Subiect 23 iunie 2021.

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

Un joc cu 3 jucători numit *Cauta obiectele*. Trei utilizatori autentificați pot juca acest joc. Fiecare jucător propune 3 poziții (din 9 posibile) pentru obiectele sale și fiecare jucător încearcă să ghicească pozițiile obiectelor celorlalți jucători. Jucătorul/Jucătorii care obține/obțin cele mai multe puncte după 3 runde, câștigă jocul.

Fiecare jucător poate să facă următoarele:

1. *Login*. După autentificarea cu succes se deschide o nouă fereastră în care sunt afișate trei câmpuri pentru introducerea pozițiilor alese de jucător (numere între 1 si 9) pentru cele 3 obiecte și un buton "Start joc". Doar după ce trei jucători se autentifică în aplicație, introduc pozițiile lor și apasă butonul de "Start joc", jocul va începe. La începerea jocului, serverul va trimite tuturor jucătorilor id-urile/username-urile celorlalți jucători și pozițiile disponibile.
   Exemplu: (ana, '_ _ _ _ _ _ _ _ _ '), (ion, '_ _ _ _ _ _ _ _ _ _'), (vasile, '_ _ _ _ _ _ _ _ _ _')
2. *Ghicește pozitia.* Fiecare jucător alege unul dintre ceilalți doi jucători și introduce o poziție pe care crede că este unul dintre obiectele jucătorului respectiv. După ce toți jucătorii au trimis propunerile (jucător și poziție) la server, acesta verifică propunerile primite astfel:
   - dacă pe poziția propusă de jucător, la jucătorul ales, este un obiect, jucătorul va primi 7 puncte.
   - dacă poziția propusă de jucător, la jucătorul ales, este lângă un obiect (obiectul este în stânga sau în dreapta poziției alese de jucător), jucătorul va primi 3 puncte.
   - dacă pe poziția propusă de jucător, la jucătorul ales, nu este un obiect și nu este lângă un obiect, jucătorul va primi 0 puncte..

   După verificarea propunerilor, serverul trimite tuturor jucătorilor pozițiile actualizate pentru fiecare jucător în care sunt marcate pozițiile propuse astfel: 'O' pe o poziție propusă pe care este un obiect, 'N' pe o pozitie propusă aflată lângă un obiect și 'C' pe o poziție propusă care nu este obiect sau lângă un obiect.
   Exemplu (ana, '_ C _ _ O _ _ _'), (ion, '_ _ _ _ _ _ _ _ _ _'), (vasile, '_ _ _ _ _ N'). Aceste informații vor apărea automat pe interfața grafică a fiecărui jucator. Toți jucătorii văd aceleași informații pe interfața grafică.

   Acest pas se repeta de încă 2 ori. La finalul celor 3 runde, serverul va trimite tuturor jucătorilor punctajul total obținut de fiecare jucător (in ordine descrescătoare a numarului de puncte obținut) și toți jucătorii vor vedea clasamentul pe interfața grafică.
3. Un serviciu REST care permite vizualizarea jucătorilor participanți la un anumit joc și a celor 3 poziții propuse de fiecare jucător la începerea jocului.
4. Un serviciu REST care permite vizualizarea celor 3 propuneri (jucător și poziția propusă) făcute de un anumit jucător în timpul unui anumit joc și punctajul obținut de acesta la fiecare rundă.

Observație:

Pozițiile propuse de jucători pentru obiectele lor, propunerile de la fiecare rundă si punctajul obținut pentru fiecare propunere se păstreaza în baza de date.

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 (cu trimiterea pozitiilor) - 0.5p
- Trimiterea unei propuneri (cu adaugare in baza de date) - 0.5 p
- Determinarea punctajelor pentru o runda - 1 p
- 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