**Chess**

**16.01.2024**

|  |  |  |
| --- | --- | --- |
| **Îndrumător:** |  | **Student:** |
| **dr. ing. Daniel Morariu** |  | **Hoadrea Răzvan-Bucur**  **223/1** |

Istoric Versiuni

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versiune** | **Descriere** | **Autor** |
| 11/12/2023 | 1.0 | Începerea aplicației | Hoadrea Răzvan |
| 22/12/2023 | 1.1 | Terminare modulului de rețea | Hoadrea Răzvan |
| 23/12/2023 | 1.2 | Începerea jocului propriu-zis | Hoadrea Răzvan |
| 26/12/2023 | 1.3 | Terminarea aplicației | Hoadrea Răzvan |

[Istoric Versiuni 1](#_Toc937418138)

[1 Specificarea cerințelor software 2](#_Toc1020124100)

[1.1 Introducere 3](#_Toc4591333)

[1.1.1 Obiective 3](#_Toc787784474)

[1.1.2 Definiții, Acronime și Abrevieri 3](#_Toc1209156602)

[1.1.3 Tehnologiile utilizate 3](#_Toc554272129)

[1.2 Cerințe specifice 3](#_Toc294105119)

[2 Funcționalitate 4](#_Toc2082598682)

[2.1 Descriere 5](#_Toc923423999)

[2.2 Fluxul de evenimente 5](#_Toc384072694)

[2.2.1 Fluxul de bază 5](#_Toc1724699958)

[2.2.2 Fluxuri alternative 5](#_Toc1743260841)

[2.2.2.1 Primul flux alternativ 5](#_Toc1236435604)

[2.2.2.2 Al doilea flux alternativ 5](#_Toc666660031)

[2.2.3 Pre-condiții 5](#_Toc381385531)

[2.2.4 Post-condiții 5](#_Toc634215315)

[3 Funcționalitate 5](#_Toc1974291106)

[3.1 Descriere 6](#_Toc704728270)

[3.2 Fluxul de evenimente 6](#_Toc1464610452)

[3.2.1 Fluxul de bază 6](#_Toc1173821029)

[3.2.2 Fluxuri alternative 6](#_Toc434026207)

[3.2.2.1 Primul flux alternativ 6](#_Toc787062051)

[3.2.2.2 Al doilea flux alternativ 6](#_Toc1166449285)

[3.2.3 Pre-condiții 6](#_Toc1426282298)

[3.2.4 Post-condiții 6](#_Toc340150819)

[4 Implementare 7](#_Toc1679427051)

[4.1 Diagrama de clase 8](#_Toc1710914423)

[4.2 Descriere detaliată 9](#_Toc1477723355)

[5 Bibliografie 10](#_Toc1501923968)

# Specificarea cerințelor software

## Introducere

Proiectul propus este dezvoltarea unui joc de șah online, focalizat pe partide între doi jucători. Scopul principal al proiectului este crearea unei platforme interconectate, unde doi utilizatori pot juca șah în timp real, fiecare conectat la joc de pe propriul său dispozitiv. Jocul va fi conceput pentru a oferi o experiență de șah autentică și captivantă, încurajând competiția și interacțiunea între jucători.

**Conectivitate în rețea:** Implementarea unei arhitecturi de rețea pentru a permite conectarea a doi jucători. Aceasta trebuie să asigure o comunicație stabilă și sigură între dispozitivele participante.

**Interfață grafică:** Dezvoltarea unei interfețe grafice intuitive și atractive pentru a facilita plasarea și mișcarea pieselor pe tablă. Interfața ar trebui să ofere o experiență de utilizare plăcută și să permită jucătorilor să se concentreze pe aspectul strategic al jocului.

**Logică de joc:** Implementarea regulilor șahului pentru a asigura desfășurarea corectă a partidelor. Acest aspect trebuie să includă validarea mutărilor, gestionarea stării tablei de șah și detectarea condițiilor de încheiere a jocului (șah-mat, remiză etc.).

### Obiective

Obiectivele proiectului sunt structurate în jurul dezvoltării unui joc de șah online complet, cu accent pe logica de joc și eficientizarea codului. Iată lista obiectivelor propuse:

**Implementarea logicii de șah:** Crearea motorului de șah care să gestioneze corect regulile și condițiile de joc. Acesta ar trebui să permită plasarea și mutarea pieselor, validarea mișcărilor, gestionarea stării tablei și detectarea finalului de joc.

**Conectivitate în rețea:** Dezvoltarea sistemului de conectivitate pentru a permite doi jucători să se conecteze și să joace în timp real. Implementarea unui protocol de comunicație sigur și eficient pentru transmiterea datelor între dispozitive.

**Interfață grafică (GUI):** Crearea unei interfețe grafice intuitive pentru a facilita plasarea și mișcarea pieselor pe tablă. Interfața ar trebui să ofere o experiență plăcută utilizatorilor și să reflecte corect starea jocului.

**Optimizarea performanțelor:** Eficientizarea codului pentru a asigura o experiență de joc fără întreruperi și o utilizare optimă a resurselor sistemului.

**Testare și depanare:** Realizarea testelor pentru a asigura corectitudinea funcționalităților și a identifica eventuale erori. Depanarea și îmbunătățirea continuă a codului pentru a asigura un produs final stabil și fiabil.

**Documentare:** Crearea unei documentații cuprinzătoare pentru a facilita înțelegerea și dezvoltarea continuă a proiectului.

### Definiții, Acronime și Abrevieri

Regulă pentru denumirea variabilelor/funcțiilor/claselor:

Nume sugestive în limba engleză care descriu precis funcționalitatea sau rolul elementului respectiv, urmărind claritatea și ușurința înțelegerii codului.

### Tehnologiile utilizate

Proiectul a fost creat în mediul de dezvoltare Blend for Visual Studio 2022 Professional folosind limbajul de programare C# utilizând platforma Windows Presentation Foundation (WPF). Pentru gestionarea comunicării în rețea între jucători, am implementat metoda de socket programming, care facilitează conexiunea și schimbul de date în timp real între dispozitivele participante.

În ceea ce privește elementele vizuale ale jocului, am integrat imagini pentru piese și tabla de șah, preluate de pe internet. Cu toate acestea, am adus modificări acestora pentru a le personaliza și a le adapta mai bine cerințelor specifice ale proiectului. Această abordare a permis obținerea unei estetici adaptate conceptului general al aplicației.

## Cerințe specifice

Funcționalități ale aplicației realizate:

**Interfața Grafică:**

* + Afișarea corectă a tablei de șah și a pieselor pe interfața grafică.
  + Implementarea unui sistem interactiv de plasare și mutare a pieselor pe tablă.
  + Design modern și intuitiv , asigurând o experiență plăcută utilizatorilor.
  + Implementarea unei interfețe clare și intuitive pentru a notifica jucătorii cu privire la starea curentă a partidei.

**Conectivitate în Rețea:**

* + Implementarea metodei de socket programming pentru gestionarea conexiunilor între doi jucători.
  + Stabilirea unei comunicări eficiente și în timp real între dispozitivele conectate.
  + Validarea și gestionarea mesajelor transmise între server și clienți în timpul partidelor.

**Logica de Șah:**

* + Dezvoltarea unui motor de șah care implementează regulile corecte ale jocului.
  + Validarea și procesarea corectă a mișcărilor pieselor pe tablă.
  + Detectarea și gestionarea stării de joc, precum șah-mat, remiză sau victorie.

**Sistem de Autentificare:**

* + Crearea unui sistem de autentificare pentru jucători.

**Securitate:**

* + Asigurarea integrității și confidențialității informațiilor în timpul partidelor online.

Aceste funcționalități reprezintă baza aplicației dezvoltate, oferind o experiență completă și autentică a jocului de șah online. În secțiunile următoare, vom detalia două dintre aceste funcționalități, evidențiind aspectele lor tehnice și importanța în cadrul proiectului.

# Funcționalitate

## Descriere

**Plasarea/Mutarea Pieselor pe Hartă**

Funcționalitatea de plasare și mutare a pieselor pe hartă reprezintă coloana vertebrală a experienței de joc în aplicația noastră. Este o funcționalitate esențială într-un joc de șah, oferind utilizatorilor capacitatea de a interacționa direct cu tabla de joc și de a experimenta aspectele strategice ale șahului.

## Fluxul de evenimente

### Fluxul de bază

Înainte de a efectua o mutare, aplicația verifică starea jocului, incluzând situații precum deconectarea adversarului de către server, părăsirea jocului sau renunțarea la acesta. Logica jocului de șah este implementată în clasa principală numită **ChessGame**, care gestionează reguli specifice, cum ar fi rocada mare, rocada mică, promovarea unui pion la sfârșitul tablei, șah mat, șah, regula a 50 de mutări fără captură, precum și verificarea pieselor necesare pentru a câștiga sau pierde meciul. Această clasă este derivă din clasa **Board**, unde sunt plasate piesele pe tabla de șah printr-o matrice de obiecte. Pentru fiecare piesă plasată, se generează pozițiile posibile, iar pentru fiecare poziție posibilă, se adaugă informații despre coliziuni cu alte piese. Astfel, se cunoaște în permanență pozițiile și coliziunile pentru fiecare piesă în parte. După fiecare mutare, se creează o mască cu tabla curentă si se va stabilii starea jocului.

### Fluxuri alternative

#### Primul flux alternativ

Fiecare tip de piesă are propria clasă specifică derivată din clasa de bază **Piece**, care, la rândul ei, derivă din interfața **IPiece**. Pentru generarea pozițiilor și coliziunilor pentru o anumită piesă, se apelează funcția specifică din clasa corespunzătoare piesei respective.

#### Al doilea flux alternativ

După ce un jucător realizează o mutare, adversarul este notificat cu privire la starea jocului prin intermediul decodificării mesajului. În acest context, în funcție de mesajul primit, tabla adversarului este actualizată, apelând funcțiile corespunzătoare pe care și jucătorul care a realizat mutarea le-a apelat anterior.

### Pre-condiții

Utilizatorul nu trebuie să se preocupe de funcționalitatea aplicației, deoarece aceasta este bine testată și toate cazurile posibile au fost tratate. Aplicația nu ar trebui să întâmpine probleme semnificative.

### Post-condiții

După realizarea oricărei mutări sau setări, ambii utilizatori sunt notificați cu privire la starea actuală a jocului. Aplicația asigură o comunicare eficientă între utilizatori, furnizând informații în timp real despre evoluția partidei de șah.

# Funcționalitate

## Descriere

Clasa **Network** este responsabilă pentru gestionarea aspectelor legate de rețea în cadrul aplicatiilor de șah. Ea facilitează comunicarea între server și clienți, permițând trimiterea și recepționarea de mesaje între aceștia. Prin intermediul acestei clase, se stabilește și menține conexiunea între server și clienți, precum și gestionarea evenimentelor specifice rețelei.

## Fluxul de evenimente

### Fluxul de bază

**Configurare Server:** Clasa începe prin a inițializa un obiect **TcpListener** pe adresa IP și portul specificate (IPAddress.Any, 8888).

**Ascultare pentru Clienți:** Se inițiază un fir de ascultare (**listenerThread**) pentru a accepta conexiuni de la clienți. În timpul ascultării, la fiecare conexiune acceptată, se creează un fir de execuție separat (**clientThread**) pentru a gestiona comunicarea cu acel client.

**Comunicare cu Clienții:** La fiecare citire de date de la client, se decodează mesajul primit și se trimite către controller-ul aplicației pentru a fi procesat. Mesajele sunt trimise și primite în format ASCII.

**Deconectare Clienți:** În cazul în care apare o deconectare sau închidere a conexiunii, clientul respectiv este eliminat din lista de clienți conectați.

Variabila **connectedClients** este o instanță a clasei **ConcurrentDictionary<string, TcpClient>**. Acesta este un tip special de colecție în .NET care poate fi accesată și modificată în mod sigur de mai multe fire de execuție în același timp, fără a necesita sincronizare explicită.

### Fluxuri alternative

#### Primul flux alternativ

**Transmitere Mesaj Către Clientul Specific:** Funcția **BroadcastMessage** transmite un mesaj specific către un client identificat prin adresa IP. Mesajul este transmis în format ASCII prin intermediul fluxului clientului.

**Deconectare Clienților:** Funcția **disconnectIPClient** trimite un mesaj de tip "shutdown" către clientul specificat, determinându-l să se deconecteze. Mesajul este trimis prin intermediul fluxului clientului și, ulterior, clientul este eliminat din lista de clienți conectați.

#### Al doilea flux alternativ

**Transmitere Mesaj Tuturor Clienților:**

**BroadcastAll**: Transmite un mesaj către toți clienții conectați, excludând cei care sunt deja implicați într-o partidă de șah (**In Game**).

**BroadcastAllExcept**: Transmite un mesaj către toți clienții conectați, cu excepția clientului specificat prin adresa IP și excluzând cei deja implicați într-o partidă de șah (**In Game**).

### Pre-condiții

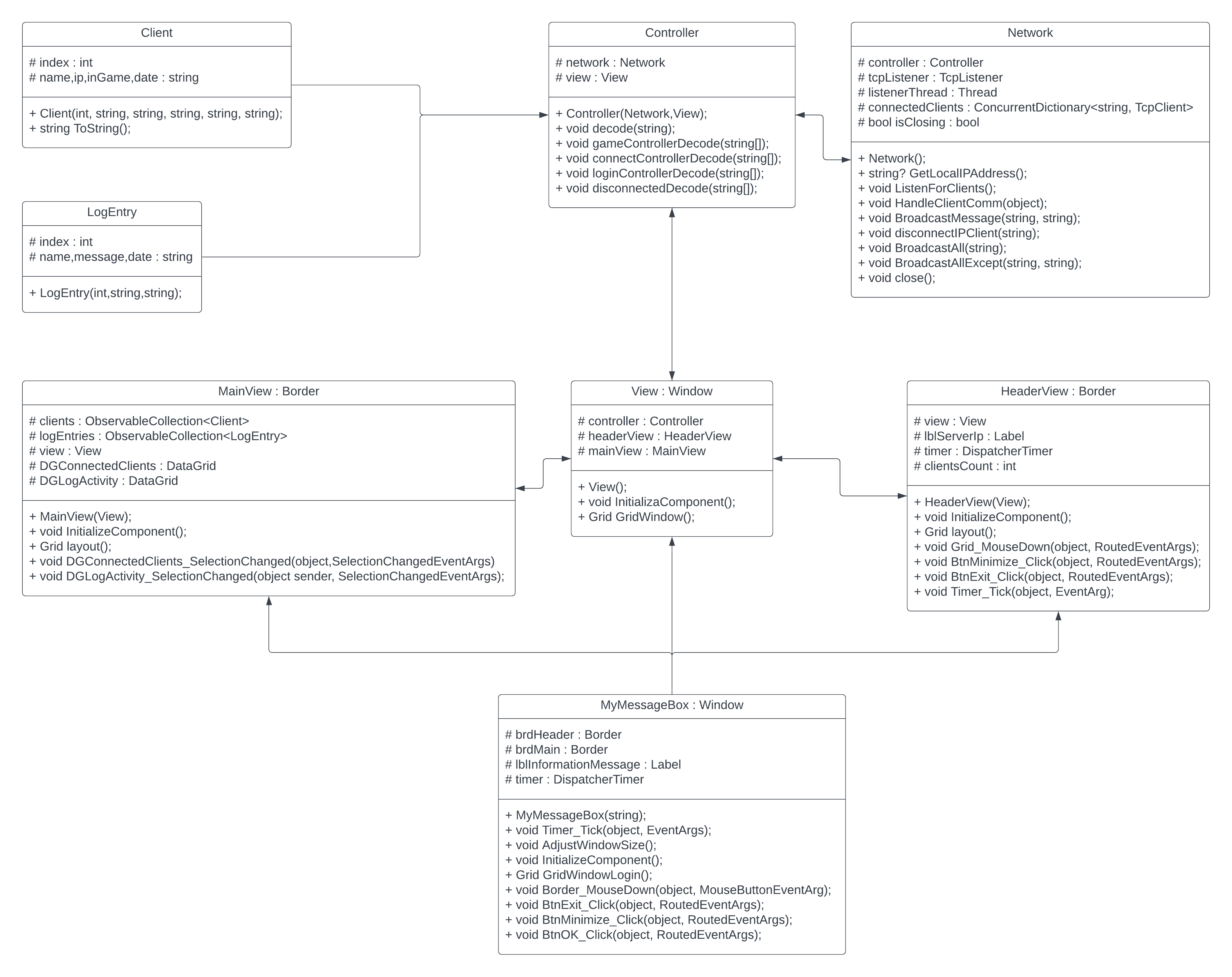
Utilizatorul nu trebuie să aibă grijă de funcționalitatea aplicației la nivel de rețea. Clasa este proiectată și testată pentru a gestiona corect toate cazurile posibile, inclusiv deconectarea sau închiderea conexiunilor.

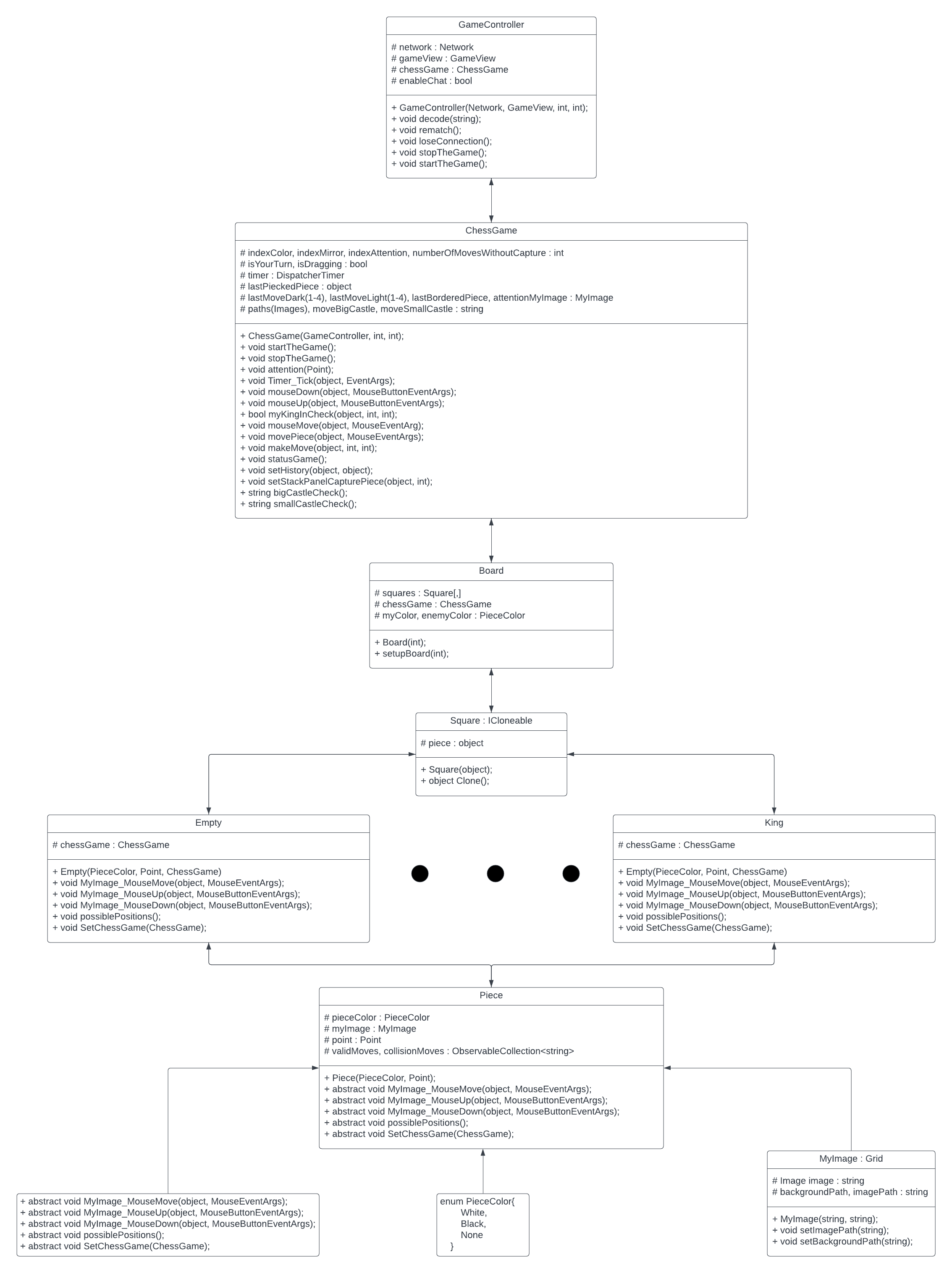
### Post-condiții

Utilizatorul nu trebuie să aibă grijă de funcționalitatea aplicației la nivel de rețea. Clasa este proiectată și testată pentru a gestiona corect toate cazurile posibile, inclusiv deconectarea sau închiderea conexiunilor.

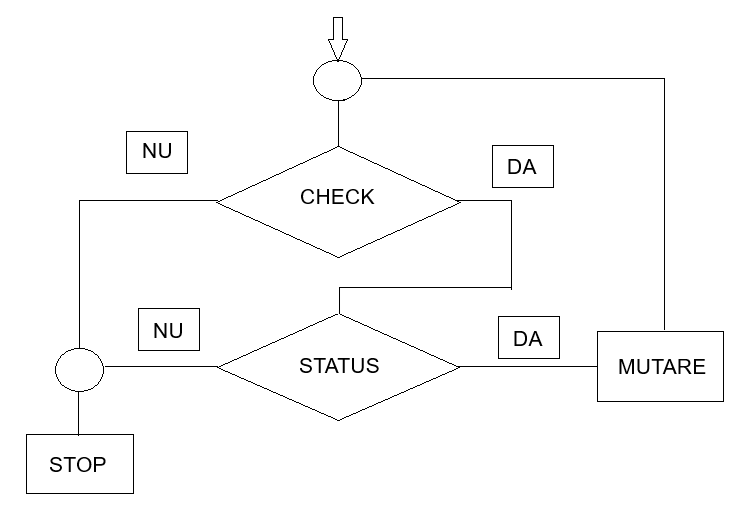
# Implementare

## Diagrama de clase





## Descriere detaliată



# Bibliografie

Soluțiile problemelor mele tehnice le-am găsit pe Stack Overflow: <https://stackoverflow.com/>