Document de Proiectare Arhitecturală

„Cooking Crisis”

Doxa Studios

GRAMA NICOLAE
CHIFAN ȘTEFAN-CRISTIAN
VÎȚOGA GEORGE-PATRICK

UNIVERSITATEA „POLITEHNICA” BUCUREȘTI

FACULTATEA DE AUTOMATICA SI CALCULATOARE

APRILIE 2021

CUPRINS

1. Introducere

1.1 Scopul sistemului

Un factor important in dezvoltarea acestei aplicații o reprezintă crearea unui sistem ușor de depanat si extins (atât ca si cod, pentru a putea adăuga noi funcționalități pe viitor, dar si din punct de vedere al scalabilitații sistemului). Este foarte important ca sistemul sa fie testat suficient in fazele inițiale ale dezvoltării.

1.2 Lista de definiții si abrevieri

- Frontend - serviciu ce livrează și este responsabil pentru interfața utilizatorului.

- Backend - serviciu ce este responsabil de logica dintre utilizatori.

- Aplicație nativa – o aplicație ce este proiectata sa ruleze pe un anumit tip de device (Windows, Linux, Mac, Android, IOS, etc..)

- Anti-cheat – un sistem ce asigura integritatea unei competiții virtuale. In general sunt programe ce se asigura ca participanții nu folosesc alte aplicații ce le-ar da un avantaj nedrept. In cazul acestui proiect se refera la sistemul ce se asigura ca participanții nu trimit date incorecte intenționat (sa mărească puterea unei cărți, sa folosească o carte pe care nu o aveau disponibila, etc..)

1.3 Documente de referința

- Documentul de specificație
- Documentație Vue v3
- Documentație Nodejs
- Documentație Express
- Documentație Mongoose
- Planificarea implementării

2. Obiective de proiectare

Pentru acest proiect putem definii următoarele obiective de proiectare:

1. Frontend – reprezintă partea interactiva a aplicației, expusa utilizatorilor. Aceasta este reprezentata de o pagina web (dar ar putea fi creata si a aplicație „nativa” de PC, IOS sau Android), ce trebuie sa fie atractiva din punct de vedere vizual, ușor de utilizat si performanta (sa nu fie o întârziere foarte mare intre acțiunile utilizatorului si răspunsul aplicației).
2. Backend – partea cea mai complexa aplicației, conținând marea majoritate a logicii sistemului. Se ocupa de transmisia de date intre clienți, servirea de informații din baza de date, asigurarea integrității meciurilor (anti-cheat), si multe altele. Totodată, ea trebuie proiectata astfel încât sa poată fi legata de mai multe tipuri de frontenduri (browser, aplicații native).
3. Database – in cazul acestei aplicații, baza de date nu este inclusa in backend, ci separata. Vom folosi o baza de date de tip document (MongoDB), aflata in cloud (Atlas). Sistemul trebuie sa se asigure ca are întotdeauna acces la date si mai ales, ca acestea sunt corecte.

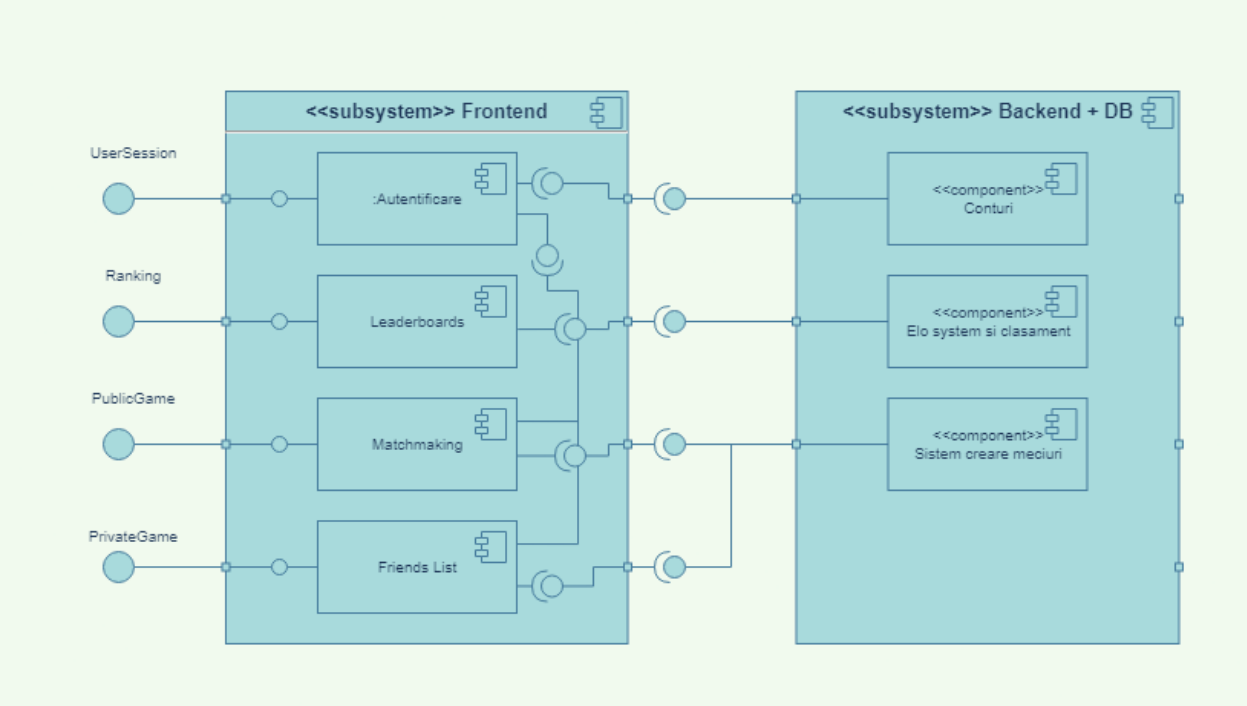
3. Arhitectura propusa

3.1 Prezentarea generala a arhitecturii sistemului

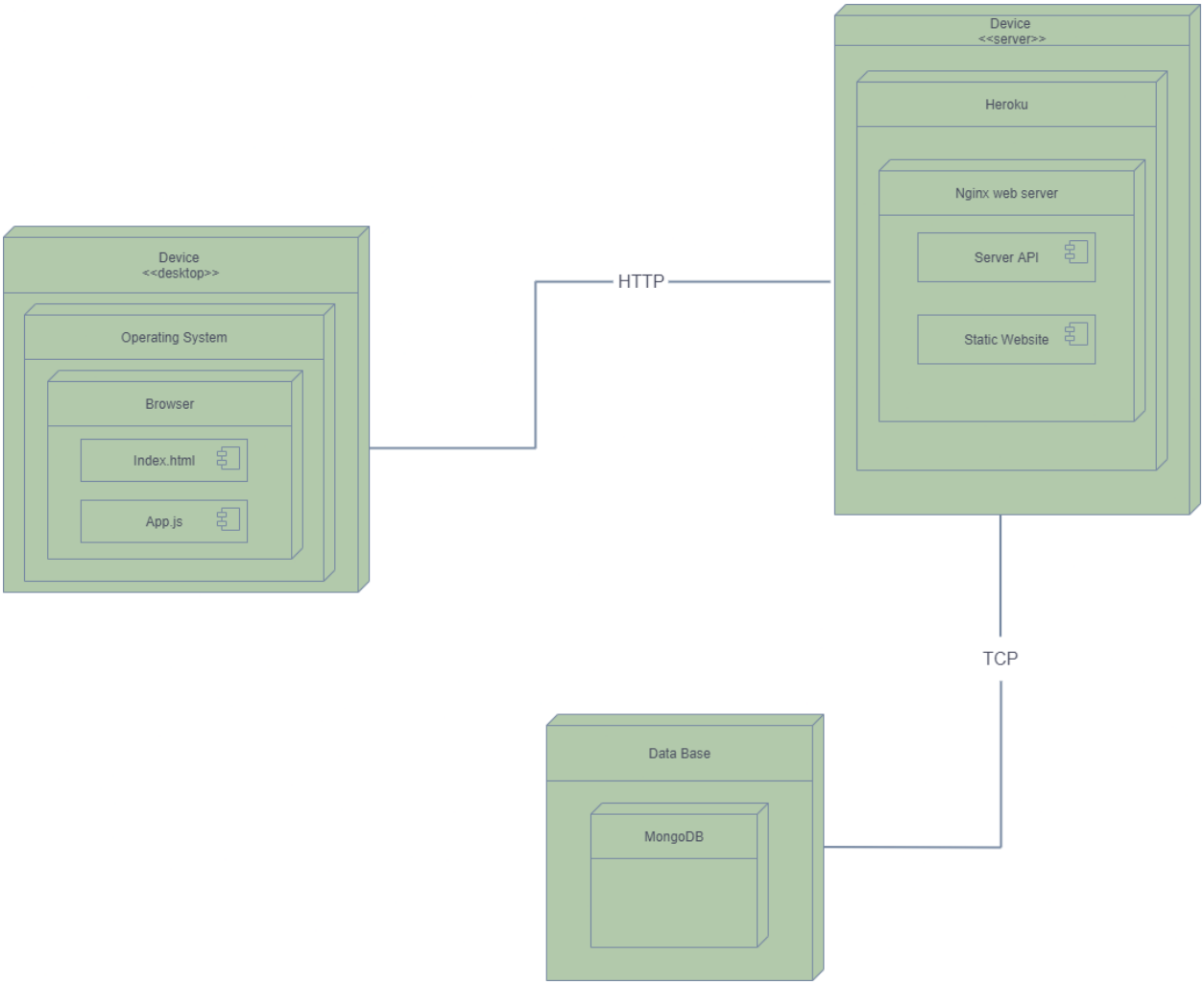
Cum a fost menționat anterior, proiectul „Cooking Crisis” înglobează 3 componente principale: frontend, backend si database (baza de date persistente). Fiecare dintre acestea definește un subsistem.

1. Componenta Frontend: oferă componenta vizuala cu care utilizatorul va interacționa. Se va folosi Vue3 (un framework de javascript) pentru a genera html, javascript si css, ce vor fi servite browserului, ce afișează mai apoi o pagina web.
2. Componenta Backend: asigura funcționarea întregului sistem. Oferă metode de comunicare cu frontendul (REST API, GraphQL, Socket.io), asigura stocarea datelor persistente (MongoDB Atlas + Mongoose). Totodată, preia o parte din „calculele” ce ar fi implementate in frontend pentru a creste securitatea datelor si a îmbunătății viteza clienților.
3. Componenta Database: trebuie sa asigure valabilitatea si siguranța datelor. Fiind un serviciu de cloud (MongoDB Atlas), trebuie doar proiectata structura „documentelor” ce vor fi stocate.

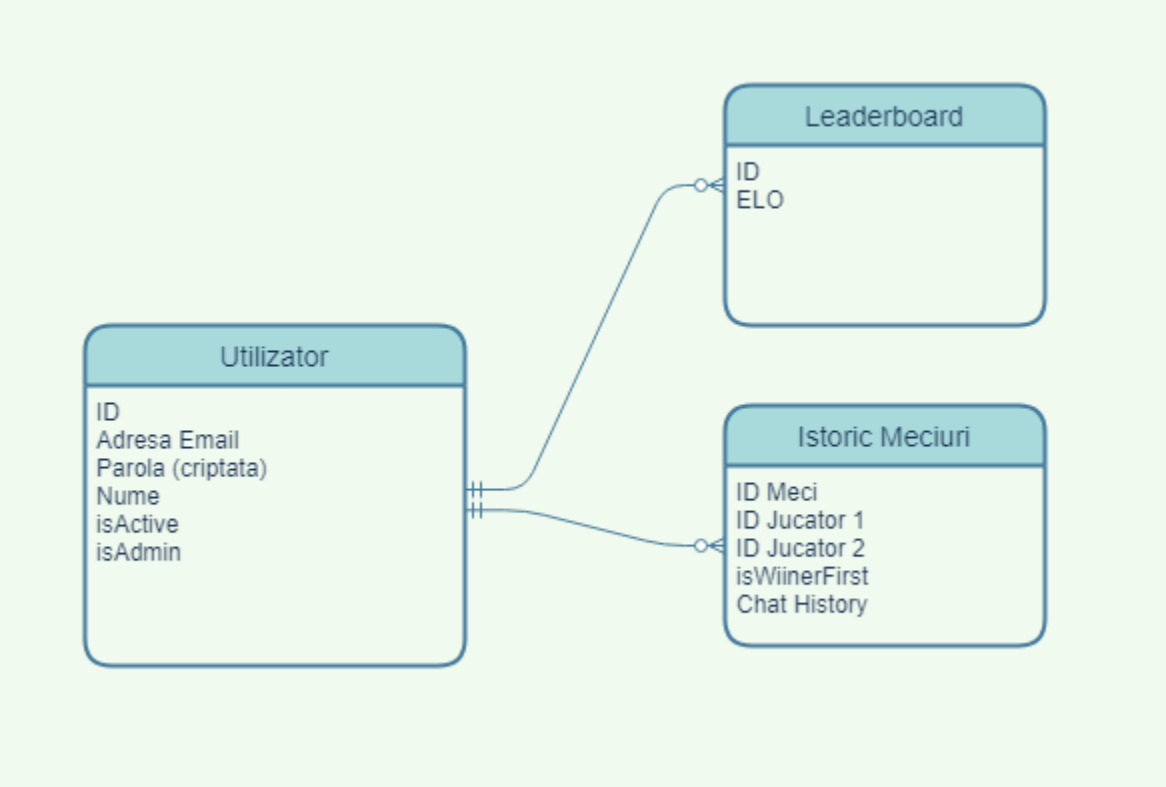
3.2 Diagrama de componente



3.3 Diagrama de distribuție



3.4 Managementul datelor persistente



3.5 Controlul accesului utilizatorilor la sistem

Accesul utilizatorilor pe platforma se face pe baza contului de utilizator. Aceștia trebuie sa se înregistreze mai întâi (jucătorii, adminii primesc conturile direct de la developer).

- Autentificarea se face pe baza numelui de utilizator/adresei de email si a parolei
- Utilizatorii își pot schimba parola si numele de utilizator (daca noul nume nu este deja folosit)

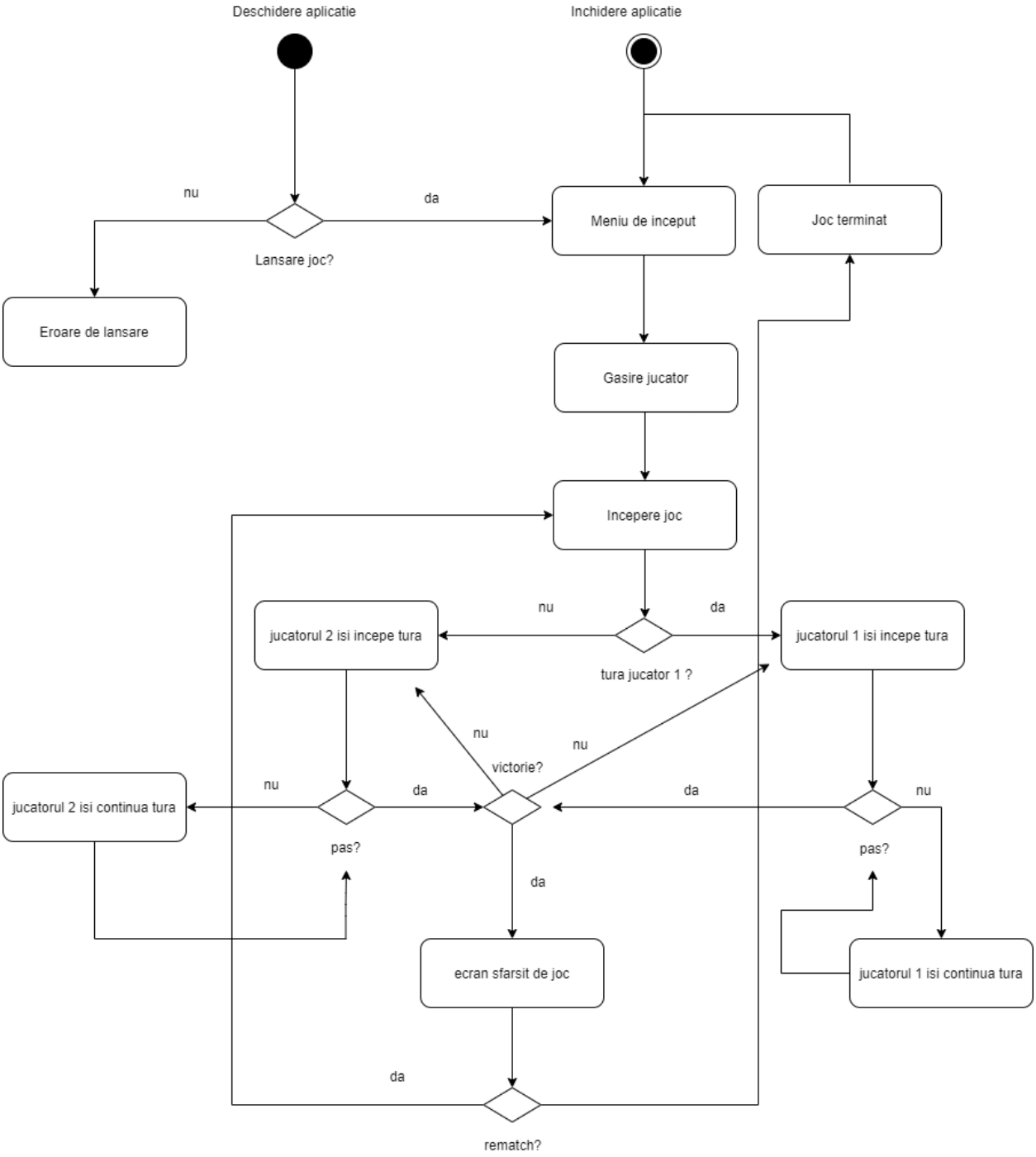
Utilizatorul „jucător”:

- Poate sa afle diferite informații legate de regulile jocului, cărți existente, leaderboard
- Poate participa la meciuri publice (ce ii pot modifica rankul si înregistrate) sau sa organizeze meciuri private, cu prietenii

Utilizatorul „administrator”:

- Acesta asigura respectarea regulilor comunității, verificând raporturi de comportament neadecvat in cadrul meciurilor publice. Acesta poate accesa acel meci (din istoricul de meciuri) si sa verifice chatul. In funcție de severitate, poate sa aplice diferite pedepse (chat ban, account suspension – perioada determinata, account ban – perioada nedeterminata).

3.6 Fluxul global al controlului



3.7 Condițiile limita

Condițiile limita ale sistemului:

- Existenta a multor query-uri in baza de date
- Număr foarte mare de meciuri concurente – supraîncărcarea rețelei
- Probleme cu serviciile cloud