UNIVERSITATEA “STEFAN CEL MARE”, SUCEAVA

FACULTATEA DE INGINERIE ELECTRICA SI STIINTA CALCULATOARELOR

SPECIALIZAREA CALCULATOARE

PROIECT DISCIPLINA POO

[Football Rating app]

Autor

Studentul Hrițuc Denis

TEMA PROIECT

TEMA SI MOTIVATIA ALEGERII

Tema proiectului constă în dezvoltarea unei aplicații pentru evaluarea jucătorilor de fotbal pe baza unor criterii statistice (rating automat) și a opiniilor utilizatorilor (rating manual).

Această aplicație are ca scop integrarea pasiunii pentru fotbal cu dezvoltarea competențelor de programare orientată pe obiect. Ratingurile automatizate vor fi generate pe baza unor factori precum goluri, pase decisive, recuperări și alți indicatori de performanță, iar utilizatorii vor putea evalua subiectiv jucătorii în urma vizionării meciurilor. Această dublă abordare asigură un echilibru între obiectiv și subiectiv și poate avea aplicații în fantezia fotbalului sau analiza sportivă.

CUPRINS

TEMA ȘI MOTIVAȚIA ALEGERII  
 1. ELEMENTE TEORETICE  
 1.1. DESCRIEREA PROBLEMEI  
 1.2. ABORDAREA TEORETICĂ A PROBLEMEI  
 1.3. ELEMENTE SPECIFICE POO  
 1.4. ALTE CAPITOLE SPECIFICE  
 2. IMPLEMENTARE  
 2.1. TEHNOLOGII FOLOSITE  
 2.2. DIAGRAMA DE CLASE  
 3. ANALIZA SOLUȚIEI IMPLEMENTATE  
 3.1. FORMATUL DATELOR DE I/O  
 3.2. STUDII DE CAZ  
 3.3. PERFORMANȚE OBȚINUTE  
 4. MANUAL DE UTILIZARE

1. ELEMENTE TEORETICE

1.1 DESCRIEREA PROBLEMEI

Proiectul presupune dezvoltarea unei aplicații care gestionează un set de jucători de fotbal, permițând atribuirea unui rating automat (calculat) și unul oferit de utilizatori. Aplicația va gestiona datele jucătorilor și va permite afișarea celor două tipuri de rating.

1.2 ABORDAREA TEORETICĂ A PROBLEMEI

Ratingul automat va fi calculat folosind o formulă ponderată în funcție de statistici specifice. Se va realiza o interfață în consolă, cu salvarea datelor în fișiere. Se vor utiliza noțiuni precum clase, vectori de obiecte, funcții virtuale (pentru posibilă extensibilitate), citire/scriere din fișiere text.

1.3 ELEMENTE SPECIFICE POO

Se vor utiliza conceptele fundamentale ale programării orientate pe obiect:

* **Încapsulare**: pentru protejarea datelor despre jucători.
* **Moștenire**: pentru a crea tipuri specializate de jucători (ex: Portar, Fundas, Atacant), fiecare cu metode proprii de calcul al ratingului automat, bazate pe statistici relevante pentru poziția respectivă.
* **Polimorfism**: pentru metode generice de afișare/calcul.
* **Clase și obiecte**: pentru modelarea entităților (jucător, statistici, rating etc.).

1.4 ALTE CAPITOLE SPECIFICE

* Structura modulară a codului
* Validarea input-ului utilizatorului
* Salvarea ratingurilor într-un fișier CSV

2. IMPLEMENTARE

2.1 TEHNOLOGII FOLOSITE

* **Limbaj**: C++
* **IDE**: Visual Studio Code
* **Altele**: Fișiere text pentru salvarea datelor

2.2 IMPLEMENTAREA FUNCȚIONALITĂȚILOR SPECIFICE

Aplicația „Player Ratings” urmărește să ofere, pentru fiecare jucător, un scor numeric calculat automat (pe baza performanțelor în meciuri) sau introdus manual de către utilizator. Pe lângă calculul punctual al unui rating pentru un singur meci, sistemul trebuie să permită:

1. **Înregistrarea fiecărui rating pe meci**, asociat jucător-oponent-dată.
2. **Calculul mediei rating-urilor pe ultimele *X* zile**, cu excluderea meciurilor mai vechi de această perioadă.
3. **Salvarea tuturor rating-urilor (automate și, eventual, manuale) într-un fișier CSV** pentru export

3. ANALIZA SOLUȚIEI IMPLEMENTATE

3.1 FORMATUL DATELOR DE I/O

**Intrare:**

* Datele de intrare pot fi introduse manual, printr-un meniu interactiv, sau preluate dintr-un fișier text (CSV) care conține, pentru fiecare jucător și meci, următoarele câmpuri:
  + **playerName** (șir de caractere): Numele complet al jucătorului.
  + **position** (șir de caractere): Poziția în teren („Forward”, „Midfielder”, „Defender” sau „Goalkeeper”).
  + **opponent** (șir de caractere): Numele echipei adversare în acel meci.
  + **matchDate** (șir de caractere, format “YYYY-MM-DD”): Data meciului.
  + **statistici meci** (valori numerice întregi sau reale):
    - Goals, Assists, Tackles, KeyPasses, SuccessfulDribbles, FinalThirdPass, ProgressiveCarries, BadPasses, Fouls, Offsides, YellowCards, RedCards, OwnGoals, UnsuccessfulDribbles, Interceptions, Clearances, AerialDuelsWon, Blocks, Saves, CleanSheets, CrossesStopped, GoalsConceded, PunchesMissed.
  + Opțional, dacă sistemul acceptă o notare manuală ulterioară:
    - **userScore** (reel): Punctaj introdus de utilizator ca override peste ratingul automat.
    - **hasUserScore** (boolean): Indicator boolean al existenței unei note manuale.

La citirea din fișier, fiecare linie este tokenizată după separatorul , și se populează structura Player p (inclusiv sub-structura PlayerStats cu toate statisticile enumerate). Apoi, prin apelul ratingManager.ratePlayer(p), se înregistrează în memoria internă obiectul MatchRating aferent.

**Ieșire:**

* În modul consoliă (CLI), datele pot fi afișate astfel:
  + Lista completă a jucătorilor cu toate meciurile lor
  + Raportul agregat “Average Ratings (Last X Days)”
  + Rezultatul căutărilor

3.2. STUDII DE CAZ

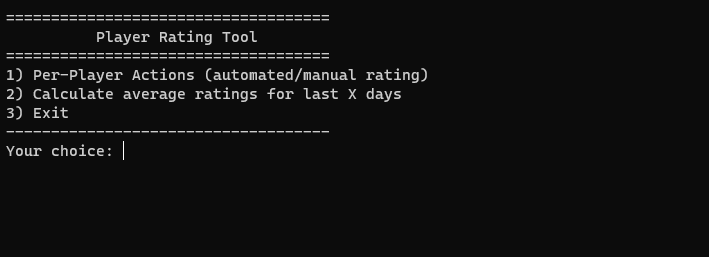
* **Înregistrare meci nou:** Adăugăm „Francesco Esposito vs. AC Milan, 2025-04-23” → ratePlayer stochează corect data și scorul calculat automat.
* **Calcul medie ultime 30 zile:** Pentru „Francesco Esposito” cu meciuri la 13, 27 și 34 zile în urmă, getAverageRatingLastDays("Francesco Esposito", 30) include doar meciurile cu ageDays < 30.
* **Filtrare multiplă:** Căutăm meciurile lui „Marco Bianchi” poziție Midfielder și adversar „Juventus” în ultimele 90 zile → se returnează doar intrările care îndeplinesc toate criteriile.
* **Export CSV:** generează un fișier cu rânduri

3.3. PERFORMANȚE OBTINUTE

* **Inserare/actualizare (ratePlayer)**: operație în O(log P + log M) per meci.
* **Calcul medie (getAverageRatingLastDays)**: O(log P + L) per jucător (L = număr meciuri).
* **Export CSV**: O(P + T) pentru scrierea tuturor intrărilor.

4. MANUAL DE UTILIZARE

**1. Meniu Principal**



In meniul principal, utilizatorul are posibilitatea de a selecta dintre trei optiuni:

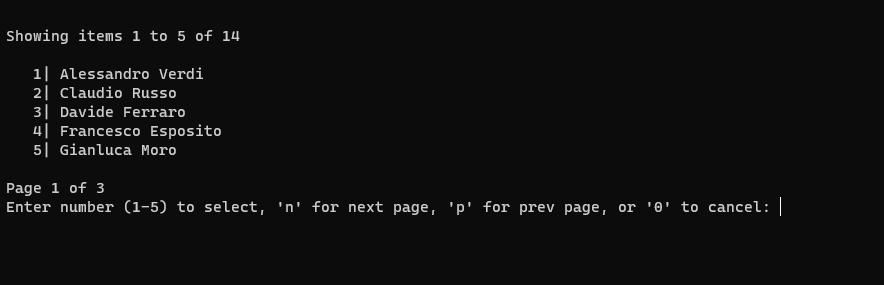
-Actiuni per player

-Media rating-urilor pe ultimele x zile

-Exit, daca este cazul.

**2.Meniul de actiuni per player**

In acest meniu, utilizatorul este intrebat daca vrea sa caute un jucator anume, sau sa i se afiseze toata lista cu jucatori. Dupa afisarea listei de jucatori, el are posibilitatea de a alege un jucator.

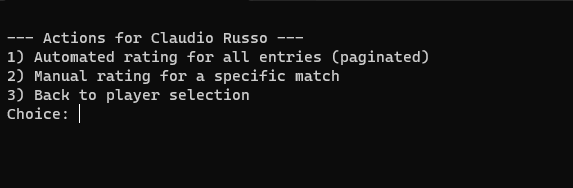


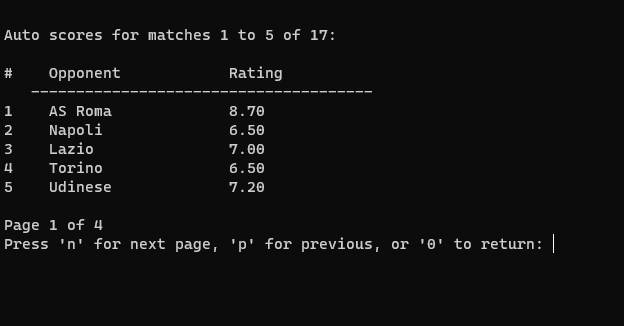
Dupa alegerea acestuia, utilizatorul poate alege din 3 optiuni:

- Calculul automat al rating-ului pentru fiecare din meciurile jucate, bazat pe statistica jucatorului in meci.

- Introducerea unui rating manual pentru unul din meciurile jucatorului, la alegere.

- Inapoi la sectiunea cu toata lista de jucatori.





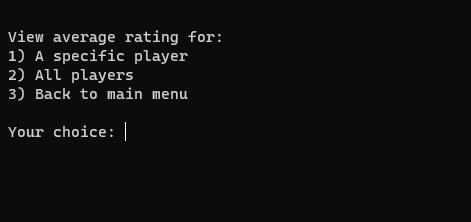
**3. Meniul de media rating-urilor pe ultimele x zile**

Aici avem posibilitatea de a calcula media rating-urilor jucatorilor pe ultimele x zile, zilele fiind introduse de utilizator. Prima data, utilizatorul are de ales dintre 3 optiuni:

-Sa vada media pentru un jucator specific, care numele ii va fi introdus la tastatura.

-Sa vada media pentru toti jucatorii.

-Inapoi la meniul principal





**CAPITOLUL V – CONCLUZII**

**Descrierea interfeței produsului**

Interfața aplicației „Player Ratings” este realizată în modul consola (CLI), cu un meniu principal care permite:

* Vizualizarea listelor de jucători și a meciurilor stocate.
* Generarea rapoartelor (medii pe ultimele X zile sau export CSV).
* Comenzile sunt numerotate, iar utilizatorul tastează cifra corespunzătoare fiecărei opțiuni pentru a naviga.

**Datele de intrare și modul de introducere**

* Datele pot fi furnizate dintr-un fișier CSV iar programul parsează automat și apelează ratePlayer pentru fiecare intrare.
* Validarea minimală se face la citire: formatul datei, câmpuri numerice, poziție validă.

**Rezultatele oferite și formatul acestora**

* Lista meciurilor (pentru fiecare jucător): afișează adversarul și rating-ul automat.
* Rapoarte de medii: “Average Ratings (Last X Days)” – pentru fiecare jucător media numerică cu două zecimale.
* Export CSV: fișier cu antet urmat de câte un rând per înregistrare, cu valorile corespunzătoare.

**Operațiile disponibile și logica acestora**

* **Adăugare meci (ratePlayer)**:

Selectează strategia (automat sau override manual).

Calculează scorul prin metoda calculateRating.

* **Calcul medie pe ultimele X zile**:

Preia todayDays = getTodayInDays().

Pentru fiecare meci al jucătorului: calculează ageDays = todayDays – dateDays.

Include în sumă doar meciurile cu 0 ≤ ageDays < X.

Returnează media scorurilor (dacă există cel puțin un meci relevant; altfel 0.0).

* **Afișare debug și medie (printAverageRatingsLastDays)**:

Pentru fiecare jucător, afișează detaliat fiecare meci cu dateDays, ageDays și score.

Apoi afișează media pe ultimele X zile.

* **Export CSV (saveRatingsToCSV)**:

Deschide fișier nou și scrie antetul.

Iterează prin toate intrările din matchRatings, scriind câte un rând de format CSV pentru fiecare „jucător–adversar”.

**CAPITOLUL VI – BIBLIOGRAFIE**

1. Cplusplus.com – C++ Reference and Tutorials, https://cplusplus.com
2. cppreference.com – C++ Reference Documentation, https://en.cppreference.com
3. OpenAI ChatGPT