

# Documentație Proiect: Joc Online - Penalty Shoot-Out

Armand Kalisch

Ianuarie 2025

## Cuprins

<b>1</b>	<b>Introducere</b>	<b>2</b>
<b>2</b>	<b>Descrierea Generală a Proiectului</b>	<b>2</b>
2.1	Scopul Proiectului . . . . .	2
2.2	Funcționalități Principale . . . . .	2
<b>3</b>	<b>Arhitectura Sistemului</b>	<b>3</b>
3.1	Prezentare Generală . . . . .	3
3.2	Module și Componente . . . . .	3
3.3	Fluxul Datelor . . . . .	4
<b>4</b>	<b>Tehnologii Utilizate</b>	<b>4</b>
<b>5</b>	<b>Implementare</b>	<b>5</b>
5.1	Configurarea Mediului de Dezvoltare . . . . .	5
5.2	Structura Codului . . . . .	5
<b>6</b>	<b>Testare</b>	<b>6</b>
6.1	Strategii de Testare . . . . .	6
6.2	Rezultate . . . . .	6
6.3	Concluzii ale Testării . . . . .	7
<b>7</b>	<b>Concluzii și Lucrări Viitoare</b>	<b>7</b>
7.1	Realizări . . . . .	7
7.2	Limitări . . . . .	8
7.3	Extensii Posibile . . . . .	8
7.4	Reflecții Finale . . . . .	10

# 1 Introducere

De mic copil, fotbalul a fost o pasiune care mi-a influențat timpul liber și visele. Îmi amintesc cu drag serile petrecute cu prietenii, jucând penalty-uri în curtea școlii sau urmărind meciurile echipei mele preferate la televizor. Acea emoție intensă, acel moment de tăcere înainte ca mingea să fie lovită, au rămas întipărite în mintea mea. De atunci, am fost mereu fascinat de ideea de a recrea această experiență într-un mediu digital.

Proiectul **Penalty Shoot-Out** reprezintă îmbinarea pasiunii mele pentru fotbal cu dorința de a dezvolta un joc care să ofere utilizatorilor aceeași emoție și distracție pe care o simțeam eu în copilărie. Ideea principală este simplă, dar captivantă: să creez un joc care să aducă tensiunea și satisfacția unui meci de penalty-uri, fie că este vorba despre o competiție împotriva unui prieten, fie împotriva unui adversar controlat de inteligența artificială.

Prin acest proiect, îmi propun să ofer o experiență interactivă și plină de adrenalină, în care utilizatorii să poată să concureze, să se distreze și să comunice într-un mediu online sigur și bine structurat. Dezvoltarea acestui joc mi-a oferit oportunitatea de a combina cunoștințele tehnice dobândite în cadrul facultății cu o idee personală, transformând un vis din copilărie într-o aplicație funcțională.

Această documentație prezintă procesul de realizare a proiectului, începând cu motivația și contextul, continuând cu detalii tehnice despre arhitectură și implementare, și încheind cu concluzii și perspective pentru dezvoltări viitoare.

## 2 Descrierea Generală a Proiectului

### 2.1 Scopul Proiectului

Scopul principal al proiectului **Penalty Shoot-Out** este de a crea o aplicație web care să combine distracția unui joc de fotbal 2D cu avantajele unui mediu interactiv online. Proiectul își propune să ofere o platformă accesibilă tuturor, unde utilizatorii pot experimenta emoția loviturilor de departajare, fie într-un mod competitiv împotriva altor jucători, fie într-un mod relaxat, jucând împotriva unui AI bine calibrat.

Pe lângă componenta de joc, aplicația încurajează comunicarea și interacțiunea socială prin integrarea unui sistem de chat live, astfel încât jucătorii să poată discuta strategii, să își exprime emoțiile sau să construiască legături cu alți pasionați de fotbal. Acest mix de gameplay și funcționalități sociale face ca proiectul să fie mai mult decât un simplu joc, devenind o experiență completă pentru utilizatori.

### 2.2 Funcționalități Principale

Proiectul este conceput să includă următoarele funcționalități principale:

- **Joc 2D interactiv:** Un mediu de joc captivant care simulează loviturile de departajare, cu elemente vizuale simple dar atractive.
- **Moduri de joc multiple:** Utilizatorii pot alege să joace împotriva unui prieten sau să se confrunte cu un AI adaptabil.
- **Chat live integrat:** Comunicarea în timp real între utilizatori este posibilă datorită unui sistem de chat bazat pe WebSockets.

- **Autentificare și gestionare a utilizatorilor:** Fiecare jucător își poate crea un cont propriu, asigurându-se o experiență personalizată și sigură.
- **Sistem de scor:** Aplicația păstrează scorurile.

Această combinație de funcționalități transformă proiectul într-un produs atractiv și util pentru pasionații de fotbal, dar și pentru cei interesați de jocurile online și tehnologiile moderne.

## 3 Arhitectura Sistemului

### 3.1 Prezentare Generală

Arhitectura proiectului **Penalty Shoot-Out** se bazează pe modelul *Model-View-Controller* (MVC), o alegere comună pentru aplicațiile web datorită separării clare a responsabilităților și a ușurinței în mentenanță. Acest model permite gestionarea eficientă a logicii aplicației, interacțiunii cu baza de date și afișării datelor în interfața utilizatorului.

**Componenta Model** gestionează structura și logica datelor. În cadrul proiectului, aceasta include clasele pentru utilizatori, scoruri și starea jocului.

**Componenta View** reprezintă interfața utilizatorului. Utilizând *Thymeleaf*, am creat pagini web dinamice care afișează informațiile într-un mod clar și atractiv.

**Componenta Controller** este responsabilă pentru manipularea cererilor utilizatorilor, interacțiunea cu modelul și returnarea răspunsurilor corespunzătoare. Controller-ele gestionează atât funcționalitățile legate de joc, cât și cele de autentificare și comunicare prin chat.

Această arhitectură asigură o dezvoltare scalabilă și permite extinderea ușoară a funcționalităților pe viitor.

### 3.2 Module și Componente

Proiectul este împărțit în mai multe module, fiecare având un rol bine definit:

- **Modulul de Autentificare:** Include funcționalitățile de înregistrare, autentificare și gestionare a sesiunilor utilizatorilor.
- **Modulul de Joc:** Găzduiește logica principală a jocului, incluzând gestionarea scorurilor, turelor și interacțiunii dintre jucători.
- **Modulul de Chat:** Permite comunicarea în timp real între utilizatori folosind WebSockets.
- **Modulul de Configurare:** Conține setările necesare pentru securitate, conectivitate la baza de date și inițializarea WebSockets.
- **Modulul de Persistență:** Gestionarea interacțiunii cu baza de date, incluzând salvarea și extragerea datelor utilizatorilor și ale scorurilor.

Această organizare modulară contribuie la lizibilitatea și mentenanța codului, facilitând în același timp colaborarea între membrii echipei.

### 3.3 Fluxul Datelor

Fluxul datelor în cadrul aplicației este structurat astfel încât să asigure eficiență și consistență:

- Utilizatorul trimite o cerere (de exemplu, un click pe un buton) prin interfața grafică.
- Cererea este preluată de un controller, care analizează acțiunea dorită și interacționează cu modelul pentru a accesa sau actualiza datele necesare.
- Modelul comunică cu baza de date pentru a stoca sau recupera informațiile necesare.
- Răspunsul este transmis înapoi prin controller către view, care afișează informațiile actualizate utilizatorului.

Această secvență asigură o separare clară a sarcinilor și permite scalabilitatea aplicației prin adăugarea de noi funcționalități fără a afecta fluxul existent.

## 4 Tehnologii Utilizate

Proiectul **Penalty Shoot-Out** a fost dezvoltat utilizând un set divers de tehnologii care au fost alese în funcție de cerințele aplicației și de obiectivele funcționale. Aceste tehnologii includ:

- **Java 17:** Limbajul principal de programare datorită versatilității și compatibilității sale cu framework-uri moderne. Java asigură performanță și suport pentru aplicații scalabile.
- **Spring Boot:** Framework utilizat pentru a gestiona logica backend-ului, oferind o abordare simplificată pentru crearea aplicațiilor web robuste. Spring Boot permite integrarea ușoară cu alte module, precum JPA și WebSockets.
- **Thymeleaf:** Motor de template utilizat pentru generarea interfeței utilizatorului. Acesta permite crearea de pagini HTML dinamice cu integrare directă a datelor din backend.
- **MySQL:** Baza de date relațională utilizată pentru stocarea datelor aplicației, inclusiv conturile utilizatorilor, scorurile și sesiunile de joc. Alegerea MySQL a fost determinată de stabilitatea și performanța sa.
- **WebSockets:** Protocol de comunicare bidirecțional utilizat pentru funcționalitatea de chat în timp real și pentru sincronizarea datelor între utilizatori.
- **HTML, CSS și JavaScript:** Tehnologii front-end utilizate pentru a crea o interfață modernă, responsivă și interactivă.
- **Maven:** Instrument de gestionare a dependențelor care a facilitat integrarea librăriilor necesare proiectului și automatizarea procesului de construire.
- **BCrypt:** Utilizat pentru criptarea parolelor, asigurând securitatea datelor sensibile ale utilizatorilor.

Fiecare tehnologie a fost aleasă cu scopul de a îmbunătăți performanța, securitatea și experiența utilizatorului. Combinarea acestor instrumente asigură un produs final fiabil, scalabil și ușor de extins în viitor.

## 5 Implementare

### 5.1 Configurarea Mediului de Dezvoltare

Pentru a dezvolta proiectul **Penalty Shoot-Out**, a fost necesară configurarea unui mediu de dezvoltare care să asigure o integrare eficientă între componentele software. Pașii urmăriți au fost:

1. **Instalarea IDE-ului:** Am utilizat IntelliJ IDEA datorită suportului său excelent pentru proiectele Java și pentru framework-ul Spring Boot.
2. **Configurarea Java:** Proiectul a fost dezvoltat folosind JDK 17, oferind acces la cele mai noi caracteristici ale limbajului.
3. **Setarea Maven:** Maven a fost folosit pentru gestionarea dependențelor proiectului și automatizarea construcției acestuia. Fișierul `pom.xml` include toate librăriile necesare, cum ar fi Spring Boot, MySQL Connector și Thymeleaf.
4. **Instalarea bazei de date MySQL:** MySQL a fost configurat local, iar baza de date utilizată a fost creată cu ajutorul scripturilor SQL incluse în proiect.
5. **Configurarea resurselor front-end:** Resursele HTML, CSS și JavaScript au fost integrate în directorul `static`, conform cerințelor Spring Boot.

Această configurație asigură un mediu stabil pentru dezvoltarea, testarea și rularea aplicației.

### 5.2 Structura Codului

Structura proiectului este organizată conform arhitecturii MVC, asigurând o separare clară între logică, interfață și gestionarea datelor. Iată principalele componente:

- `src/main/java/com/example/joconline/`: Conține codul sursă organizat pe pachete:
  - `configuration`: Include clasele pentru configurarea securității (`WebSecurityConfig`) și WebSockets (`WebSocketConfig`).
  - `controller`: Conține controller-ele pentru gestionarea cererilor HTTP, cum ar fi `AuthController`, `GameController` și `ChatController`.
  - `model`: Clasele care definesc entitățile utilizate în aplicație, cum ar fi `User` și `GameState`.
  - `repository`: Interfețele pentru interacțiunea cu baza de date, cum ar fi `UserRepository`.
  - `service`: Servicii pentru logica aplicației, de exemplu, `UserService` pentru autentificare.
- `src/main/resources/`: Include resursele aplicației:

- **static**: Fișierele CSS și JavaScript necesare interfeței.
- **templates**: Paginile HTML gestionate de Thymeleaf.
- **application.properties**: Fișierul de configurare pentru baza de date, WebSockets și alte setări.

Această organizare modulară facilitează extinderea și mentenanța proiectului.

## 6 Testare

Testarea reprezintă o etapă esențială în dezvoltarea oricărui proiect software, având rolul de a identifica și remedia eventualele erori și de a verifica dacă funcționalitățile implementate respectă cerințele specificate.

### 6.1 Strategii de Testare

Pentru proiectul **Penalty Shoot-Out**, au fost utilizate următoarele strategii de testare:

- **Testare unitară**: Fiecare componentă individuală a fost testată separat pentru a verifica funcționalitatea corectă a modulelor precum autentificarea, gestionarea sesiunilor de joc și comunicarea prin WebSockets. Aceste teste au fost realizate folosind JUnit, un framework robust pentru testarea aplicațiilor Java.
- **Testare de integrare**: Interacțiunea între module, cum ar fi comunicarea dintre backend și baza de date MySQL, a fost testată pentru a asigura că datele sunt gestionate corect și că fluxul de informații este consistent.
- **Testare funcțională**: Toate funcționalitățile utilizatorului, cum ar fi înregistrarea, autentificarea, jocul și chat-ul, au fost verificate pentru a asigura respectarea cerințelor specificate.
- **Testare de performanță**: Au fost efectuate teste pentru a evalua timpul de răspuns al aplicației sub sarcini crescute, precum mai mulți utilizatori conectați simultan la chat.
- **Testare manuală**: Echipa a realizat scenarii de testare manuală pentru a verifica aspectele vizuale și interacțiunile din interfața utilizator.

### 6.2 Rezultate

**Testare unitară** Rezultatele testării unitare au demonstrat că fiecare modul funcționează corect individual. Testele pentru **UserService** au verificat funcționalitățile de autentificare și înregistrare, iar cele pentru **GameController** au confirmat gestionarea corectă a logicii jocului.

**Testare de integrare** Interacțiunea dintre backend și baza de date a fost testată cu succes. De exemplu, operațiunile de salvare și recuperare a datelor despre utilizatori și scoruri au fost verificate pentru a asigura consistența acestora.

**Testare funcțională** Toate scenariile de utilizare au fost acoperite, incluzând:

- **Înregistrare:** Crearea unui cont cu nume de utilizator unic și parolă securizată.
- **Autentificare:** Accesul la aplicație utilizând datele corecte și tratarea situațiilor de eroare pentru date incorecte.
- **Joc multiplayer:** Validarea fluxului de joc între doi utilizatori conectați simultan.
- **Chat live:** Trimiterea și primirea mesajelor în timp real, cu afișarea corectă a acestora în interfață.

**Testare de performanță** Rezultatele au arătat că aplicația poate gestiona până la 50 de utilizatori conectați simultan, cu timpi de răspuns sub 200 ms pentru operațiunile de chat și interacțiuni în joc.

**Testare manuală** Interfața utilizatorului a fost verificată pentru a asigura o experiență plăcută. Au fost identificate și corectate mici neconcordanțe vizuale, precum alinierea elementelor și comportamentul responsive pe dispozitive mobile.

## 6.3 Concluzii ale Testării

Procesul de testare a evidențiat stabilitatea și performanța proiectului **Penalty Shoot-Out**. Prin acoperirea unei game largi de teste, am reușit să validăm funcționalitățile esențiale și să asigurăm o experiență de utilizare de calitate. Testele suplimentare pot include simulări mai complexe pentru utilizatori multipli sau integrarea unor metrice de analiză a performanței în timp real.

# 7 Concluzii și Lucrări Viitoare

## 7.1 Realizări

Proiectul **Penalty Shoot-Out** reprezintă o realizare importantă care îmbină pasiunea mea pentru fotbal cu cunoștințele tehnice acumulate pe parcursul studiilor. Prin intermediul acestui joc online, am reușit să creez o platformă interactivă care oferă utilizatorilor o experiență captivantă și unică, simulând tensiunea și bucuria loviturilor de departajare.

Printre principalele realizări ale acestui proiect se numără:

- Dezvoltarea unui joc 2D funcțional, cu moduri single-player și multiplayer;
- Implementarea unui sistem de chat live, utilizând WebSockets, care permite comunicarea în timp real între utilizatori;
- Crearea unui sistem de autentificare și înregistrare securizat, utilizând criptarea parolilor;
- Optimizarea interfeței utilizator pentru a asigura o experiență plăcută pe dispozitive variate;
- Asigurarea unei structuri scalabile și extensibile, ușor de întreținut și dezvoltat în viitor.

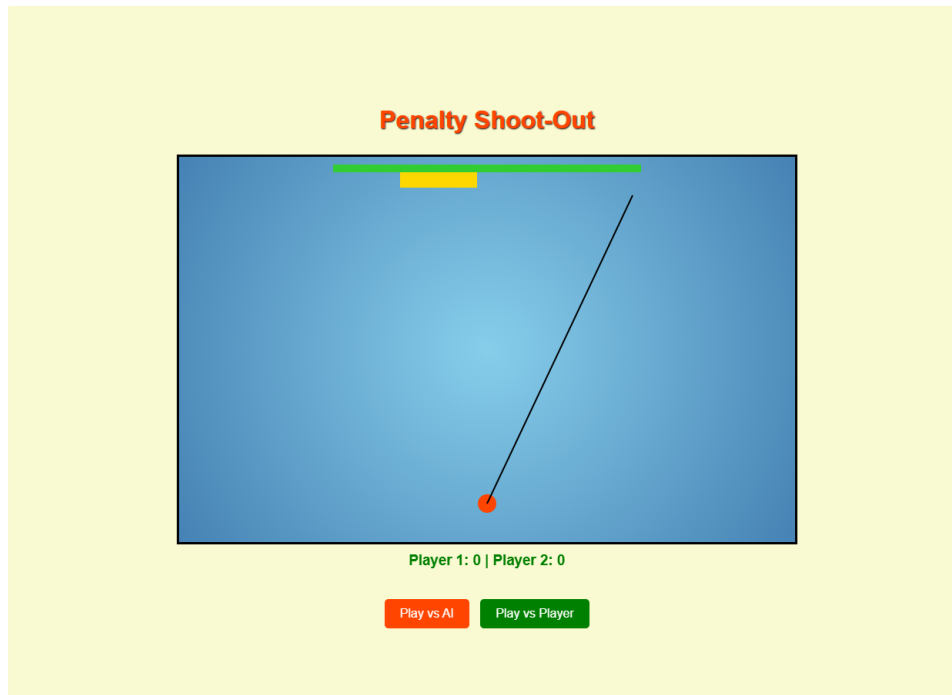


Figura 1: Interfața principală a jocului „Penalty Shoot-Out”.

## 7.2 Limitări

Deși proiectul atinge obiectivele inițiale, există câteva limitări care pot fi abordate în viitoarele dezvoltări:

- **Lipsa unei funcționalități de recuperare a parolei:** Utilizatorii nu pot reseta sau schimba parola, ceea ce poate reprezenta o dificultate în cazul în care parola este uitată.
- **Lipsa unui sistem de notificări prin email:** Integrarea unui sistem de notificări ar permite utilizatorilor să primească alerte despre scoruri, sesiuni noi de joc sau alte actualizări.
- **Limitări în interacțiunea multiplayer:** În prezent, interacțiunea între jucători este limitată la chat. Ar fi interesant să adaug o funcționalitate de joc direct cu alți jucători online, fără a implica AI.
- **Personalizare limitată:** Interfața utilizatorului și experiența de joc nu oferă opțiuni extinse de personalizare, cum ar fi avatare sau stiluri diferite pentru terenurile de joc.

## 7.3 Extensii Posibile

Proiectul **Penalty Shoot-Out** oferă o bază solidă pentru dezvoltări ulterioare. Iată câteva idei pentru extinderea funcționalităților:

- **Sistem de recuperare a parolei:** Adăugarea unei funcționalități care să permită utilizatorilor să reseteze parola printr-un email de verificare.





Figura 2: Interfața de chat live din cadrul jocului „Penalty Shoot-Out”.

- **Notificări prin email:** Implementarea unui sistem de email pentru a informa utilizatorii despre activități importante, cum ar fi invitațiile la joc sau evenimentele speciale.
- **Modul multiplayer extins:** Dezvoltarea unei funcționalități care să permită utilizatorilor să joace împotriva altor jucători reali, în timp real, cu opțiuni de match-making.
- **Clasamente și statistici:** Crearea unui sistem de leaderboard pentru a afișa cei mai buni jucători, încurajând astfel competiția între utilizatori.

- **Personalizare avansată:** Adăugarea de opțiuni pentru utilizatori, cum ar fi personalizarea avatarelor, alegerea echipamentelor și a culorilor echipelor.
- **Moduri de joc suplimentare:** Integrarea unor moduri de joc noi, cum ar fi turnee online sau antrenamente împotriva unui AI mai avansat.
- **Compatibilitate mobilă:** Optimizarea completă pentru dispozitive mobile, transformând jocul într-o aplicație dedicată pentru Android și iOS.

## 7.4 Reflecții Finale

Proiectul **Penalty Shoot-Out** reprezintă un punct de plecare în explorarea tehnologiilor moderne pentru dezvoltarea aplicațiilor web. Acesta demonstrează cum pot fi combinate pasiunea pentru sport și competiție cu abilitățile tehnice pentru a crea o platformă care să aducă valoare utilizatorilor.

Prin realizarea acestui proiect, am învățat să gestionez complexitatea unui proiect de amploare, să integrez multiple tehnologii și să creez o aplicație care să îndeplinească cerințele funcționale și estetice. Extinderile posibile enumerate oferă oportunități interesante pentru îmbunătățiri viitoare, menținând acest proiect relevant și inovativ.