

# SQUAD MANAGER

## SPORT TEAM MANAGEMENT APP

Developed by

- ↳ Avian Silviu
- ↳ Dijmărescu Cristina
- ↳ Ionescu Alexandru
- ↳ Stan Ana-Maria

## DESCRIERE APLICATIE

“Squad Manager” este o aplicație web care ajută în gestionarea unui club sportiv de fotbal. Aplicația a fost dezvoltată special pentru a ușura munca cluburilor sportive, ajutându-i pe oamenii ce activează în industria sportivă să gestioneze mai facil clubul, să aibă acces oricând la toate informațiile echipei.

Acest software preia din atribuțiile managerului clubului, memorând toate datele importante de la nivelul clubului. Astfel, aplicația este mereu pregătită să îți ofere orice informație ai nevoie referitoare la echipa ta.

Aplicația are mai multe roluri pentru utilizator, inclusiv administrator, membru al staffului și jucător, fiecare cu acces diferit la functionalitatile aplicatiei.

In Squad Manager sunt integrate si functionalitati ale altor aplicatii, cum ar fi Excel, PDF pentru a exporta usor date și API-urile meteo pentru a vizualiza datele meteorologice corespunzatoare meciurilor viitoare.

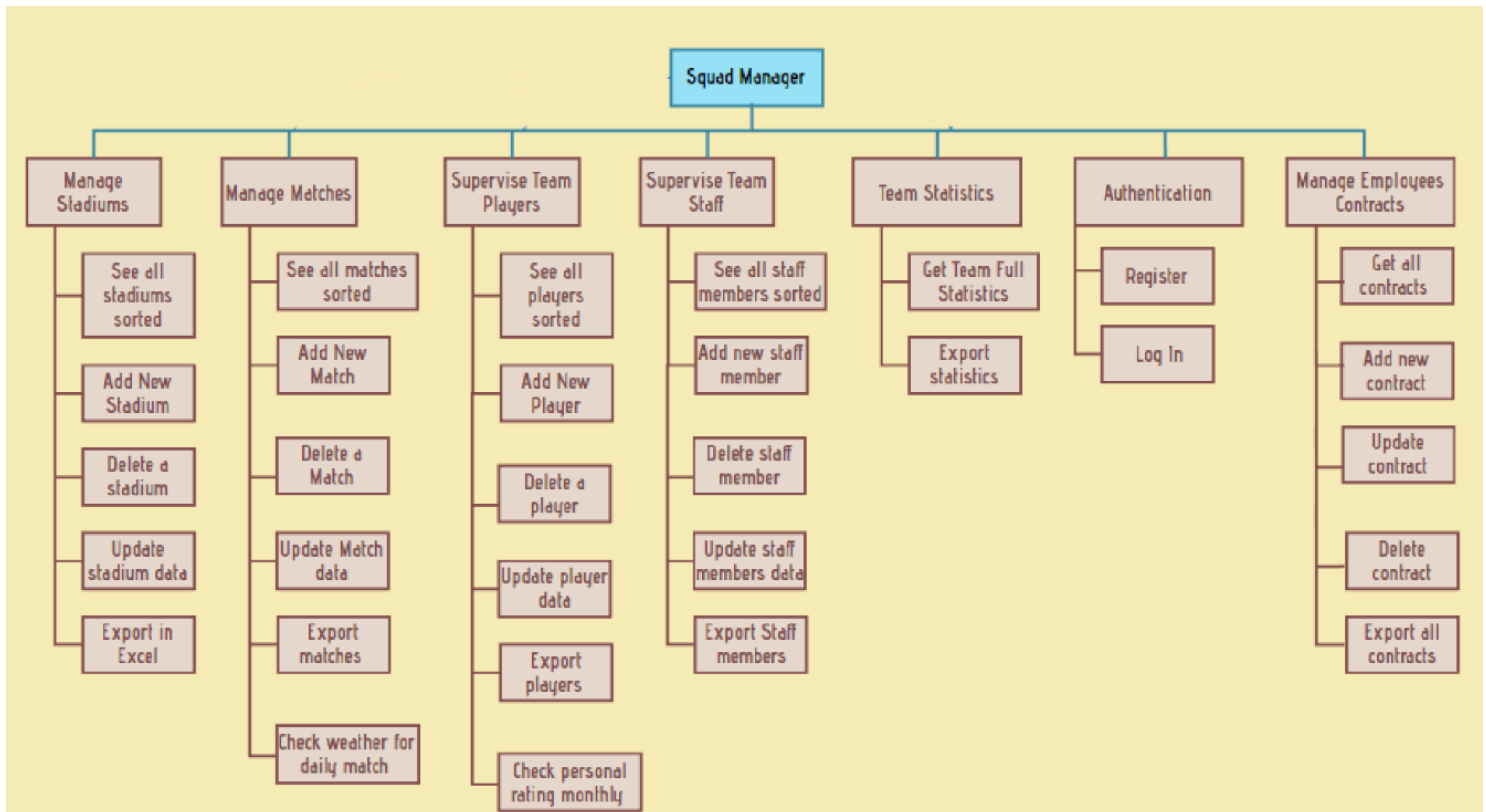
Datele aplicatiei sunt stocate cu ajutorul platformei Firestore, astfel Firebase este folosita ca baza de date principală pentru a salva datele fiecărei secțiuni din club. Scopul principal al aplicației este de a îmbunătăți organizarea și performanța clubului, oferind o modalitate ușoară pentru membrii clubului de a accesa și gestiona datele și activitățile clubului.

### **Atributele principale ale aplicației Squad Manager sunt:**

- Simplitate in utilizare - aplicația este ușor de utilizat de catre utilizatori cu diferite niveluri de experiență tehnică
- Performanța - aplicația poate gestiona numarul de utilizatori prezenti in cadrul unui club sportiv și poate raspunde rapid cerintelor acestora.
- Scalabilitatea - aplicația are posibilitatea de a fi imbunatatita pentru a gestiona un numar ridicat de utilizatori și date in viitor, daca clubul isi maresta echipa.

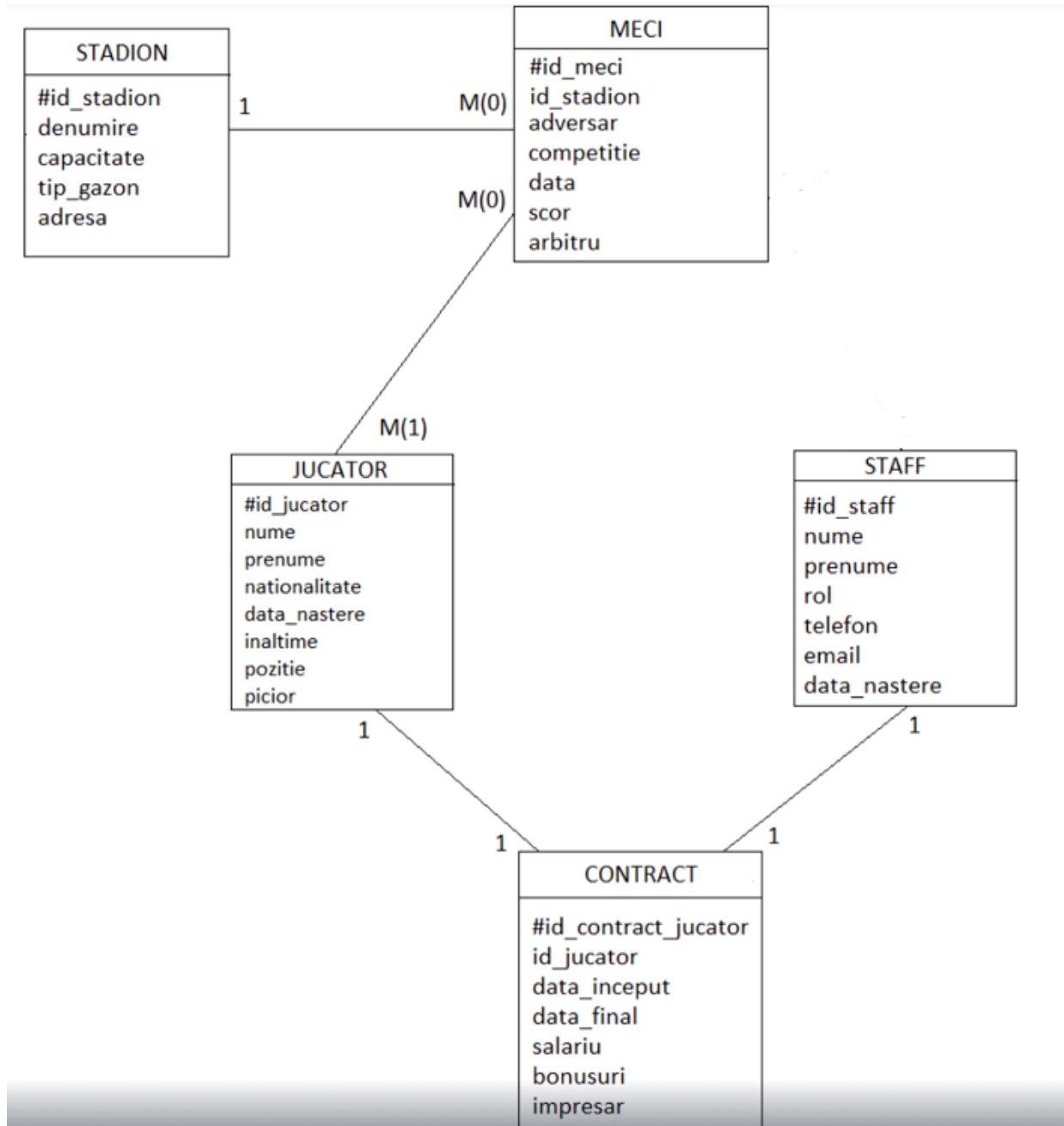
- Accesibilitatea - aplicația este accesibilă utilizatorilor de pe diferite dispozitive și din orice locație.
- Managementul datelor - aplicația oferă o modalitate eficientă de a stoca, gestiona și accesa datele importante pentru club, utilizând platforma Firebase.
- Roluri și permisiuni specifice utilizatorului - aplicația are roluri și permisiuni diferite pentru utilizatorii din club (jucator, staff sau admin)
- Integrarea - aplicația este integrată cu servicii și platforme externe pentru a oferi mai multe funcționalități utilizatorilor, cum ar fi exportul datelor din pagini în Excel/PDF și afisare vreme pentru meciurile viitoare folosind API-uri.
- Generare statistici - aplicația oferă funcționalitatea de vizualizare și exportare statistici pentru a ajuta clubul să analizeze și să ia decizii bazate pe date concrete.
- Îmbunătățirea continuă - aplicația poate fi actualizată și îmbunătățită continuu pentru a se adapta nevoilor dinamice ale clubului.

# FUNCTIONAL DECOMPOSITION DIAGRAM



In aceasta diagrama am inclus toate controllerele din aplicatie (paginile principale ale aplicatiei). Pentru fiecare dintre acestea am adaugat toate metodele utilizate pentru a efectua CRUD la nivelul bazei de date.

# ENTITY RELATIONSHIP DIAGRAM



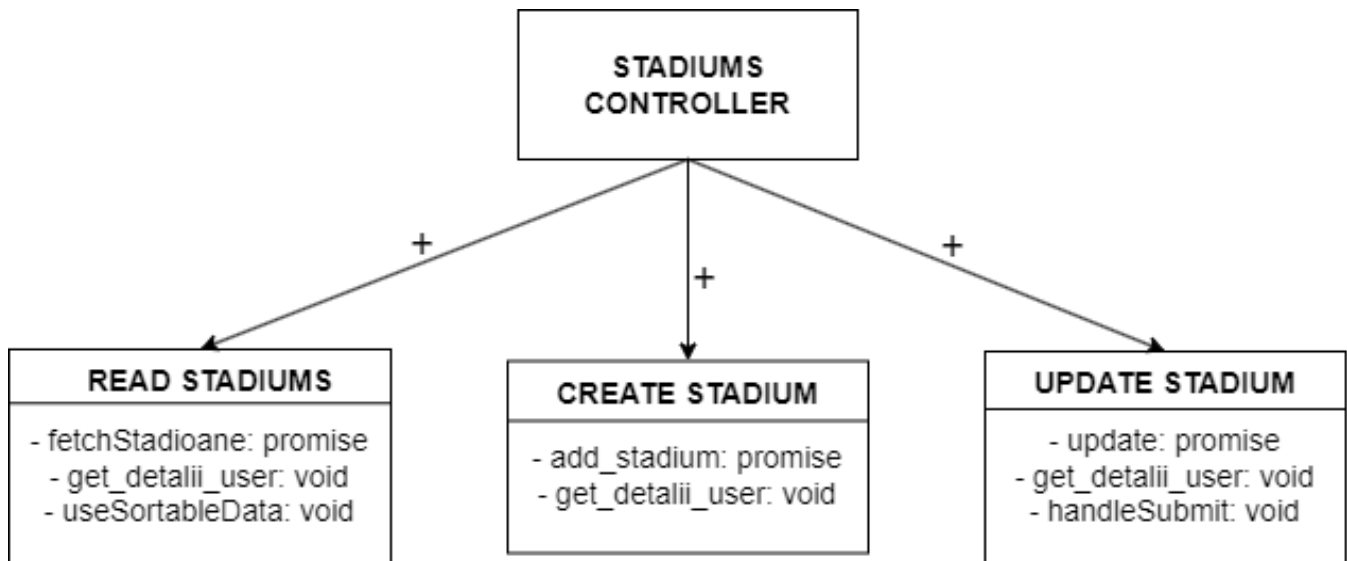
Aplicatia Squad Manager are in spate o baza de date realizata cu ajutorul Firebase, compusa din 5 entitati: stadion, meci, jucator, staff si contract.

Tabelele Stadion si Meci sunt legate printr-o relatie de tip One to Many: pe un stadion se pot juca mai multe meciuri, iar un meci este obligatoriu sa se joace pe un singur stadion.

Intre Meci si Jucator vom avea o relatie Many to Many deoarece intr-un meci participa mai multi jucatori, iar un jucator poate fi utilizat in mai multe meciuri.

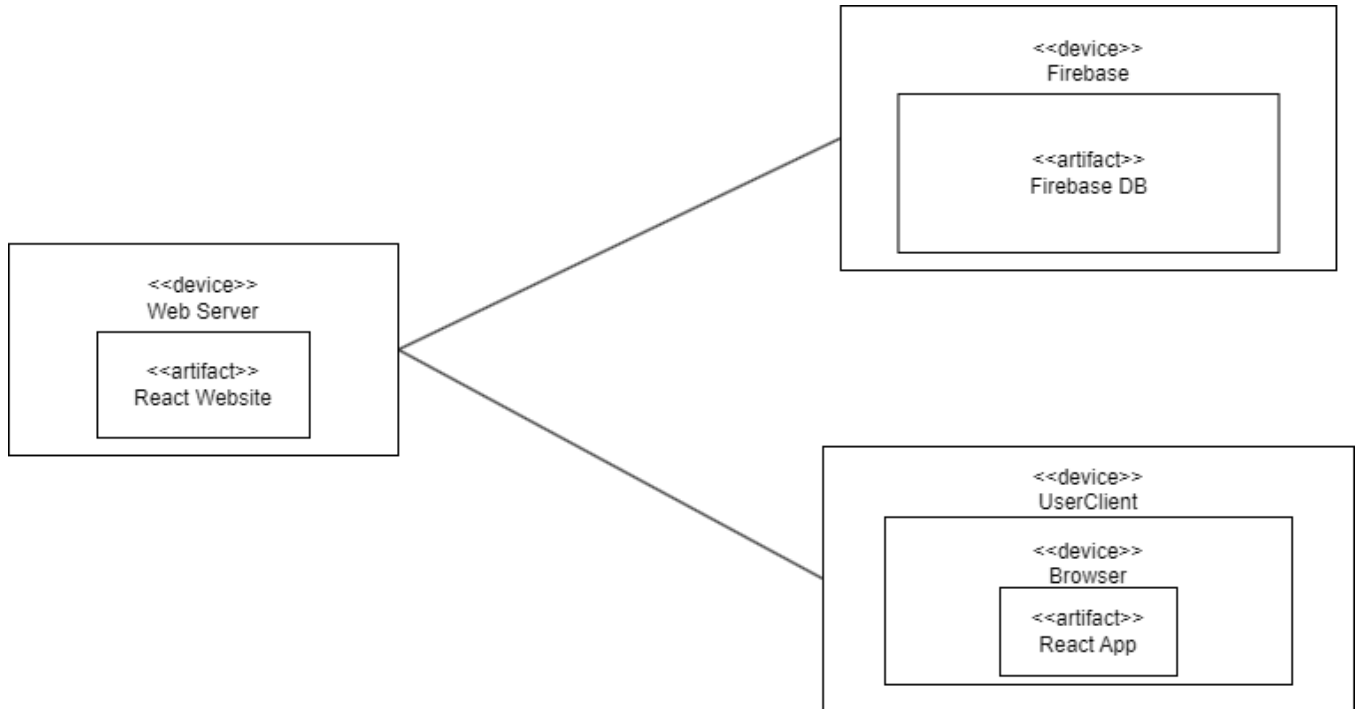
Entitatile Jucator, Contract si Staff sunt legate prin relatii de tip One to One, fiecare jucator, respectiv membru al staffului avand un contract unic semnat cu clubul sportiv respectiv.

## PACKAGE DIAGRAM



In aceasta diagrama am prezentat controllerul pentru stadioane in care avem adaugate 3 pagini principale pe care le pot accesa utilizatorii. Aceste pagini sunt publice, pot fi accesate din afara controllerului pentru ca au fost definite rute speciale catre acestea. Fiecare dintre cele 3 pagini contine cateva functii utlizate pentru a realiza modificari la nivelul bazei de date. Aceste functii sunt accesibile doar la nivelul controllerului respectiv.

## DEPLOYMENT DIAGRAM



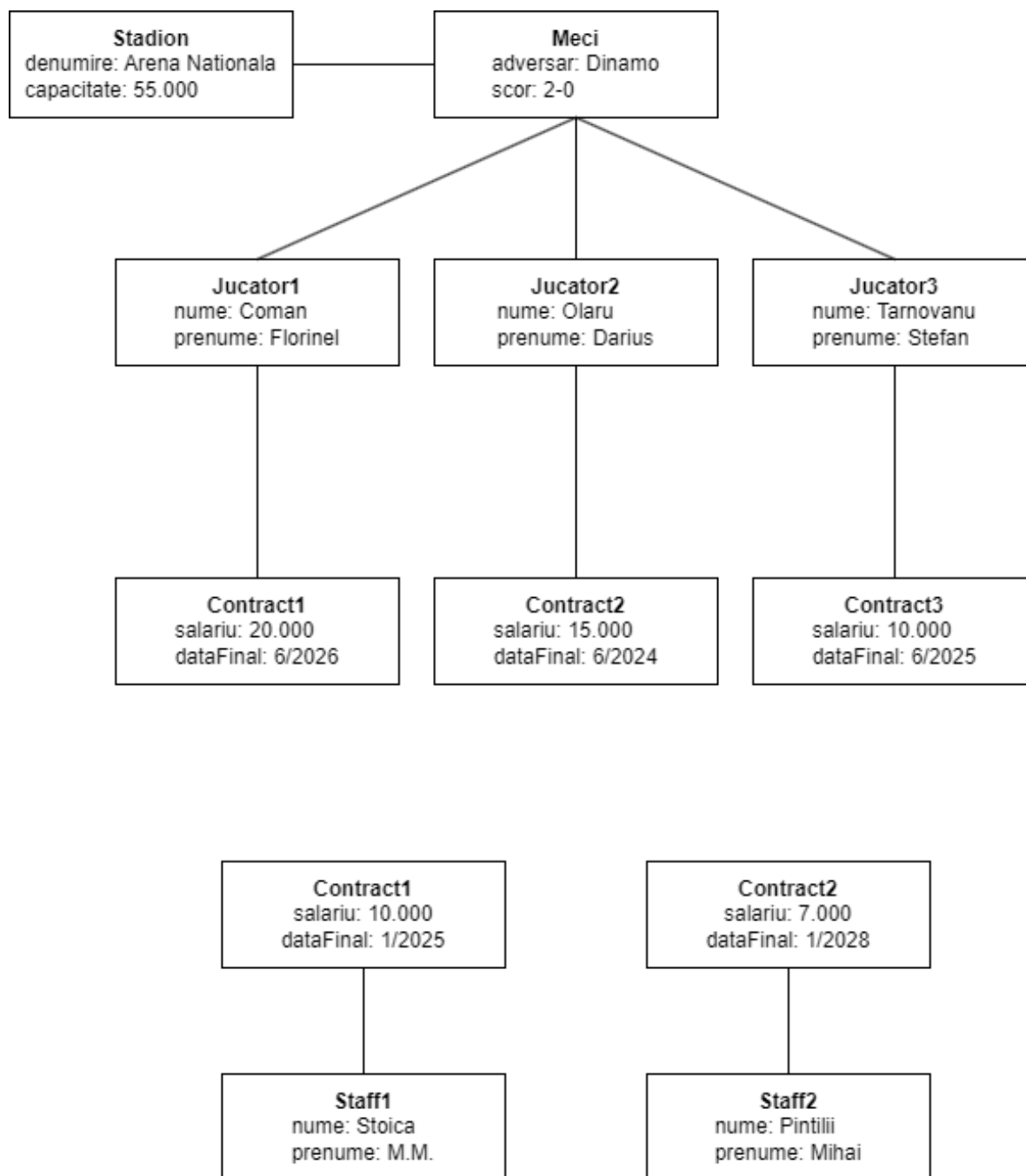
Squad Manager este o aplicație web dezvoltată în framework-ul React, utilizând JavaScript ca tehnologie principală.

Aplicația oferă o modalitate eficientă de stocare, gestionare și accesare a datelor importante pentru club, utilizând platforma Firebase.

User-ul aplicației utilizează browserul pentru a accesa platforma Squad Manager.



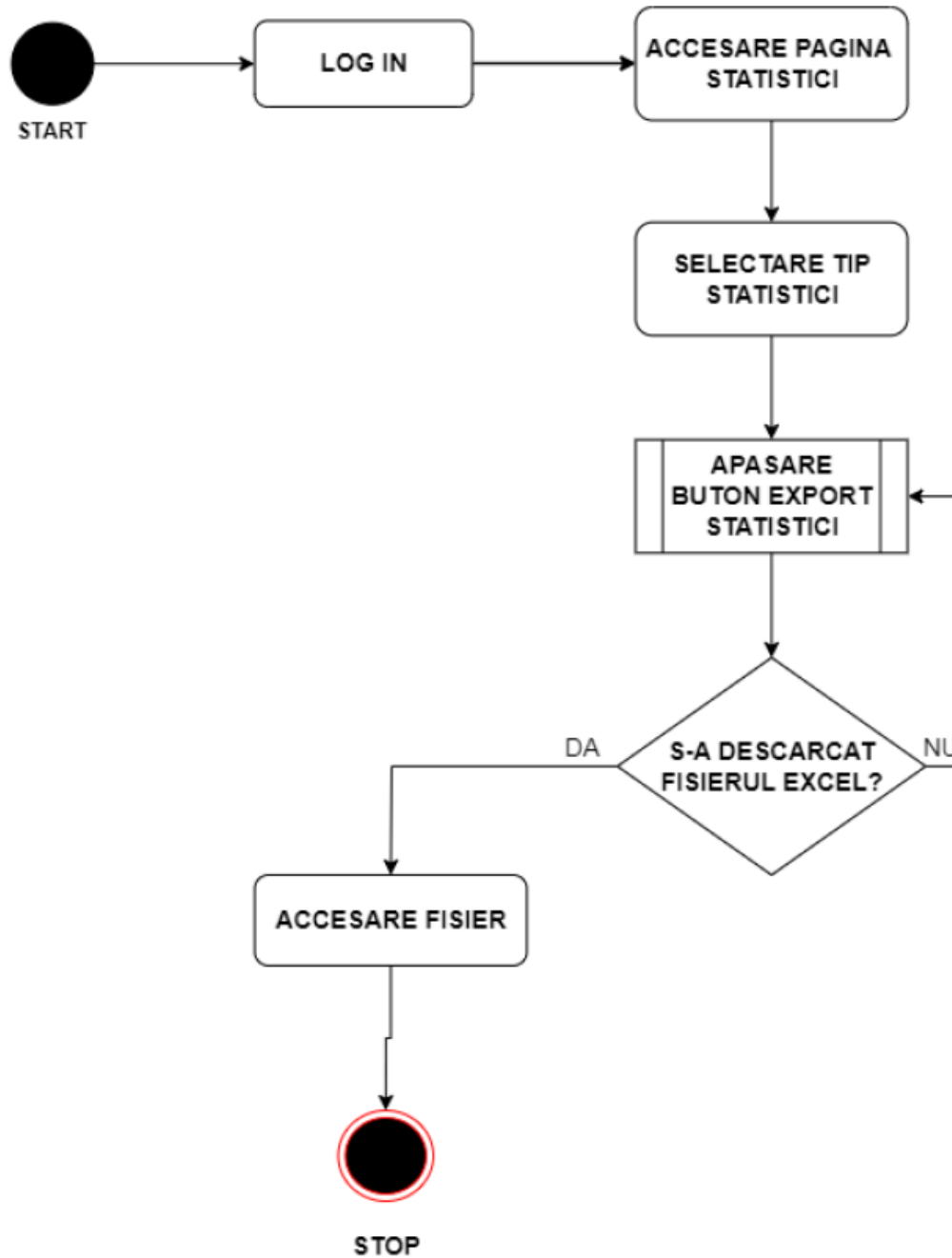
## OBJECT DIAGRAM



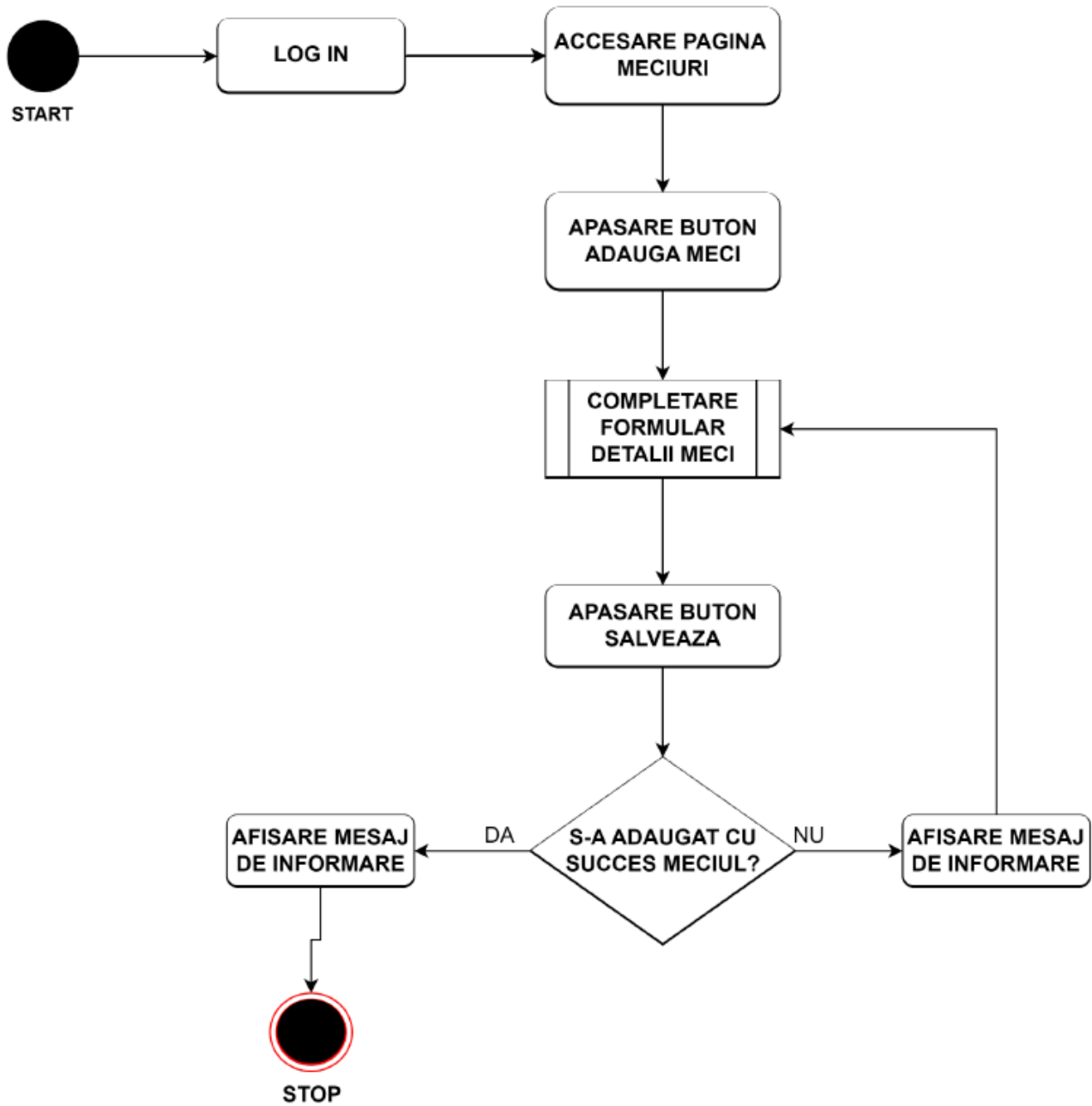
Pornind de la diagrama de clase ER prezentata anterior, am realizat diagrama de obiecte reprezentand printr-un exemplu concret toate clasele de la nivelul bazei de date.

# ACTIVITY DIAGRAMS

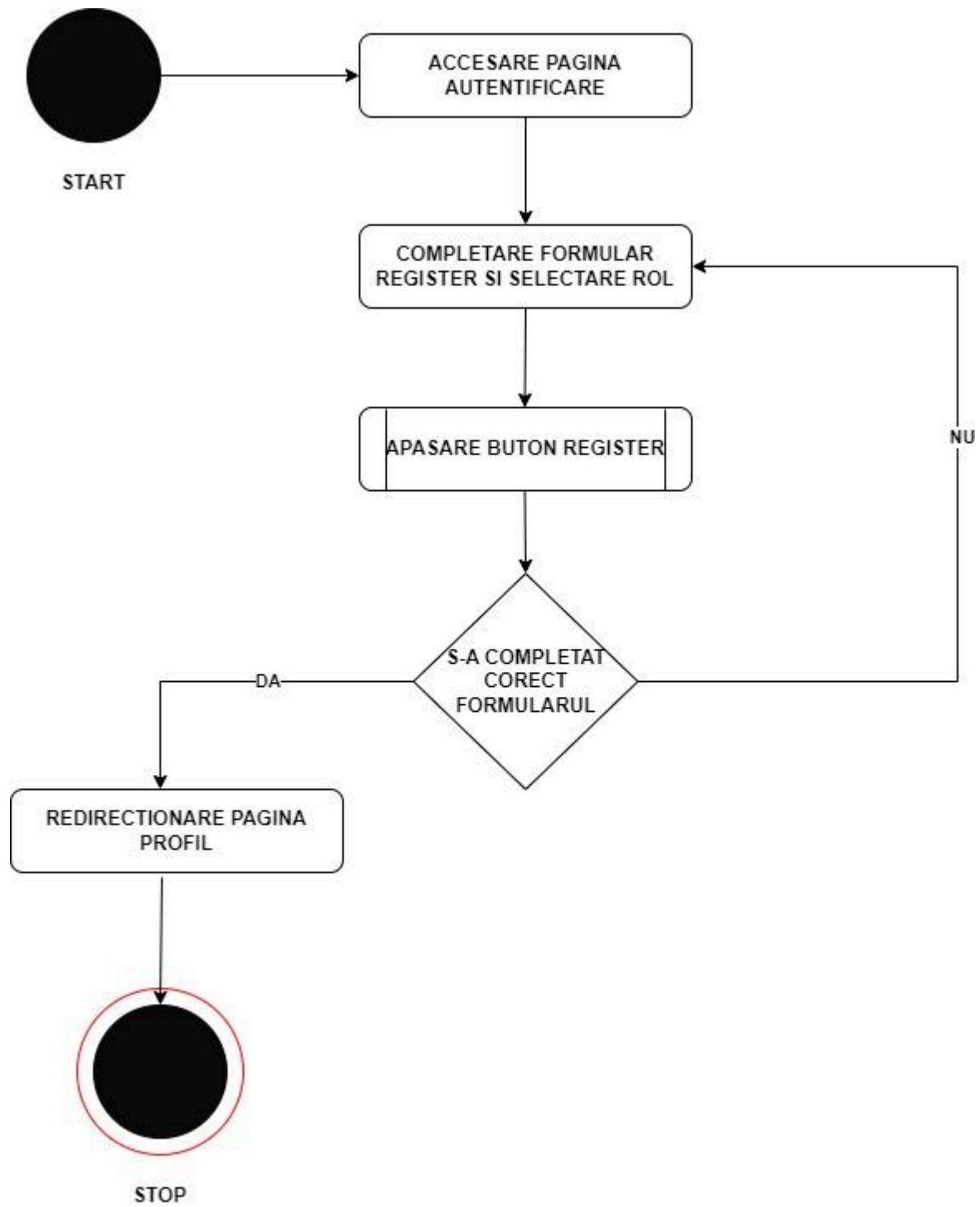
- Exportare statistici in Excel



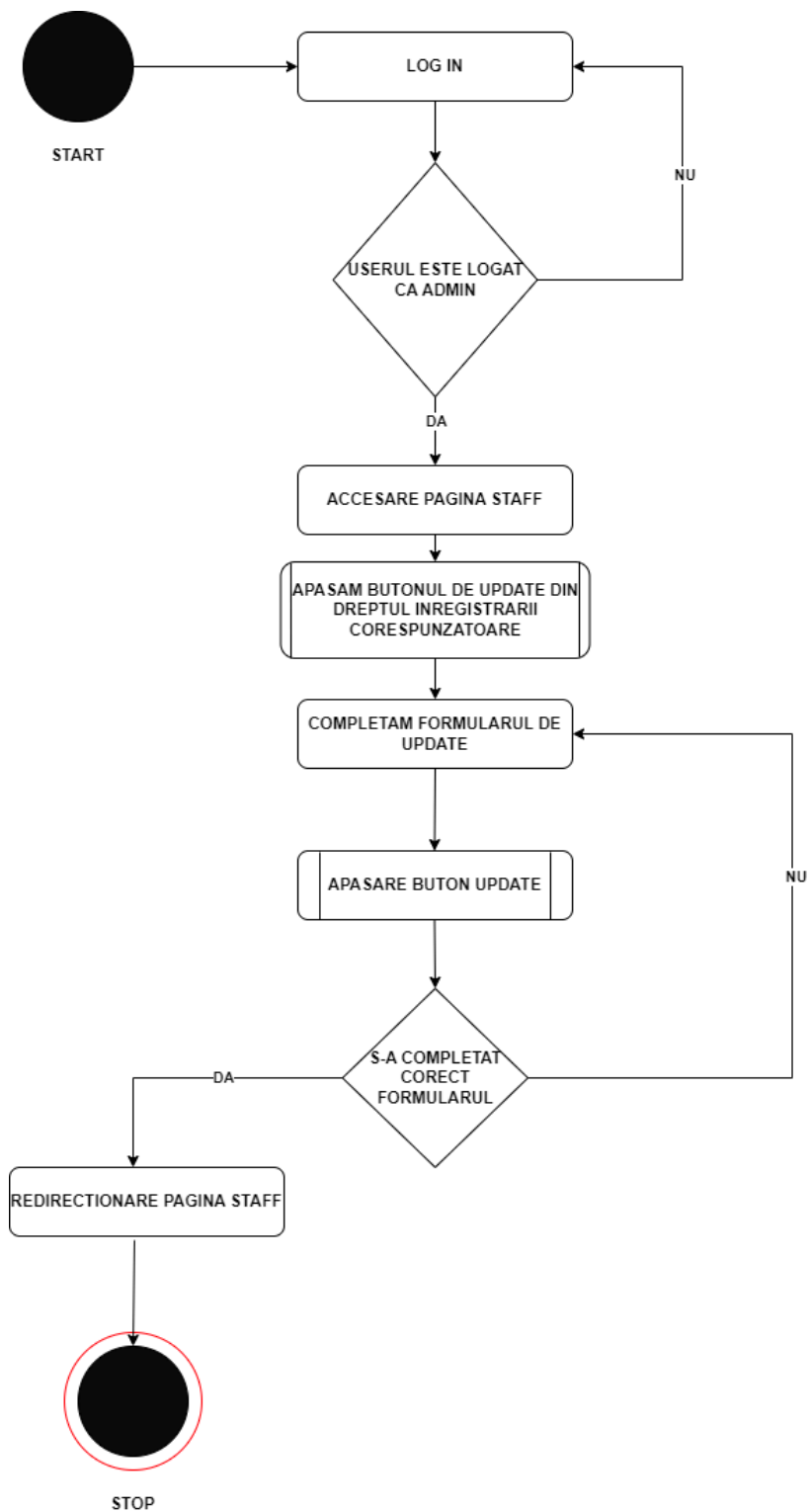
- Adaugare meci nou in pagina dedicata meciurilor



- Register



- **Update inregistrare din pagina Staff**



# PROJECT CHARTER DOCUMENT

TITLE		PROJECT NAME	
WEB APPLICATION		SQUAD MANAGER	
TEAM			
PROJECT MANAGER	PROJECT TEAM MEMBERS		PROJECT STAKEHOLDERS
Avian Silviu Dijmarescu Cristina	Avian Silviu Dijmarescu Cristina	Ionescu Alexandru Stan Ana-Maria	Chirita Claudia - profesor FMI Razvan Burleanu - presedinte FRF Gino Iorgulescu - presedinte LPF
SPECS			
BUSINESS CASE		PROJECT PURPOSE	
Imbunatatirea performantelor sportive cu cel putin 10% Imbunatatirea satisfactiei jucatorilor si staffului cu 30% Transparenta privind rezultate individuale si colective Monitorizarea eficienta a randamentului staffului		Ofera cluburilor sportive posibilitatea de a manageria cu usurinta activitatile echipei si de a imbunatati performantele.	
PROJECT DELIVERABLES		PROJECT SCOPE	
Livrabil intermediar (planificare) Livrabil final (aplicatia)		Implementarea unei aplicatii usor de utilizat, avand pagini dedicate fiecarei sectiuni dintr-un club sportiv si posibilitate de monitorizare statistici.	
PROJECT BENEFITS		PROJECT RISKS	
Manageriera tuturor sectiunilor din club Planificarea si monitorizarea activitatilor sportive Reducerea ratei de eroare in statisticile clubului Obtinerea unui avantaj competitional		Risc depasire timp alocat datorita utilizarii unor noi tehnologii (React si Firebase) si adaptarii cu platforma GitHub.	
RESOURCES		PROJECT BUDGET	
React Visual Studio Code HTML + CSS	Firebase GitHub Cursuri FMI	Bugetul este exprimat in numarul de ore disponibile pentru a fi alocate proiectului: ~ 100 ore	
MILESTONES			
STARTING DATE	MILESTONE COMPLETION DATES		PROJECT COMPLETION DATE
25 Noiembrie 2023	Planificare	1/12/2023	20 Ianuarie 2024
	Backend design	25/12/2023	
	Frontend design	10/01/2024	
	Testare	15/01/2024	
	Go Live	20/01/2024	

# DESIGN PATTERNS USED IN THE APPLICATION

## Motivele pentru care am folosit Design Pattern-urile:

- Functional Component - Exemplu: Pagina Home

Ele sunt mai simple și sunt folosite pentru componente care trebuie să fie randate, neavând nevoie de state cum ar fi pagina de Home.

- Container Component - Exemplu: Weather API

Componentele containerului sunt adesea folosite pentru a gestiona logica complexă cu ajutorul state-urilor, cum ar fi preluarea datelor din Weather API sau gestionarea interacțiunilor utilizatorului.

- Compound Components - Exemplu: Weather API

Rolul principal al unei componente de container compus este de a acționa ca un container de nivel superior care gestionează starea și logica pentru mai multe componente ale containerului copil. În cazul nostru, widget-ul de vreme primește state-ul de la părintele său.

- Conditional Rendering - Exemplu: NavBar-ul se modifică în funcție de rol

Randarea condiționată este adesea folosită de noi pentru a crea interfețe dinamice și receptivă, unde conținutul afișat pe ecran se poate modifica în funcție de acțiunile utilizatorului sau de alte evenimente.

- Controlled Components? - Exemplu: Formulare de Add si Update

Rolul principal al componentelor controlate este de a se asigura că starea componentei este întotdeauna sincronizată cu starea componentei părinte. În aplicație noastră comunicăm formularul cu field-urile din cadrul acestuia.

- React Hooks? - Exemplu: La afisarea jucatorilor

React Hooks sunt funcții care ne-au permis să utilizăm state-uri și alte caracteristici React în componente funcționale, făcându-le mai puternice și versatile. Am folosit `useEffect` pentru a afișa lista jucătorilor la încărcarea paginii.

## Benefits and Tradeoffs

### → Functional Component

Beneficii:

- simplitate
- performanță
- posibilitatea de refolosire
- usurinta testarii

Compromisuri:

- functionalitate limitată

### → Container Component

Beneficii:

- performanță
- posibilitatea de refolosire
- usurinta testarii



Compromisuri:  
-complexitate

#### → Compound Components

Beneficii:  
-performanta

Compromisuri:  
-complexitate

#### → Conditional Rendering

Beneficii:  
-interfata dinamica

Compromisuri:  
-complexitate  
-posibilitatea de refolosire limitata

#### → Controlled Components

Beneficii:  
-comportament predictibil

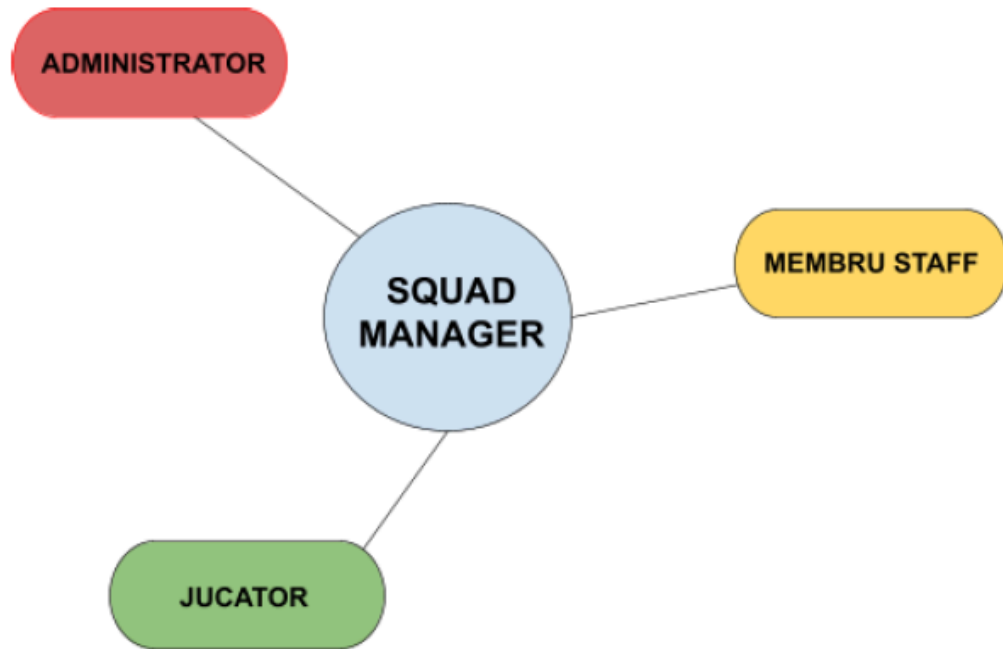
Compromisuri:  
-complexitate  
-posibilitatea de refolosire limitata

#### → React Hooks

Beneficii:  
-cod mai scurt

Compromisuri:  
-complexitate

# USER ROLES



- **ADMINISTRATOR:**
  - are acces deplin la toate functionalitatile din cadrul aplicatiei
  - in pagina Profil are acces la datele sale personale
- **MEMBRU STAFF:**
  - in pagina Stadioane are acces la toate functionalitatile
  - in pagina Meciuri are acces la toate functionalitatile
  - in pagina Jucatori are acces la toate functionalitatile
  - in pagina Staff are acces doar in a vizualiza datele
  - in pagina Contracte nu are acces
  - in pagina Profil are acces la datele sale personale

- JUCATOR:
  - in pagina Stadioane are acces doar in a vizualiza datele
  - in pagina Meciuri are acces doar in a vizualiza datele
  - in pagina Jucatori are acces doar in a vizualiza datele
  - in pagina Staff are acces doar in a vizualiza datele
  - in pagina Contracte nu are acces
  - in pagina Profil are acces la datele sale personale
  
- UTILIZATOR NELOGAT:
  - in pagina Stadioane are acces doar in a vizualiza datele
  - in pagina Meciuri are acces doar in a vizualiza datele
  - in pagina Jucatori are acces doar in a vizualiza datele
  - in pagina Staff are acces doar in a vizualiza datele
  - in pagina Contracte nu are acces

# CONTRIBUTII PERSONALE

## AVIAN SILVIU:

- Configurare Firebase
- Exportare in Excel/Pdf
- Open Weather API
- Code refactoring
- Activity diagram export
- Package diagram

## DIJMARESCU CRISTINA:

- Stadium page
- Adaugare elemente in baza de date
- Configurare entitati Firebase
- Class diagram ER
- Activity diagram adaugare meci

## STAN ANA:

- Player page
- Staff page
- CSS design
- Objects Diagram
- Activity diagram update staff

## IONESCU ALEXANDRU:

- Register
- Login
- Match, Contracts and Statistics pages
- Deployment diagram
- Activity diagram register