

## Documentul de Proiectare a Soluției Aplicatiei Software - *Software Design Document* -



Echipa: Hawks

Membrii:

- Project Manager: Silviu-Teodor Stancioiu, 343C1
- Team Lead: Adrian Stefan, 343C5
- Developer: Bogdan Penisoara, 343C5
- Developer: Catalin Gheorghe, 343C1
- Developer: Vlad Teodor, 342C5
- Tester: Clara Maria Ciarnau, 343C5

### 1. Scopul Documentului

---

Scopul acestui document este de a descrie soluția implementării jocului propus, și anume, “Football Card Game”. Acest document servește rolul unui ghid pentru echipa de dezvoltare. Va face ușoară urmărirea cerințelor aplicației și a mijloacelor prin care ea va fi dezvoltată.

### 2. Obiective

Enunțul cerinței care a stat la baza acestui joc poate fi văzut [aici](#). Obiectivul final este crearea unui joc multiplayer de tip “turn based card-game” (ex: Gwent, Magic: The Gathering, Hearthstone )

Cerințe obligatorii:

- joc multiplayer cu exact 2 jucători
- se folosește o tablă de joc și cărți din lumea fotbalului (jucători de fotbal)
- maximum 3 runde, câștigător fiind cel care câștigă primul 2 runde
- o rundă este câștigată dacă totalul punctelor cartilor de pe teren (atac + apărare) este mai mare decât totalul punctelor adversarului.
- în caz de egalitate (într-o rundă) scorul devine 1-1 de la 0-0 , 2-1 de la 1-0 , 1-2 de la 0-1. Iar dacă scorul deja este 1-1, se da cu banul pentru a determina câștigătorul.
- corectitudinea aplicării acțiunilor
- fiecare jucător vede numai cărțile lui din mână și cele de pe tablă de joc

Tipuri de carti:

- Carti de personaje. (Maxim 500, minim 50) *La inceputul jocului se aleg aleator 24 de carti de personaje din pachet, dintre care jucatorul pastreaza 11.*
- Carti functionale. (Maxim 40, minim 20) *La inceputul jocului se aleg aleator 10 carti functionale, dintre care jucatorul pastreaza 5.*
- Carti lider. (Maxim 40, minim 5) *Se alege de catre jucator la inceputul jocului, este mereu vizibila chiar si de adversar.*

Tabla de joc:

- Consta in 4 linii unde pot fi plasate cartile. Ofensiva, mijloc, defensiva, portar. Fiecare linie ofera bonusuri diferite cartilor. Pot fi plasate oricate carti in fiecare linie, mai putin in linia “portar” unde este permisa doar o carte.

### 3. Continutul Documentului

---

Acest document contine 4 sectiuni esentiale:

- modelul datelor
- modelul arhitectural
- modelul interfetei cu utilizatorul
- elemente de testare

### 4. Modelul Datelor

Jocul nu dispune de o baza de date propriu-zisa. Informatiile despre carti vor fi tinute pe server intr-un format *json*.

Carti de personaje. Vor contine urmatoarele campuri:

- “name” - numele jucatorului
- “type” - pozitia pe care joaca de obicei
- “attack” - puncte de atac
- “defence” - puncte de defensiva
- “club” - clubul din care face parte
- “country” - tara de origine
- “image” - link catre o imagine cu jucatorul (va fi afisata pe carte)

Carti functionale. Vor contine urmatoarele campuri:

- “name” - numele cartii
- “description” - descrierea actiunii

Carti lider. Vor contine urmatoarele campuri:

- “name” - numele antrenorului
- “description1” - descrierea efectului permanent

- “description2” - descrierea efectului activabil

Vor exista si 2 obiecte de tip jucator care vor avea urmatoarele campuri:

- lista cu cartile din mana
- un obiect de tip board

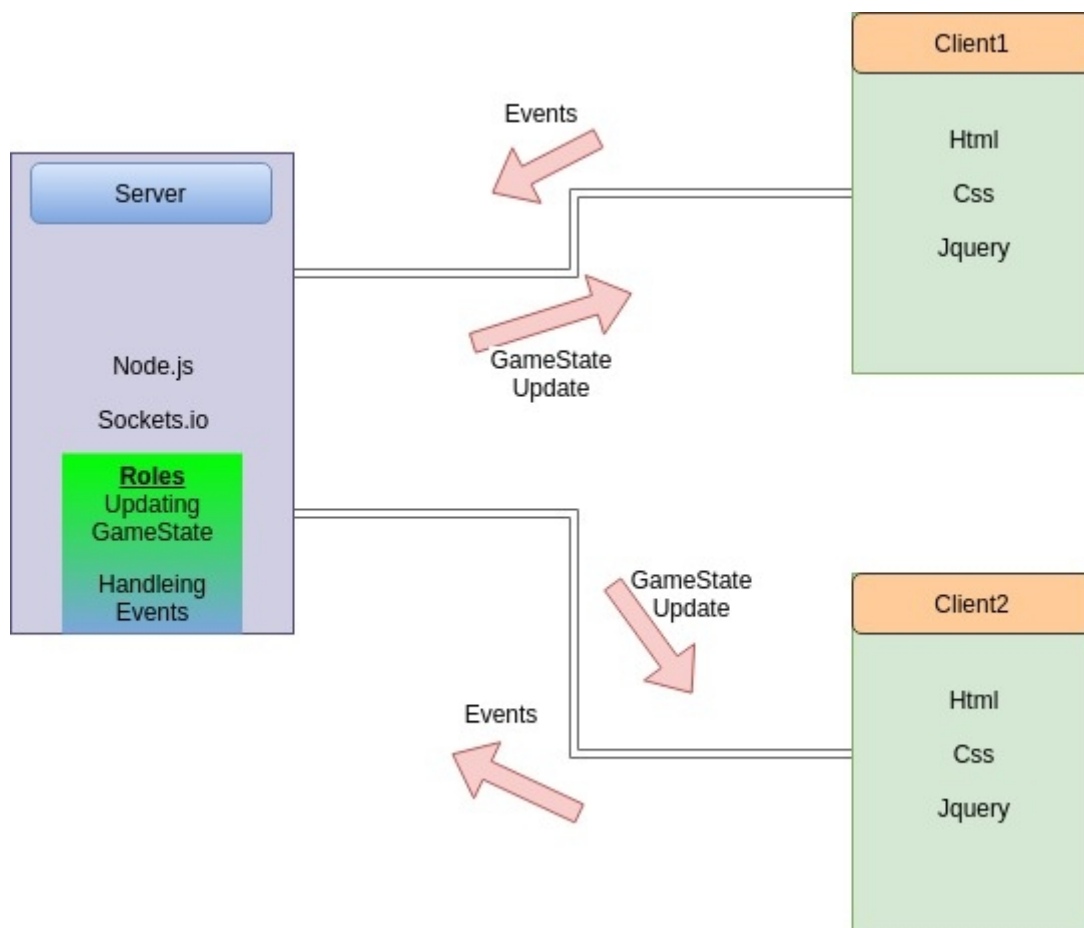
Un obiect de tip board va avea:

- 3 liste pentru fiecare linie a tablei de joc + un loc liber pentru portar
- o carte lider

## 5. Modelul arhitectural

Aplicatia va fi un joc intr-o pagina web. Va avea o arhitectura de tip client-server.

Partea de back-end va fi dezvoltat in javascript prin socket.io si node.js, iar front-endul va fi facut in html si css. Interactiunea server-client facandu-se prin jquery care trimite eventuri la server.



Exista un server tcp care deschide un port prin intermediul runtime environmentului ‘node.js’ si asteapta ca 2 clienti sa se conecteze pentru a putea incepe jocul. Clientii sunt reprezentati de 2 conexiuni tcp initiate printr-un browser web.

Folosind runtime environmentul ‘node.js’ ne asiguram ca nu suntem dependenti de platforma in cazul serverului.

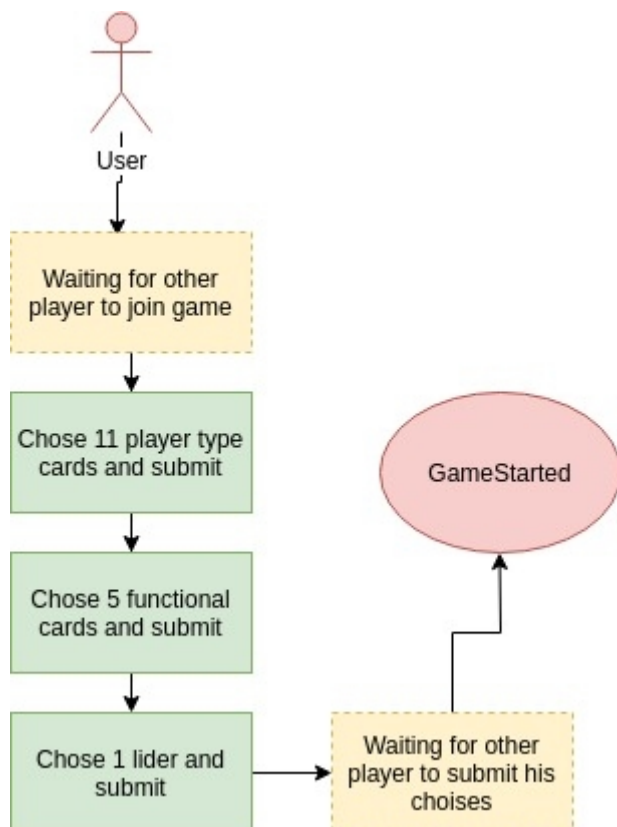
Serverul intervine in validarea mutarilor realizate de clienti si notificarea tuturor clientilor despre starea actuala a jocului, in cazul in care ea se schimba.

## 6. Modelul interfetei cu utilizatorul

Utilizatorul interactioneaza cu jocul prin intermediul unui browser web. Exista pagina de pornire in care se asteapta ca 2 jucatori sa se conecteze la server, iar in acel moment un buton de start-game devine valabil.

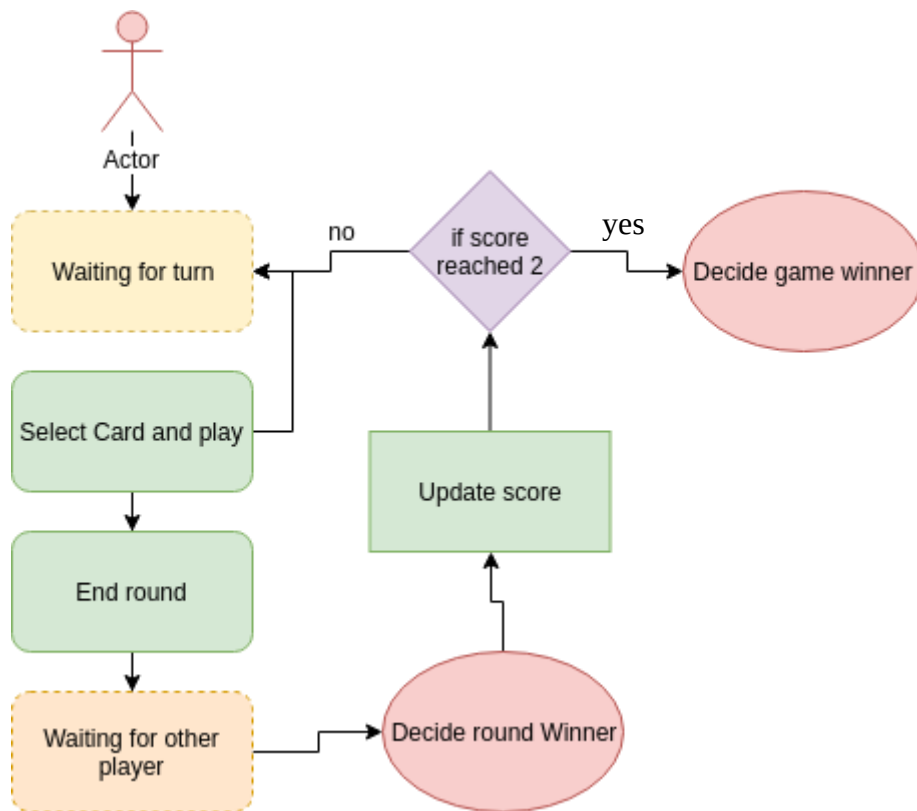
Din aceasta pagina se merge intr-o alta pagina in care trebuie sa se selecteze 11 carti jucator din 22 disponibile. Se confirma alegerea si apoi se aleg alte 5 carti functionale din 10 disponibile, respectiv se alege o carte lider din toate cartile lider existente.

Aceste actiuni se pot urmarii in urmatoarea diagrama.



Dupa inceperea jocului este afisata o tabla cu 4 linii : atac, mijloc, defensiva, portar. Cartile din mana in partea de jos a ecranului si cartea lider in partea stanga.

Prin evenimente de tip click se selecteaza carti si se seteaza in partea dorita a terenului. Cand jucatorul doreste poate apasa pe un buton de 'terminare runda', nu vor mai putea adauga carti pe teren si se va astepta si celalalt jucator pentru ca runda sa fie calculata si a desemna castigatorul.



Este important ca toate informatiile despre carti sa fie usor de accesat si toata interactiunea cu jocul sa fie prin intermediul clickurilor pentru a face jucarea usoara.

## 7.Elementele de testare

Testarea va consta in rularea aplicatiei pentru a vedea daca exista bugguri de logica, afisaj etc. Rezultatele testarii manuale vor fi comunicate dezvoltatorilor si se vor crea tichete pe github pentru rezolvarea problemelor gasite.

Jocul nu necesita resurse computationale intense ale serverului sau a calculatoarelor clientilor. Iar traficul de internet este de asemenea scazut si enevtuala viteza scazuta nu cauzeaza probleme de tip 'lag'.