

# RAPORT DE PROIECT

## Aplicație Web pentru Managementul Evenimentelor Sportive ("Poli Olympics")

### 1. Introducere

Prezentul raport detaliază etapele de analiză, proiectare și implementare a aplicației web "Poli Olympics", o soluție software destinată digitalizării procesului de organizare a competițiilor sportive. Aplicația a fost dezvoltată utilizând ecosistemul **Java Spring Boot** pentru backend și **Thymeleaf** împreună cu framework-ul **Bootstrap** pentru frontend, având ca obiectiv principal facilitarea gestionării meciurilor, echipelor și a scorurilor într-un mod securizat și eficient.

Motivația alegerii acestei teme provine din necesitatea de a elimina procesele manuale (hârtie/Excel) în organizarea turneelor și de a oferi o transparență totală asupra rezultatelor în timp real.

### A. Specificare funcționalități de bază și suplimentare

Arhitectura aplicației a fost gândită modular, împărțind funcționalitățile în două categorii: cele esențiale funcționării platformei (Core Features) și cele care aduc un plus de valoare și complexitate tehnică (Advanced Features).

#### A.1. Funcționalități de Bază (Core)

ACESTE FUNCȚIONALITĂȚI ASIGURĂ INFRASTRUCTURA MINIMĂ NECESARĂ DESFĂȘURĂRII ACTIVITĂȚII UTILIZATORILOR:

##### 1. Gestionează Conturile de Utilizator (CRUD):

- Posibilitatea de înregistrare a unui utilizator nou folosind o adresă de email și o parolă.
- Stocarea securizată a parolelor în baza de date Microsoft SQL Server prin criptare folosind algoritmul **BCrypt**.
- Autentificarea clasică (Login) pe baza credențialelor stocate.

##### 2. Vizualizarea Meciurilor și a Clasamentului:

- Orice utilizator autentificat are acces la lista completă a meciurilor programate.

- Vizualizarea scorurilor actualizate pentru meciurile încheiate sau în desfășurare.

### 3. Gestiunea Echipelor (Read-Only pentru useri):

- Afisarea listei de echipe participante, a siglelor acestora și a detaliilor generale.

## A.2. Funcționalități Suplimentare și Avansate

Acestea reprezintă elementele de inovație tehnică ale proiectului:

### 1. Autentificare prin OAuth2 (Social Login):

- Integrarea cu **Google Identity Platform** și **Facebook Login (Meta for Developers)**. Aceasta permite utilizatorilor să se autentifice cu un singur click, fără a mai fi necesară crearea manuală a unui cont.
- Sistemul preia automat datele publice (Nume, Email, Poză de profil) și creează o intrare în baza de date locală, mapând utilizatorul social la un cont intern.

### 2. Logică de Turneu Knockout (Eliminatoriu):

- Implementarea unui mecanism automat de eliminare. Administratorul are opțiunea, în momentul introducerii scorului, să marcheze echipa pierzătoare ca fiind "Eliminată".
- **Filtrare Inteligentă:** Sistemul previne erorile umane; atunci când se creează un meci nou, liste de selecție ("dropdown") afișează *doar* echipele care nu au fost eliminate anterior.

### 3. Panou de Administrare Complex:

- Interfață dedicată administratorilor, separată de cea a utilizatorilor obișnuiți, protejată prin Spring Security (Role-Based Access Control).
- Posibilitatea de a adăuga, edita și șterge meciuri și echipe.

---

## B. Descrierea claselor, metodelor și atributelor proiectului

Aplicația respectă arhitectura **MVC (Model-View-Controller)** specifică Spring Boot, asigurând o separare clară a logicii de business de interfața cu utilizatorul și de stratul de persistență a datelor.

### B.0. Platforme și medii de lucru (Tehnologii folosite)

Demonstrarea funcționalităților și implementarea soluției s-au realizat utilizând următoarea stivă tehnologică, optimizată pentru mediul **Windows**:

- **IntelliJ IDEA CE:** Mediul de dezvoltare principal (IDE) folosit pentru scrierea codului Java și gestionarea proiectului.
- **Maven:** Instrumentul de automatizare a build-ului, folosit pentru gestionarea dependențelor (Spring Security, JPA, SQL Driver).
- **H2 Console / SQL Server:** Utilizate pentru vizualizarea și interogarea directă a tabelelor bazei de date.
- **POSTMAN:** Unealtă esențială pentru depanarea și testarea endpoint-urilor de tip REST API, asigurând că backend-ul returnează datele corecte înainte de integrarea cu frontend-ul.
- **Arhitectura MVC (Model-View-Controller):** Soluția este structurată riguros pe acest şablon pentru a separa logica de date (Model), interfața utilizator (View) și fluxul de control (Controller).

## B.1. Stratul de Model (Entities)

Clasele care mapează tabelele din baza de date SQL Server.

- **Clasa UserAccount:**
  - *Atribute:* Long id (PK), String username, String email, String password (hash), String role (ADMIN/USER), String provider (LOCAL/GOOGLE/FACEBOOK), String imagePath.
  - *Rol:* Reprezintă utilizatorul sistemului. Atributul provider este esențial pentru a distinge între userii creați manual și cei veniți prin OAuth2.
- **Clasa Team (Echipă):**
  - *Atribute:* Long id, String numeEchipa, String sport, boolean eliminated.
  - *Atributul Cheie:* eliminated (tip BIT în SQL) este "flag-ul" care dictează dacă echipa mai poate fi selectată pentru meciuri viitoare.
- **Clasa Meci:**

- *Atribute*: Long id, Team echipaAcasa (ManyToOne), Team echipaDeplasare (ManyToOne), Integer scorAcasa, Integer scorDeplasare, LocalDateTime dataMeciului.
- *Rol*: Leagă două instanțe ale clasei Team și stochează rezultatul confruntării.

## B.2. Stratul de Repository (Data Access)

Interfețe care extind JpaRepository pentru manipularea datelor.

- **TeamRepository:**
  - Metoda List<Team> findByEliminatedFalse(): O interogare derivată critică ce returnează doar echipele active, folosită la popularea formularului de creare meci.
- **UserRepository:**
  - Metoda Optional<UserAccount> findByEmail(String email): Folosită atât la login-ul clasic, cât și la cel prin social media pentru a verifica existența utilizatorului.

## B.3. Stratul de Controller și Securitate

- **AdminController:**
  - Gestioneză rutele protejate (/admin/\*\*).
  - Metoda updateMatchScore(...): Primește scorul și parametrii eliminaGazdele/eliminaOaspetii. Apelează logică de business pentru a seta team.setEliminated(true).
- **SecurityConfig:**
  - Clasa de configurare unde se definește SecurityFilterChain.
  - Aici sunt configurate rutele pentru OAuth2: .oauth2Login().userInfoEndpoint(...).
- **FacebookLoginSuccessHandler & GoogleLoginSuccessHandler:**
  - Clase ce implementează AuthenticationSuccessHandler.
  - Metoda onAuthenticationSuccess(...): Se apelează automat după ce Google/Facebook confirmă identitatea. Extrage email-ul și numele din token, verifică baza de date și, dacă utilizatorul nu există, îl creează "on-the-fly".

## C. Descrierea elementelor funcționale oferite prin interfață grafică

Interfața grafică (GUI) a fost construită punând accent pe **User Experience (UX)** și **Responsiveness**, utilizând biblioteca de stiluri sb-admin-2.css bazată pe Bootstrap 4. Aceasta asigură o afișare corectă atât pe desktop, cât și pe dispozitive mobile.

#### C.0. Realizarea componentei Front-end (Interfața Utilizator):

1. **Integrarea paginilor dinamice cu Thymeleaf:** Utilizarea motorului de template-uri pentru a afișa datele din baza de date direct în HTML (ex: listarea meciurilor sau a echipelor).
2. **Stilizarea interfeței cu Bootstrap:** Implementarea unui design responsiv care să permită utilizarea aplicației pe orice dispozitiv (Desktop/Mobil).
3. **Comunicarea cu Controller-ele Spring:** Crearea formularelor HTML care trimit cereri de tip POST către server (pentru logare sau actualizarea scorurilor).
4. **Integrarea butoanelor de Social Login:** Implementarea părții vizuale a autentificării cu Facebook și Google, asigurând redirectarea corectă a utilizatorului.

#### C.1. Pagina de Autentificare (Login)

Este punctul de intrare în aplicație. Designul este minimalist, centrat pe cardul de login.

- **Elemente funcționale:**
  - Formular clasic (Email/Parolă) cu validare vizuală.
  - **Butoane Sociale:** Două butoane distincte, stilizate în culorile oficiale de brand ("Google Red" și "Facebook Blue"). Acestea conțin logo-urile FontAwesome și redirectionează utilizatorul către rutele /oauth2/authorization/....

#### C.2. Dashboard (Panoul Principal)

După autentificare, utilizatorul este întâmpinat de un Dashboard interactiv.

- **Sidebar (Meniu lateral):** Oferă navigare rapidă către secțiunile "Meciuri", "Echipe", "Profil". Este retractabil pe mobil.
- **Carduri Statistice:** Afisează numărul total de echipe înscrise și meciuri jucate.

#### C.3. Modulul de Administrare Meciuri

Aceasta este cea mai complexă interfață, accesibilă doar administratorilor.

- **Tabelul de Meciuri:** Listă paginată cu toate meciurile. Fiecare rând are buton de "Editare".
  - **Formularul de Adăugare Meci:**
    - Conține dropdown-uri (<select>) populate dinamic prin Thymeleaf.
    - *Logică vizuală:* În aceste liste apar exclusiv echipele neeliminate.
  - **Modalul de Editare Scor & Eliminare:**
    - Permite introducerea scorului numeric.
    - Include două **Checkbox-uri Roșii** (ex: "Elimină Echipa Gazdă"). Bifarea acestora activează logica de knockout în backend.
- 

## D. Aspecte privind testarea aplicației

Asigurarea calității software (QA) a fost realizată printr-o combinație de testare manuală și verificare a logicii de business.

### D.1. Testarea Securității (Security Testing)

- **Scenariul OAuth2:** S-a testat fluxul complet de autentificare. S-a verificat comportamentul aplicației atunci când un utilizator are deja cont cu o adresă de email și încearcă să se logheze ulterior cu Facebook folosind același email (sistemul trebuie să le asocieze corect).
- **Testarea Rolurilor:** S-a verificat tentativa de accesare a rutei /admin/adauga-meci de pe un cont de utilizator standard. Rezultatul așteptat și obținut a fost redirecționarea către o pagină de "Access Denied" (403 Forbidden).

### D.2. Testarea Logicii de Eliminare (Unit & Integration)

- S-a validat scenariul critic:
  1. Echipa A joacă cu Echipa B.
  2. Adminul setează scorul 3-0 și bifează "Elimină Echipa B".
  3. Adminul încearcă să creeze un meci nou.
- 4. **Rezultat Valid:** Echipa B nu mai apare în lista de selecție pentru noul meci.
- S-a verificat persistența datelor în SQL Server, asigurându-se că coloana Is\_Eliminated trece din 0 (false) în 1 (true).

### **D.3. Testarea Interfeței (Cross-Browser)**

Aplicația a fost testată pe Google Chrome, Mozilla Firefox și Microsoft Edge pentru a asigura compatibilitatea stilurilor CSS (sb-admin-2), în special a butoanelor de social login și a tabelelor responsive.

---

## **E. 5 Idei despre cum poate fi îmbunătățită aplicația**

Deși aplicația este funcțională și robustă, există direcții clare de dezvoltare ulterioară pentru a o transforma într-un produs comercial complet:

### **1. Generarea Automată a Diagramelor de Turneu (Tournament Brackets):**

- Implementarea unei vizualizări grafice de tip "copac" (tree view) care să arate parcursul echipelor (optimi, sferturi, semi-finale, finală). Aceasta s-ar genera automat pe baza câștigătorilor din meciurile anterioare.

### **2. Sistem de Notificări prin Email:**

- Integrarea serviciului JavaMailSender pentru a trimite notificări automate utilizatorilor abonați atunci când scorul unui meci favorit se modifică sau când este programat un meci nou important.

### **3. Modul de Pariuri Virtuale (Prediction Game):**

- Adăugarea unei funcționalități de "Gamification" unde utilizatorii pot pronostica rezultatele meciurilor înainte de începere. Se poate implementa un clasament al utilizatorilor (Leaderboard) bazat pe acuratețea predicțiilor.

### **4. API REST pentru Aplicație Mobilă:**

- Expunerea funcționalităților actuale prin RestControllers care să returneze date în format JSON. Acest lucru ar permite dezvoltarea unei aplicații mobile native (Android/iOS) care să consume datele din același backend.

### **5. Export Avansat de Date (Reporting):**

- Implementarea funcționalității de export a clasamentelor și a istoricului de meciuri în formate portabile precum PDF (pentru tipărire oficială) sau Excel (pentru analiză statistică detaliată).
- 

## **Concluzii**

Proiectul "**Poli Olympics**" demonstrează capacitatea de a integra tehnologii moderne (Spring Security, OAuth2, SQL Server) într-o arhitectură coerentă și scalabilă. Prin implementarea sistemului de autentificare hibrid (Local + Social) și a logicii complexe de turneu eliminatoriu, aplicația depășește stadiul unui simplu CRUD, oferind o soluție viabilă pentru managementul competițiilor sportive reale. Utilizarea standardelor industriale în codare și securitate asigură o bază solidă pentru dezvoltări ulterioare.