

UNIVERSITATEA NAȚIONALĂ DE ȘTIINȚĂ ȘI TEHNOLOGIE
POLITEHNICA BUCUREȘTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL DE CALCULATOARE



PROIECT DE DIPLOMĂ

Ligă Sportivă de baschet

Surdu Cristian

Coordonator științific:

Prof. dr. ing. Costin Anton Boianțiu

BUCUREȘTI

2024

NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
POLITEHNICA BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE DEPARTMENT



DIPLOMA PROJECT

Basketball League

Surdu Cristian

Thesis advisor:

Prof. dr. ing. Costin Anton Boiangiu

BUCHAREST

2024

CUPRINS

Sinopsis	2
Abstract.....	2
1.1 Context.....	3
1.2 Problema.....	4
1.3 Obiective	5
1.4 Structura lucrării	6
2 Analiza și specificarea cerințelor.....	7
2.1 Cerințe de sistem	7
2.2 Cerințe funcționale	7
3 Studiu de piață / Abordări existente.....	11
3.1.1 NBA.com	11
3.1.2 AMBB	12
3.1.3 ESPN	13
4 SOLUTIA PROPUSA	14
4.1 Arhitectura si infrastructura aplicației	14
4.2 Baza de date.....	15
4.3 Fluxul aplicației	17
5 Detalii de implementare	19
5.1 Tehnologii.....	19
5.2 Backend.....	21
5.3 Frontend.....	24
6 Studiu de caz / Evaluarea rezultatelor	30
7 Concluzii	31
7.1 Dezvoltări ulterioare	31
8 Bibliografie	32
9 Anexe	34

SINOPSIS

Proiectul dezvoltat se adresează problemelor comune în managementul ligilor de baschet din toată lumea: monitorizarea și administrarea detaliată a activităților din cadrul acestora. Platforma software propune un sistem de tipul client-server care susține fluxul unei ligi sportive dedicate exclusiv baschetului. Încadrarea utilizatorilor care sunt destinați să folosească platforma se împarte în trei mari categorii: căpitan care are datoria principală de a crea echipa și a decide care va fi lotul component al echipei pe bază unei analize amănunțite a atributelor celorlalți jucători, jucătorul normal care are datoria de a înscrie cât mai multe puncte într-un mod cât mai eficient pentru a putea fi selectat ulterior la o echipa mai bună și administrator care are rolul de a simula și crea meciuri. Pe lângă aceste atribute principale cele două tipologii de utilizatori împart și alte funcționalități din cadrul aplicației. Logica componentelor din cadrul sistemului este menită să ușureze viața atleților profesioniști dar și a proprietarilor/antrenorilor de echipe pentru ca aceștia să se poată concentra exclusiv pe partea fizică a jocului.

Sunt prezente numeroase surse de informare precum statistici și analize pentru a putea îndruma clar echipele înscrise către obiectivele lor și a oferi o imagine generală asupra organizațiilor sportive implicate. Pe lângă cele deja menționate precum crearea de echipe, statistici procentuale direct proporționale cu coșurile înscrise/ratate de fiecare jucător există și o posibilitate de simulare a unui meci direct între două echipe. Motivația din spatele alegerii acestei teme este însăși propria experiență din cadrul unei organizații sportive în care am fost implicat pe o durată de zece ani atât la nivel de juniorat cât și profesionist. Viața de atlet m-a format ca om și m-a încurajat să îmi doresc ulterior o implicare directă în partea organizatorică a sportului. Polivalența de a fi atât jucător cât și antrenor, să înțeleg mecanicele din spatele fiecărei schimbare tactică, negocieri pentru contractele jucătorilor cât și importanța fiecărui meci în parte pentru a putea calcula și estima atingerea scopului final, câștigarea unui trofeu.

ABSTRACT

The developed project addresses common issues regarding the management of basketball leagues worldwide: detailed monitoring and management of activities within them. The software platform proposes a client-server system that supports the workflow of a sports league dedicated exclusively to basketball. The users intended to use this platform are divided into three main categories: the captain, who has the main focus of developing a team from scratch by deciding the components of the roster based on a thorough analysis of the other players' attributes, and the regular player, who has the duty to stand out by scoring as many points as possible with a pinpoint accuracy in order to be selected later for a better team and the admin who needs to create and simulate games. In addition to these main attributes, the two types of users share other functionalities within the application. The logic of the system components is designed to make the lives of professional athletes and team owners/coaches easier so they can focus exclusively on the physical part of the game. There are numerous sources of information such as statistics and analyses to clearly

guide the registered teams toward their objectives and provide a general overview of the involved sports organizations. Besides the already mentioned features, like team creation, percentage statistics directly proportional to the points scored/missed by each player, there is also the possibility to simulate a direct match between two teams.

The motivation behind choosing this theme is my own experience within a sports organization where I was involved for ten years, both at the junior and professional levels. The life of an athlete made me the person I am today and encouraged me to later want a direct involvement in the organizational side of the sports. The versatility of being both a player and a coach, to understand the mechanics behind each tactical change, negotiations for player contracts, and the importance of each match to calculate and estimate the achievement of the final goal, winning a trophy.

1 INTRODUCERE

1.1 Context

Sportul reprezintă atât o activitate recreativă, menită să te separe de problemele cotidiene permițându-ți să te focusezi pe un ideal scurt și concis în decursul a câteva ore, să te distrezi, cât și o viitoare carieră de succes pentru tinerii talentați și muncitori care sunt dispuși să își sacrifice tinerețea în scopul perfecționării calităților fizice. Pentru mulți atleți atât din România cât și din străinătate, sportul profesionist încă de la o vârstă fragedă devine singura cale de scăpare dintr-o situație nefavorabilă. Mulți jucători tineri încep să se perfecționeze pe o anumită ramură fie ea fotbal, baschet sau alt sport cu visul clar de a ajunge să joace într-o ligă profesionistă, să prindă contracte și sponsorizări cu care își pot asigura un trai de viitor, astfel pentru ei sportul devine un stil de viață și totodată un loc de muncă.

Deși la prima vedere această alegere pare perfectă pentru mulți, realitate este cu totul alta. Rata de succes a unui atlet în zilele de astăzi este undeva sub doi la sută în funcție de sport după cum confirmă următorul articol [1], lucru care obligă acești tineri să ia decizii radicale încă din juniorat, într-un moment al vieții în care nu sunt suficienți de maturi pentru astfel de decizii legate de viitorul lor. Soluția pentru majoritatea este una simplă și clară, ligile de juniori. Fie că ești la început de drum, fie că ești deja un sportiv de renume, nimic nu ar fi posibil fără o organizație care se îți pună la dispoziție un teren cu toate facilitățile necesare, un loc în care fanii se pot aduna să te susțină și un premiu final. Sportul este total dependent de fani, sursa principală de venit în orice sport, iar fanii vor fi mereu atrași de o ligă care unește echipele lor preferate într-o singură unitate centrală care poate fi mediatizată internațional. Ligile sportive nu sunt o soluție apărută recent, iar sportul ca și competiție poate fi găsit încă din antichitate prin prisma jocurilor olimpice [2], prima și cea mare liga sportivă polivalentă din lume, care adună miliarde de spectatori chiar și în zilele noastre. Răspunsul la întrebarea de ce liga sportivă este simplu, sportul pentru oameni este aproape la fel de vital ca apa fiind consumat zilnic în toate colțurile lumii, iar liga pentru un

sport reprezintă scena pe care acesta se poate desfășura, fiind scheletul care ține toate celelalte componente într-un singur loc. Sportul unește oamenii indiferent de naționalitate sau religie pentru a susține un scop comun.

1.2 Problema

Sportul, ca orice alt domeniu ar fi inutil fără o structură și un set de reguli. Așa cum elevii au nevoie de școală pentru a putea ști ce să învețe și să se formeze atât din punct de vedere social cât și academic, ligile sportive reprezintă unitatea centrală care promovează activitățile fizice dintre participanții acestora. Acestea stabilesc un cadru organizat și profesionist pentru diferite competiții, asigurând un spațiu propice pentru desfășurarea în siguranță a activităților de acest gen. Totodată datorită naturii competitive care ne face zilnic să evoluăm pot apărea oricând probleme și neînțelegerilor între participanți datorită diferențelor de opinii, lucru care este totodată și el rezolvat prin existența persoanelor calificate care pot să ia decizii în cadrul acestor competiții, adică arbitrii. Ligile creează calendare și programe clare și detaliate la care echipele participante trebuie să adere astfel încât să se asigure o optimizare logistică. De asemenea ligile stabilesc înaintea oricărui sezon un set de reguli și reglementări, care trebuie respectate de toți participanții, astfel asigurând corectitudinea și meritul câștigătorilor, cât și protecția fizică și psihică a celor implicați prin respectarea unei anumite conduite bine stabilite.

Toată lumea a auzit măcar o dată sfatul părintesc „Talentul fără muncă nu înseamnă nimic”, însă problema se întinde mai departe de atât. În lumea sportului talentul și munca fără un mijloc de expunere socială este la fel de inutil. Ligile sportive oferă tinerilor cel mai crucial avantaj, o platformă unde își pot arăta talentele. Numeroasele turnee și competiții la care aceștia participă îi ajută atât pe plan tehnic pentru a își dezvolta în continuare atributele sportive, a învăța cum să colaboreze într-o echipă și a primi numeroase alte oportunități de antrenament, cât și social prin prezența unui centru în care oamenii cu aceleași interese se adună într-un singur loc pentru a urmări viitori jucători. Acest beneficiu ne aduce la următoarea problemă majoră, finanțarea. Echipele au nevoie de sponsori pentru a putea călători în diferite orașe și a putea juca în diferite turnee iar ligile care au o expunere vastă în mediul online după cum exemplifică acest articol [3], sunt cele care atrag cei mai mulți sponsori și parteneri financiari, asigurând astfel fondurile necesare pentru funcționarea propice a echipelor.

Nu în ultimul rând ligile sportive ajută cu promovarea sportului în sine fără de care totul ar dispărea. Dacă lumea nu ar mai fi interesată de sport și ar alege să își petreacă timpul liber făcând alte activități recreative, toate celelalte componente dispar ca într-un domino. Oamenii au nevoie de o ligă care să promoveze meciuri și jucători, pe care fanii să-i îndrăgească și să îi motiveze să își rupă din timpul personal pentru a susține echipa favorită atât prin simpla prezență fie ea fizică sau virtuală cât și prin echipamente și alte comodități pe care aceștia le cumpără doar fiindcă au emblema clubului respectiv.

1.3 Obiective

Obiectivele acestui proiect sunt de a ataca problemele exemplificate anterior prin centralizarea tuturor facilităților care eficientizează procesul de interacțiune între utilizator și sistem într-o singură platformă multifuncțională. Prin rezolvarea provocărilor curente se dorește depășirea barierei tradiționale, mai exact pixul și foile de statistici printate după fiecare meci, prin utilizarea instrumentelor digitale. Astfel utilizatorii se pot bucura de o platformă prietenoasă și intuitivă, care este gândită specific pentru nevoile lor, iar componenta tehnică a echipelor se poate bucura de o eficientizare majoră a timpului, putând să își concentreze resursele pentru recrutarea de noi talente, calcularea procentajelor de victorii/înfrângeri și alte statistici importante.

Platforma vrea să ofere funcționalități pentru toate tipologiile de utilizatori, atât jucători cât și echipe, indiferent de scopul pe care îl au acestea atunci când hotărăsc să se autentifice pe platforma respectivă. Echipa își pot gestiona singure înscrierile prin intermediul funcționalității de invitații, iar jucătorii pot accepta sau respinge aceste invitații în funcție de propriile preferințe. Totodată vor dispune și de un sistem de notificări pentru a putea naviga ușor către pagina de profil atunci când sunt invitați într-o echipă. Odată acceptată o invitație jucătorii nu trebuie să își mai bată capul cu refuzul celorlalte invitații în cazul în care erau doriți de mai multe echipe deoarece ele vor fi anulate automat, iar căpitani care au făcut respectivele invitații vor fi notificați.

La nivel de analiză, platforma dispune de numeroase metrice utile atât jucătorilor cât și echipelor cu scopul de a asigura o ierarhie cât mai precisă și detaliată. Jucătorii vor fi monitorizați în funcție de procentajul de coșuri înscrise, fiind separate pe 3 categorii în funcție de distanța de la care se aruncă la coș: un punct, două puncte sau trei puncte. Echipa sunt ierarhizate pe baza a numeroși și diferiți factori prezenți. Există numeroase filtre în tabelul acestora precum: număr de meciuri câștigate, număr de puncte marcate în sezon cât și filtre de căutare în funcție de nume. Jucătorii primesc o analiză și mai detaliată, ei fiind monitorizați nu numai din punctul de vedere al volumului de puncte marcate cât și al ratei de succes atunci când aruncă la coș, ierarhizarea lor fiind structurată pe baza procentului de înscrieri din fiecare poziție de pe teren.

Există totodată și posibilitatea de simulare de meciuri, care creează un meci direct între două echipe din liga competițională și afișează în timp real scorurile pentru fiecare echipă cât și statisticile complete ale fiecărui jucător prezent în meciul respectiv, urmată de o istorie a meciurilor jucate cu scopul de a înțelege cum se clasează fiecare echipă împotriva celorlalte, unde mai au de lucrat, și cum au evoluat pe parcursul sezonului. Meciurile sunt totodată și programate astfel încât fiecare echipă să știe clar ce calendar competițional are și cât timp are să se pregătească pentru fiecare meci. Platforma va fi accesibilă tuturor celor interesați de acest sport, indiferent de nivelul de abilitați sau resurse financiare, liga promovând incluziunea și diversitatea tocmai în ideea dezvoltării fiecărei echipe ca într-un

final nivelul să fie unul ridicat pentru toți participanții, iar fiecare utilizator va avea un cont propriu în care își va trece atât datele personale cât și atribute fizice pentru a eficientiza și mai mult procesul de selecție în timp ce se păstrează securitatea și integritatea datelor.

1.4 Structura lucrării

În această lucrare se vor prezenta toate etapele derulate în cadrul dezvoltării platformei mele web dedicată ligilor de baschet, cât și motivele care m-au determinat să aleg această ramură. În capitolul de analiză a cerințelor se discută adevăratele nevoi ale platformei pentru a putea funcționa la cote maxime în modul în care a fost prevăzută cât și cerințele funcționale, nefuncționale și cele de sistem. Totodată se prezintă numeroase cazuri de uz alături de diferite diagrame pentru o mai bună înțelegere a întreg ansamblului și a structurii platformei.

În capitolul care se concentrează pe studiul pieței sunt prezente descrieri succinte legate de alte platforme similare care au servit ca sursă de inspirație și ghid în dezvoltarea ligii mele de baschet, printr-o analiză concisă și detaliată a punctelor de interes, a lipsurilor pe care acestea le au, cât și a punctelor slabe și tari pe care eu le-am considerat eu relevante. Fiecare platformă este detaliată atât din punct de vedere funcțional cât și al impactului pe care îl are în piață alături de o imagine reprezentativă pentru fiecare.

Capitolul în care este descrisă soluția propusă, este destinat descrierii detaliate a platformei pentru a înțelege exact cum funcționează și cum arată arhitectura acesteia în spate. Se prezintă toate modulele aplicației și modul în care acestea se leagă unul de celălalt, detalii legate de designul bazei de date, cum e gândit serverul dar și partea de interfață a utilizatorului. Totodată se prezintă un ghid pas cu pas al funcționării bazat pe fluxul de utilizare al funcționalităților.

Capitolul următor se concentrează pe detalierea specifică legată de fiecare componentă în parte și de codul propriu zis împreună cu tehnologiile folosite pentru a dezvolta sistemul. Tot aici se prezintă și motivații pentru alegerea fiecărei tehnologii în parte pentru a înțelege cum funcționează ele și ce plusuri aduc platformei. Partea de testarea a funcționalității și a rezultatelor obținute în urma acestor teste este exemplificată în capitolul de evaluare a rezultatelor. Fiind vorba despre dezvoltarea unei platforme web se discută atent în aceasta secțiunea și despre problemele întâmpinate pe parcurs.

Acest capitol este urmat apoi de capitolul final care are rol de rezumat al întregii lucrări, reamintind punctele cheie din fiecare capitol prezentat anterior cât și o scurtă motivație și părere personală legată de modalitatea în care platforma mea a rezolvat problemele propuse, totodată prezentând și modalități de îmbunătățire pe viitor a acesteia prin anumite funcționalități pe care n-am reușit să le implementez dar care mi se par utile pe viitor pentru utilizatori. Există și o parte de bibliografie unde se redau linkurile utile dar și alte lucrări sau cărți care au fost folosite pentru inspirație și pentru anumite statistici folosite.

2 ANALIZA ȘI SPECIFICAREA CERINȚELOR

Acest capitol va analiza cerințele manageriale ale unei ligi de baschet din perspectivele tipurilor de utilizatori care se autentifică pe o astfel de platformă: căpitani de echipă , jucători, administratori de ligă. Diferitele tipuri de scenarii vor fi luate în considerare pentru a genera o listă cât mai detaliată a funcționalităților care să cuprindă toate posibilele situații care vizează fluxul platformei. În urma discuțiilor pe care le-am avut cu nenumărați foști antrenori, jucători dar și manageri de ligi deja existente prin prisma experienței personale de fost jucător profesionist de baschet, vizualizați figura A1 din Anexe, reiese că un procent majoritar dintre aceștia își doresc o schimbare în complexitatea cu care aplicațiile deja existente se prezintă în fața utilizatorilor, a incapacității de a te înscrie într-o ligă sportivă fără să ai o echipă deja prestabilită, cât și a lipsei de ligi oficiale dedicate persoanelor care vor să activeze doar ca jucători amatori dar își doresc un mediu profesionist alături de oficiali ai ligii și arbitrii licențiați. Diferitele tipuri de utilizatori se prezintă cu următoarele cerințe:

- Administratorii de ligă sportivă au diferite responsabilități precum: locații meciuri, ore meciuri, echipe participante, managerierea meciurilor.
- Căpitani echipele sunt responsabili de: componență echipă, analiza oponentilor.
- Jucătorii au ca funcționalități: găsire echipe, menținere statistici, verificare orar meciuri.

2.1 Cerințe de sistem

Pentru o funcționare cat mai optima recomandarea este:

- **Sistem de operare:** Windows 10(recomandat)
- **Browser:** Google Chrome sau Mozilla Firefox
- **Dispozitiv:** Laptop sau Desktop
- Conexiune la internet

2.2 Cerințe funcționale

In următoarele puncte voi explica pe scurt funcționalitățile implementate si cerințele acestora

- Administrator: Ștergere echipa, Ștergere utilizator, Programare meciuri.
- Utilizator:

Înregistrare/Autentificare - In urma accesării platformei, dacă utilizatorul este unul nou este nevoit să își creeze un cont cu mail, parolă dar si poză de profil si înălțime.

Pagina de autentificare se bazează pe credențialele create anterior la pasul de înregistrare.

Actualizare profil: Utilizatorul își poate schimba nume, imagine de profil sau poza de profil.

Pagina de Lista Echipe - Vizionarea diferitelor echipe, statisticilor fiecărei echipe cât și accesarea detaliată prin link a unei echipe anume pentru a vedea componenta acesteia și detalii succinte despre fiecare jucător. Există diferite tipuri de filtre pentru a găsi/ordona echipele în funcție de criteriile de căutare pe care le are utilizatorul.

Pagina de Listă jucători - Vizionarea tuturor jucătorilor dintr-o ligă sportivă, statistici detaliate a acestora cât și hyperlink către pagina de profil a jucătorului respectiv. Pagina dispune de filtre personalizate de ordonare/căutare a jucătorilor din liga sportivă.

Pagina de profil - Oferă detalii personale despre utilizator/jucătorul autentificat, procente aruncări din diferite poziții pe teren pentru a putea înțelege mai bine unde are nevoie de antrenament cât și funcționalități de acceptare și respingere invitații în diferite echipe din liga sportivă și un buton de „Închiriere Echipă” în caz că playerul dorește o schimbare în carieră.

Simulare de meci - Motorul care ține liga în viață simulând meciuri între diferite echipe având ca rezultat actualizarea diferitelor statistici atât pentru jucători cât și pentru echipe în funcție de scor, coșuri marcate/ratate. Totul este afișat în interfață în timp real (fiecare coș este variabil între coș de 3 puncte, 2 puncte sau 1 punct) iar utilizatorul va putea vedea fiecare schimbare atât din punct de vedere al punctelor înscrise cât și a celorlalte statistici.

Vizualizare Istoric Meciuri - Posibilitatea de a vizualiza istoricul meciurilor deja jucate și a accesa un anumit meci pentru a vedea detalii importante despre meci precum ora, locația, echipe participante, scor, scor play by play pentru a vedea cine a dat fiecare coș plus toate celelalte statistici prezentate în simularea meciului.

Posibilitatea de a vizualiza meciurile programate pentru a afla detalii legate de oră, locație și oponent.

Prezentare vizuală în Figura A2 din Anexe.

2.3 Cerințe non-funcționale

Acest subcapitol include cerințele non-funcționale care se ocupă de condițiile de funcționare ca total a platformei în pofida funcționării unitare a fiecărui modul în parte:

- **Securitate** - Datele personale ale jucătorilor și echipelor sunt grija primordială a platformei așa că autentificarea se face pe baza un JWT token care asigură o criptare unică pentru fiecare parolă, asigurând integritatea datelor fiecărui utilizator
- **Performanța** - Sistemul este optimizat în așa fel încât să asigure un timp de răspuns prompt a tuturor funcționalităților prezente precum vizionarea meciurilor, actualizarea scorului și statisticilor jucătorilor cât și restul componentelor. Toate aceste componente sunt integrate într-o arhitectură modernă și eficientă menită să maximizeze experiența pozitivă a utilizatorului pe platformă.
- **Fiabilitatea** – Se asigură persistența datelor, fiind extrem de importante pentru desfășurarea conformă a activităților echipelor și a jucătorilor din ligă.
- **Portabilitatea** – Liga de baschet este concepută să ruleze fără efort pe orice sistem de calcul care dispune de un sistem de operare (Windows, Linux, macOS) și de un motor de căutare.
- **Utilizabilitate** – Interacțiunea dintre utilizator și platformă este gândită să mulțumească nevoile tuturor tipurilor de utilizatori, furnizând o interfață intuitivă și prietenoasă care se mulează pe orice persoană indiferent de nivelul acestora de familiarizare cu tehnologia. Poziția pe care o ocupi pe platforma nu îți afectează în niciun fel experiența de utilizare a acesteia, administrator, căpitan respectiv jucător rutele de accesare a diferitelor funcționalități sunt vizibile și ușor de accesat.
-

2.4 Cazuri de utilizare

2.4.1 Creare echipa

Precondiție: Utilizatorul este autentificat ca și utilizator de tip Capitan

Fluxul de bază:

- Utilizatorul se autentifica în contul său
- Sistemul verifică validitatea credențialelor
- Utilizatorul selectează pagina de „Creare/Ștergere Echipă”
- Utilizatorul completează câmpurile de nume și an pentru echipa respectivă
- Dacă nu există altă echipă cu același nume se creează echipa și utilizatorul este automat atribuit echipei respective în calitate de căpitan

Post condiție: Utilizatorul poate vedea echipa și detalii despre ea în lista de echipe și să invite alți jucători să se alăture echipei. Exemplu grafic disponibil Figura A3 din Anexa.

2.4.2 Ștergere echipa

Precondiție: Utilizatorul este autentificat ca si utilizator de tip Administrator

Fluxul de bază:

- Utilizatorul se autentifică în contul său de administrator
- Sistemul verifică validitatea credențialelor
- Utilizatorul selectează pagina de „Creare/Ștergere echipă”
- Utilizatorul completează câmpul aferent numelui echipei pentru a șterge echipa respectivă din sistem

2.4.2 Simulare meci

Utilizatorul are un cont complet funcțional:

Flux de bază:

- Utilizatorul se autentifică în contul său
- Sistemul verifica validitatea credențialelor
- Utilizatorul selectează pagina de „ Manageriere Meciuri”
- Utilizatorul completează câmpurile necesare creării unui meci nou (nume echipe, dată, locație)
- Sistemul generează fiecare secvență dintr-un meci si o afișează in timp real

Post condiție: Utilizatorul poate vedea detalii legate de meciul terminat in urma simulării in lista de meciuri jucate. Exemplu grafic disponibil Figura A4 din Anexe.

2.4.3 Trimitere invitație alăturare echipă:

Utilizatorul are un cont complet funcțional de tip Căpitan:

Flux de bază

- Utilizatorul se autentifică in contul său
- Sistemul verifica validitatea credențialelor
- Utilizatorul selectează pagina de „Profil Utilizator”
- Utilizatorul completează câmpurile necesare(nume echipa, email jucător dorit)

Post condiție: Invitația este trimisă pentru a fi aprobată sau refuzată de jucătorul invitat.

2.4.4 Acceptare invitație alăturare echipa:

Utilizatorul are un cont complet funcțional:

Flux de bază:

- Utilizatorul se autentifică în contul său
- Sistemul verifică validitatea credențialelor
- Utilizatorul selectează pagina de „Profil Utilizator”
- Utilizatorul verifică secțiunea de „Invitațiile tale”
- Selectează butonul ”Acceptă” sau „Refuză”

Post condiție: Invitația a fost acceptată și jucătorul s-a alăturat echipei iar celelalte invitații au primit refuz automat sau refuz direct la invitație. Exemplu grafic disponibil Figura A5 din Anexe.

3 STUDIU DE PIAȚĂ / ABORDĂRI EXISTENTE

3.1 Abordări existente

În ultimii ani tehnologia a evoluat semnificativ, reușind să pătrundă în majoritatea domeniilor printre care se numără și cel sportiv, mai exact al baschetului. Nu este un secret faptul că pe piața din ziua de azi există numeroase aplicații care să se concentreze ori în jurul ligilor mari deja consacrate, ori în jurul sporturilor mai populare precum fotbalul. În acest capitol voi descrie câteva platforme care au servit ca inspirație pentru crearea platformei mele, exemplificând funcționalitățile lor de bază și motivele pentru care sunt atât de populare în acest domeniu cât și probleme pe care aceste platforme le au și pe care eu am încercat să le rezolv prin funcționalitățile mele.

3.1.1 NBA.com

Cea mai populară aplicație destinată exclusiv celei mai mari ligi de baschet de pe glob, platforma oficială prezentă aici [4], „National Basketball League Association” care iese în evidență prin popularitatea pe care o are la nivel atât național în cadrul Americii cât și internațional. Este un lider al profesionalismului, adoptând cele mai noi tehnologii și soluții pentru a maximiza interacțiunea cu fanii, principala sa sursă de venit. Platforma este una complexă cu numeroase funcționalități precum statistici exacte cu o plajă largă de utilități de la cele ale jucătorilor și echipelor până la cele legate de impactul zonei geografice și a veniturii fiecărei echipe. Totodată platforma

se remarcă prin multitudinea de conținuturi media, meciuri, interviuri post-game cât și legături cu diferite posturi tv și de jurnalism care pătrund adânc în viețile atleților ajutând fanii să simtă o conexiune cât mai strânsă cu jucătorul lor preferat. Această platformă poate fi considerată un magnat în industria sportivă, având o putere de se mula pe cerințele și trendurile actuale, reușind totodată să își păstreze integritatea și nuanța pe care o cultivă încă din anul 1920 când a apărut pentru prima oară această ligă sportivă. De asemenea aplicația funcționează și ca o rețea de social media, permițând fanilor să lase comentarii, fiind legată cu toate rețele de socializare existente astfel permițând o globalizare rapidă și o întindere extinsă în toate colțurile globului.

Principalele funcționalități sunt reprezentate de statistici avansate verificate de analiști profesioniști din lumea sportului, program al meciurilor și al turneeelor disponibil non-stop, ierarhie a echipelor și a jucătorilor în funcție de o multitudine de metrice, conținut multimedia de cea mai înaltă calitate cât și integrare mobilă. Deși este o platformă atât de complexă care pare să mulțumească absolut orice cerință din partea consumatorilor, însăși această complexitate vine la pachet cu câteva probleme pe care platforma mea a încercat să le rezolve. Multitudinea de informații și variante de a naviga platforma face greu pentru consumatorul ocazional care vrea doar să își urmărească echipa preferată să înțeleagă cum funcționează întreg ansamblul de linkuri, butoane și postări.

Timpul necesar în familiarizarea cu platforma poate îngreuna sau chiar speria potențiali utilizatori, mai ales dacă vârsta lor este una înaintată sau nu au avut tangență până acum cu tehnologia. Totodată dat fiind faptul că este dedicată exclusiv unei ligi bine consacrate din America, există o lipsă totală de personalizare care să îți permită să îți lași amprenta în această aplicație atât ca fan cât și ca jucător sau antrenor. Este limitată la uz strict dedicat ligii „NBA” deci nu poate primi echipe sau ligi internaționale lucru care împiedică ligile mici sau locale să facă vreo mișcare pentru a evolua. Exemplu grafic Figura B1 din Anexe.

3.1.2 AMBB

Cea mai populară platformă web dedicată baschetului din România, platforma oficială prezentă aici [5], și principala sursă de inspirație în sistemul pe care l-am implementat eu. „AMBB” sau Asociația Municipală Baschet București reprezintă principala ligă de baschet din București și o filială importantă a Federației Române de baschet și se ocupă cu tot ce înseamnă organizare de meciuri atât între echipe profesionale cât echipe de amatori sau meciuri de tip streetball.

Principalul atu al acestei platforme este faptul că este foarte ușor accesibilă masei de utilizatori fiind simplă și intuitivă pentru consumatorul de rând. Oferă diferite funcționalități cum ar fi cele deja prezente și în alte ligi statistici, ierarhii, program

meciuri dar și legătura strânsă cu rețelele de socializare. Totodată se remarcă prin faptul că este o platforma polivalenta care nu susține doar liga în sine cât și baschetul în general, oferind o secțiune de „NEWS” pentru a afla cele mai recente discuții și rezultate din lumea sportivă, interacțiunea cu fanii fiind de asemenea un prim plan al platformei.

Se prezintă și o parte de istorie a ligii pentru a familiariza consumatorul cu ceea ce înseamnă „AMBB”, cum s-a creat și ce a făcut ca să ajungă una din cele mai mari ligi de baschet din țară. Se prezintă totodată pe lângă liga în sine și informații despre celelalte ligi din țară, profitând de legătura strânsă pe care o are cu Federația Română de baschet și cu baza lor de date. Un dezavantaj al aplicației este lipsa posibilității de a efectua transferuri și a crea echipe cu componentă parțială. Pentru ca o echipă să fie primită în această ligă, este nevoie de pași pre organizatorici care sunt făcuți în afară platformei iar echipa se înregistrează abia după ce are un lot complet, direct pe platformă. Acest lucru nu este o problemă pentru echipele deja consacrate, care au o componentă completă însă devine o problemă majoră atunci când ești un împătimit al sportului care nu are însă 11 colegi de echipă ca să poată înscrie echipa în ligă și este nevoit să renunțe sau să caute persoane pentru a completa lotul, lucru care se întâmplă des dat fiind faptul că platforma este dedicată atât profesioniștilor cât și amatorilor. Exemplu grafic disponibil Figura B2 din Anexe.

3.1.3 ESPN

Tot o platformă populară americană , platforma oficială prezentă aici [6], ”ESPN” este un etalon în lumea fanaticilor ai baschetului. Este un lider internațional în materie de analiza sportiva și destinația tuturor noutăților legate de orice sport și conținutului video de înalta calitate, disponibil la orice ora indiferent de poziția geografică. Este frecventat de milioane de oameni global, oferind o gamă largă de funcționalități și servicii.

Are o structură intuitivă, care îți oferă majoritatea necesităților direct pe pagina de „HomePage”.Principalele elemente sunt: „Top Stories” care reprezintă articole de actualitate și noutățile zilnice, asemănător cu vechile ziare, în care se discută cele mai importante și relevante noutăți din lumea sportului precum transferuri, posibile contracte ale jucătorilor și antrenorilor cât și actualizări a listei de accidentări și a lotului componente pentru fiecare echipă, scoruri și rezultate finale a celor mai importante meciuri din toate ligile naționale și internaționale cât și scurtmetraje cu cele mai importante faze de joc.

Ceea ce pune în prim plan această platformă în comparație cu cele deja existente pe piața este „Fantasy Sports”, o componentă dedicată exclusiv fanilor, în care aceștia pot să își creeze propriile echipe cu jucători actuali sau pensionați pe care îi cumpără cu bani virtuali pe baza rezultatelor echipei lor imaginare într-un turneu

exclusiv online sau a transferurilor propriilor jucători din lot care își modifica valoarea în funcție de performanțele lor din lumea reală. Se mai pot simula meciuri, turnee, statistici, practic devenind proprii lor manageri de echipă, îmbunătățind semnificativ experiența de utilizator. Exemplu grafic disponibil Figura B3 din Anexe.

4 SOLUTIA PROPUSA

4.1 Arhitectura si infrastructura aplicației

Platforma Basketball League este bazată pe o arhitectură de tip 3-tier, exemplificată aici [7], care împarte aplicația în trei mari categorii importante: nivelul de prezentare sau interfața utilizatorului, singurul cu care utilizatorul interacționează în mod direct prin diferitele componente prezente, urmată de nivelul de aplicație care se ocupă de logica din spatele platformei și manipularea datelor prin diferitele tipuri de filtrări, sortări și redirectionări ale informațiilor care sunt primite din ultimul nivel, nivelul de date, în care se salvează toate variabilele definite pentru a crea în sine platforma.

Acest framework este capabil să ruleze fiecare nivel pe un sistem de operare diferit împărțind funcționalitățile logic. Acest lucru vine la pachet cu un număr ridicat de beneficii pe care le-am luat în considerare precum: dezvoltarea mai rapidă datorită faptului că fiecare nivel se poate dezvolta independent de celelalte astfel crescând scalabilitatea în timp ceea ce reduce riscul de impact generat de apariția problemelor la un alt nivel, neafectând performanța celorlalte componente.

Totodată din cauza faptului că nivelul de prezentare și cel de date nu comunică direct, se evită posibile slăbiciuni de securitate precum SQL injection sau alte tipuri de exploatare, și se păstrează integritatea datelor prin numeroase verificări.

Nivelul de prezentare sau mai ușor interfața utilizatorului este, așa cum am menționat anterior, singurul nivel pe care utilizatorul îl vede și cu care are o interacțiune directă prin diferitele funcționalități pe care acesta le prezintă. Acest tip de nivel rulează pe un browser web prin intermediul unui server și este împărțit într-o ramură funcțională care definește modul în care utilizatorul poate naviga și accesa diferitele funcționalități, și una vizuală a designului și a modului în care datele sunt afișate în cadrul interfeței.

Nivelul de backend, cunoscut și ca nivelul de mijloc, reprezintă motorul principal al platformei, unde toate datele sunt procesate, verificate și ulterior modificate pentru a putea construi logica aplicației. Neavând o aplicație cu un număr ridicat de date care trebuie procesate, am putut beneficia de toate atuurile acestuia, depanarea și testarea fiind foarte ușor de realizat având totul organizat într-un singur proces. Acesta preia informațiile oferite de baza de date, le prelucrează prin diferite filtre ,actualizări sau ștergeri, pentru a putea fi ulterior transmise către interfața cu care interacționează

utilizatorul. Am ales să merg pe o structură clasică de tip SpringBoot care se împarte în foldere dedicate cu rol clar.

Entity reprezintă cumulusul de obiecte de tip entitate care definesc cum va arată baza de date și ce variabile va avea fiecare tip de entitate.

Repository reprezintă folderul în care se definesc toate interfețele necesare implementării logicii funcționale prezentă în folderul Service care conține majoritatea logicii ulterior utilizată în Controller, care se ocupă de comunicarea dintre API urile din backend și interfață prin diferite endpointuri.

Pe partea de securitate, alegerea a fost, conform [8], JSON Web Tokens care asigură că datele fiecărui utilizator sunt securizate, fiind cel mai important aspect al credibilității aplicației. Folosind acest tip de autentificare parolele nu sunt salvate explicit în baza de date fiind mai degrabă folosite pentru a genera un token care este verificat înainte de orice apel în platformă, astfel fiecare sesiunea să se desfășoare în siguranță. Acest token este utilizat ca și prefix pentru orice utilizare a APIurilor.

Legătura între nivelul de afișare și cel logic se face prin cereri HTTP, și a API urilor pe baza conceptelor RESTful [9]. Un API REST este o interfață care permite interacțiunea între serviciile web printr-un set de protocoale, sau mai ușor, un contract între cel care produce informația și cel care o folosește în cazul nostru frontendul și backendul. Fiecare cerere este independentă de celelalte făcându-se prin diferite URL-uri sau căi cu diferite tipuri: GET, POST, PUT, DELETE în format JSON.

4.2 Baza de date

Pentru structurarea datelor, Figura A6 din Anexe, am decis să folosesc baze de date relaționale prin intermediul MySQL, păstrând datele sub formă de tabele împărțite în următoarele entități: User, Game, Game_Event, Invitation, Player, Team, fiecare având coloane de tip string, date calendaristice, integer și boolean.

Tabela User reține datele utilizatorilor platformei. Fiecare utilizator este identificat atât prin cheia primară de tip identificator unic care se generează automat la crearea unui utilizator nou cât și prin emailul acestora care poate fi dat ca parametru diferitelor API uri pentru a ușura logica de căutare și filtrare a acestora. Se mai găsesc și diferite date precum nume, prenume și rol, care împart utilizatorii în trei categorii jucător normal, căpitan și administrator cât și șirul de caractere necesar încărcării pozelor de profil pentru conturile acestora. Emailul fiind o resursă unică pentru fiecare entitate este folosit ca nume de utilizator în procesul de autentificare alături de parolă, care este salvată în baza de date criptată.

Pentru a ușura prelucrarea datelor din intermediul aplicației, am salvat în această tabelă și o entitate de tip jucător aferentă fiecărui utilizator, care se generează și ea

automat atunci când se creează un cont nou, aceste două tabele împărțind date între ele. Pentru stocarea pozelor am luat în considerare cea mai eficientă metodă care să se muleze pe logica aplicației mele cea de codificare base64 a pozelor în șiruri de caractere. Astfel aplicația poate rula pe orice sistem deoarece pozele sunt salvate direct în baza de date nefiind nevoie de spațiu extern.

Tabela Player reține datele aferente fiecărui jucător din liga sportivă. La fel ca la User, jucătorii sunt identificați prin codul de identificare unic. Această tabelă este cea mai complexă și conține o varietate de atribute care modelează informațiile esențiale despre un sportiv, cele mai importante fiind variabilele de tipul points, rebounds, assists și games_played care sunt ulterior folosite pentru a calcula toate statisticile necesare.

Tabela Player mai conține și o relație de tip many-to-one cu entitatea Team pentru a indica echipa din care face parte fiecare jucător dar și pentru a putea atribui un număr extins de jucători la o singură echipă. Tot pentru a putea respecta relația de mai devreme dintre utilizator și jucător, tabela mai descrie și o relație de tip one-to-one cu entitatea User pentru a ușura atribuirea de nume, prenume și alte date din tabela User în tabela Player. Am folosit diferite adnotări Lombok pentru a ajuta cu serializarea din JSON și a asigura o mapare corectă a datelor în același timp evitând referințele circulare.

Tabela Game conține datele relevante care fac posibilă crearea unui meci nou între două echipe. Coloanele clasice precum scorurile echipelor, locația și data sunt salvate aici alături de statusul fiecărui meci pentru a putea ști dacă un meci este programat, în desfășurare sau terminat. Totodată regăsim și aici doua relații de tip many-to-one între meciuri și echipele aferente fiecărui meci pe baza id-ului fiecărei echipe din ligă. Această entitate încapsulează comportamentul cheie al platformei asigurând o eficientizare a manipulării datelor implicate.

Alături de tabela Game există și tabela GameEvent care este relaționată cu Game pe baza unei legături many-to-one fiecare meci având o multitudine de evenimente care se petrec în timpul simulării acestuia. Fiecare eveniment dintr-un meci este mapat cu tipul de eveniment, descrierea evenimentului dar și un timestamp care indică momentul de timp când evenimentul a avut loc.

Tabela Invitation este cea care modelează datele și comportamentul aferent invitațiilor trimise de căpitani către alți jucători pentru a îi putea recruta în echipele lor. Datele din tabelă sunt împărțite astfel: două relații de tipul many-to-one între invitații și echipele care invită jucătorul și jucătorul respectiv care este invitat. Totodată se indică și statutul fiecărei invitații Pending, Accepted, Declined pentru a putea fi ulterior folosite în logica aplicației. Fiecare variabilă și relație este bine definită pentru o modularitate mai clară și pentru a evita problemele de separare.

Tabela GameEvents este cea care salvează toate acțiunile de joc din timpul unei simulări. Aceasta cuprinde o relație de tipul many-to-one cu tabela Game pentru a putea referi mai multe faze de joc la același meci, o variabilă pentru tipul de acțiune care se petrece în timpul unui meci precum: coș marcat, coș ratat, pasă decisivă sau recuperare cât și o variabilă de tip data pentru a putea reține exact momentul la care se petrece fiecare acțiune de joc.

În toate entitățile m-am folosit de adnotările oferite de Lombok pentru a face codul mai ușor de înțeles și vizualizat, lucru care m-a ajutat și la depanarea acestuia plus serializarea JSON pentru maparea datelor în clase și obiecte.

4.3 Fluxul aplicației

Un utilizator care utilizează pentru prima oară platformă este întâmpinat de pagină de „Autentificare/înregistrare” unde este întrebat dacă are deja un cont activ sau este nevoit să își creeze unul nou. Dat fiind faptul că utilizatorul nostru este unul nou acesta va selecta pagina de Înregistrare unde îi este prezentat un formular tip în care trebuie să completeze diferite date personale pentru a putea fi înregistrat. Este nevoit să își atribuie un email și o parolă unică, înălțimea pentru a putea fi recrutat mai ușor de alți căpitani dacă este un jucător normal, o imagine de profil și rolul pe care dorește să îl aibă. În urma înregistrării cu succes este întors pe pagina de Autentificare unde este rugat să completeze emailul și parolă cu care s-a înregistrat. După ce s-a autentificat pe platforma acesta poate vedea toate componentele ligii.

Structura platformei este bazată pe o bară de căutare care include toate paginile separate prin care utilizatorul poate naviga fără vreo restricție. Prima pagină „Listă Echipe” este cea de afișaj al echipelor în care se prezintă o listă cu toate echipele din liga sportivă, în care sunt afișate statistici exacte legate de fiecare cum ar fi puncte totale, pase decisive totale, recuperări totale, numele echipei și numărul de meciuri jucate până acum în ligă. Pe lângă detaliile menționate anterior fiecare linie din tabel reprezintă o rută care odată apăsată redirecționează utilizatorul pe o altă pagină în care este prezentat lotul component și detalii succinte legate de fiecare jucător din acel lot. Pagina dispune și de numeroase filtre care sortează echipele în funcție de ierarhia punctajelor, recuperărilor dar și a meciurilor câștigate sau pierdute atât crescător cât și descrescător, cât și posibilitatea de a căuta după nume fiecare echipă în parte.

Bazat pe aceeași structură este și pagina „Listă Jucători”, care vine la pachet cu un număr de filtre specifice acestei entități dar și o tabelă mai vastă care afișează numele echipei aferente fiecărui jucător cât și înălțimea acestuia. În cazul în care utilizatorul este un căpitan în căutare de jucători noi pentru propria echipă acesta poate filtra sportivii în funcție de statutul acestora, dacă sunt liberi de contract ,adică nu aparțin vreunei echipe sau dacă sunt deja contractați. La fel ca în cazul listei de echipe, fiecare linie reprezintă o rută către o pagină separată care descrie în detaliu fiecare jucător și

în care se pot afla detalii personale precum nume complet și adresa de email în cazul în care un căpitan dorește să recruteze jucătorul respectiv.

Următoarea pagina este cea de „Creare Echipa/Ștergere” echipă în care un jucător autentificat ca și căpitan își poate crea o nouă echipă completând numele și anul înființării echipei urmând a fi înregistrată în baza de date iar căpitanul este automat înrolat în cadrul acelei echipe. Un jucător autentificat ca jucător normal nu poate utiliza această funcționalitate deoarece nu dispune de drepturile necesare. Tot pe aceasta pagină există și funcționalitatea de desființare echipă care șterge echipa din baza de date și eliberează de contract toți jucătorii care erau parte din acea echipă, funcționalitate exclusivă tot căpitanului unei echipe. Atât ca jucător normal cât și ca și căpitan de echipă poți naviga către Pagina de Profil Utilizator, în care este prezentat orice detaliu atât personal cât și tehnic despre tine.

Pagina este împărțită în trei componente individuale. Prima este afișaj datelor personale alături de imaginea de profil. Alături în partea dreaptă a paginii este lista cu invitații pe care jucătorul le-a primit de la alte echipe cât și un formular de trimitere invitație în care căpitanul trebuie să completeze emailul jucătorului pe care vrea să-l recruteze și numele echipei care este autocompletat și nevalid schimbărilor pentru a nu putea invita un jucător într-o echipă din care tu la rândul tău nu faci parte. Tot aici se afla și o listă cu invitațiile trimise de căpitan către alți jucători și statul în care se afla acestea în acel moment. În partea de jos a profilului acesta își poate vizualiza statisticile procentuale din meciuri calculate pe baza numărului de meciuri pe care acesta l-a jucat până în momentul de față. Există și o scurtătură în bara de navigare a platformei, un clopoțel în care există un număr îngroșat de notificări aferent numărului de invitații neacceptate de către jucător care odată apăsă îl duce direct la pagina lui de profil.

Cea mai complexă pagină este cea de „Manageriere Meciuri” în care utilizatorul de tip administrator poate simula propriu-zis meciuri. Și aici există trei mari funcționalități care împart pagina. Prima parte este cea de simulare efectivă în care se completează numele celor două echipe, locația și data meciului. În urma creării unui meci acesta este pus în tabela de meciuri programate. Administratorul după se ocupă de simularea meciurilor prin selectarea meciului care trebuie simulat. Se afișează după cele două echipe în paralel alături de lista cu jucători din fiecare echipă și un scor live. Aici intervine websocketul care trimite datele către UI cu fiecare schimbare de scor care este afișată în timp real, actualizând atât scorul pentru fiecare echipă cât și fiecare statistică pentru jucătorii implicați în meciul respectiv. Simularea are un cronometru care face meciul să fie cât mai realist și pentru a da echipelor șansa să schimbe soarta meciului în orice secunda. La finalul simulării, meciul este mutat din tabela de meciuri programate în tabela de meciuri jucate în care se prezintă numele echipelor și scorurile finale.

Aceasta tabelă este accesibilă oricărui utilizator din platformă indiferent de rolul pe care îl are, și îi este oferită posibilitatea de a afla detaliile meciurilor în timp real deși nu a putut vedea simularea propriu-zisă a meciului. El trebuie să apese pe unul din meciuri și este redirecționat către o altă pagină de Detalii meci în care este salvată întreaga simulare de meci cu fiecare schimbare de scor și de statistici pentru fiecare jucător. Aici utilizator, indiferent de rolul pe care îl are în aplicație, este redirecționat pe o pagină tip în care se prezintă terenul de baschet, scorurile echipelor și fazele de meci care au actualizat la fiecare secundă acel scor

5 DETALII DE IMPLEMENTARE

Acest capitol surprinde detaliat dezvoltarea punctuală a fiecărei componente din punct de vedere al tehnologiilor folosite și a codului propriu-zis. Platforma ligii de baschet este împărțită în cele două mari module. Modulul de backend care se ocupă cu logica și modelarea datelor, transmise ulterior prin API-uri prin diferite endpointuri. Modulul de frontend se ocupă cu preluarea datelor, maparea lor în diferite constante, verificări suplimentare și afișarea ulterioară a datelor într-un format modern prin JavaScript și TailWind și CSS.

5.1 Tehnologii

În dezvoltarea platformei mele am ales să utilizez următoarele tehnologii: SpringBoot pentru partea de backend, React pentru frontend combinat cu CSS pentru styling și TailWind pentru modelarea componentelor interioare din fiecare pagină. Pentru baza de date am mers pe una relațională deoarece îmi oferă o mai bună organizare a datelor în tabele, iar pentru partea de simulări meciuri am utilizat Websockets care oferă o comunicare în timp real. Pentru mediul de dezvoltare am folosit atât IntelliJ pentru partea de logică a aplicației, oferind suport în SpringBoot și MySQL iar pentru frontend am folosit Visual Studio Code.

SpringBoot, conform documentației oficiale [10], este un framework folosit pentru a crea aplicații backend complexe cu numeroase avantaje. În primul rând, este integrat într-un ecosistem larg cunoscut pentru simplitate și viteză de dezvoltare. Oferă totodată o documentație vastă și ușor de interpretat alături de numeroase forumuri. Datorită popularității acesta ușurează depanarea problemelor tehnice apărute pe parcurs. SpringBoot mai suportă și micro serviciile arhitecturii care permit o mentenanță independentă a componentelor utilizate.

Nu în ultimul rând, este abil în îmbinarea lui cu servere embedded precum Tomcat, Jetty și Undertow care reduc nevoie de servere externe. Java este un limbaj de programare orientat obiect care rulează pe milioane de platforme de la computere la telefoane și console de jocuri. Motivul principal pentru care Java a devenit un limbaj atât de popular este

portabilitatea care permite unui program scris pentru un computer să fie adaptat pentru un telefon fără prea mult efort.

Totodată motivul pentru care am ales Java ca principal limbaj de programare este versatilitatea și modelul de arhitectură bazat pe obiecte care simulează foarte mult viață reală, oferindu-mi o perspectivă familiară când vine vorba de legătura dintre clase. Totodată Java oferă un suport vast pentru lucrul în paralel, și permite platformelor web să suporte numeroase procese în același timp. Nu în ultimul rând, acest limbaj de programare prin bibliotecile lui, este util în dezvoltarea de APluri pentru criptografie și comunicare securizată.

Websockets ,conform [11], reprezintă un protocol de comunicare bidirecțională și full-duplex, dependent de o singură conexiune de tip TCP. Sunt folosiți în special pentru comunicarea în timp real dintr-o aplicație deoarece oferă o conexiune live de lungă durată. Este o tehnologie esențială pentru funcționalități care necesită schimbări în timp real cum ar fi schimburi de mesaje sau scoruri live. Aceștia permit schimbul instant de date dintre client și server, astfel reducând overheadul asociat cu pollingul Http. Este menținută constant o conexiune deschisă care suportă scalabilitatea unui număr mare de conexiuni simultane, fiind astfel foarte rapid în răspunsuri.

React, conform [12], este o librărie puternic influențată de JavaScript care este menită dezvoltării de aplicații frontend și interfețe pentru utilizatori. Este menită în primul rând să ofere o interfață dinamică și interactivă datorită arhitecturii bazată pe componente. Acest tip de arhitectură încurajează reutilizarea unor componente, oferind astfel un proiect mai curat și ușor de înțeles indiferent de dimensiunea acestuia. Totodată dispune de un DOM virtual, o interfață de programare pentru HTML și XML care structurează ierarhic aplicația, asigurând astfel o îmbunătățire a performanței, îmbunătățind doar părțile necesare din DOM păstrând astfel o performanță ridicată. Componentele sunt menite să structureze codul în bucăți simple și clare, care se pot ulterior folosi în toată aplicația, iar modificările aduse unei componente nu afectează tot proiectul, mai degrabă doar cel asociat acelei componente.

CSS,conform [13], sau Cascading Style Sheets, este limbajul folosit pentru stilizarea interfeței utilizatorului. Acesta oferă o interfață prietenoasă și ușor de utilizat, fișierele .css venind ca un ataș pentru componentele React de pe platformă, segmentând diferitele butoane și tabele într-un mod plăcut și modern. Acest limbaj pune la dispoziție și un număr de frameworkuri pe care dezvoltatorii le pot utiliza pentru a implementa funcționalități mai complexe. Ele sunt compatibile cu multiple motoare de căutare lucru care reduce apariția problemelor tehnice și totodată acestea nu necesită o muncă în plus deoarece vin ca un schelet tip, nefiind nevoie să te adâncești prea mult în codul de CSS pentru a rezolva diferite taskuri. Eu am ales să folosesc Tailwind deoarece permite utilizatorului să înfrumusețeze interfața fără limitări, fiind foarte bine documentat.

MySQL, conform [14], a fost alegerea mea pentru configurare și crearea unei baze de date aferente platformei mele. MySQL este cea mai populară bază de date de tip open-source. Fiind folosită de majoritatea marilor companii precum Facebook, Twitter și Netflix. Deoarece este open-source aceasta include numeroase funcționalități dezvoltate de milioane de utilizatori pe durata a mai bine de 25 de ani.

Bazele de date se împart în două mari categorii relaționale și non-relaționale. Cele relaționale așa cum este cea aleasă de mine împart datele în tabele, în pofida păstrării datelor într-un singur loc. Organizarea acestora este foarte clară, optimizată și oferă un mediu foarte flexibil datorită obiectelor organizate precum tabele cu coloane și rânduri, fiind urmărite de numeroase reguli între diferitele tabele pentru a păstra baza de date organizată și a nu avea niciodată duplicate sau date lăsate singure. Totodată este o bază de date de tip client-server care constă în nenumărate servere SQL care lucrează în paralel pentru a suporta diferite librării și API uri. Este o bază de date rapidă și scalabila care e foarte ușor de setat și utilizat.

JavaScript, conform [15], este un limbaj de programare foarte polivalent folosit în special pentru dezvoltarea de platforme web și aplicații dinamice și interactive. Popularitatea lui a dus la apariția multitudinilor de librării și frameworkuri care să eficientizeze scrierea de cod prin bucăți reutilizabile deja implementate, totodată ușurând configurările. Funcționalitățile cu care iese în evidență pe partea de front-end sunt animații, bara de căutare, butoane și multe alte elemente interactive menite să atragă utilizatorii paginilor respective. Totodată oferă și actualizări în timp real a paginii fără a fi nevoie să o reîncarci.

Postman, conform [16], este o platformă care are ca scop principal construirea și depozitarea de API-uri. Aceasta pune la dispoziție o suită de colecții în care poți salva diferite tipuri de cereri într-o unitate centrală pe care o poți ulterior partaja cu alte persoane. Oferă suport pentru mai multe tipuri de API-uri cât și metrice, documentație și teste manuale. Totodată permite integrarea de pipeline-uri.

5.2 Backend

Pentru arhitectura backendului am ales structura clasică împărțită în următoarele module, conform [17]: Entity, Repository, Service, DTO, Controller și Configuration (Figura 1). Această separare clară între modulele sistemului asigură o izolare a posibilelor probleme de depanare cât și un cod mai lizibil și mai ușor de înțeles. Fiecare componentă are propriul rol în sistem și reprezintă diferitele nivele ale aplicației. Entitățile reprezintă nucleul datelor din aplicație. Acestea sunt responsabile prin adnotările Lombok, să creeze tabele necesare în baza de date (@Entity), indicând

persistenta acestora. Fiecare entitate corespunde unui tabel iar fiecare variabilă din entitate corespunde unei coloane din acel tabel.

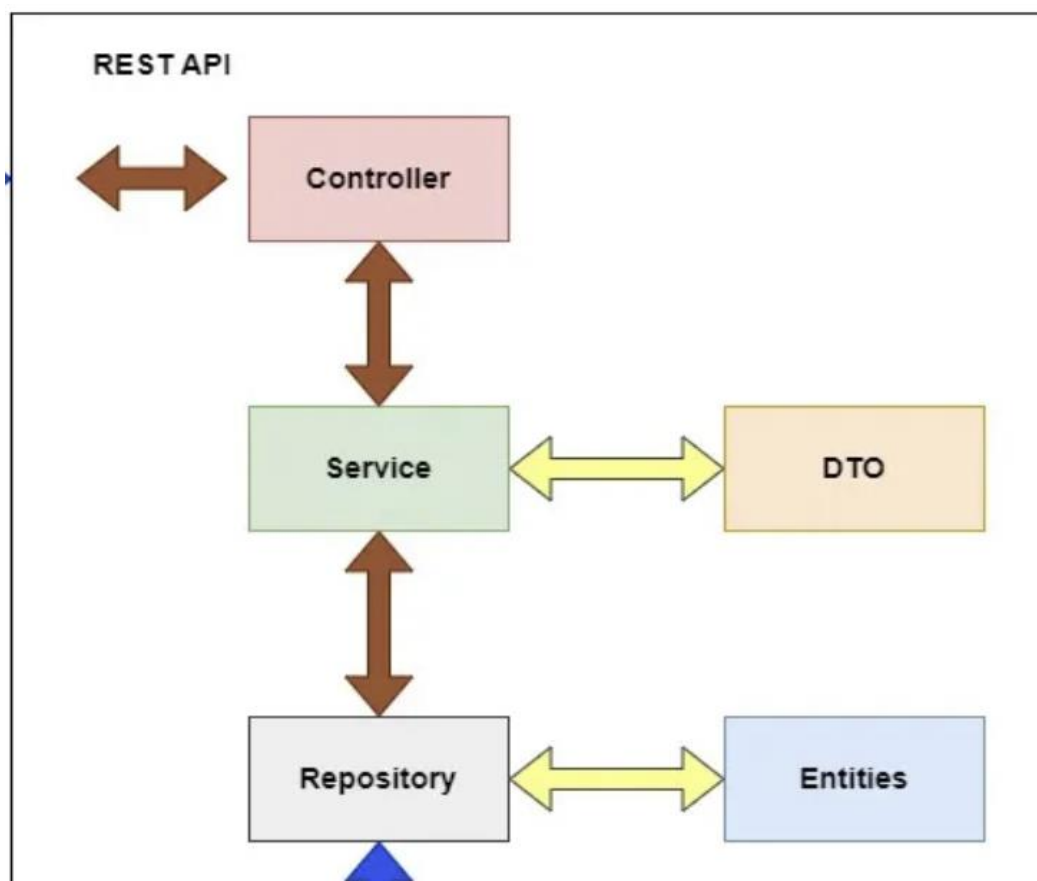


Figura 1. Exemplu arhitectura platforma SpringBoot

Eu mi-am împărțit entitățile în 3 componente majore "Team" care definește datele necesare creării și manipulării unei echipe, "Player" care definește tot date necesare despre jucătorii fiecărei echipe și "Game" care folosește cele două entități anterioare pentru a crea un meci de baschet. Pe lângă acestea există și entități pentru descrierea utilizatorilor "Users", pentru evenimentele dintr-un meci "GameEvents" și pentru crearea invitațiilor în alte echipe clasa "Invitation".

În modulul Repository am definit interfețele necesare fiecărei componente exemplificată anterior prin operațiile de tip CRUD(Create,Read,Update,Delete) care se aplică direct pe entități. Ele sunt implementate automat de către Spring Data JPA astfel reducând necesitatea de cod inutil. Totodată pentru a eficientiza extragerea de date din baza de date, și nu a aduce date inutile care ar încetini procesul de exploatare a acestora am folosit și EntityGraphuri pentru a face fetch doar la asocierile de care

aveam nevoie în mod punctual, cum am procedat în cazul găsirii meciului cu status încheiat, folosind ca attribute cele două echipe implicate.

În modulul de Service am definit logica efectivă a platformei. Acest nivel este împărțit în două pachete cel de interfețe și cel de implementare. Pachetul de interfețe conține planul de structură al fiecărei clase. În acesta se definesc toate metodele care ulterior sunt implementate efectiv în pachetul ServiceImpl. Aici mi-am prelucrat toate datele primite din baza de date, am efectuat filtrări, sortări și alte verificări pentru diferite cazuri speciale din logica aplicației, precum restricționarea invitațiilor către echipe din care căpitanul nu face parte și imposibilitatea de a crea două meciuri în același timp și în aceeași locație.

Cea mai complexă metodă din acest model este cea a simulării unui meci, Figura C1 din Anexe , încât ea cuprinde numeroase alte componente din interiorul aplicației. Logica acestei funcționalități este de a crea un meci între două echipe deja existente, ulterior ele fiind supuse unui timer de 10 secunde. Inițial statisticile pentru fiecare jucător din meci se setează la zero pentru a putea calcula procentajele corecte. Se alege aleator o echipă care va înscrie, iar din acea echipă se alege un jucător pe baza unui algoritm, Figura C2 din Anexe, care selectează din lista de jucători unul pe baza tuturor statisticilor aferente acestuia, înălțime, coșuri marcate, rata de a marca. Pentru a nu avea un singur jucător care înscrie mereu, această pondere calculată este doar orientativă și este folosită pentru a crește șansa de a fi ales dar alegerea rămâne tot aleatorie, sistemul fiind asemănător unei cote. Fazele meciului sunt împărțite și ele aleator astfel încât să simuleze cât mai bine un meci real în care o echipă poate ataca de mai multe ori la rând, împărțirea fazelor de joc nefiind una egală. Se calculează astfel câte coșuri dă fiecare jucător dintr-un pool de aruncări care pot fi atât marcate cât și ratate. Aruncările sunt împărțite în 3 tipuri de un punct, două sau trei și acestea sunt ulterior manipulate pentru a actualiza scorul fiecărei echipe în timp real folosind pachete de date transmise printr-un canal de comunicații de tipul WebSocket.

Pe lângă aruncările la coș fiecare jucător poate impacta meciul și prin alte acțiuni de tipul „pase decisive, recuperări sau schimbări de posesie de la adversar. La finalul cronometrului, fiecare jucător este interogată pentru a putea calcula statisticile exacte în funcție de numărul de meciuri jucate și apoi sunt actualizate aceste statistici în tabelele corespunzătoare din baza de date. „Data Transfer Object” (DTO) este modulul folosit pentru transferul de date între celelalte module. Acesta convertește un răspuns din baza de date într-o variantă mai concretă care mapează doar variabilele care sunt folosite în logica respectivă.

Totodată am folosit clase DTO pentru a agrega date din diferite entități într-o singură unitate centrală care m-a ajutat să transform datele înainte de a le trimite către client, datorită faptului că majoritatea entităților mele conțin sau sunt dependente de celelalte

tabele răspunsurile primite erau foarte complexe și conțineau informații inutile în logica funcțională respectivă.

Modulul de controller este cel care se ocupa efectiv de transmiterea și primirea de cerințe de la client. El comunică direct cu interfața utilizatorului prin cereri HTTP care sunt ulterior mapate prin „endpoints”, adrese ale paginii web. Acești cereri au fost împărțite în câteva categorii principale: `@PostMapping` care se ocupă cu trimiterea de date către client, `@GetMapping` care cere date de la server și `@DeleteMapping` care trimite o cerere de ștergere a unei resurse. Acest nivel nu conține logică propriu-zisă folosind metodele definite anterior în nivelul de Service, astfel păstrând o lizibilitate a codului și o modelare și depanare mai ușoară ulterior, singurele operații suplimentare prezente aici fiind validări și verificări înainte sau după primirea/trimiterea datelor.

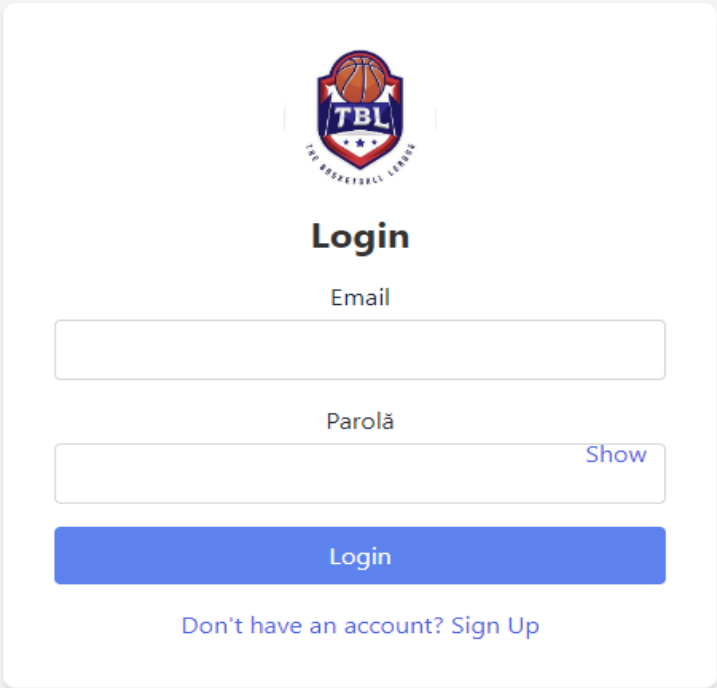
Nu în ultimul rând, modulul Configuration este cel care se ocupă de definirea și configurarea întregului proiect prin Beanuri, obiecte care sunt definite de adnotații, și ale cărui ciclu de viață este manageriat automat de SpringBoot pentru a fi ulterior refolosite și a injectate în alte clase. Tot aici am definit și configurația pentru SpringSecurity, componenta care se ocupă de logica de înregistrare și autentificare pe platformă. Eu am ales să folosesc un JWT Token care este un obiect JSON encodat folosit ca payload în requesturile făcute către server. Astfel am definit câteva reguli de ierarhie pentru diferite API-uri cum ar fi "CreateTeam" care poate fi folosit doar de către un utilizator autentificat ca și "CAPTAIN", cât și "SimulateGame", "CreateGame" și "DeleteTeam" care sunt accesibile doar utilizatorilor de tip "ADMIN". Acest tip de aplicație m-a ajutat să mențin o tehnică bine structurată de dezvoltare și testare, reușind să separ fiecare funcționalitate în componente individuale.

5.3 Frontend

Pentru interfața utilizatorului am folosit React și Tailwind CSS. Configurația proiectului este una clasică pentru orice proiect React bazată în special pe fișierul de intrare `index.js` (fișierul unde se realizează rularea), iar dependențele se afla în `package.json`. M-am folosit de Tailwind și CSS pentru a formata și înfrumuseța paginile cu care interacționează utilizatorul. Fișierul `index.js` conține setările necesare ca și rădăcina a DOM-ului [18], prin componenta `App` care reprezintă nucleul aplicație, aici plasându-se toate funcțiile care creează legături cu toate celelalte componente din compoziția proiectului și modul de navigare printr-un `AppBar`. Rutele aplicației sunt definite prin biblioteca `react-router-dom` care instanțiază rute de tip normal și private pentru paginile care pot fi accesate doar după autentificarea pe platformă de către utilizator. Proiectul este împărțit în fișiere pe baza componentelor și a funcționalităților prezente în acestea: `Game`, `Login/Register`, `Player` și `Team`.

Formular de Înregistrare și Autentificare ,Figura 2 si Figura 3, este unul minimalist în care sunt implementate doar funcționalitățile de bază pentru a păstra un afișaj ușor de înțeles și simplist cu un dropdown pentru selectarea tipului de utilizator și un client Axios

care se ocupă de trimiterea datelor de tip imagini. Intrat pe aplicație ,utilizatorul poate să navigheze între toate paginile print-un AppBar poziționat deasupra paginilor.



The image shows a login form for the TBL Basketball League. At the top is the league's logo, which features a basketball and the letters 'TBL' inside a shield-like shape, with 'TBL BASKETBALL LEAGUE' written around it. Below the logo is the word 'Login' in a bold, black font. Underneath 'Login' are two input fields: the first is labeled 'Email' and the second is labeled 'Parolă' (Password). To the right of the password field is a blue link labeled 'Show'. Below these fields is a blue button with the text 'Login'. At the bottom of the form is a blue link that says 'Don't have an account? Sign Up'.

Figura 2. Pagina de Autentificare

Register

Nume

Prenom

Email

Parolă

Height

Role

Select a role

Profile Picture

Choose File No file chosen

Înregistrare

[Already have an account? Sign In](#)

Figura 3. Pagina de Înregistrare

Paginile "Listă Echipa" și "Listă jucători", Figura 4 și Figura 5, sunt structurate într-un mod similar folosind un tabel în care sunt afișate toate statisticile legate de fiecare jucător și fiecare echipă, însoțite de casete de selecție care permit aplicarea diferitelor filtre și sortări pentru a găsi mai eficient jucătorii sau echipele favorite. Fiecare jucător se poate filtra în funcție de numărul de puncte, recuperări, pase decisive și aparența la o echipă. Tabela de echipe conține aceleași filtre doar că în loc de apartenența la o echipă are număr de meciuri câștigate și pierdute. Totodată există în ambele pagini o bară de căutare pentru o selecție specifică în funcție de nume. Ambele tabele conțin rute de acces către pagini de afișaj detaliat al componentelor din acestea în funcție de echipă sau jucătorul selectat din tabel.

7/4/2024, 3:00:00 AM
SSS vs PPP

7/17/2024, 3:00:00 AM
SSS vs PPP

7/23/2024, 3:00:00 AM
PPP vs SSS

Team List

Create Team

Player List

UserProfilePage

Manage Games

Logout

Player List

Sort by: Points

Order: Ascending

Search by First Name: First Name

Search by Last Name: Last Name

Show Players Without Team

Show Players With Team

Show All Players

Name	Height	Points per Game	Rebounds per Game	Steals per Game	Assists per Game	Team
admin admin	999	0.0	0.0	0.0	0.0	No Team
Surdu Cristian	201	2.0	1.0	1.5	0.5	SSS
player1 player1	187	2.0	1.0	1.0	0.5	SSS
player2 player2	157	3.0	1.0	0.5	0.0	SSS
player3 player3	173	3.0	1.0	0.5	0.5	PPP
player4 player4	189	4.0	0.5	0.5	1.0	PPP
Popica Alexandru	183	5.0	2.5	1.5	1.0	PPP

Figura 4. Pagina pentru listă de jucători

7/17/2024, 3:00:00 AM
SSS vs PPP

7/23/2024, 3:00:00 AM
PPP vs SSS

7/4/2024, 3:00:00 AM
SSS vs PPP

Team List

Create Team

Player List

UserProfilePage

Manage Games

Logout

Team List

Sort by: Points

Order: Ascending

Search: Search by team name

Team Name	Games Won	Games Lost	Total Points	Total Rebounds	Total Steals	Total Assists
Test	0	0	0.0	0.0	0.0	0.0
SSS	0	2	7.0	3.0	3.0	1.0
PPP	2	0	12.0	4.0	2.5	2.5

Figura 5. Pagina pentru listă echipe

Pagina de profil, Figura 6, prezintă un design modern în care se afișează detaliat toate datele aferente utilizatorului înregistrat pe platformă, printr-un format segmentat, partea de sus a paginii conține informații personale, rubrică de invitații care împarte invitațiile în doua categorii: invitații primite de la alte echipe pe care jucătorul el poate accepta sau refuza, și invitații trimise către alți jucători alături de un istoric al statusul acestor invitații. Mai exista și o poză de profil iar partea de jos un tabel cu statisticile avansate ale jucătorului si un buton prin care jucătorul se poate înlătura din componența echipei actuale în care joaca. Pagina de "Creare/Ștergere Echipa" este cea mai minimalistă din toată platforma încât conține doar un formular de completare a numelui și anului de înființare al echipei iar în partea de jos un buton pentru ștergerea unei echipe deja existente.

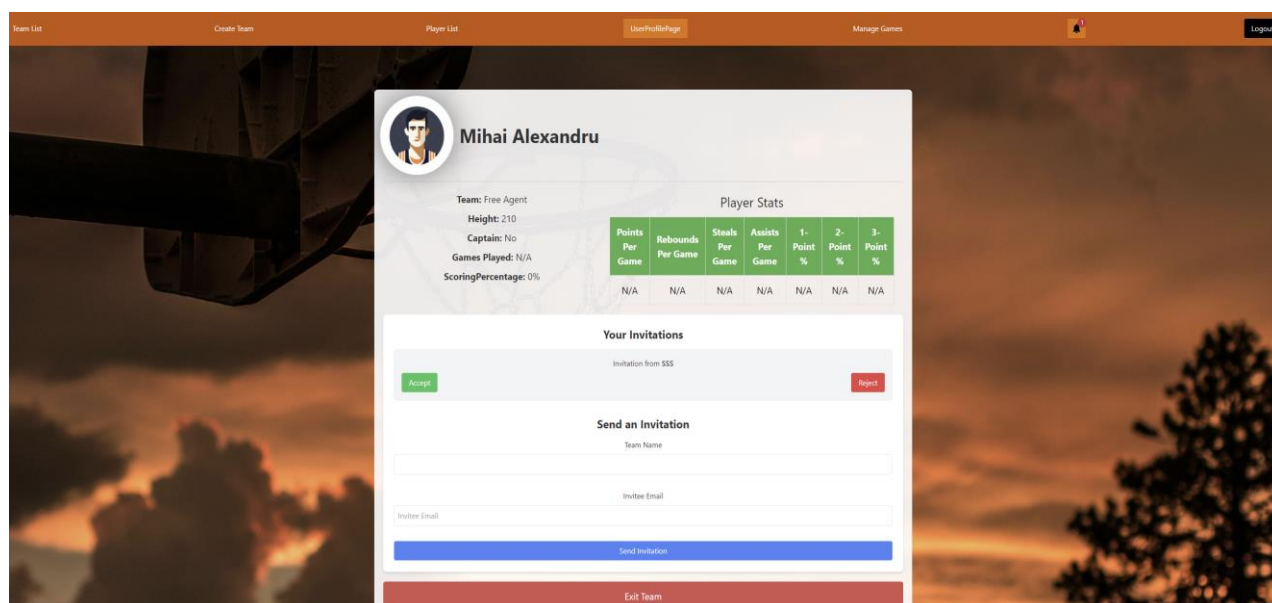


Figura 6. Pagina de profil utilizator

Pagina de "Manegeriere a Meciurilor", Figura 7, conține tot un format segmentat în care logica propriu-zisă a creării și simulării unui meci a fost plasată în partea superioară iar informațiile legate de meciuri în partea inferioară a paginii. Există și aici două tabele cu istoricul meciurilor deja jucate și cu cel al meciurilor programate rămânând la latitudinea utilizatorului autentificat ca și administrator să selecteze ordinea în care meciurile programate vor fi simulate. Se prezinta detalii esențiale legate de meciuri precum echipe participante, scoruri, locația în care se va juca meciul cat si data si ora la care acesta e programat. Prin intermediul componentei NavLink conform documentației oficiale [19], utilizatorii pot accesa expunerea detaliată a meciurilor jucate doar cu o apăsare de buton într-o interfață separată, unde se detaliază fiecare faza de meci ,alături de jucătorul care a creat faza respectiva si numărul de puncte pe care a reușit sa îl înscrie, prezent Figura 8.

Pentru un afișaj dinamic am ales să implementez și o bară dinamică care prezintă toate meciurile deja jucate într-o buclă infinită care se derulează deasupra NavBarului.

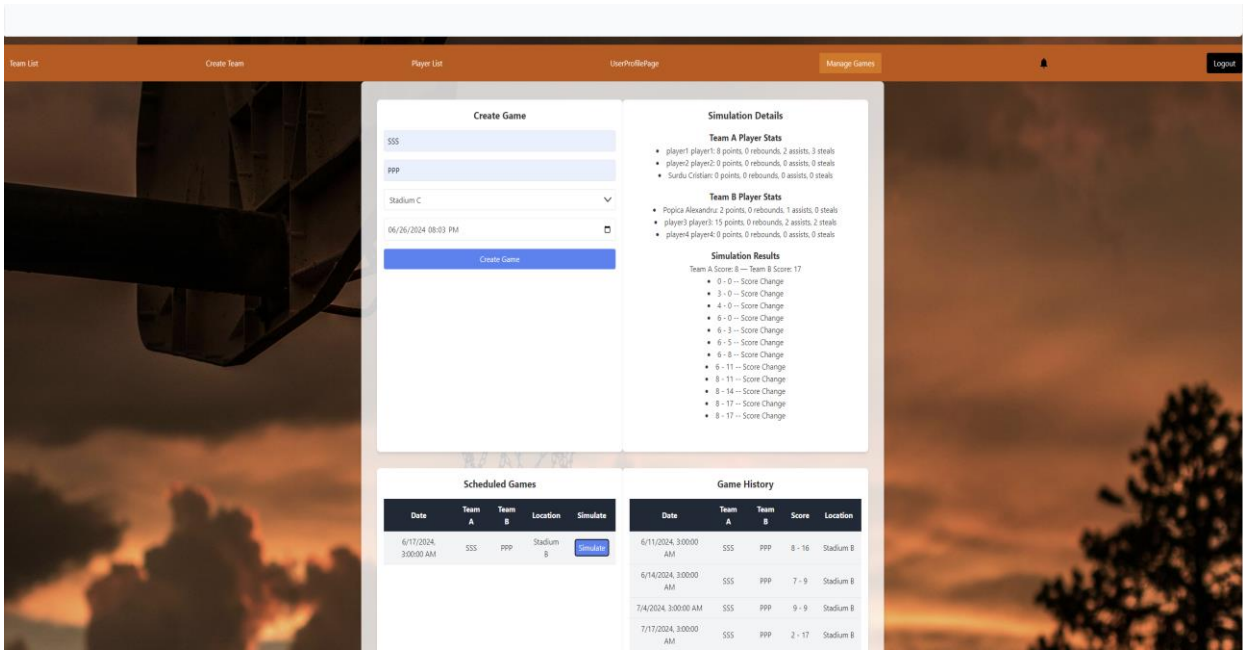


Figura 7. Pagina de manageriere a meciurilor

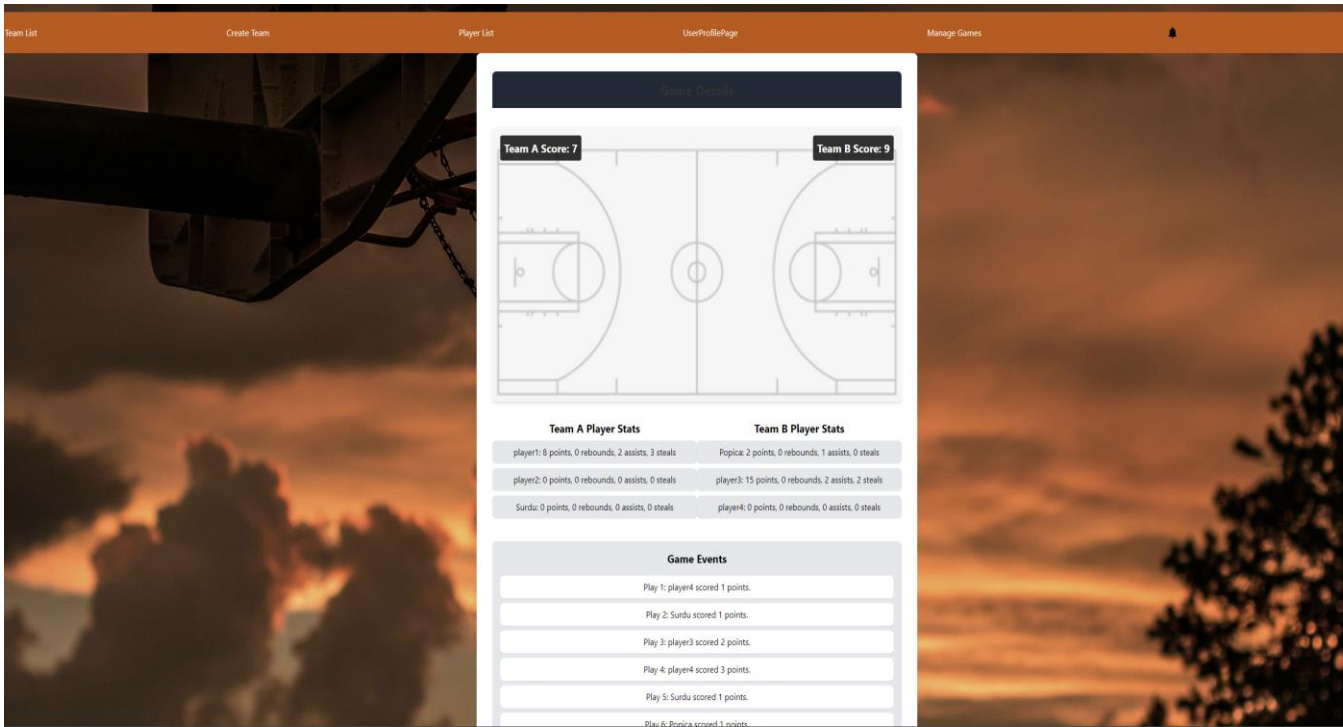


Figura 8. Pagina de detalii meci jucat

6 STUDIU DE CAZ / EVALUAREA REZULTATELOR

Pentru acest capitol am efectuat atât testare manuală a fiecărei funcționalități pentru a mă asigura că totul funcționează conform logicii explicate anterior cât și teste unitare pentru fiecare metodă în parte. Pentru partea de backend am realizat în primul rând testare manuală cu ajutorul platformei Postman, Figura C3 din Anexe, în care am creat câte o cerere pentru fiecare API din controllere alături de cereri cu diferite date atât valide cât și invalide pentru a vedea ce răspunsuri am de la server. Am realizat atât cereri bazate pe date invalide, metode greșite de cerere (Put în loc de Post) cât și utilizatori cu permisiuni necorespunzătoare pentru a verifica securitatea aplicației si am mapat fiecare tip de eroare 404,403,400 la cate un mesaj de validare a acelei erori pentru a putea înțelege mai ușor care era problema.

Pentru testarea pachetelor trimise prin websocket am folosit tot Postman și am creat o colecție nouă dedicată acestora prin care am monitorizat pachetele trimise. Pentru testarea interfeței utilizatorului, am navigat prin diferitele pagini și am încercat să creez situații nefavorabile pentru a testa manual platforma. Am probat navigarea de la o pagină la alta să fiu sigur că rutele sunt setate corespunzător, am încărcat date invalide de tipul șiruri de caractere prea lungi sau cu format invalid, tipuri de utilizatori invalizi cât și navigarea prin paginile separate să fiu sigur că tabele afișează toate datele aduse din backend și că derularea paginii vertical in sus si in jos păstrează formatul tabelor intact. Totodată atunci când am dat ca input date invalide sau am încercat acțiuni nepermise pe platformă m-am asigurat că primesc logurile corespunzătoare și intuitive în interfață. Probleme întâmpinate pe parcursul platformei au fost atât de natură logică cât și funcțională.

Prima problemă cu care m-am confruntat a fost cea legată de politicile CORS. CORS, conform [20], asigură securitatea platformei, prin eliminarea posibilității de atacuri cross-site scripting (XSS) prin specificarea exactă a permisiunilor de acces asupra domeniului. Depanarea a fost anevoioasă inițial până am înțeles exact cum funcționează aceste tipuri de erori. Am verificat consola din browser cât și ce tipuri de prefixe se trimiteau la server, iar nu în ultimul rând m-am asigurat că cererile erau făcute folosind http nu https. Pentru a rezolva problema a fost nevoie să creez o configurație CORS în care am setat originile permise cât și a metodelor HTTP împreună cu prefixele necesare și a cookiurilor de autentificare.

A doua problemă la care am fost blocat un timp mai îndelungat a fost cea legată de configurarea canalului de websocket care a fost strâns legată de configurațiile CORS despre care am povestit mai devreme. A fost și aici nevoie de configurare a securității prin includerea endpointurilor pentru WebSocket atât în clasa personalizată pentru el cât și în cea de configurare CORS. Cea mai utilă metodă de depanarea a fost Postman și

colecția dedicată pentru Websocket cu care vine m-a ajutat mult să înțeleg exact cum se trimiteau acele pachete de date și unde pierdeam mai exact date.

Ultima problemă cu care m-am confruntat a fost legat de permisiunile diferitelor tipuri de utilizatori, mai exact în fișierul de configurare deși specificam explicit în filtrul de sufixe ce API url uri se pot folosi în funcție de rolul utilizatorului pe platformă care e verificat prin tokenul JWT pățeam că unele API uri să funcționeze indiferent de rol așa că am adăugat verificări suplimentare în logica de frontend că să fiu sigur că funcționalitățile sunt împărțite corespunzător în funcție de rolul aferent.

7 CONCLUZII

Basketball League este o platformă web care are ca scop principal mulțumirea maselor de jucători atât amatori cât și profesioniști care își doresc să activeze într-o echipă indiferent de vârstă și nivel de pregătire baschetbalistică. Sunt de părere că această platformă exploatează o nișă în domeniul sportiv care vizează atât segmentul de jucători profesioniști care își doresc o rampă de lansare în zona baschetului profesionist cât și fanii care își doresc să simtă emoțiile jocului în fața unui public, cu o miză la mijloc. Pune la dispoziție o multitudine de utilități, gândite să ușureze viața utilizatorilor pe platformă.

Implementarea este bazată pe o arhitectură în 3 nivele, fiecare cu rol propriu care împreună asigură funcționarea corectă și eficientă și oferă o fereastră între sistem și utilizator prietenoasă și intuitivă. Nivelul logic al platformei este unul eficient care prelucrează cu atenție datele primite de la baza de date pentru a oferi mereu cele mai detaliate și corecte date utilizatorilor păstrând integritatea ligii sportive.

În concluzie, sunt de părere că platforma prezentată în acest document rezolvă o multitudine de probleme din lumea sportivă, prin posibilitatea de a te înscrie ca jucător independent fără să fi legat vreo echipă cât și simplitatea și intuitivitatea interfeței de care se lovește utilizatorul în prima secundă când alege să folosească acest sistem, oferind soluții moderne la diferite probleme cu care se confruntă zi de zi atât foști jucători profesioniști care nu mai sunt la nivelul necesar să joace într-o ligă națională cât și fanilor amatori de sport.

7.1 Dezvoltări ulterioare

Platforma are implementate și funcționale toate metodele de bază pentru a putea fi lansată în producție și a primi un feedback pozitiv din partea utilizatorilor, totuși, în opinia mea există loc de îmbunătățiri și modalități de a face sistemul unul mai performant și mai actual decât este deja, platforma implementată reprezentând o baza

solidă pentru viitoare actualizări și metodologii noi mai moderne, deoarece a fost gândită să fie cât mai versatilă posibil.

Prima dezvoltare la care mă gândesc este implementarea acesteia pe un sistem mobil, dat fiind faptul că majoritatea persoanelor din ziua de azi își petrec statistic nu mai puțin de 5-6 ore pe zi pe telefoanele mobile, o aplicație mobilă ar fi următorul pas pentru liga sportivă atât din punct de vedere comercial cât și tehnic încât tranziția nu ar implica un cumul mare de muncă sau configurări. Dacă platforma va fi vreodată expusă producției în masă, acest tip de dezvoltare mobilă pare să fie cel mai logic pas din punct de vedere al profitului, cât și al costurilor de producție încât o astfel de schimbare nu este dificilă din punct de vedere tehnic.

A doua dezvoltare care nu a fost tratată este cea de chat live direct pe aplicație. În momentul de față singura modalitate de comunicare între utilizatori este prin intermediul invitațiilor și al emailul prezent în pagina de profil, însă există situații în care o comunicare mai detaliată este necesară înaintea unei colaborări între un jucător și o echipa, iar un sistem de live chat ar fi util în rezolvarea acestei probleme.

Sunt de părere că această funcționalitate va aduce un răspuns majoritar pozitiv din partea utilizatorilor în primul rând fiindcă ar familiariza foarte mult platforma cu gusturile actuale ale persoanelor care sunt conectate la internet și la mesaje digitale în fiecare zi. Totodată va aduce și un plus de comunicare între componentele echipelor prezente în ligă, iar informațiile interesante și utile vor circula mult mai repede.

Ultima dezvoltare la care m-am gândit este posibilitatea de streaming live de meciuri. Deși eu pot simula meciuri între diferite echipe, acestea nu sunt prezente în format video pentru ca fanii să le poată descărca și salva în așa fel încât să le vadă ulterior în timpul liber sau se bucure chiar în acel moment alături de prieteni așa cum sunt televizate meciurile din ziua de azi.

8 BIBLIOGRAFIE

- [1] Estimated probability of competing in professional athletics. NCAA. [Online]. Disponibil: <https://www.ncaa.org/sports/2015/3/6/estimated-probability-of-competing-in-professional-athletics.aspx>
- [2] Olympic History - From Ancient Greek Olympics to Modern Olympic Games. [Online]. Disponibil: <https://olympics.com/ioc/ancient-olympic-games/history>
- [3] Marani, I. N., Yulianti, E., Rhewindinar, Subarkah, A., Widyaningsih, H., & Jauhari, M. (2021). Effects of Mass Media in Sports. In Proceedings of the 5th International Conference on Sport Science and Health (ICSSH 2021). Advances in Health Sciences Research, 45. Atlantis Press. <https://doi.org/10.2991/ahsr.k.220203.014>

- [4] Platforma web NBA. [Online], Disponibil: <https://www.nba.com/>. [Accesat 10.04.2024]
- [5] Platforma web AMBB. [Online], Disponibil: <https://baschetbucuresti.ro/> [Accesat 12.04.2024]
- [6] Platforma web ESPN. [Online], Disponibil: <https://www.espn.com/> [Accesat 13.04.2024]
- [7] What is three-tier architecture IBM/ [Online]. Disponibil: <https://www.ibm.com/topics/three-tier-architecture>
- [8] M. Jones, J. Bradley, N. Sakimura. (2015). RFC 7519 - JSON Web Token (JWT). Internet Engineering Task Force (IETF). Disponibil: <https://datatracker.ietf.org/doc/html/rfc7519>
- [9] What is a RESTful API Amazon Services/ [Online]. Disponibil: <https://aws.amazon.com/what-is/restful-api/> [Accesat 16.03.2024]
- [10] Documentația oficială SpringBoot [Online], Disponibil: <https://spring.io/projects/spring-boot>. [Accesat 15.03.2024]
- [11] Fette, I., & Melnikov, A. (2011). The WebSocket Protocol. Internet Engineering Task Force (IETF). Disponibil: <https://datatracker.ietf.org/doc/html/rfc6455>
- [12] What is React.js? - Brief Framework Intro / Overview. [Online]. Disponibil: <https://www.sanity.io/glossary/react-js> [Accesat 17.03.2024]
- [13] What are the uses of CSS? [Online], Disponibil: <https://www.javatpoint.com/what-are-the-uses-of-css>
- [14] Letkowski, J. (2015). Doing database design with MySQL. *Journal of Technology Research*, Western New England University.
- [15] Koch, S. (2003). *JavaScript: A Programmer's Companion from Basic Through DHTML, CSS and DOM*. United Kingdom: Wiley. [Carte].
- [16] Postman official documentation. [Online]. Disponibil: <https://www.postman.com/product/what-is-postman/> [Accesat 6.04.2024]
- [17] The architecture of the Spring Boot REST applications. [Online]. Disponibil: <https://medium.com/@piyumisudusinghe/the-architecture-of-the-spring-boot-rest-applications-3643e905c5f8> [Accesat 4.04.2024]
- [18] What is Dom in React – Javatpoint. [Online]. Disponibil: <https://www.javatpoint.com/what-is-dom-in-react> [Accesat 10.03.2024]
- [19] NavLink v6.24.0. [Online]. Disponibil: <https://reactrouter.com/en/main/components/nav-link> [Accesat 28.03.2024]

[20] Amazon Web Services. *What is CORS? - Cross-Origin Resource Sharing Explained* [Online]. Disponibil: <https://aws.amazon.com/what-is/cross-origin-resource-sharing/> [Accesat 20.03.2024]

9 ANEXE

9.1 Anexa A - Grafice si diagrame

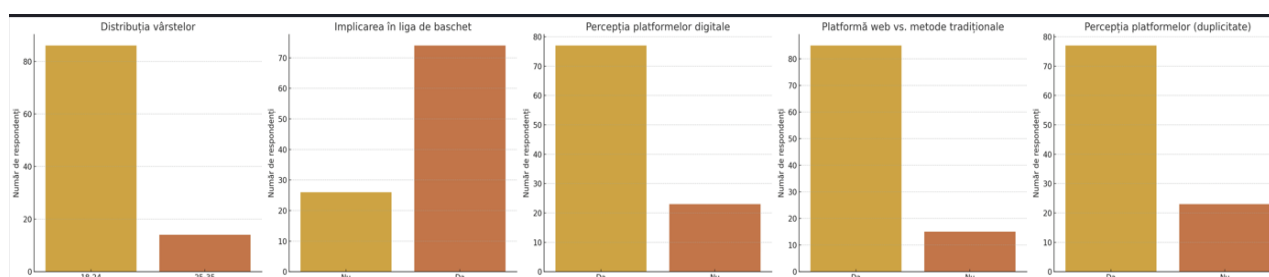


Figura A1. Grafice reprezentative pentru funcționalitățile ligii de baschet

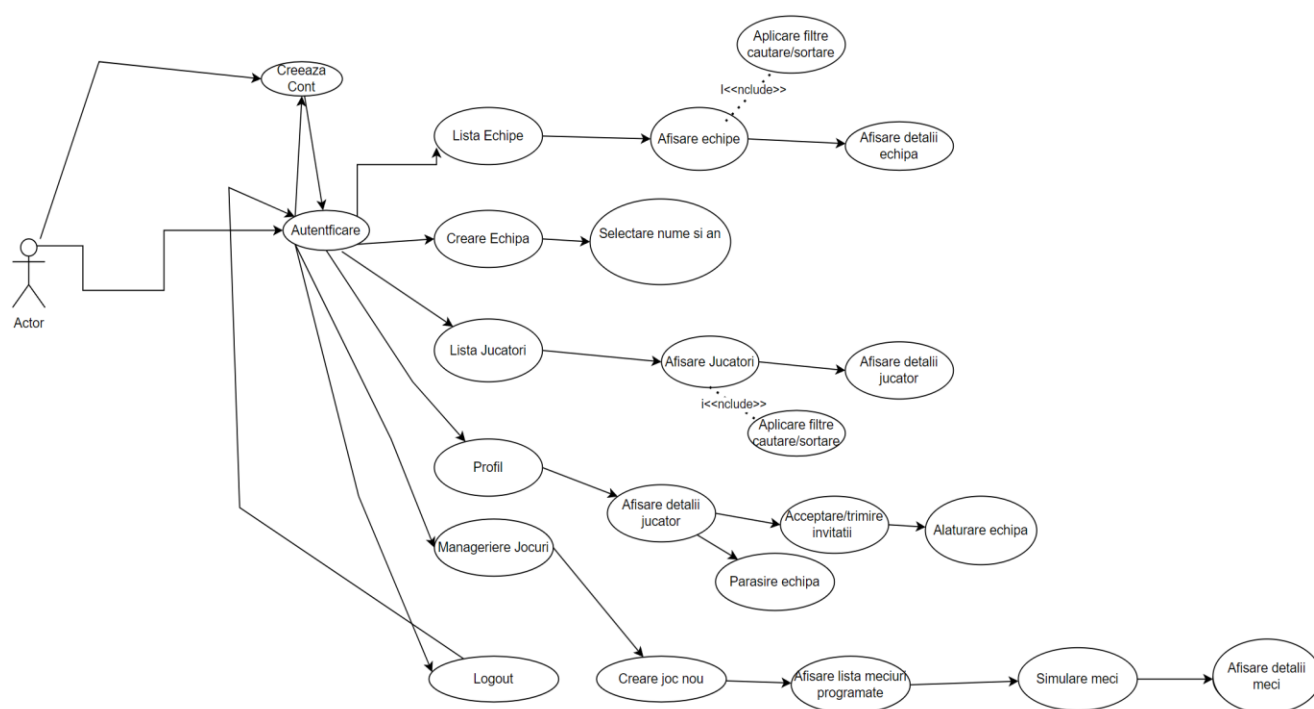


Figura A2. Diagrama de cazuri de utilizare pentru utilizatorul clasic

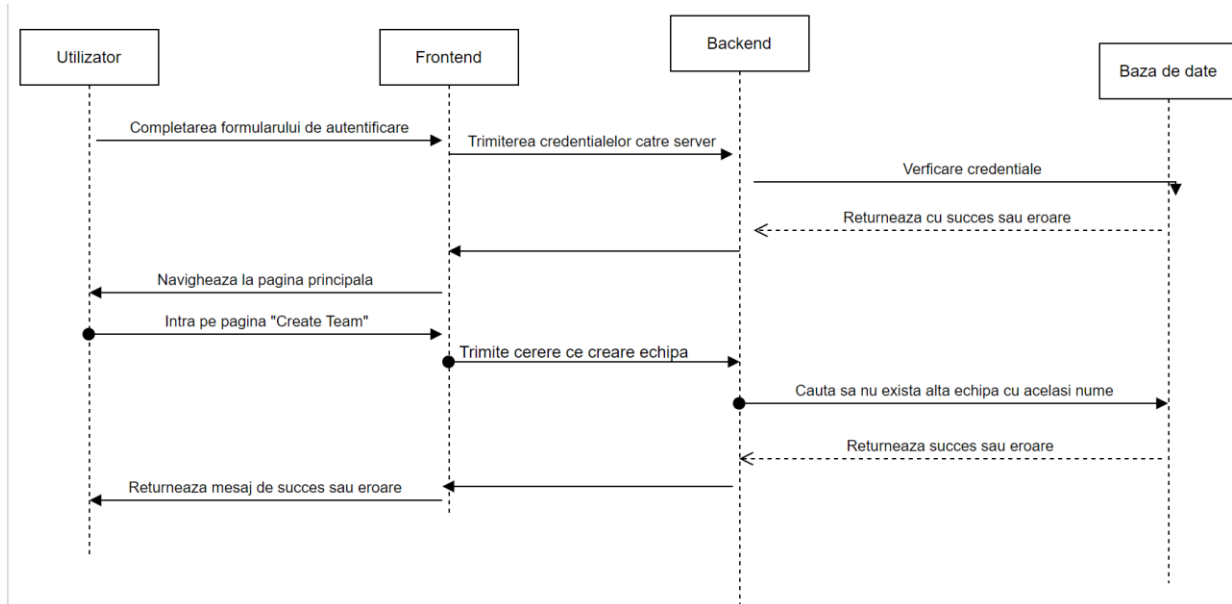


Figura A3. – Diagrama de secvență a scenariului de creare echipa

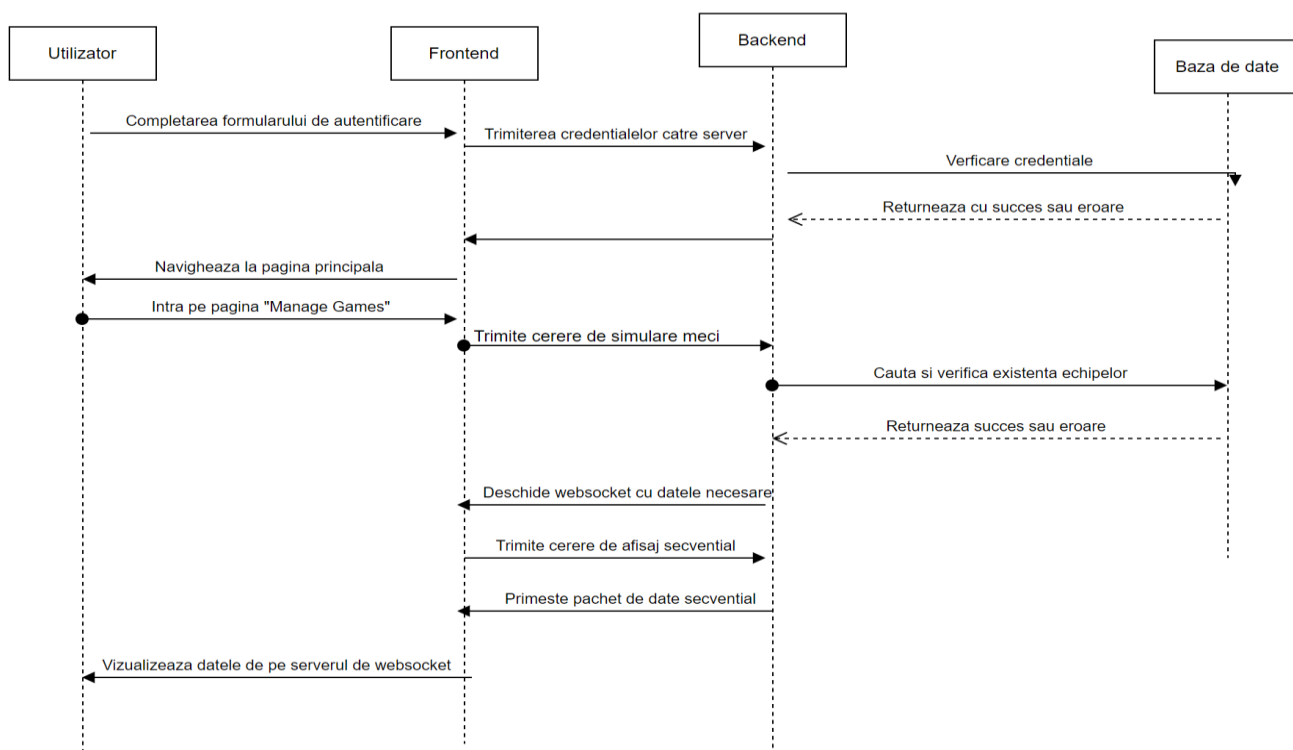


Figura A4. Diagrama de secvență pentru scenariul de simulare meci

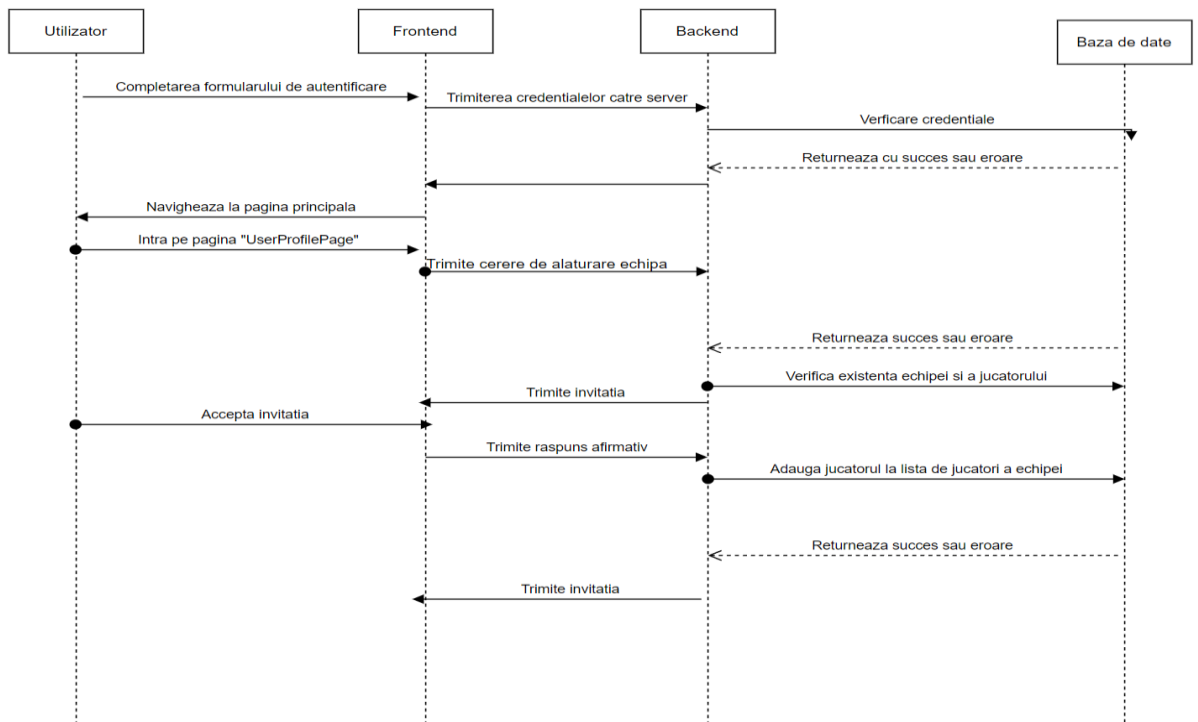


Figura A5. Diagrama de secvență a scenariului trimitere si acceptare invitație echipa

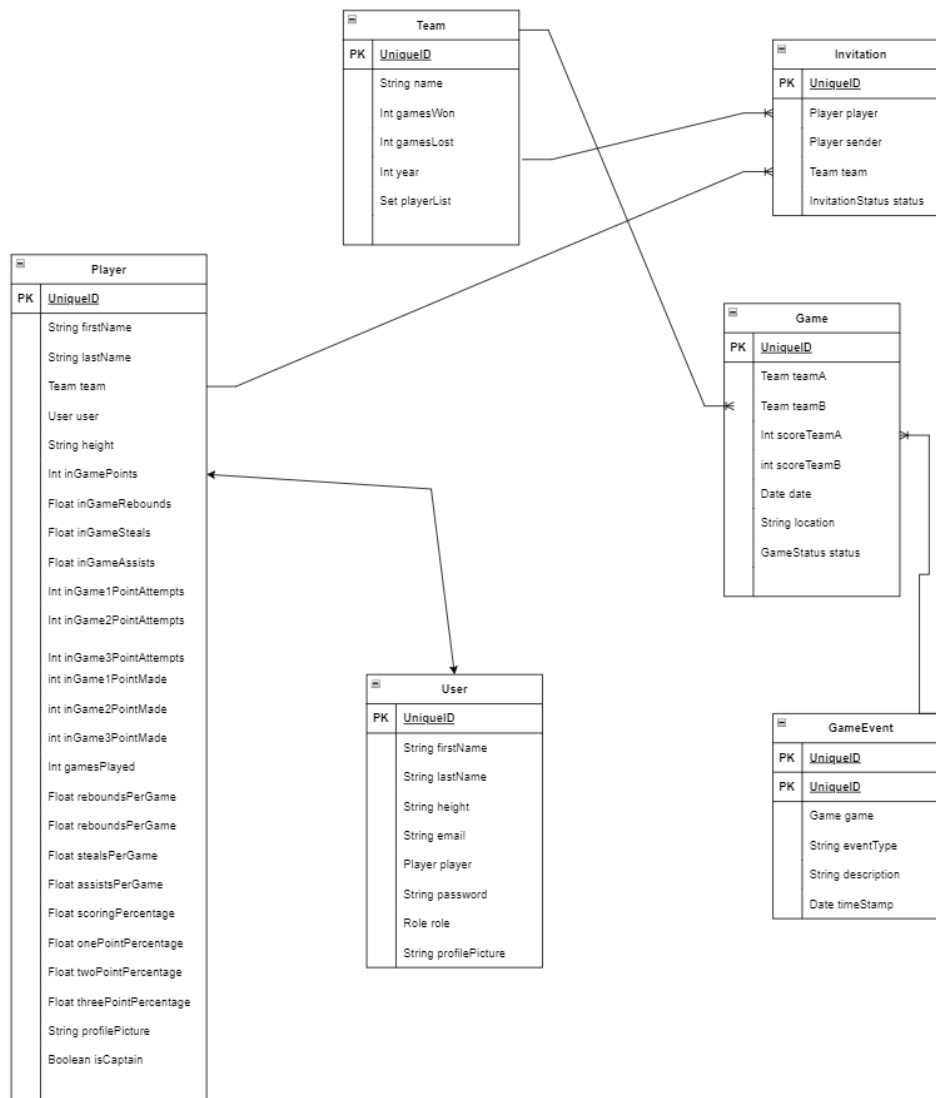


Figura A6. Schema bazei de date

9.2 Anexa B – Exemple platforme similare

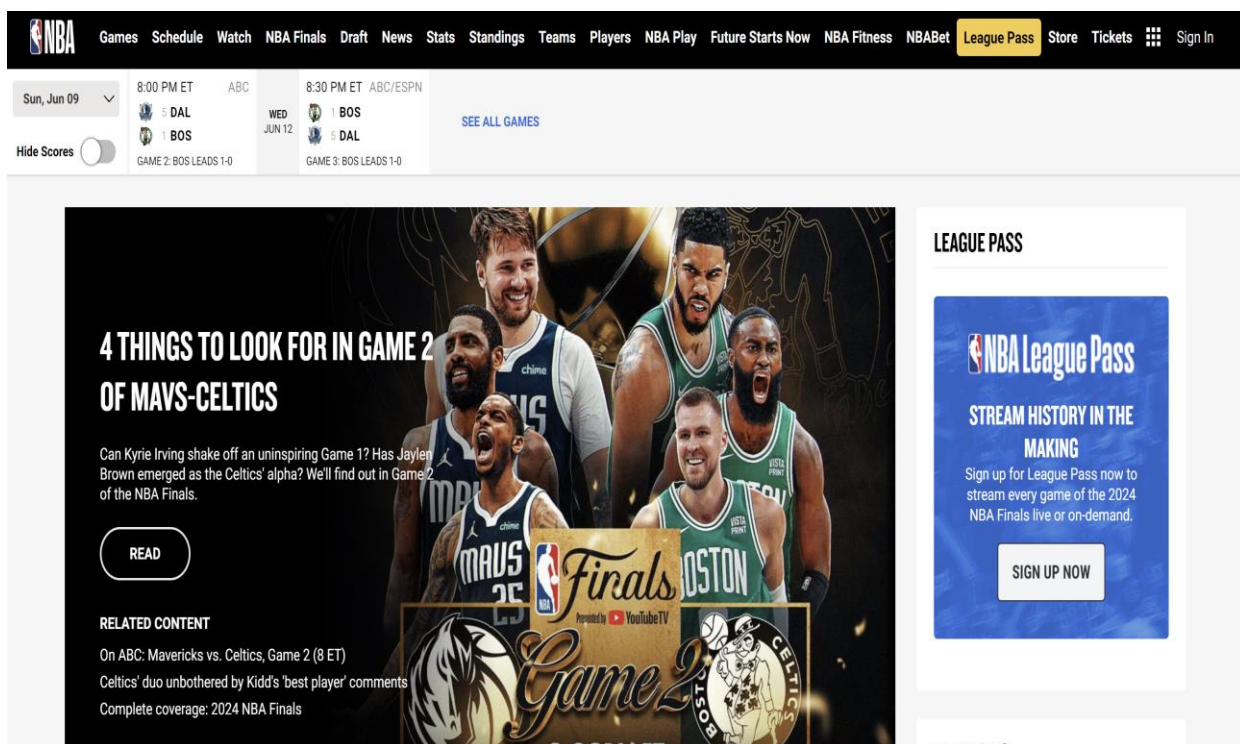


Figura B1. Pagina principala din cadrul aplicație web NBA.com



Figura B2. Pagina principala din cadrul aplicației web AMBB.ro

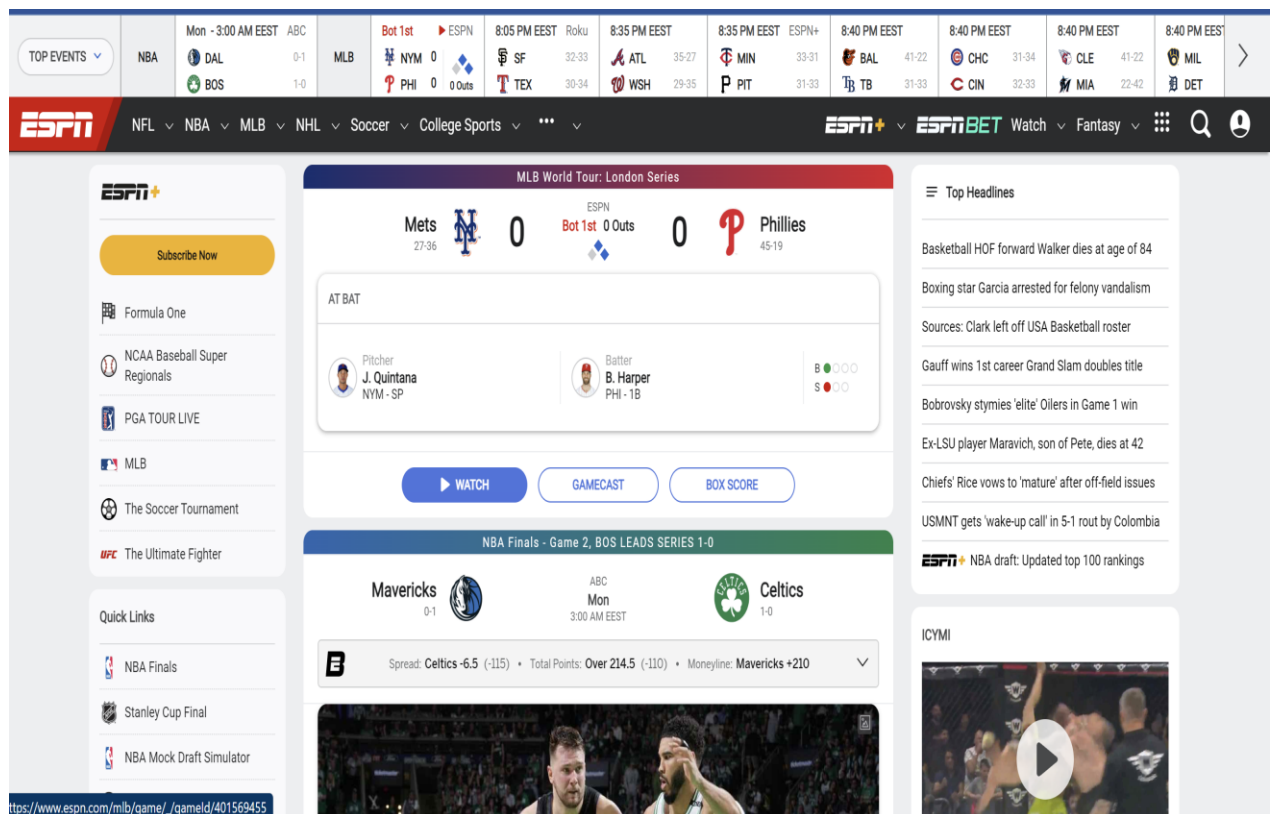


Figura B3. Pagina principala ESPN.com

9.3 Anexa C- Exemple cod si Colecție Postman

```

1 def simulateGame(game):
2     resetPlayerStats(game)
3     game.status = "IN_PROGRESS"
4     saveGame(game)
5     notifyGameplay(game)
6
7     startTime = currentTime()
8     gameDuration = 10000
9
10    while currentTime() - startTime < gameDuration:
11        updateTeamA = randomBoolean()
12
13        player = selectScoringPlayer(game.teamA if updateTeamA else game.teamB)
14        pointsType = randomInt(1, 3)
15        scoreChange = calculateScoreChange(pointsType)
16        attemptsChange = calculateAttemptsChange(scoreChange)
17
18        reboundsChange = randomInt(0, 1)
19        stealsChange = randomInt(0, 1)
20        assistsChange = randomInt(0, 1)
21
22        if updateTeamA:
23            game.scoreTeamA += scoreChange
24        else:
25            game.scoreTeamB += scoreChange
26
27        updatePlayerInGameStats(player, scoreChange, pointsType, attemptsChange, reboundsChange, stealsChange, assistsChange)
28        notifyPlayerUpdate(player)
29
30        event = createGameEvent(game, "SCORE", player.name + " scored " + scoreChange + " points.")
31        saveGameEvent(event)
32
33        sleep(1000)
34        saveGame(game)
35        notifyGameplay(game)
36
37    game.status = "FINISHED"
38    saveGame(game)
39    updatePlayerStatsAfterGame(game.teamA.players)
40    updatePlayerStatsAfterGame(game.teamB.players)
41    updateTeamResults(game)
42    notifyGameplay(game)

```

Figura C1. Pseudocod metodă simulare meci

```

def selectScoringPlayer(team):
    players = getPlayersByTeam(team)

    totalWeight = sum(calculatePlayerWeight(player) for player in players)
    if totalWeight == 0:
        return randomChoice(players) # Fallback in case all weights are 0

    randomValue = randomDouble() * totalWeight
    for player in players:
        randomValue -= calculatePlayerWeight(player)
        if randomValue <= 0:
            return player

    return players[-1] # Fallback in case of rounding issues

def calculatePlayerWeight(player):
    pointsPerGameWeight = player.pointsPerGame
    onePointPercentage = player.onePointPercentage
    twoPointPercentage = player.twoPointPercentage
    threePointPercentage = player.threePointPercentage

    scoringPercentageWeight = averageOfPositiveValues(onePointPercentage, twoPointPercentage, threePointPercentage)
    baseWeight = 1.0 # Base weight to avoid complete exclusion
    experienceFactor = 1.0 if player.gamesPlayed > 0 else 0.1
    heightFactor = parseHeight(player.height)

    totalWeight = baseWeight + (pointsPerGameWeight + scoringPercentageWeight + heightFactor) * experienceFactor
    return totalWeight

```

Figura C2. Pseudocod metoda alegere jucător care să marcheze

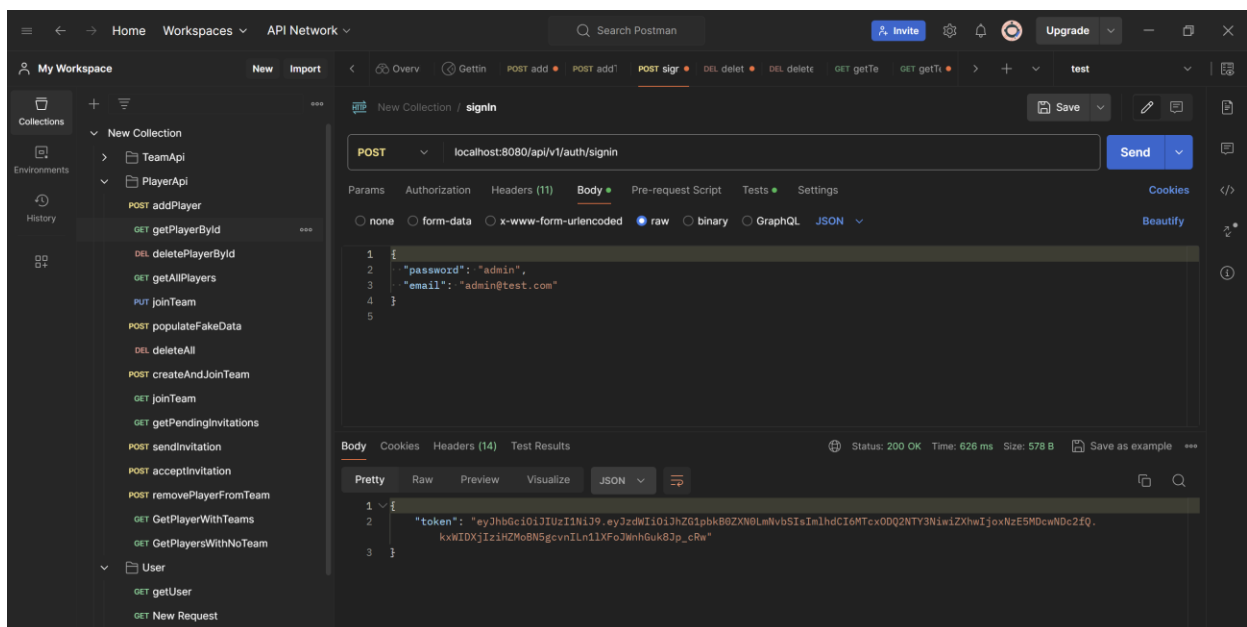


Figura C3. Colecție teste manuale