



UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA

DE

MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ



LUCRARE DE LICENȚĂ

COORDONATOR ȘTIINȚIFIC
Lect. Dr. Sorina-Nicoleta Preduț

ABSOLVENT
Andreea Miruna Postolache

BUCUREȘTI

2023



UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA

DE

MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ



TRI România -

Platformă de Administrare a Federației de Triatlon și

Monitorizare a Performanțelor Sportivilor

COORDONATOR ȘTIINȚIFIC
Lect. Dr. Sorina-Nicoleta Preduț

ABSOLVENT
Andreea Miruna Postolache

BUCUREȘTI

2023

Rezumat

„TRI România” este o aplicație web prin intermediul căreia se gestionează activitatea Federației Române de Triatlon, monitorizând, de asemenea, performanțele sportivilor. Utilizatorii autentificați au acces la toate competițiile de triatlon organizate, la rezultatele obținute de participanți, la recenziile sportivilor despre concursuri și la profilurile tuturor persoanelor înregistrate pe platformă.

Pentru a oferi un mediu prietenos și interactiv, aplicația permite utilizatorilor logați să publice postări la care pot reacționa și scrie comentarii. De asemenea, aceștia au la dispoziție o pagină de chat în timp real unde pot alege să vorbească cu orice persoană doresc.

O altă funcționalitate importantă este înscrierea sportivilor la competiții de către antrenorul acestora direct din aplicație. Această acțiune este urmată de trimiterea unei notificări spre sportivul care a fost adăugat la concurs pentru a-l înștiința cu privire la acest aspect. Totodată, dacă sportivul este abonat la știri, va primi și pe email această informare.

Abstract

„TRI Romania” is a web application through which is managed the activity of the Romanian Triathlon Federation, also monitoring the athletes' performances. Authenticated users have access to all organized triathlon competitions, to the results obtained by the participants, to the athletes' reviews about the competitions and to the profiles of all people registered on the platform.

To provide a friendly and interactive environment, the application allows logged-in users to publish posts to which they can react and write comments. They also have a live chat page where they can choose to talk to anyone they want.

Another important functionality is the registration of athletes to competitions by their coach directly from the application. This action is followed by a notification being sent to the athlete who has been added to the competition to inform him or her about this. Also, if the athlete is subscribed to the news, he will also receive this information by email.

Cuprins

CAPITOLUL 1 – INTRODUCERE.....	8
1.1. Motivația alegerii temei	8
1.2. Scopul aplicației	8
1.3. Aplicații similare	8
1.4. Contribuția proprie	10
1.5. Structura lucrării.....	11
CAPITOLUL 2 - TEHNOLOGII FOLOSITE	12
2.1. ASP.NET Core Web API	12
2.1.1.Date generale și utilizarea în aplicație	12
2.1.2.Trimiterea de email-uri.....	13
2.2. SignalR.....	15
2.2.1. Date generale.....	15
2.2.2. Utilizarea SignalR în aplicație.....	17
2.2.3. Motivație alegere SignalR	22
2.3. Angular.....	23
2.3.1. Date generale și utilizarea în aplicație	23
2.3.2. Motivație alegere Angular	26
2.4. Ng-Bootstrap	26
CAPITOLUL 3 – ARHITECTURA APLICAȚIEI	27
3.1. Prezentarea generală a arhitecturii	27
3.2. Baza de date	27
3.3. Backend.....	28

3.3.1.Entity Framework Core	28
3.3.2.ASP.NET Core Identity	29
3.3.3.Repository pattern.....	30
3.3.4.Arhitectura Traditional „N-Layer”	31
3.4. Frontend	32
3.4.1.Arhitectura interfeței.....	32
3.4.2.Grafice - Ng2-charts.....	33
3.4.3.Google API.....	35
3.4.4.YouTube Video Player.....	36
3.4.5.Fluxul de funcționare al interfeței	36
CAPITOLUL 4 – PREZENTAREA APLICAȚIEI	37
4.1. Înregistrare și autentificare.....	37
4.2. Interfața administratorului.....	39
4.2.1.Gestionarea videoclipurilor	39
4.2.2.Gestionarea cluburilor	40
4.2.3.Gestionarea probelor.....	41
4.2.4.Gestionarea competițiilor, rezultatelor și locațiilor acestora	41
4.2.5. Gestionarea tipurilor de competiții	43
4.2.6. Gestionarea recenziilor, comentariilor și postărilor.....	44
4.2.7. Gestionarea formularelor.....	44
4.3. Interfața utilizatorului neautentificat.....	44
4.3.1. Pagina principală.....	44
4.3.2. Pagina „Cluburi”.....	44
4.3.3. Pagina „Competiții”	45

4.3.4. Pagina „Statistici”.....	46
4.3.4.1. Afișarea topului de 5 sportivi.....	46
4.3.4.2. Afișarea topului de 5 cluburi.....	46
4.4. Interfața comună utilizatorilor cu rol de sportiv și antrenor.....	47
4.4.1. Pagina „Cluburi”.....	47
4.4.2. Pagini dedicate postărilor	47
4.4.2.1. Fluxul de postări	47
4.4.2.2. Postările proprii.....	48
4.4.3. Pagina de Chat.....	48
4.4.4. Pagina de Rezultate.....	50
4.4.5. Pagina de detalii specifică unui concurs.....	50
4.5. Interfața utilizatorului cu rol de sportiv.....	51
4.5.1. Pagina de detalii pentru competițiile proprii	51
4.5.2. Pagina notificărilor.....	51
4.5.3. Pagina de profil a sportivului.....	51
4.5.3.1. Profilul din perspectiva deținătorului.....	51
4.5.3.1.1. Editarea și datele profilului	51
4.5.3.1.2. Vizualizarea antrenorului	52
4.5.3.1.3. Completarea fișei anuale	52
4.5.3.1.4. Vizualizarea fișelor completate	52
4.5.3.1.5. Vizualizarea și editarea recenziilor.....	53
4.5.3.1.6. Best of și istoricul rezultatelor sportivului	53
4.5.3.2. Profilul sportivului din perspectiva publică.....	54
4.6. Interfața utilizatorului cu rol de antrenor	54

4.6.1. Înscrierea sportivilor la competiții.....	54
4.6.2. Pagina de profil a antrenorului	55
4.6.2.1. Profilul din perspectiva deținătorului.....	55
4.6.2.1.1. Editarea și datele profilului	55
4.6.2.1.2. Vizualizarea sportivilor proprii	56
4.6.2.1.3. Studiarea statisticilor	56
4.6.2.2. Profilul antrenorului din perspectiva publică	57
4.7. Meniul aplicației.....	57
CONCLUZII	58
SURSELE IMAGINILOR PRELUATE DIN APLICAȚIE	59
BIBLIOGRAFIE	61

CAPITOLUL 1 – INTRODUCERE

1.1. Motivația alegerii temei

Fiind o fire activă și dornică de mișcare, părinții au decis să mă introducă în lumea sportului încă de la vârsta de patru ani. De-a lungul timpului am practicat sporturi de performanță, precum: dansul modern, înotul și triatlonul. De asemenea, participarea la numeroase concursuri atât în țară, cât și în străinătate a avut un rol important în formarea ca sportiv, dar și ca individ. Pe parcursul carierei sportive am înțeles importanța disciplinei, mentalității puternice, dozarea efortului în vederea atingerii obiectivelor propuse. Totodată, am învățat să apreciez și să mă bucur pentru reușitele altora, să îi ajut pe cei care au nevoie și să îmi doresc să devin cea mai bună versiune a mea.

Alegerea temei de licență a fost un proces natural, deoarece din momentul în care am învățat să fac aplicații web mi-am dorit să îmbin pasiunea pentru sport cu cea pentru informatică și produsul final să aibă un impact pozitiv asupra iubitorilor de sport. Am optat să fac o aplicație web care să aibă în centrul atenției triatlonul, pentru că de când am trecut de la înot la această disciplină sportivă am observat anumite lipsuri pe care le are site-ul federației de triatlon comparativ cu cel de înot, elementele lipsă fiind utile pentru sportivi.

1.2. Scopul aplicației

„TRI România” este o aplicație web concepută cu scopul de a reuni întreaga comunitate de triatlon, oferindu-le sportivilor și antrenorilor posibilitatea de a urmări competițiile, evoluțiile sportivilor, de a interacționa cu postările utilizatorilor și de a comunica între ei.

1.3. Aplicații similare

- „Triatlon Romania”

„Triatlon Romania” este o aplicație web, reprezentând site-ul oficial al Federației Române de Triatlon. Navigarea pe acest site nu implică crearea unui cont, decât dacă utilizatorul dorește să facă o înscriere la o competiție sau să completeze formularul de

licențiere pentru care are nevoie de un cont de RaceHub – platformă care administrează competițiile sportive din mai multe arii de activitate.

De asemenea, această aplicație oferă utilizatorilor informații detaliate despre competițiile desfășurate în anul curent, precum și cele din anii precedenți. Aceste informații includ detalii despre locație, data de desfășurare, programul, categoriile și distanțele implicate în concurs. Totodată, utilizatorii au acces la rezultatele competițiilor, calendarul competițional, informații despre procesul de licențiere, aspectele organizatorice, cluburile afiliate, antrenori și sportivii care fac parte din lotul național. Mai mult, site-ul include secțiuni dedicate promovării fotografiilor și videoclipurilor de la diverse competiții, precum și articole legate de activitățile desfășurate de către federație.

- „AMNB”

„AMNB” reprezintă o aplicație web prin intermediul căreia Asociația Municipală de Natație București furnizează informații despre competiții, evoluția sportivilor, anunțuri, cluburi și statistici.

Prin intermediul acestei aplicații, utilizatorii pot accesa detalii referitoare la competiții, precum: locația, perioada de înscriere și cea de desfășurare, programul, lista de start, regulamentul, lista de rezultate, dar și topul cluburilor ordonate în funcție de numărul de puncte acumulate de sportivi în concurs.

De asemenea, „AMNB” prezintă informații despre cluburi: o scurtă descriere, detalii de contact și despre locație, sportivii afiliați și website-ul acestora.

În plus, aplicația oferă statistici care surprind: campionii asociației – cei mai buni sportivi în funcție de punctele obținute la competiții, topul cluburilor – în funcție de numărul de puncte obținute de sportivi și cei mai buni timpi înregistrați pentru fiecare probă.

Un alt aspect semnificativ al aplicației este capacitatea utilizatorului de a căuta sportivi, oferindu-i astfel posibilitatea de a accesa istoricul participărilor acestora la competiții, evoluția lor la diverse probe și cei mai buni timpi obținuți la fiecare probă în parte.

- „Cyprus Triathlon Federation”

„Cyprus Triathlon Federation” este platforma prin intermediul căreia Federația de Triatlon din Cipru distribuie utilizatorilor informații despre: calendarul competițional,

rezultatele obținute de sportivi, cluburile afiliate și noutăți, nefiind necesar crearea unui cont. De asemenea, include și o secțiune de fotografii din cadrul competițiilor.

Mai mult, utilizatorii pot afla detalii referitoare la competiții, precum: data de desfășurare, locația, informații privind înscrierea, regulile, dar și site-ul oficial al competiției.

1.4. Contribuția proprie

Spre deosebire de aplicațiile mai sus menționate, „TRI România” oferă utilizatorilor care au cont (sportivii și antrenorii acestora) posibilitatea de a comunica prin mesaje, de a crea postări – ceea ce implică și scrierea de comentarii, respectiv oferirea de reacții. În ceea ce privește sportivii, aceștia pot redacta recenzii pentru competițiile la care au luat parte, pot fi notificați atât în cadrul aplicației, cât și pe adresa de email după ce antrenorul îi înscrie la o competiție. De asemenea, antrenorii pot fi notificați pe adresa de email când administratorul introduce o nouă competiție.

Secțiunea de chat prezintă în aplicație a fost implementată folosind biblioteca SignalR care permite comunicarea în timp real. Pagina conține o listă a persoanelor care sunt online, dar și a celor care sunt offline, utilizatorul putând trimite mesaje ambelor categorii. Un individ este considerat online dacă acesta se află pe pagina de chat, altfel este considerat a fi offline. Un alt aspect important de menționat este faptul că user-ul poate vedea în bara de meniu numărul de mesaje necitite, iar la intrarea în secțiunea de chat în partea superioară va exista o listă a persoanelor care i-au trimis mesaje și care nu au primit răspuns, alături de numărul specific de mesaje necitite. Doar dacă utilizatorul va răspunde la mesaje, acestea vor fi considerate citite.

Un alt aspect important al platformei este faptul că antrenorul își poate înscrie sportivii la competițiile dorite direct din aplicație. Dacă înscrierea are loc cu succes, atunci sportivul va primi o notificare în aplicație prin care este informat la ce competiție și la ce probă a fost înscris. Dacă sportivul a selectat să primească notificări, atunci acesta va primi și pe email un mesaj care să îl atenționeze cu privire la competiția la care a fost înscris.

Mai mult, când administratorul aplicației introduce o nouă competiție atunci toți antrenorii care au selectat opțiunea să fie notificați vor primi un email prin care vor fi anunțați de introducerea noului concurs în calendarul competițional.

1.5. Structura lucrării

În ceea ce privește structura lucrării, aceasta conține patru capitole care la rândul lor pot avea un număr variabil de subcapitole. Primul capitol este cel introductiv care pune accentul pe scopul și motivul alegerii temei abordate și asupra contribuțiilor proprii. Următorul capitol cuprinde tehnologiile utilizate, fiecare având un subcapitol asociat în care se detaliază modul de funcționare, utilizarea în aplicație și motivul alegerii acestora. Cel de-al treilea capitol surprinde descrierea arhitecturii aplicației, focalizându-se asupra bazei de date și a părților de backend și frontend. Următorul capitol este cel în care se prezintă aplicația, fiind împărțit în subcapitole ce corespund unui tip de interfață. Mai mult, sunt descrise și metodele de înregistrare și autentificare, dar și meniul aplicației. La aceste capitole se adaugă concluziile, precizările surselor de unde au fost preluate imaginile utilizate în interfața aplicației și bibliografia.

CAPITOLUL 2 - TEHNOLOGII FOLOSITE

2.1. ASP.NET Core Web API

2.1.1. Date generale și utilizarea în aplicație

.NET este o platformă creată de Microsoft ce are ca scop dezvoltarea mai multor tipuri de aplicații. Aceasta îi permite dezvoltatorului să folosească unul dintre limbajele de programare, precum: C#, F#, Visual Basic, precum și să aleagă un editor de cod după bunul plac. Totodată, .NET pune la dispoziție o gamă variată de biblioteci care pot fi folosite în crearea de aplicații pentru web, mobile, jocuri și pentru alte domenii.[15]

ASP.NET este un framework web creat de Microsoft folosit pentru construirea de aplicații web și servicii, având la bază platforma .NET.

În anul 2016, a fost lansat framework-ul open-source, cross-platform ASP.NET Core care a presupus o reproiectare a versiunii anterioare de ASP.NET pe partea de Windows. Acesta a fost proiectat astfel încât să fie mai performant, permițând evoluția rapidă a: componentelor de runtime, API-urilor, compilatoarelor și limbajelor utilizate în dezvoltarea aplicațiilor. Mai mult, acest framework este stabil, asigurând funcționarea corespunzătoare a aplicațiilor indiferent de modificările, respectiv actualizările aduse elementelor enumerate mai sus. În plus, ASP.NET Core poate fi rulat atât pe Windows, Linux, macOS, cât și pe Docker.[16]

În final, ASP.NET Core Web API este un framework care pentru realizarea legăturii dintre server și client se folosește de protocoalele HTTP. Modul simplificat de funcționare al acestuia este următorul: clientul face un request de tip HTTP care este preluat de API controller-ul corespunzător. Acesta din urmă, prin apelarea și executarea metodei din manager, obține informațiile cerute care după un eventual proces de prelucrare sunt transmise clientului sub formă de JSON. Acest proces poate fi observat în *Figura 2.1.* [30]

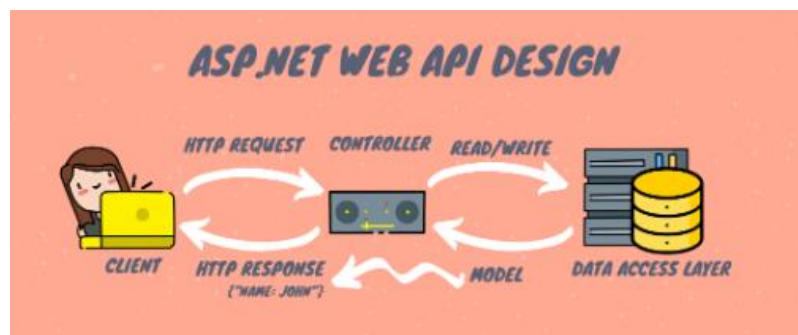


Figura 2.1. Reprezentare model ASP.NET Core Web Api, preluat din sursa:

<https://www.telerik.com/blogs/aspnet-core-beginners-web-apis>

În ceea ce privește backend-ul aplicației „TRI România”, arhitectura este compusă din trei niveluri cu roluri bine definite, precum: Controller, Manager și Repository.

Controllerul este o clasă care conține metode HTTP, precum: `HttpPost`, `HttpGet`, `HttpPut`, `HttpPatch`, `HttpDelete`, acestea fiind utilizate pentru a gestiona cererile HTTP primite de la client. De asemenea, controllerul are scopul de a ruta și redirecționa cererile către metodele corespunzătoare din manager, care de altfel este injectat în controller.

Managerul este o clasă intermediară între controller și repository. Acesta cuprinde implementările metodelor din controller care pot presupune efectuarea de prelucrări, validări sau alte operații necesare aplicației. Pentru a putea accesa și manipula datele din baza de date, managerul face apeluri către metodele din repository.

Repository-ul este o clasă care are acces la baza de date și conține metode de tip *CRUD* (Create, Read, Update, Delete) prin care se pot crea noi elemente, citi, actualiza sau chiar șterge elemente din baza de date.

Astfel, controllerul primește cererea HTTP de la client și o direcționează spre metoda managerului potrivit. Acesta din urmă execută funcția aferentă cererii, iar pentru obținerea datelor necesare din baza de date apelează repository-ul. Datele sunt apoi returnate către controller și apoi transmise către client în format JSON.

2.1.2. Trimiterea de email-uri

Aplicația „TRI România” folosește sistemul de trimitere de mesaje pe email în două cazuri: când antrenorul finalizează cu succes înscrierea unui sportiv în competiția aleasă, dar și în momentul în care administratorul platformei adaugă o nouă competiție. Un aspect

important cu privire la distribuirea email-urilor este că acestea sunt trimise destinatarilor doar dacă aceștia au selectat că doresc să fie abonați la știri.

Pentru a fi posibilă trimiterea de email-uri, s-a instalat biblioteca MailKit care permite utilizatorilor să trimită, să primească și să gestioneze mesajele de email. De asemenea, aceasta are la bază biblioteca MimeKit care se concentrează pe crearea și parsarea mesajelor prin folosirea MIME (*Multipurpose Internet Mail Extensions*) – standard pentru formatarea atașamentelor în email-uri, precum: audio, video, imagini și alte tipuri de fișiere. [25]

Dintre protocoalele acceptate de MailKit s-a ales SMTP (*Simple Mail Transfer Protocol*) pentru care s-a folosit portul SSL 465. Totodată, s-a decis folosirea drept host a SMTP-ului pentru Yahoo, anume: *smtp.mail.yahoo.com*. Pașii următori au presupus alegerea mailului care va trimite mesajele, numele cu care va apărea acesta când destinatarii văd mailul și parola. Toate aceste setări pentru email se găsesc în partea de backend în fișierul *appsettings.Development.json*.

În cele din urmă se creează un serviciu care, preluând setările făcute pentru email, construiește o funcție care are rolul de a trimite un mesaj pe email. Funcția primește ca input un model ce conține: emailul, numele destinatarului, titlul și conținutul emailului, iar ca output întoarce dacă emailul a fost sau nu trimis cu succes. În interiorul metodei se construiește un obiect de tipul *MimeMessage* care va reprezenta mesajul emailului.

Totodată, prin intermediul *MailboxAddress* se stabilesc numele și adresele expeditorului și destinatarului care sunt adăugate mesajului prin câmpurile de *From* și *To*. Mai mult, prin câmpul *Subject* este adăugat titlul email-ului.

Pentru conținutul email-ului, se utilizează un obiect *BodyBuilder* – ajută la crearea corpurilor de email sub standardele comune MIME. În cazul „TRI România” se setează doar conținutul text al email-ului prin utilizarea proprietății *TextBody*. După ce câmpul *Body* al mesajului este inițializat cu conținutul email-ului se stabilește conexiunea cu serverul SMTP prin *SmtplibClient* și se realizează autentificarea la contul expeditorului cu datele din setările emailului. Următorul pas este trimiterea mesajului creat, urmând deconectarea de la server și eliberarea resurselor utilizate.

2.2. SignalR

2.2.1. Date generale

SignalR este o bibliotecă dezvoltată de Microsoft, având ca scop adăugarea de funcționalități care să aibă loc în timp real în aplicațiile web. Transmiterea în timp real presupune capacitatea distribuirii instantanee a conținutului de la nivelul serverului către toți clienții conectați, fără ca aceștia să fie nevoiți să solicite explicit informații noi de la server.[13]

Printre exemplele de utilizare a SignalR se numără: crearea unui chat, apariția de date noi la refreshul paginii, platforme ce presupun aducerea mai multor persoane împreună pentru a colabora, formulare, notificări, jocuri în timp real, actualizări ale progresului muncii, și altele. [13]

Aspectele importante ale SignalR sunt legate de: transport, server, hub, protocoale, utilizatori, grupuri și conexiuni.[17]

1. Transport

Pentru a gestiona comunicarea în timp real, SignalR folosește una din următoarele metode de transport care se potrivește cel mai bine serverului și clientului său (a se vedea *Figura 2.2.*): *WebSocket* – permite conexiuni bidirecționale și persistente între server și client; *Server-Sent Events* – presupune stabilirea unei conexiuni de lungă durată între client și server, permițând trimiterea de date de tip text doar spre client, după ce acesta din urmă face cerere la server; *Long Polling* – presupune o serie de cereri din partea clientului și răspunsuri din partea serverului. [26]

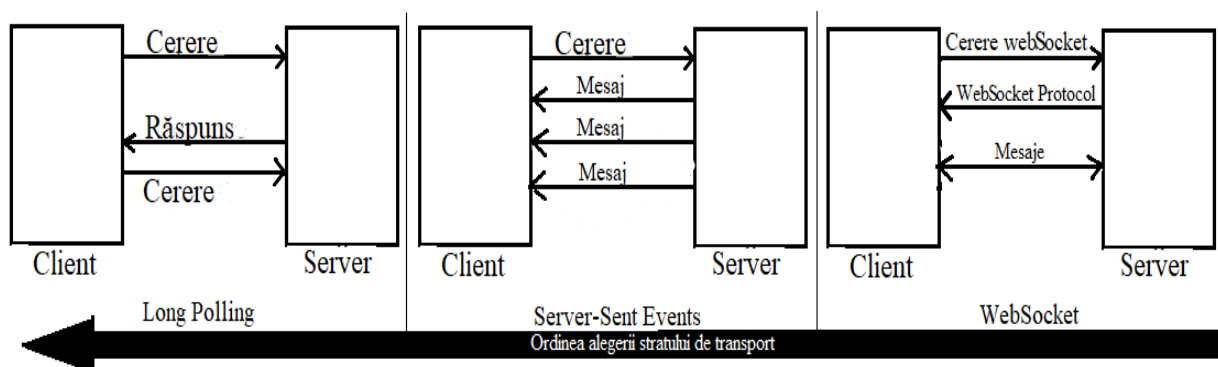


Figura 2.2. Reprezentarea nivelurilor de transport

2. Server

Serverul este reprezentat de o clasă care moștenește clasa Hub. Mai mult, acesta conține o serie de metode accesibile pentru clienții care fac cereri la server. De asemenea, serverul conține și evenimente la care se pot abona clienții, permițându-le să primească notificări în timp real atunci când acestea au loc.

3. Hub

În ceea ce privește SignalR, un hub reprezintă o componentă importantă, fiind utilizat pentru realizarea comunicării între servere și clienți. Totodată, acesta reprezintă un *pipeline* de nivel înalt care permite apelarea reciprocă de metode între cele două părți implicate.

4. Protocoale

SignalR prezintă incorporate două protocoale, anume: protocol text bazat pe JSON – cel implicit și un protocol binar ce are la bază *MessagePack* care creează mesaje de dimensiuni mai mici față de cele generate cu JSON.

5. Utilizatori

Utilizatorii sunt cei care folosesc aplicația, aceștia având posibilitatea de a face parte dintr-un grup sau de a acționa individual. Fie că este vorba de utilizatorii care fac parte dintr-un grup sau de un user care folosește platforma prin intermediul mai multor dispozitive aceștia vor fi notificați în același timp în cadrul grupului, respectiv pe toate dispozitivele.

6. Grupuri

Pentru a trimite date doar spre un ansamblu de utilizatori, serverul poate crea grupuri. Acestea necesită un nume unic după care pot fi identificate, iar printre acțiunile posibile se numără: adăugarea, respectiv ștergerea unei conexiuni din grup.

7. Conexiuni

Când un client se conectează la un hub îi este asignat un identificator unic care este știut doar de acesta și server. Mai mult, utilizatorul are o conexiune unică cu serverul.

De asemenea, pentru fiecare tip de hub există o singură conexiune, ceea ce presupune faptul că hub-urile nu vor interfera, ci vor avea propriile conexiuni de clienți, funcționând separat.

Modul de funcționare al SignalR este prezentat mai jos:

- Clientul trimite o cerere HTTP serverului.
- SignalR, în urma procesării cererii, determină tipul potrivit pentru transferul datelor între cele două părți, luând în considerare mai întâi cele mai bune metode și apoi cele mai puțin eficiente.
- Dacă serverul și clientul acceptă WebSockets atunci conexiunea dintre aceștia se transformă într-o conexiune de tipul WebSocket, în caz contrar se încearcă transportul de tipul Server-Sent Events. Dacă nici acest transport nu este acceptat, atunci se încearcă transportul Long Polling.
- Fiind stabilit tipul de transport al datelor, SignalR va realiza cu ajutorul hub-urilor comunicarea dintre client și server.

2.2.2. Utilizarea SignalR în aplicație

„TRI România” oferă utilizatorilor logați posibilitatea de a comunica prin intermediul unui chat care a fost implementat cu ajutorul SignalR. În cele ce urmează se vor prezenta pașii ce au dus la construirea platformei de comunicare atât pe partea de backend, cât și pe partea de frontend.

1. Backend

Pentru a beneficia de serviciile acestei biblioteci, s-a adăugat serviciul SignalR în fișierul *Program.cs* al aplicației. Privind implementarea logicii de comunicare pentru funcționalitatea de chat s-a creat clasa ChatHub, moștenitoare a clasei Hub și s-a stabilit ruta pentru acesta – când ruta este accesată de clienți, atunci aceștia vor fi conectați la ChatHub.

Clasa ChatHub conține metode și evenimente care gestionează interacțiunea dintre clienți și server. Dintre metodele de bază ale clasei se pot remarca:

1. Conectarea – apelată atunci când un client se conectează la serverul de chat. Aceasta adaugă codul de conexiune al clientului în grupul de „Chat” și trimite un eveniment de tipul „ConnectedUser” spre client pentru a-l notifica cu privire la faptul că a fost conectat cu succes.
2. Deconectarea – apelată atunci când un client se deconectează de la serverul de chat. Aceasta elimină codul de conexiune al clientului din grupul de „Chat”,

actualizează codul său de conexiune la *null*, iar disponibilitatea ca *false*. De asemenea, sunt apelate funcțiile de afișare a utilizatorilor online și offline.

3. Adăugarea conexiunii pentru un utilizator cu email-ul primit ca input – pentru utilizatorul căruia îi aparține email-ul se actualizează codul de conexiune cu valoarea conexiunii curente, iar disponibilitatea acestuia este actualizată la valoarea *true*. Sunt apelate și funcțiile de afișare a utilizatorilor online și offline.
4. Crearea chatului privat – metoda primește ca input un model ce conține date precum: emailul expeditorului, destinatarului, mesajul din conversație și data trimiterii acestuia. Se verifică disponibilitatea expeditorului și destinatarului corespunzători adresei de email din mesaj. Există două cazuri posibile când vine vorba de crearea unui chat: când cel care trimite mesajul este disponibil, dar celui căruia îi trimite nu este disponibil, respectiv când ambii utilizatori sunt disponibili.

În primul caz, mesajele necitite și care așteptau un răspuns de la expeditor (dacă este cazul) sunt considerate citite; se creează un grup privat în care este adăugat utilizatorul care a inițiat conversația prin funcția *AddToGroupAsync*; mesajul trimis este salvat în baza de date, unde proprietatea *citireMesaj* are valoarea *false* – având în vedere faptul că destinatarul nu a putut vedea direct mesajul, fiind indisponibil.

În cel de-al doilea caz, disponibilitatea celor doi utilizatori este actualizată la valoarea *false*, aceștia devenind indisponibili. În cazul în care unul dintre ei avea mesaje necitite, acestea sunt considerate a fi citite; se apelează funcția care afișează utilizatorii online. Se creează un nume de grup format din adresele de email ale celor doi participanți și este adăugat în lista de grupuri. În grupul privat este adăugat user-ul care a inițiat conversația, dar și destinatarul, prin folosirea codurilor de conexiune. De asemenea, mesajul creat este salvat în baza de date, dar de această dată proprietatea *citireMesaj* are valoarea *true*, întrucât destinatarul a putut vedea mesajul. În cele din urmă, prin metoda *SendAsync* se trimite evenimentul „OpenPrivateChat” spre destinatar pentru a se deschide chat-ul privat în interfața acestuia.

5. Primirea de mesaje private (a se vedea *Extras cod-sursă 2.1.*) – metoda primește ca input același model ca în cazul funcției de mai sus. Se verifică disponibilitatea

expeditorului și destinatarului și se formează două cazuri: când cel care a trimis mesajul este disponibil, iar cel care îl primește este indisponibil, respectiv când ambii utilizatori sunt indisponibili (ceea ce înseamnă că aceștia poartă o conversație unul cu altul) sau sunt disponibili.

```
public async Task ReceivePrivateMessage(ConversatieModel message)
{
    var codUtilizatorFrom = userManager.Users.Where(u =>
u.Email.Equals(message.codUtilizator)).Select(u =>
u.Id).FirstOrDefault();
    var codUtilizatorTo = userManager.Users.Where(u =>
u.Email.Equals(message.codUtilizator2)).Select(u =>
u.Id).FirstOrDefault();
    var disponibilitateFrom =
manager.GetDisponibilitate(codUtilizatorFrom);
    var disponibilitateTo =
manager.GetDisponibilitate(codUtilizatorTo);

    if (disponibilitateFrom == true && disponibilitateTo == false)
    {
        string privateGroupNam =
GetPrivateGroupName(message.codUtilizator, "");
        var newMesaj = new ConversatieCreateModel
        {
            codUtilizator = userManager.Users.Where(u =>
u.Email.Equals(message.codUtilizator)).Select(u =>
u.Id).FirstOrDefault(),
            codUtilizator2 = userManager.Users.Where(u =>
u.Email.Equals(message.codUtilizator2)).Select(u =>
u.Id).FirstOrDefault(),
            mesajConversatie = message.mesajConversatie,
            citireMesaj = false
        };
        manager.CreateMesaj(newMesaj);
        await
Clients.Group(privateGroupNam).SendAsync("NewPrivateMessage",
message);
    }
    if ((disponibilitateFrom == false && disponibilitateTo ==
false) || (disponibilitateFrom == true && disponibilitateTo ==
true))
    {
        string privateGroupName =
GetPrivateGroupName(message.codUtilizator,
message.codUtilizator2);
        var esteGrup = groups.Contains(privateGroupName
        if (esteGrup == false)
        {
            await
Clients.Group(GetPrivateGroupName(message.codUtilizator,
"")).SendAsync("NewPrivateMessage", message);
            await CreatePrivateChat(message);
        }
    }
}
```

```

        else
        {
            var newMesaj = new ConversatieCreateModel
            {
                codUtilizator = userManager.Users.Where(u =>
u.Email.Equals(message.codUtilizator)).Select(u =>
u.Id).FirstOrDefault(),
                codUtilizator2 = userManager.Users.Where(u =>
u.Email.Equals(message.codUtilizator2)).Select(u =>
u.Id).FirstOrDefault(),
                mesajConversatie = message.mesajConversatie,
                citireMesaj = true
            };
            manager.CreateMesaj(newMesaj);
            await
Clients.Group(privateGroupName).SendAsync("NewPrivateMessage",
message);
        }
    }
}

```

Extras cod-sursă 2.1. – Metoda de primire a mesajelor private

În primul caz, se creează un mesaj care este salvat în baza de date, proprietatea *citireMesaj* având valoarea false, deoarece destinatarul nu poate citi mesajul când este indisponibil. Mesajul este trimis către grupul privat corespunzător – cel care îl conține pe expeditor – prin evenimentul „NewPrivateMessage”.

În cel de-al doilea caz, se verifică dacă grupul format din email-urile expeditorului și destinatarului este printre grupurile aflate în sesiunea curentă. Dacă grupul nu există este echivalent cu faptul că destinatarul indisponibil căruia îi scria utilizatorul a devenit disponibil în timpul în care expeditorul îi scria mesaje, fără ca acesta să fi închis fereastra de chat. Drept urmare, mesajul trimis de expeditor va fi transmis către grupul privat corespunzător, iar destinatarul îl va putea primi. Apoi se apelează funcție de creare a chat-ului privat cu mesajul trimis ca input cu scopul de a crea o conversație privată între cei doi utilizatori. Dacă grupul exista deja, atunci mesajul este salvat în baza de date, cu proprietatea *citireMesaj* având valoarea *true*, urmând ca mesajul să fie transmis grupului privat corespunzător.

6. Închiderea chatului privat – primește ca input adresele de email ale celor doi utilizatori.

Dacă după verificarea disponibilității acestora, ambii sunt indisponibili, atunci li se actualizează disponibilitatea la valoarea *true*. Totodată, sunt apelate funcțiile care afișează utilizatorii online și cea care afișează pentru utilizatorul curent persoanele împreună cu numărul de mesaje necitite. În continuare se va șterge grupul privat format de cei doi utilizatori din lista de grupuri, se va transmite participanților din grupul privat un eveniment de tipul „ClosePrivateChat” pentru a-i anunța de închiderea chatului și apoi se înlătură cele două conexiuni din grup.

Dacă doar expeditorul este disponibil, se apelează funcția care afișează utilizatorii online și cea care afișează pentru utilizatorul curent persoanele împreună cu numărul de mesaje necitite. De asemenea, participanții grupului în cauză sunt notificați prin evenimentul „ClosePrivateChat” că chatul va fi închis, iar conexiunea expeditorului este eliminată din grupul privat.

2. Frontend

Pagina aferentă chatului poate fi accesată de către utilizatorii logați din bara de meniu. Când user-ul accesează pagina de mesagerie, se apelează metoda prin care se creează conexiunea cu serviciul de chat.

Funcția mai sus menționată cuprinde mai multe acțiuni. Dintre acestea se remarcă construirea obiectului *chatConnection* de tipul „HubConnection” prin folosirea „HubConnectionBuilder”, ce face parte din biblioteca SignalR, care este configurat să folosească transportul bazat pe HTTP pentru a se conecta la URL-ul specific, care coincide cu cel setat în fișierul *Program.cs* din backend, dar și să se reconecteze automat în cazul pierderii conexiunii. După ce conexiunea este creată, aceasta va fi pornită, iar eventualele erori vor fi afișate în consolă.

De asemenea, metoda conține pentru fiecare eveniment creat de server o implementare, precum: „ConnectedUser” care implică apelarea funcțiilor care invocă metodele: adăugarea conexiunii pentru un utilizator cu email-ul primit ca input, respectiv cea prin care se returnează persoanele alături de numărul de mesaje necitite pentru utilizatorul actual. În ceea ce privește implementarea evenimentelor: „OnlineUsers”, „OfflineUsers” și „UsersMessagesNr” rezultatul întors de la server va fi adăugat în lista specifică.

Totodată, evenimentul „OpenPrivateChat” primește ca parametru un model ce conține: adresele de email ale expeditorului și destinatarului, mesajul trimis de cel care a inițiat

conversația și data la care a fost trimis. Implementarea surprinde setarea faptului că a fost inițiată conversația privată, urmând să fie deschisă o fereastră de modal care cuprinde numele destinatarului, conversațiile trecute – dacă există și în partea de jos un câmp pentru scrierea și trimiterea de mesaje.

În cazul „NewPrivateMessage”, acesta primește același model ca evenimentul de mai sus și îl adaugă la lista de mesaje private.

Un alt eveniment implementat este „ClosePrivateChat” care resetează lista mesajelor private, setând faptul că a fost încheiată conversația privată. Mai mult, se apelează funcția care invocă metoda prin care se returnează persoanele alături de numărul de mesaje necitite pentru utilizatorul actual. La final, se închid ferestrele de modal care au fost menite comunicării între cei doi utilizatori.

O altă metodă importantă strâns legată de partea de chat este trimiterea mesajelor private. Aceasta este apelată atunci când utilizatorul apasă pe butonul de trimitere al mesajului și primește ca input date, precum: adresa de email a destinatarului, mesajul conversației și data de trimitere. Se creează mesajul ce conține pe lângă parametrii din input și emailul utilizatorului curent. Dacă conversația nu a fost setată ca fiind inițiată, atunci se setează și este invocată metoda de creare a chatului privat ce primește ca parametru mesajul construit mai sus, acesta fiind adăugat și în lista mesajelor private. Altfel, este invocată metoda de primire a mesajelor private ce primește ca parametru mesajul creat mai sus.

Mai mult, când utilizatorul închide fereastra de modal a chatului este apelată metoda de închidere a canalului de mesagerie privat ce are ca parametru emailul destinatarului. Astfel, este invocată metoda închiderii chatului privat ce primește ca input adrese de email ale utilizatorului curent și destinatarului, în această ordine.

2.2.3. Motivație alegere SignalR

Pentru a crea o aplicație de chat în timp real există mai multe opțiuni de implementare, dintre care se pot menționa: Firebase Realtime Database, Socket.IO, WebSocket, SignalR și altele.

Firebase, fiind un serviciu de cloud construit de Google, oferă dezvoltatorului posibilitatea să păstreze datele aplicației în una dintre bazele de date non-relaționale: *Firebase Realtime Database* și *Cloud Firestore*. Astfel, orice modificare adusă la una din aceste baze de date se va propaga și la nivelul clientului în timp real. Acest fapt este posibil datorită

folosirii WebSockets care mențin o conexiune bidirecțională persistentă între client și server.[1]

Socket.IO este o bibliotecă open-source, bazată pe WebSocket care presupune comunicarea bidirecțională între client și server. Aceasta dispune de servicii, ca: înlocuirea WebSocket cu HTTP Long Polling în sistemele unde cel dintâi nu este acceptat, detectarea deconectărilor, reconectarea automată, transmiterea de mesaje tuturor utilizatorilor sau doar a unei părți.[2]

WebSocket este un protocol care oferă un canal persistent de comunicare bidirecțional între un client și un server web printr-o conexiune de tip TCP (presupune schimbarea de mesaje într-o rețea). Modul de funcționare constă în trimiterea unei cereri de la client spre server prin care se cere trecerea de la conexiunea HTTP la WebSocket. Serverul trimite un răspuns prin care anunță dacă acceptă sau nu tipul de conexiune propus. În caz pozitiv, conexiunea WebSocket este stabilită, iar clientul și serverul pot transmite mesaje bidirecțional.[3]

Dintre aceste modalități am ales biblioteca SignalR, pentru că spre deosebire de WebSocket aceasta are în plus funcționalități, ca: reconectare automată, posibilitatea alegerii de către server dintre mai multe tipuri de transporturi, transmiterea mesajelor fie la o parte fie la toți utilizatorii. Mai mult, Socket.IO nu era o alegere potrivită, pentru că partea de backend a aplicației folosește .NET și nu Node.js care era util acestuia. Același motiv poate fi folosit pentru a motiva respingerea implementării cu ajutorul Firebase Realtime Database, la care se poate adăuga și efortul mai mare privind configurarea platformei Firebase.

2.3. Angular

2.3.1. Date generale și utilizarea în aplicație

Angular este unul dintre cele mai folosite framework-uri de frontend pentru aplicațiile web, având peste 1.7 milioane de dezvoltatori, creatori de conținut și autori. Acesta a fost dezvoltat de Google și are la bază TypeScript – limbaj de programare care extinde JavaScript [28], HTML și CSS.[5]

Arhitectura Angular este de tip MVC (*Model-View-Controller*), fiind bazată pe componente. Principalele elemente care stau la baza creării unei aplicații Angular, sunt

reprezentate de: componente, module, template-uri, metadata, servicii, injectare de dependențe, directive și rutare, așa cum se poate vedea în *Figura 2.3*. [6]

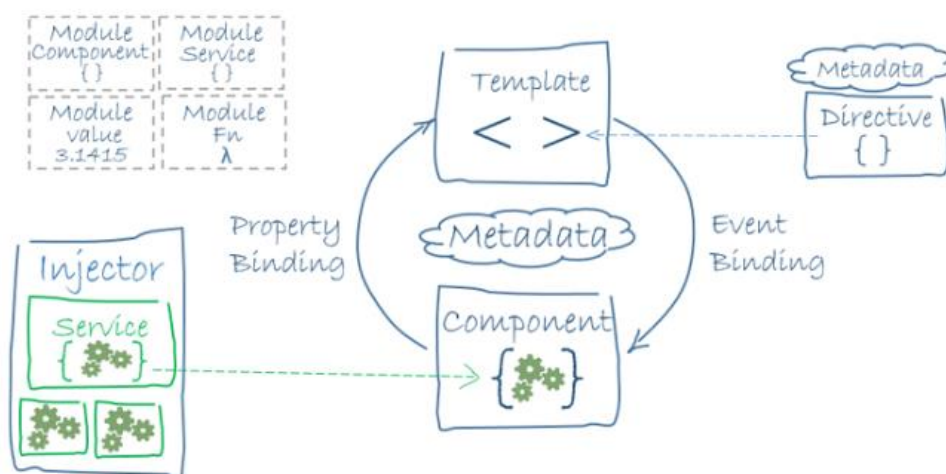


Figura 2.3. Reprezentare model Angular, preluat din sursa: <https://angular.io/guide/architecture>

1. Componente

În Angular, componentele sunt identificate prin marcatorul `@Component`, fiind plasat la începutul definirii acestora. Componentele definesc elementele de bază ale aplicației, cuprinzând: un template de tip HTML, stilizare prin CSS, SCSS, Sass, Less sau Stylus și o clasă de tip TypeScript. Acesta din urmă conține logica componentei și pentru implementarea acesteia se pot apela funcționalitățile din servicii.

2. Module

În Angular, modulele sunt identificate prin marcatorul `@NgModule`, plasat la începutul definirii acestora. NgModules cuprind o serie de componente care au funcționalități din același domeniu, temă. Acestea oferă componentelor un context prin care se pot importa modulele necesare, cât și posibilitatea exportării acestora.

Modulul de bază al unui proiect de tip Angular este *app.module* care definește configurarea inițială a aplicației. Acesta include secțiuni precum: declarații, importuri, dar și bootstrap – care are rolul de a lansa aplicația.

Organizarea componentelor în module facilitează dezvoltarea aplicațiilor complexe, oferind o structură clară și ușor de parcurs și înțeles. Mai mult, posibilitatea reutilizării de cod în alte module este marcată prin exportarea componentelor care aparțin unor module.

3. Template-uri

Un fișier de tipul HTML poate fi considerat un template dacă acesta conține atât sintaxa de template specifică Angular (**ngIf*, **ngFor*, *{{proprietate}}*, *<numeComponentă>* etc.), cât și elemente tipice HTML. De asemenea, rolul unui template este de a ghida framework-ul în ceea ce privește randarea componentei de care aparține. [7]

4. Metadata

În ceea ce privește componentele, opțiunile de configurare, precum: selectorul, url-ul template-ului, url-urile specifice stilizărilor, reprezintă metadata. Astfel, aceasta din urmă are rolul de a specifica framework-ului cum să creeze și de a-i specifica legăturile spre partea de HTML și cea de stilizare a componentei. [8]

În general, metadata folosește configurările claselor pentru a ghida framework-ul Angular în procesul de creare, gestionare și afișare a claselor în cadrul aplicației.

5. Servicii

Pentru a crea funcționalități care să poată fi utilizate de orice componentă a aplicației se adaugă clase de tipul *service*.

6. Injectarea dependențelor

Pentru a accesa funcționalitățile oferite de alte servicii în cadrul unei componente, este necesară injectarea dependențelor acestora în constructorul componentei respective care se face prin adăugarea instanțelor serviciilor ca parametri în constructor. Astfel, acestea devin disponibile în cadrul componentei pentru a fi utilizate ulterior în cod.

7. Directive

Directivele au rolul de a ghida framework-ul privind manipularea DOM-ului în momentul randării template-urilor. Acestea sunt de mai multe tipuri: directive structurale (**ngIf*, **ngFor*), de atribut (*ngModel*) și componente – fiind extinderea directivelor.[9]

8. Rutare

Cu ajutorul modului Router, Angular permite dezvoltatorului să creeze rute pentru componentele și modulele dorite. Mai mult, acesta îi oferă posibilitatea navigării spre o rută dată ca parametru.[10]

2.3.2. Motivație alegere Angular

Pentru implementarea interfeței aplicației „TRI România” s-a folosit framework-ul Angular având în vedere toate elementele menționate mai sus pe care le oferă. În plus, spre deosebire de React, acesta are incorporate mai multe instrumente ce privesc gestionarea sarcinilor de bază [24], iar față de Vue.js – JavaScript framework, arhitectura aplicațiilor dezvoltate de Angular este mult mai bine structurată și mai clară, ceea ce facilitează dezvoltarea aplicațiilor mai complexe [29].

2.4. Ng-Bootstrap

Ng-Bootstrap este o bibliotecă care folosește Bootstrap 5 CSS, având API-uri specifice pentru Angular. Dintre componentele puse la dispoziție utilizatorilor se numără: alerte, carusel de poze, alegerea de date, ferestre de modal, popover – apariția unei casete text în urma unei acțiuni, tipuri de recenzii, alegerea timpului sub formă de oră, minute, secunde, dar și altele.[11]

Pentru a avea acces la aceste elemente, se va instala biblioteca ng-bootstrap în aplicația de Angular. Astfel, „TRI România” folosește o parte din componentele mai sus menționate, precum: caruselul de poze care este prezent pe pagina principală a aplicației, alegerea de date și de timp – în cazul adăugării de competiții pentru stabilirea datelor și orelor de desfășurare a acestora, dar și în stabilirea limitei de timp pentru o probă. De asemenea, au fost utilizate și modale pentru fereastra de chat, editarea și respectiv pentru adăugarea de elemente. Mai mult, popover s-a folosit pentru interfața administratorului prin care era informat de rolul butonului peste care trecea cu cursorul, dar și pentru a anunța antrenorul că nu are sportivi în momentul în care vrea să vadă pe cine antrenează, sau pentru a atenționa sportivul că nu a participat la nicio competiție când acesta dorește să acceseze concursurile la care a luat parte. Pe lângă aceste componente au fost utilizate și cele pentru crearea de tabele, dar și de recenzii sub formă de inimă.

Cu scopul îmbunătățirii interfeței s-a instalat librăria open-source de iconițe a framework-ului Bootstrap care au fost preluate de aplicație sub formă de HTML. Acestea au fost folosite în scopuri diverse: pentru mesaje, notificări, adăugarea de elemente, butoanele de reacție și de comentariu din dreptul postărilor și altele.

CAPITOLUL 3 – ARHITECTURA APLICAȚIEI

3.1. Prezentarea generală a arhitecturii

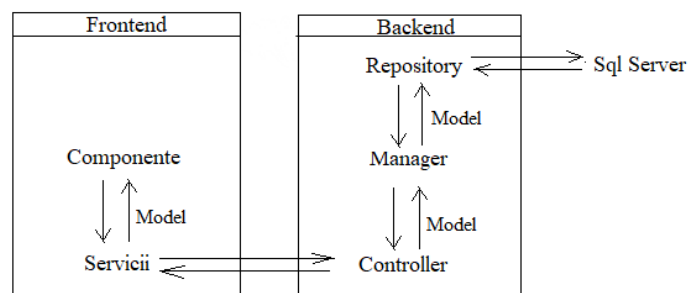


Figura 3.1. Arhitectura generală a aplicației

3.2. Baza de date

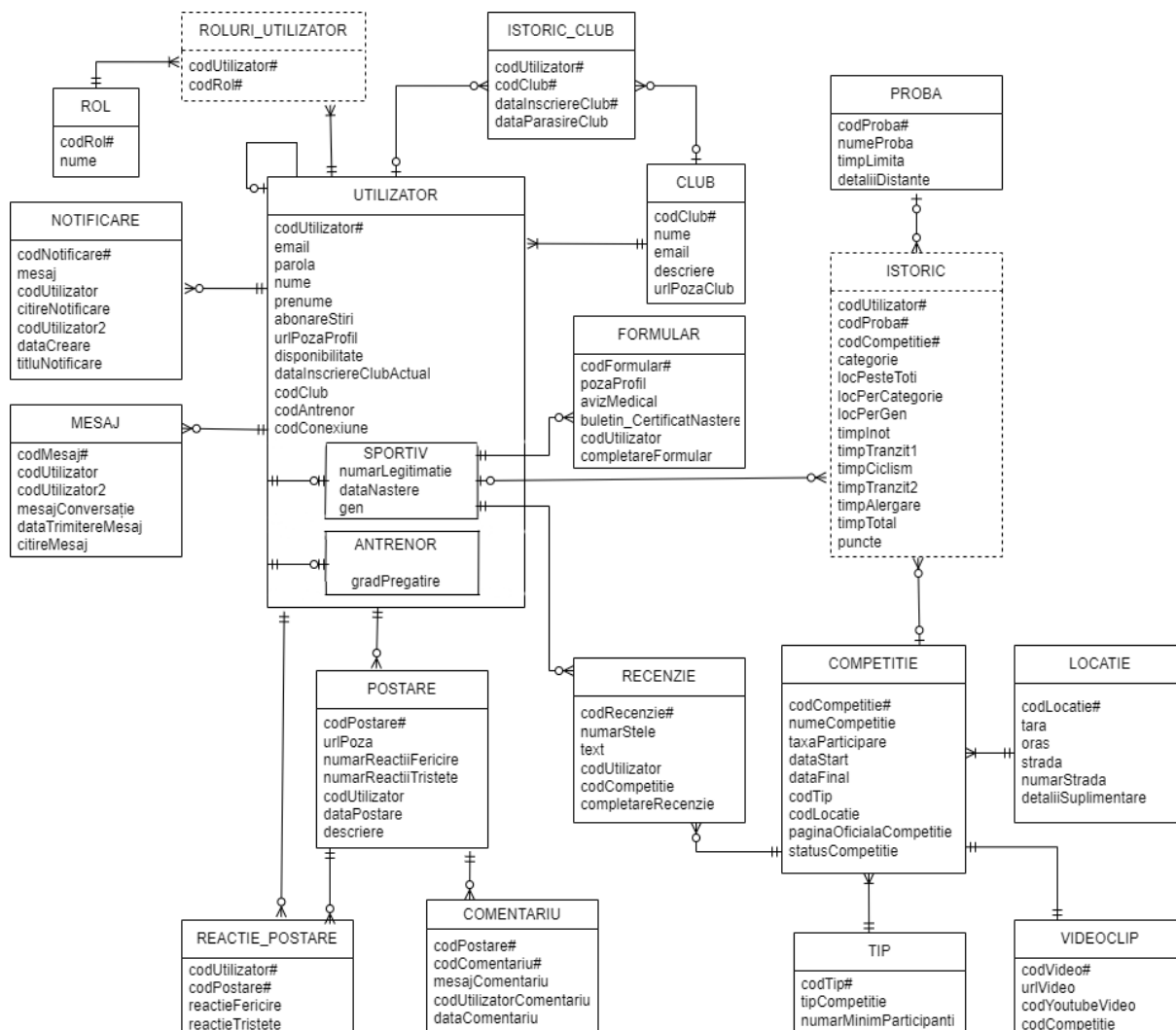


Figura 3.2. Diagrama conceptuală a bazei de date

Pentru crearea unei baze de date locale, s-a folosit SSMS (*Microsoft SQL Server Management Studio*) – un mediu folosit pentru gestionarea și administrarea bazelor de date. În ceea ce privește aplicația dezvoltată, baza de date cuprinde 18 tabele, predominând relațiile binare. De asemenea, există și o relație de tip 3 ce are loc între tabelele SPORTIV, PROBĂ și COMPETIȚIE, respectiv și o relație recursivă asociată tabelului UTILIZATOR.

Se poate remarca faptul că entitatea UTILIZATOR conține două subentități: SPORTIV și ANTRENOR. Pentru a se afla sportivii unui antrenor s-a recurs la o relație recursivă în cazul entității UTILIZATOR. Mai mult, baza de date conține și entitățile ROL și ROLURI_UTILIZATOR.

Aplicația se poate diviza astfel: partea care are în centru socializarea și cea care se concentrează asupra urmăririi rezultatelor sportivilor. Prima parte cuprinde entitățile: NOTIFICARE – anunțurile antrenorului spre sportiv când îl înscrie la o competiție, MESAJ – conversațiile utilizatorilor, POSTARE, COMENTARIU – entitate dependentă de POSTARE, REACȚIE_POSTARE – reacțiile utilizatorilor la postări. Cea de-a doua parte este formată din entitățile: CLUB, ISTORIC_CLUB – reține istoricul cluburilor în care au activat sportivii și antrenorii, facilitând asocierea corectă a rezultatelor sportivilor la competiții cu clubul la care erau afiliați în momentul concursului, FORMULAR – stochează elementele necesare licențierii sportivilor, PROBĂ, COMPETIȚIE, ISTORIC – rezultatele sportivilor la probele din cadrul competițiilor la care au luat parte, TIP – tipul competiției, LOCAȚIE – zona unde are loc competiția, VIDEOCLIP – contribuie la vizionarea competiției, RECENZIE – atribuția sportivilor de a-și spune părerea legată de concursul la care au participat.

Pentru accesarea bazei de date locală, stringul de conexiune al acesteia se va adăuga în fișierul appsettings.Development.json care conține configurările aplicației.

3.3. Backend

3.3.1. Entity Framework Core

Entity Framework Core este o versiune *open-source*, *cross-platform* a Entity Framework, *mapper* relațional de obiecte, care permite crearea tabelelor bazei de date, stabilirea relațiilor între acestea prin intermediul claselor din .NET utilizând limbajul C#, fără a fi nevoie de a scrie cod SQL. De asemenea, acesta oferă și posibilitatea realizării interogărilor LINQ [18]. Pentru a adăuga tabelele sau modificările aduse acestora în baza de

date se folosesc migrări, de unde reiese faptul că dintre cele trei abordări ale acestui framework: Code First, Database First și Model First s-a ales prima metodă.

Pentru a face legătura între clasele specifice tabelelor și baza de date trebuie definită o clasă de context care să extindă clasa `IdentityDbContext`. În fișierul *Program.cs* se adaugă un serviciu pentru adăugarea contextului, menționându-se stringul de conexiune care va fi utilizat pentru conectarea la baza de date. Prin intermediul obiectelor de tipul `DbSet` se specifică tabelele care se vor adăuga în baza de date, iar pentru stabilirea relațiilor între acestea se va suprascrie metoda `OnModelCreating`.

3.3.2. ASP.NET Core Identity

ASP.NET Core Identity este un framework care oferă dezvoltatorului o serie de clase cu proprietăți concepute special pentru a facilita implementarea funcționalității de autentificare într-o aplicație. Acesta poate fi folosit împreună cu o bază de date pentru a stoca informații despre utilizatori, rolurile, datele acestora și alte date relevante. [4]

„TRI România” este o aplicație care folosește Identity pentru clasele: `Utilizator`, `Rol` și `RoluriUtilizator` care moștenesc `IdentityUser`, `IdentityRole` - având cheile primare de tipul de bază - string și `IdentityUserRole`. În cazul clasei `Utilizator` pe lângă proprietățile moștenite se mai adaugă și altele comune atât sportivilor, cât și antrenorilor care sunt două clase ce extind clasa `Utilizator`.

În ceea ce privește rolurile pe care le pot avea utilizatorii aplicației se pot menționa: sportiv, antrenor și administrator. Pentru acestea se creează autorizații care permit blocarea anumitor opțiuni ale CRUD pentru utilizatorii care nu au rolul potrivit specificat în câmpul `Roles` al funcției `Authorize` care este expusă la începutul metodei din controller (a se vedea *Extras cod-sursă 3.1.*). Adăugarea serviciului de autorizare se face în fișierul *Program.cs* unde se adaugă *policy-uri* care sunt specifice rolurilor aplicației. În acest fel, autorizarea contribuie la securizarea aplicației prin protejarea metodelor CRUD de a fi folosite de utilizatorii nepotriviți.

```
[HttpGet("byEmail/{email}")]
[Authorize(Policy = "AntrenorUtilizator")]
public async Task<IActionResult> GetAntrenorById([FromRoute]
string email)
{
    var antrenor = manager.GetAntrenorInfo(email);
    return Ok(antrenor);
}
```

Extras cod-sursă 3.1. – Metodă de tip `HttpGet` din controller autorizată

În cadrul aplicației dezvoltate, procesul de autentificare este unul important. Înregistrarea presupune completarea câmpurilor specifice rolului ales de utilizator: sportiv sau antrenor, parola și adresa de email trebuind să respecte modelul regex-ului asociat. În ceea ce privește parola, aceasta contribuie la securizarea platformei prin stocarea sa în baza de date într-un format criptat. Pentru realizarea procesului de criptare a parolei, se utilizează funcția `CreateAsync` din framework-ul Identity, care primește ca parametri utilizatorul și parola asociată acestuia. Prin apelarea metodei `UpdatePasswordHash`, se obține parola în format hash, asigurând astfel și confidențialitatea acestor informații sensibile. Autentificarea presupune următorii pași: verificarea existenței adresei de email în baza de date, urmată de examinarea asocierii dintre email și parolă. Dacă logarea este împlinită cu succes atunci se creează un token, altfel se afișează un mesaj din care reiese scrierea incorectă a email-ului sau parolei. În cele ce urmează se va descrie metoda de creare a token-ului. În primul rând, se generează o cheie de securitate simetrică folosind cheia secretă extrasă din fișierul de configurare. Următorul pas constă în specificarea credențialelor necesare semnării token-ului, incluzând cheia de semnare și algoritmul de criptare utilizat. Se definește un descriptor al token-ului care conține informații precum timpul de expirare, credențialele de semnare și subiectul, care este legat de rolul utilizatorului. În final, se utilizează un obiect de tipul `JwtSecurityTokenHandler` pentru a crea tokenul, utilizând descriptorul definit anterior.

3.3.3. Repository pattern

Un repository este o clasă folosită pentru a gestiona acțiunile de creare, afișare, actualizare și ștergere asociate unui obiect dintr-o clasă specifică. *Repository pattern* presupune separarea procesului de modelare a datelor din baza de date de restul aplicației, contribuind la crearea unei structuri modulare, ceea ce facilitează gestionarea, monitorizarea și testarea codului mai ușor. Un avantaj al utilizării repository pattern este eliminarea interogărilor duplicate spre baza de date, prin concentrarea într-o singură clasă a tuturor metodelor specifice unei entități.[19,20]

În cadrul unui repository se injectează contextul aplicației, care este necesar pentru a stabili conexiunea cu baza de date și pentru a implementa metodele CRUD corespunzătoare. De asemenea, aceste funcții sunt definite în interfața asociată repository-ului cu scopul de a fi utilizate în clasele de manager care reprezintă legătura între repository și controllere. În cazul metodelor care afișează date, acestea returnează un model specific, care este o reprezentare a

entității respective. Acestea sunt utilizate pentru a transfera datele în nivelul superior – manager.

3.3.4. Arhitectura Traditional „N-Layer”

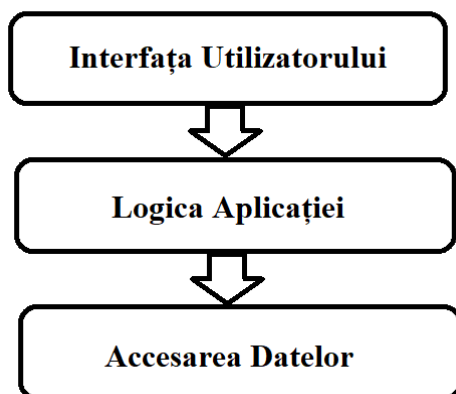


Figura 3.3. Nivelurile aplicației

Structurarea backend-ului platformei „TRI România” în controllere (vezi *Extras cod-sursă 3.4.*), manageri (vezi *Extras cod-sursă 3.3.*) și repository (vezi *Extras cod-sursă 3.2.*) corespunde celor trei layere fundamentale ale unei aplicații (vezi *Figura 3.3.*): interfața utilizatorului, logica aplicației și accesul la date. Acestea comunică prin intermediul modelelor care au rolul de a transfera date între niveluri.[21]

```
public class VideoclipRepo: IVideoclipRepo
{
    private GestionareFederatieTriatlonContext db;
    public VideoclipRepo(GestionareFederatieTriatlonContext db)
    {
        this.db = db;
    }

    public void Create(Videoclip videoclip)
    {
        db.Videoclipuri.Add(videoclip);
        db.SaveChanges();
    }
}
```

Extras cod-sursă 3.2. – Repository cu metoda de creare

```
public class VideoclipManager: IVideoclipManager
{
    private readonly IVideoclipRepo videoRepo;
    public VideoclipManager(IVideoclipRepo videoRepo)
```

```

    {
        this.videoRepo = videoRepo;
    }

    public void Create(VideoModelCreate videoCreate)
    {
        var newVideo = new Videoclip
        {
            urlVideo = videoCreate.urlVideo,
            codYoutubeVideo = videoCreate.codYoutubeVideo,
            codCompetitie = videoCreate.codCompetitie
        };
        videoRepo.Create(newVideo);
    }

```

Extras cod-sursă 3.3. – Implementare metoda Create în Manager

```

[Route("api/[controller]")]
[ApiController]
public class VideoclipController : ControllerBase
{
    private readonly IVideoclipManager manager;

    public VideoclipController(IVideoclipManager manager)
    {
        this.manager = manager;
    }

    [HttpPost]
    [Authorize(Policy = "AdminUtilizator")]
    public async Task<IActionResult> Create(FromBody]
VideoModelCreate model)
    {
        manager.Create(model);
        return Ok();
    }
}

```

Extras cod-sursă 3.4. – Funcția de creare în cadrul Controller-ului

3.4. Frontend

3.4.1. Arhitectura interfeței

Interfața aplicației este împărțită în module, componente, servicii și modele, respectând arhitectura MVC, elementele fiind descrise în capitolul Tehnologii în secțiunea Angular. Modul prin care interacționează acestea este detaliat mai jos.

Clasele de tipul serviciu facilitează interacțiunea cu serverul prin efectuarea cererilor HTTP, utilizând obiectul de tipul `HttpClient` definit în constructor. Acestea includ opțiuni HTTP, precum: tipul conținutului (json) și autentificarea prin adăugarea unui antet de tip *Bearer* care conține token-ul preluat din `localStorage`. De asemenea, prin intermediul acestor clase, se realizează trimiterea și primirea datelor către și de la backend, informațiile fiind gestionate și reținute sub formă de modele.

Componentele, prin fișierele de `typescript`, pe lângă crearea funcțiilor specifice apelează metodele corespunzătoare din servicii pentru a prelua sau modifica date. Rezultatele sunt salvate în variabile care apoi sunt utilizate în fișierele de HTML cu scopul afișării, creării, editării sau ștergerii informațiilor.

Astfel, componentele și serviciile colaborează pentru a gestiona interacțiunea cu datele și pentru a asigura funcționalitatea adecvată în cadrul aplicației.

3.4.2. Grafice - Ng2-charts

Aplicația „TRI România” folosește grafice pentru a putea oferi utilizatorului o reprezentare vizuală a statisticilor legate de rezultatele sportivilor și de cele ale cluburilor sportive. Astfel, se instalează pachetul `ng2-charts` și biblioteca `Chart.js`.

`Chart.js` este o bibliotecă *open-source* care oferă acces la diferite tipuri de grafice care pot fi personalizate prin *plugin-uri* care permit: stilizarea, interacțiunea, adăugarea de caracteristici și gestionarea surselor de date. Totodată, aceasta are la bază `TypeScript`, fiind compatibilă cu framework-uri de JavaScript printre care se numără și `Angular`. Pentru a integra această bibliotecă s-a ales folosirea `ng2-charts`.^[12]

`Ng2-charts` este un pachet *wrapper* dezvoltat special pentru `Angular`. Pentru a putea fi utilizat trebuie importat modulul `NgChartsModule`. Pachetul cuprinde opt tipuri de grafice, fiecare având la bază opțiunile: tipul de grafic, datele care trebuie interpretate de grafic (setul de date, etichete și alte proprietăți caracteristice tipului de grafic ales), seturi de date, etichete pentru acestea din urmă, opțiuni (de exemplu stabilirea titlului graficului) și afișarea legendei.^[22]

S-au utilizat trei grafice de tipuri diferite: *pie*, *bar* și *doughnut*. Pentru primele două s-a folosit o structură de implementare, ultima având o altă abordare.

Prima modalitate (vezi *Extras cod-sursă 3.5.*) constă în crearea unui obiect de tip Chart care primește ca parametri: id-ul canvasului asociat și configurarea graficului prin specificarea proprietăților corespunzătoare dintre cele menționate mai sus. Setul de date în acest caz s-a populat cu datele obținute în urma apelării unei metode din serviciu. În cadrul fișierului HTML, obiectul este transmis prin interpolare în componenta de tip canvas, id-ul corespunzând cu cel din obiectul creat, dimensiunile canvasului fiind setate corespunzător.

```
this.graficulPodium = new Chart('canvasul',{
    type:'pie',
    options: {
        plugins: {
            title: {
                display: true,
                text: 'Procent sportivi medaliați/nemedaliați '
            }
        }
    },
    data:{
        labels: ['Medaliați','Nemedaliați'],
        datasets:[{
            data:this.statisticaPodium,
            label:'Medaliere',
        }],
    })
})
```

Extras cod-sursă 3.5. – Crearea unui grafic de tip *pie*

Cea de-a doua opțiune a implicat crearea unui obiect ChartOptions care primește tipul graficului dorit, fiind urmat de specificarea opțiunilor acestuia. În plus, valorile pentru etichete și date se atribuie prin apelarea funcțiilor dintr-un serviciu. În ceea ce privește fișierul HTML (a se vedea *Extras cod-sursă 3.6.*), se folosește directiva baseChart în cadrul elementului canvas, iar proprietățile, cum ar fi tipul de grafic, setul de date, etichetele, opțiunile și dimensiunile canvasului, sunt atribuite corespunzător.

```
<div style="display: block; margin-left:40%;margin-right: 40%;
margin-top: 2%;">
    <canvas baseChart
    [type]=" 'doughnut' "
```

```
[datasets]="dataPie"  
[labels]="etichete"  
[options]="pieOption"  
width="400"  
height="400"  
>  
</canvas>  
</div>
```

Extras cod-sursă 3.6. – Crearea unui grafic de tip *doughnut*

3.4.3. Google API

Google Maps Platform oferă utilizatorilor acces la un ansamblu variat de API-uri și SDK-uri. Acestea permit dezvoltatorilor să integreze sau să preia date din Google Maps pe care să le utilizeze în aplicațiile create. Astfel, programatorii pot avea acces la diferite elemente, precum: hărți (afișarea hărților în diferite formate, vizualizare 360° a străzilor, etc.) , rute (planificarea acestora, detalii despre trafic, etc.) și locuri (detalii, opțiunea de căutare a unei locații specifice, etc). [14]

Maps JavaScript API oferă posibilitatea utilizatorului de a adăuga și personaliza hărți interactive, marcatori, rute și alte elemente grafice cu scopul de a crea experiențe de navigare complexe și interactivitate în aplicațiile dezvoltate.

Google Geocoding API este un alt serviciu oferit de platforma Google Maps, prin care adresele sunt transformate în coordonate geografice: latitudine și longitudine. În urma acestei convertiri harta poate fi centrată în dreptul punctelor de coordonate aflate. [27]

Pentru a afișa locul desfășurării competițiilor, platforma „TRI România” folosește cele două tipuri de API descrise mai sus. Se utilizează Geocoding pentru a transforma adresa asociată competiției în coordonate geografice precise. Aceasta implică efectuarea unei solicitări de tip fetch către un URL specific pentru geocoding, care include adresa locației competiției (obținută prin apelarea metodei de afișare din serviciu) și cheia API-ului Google Maps. Rezultatul primit este un fișier JSON care conține informații, printre care și câmpul *geometry*, care include detalii despre locație, cum ar fi longitudinea și latitudinea. Se construiește un obiect de tip *google.maps.Map* care primește ca parametri: elementul HTML de tip div în care va fi afișată harta și opțiunile de afișare, precum: centrarea pe baza coordonatelor geometrice și nivelul de *zoom*. Mai mult, se adaugă un marcator pe hartă prin

intermediul unui obiect de tip *google.maps.Marker* care primește ca date poziția marcatorului, harta pe care va fi afișat și un titlu corespunzător.

3.4.4. YouTube Video Player

Ngx-youtube-player este o componentă Angular, fiind bazată pe IFrame Player API furnizat de aplicația YouTube care permite controlarea și integrarea unui videoclip de pe platforma YouTube (redarea, întreruperea, oprirea, etc), prin funcții JavaScript. [23]

Componenta Angular permite o integrare mai ușoară a videoclipurilor YouTube fără a necesita gestionarea manuală cu API-ul menționat mai sus. De asemenea, aceasta conține următoarele atribute care trebuie completate în fișierul de HTML: *videoId* – pentru încărcarea videoclipului cu identificatorul specificat, calitatea, dimensiunile playerului YouTube, variabilele de configurare și evenimentul care este declanșat când videoclipul poate fi redat.[23] Aplicația dezvoltată utilizează redarea videoclipurilor din YouTube pentru ca utilizatorii să poată urmări concursurile atât în timp real, cât și după finalizarea concursului.

3.4.5. Fluxul de funcționare al interfeței

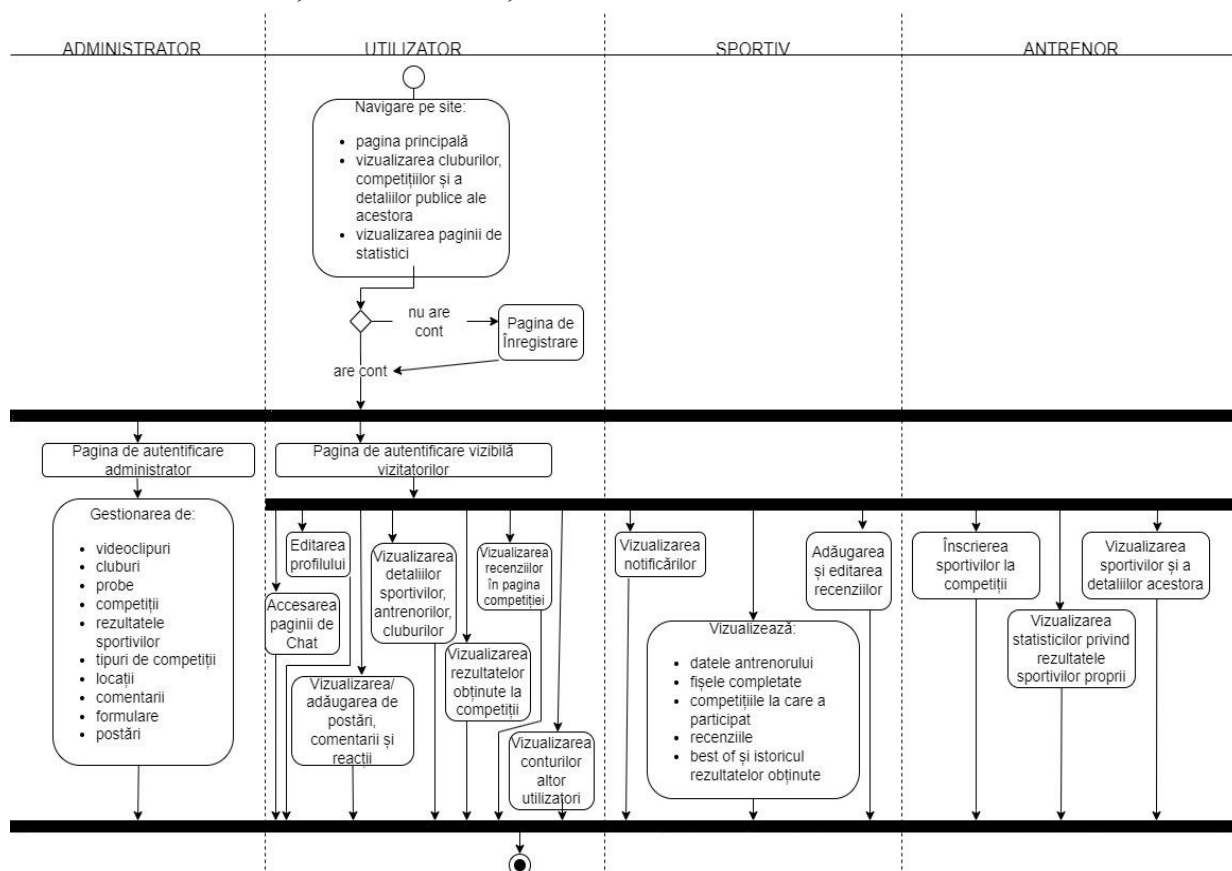


Figura 3.4. Fluxul de funcționare al interfeței

CAPITOLUL 4 – PREZENTAREA APLICAȚIEI

4.1. Înregistrare și autentificare

Aplicația „TRI România” permite vizitatorilor să navigheze pe site, aceștia având acces doar la paginile care i-ar putea inspira să se înscrie la unul dintre cluburile Federației Române de Triatlon.

Dacă cel care vizitează site-ul este un antrenor sau sportiv nou venit în lumea triatlonului, acestuia i se va aduce la cunoștință link-ul prin intermediul căruia se va putea înregistra în aplicație (vezi *Figura 4.1.*), beneficiind de funcționalitățile acesteia.

The screenshot displays the registration interface of the 'TRI România' application. At the top, a navigation bar contains a logo and four menu items: 'Cluburi', 'Competiții', 'Statistici', and 'Logare'. The main content area features a large background image of triathletes swimming in the ocean. Overlaid on this is a white registration form titled 'Formular de înregistrare'. The form contains the following elements: an 'Email*' input field with a red error message 'Introduceți o parolă validă'; a 'Parola*' input field with a red error message 'Cel puțin o literă mare, o literă mică, o cifră, un caracter special. Cel puțin 8 caractere.'; a 'Rol*' dropdown menu; 'Nume*' and 'Prenume*' input fields; a section for 'Abonare știri' with 'Da' and 'Nu' radio buttons; a 'Choose File' button with the text 'No file chosen'; and a 'Cod club*' input field with the value '0'. A prominent yellow button labeled 'Înregistrare' is positioned at the bottom of the form. Below the button, a link 'Ai deja cont? Logare' is visible.

Figura 4.1. – Pagina de înregistrare

Formularele de înregistrare includ un câmp denumit *Rol* care permite utilizatorilor să selecteze între două tipuri: sportiv sau antrenor (vezi *Figura 4.2.*). Diferențele dintre aceste două categorii se reflectă în câmpurile ce trebuie completate: antrenorii trebuie să menționeze gradul de pregătire, în timp ce sportivii trebuie să specifice adresa de email a antrenorului lor, numărul legitimației, data nașterii și genul. De asemenea, câmpurile de email și parolă sunt supuse validării. Parola trebuie să conțină cel puțin o literă mică, o literă mare, o cifră, un caracter special și să aibă minim opt caractere. În cazul email-ului, acesta trebuie să respecte formatul unei adrese de email valabile. După ce toate datele sunt completate, butonul de înregistrare devine disponibil, iar după crearea contului, utilizatorul va fi redirecționat către pagina de autentificare.

Formular de înregistrare

Email*

Introduceți o parolă validă

Parola*

Cel puțin o literă mare, o literă mică, o cifră, un caracter special. Cel puțin 8 caractere.

Antrenor

Nume*

Prenume*

Abonare știri* ☐ Da ☐ Nu

Choose File No file chosen

Cod club* 0

Grad Pregătire*

Înregistrare

Formular de înregistrare

Email*

Introduceți o parolă validă

Parola*

Cel puțin o literă mare, o literă mică, o cifră, un caracter special. Cel puțin 8 caractere.

Sportiv

Nume*

Prenume*

Abonare știri* ☐ Da ☐ Nu

Choose File No file chosen

Email antrenor*

Cod club* 0

Numar legitimitate* 0

Data naștere* mm/dd/yyyy

Gen* ☐ F ☐ M

Înregistrare

Figura 4.2. – Formulare de înregistrare pentru antrenor și pentru sportiv

În ceea ce privește pagina de autentificare, utilizatorii au de completat câmpurile de email și parolă, iar dacă unul dintre acestea este introdus greșit se va afișa un mesaj de eroare. Butonul de logare devine activ când cele două câmpuri sunt completate (vezi *Figura 4.3.*).

Logare

Email* mirunapami@yahoo.com

Parola*

Email sau parolă incorecte.

Logare

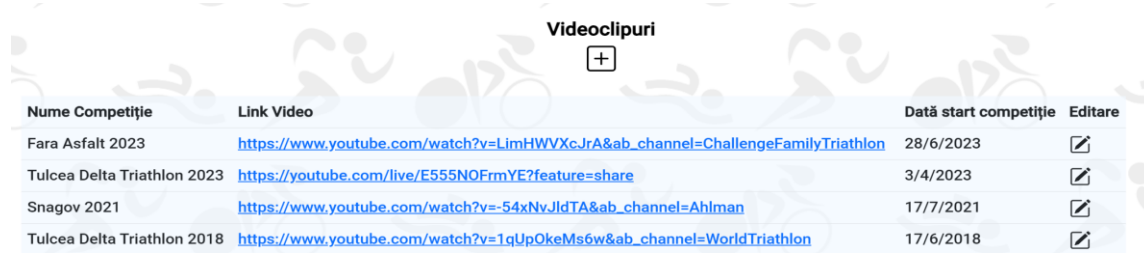
Figura 4.3. – Formularul de autentificare

Un aspect important, ce ține de securitatea site-ului, este acela că autentificarea administratorului se face pe baza unui link cunoscut doar de acesta, iar câmpurile ce trebuie completate sunt aceleași cu cele menționate mai sus.

4.2. Interfața administratorului

4.2.1. Gestionarea videoclipurilor

Pagina „Videoclip” conține un tabel în care se specifică link-ul spre videoclipul asociat competiției, ordonarea fiind făcută descrescător după data de start a concursului (vezi *Figura 4.4.*).



Nume Competiție	Link Video	Data start competiție	Editare
Fara Asfalt 2023	https://www.youtube.com/watch?v=LimHWVXcJrA&ab_channel=ChallengeFamilyTriathlon	28/6/2023	
Tulcea Delta Triathlon 2023	https://youtube.com/live/E555NOFrmYE?feature=share	3/4/2023	
Snagov 2021	https://www.youtube.com/watch?v=-54xNvJldTA&ab_channel=Ahlman	17/7/2021	
Tulcea Delta Triathlon 2018	https://www.youtube.com/watch?v=1qUpOkeMs6w&ab_channel=WorldTriathlon	17/6/2018	

Figura 4.4. – Pagina „Videoclip” din interfața administratorului

De asemenea, există și opțiunea de editare a link-ului și a codului videoclipului pentru o competiție aleasă, ce presupune deschiderea unei ferestre de modal (vezi *Figura 4.5.*).

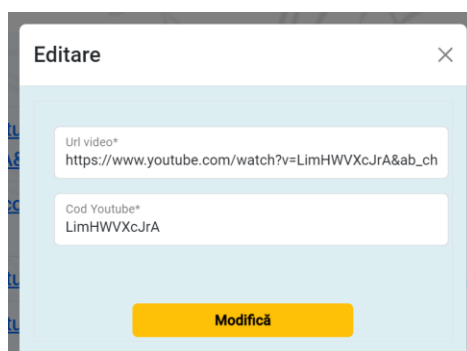


Figura 4.5. – Modal de editare a unui videoclip

Mai mult, administratorul poate adăuga un nou videoclip pentru competițiile active care nu au unul asociat. Se va trece link-ul și codul specific acestuia. (vezi *Figura 4.6.*).

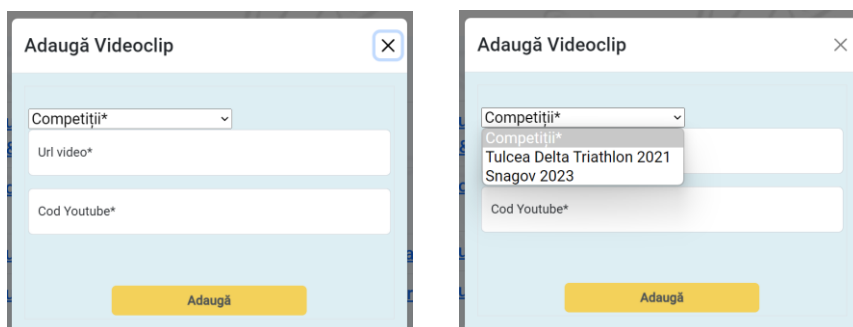


Figura 4.6. – Modal de adăugare a unui videoclip

4.2.2. Gestionarea cluburilor

Pagina destinată gestionării cluburilor surprinde afișarea datelor despre cluburi, precum: codul, poza, numele, adresa de email și descrierea într-un format tabelar (vezi *Figura 4.7.*) care prezintă pe ultima coloană posibilitatea de editare – modificarea tuturor câmpurilor, exceptând codul (vezi *Figura 4.8.*).

Cluburi					
+					
Cod	Poză	Nume	Email	Descriere	Editare
1004		Bucharest Sport Club	bsc@gmail.com	Cea mai importantă sarcină a celor mari este de a-i educa inteligent pe cei mici. Prin programe bine puse la punct și susținute zilnic, sportivii vor putea obține rezultate din ce în ce mai bune. Antrenorii nostri sunt foști sportivi de performanță.	
1		CSA Steaua	contact@csasteaua.ro	Clubul CSA Steaua își dorește să crească cei mai buni sportivi ai timpului alături de antrenori bine pregătiți. Sediul clubului se găsește pe Bdul. Ghencea, nr. 35, sector 6	
3		Delfinul Bucuresti	delfinul@yahoo.com	Delfinul București este un club unit, unde sportivii sunt ajutați atât de preparatorii fizici, cât și de antrenori să ajungă la performanțele visate.	

Figura 4.7. – Pagina „Club” din interfața administratorului

Editare

Schimbă poza clubului

Choose File No file chosen

Nume

CSA Steaua

Email

contact@csasteaua.ro

Descriere

Clubul CSA Steaua își dorește să crească cei mai buni sportivi ai timpului alături de antrenori bine pregătiți. Sediul clubului se găsește pe Bdul. Ghencea, nr. 35, sector 6

Modifică

Figura 4.8. – Modal de editare a unui club

De asemenea, apăsarea iconiței sub formă de plus declanșează deschiderea unei ferestre de modal care presupune completarea tuturor spațiilor pentru adăugarea noului club (vezi *Figura 4.9.*).

Adaugă club

Choose File No file chosen

Nume*

Email*

Descriere

Adaugă

Figura 4.9. – Modal de adăugare a unui club

4.2.3. Gestionarea probelor

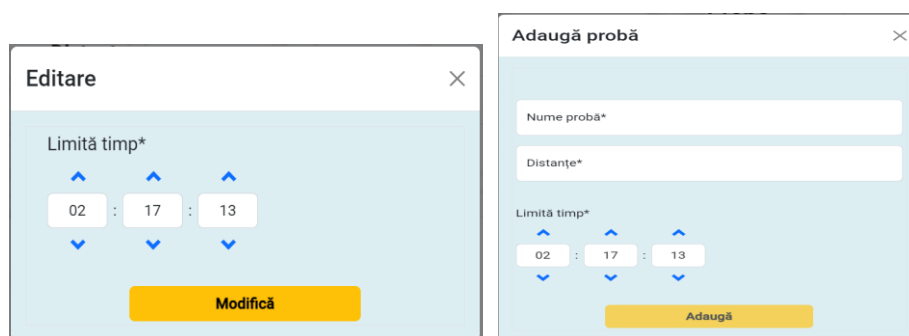
Pagina destinată probelor conține un tabel ce cuprinde informații legate de: numele, timpul limită și distanțele asociate probei (vezi *Figura 4.10.*).



Proba	Timp Limită	Distanțe	Editare
Olimpic	2:17:13	1.5 km inot, 40km ciclism, 10km alergare	
Sprint	01:50:00	700m inot, 12km bicicleta, 5km alergare	
Supersprint	00:50:00	500m inot, 6km bicicleta, 2.5km alergare	

Figura 4.10. – Pagina „Probă” din interfața administratorului

Această pagină oferă posibilitatea de editare care presupune deschiderea unei ferestre de modal ce permite administratorului schimbarea timpului limită al probei dorite. Pentru introducerea de noi probe se va apăsa pe iconița aflată în partea de sus a paginii care va deschide o pagină de modal unde se vor introduce datele corespunzătoare (vezi *Figura 4.11.*).



Editare

Limită timp*

02 : 17 : 13

Modifică

Adaugă probă

Nume probă*

Distanțe*

Limită timp*

02 : 17 : 13

Adaugă

Figura 4.11. – Modal de editare și adăugare a unei probe

4.2.4. Gestionarea competițiilor, rezultatelor și locațiilor acestora

În momentul în care administratorul apasă pe elementul „Competiție” din bara de meniu se va deschide o pagină care va conține atât informații cu privire la competiții, cât și la locațiile acestora.



Competiții și Locații													
Competiție	Taxă Participare	Data Start	Data Final	Pagina oficială	Status	Cod Locație	Tip	Editare	Despre	Cod Locație	Țara	Oraș	Strada
Fara Asfalt 2023	200	28/6/2023 ora:10:30	29/6/2023 ora:13:30	https://www.faraasfalt.ro/triatlon-la-mare/traseu/	activa	2002	nationala			4	Romania	Tulcea	Lac Ciuperca
Snagov 2023	100	25/5/2023 ora:9:15	25/5/2023 ora:18:25		activa	1002	nationala			1002	Romania	Snagov	Complex Sportiv Național Nicolae Navasart, Izvorani
										2002	Romania	Mnagalia	Pescaria 2 Mai

Figura 4.12. – Pagina „Competiție” din interfața administratorului

În ceea ce privește tabelul locațiilor acesta cuprinde date, precum: codul locației, țara, orașul și alte informații opționale – denumirea și numărul străzii și detalii suplimentare. Apăsarea iconiței de editare declanșează deschiderea unei ferestre de modal prin care se pot modifica toate datele din tabel, mai puțin codul. Mai mult, pentru adăugarea unei locații se folosește butonul din partea de sus a paginii care deschide o pagină de modal asemănătoare celei pentru editare (vezi *Figura 4.13.*). De asemenea, ștergerea unei locații se face prin apăsarea simbolului sub formă de coș de gunoi din dreptul acesteia.

Figura 4.13. – Modal de editare și adăugare a unei locații

În cazul competițiilor, tabelul asociat cuprinde informații cu privire la: denumirea, taxa de participare, data de start și de final, pagina oficială, statusul (activa/anulata), codul locației unde se desfășoară și tipul concursului. Totodată, pentru modificarea datelor afișate în tabel pentru un concurs anume se va apăsa butonul de editare din dreptul acestuia care va deschide un modal de editare, iar pentru adăugarea unui nou concurs – butonul „Adaugă competiție” (vezi *Figura 4.14.*).

Figura 4.14. – Modal de editare și adăugare a unei competiții

Mai mult, pentru a putea modifica rezultatele participanților la o competiție anume se selectează butonul din secțiunea „Despre” din dreptul concursului dorit. Astfel, administratorul va fi redirecționat spre o altă pagină care conține rezultatele tuturor

participanților ordonate după probă (vezi *Figura 4.15.*), atribuția acestuia fiind aceea de a modifica timpii și locurile obținute pentru a fi în conformitate cu timpii rezultați din cronometrare și de a oferi numărul de puncte corespunzătoare (vezi *Figura 4.16.*).

Concurenții competiției Tulcea Delta Triathlon 2018

Număr Legitimăție	Probă	Competiție	Categorie	Loc General	Loc Gen	Loc Categorie	Timp Total	Înot	Tranzit 1	Ciclism	Tranzit 2	Alergare	puncte	Editare
1	Sprint	Tulcea Delta Triathlon 2018	AG 45-49 FEMININ	4	4	1	01:21:20	00:15:00	00:01:00	00:30:00	00:00:20	00:30:00	40	
4	Sprint	Tulcea Delta Triathlon 2018	CADETI FEMININ	2	2	1	01:07:20	00:11:00	00:01:00	00:29:30	00:00:20	00:25:30	80	
21	Sprint	Tulcea Delta Triathlon 2018	JUNIORI I 18-19 FEMININ	3	3	1	01:07:55	00:11:00	00:00:40	00:30:00	00:00:15	00:26:00	60	
2	Sprint	Tulcea Delta Triathlon 2018	UNDER 23 FEMININ	1	1	1	01:02:52	00:09:54	00:00:50	00:28:53	00:00:10	00:23:05	100	
1	Supersprint	Tulcea Delta Triathlon 2018	AG 50-54 FEMININ	3	2	1	00:40:16	00:06:50	00:00:27	00:21:28	00:00:11	00:11:20	80	
2	Supersprint	Tulcea Delta Triathlon 2018	UNDER 23 FEMININ	999	999	999	99:99:99	99:99:99	99:99:99	99:99:99	99:99:99	99:99:99	0	
19	Supersprint	Tulcea Delta Triathlon 2018	UNDER 23 FEMININ	999	999	999	99:99:99	99:99:99	99:99:99	99:99:99	99:99:99	99:99:99	0	
10	Supersprint	Tulcea Delta Triathlon 2018	UNDER 23 FEMININ	2	1	1	00:40:07	00:06:47	00:00:30	00:21:29	00:00:10	00:11:11	100	
17	Supersprint	Tulcea Delta Triathlon 2018	UNDER 23 MASCULIN	1	1	1	00:35:06	00:06:12	00:00:20	00:19:15	00:00:10	00:09:19	100	
50	Supersprint	Tulcea Delta Triathlon 2018	UNDER 23 MASCULIN	999	999	999	99:99:99	99:99:99	99:99:99	99:99:99	99:99:99	99:99:99	0	

Figura 4.15. – Pagina de rezultate pentru o competiție din interfața administratorului

Valorile inițiale pentru timpii și locuri sunt ca cele din dreptul sportivilor care au legitimația 2, 19 sau 50, acestea fiind trecute astfel la înscrierea sportivului la concurs.

Modifică rezultatele

locPesteToti
999

locPerGen
999

locPerCategorie
999

timpTotal
99:99:99

puncte
0

timpInot
99:99:99

timpTranzit1
99:99:99

timpCiclism
99:99:99

timpTranzit2
99:99:99

timpAlergare
99:99:99

Modifică

Figura 4.16. – Modal de editare a rezultatelor obținute la o competiție

4.2.5. Gestionarea tipurilor de competiții

Pagina afișată în urma apăsării elementului „Tip” din bara de meniu conține un buton de adăugare a unui nou tip de concurs, un tabel cu detalii despre tipuri și opțiunea de editare a numărului de participanți pentru un tip anume. Adăugarea și editarea se realizează prin completarea câmpurilor din ferestrele de modal (vezi *Figura 4.17.*).

Tipuri

+

Tip	Număr minim participanți	Editare
internacionala	28	
nationala	10	

Adaugă Tip

Tip competiție*

Număr minim participanți*

0

Adaugă

Editare

Număr minim participanți

28

Modifică

Figura 4.17. – Pagina „Tip” din interfața administratorului și modalele de editare și adăugare a unui tip

4.2.6. Gestionarea recenziilor, comentariilor și postărilor

Limbajul și conținutul promovat în cadrul aplicației trebuie să fie unul decent, și de aceea administratorul poate șterge atât comentarii, recenzii care nu au mesaje tocmai potrivite, cât și postări pe baza pozelor distribuite.

4.2.7. Gestionarea formularelor

Pentru a reînnoi licențele sportivilor, administratorul are acces la fișierele încărcate de aceștia: poza de profil, avizul medical și buletinul sau certificatul de naștere. Acestea sunt afișate sub formă de tabel, putând fi filtrate după anul dorit.

4.3. Interfața utilizatorului neautentificat

4.3.1. Pagina principală

Pagina de deschidere a aplicației are ca scop familiarizarea vizitatorului cu ceea ce înseamnă triatlonul. În prima parte a paginii se regăsește un carusel cu cinci imagini, reprezentând: fostul președinte al federației, noua generație de sportivi și cele trei probe ale triatlonului: înot, ciclism și alergare. De asemenea, sunt menționați și sponsorii Federației Române de Triatlon, iar în partea de jos se află motto-ul federației, aparținând fostului președinte al acesteia, precum și link-uri către platformele de socializare.

4.3.2. Pagina „Cluburi”

Pagina „Cluburi” oferă vizitatorilor posibilitatea de a descoperi cluburile afiliate federației și de a afla detalii despre acestea prin apăsarea butonului „Despre” (vezi *Figura 4.18*).

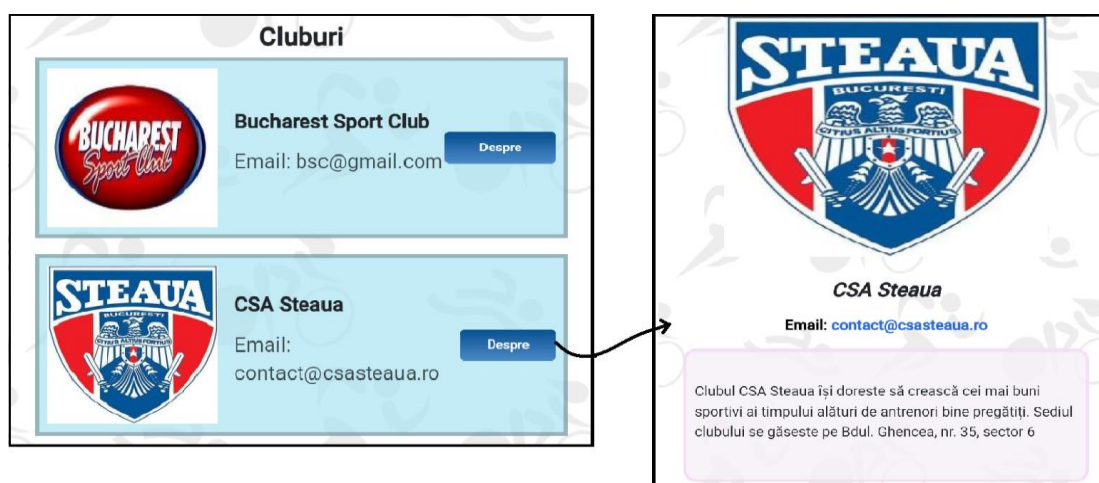


Figura 4.18. – Pagina „Cluburi” și pagina cu detalii despre cluburi

4.3.3. Pagina „Competiții”

Pagina „Competiții” listează toate concursurile care au avut sau vor avea loc, menționând detalii legate de perioada de desfășurare și tipul competiției, restul de informații regăsindu-se pe pagina competiției prin apăsarea butonului „Despre”. Mai mult, în cazul în care o competiție este anulată, atunci utilizatorul nu va putea accesa pagina concursului (vezi *Figura 4.19.*).

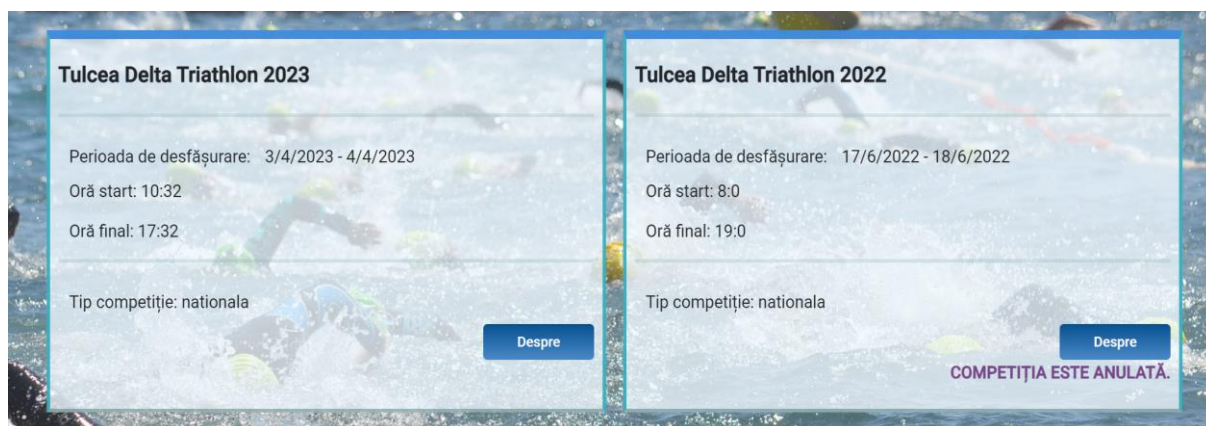


Figura 4.19. – Pagina „Competiții”

Dintre informațiile regăsite în pagina competiției care nu se regăsesc și în pagina principală, cele mai importante sunt: locația concursului prezentată pe o hartă interactivă, elementul care prezintă scurgerea timpului până la momentul începerii evenimentului și site-ul oficial al competiției, dar și rezultatul recenziilor sportivilor (vezi *Figura 4.20.*).



Figura 4.20. – Componente din pagina unei competiții care va avea loc versus care a avut loc

4.3.4. Pagina „Statistici”

4.3.4.1. Afișarea topului de 5 sportivi

Pentru recunoașterea campionilor federației de triatlon, „TRI România” oferă utilizatorului posibilitatea de a-i vedea pe cei mai buni 5 sportivi. De asemenea, sunt puse la dispoziție patru componente pe baza cărora se poate realiza filtrarea: categorie, club, probă și an (vezi *Figura 4.21.*).



Figura 4.21. – Top 5 sportivi

4.3.4.2. Afișarea topului de 5 cluburi

Pentru a-și putea face o imagine mai clară asupra performanțelor unui club, sau pentru a observa evoluția acestuia, utilizatorul poate urmări statisticile cu cele mai bune 5 cluburi pe toți anii sau în funcție de unul ales (vezi *Figura 4.22.*).



Figura 4.22. – Top 5 cluburi

4.4. Interfața comună utilizatorilor cu rol de sportiv și antrenor

4.4.1. Pagina „Cluburi”

Utilizatorii autentificați cu rolurile precizate mai sus, când intră pe pagina unui club, beneficiază de posibilitatea de a accesa informații cu privire la sportivii și antrenorii care fac parte din clubul respectiv și de a-i căuta după numele de familie sau prenume. Mai mult, numele tastat nu trebuie să fie neapărat complet. Un exemplu de căutare este prezent în *Figura 4.23*. Mai mult, apăsarea butonului „Detalii” din dreptul sportivului sau antrenorului va redirecționa utilizatorul spre profilul public al acestuia.

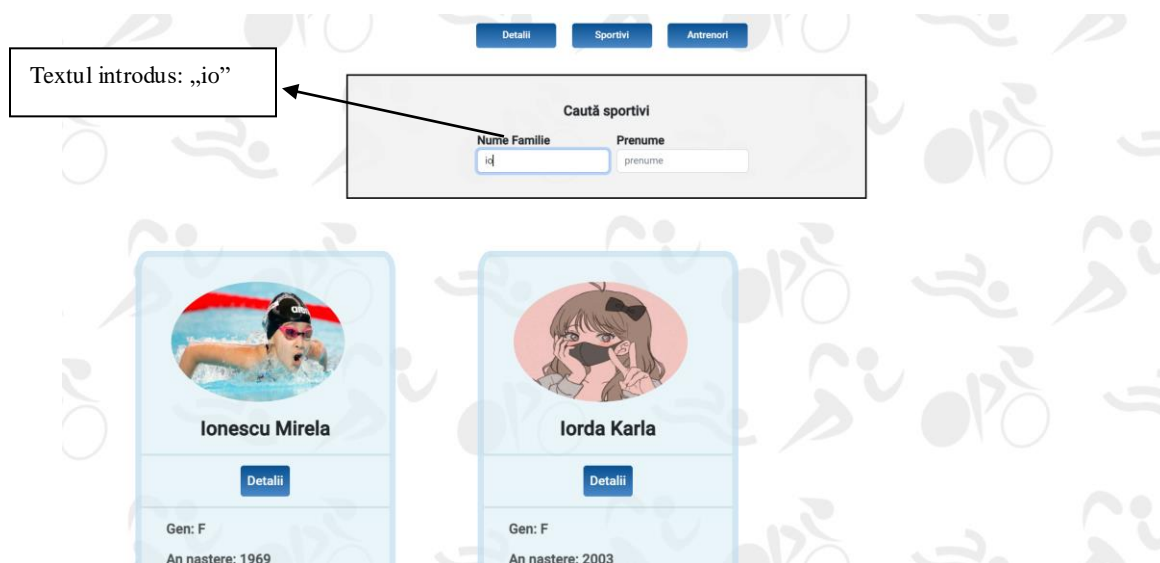


Figura 4.23. – Metoda de căutare a sportivilor/antrenorilor din pagina de detalii a unui club

4.4.2. Pagini dedicate postărilor

4.4.2.1. Fluxul de postări

Bara de meniu, prin elementul „Postări” permite utilizatorilor vizați să vizualizeze și să interacționeze cu postările celorlalți prin oferirea de reacții și comentarii (vezi *Figura 4.24*). Totodată, aceștia au și opțiunea de a adăuga o postare – imaginea și descrierea asociată, prin apăsarea pe iconița de la începutul paginii.

În ceea ce privește reacțiile, utilizatorul știe pe care a ales-o astfel: reacția tristă – dacă este colorată cu albastru, reacția fericită – colorată în galben. Dacă acesta dorește să își anuleze reacția sau să o schimbe, mai întâi trebuie să apese pe reacția activă, iar apoi va putea decide dacă va mai da sau nu o altă reacție.

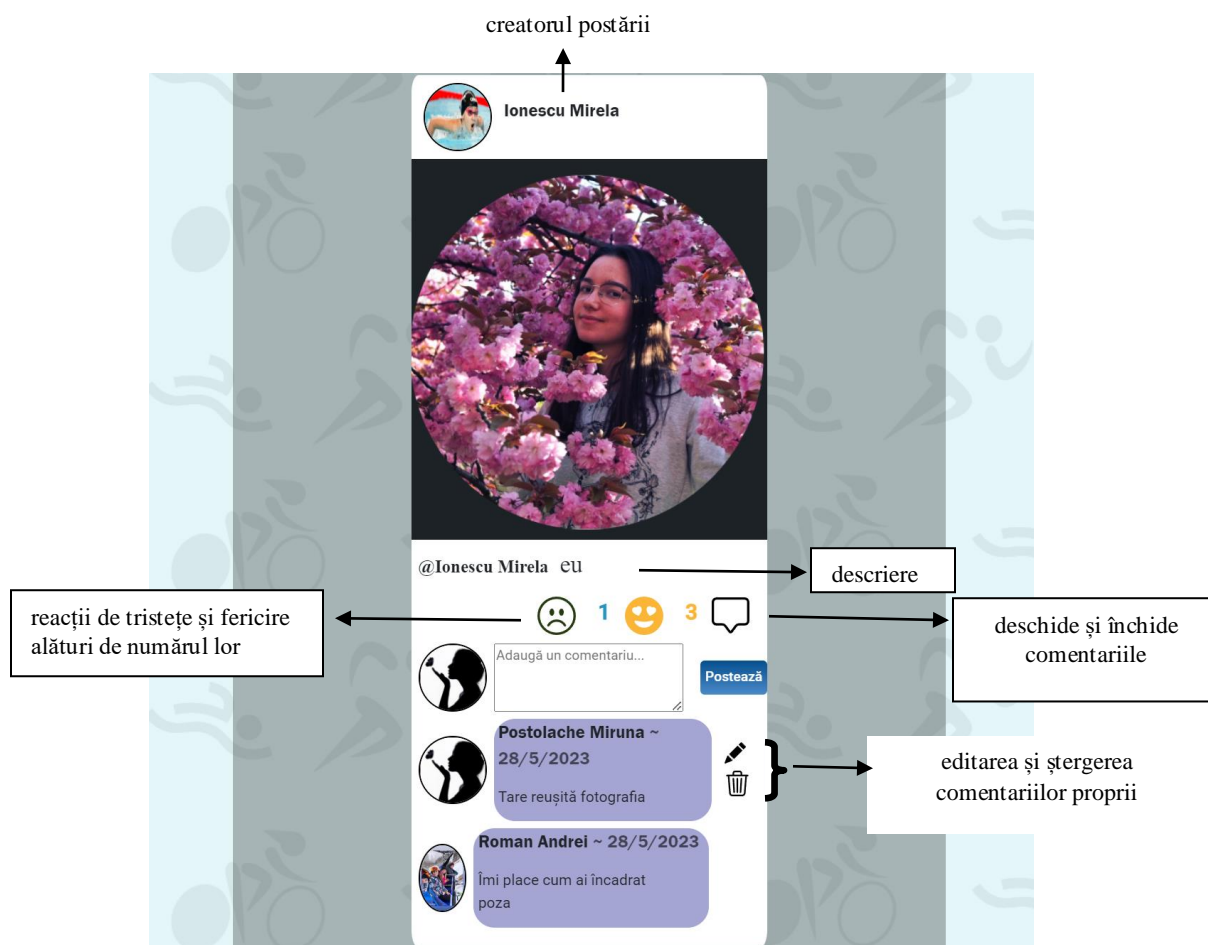


Figura 4.24. – Prezentarea postărilor și interacțiunea cu acestea

Mai mult, prin apăsarea pictogramei specifice creatorului postării, utilizatorul va fi redirecționat spre profilul public al acestuia care pe lângă alte detalii va avea și o secțiune în care sunt afișate postările acestuia.

4.4.2.2. Postările proprii

Pagina care conține doar postările create de utilizator poate fi accesată din profilul său. Spre deosebire de acțiunile disponibile pentru postările din cadrul paginii „Postări”, sau cele din pagina profilului public, această pagină conține și opțiunea ștergerii postării selectate. În plus, pictograma creatorului nu va redirecționa utilizatorul spre profil, deoarece doar creatorul are acces la această pagină.

4.4.3. Pagina de Chat

Cu scopul de a face comunicarea mai ușoară între sportivi și antrenori, aplicația „TRI România” oferă posibilitatea acestora de a trimite mesaje atât spre utilizatorii online – cei care se află pe pagina destinată conversațiilor, cât și spre cei offline. Accesul la pagina de chat este

posibil prin intermediul barei de meniu, prin apăsarea butonului cu un design sugestiv care prezintă în centru și numărul mesajelor necitite (vezi *Figura 4.25.*).

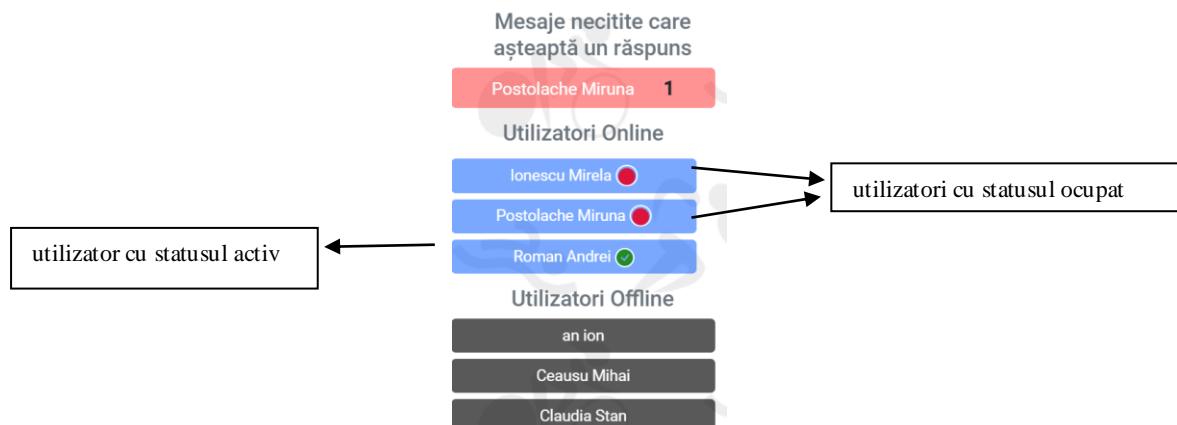


Figura 4.25. – Așezarea utilizatorilor în pagina de Chat

Pentru a începe o conversație, utilizatorul trebuie să dea dublu click pe numele persoanei cu care dorește să vorbească, fie că aceasta este online sau offline și se va deschide o fereastră de modal (vezi *Figura 4.26.*).

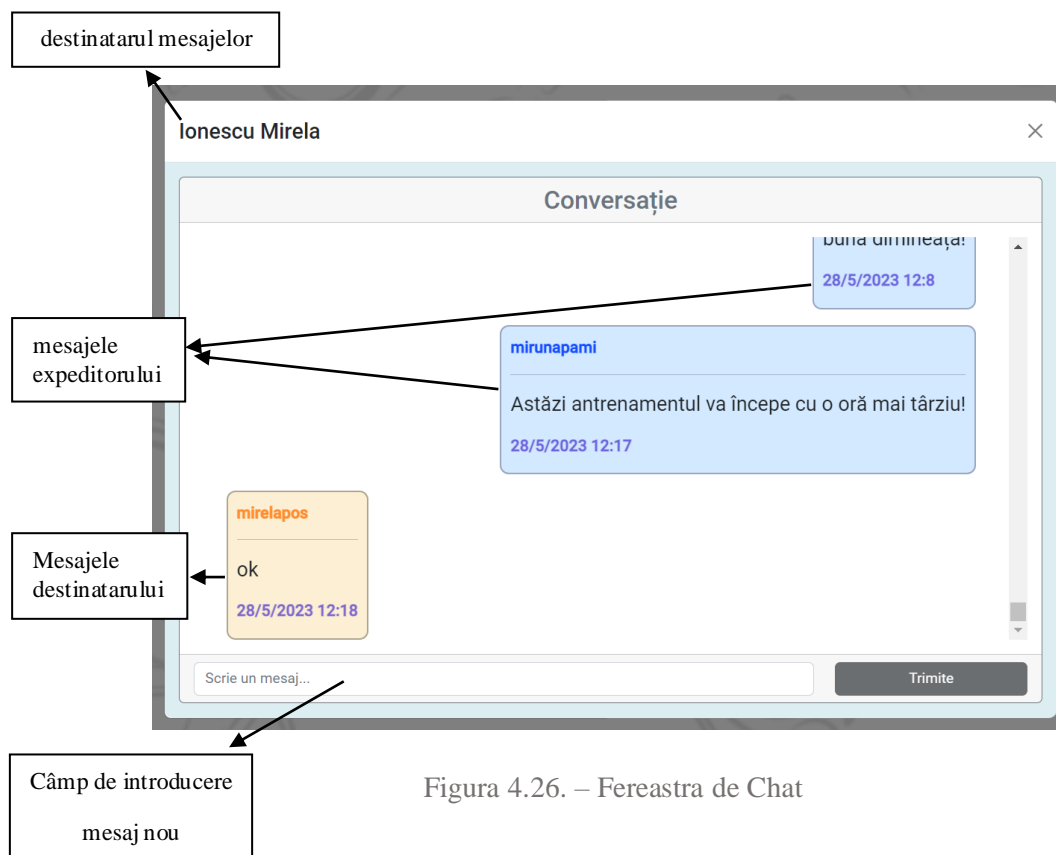


Figura 4.26. – Fereastra de Chat

4.4.4. Pagina de Rezultate

Pentru fiecare competiție există o pagină care conține rezultatele tuturor sportivilor înscriși la concursul respectiv. Acestea sunt afișate sub forma unui tabel și cuprind informații cu privire la: numele și prenumele sportivului, categoria, clubul din care face parte, timpul total în care a parcurs cursa, poziția ocupată (în clasamentul general, în funcție de gen și în funcție de categorie) și punctele obținute. Mai mult, rezultatele pot fi filtrate în funcție de categoriile, probele și cluburile participante.

În partea superioară se află butonul „TOP CLUBURI” care în urma apăsării afișează primele cinci cluburi în funcție de numărul total de puncte obținute de sportivii lor. Acestea sunt reprezentate atât sub forma întâlnită în pagina de statistici, cât și ca diagramă reprezentată ca *doughnut* (vezi Figura 4.27.). Cea din urmă oferă o vizualizare mult mai bună asupra diferențelor de punctaj între cluburi.

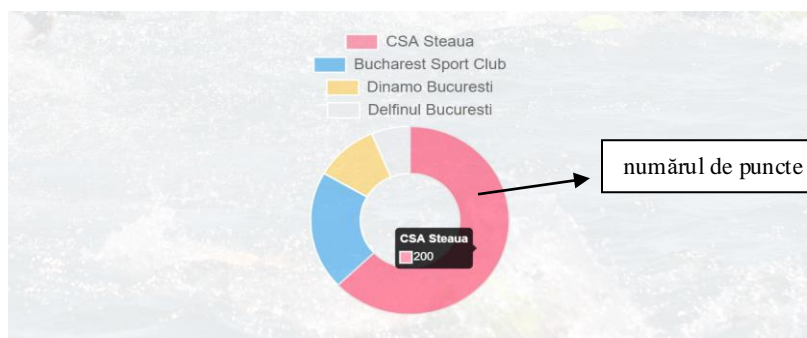


Figura 4.27. – Diagramă de tip *doughnut* pentru top 5 cluburi

4.4.5. Pagina de detalii specifică unui concurs

Față de interfața afișată pentru utilizatorii neautentificați, pagina de detalii a unei competiții conține în plus un câmp în care sunt afișate recenziile sportivilor participanți (vezi Figura 4.28.) și un buton care redirecționează utilizatorul spre pagina cu rezultate.



Figura 4.28. – Componenta de recenzii

4.5. Interfața utilizatorului cu rol de sportiv

4.5.1. Pagina de detalii pentru competițiile proprii

Din pagina profilului său, sportivul poate accesa competițiile la care a fost înscris. Pagina de detalii, față de funcționalitățile prezentate mai sus, are în plus posibilitatea redactării unei recenzii prin oferirea unui număr de inimi și a unui mesaj. După ce recenzia a fost creată, butonul care permite scrierea acesteia nu va mai fi vizibil.

4.5.2. Pagina notificărilor

În momentul în care antrenorul finalizează înscrierea unui sportiv în cadrul unei competiții, acesta din urmă primește în pagina de notificări o înștiințare cu privire la concursul și proba la care va participa (vezi *Figura 4.29.*). Pagina care conține aceste notificări poate fi accesată din bara de meniu prin apăsarea butonului sub formă de plic care are în centru numărul notificărilor necitite. Pentru marcarea notificărilor ca fiind citite, sportivul trebuie să citească fereastra deschisă prin apăsarea butonului „Detalii”. La închiderea acesteia, notificarea va fi considerată citită, iar numărul de notificări necitite se va actualiza.

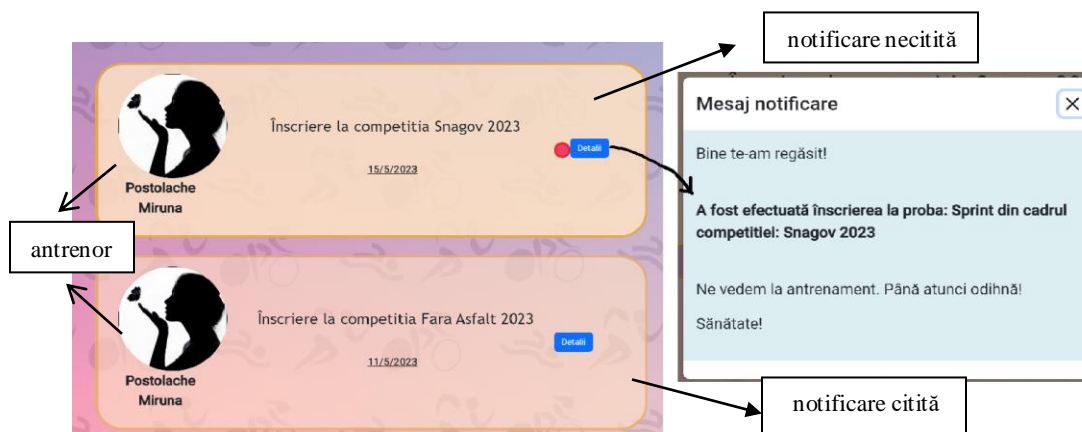


Figura 4.29. – Pagina de notificări

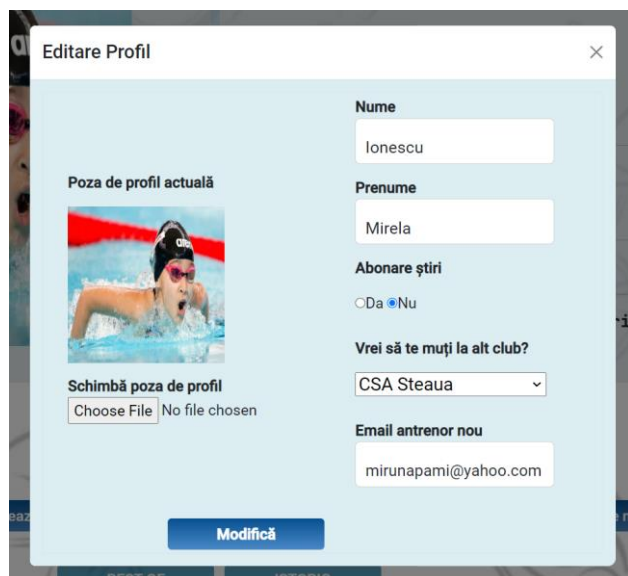
4.5.3. Pagina de profil a sportivului

4.5.3.1. Profilul din perspectiva deținătorului

4.5.3.1.1. Editarea și datele profilului

Pagina de profil îi permite utilizatorului autentificat cu rol de sportiv să gestioneze și să vizualizeze datele introduse la înregistrare, precum: poza de profil, numele complet, genul, anul nașterii, denumirea clubului actual, numărul legitimației, numele antrenorului curent și informații legate de abonarea la știri – primirea notificărilor pe adresa de email.

În cazul în care sportivul dorește modificarea datelor din profilul său, acesta trebuie să apese butonul „Editare profil” care va deschide o fereastră de modal ca în *Figura 4.30*.



Dacă sportivul decide să se mute la alt club, acesta îl poate selecta din lista cluburilor afiliate federației, trebuind să introducă și adresa de email a noului antrenor (dată de acesta). După ce modificarea a fost făcută, în istoricul cluburilor, pentru sportivul curent, este trecută perioada petrecută la clubul anterior fapt ce ajută la stabilirea corectă a legăturii club – competiție.

De asemenea, sportivul se poate muta și la un alt antrenor din cadrul aceluiași club, fapt ce nu implică actualizarea istoricului de cluburi.

Figura 4.30. – Modal de editare a profilului sportivului

4.5.3.1.2. Vizualizarea antrenorului

În cadrul paginii de profil a sportivului se poate accesa profilul public al antrenorului prin apăsarea butonului „Antrenorul meu”.

4.5.3.1.3. Completarea fișei anuale

Pentru a-și reînnoi legitimația de triatlon, sportivii trebuie să completeze anual un formular unde trebuie să încarce documente, precum: poza de profil, avizul medical și buletinul sau certificatul de naștere. După ce se încarcă toate fișierele în format *.png* și se apasă butonul prin care se finalizează completarea formularului, acesta va fi inactiv pentru o perioadă de un an.

4.5.3.1.4. Vizualizarea fișelor completate

Atunci când sportivul apasă butonul „Fișele mele” din profilul său, acesta este redirectionat către o pagină care conține toate formularele pe care le-a completat. Pentru fiecare dintre acestea, se afișează data completării și un buton – disponibil pentru a accesa pagina fișei respective unde sportivul poate vedea cele trei documente încărcate asociate formularului.

4.5.3.1.5. Vizualizarea și editarea recenziilor

În pagina de profil se regăsește butonul „Recenziile mele” care redirecționează sportivul spre o pagină care conține în ordine invers cronologică toate recenziile oferite. Mai mult, acestea au opțiunea de a edita mesajul recenziei (vezi *Figura 4.31.*).



Figura 4.31. – Pagina de afișare și editare a recenziilor

4.5.3.1.6. Best of și istoricul rezultatelor sportivului

„TRI România” îi oferă posibilitatea sportivului să își urmărească evoluția în cadrul triatlonului prin afișarea în pagina profilului a două componente importante: best of și istoricul participărilor.

Cea dintâi presupune afișarea, pentru fiecare probă la care a luat parte, a celui mai bun timp obținut și a altor informații semnificative ca în *Figura 4.32.*

Probă: Sprint		Timp Total: 01:17:00		Loc/Perioadă concurs Snagov 2021 17/7/2021
ÎNOT	TRANZIT1	CICLISM	TRANZIT2	ALERGARE
00:20:00	00:01:00	00:30:00	00:01:00	00:25:00

Figura 4.32. – Componenta Best of sportiv

Cea de-a doua componentă presupune afișarea detaliată a rezultatelor pentru toate competițiile la care a luat parte. În cazul din *Figura 4.33*, pentru prima competiție nu s-au trecut încă rezultatele, datele fiind cele din timpul înregistrării sportivului la acest concurs.

Fara Asfalt 2023 (28/5/2023 - 29/5/2023)

Probă	Timp Total	ÎNOT	TRANZIT1	CICLISM	TRANZIT2	ALERGARE	Loc General	Loc Gen	Loc Categorie	Puncte
Sprint	99:99:99	99:99:99	99:99:99	99:99:99	99:99:99	99:99:99	999	999	999	0

Tulcea Delta Triathlon 2018 (17/6/2018 - 18/6/2018)

Probă	Timp Total	ÎNOT	TRANZIT1	CICLISM	TRANZIT2	ALERGARE	Loc General	Loc Gen	Loc Categorie	Puncte
Supersprint	00:35:06	00:06:12	00:00:20	00:19:15	00:00:10	00:09:19	1	1	1	100

Figura 4.33. – Componenta Istoric sportiv

4.5.3.2. Profilul sportivului din perspectiva publică

Pagina publică a profilului unui sportiv cuprinde datele generale despre acesta, precum: numele complet, genul, anul nașterii și clubul din care face parte, dar și o imagine de profil. În plus, profilul public conține trei butoane care oferă acces la best of-ul, istoricul și postările sportivului curent. Prin expunerea publică a primelor două, se oferă posibilitatea, în special a sportivilor, de a se compara între ei și de a identifica secțiunile unde mai au de făcut îmbunătățiri pentru a atinge sau chiar a depăși performanțele altora.

4.6. Interfața utilizatorului cu rol de antrenor

4.6.1. Înscrierea sportivilor la competiții

Utilizatorii autentificați drept antrenori au posibilitatea de a-și înscrie sportivii la competiții. Pentru a face posibil acest lucru, antrenorul trebuie să aleagă concursul unde își înscrie sportivii din pagina „Competiții” accesibilă din bara de meniu. Aflat în pagina ce cuprinde detaliile competiției, spre deosebire de interfața specifică sportivului, antrenorul are în plus butonul pentru înscrierea sportivilor care deschide o fereastră de modal. Aceasta din urmă conține trei câmpuri de completat: numărul legitimației sportivului care urmează să fie înscris, numele probei și categoria curentă a acestuia, aleasă dintr-o listă de opțiuni. Ultimele două prezintă reguli de validare: proba – să fie introdusă corect („Sprint”/ „Supersprint”/ „Olimpic”/ „Ironman”/ „Half Ironman”) și categoria – vârsta sportivului să corespundă cu cea a intervalului categoriei alese și genul să fie corespunzător.

În urma finalizării înscrierii, dacă sportivul avea selectată opțiunea de a fi abonat la știri, atunci federația îi va trimite pe adresa de email o înștiințare cu privire la proba și concursul la care a fost adăugat, pe lângă notificarea primită în aplicație (vezi *Figura 4.34.*).

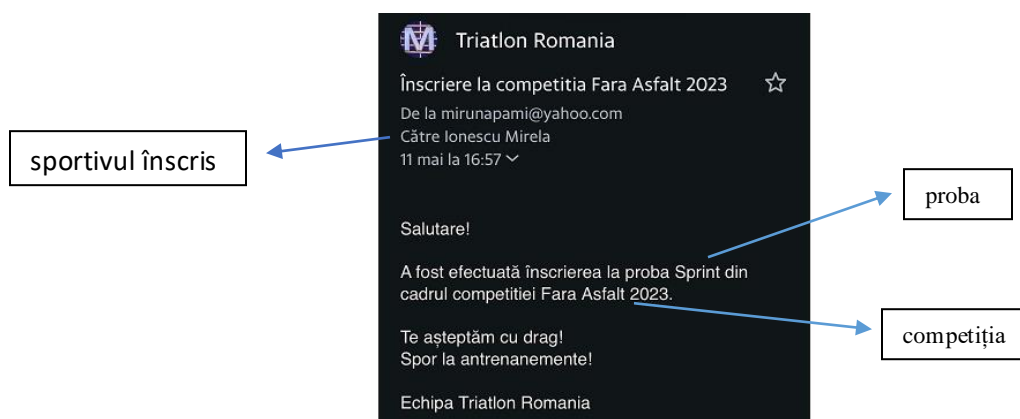


Figura 4.34. – Email primit de sportiv după efectuarea înscrierii

4.6.2. Pagina de profil a antrenorului

4.6.2.1. Profilul din perspectiva deținătorului

4.6.2.1.1. Editarea și datele profilului

Pagina de profil îi permite utilizatorului autentificat cu rol de antrenor vizualizarea datelor, precum: poza de profil, numele complet, gradul de pregătire, denumirea clubului actual, adresa de email și informații legate de abonarea la știri – primirea notificărilor pe adresa de email. Pentru modificarea de date din profilul său, acesta trebuie să apese butonul „Editare profil” care va deschide o fereastră de modal ca în *Figura 4.35*.

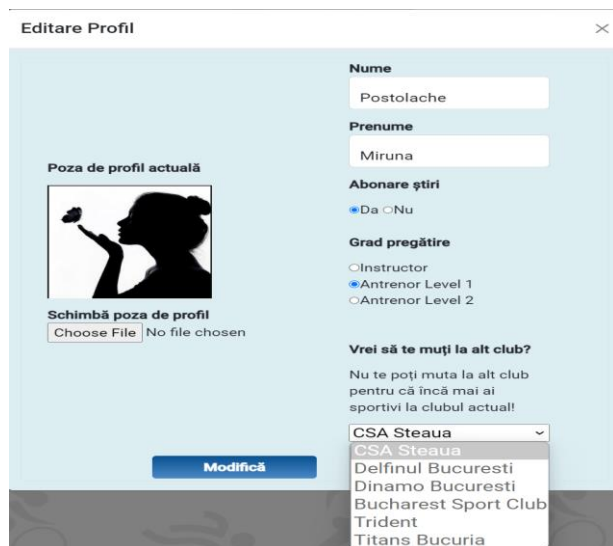


Figura 4.35. – Modal de editare a profilului antrenorului

Un antrenor se poate muta la un alt club doar dacă nu mai are asociat niciun sportiv din clubul la care este încă angajat.

În cazul în care mutarea este posibilă, se va insera perioada în care a profesat la fostul club în istoricul cluburilor.

Altfel, opțiunile din lista cluburilor afiliate federației sunt blocate. În acest caz, antrenorul trebuie să aștepte transferul sportivilor săi la alt antrenor.

4.6.2.1.2. Vizualizarea sportivilor proprii

În pagina de profil există butonul „Sportivii mei” care în urma apăsării afișează toți sportivii antrenorului logat. Aceștia pot fi filtrați în funcție de gen și de anul de naștere, valorile inițiale fiind „toate genurile” și „toți anii” (vezi *Figura 4.36.*).

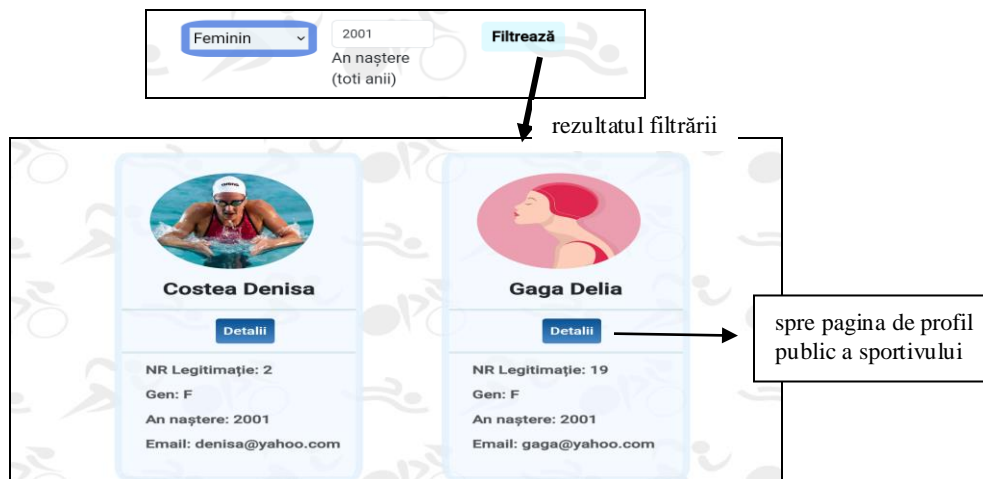


Figura 4.36. – Sportivii antrenorului din pagina de profil

4.6.2.1.3. Studiarea statisticilor

Pentru a avea o imagine mai clară asupra rezultatelor sportivilor la competiții, în profilul antrenorului, prin apăsarea butonului „Statistici” se deschide o componentă ce conține două tipuri de grafice cu roluri diferite ca în *Figura 4.37.* Cel din stânga contorizează numărul de medalii obținut pentru fiecare loc, pe când celălalt prezintă procentul sportivilor medaliați și nemedaliați în cadrul tuturor competițiilor sau a uneia specifice aleasă din câmpul de filtrare. Un sportiv este considerat medaliat dacă a ocupat cel puțin un loc pe podium.

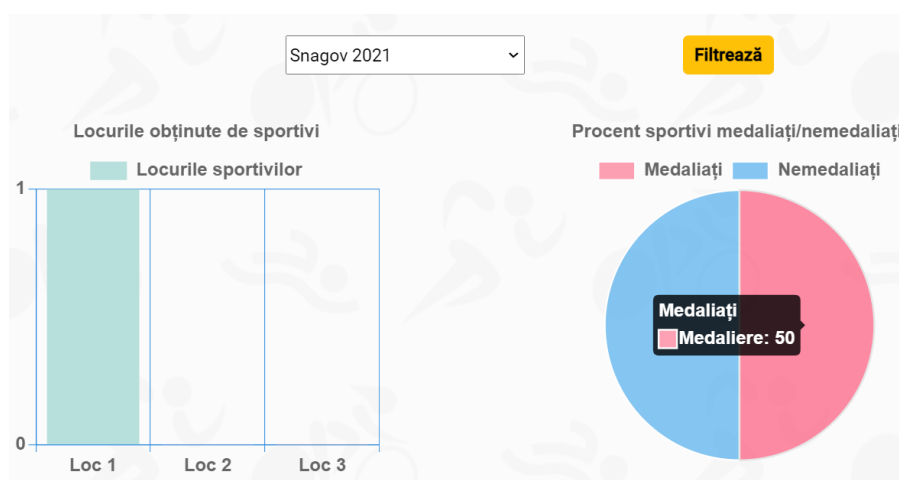


Figura 4.37. – Statistici privind rezultatele sportivilor unui antrenor

4.6.2.2. Profilul antrenorului din perspectiva publică

Pagina publică a profilului unui antrenor cuprinde datele generale despre acesta, precum: numele complet, gradul de pregătire și clubul din care face parte, dar și o imagine de profil. În plus, aceasta conține și postările adăugate de antrenor.

4.7. Meniul aplicației

Meniul este diferit în funcție de tipul utilizatorului.



Figura 4.38. – Meniul aplicației pentru vizitator

În cazul vizitatorilor, bara de meniuri conține link-uri spre paginile „Cluburi”, „Competiții”, „Statistici” și „Logare” (vezi Figura 4.38.).



Figura 4.39. – Meniul aplicației pentru antrenor (stânga) și sportiv (dreapta)

În ceea ce privește autentificarea ca antrenor, meniului i se adaugă poza de profil care duce utilizatorul spre pagina de profil, „Postări”, iconița de mesagerie care redirecționează spre pagina de Chat și opțiunea de „Logout”. Spre deosebire de acesta din urmă, bara de meniuri din interfața de sportiv are în plus iconița sub formă de plic care trimite utilizatorul spre pagina de notificări (vezi Figura 4.39.).



Figura 4.40. – Meniul aplicației pentru administrator

Administratorul are cea mai complexă bară de meniuri, conținând un număr de 10 opțiuni care fac legătura spre paginile: „Videoclip”, „Club”, „Probă”, „Competiție”, „Tip”, „Recenzie”, „Comentariu”, „Formular”, „Postare” și butonul de „Logout” (vezi Figura 4.40.).

În toate variantele de meniuri, în partea stângă se află sigla aplicației.

CONCLUZII

Această lucrare cuprinde prezentarea scopului, a tehnologiilor și a modului de funcționare al aplicației „TRI România”, dezvoltată cu ajutorul Angular, ASP.NET Core Web API, SignalR, Ng-Bootstrap și a API-urilor de Google Maps.

Aplicația a fost concepută pentru a le oferi antrenorilor și sportivilor din cadrul triatlonului o platformă prin care să interacționeze mult mai ușor și să poată primi informații de interes legate de competițiile desfășurate și rezultatele obținute la acestea. Mai mult, acesta este cel mai amplu proiect dezvoltat în perioada facultății, în care deși am folosit două tehnologii care au fost prezentate pe parcursul anilor de studii (Angular, ASP.NET) am avut ocazia să le aprofundez și să descopăr aspecte noi. Ceea ce am folosit pentru prima dată în această aplicație au fost API-urile de la Google Maps, Ng-Bootstrap, SignalR și integrarea de videoclipuri din aplicația YouTube.

Pe parcursul implementării am întâlnit o serie de provocări pe care le-am depășit prin efort și determinare, având în minte obiectivul de a crea un produs final valoros pentru comunitatea de triatlon prin funcționalitățile pe care le oferă. Un exemplu în acest sens poate fi realizarea unui chat în timp real, având în vedere că a fost prima interacțiune cu biblioteca SignalR. La început am întâmpinat dificultăți în înțelegerea funcționării acesteia, însă, în urma unei perioade de cercetare, am reușit să implementez versiunea de chat prezentă în aplicație.

Consider că „TRI România” este o aplicație care și-a îndeplinit scopul stabilit inițial și care de asemenea poate fi perfecționată prin aducerea de îmbunătățiri, precum: introducerea plății cu cardul direct din platformă pentru achitarea taxei de participare la înscrierea sportivilor de către antrenor. În plus, se pot adăuga mai multe informații cu privire la programul și traseul competițiilor, dar și a unei pagini unde să se posteze anunțuri legate de evenimentele organizate de Federația Română de Triatlon. De asemenea, o altă opțiune de extindere a platformei este includerea regulamentelor interne ale federației.

Astfel, aplicația „TRI România” oferă utilizatorilor o gamă largă de funcționalități care sunt descrise în capitolul anterior, având o interfață primitoare și ușor de utilizat. Totodată, există potențialul de a îmbunătăți și mai mult aplicația prin adăugarea funcțiilor menționate anterior și a altora, în conformitate cu cerințele viitoare. Acest fapt asigură că aplicația rămâne actualizată și capabilă să satisfacă în continuare nevoile utilizatorilor în concordanță cu evoluțiile din domeniul triatlonului.

SURSELE IMAGINILOR PRELUATE DIN APLICAȚIE

În cele ce urmează se vor preciza sursele de unde au fost preluate imaginile utilizate în aplicație. Un aspect important de menționat este faptul că pozele au fost obținute gratuit.

- Sigla aplicației face parte din sigla folosită de site-ul oficial al Federației Române de Triatlon: <https://triatlonromania.ro/wp-content/uploads/2023/01/Triatlon-Romania-2023-10-ani-logo-web.webp>
- Dintre pozele care fac parte din carusel doar două (prima și ultima) au fost preluate de pe următoarele site-uri:
 - <https://www.biciclistul.ro/wp-content/uploads/Peter-Klosz-presedinte-Federatia-Romana-de-Triatlon.jpg>
 - <https://pixabay.com/photos/sports-race-run-triathlon-2984610/>
- Imaginile de fundal:
 - <https://www.pexels.com/photo/unrecognizable-men-on-powerboat-on-wavy-sea-near-swimming-athletes-5687400/>
 - <https://www.pexels.com/photo/people-swimming-on-body-of-water-1936939/>
 - <https://www.pexels.com/photo/people-swimming-on-sea-water-5094957/>
 - https://www.freepik.com/free-vector/group-cute-little-girl-boy-holding-gold-trophy-podium_29006329.htm#&position=1&from_view=search&track=ais
 - <https://wall.alphacoders.com/big.php?i=1026345>
- Siglele sponsorilor și ale rețelelor de socializare:
 - <https://seeklogo.com/vector-logo/307866/orbea-azul>
 - https://trisport-120ac.kxcdn.com/image/cache/catalog/trisport_logo_blue2019-211x100-211x100.png
 - <https://logowik.com/brenntag-logo-vector-svg-pdf-ai-eps-cdr-free-download-17629.html>
 - <https://worldvectorlogo.com/logo/enel-group-1>
 - <https://seeklogo.com/vector-logo/130518/spiuk>
 - <https://triatlonromania.ro/wp-content/uploads/2020/05/Gold-Nutrition-logo.jpg>
 - https://global-uploads.webflow.com/60f984be4b08cd98e3d7b4/615437e03eba59ce7f2aa7a9_Project%20icon.svg
 - <https://logowik.com/garmin-vector-logo-4443.html>
 - <https://logowik.com/tyr-sport-logo-vector-40233.html>

- <https://www.dendrio.com/wp-content/uploads/2019/02/Dendrio-YJU-color.jpg>
 - <https://www.logotypes101.com/logo/perskindol>
 - <https://triatlonromania.ro/wp-content/uploads/2022/12/sloop-logo.webp>
 - <https://icons8.com/icons/set/instagram>
 - <https://icons8.com/icons/set/facebook>
 - <https://www.iconpacks.net/free-icon/youtube-logo-2431.html>
- O mare parte a pozele de profil, a imaginilor asociate cluburilor și cele din cadrul postărilor au fost preluate gratuit din diverse site-uri.

BIBLIOGRAFIE

- [1] ABLY Platform, *SignalR vs. Firebase: Learn about their differences, similarities, and use cases*, 19 mai 2023, <https://ably.com/topic/signalr-vs-firebase>, data ultimei accesări: 31.05.2023.
- [2] ABLY Platform, *SignalR vs. Socket.IO: which one is best for you?*, 09 mai 2023, <https://ably.com/topic/signalr-vs-socketio>, data ultimei accesări: 31.05.2023.
- [3] ABLY Platform, *SignalR vs. WebSocket: Key differences and which to use*, 08 noiembrie 2022, <https://ably.com/topic/signalr-vs-websocket>, data ultimei accesări: 31.05.2023.
- [4] Rick Anderson, *Introduction to Identity on ASP.NET Core*, <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-7.0&tabs=visual-studio>, data ultimei accesări: 31.05.2023.
- [5] Angular, *What is Angular?*, <https://angular.io/guide/what-is-angular#what-is-angular>, data ultimei accesări: 31.05.2023.
- [6] Angular, *Introduction to Angular concepts*, <https://angular.io/guide/architecture>, data ultimei accesări: 31.05.2023.
- [7] Angular, *Introduction to components and templates*, <https://angular.io/guide/architecture-components#templates-and-views>, data ultimei accesări: 31.05.2023.
- [8] Angular, *Introduction to components and templates*, <https://angular.io/guide/architecture-components#component-metadata>, data ultimei accesări: 31.05.2023.
- [9] Angular, *Introduction to components and templates*, <https://angular.io/guide/architecture-components#directives>, data ultimei accesări: 31.05.2023.
- [10] Angular, *Introduction to components and templates*, <https://angular.io/guide/architecture#routing>, data ultimei accesări: 31.05.2023.
- [11] Bootstrap widgets, <https://ng-bootstrap.github.io/#/components/table/overview>, data ultimei accesări: 31.05.2023.
- [12] Chart.js, <https://www.chartjs.org/docs/latest/>, data ultimei accesări: 31.05.2023.

- [13] Patrick Fletcher, *Introduction to SignalR*, <https://learn.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr>, data ultimei accesări: 31.05.2023.
- [14] Google Maps Platform, *Google Maps Platform Documentation*, <https://developers.google.com/maps/documentation>, data ultimei accesări: 31.05.2023.
- [15] Microsoft, *What is .NET?*, <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>, data ultimei accesări: 30.05.2023.
- [16] Microsoft, *What is ASP.NET Core?*, <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core>, data ultimei accesări: 30.05.2023.
- [17] Microsoft, *What is ASP.NET Core SignalR?*, <https://learn.microsoft.com/en-us/training/modules/aspnet-core-signalr/2-what-is-signalr>, data ultimei accesări: 31.05.2023.
- [18] Microsoft, *Entity Framework documentation hub*, <https://learn.microsoft.com/en-us/ef/>, data ultimei accesări: 31.05.2023.
- [19] Microsoft, *Design the infrastructure persistence layer*, 21 februarie 2023, <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-design>, data ultimei accesări: 31.05.2023.
- [20] Mukesh Murugan, *Repository Pattern in ASP.NET Core – Ultimate Guide*, 24 aprilie 2021 <https://codewithmukesh.com/blog/repository-pattern-in-aspnet-core/>, data ultimei accesări: 31.05.2023.
- [21] Mehmet Ozkaya, *Layered (N-Layer) Architecture*, 6 septembrie 2021, <https://medium.com/design-microservices-architecture-with-patterns/layered-n-layer-architecture-e15ffdb7fa42>, data ultimei accesări: 31.05.2023.
- [22] npm, *ng2-charts*, <https://www.npmjs.com/package/ng2-charts>, data ultimei accesări: 31.05.2023.
- [23] npm, *ngx-youtube-player*, <https://www.npmjs.com/package/ngx-youtube-player>, data ultimei accesări: 31.05.2023.
- [24] Nihar Raval, *React vs Angular: Which JS Framework to Pick for Front-end Development?*, 27 februarie 2023 <https://radixweb.com/blog/react-vs-angular#advantages-tools>, data ultimei accesări: 31.05.2023.

- [25] Margaret Rouse, Multipurpose Internet Mail Extensions, 01 februarie 2017, <https://www.techopedia.com/definition/1693/multipurpose-internet-mail-extensions-mime>, data ultimei accesări: 31.05.2023.
- [26] Karan Pratap Singh, *System Design Long polling, WebSocket, Server-Sent Events (SSE)*, 16 septembrie 2022, <https://dev.to/karanpratapsingh/system-design-long-polling-websockets-server-sent-events-sse-1hip>, data ultimei accesări: 31.05.2023.
- [27] Raja Tamil, *Google Maps API Geocode Explained [2022]*, 17 aprilie 2023, <https://softauthor.com/google-maps-api-geocode-javascript/>, data ultimei accesări: 31.05.2023.
- [28] TypeScript, <https://www.typescriptlang.org/>, data ultimei accesări: 31.05.2023.
- [29] Shanika Wickramasinghe, *Angular vs Vue: A Head-to-Head Comparison*, 17 februarie 2023, <https://kinsta.com/blog/angular-vs-vue/#components-and-extensibility>, data ultimei accesări: 31.05.2023.
- [30] Assis Zang, *ASP.NET Core for Beginners: Web APIs*, 19 noiembrie 2021, <https://www.telerik.com/blogs/aspnet-core-beginners-web-apis>, data ultimei accesări: 31.05.2023.