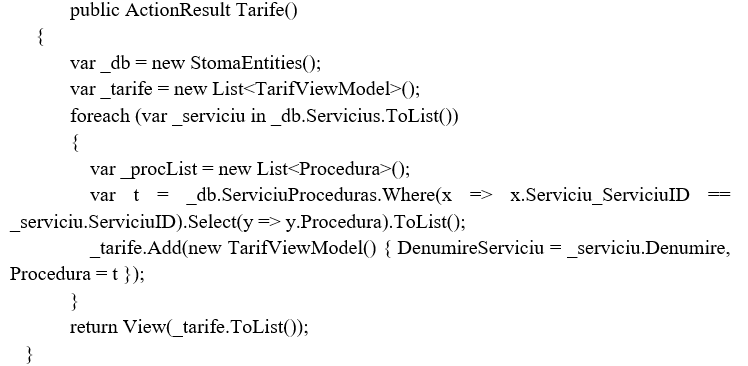
Aplicația mea se distinge prin funcționalitatea sa cheie de a permite programări online. În arhitectura aplicației, funcția Index din controller gestionează afișarea formularului pe website, unde utilizatorii pot selecta data, ora și procedura dorită. Aceste informații sunt ulterior salvate în baza de date utilizând metoda SaveProg.

Pentru a afișa orele disponibile, am dezvoltat două liste: una cuprinde orele standard de lucru ale medicului, iar cealaltă include orele deja rezervate pentru o dată specifică. Prin compararea celor două liste, eliminăm orele ocupate, permițând afișarea celor libere pentru programare.

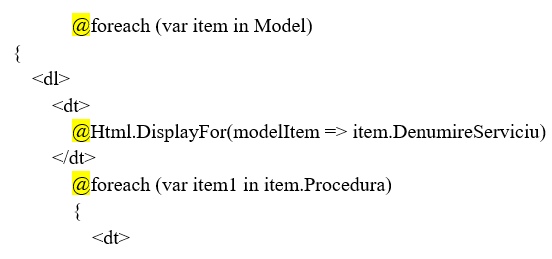


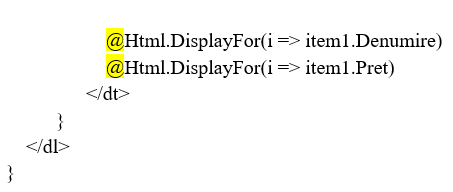


O altă funcționalitate importantă este prezentarea detaliată a procedurilor, serviciilor și prețurilor asociate. Am creat un model de vizualizare care include denumirea fiecărui serviciu și o listă a procedurilor asociate, fiecare având prețul specificat în baza de date.



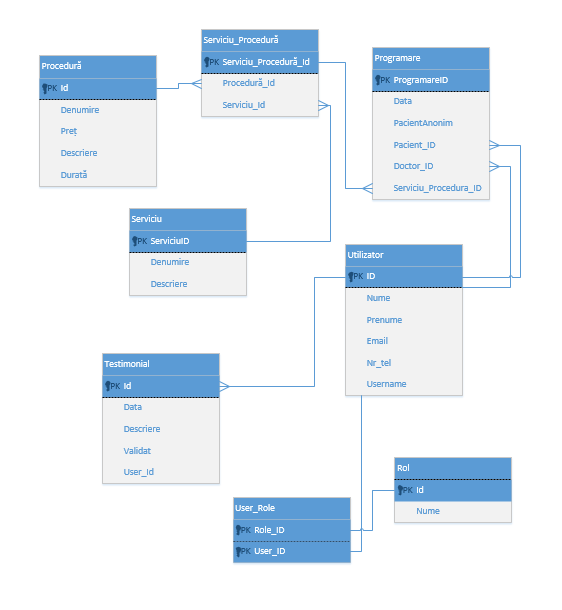
Metoda din controller procesează această listă de servicii, extrăgând procedurile corespunzătoare pentru fiecare serviciu. Aceste detalii sunt apoi afișate în pagina web, unde fiecare serviciu este listat alături de procedurile și prețurile respective.





## **5.2 Structura fizică a datelor**

În ilustrația următoare este prezentată structura bazei de date, evidențiind configurația fiecărei tabele. Conform detaliilor furnizate anterior în secțiunea dedicată informațiilor din baza de date, tabelele sunt organizate într-un mod care permite efectuarea eficientă a operațiunilor de manipulare a datelor, în concordanță cu necesitățile beneficiarului. Relațiile dintre tabele sunt stabilite prin intermediul cheilor primare și secundare, iar tipurile de date utilizate cel mai frecvent includ varchar, integer și date, pentru a se potrivi cu specificațiile definite.



Figură 13. Structura bazei de date

## **5.3 Tehnologii specifice**

Am optat pentru framework-ul .NET în dezvoltarea aplicației mele web, având în vedere că acesta furnizează o infrastructură coerentă și versatilă, adaptabilă atât pentru aplicații desktop Windows cât și pentru soluții web. Framework-ul .NET integrează o gamă largă de tehnologii și limbaje de programare, printre care ASP, XML, OOP, LINQ și C#, facilitând astfel reutilizarea eficientă a codului și asigurându-i portabilitatea pe diverse platforme Windows. Această flexibilitate permite dezvoltatorilor să creeze aplicații robuste, scalabile și ușor de întreținut.

Am ales limbajul C# datorită adaptabilității și puterii sale în implementarea principiilor de programare orientată pe obiect, cum sunt încapsularea, moștenirea și polimorfismul. C# îmbunătățește și extinde caracteristicile limbajului C++, adăugând construcții de limbaj care favorizează scrierea de cod sigur și eficient, reducând riscurile de erori comune în C++. Acesta include gestionarea automată a memoriei, colectarea gunoiului și tipuri de date sigure, făcându-l ideal pentru dezvoltarea de software enterprise.

Platforma ASP.NET MVC este baza pentru structura aplicației mele web. Separând aplicația în trei componente principale - model, view și controller - ASP.NET MVC nu doar că simplifică dezvoltarea prin clarificarea rolurilor fiecărui segment, dar și îmbunătățește testabilitatea și întreținerea aplicației. Oferă o alternativă robustă la pattern-ul Web Forms, sprijinind dezvoltarea de interfețe utilizator dinamice și reutilizabile, și integrează funcționalități avansate ASP.NET, cum ar fi autentificarea utilizatorilor și gestionarea sesiunilor.

Pentru gestionarea datelor, Microsoft SQL Server a fost alegerea evidentă, datorită performanței și fiabilității sale în manipularea bazelor de date de mari dimensiuni. SQL Server, suportând standardul de interogare SQL, este un pilon pentru soluții de baze de date enterprise, oferind capabilități extinse pentru gestionarea datelor. Versiunea 2008 a introdus funcționalități inovative, precum suportul pentru stocarea și gestionarea datelor multimedia și noi tipuri de date, inclusiv geometry și geography, care permit o manipulare mai precisă și eficientă a datelor spațiale și geografice, deschizând noi perspective pentru dezvoltarea de aplicații avansate în diverse domenii, de la GIS la analiza datelor.

# 

# **III.TESTAREA**

## **Testarea**

Testarea reprezintă o etapă crucială în finalizarea oricărui proiect, esențială pentru evaluarea modularității, fluxurilor de date și a codului. Aceasta analizează complexitatea sistemului în dezvoltare. O validare amănunțită a fost realizată pentru toate componentele critice. Această fază este fundamentală în dezvoltarea sistemului, garantând calitatea serviciilor, designul și codul programului.

**Obiectivele Testării**

Scopul principal al testării este identificarea erorilor prin executarea programelor. Aceasta vizează proiectarea de teste eficiente, capabile să descopere diverse tipuri de erori în cel mai scurt timp posibil. Testele validează faptul că funcționalitățile sistemului corespund cerințelor specificate și că performanța este la nivelul așteptat.

**Metodologia Testării**

Metodologia testării este structurată pe mai multe niveluri pentru a asigura acoperirea completă a aplicației:

1. **Testarea Blackbox**: La acest nivel, testarea se concentrează pe experiența utilizatorului, evaluând toate funcționalitățile aplicației pentru a asigura că sunt conform așteptărilor.
2. **Testarea Whitebox**: Aceasta implică un examen detaliat al fiecărui modul al aplicației. Testarea începe de la verificarea componentelor individuale și progresează spre teste integrate, pentru a evalua funcționarea armonioasă a întregului sistem.
3. **Testarea Componentelor**: Focalizată pe cele mai mici unități ale aplicației, aceasta etapă urmează după ce codul a fost dezvoltat și verificat pentru a se asigura că este conform cu designul la nivel de componentă. Fiecare componentă este testată individual pentru a verifica funcționalitatea corectă.
4. **Testarea Integrată**: După testarea componentelor individuale, este esențial să se testeze cum funcționează acestea împreună. Această etapă vizează dezvoltarea unei strategii care confirmă integrarea eficientă a tuturor componentelor în sistem.

Aceste metode de testare contribuie la asigurarea unui produs final de înaltă calitate, fiabil și performant, respectând toate cerințele inițiale.

### **6.1 Testarea manuală**

Pentru a asigura o testare eficientă, este esențial să planificăm meticolos activitățile de testare încă de la etapele inițiale de proiectare sau dezvoltare a software-ului. Pe parcursul definirii entităților de date, modulelor, obiectelor, claselor și funcțiilor, este prudent să elaborăm "scenarii" de testare pentru fiecare, cunoscute în domeniul software ca test cases.

Am pregătit un test case specific pentru fiecare dintre următoarele module:

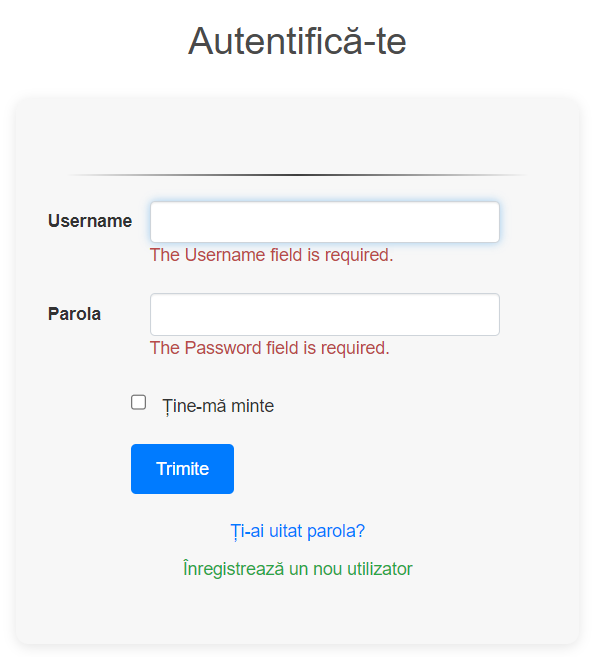
1. Autentificarea utilizatorilor
2. Înregistrarea noilor utilizatori
3. Realizarea programărilor

**Autentificarea utilizatorilor**

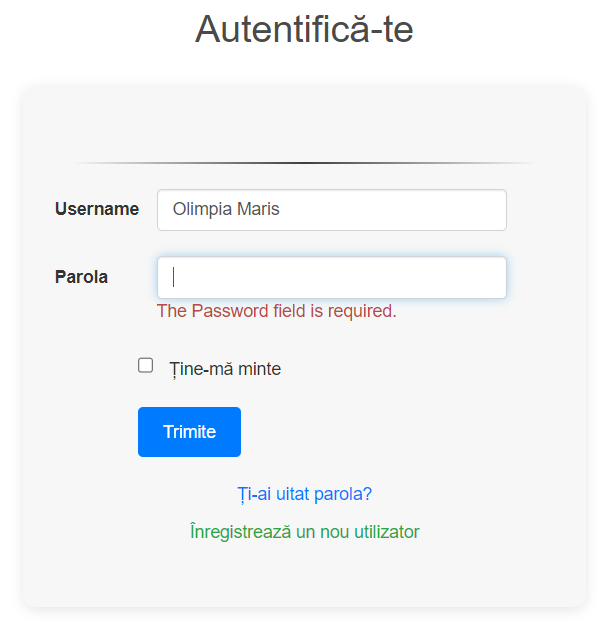
1. **Accesare buton "Autentificare"** - Selectarea opțiunii „Autentificare” din meniul principal trebuie să redirecționeze către pagina de autentificare pentru completarea datelor.



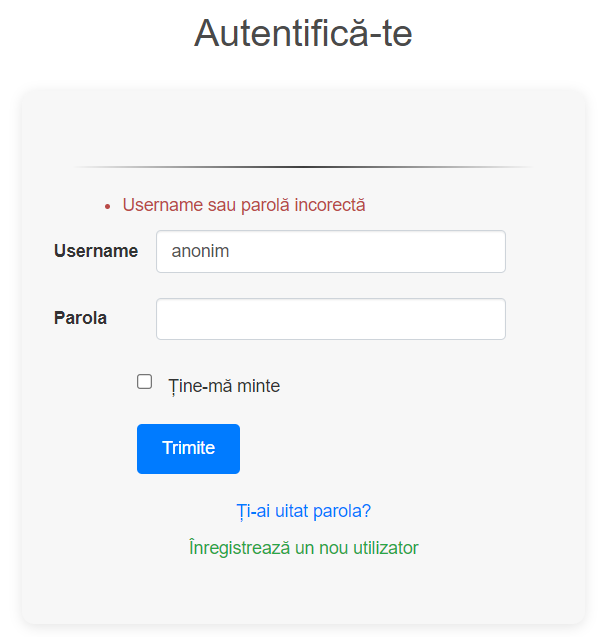
1. **Trimitere formular gol** - Dacă se apasă „Trimite” fără a introduce numele de utilizator sau parola, pagina nu ar trebui să se schimbe, iar un mesaj de eroare ar trebui să apară pentru fiecare câmp necompletat.



1. **Introducere parolă incorectă sau absentă** - La introducerea corectă a numelui de utilizator, dar fără parolă sau cu o parolă greșită, și apoi apăsând pe „Trimite”, pagina ar trebui să rămână aceeași, afișând un mesaj de eroare corespunzător.



1. **Eroare la username sau parolă** - Dacă se introduce un username sau o parolă incorecte, sau ambele, și se selectează „Trimite”, utilizatorul ar trebui să rămână pe aceeași pagină, cu mesaje de eroare afișate pentru datele introduse greșit.



1. **Autentificare reușită** - Cu un username și o parolă corecte, după apăsarea butonului „Trimite”, trebui să fie redirecționat către pagina principală, unde în colțul dreapta sus apare un mesaj de salut personalizat cu numele de utilizator.

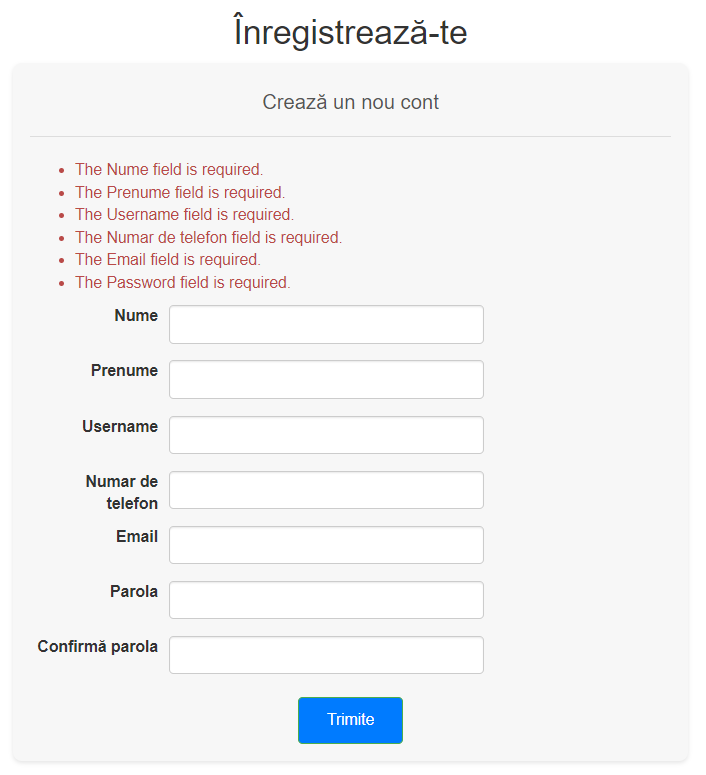


**Înregistrarea noilor utilizatori**

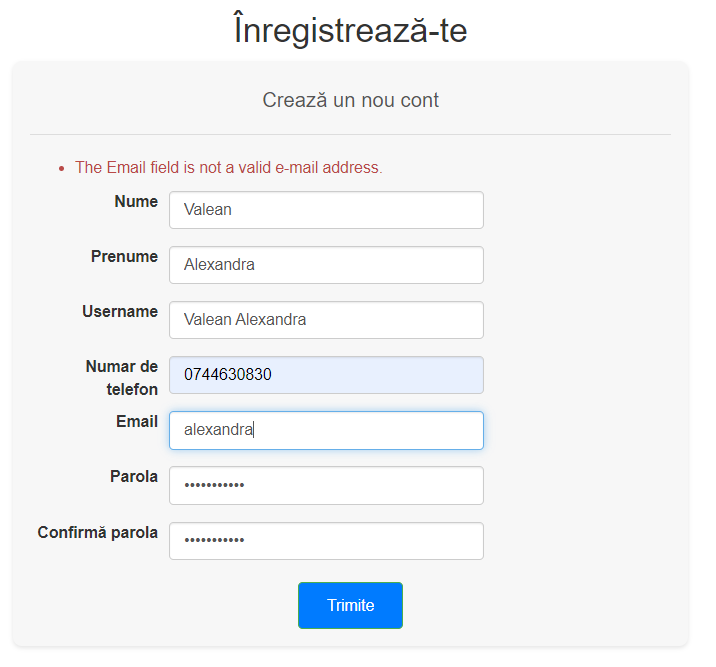
1. **Accesare opțiune înregistrare** - Selectând „Înregistrare” din meniul navigațional, sistemul ar trebui să mă direcționeze către formularul unde pot introduce datele necesare.



1. **Trimitere formular incomplet** - Apăsarea butonului „Trimite” fără a completa toate câmpurile necesare nu ar trebui să schimbe pagina, ci se va afișa un mesaj de eroare indicând câmpurile care necesită completare.



1. **Eroare la completare** - Dacă unul dintre câmpuri este completat greșit și se apasă „Trimite”, pagina ar trebui să rămână neschimbată, cu un mesaj de eroare vizibil pentru câmpul respectiv.

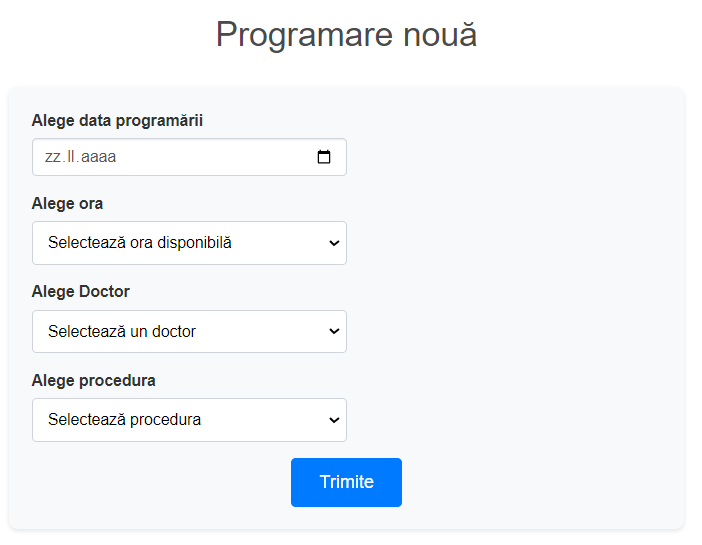


1. **Înregistrare cu succes** - După completarea corectă a tuturor datelor și apăsarea butonului „Trimite”, utilizatorul ar trebui să fie redirecționat către pagina principală, unde un mesaj de salutare personalizat cu numele de utilizator ar trebui să fie vizibil în colțul drept sus al paginii, indicând că autentificarea a fost realizată automat cu datele nou introduse.

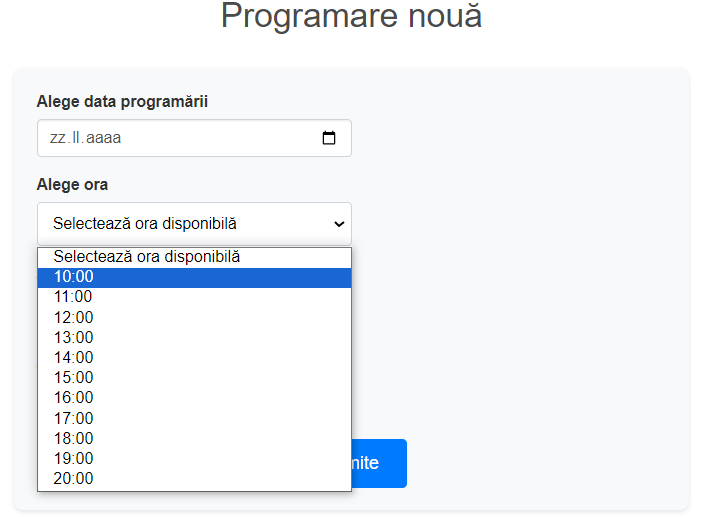


**Realizarea programărilor**

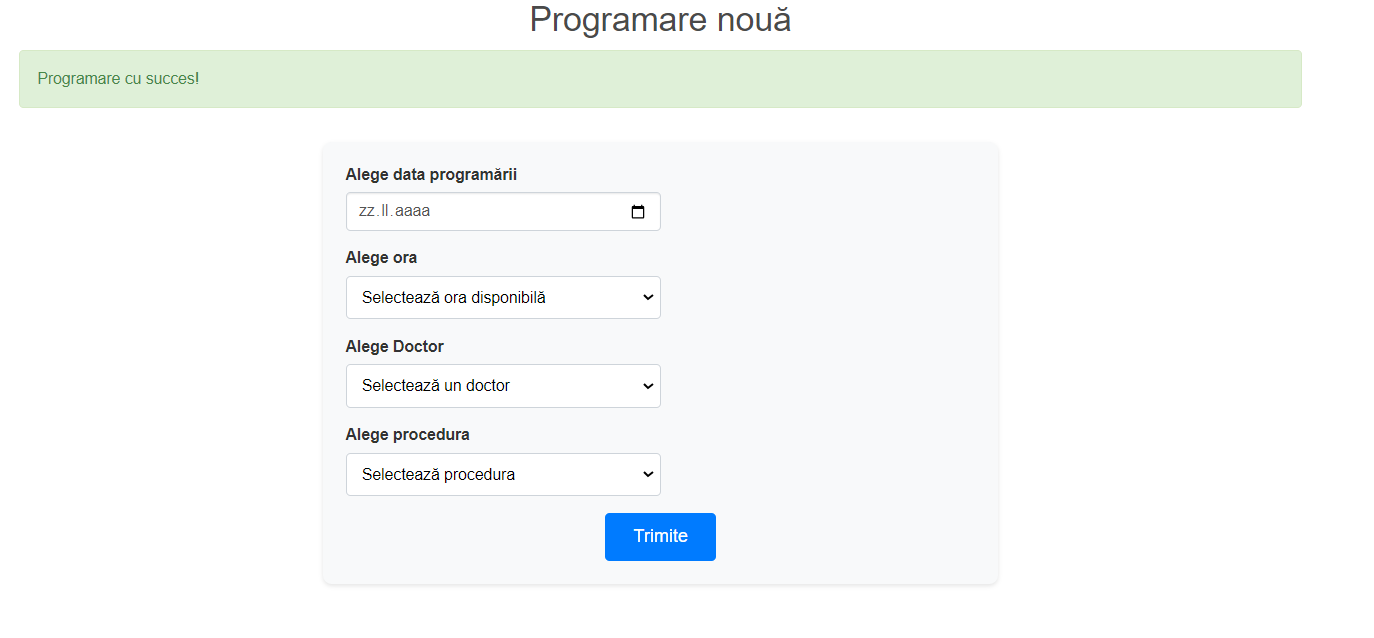
1. **Accesarea formularului de programare** - La selectarea opțiunii „Programare” din meniu, utilizatorul este redirecționat către pagina de completare a formularului. Aici, selectând câmpul pentru dată, se va deschide un calendar pentru a alege ziua dorită, iar la alegerea orei, se va afișa o listă cu orele disponibile ale medicului.



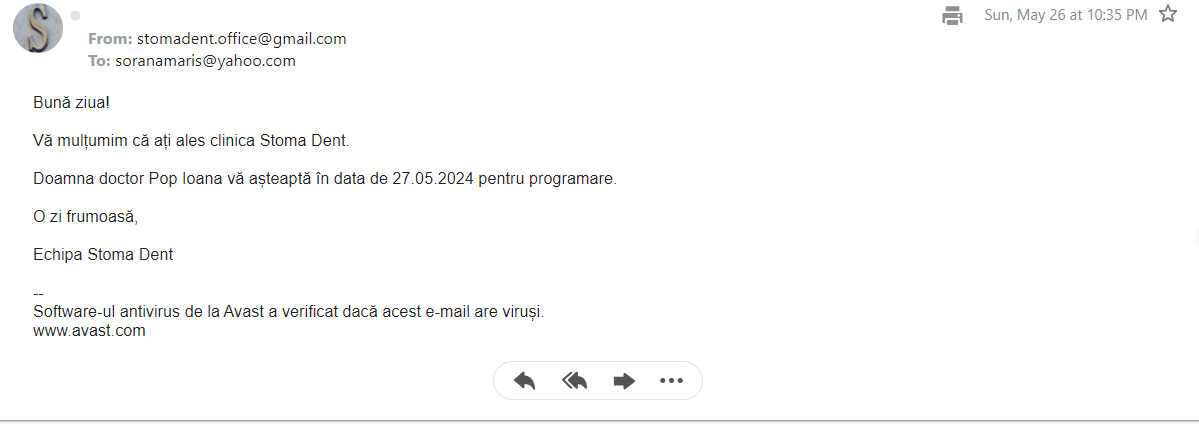
1. **Selecție dată, procedura și medic** - După alegerea datei, utilizatorul poate selecta un medic, iar în funcție de alegerea făcută, lista de ore disponibile va fi actualizată pentru a reflecta disponibilitatea medicului în ziua aleasă.



1. **Completare și confirmare înregistrare programare** - După completarea tuturor câmpurilor necesare și apăsarea butonului „Trimite”, informațiile despre programare sunt salvate. Un mesaj de confirmare că programarea a fost realizată cu succes va apărea direct pe pagina curentă.



1. **Primirea confirmării pe email** - Ulterior, clientul va primi un email de confirmare care include detaliile programării, asigurând comunicarea eficientă a tuturor informațiilor necesare.



# 

# **IV .MENTENANȚĂ**

**8. Operații de întreținere**

Întreținerea unui site web este un proces esențial pentru asigurarea funcționării fluente a unei aplicații. Această activitate poate fi efectuată periodic sau după cum necesită situația. Unul dintre principalele obiective ale întreținerii este actualizarea site-ului pentru a menține informațiile relevante și pentru a garanta că tehnologia folosită este la zi. Aceasta ajută la securizarea site-ului și permite utilizatorilor să fie la curent cu ultimele noutăți.

Serviciile de întreținere cuprind:

* Gestionarea bazelor de date
* Adăugarea pe site a serviciilor noi oferite de firmă, inclusiv informații, promoții, fotografii și galerii foto
* Verificarea corectitudinii funcționării site-ului
* Implementarea ultimelor tehnologii web
* Înregistrarea site-ului în directoare web specializate

Pentru întreținerea site-ului nostru, avem în vedere următoarele elemente:

* Pe viitor, este posibil ca medicul să opteze pentru un redesign al paginilor, dorind o interfață mai accesibilă și atractivă. Putem include efecte vizuale inovative și mai multe imagini reprezentative pentru cabinet sau tratamentele oferite.
* Având în vedere că în momentul de față avem un singur medic, expansiunea serviciilor prin angajarea altor specialiști ar necesita modificări la nivel de back-end, cum ar fi actualizările bazei de date.
* Dacă site-ul devine popular și multe programări se fac online, există posibilitatea extinderii pe dispozitive mobile. Echipa noastră ar putea să dezvolte aplicația pentru platformele iOS și Android, oferind astfel accesibilitate crescută.

# **Concluzii**

Exploatarea potențialului Internetului este esențială pentru succesul oricărei afaceri noi, mai ales într-un mediu competitiv unde prezența online este decisivă. Un website, odată creat, rămâne accesibil permanent, oferind informații despre servicii la orice oră din zi și din orice colț al lumii. Acestea pot fi actualizate sau modificate oricând, în funcție de nevoile beneficiarului.

Proiectarea unei aplicații web pentru un cabinet stomatologic a reprezentat scopul principal al acestei lucrări, având ca obiectiv transformarea metodelor de gestionare a cabinetului și îmbunătățirea prezentării serviciilor oferite. Am dezvoltat un sistem administrativ prin care medicul poate eficientiza gestionarea serviciilor și a pacienților, precum și un modul de interfață utilizator care permite pacienților să navigheze ușor pe site pentru a găsi informații, a face programări online și a oferi feedback.

Elaborarea acestei lucrări de licență a consolidat cunoștințele acumulate pe parcursul studiilor universitare, îmbinând teoria cu practica și permițându-mi să aplic diverse tehnologii învățate. Procesul de dezvoltare a aplicației de la analiza inițială până la implementarea finală mi-a arătat cât de importantă este definirea clară a obiectivelor și cerințelor și cum acestea influențează designul proiectului. Testarea, de asemenea, a avut un rol crucial în verificarea funcționalităților și stabilitatea aplicației.

În final, experiența dobândită în gestionarea timpului și în depășirea obstacolelor întâmpinate în timpul dezvoltării software-ului m-a învățat că estimarea exactă a duratei necesare pentru finalizarea unei aplicații poate fi adesea imprevizibilă. Această experiență a fost valoroasă pentru înțelegerea completă a procesului de creare a unei aplicații web, îndeplinind toți pașii necesari pentru a satisface cerințele beneficiarului și pentru a atinge obiectivele stabilite.

# **Bibliografie**

* Adam Freeman “[Pro ASP.NET Core MVC 2](https://www.amazon.in/Pro-ASP-NET-Core-MVC-2/dp/1484241118/ref=sr_1_1?crid=3VXWM238JX4KZ&keywords=Pro+ASP.NET+Core+MVC+2&qid=1680708141&s=books&sprefix=head+first+c%23+a+learner+s+guide+to+real-world+programming+with+c%23%2C+xaml%2C+and+.net%2Cstripbooks%2C539&sr=1-1)”
* [Christian Nagel](https://www.libris.ro/librarie-online?fsv_77564=Christian%20Nagel) “Professional C# and .Net”
* [Daniela Alexandra Crișan](https://www.ujmag.ro/autori/daniela-alexandra-crisan) . “Programarea aplicatiilor folosind limbajul C# si platforma .NET”.
* Joseph Albahari “[C# 9.0 in a Nutshell: The Definitive Reference](https://www.amazon.in/9-0-Nutshell-Definitive-Reference-Grayscale/dp/8194722926/ref=sr_1_1?crid=1Q5QR95KATG3B&keywords=%22C%23+9.0+in+a+Nutshell%3A+The+Definitive+Reference&qid=1680700079&s=books&sprefix=c%23+9.0+in+a+nutshell+the+definitive+reference%2Cstripbooks%2C323&sr=1-1)”
* [Mark J. Price](https://www.kobo.com/ro/ro/author/mark-j-price). “Apps and Services with .NET 8”.
* Robert Martin “[Clean Architecture: A Craftsman’s Guide to Software Structure and Design](https://www.amazon.in/Clean-Architecture-Craftsmans-Software-Structure/dp/0134494164/ref=sr_1_1?keywords=Clean+Architecture%3A+A+Craftsman%27s+Guide+to+Software+Structure+and+Design&qid=1680707677&s=books&sr=1-1)”
* Documentația oficială Microsoft Visual Studio 2013
* „Code Behind vs. Code Inline”, Microsoft .NET Framework, Microsoft.
* “Web Application Architecture”, principles, protocols and practices by Leon Shklar, Richar Rosen
* <http://www.rasfoiesc.com/educatie/informatica/internet/Medii-de-dezvoltare-web-NET-SI53.php>
* <https://learn.microsoft.com/en-us/dotnet/csharp/>