Subiect 7 iulie 2021 - specializarea informatica.

Proiectați și implementați o aplicație client-server pentru următoarea problemă.

Un joc cu 3 jucători numit *Ghicește modelul*. Fiecare jucător propune un model format din cel puțin 6 caractere propuse de server și fiecare jucător încearcă să ghicească modelul propus de ceilalți jucători. Jucătorul/Jucătorii care obține/obțin cele mai multe puncte după 3 runde, câștigă jocul. Fiecare utilizator poate să facă următoarele:

1. *Login*. După autentificarea cu succes se deschide o nouă fereastră în care sunt afișate caracterele propuse de server pentru crearea modelului (ex. {'*', '-','~','^'}), un câmp pentru introducerea modelului propus de jucător (ex. "-^**^-") și un buton "Start joc". Doar după ce trei jucători se autentifică în aplicație și apasă butonul de "Start joc", jocul va începe. La începerea jocului, serverul va trimite tuturor jucătorilor id-urile(username) celorlalți jucători și lungimea modelelor propuse de fiecare.
   Exemplu: (ana, '_ _ _ _ _ _ _ _ '), (ion, '_ _ _ _ _ _ _ '), (vasile, '_ _ _ _ _ _ _ ')
2. *Ghicește modelul.* Fiecare jucător alege unul dintre ceilalți doi jucători, încearcă să ghicească modelul acestuia și trimite propunerea lui serverului. După ce toți jucătorii au trimis propunerile la server, acesta va verifica propunerile primite și va decide punctajul obținut de fiecare jucător, astfel:
   - dacă propunerea primită este corectă, jucătorul va primi un număr de puncte egal cu numărul de caractere din model;
   - dacă propunerea primita are un număr diferit de caractere față de model, jucătorul va primi -2 puncte;
   - dacă în propunere există caractere aflate pe poziția corectă, jucătorul va primi câte 1 punct pentru fiecare caracter aflat pe poziția corectă;
   - dacă în propunere există caractere conținute în model, dar care nu se află pe poziția corectă, jucătorul va primi câte 0.5 puncte pentru fiecare astfel de caracter;

   Exemplu: Pentru mulțimea {'*', '-','~','^'} și modelul "-^**^-", propunerea "^**^-" va primi -2 puncte, propunerea "^~**~^" va primi 3 puncte deoarece cele 2 caractere '*' sunt pe pozițiile corecte (2x1p) și cele 2 caractere '^' sunt în model, dar nu pe pozițiile din propunere (2x0.5p).

   După verificarea propunerilor, serverul trimite tuturor jucătorilor toate modelele în care sunt marcate caracterele ghicite corect (se marchează doar caracterele ghicite pe poziția corectă, ex. _ _* * _ _) și punctajul obținut de fiecare jucător la runda respectivă. Aceste informații vor apărea automat pe interfața grafică a fiecărui jucator.

   Acest pas se repeta de încă 2 ori. La finalul celor 3 runde, serverul va trimite tuturor jucătorilor punctajul obținut de fiecare jucător și clasamentul final. Toți jucătorii vor vedea clasamentul pe interfața grafică (în ordine descrescătoare a numărului de puncte obținut).
3. Un serviciu REST care permite vizualizarea modelelor trimise de fiecare jucător pentru un anumit joc.
4. Un serviciu REST care permite vizualizarea celor 3 propuneri (jucător și modelul propus) făcute de un anumit jucător în timpul unui anumit joc și punctajul obținut de acesta la fiecare rundă.

Observații:

1. Modelele propuse de jucători, propunerile de la fiecare rundă și punctajul obtinut pentru fiecare propunere se păstrează în baza de date.
2. Mulțimea de caractere folosită pentru crearea unui model este aceeași pentru toți jucătorii.

Cerințe:

- Aplicația poate fi dezvoltată în orice limbaj de programare (Java, C#, etc).
- Datele vor fi preluate/salvate dintr-o bază de date relațională.
- Pentru o entitate (exceptând jucator) se va folosi un instrument ORM pentru stocarea datelor (nu se accepta SpringJPA).
- Pentru testarea serviciilor REST se va folosi o extensie a unui browser web/aplicație (Postman, REST client, etc).

Barem:

- Login - 0.5p
- Logout -0.5
- Start joc - 0.5p
- Trimiterea propunerii pentru un anumit model (cu adaugare in baza de date) - 0.5 p
- Determinarea punctajelor pentru o runda - 1p
- Determinarea clasamentului la finalul jocului - 1 p
- Actualizarea automata a ferestrelor (pentru toate cazurile) -1.5p
- Serviciu REST 1 - 1p
- Serviciu REST 2 - 1p
- Folosire ORM (cu executie) -1p
- Structura bazei de date si popularea ei -0.5p
- Arhitectura si proiectare -1 p