

## Subiect 24 iunie 2021.

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

Un joc cu 3 jucători numit *Ghicește cuvântul*. Fiecare jucător propune un cuvânt format din cel puțin 6 litere și fiecare jucător încearcă să ghicească cuvintele propuse 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 este afișat un câmp pentru introducerea cuvântului propus de jucător și un buton “Start joc”. Doar după ce trei jucători se autentifică în aplicație, introduc cuvântul 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 pentru fiecare litera a cuvântului propus de fiecare jucător.

**Exemplu:** (ion, ‘ \_ \_ \_ \_ \_ \_ ’), (ana, ‘ \_ \_ \_ \_ \_ ’), (vasile, ‘ \_ \_ \_ \_ \_ \_ ’)

2. *Ghicește litera*. Fiecare jucător alege unul dintre ceilalți doi jucători și introduce o litera care crede că apare în cuvântul propus de jucătorul respectiv. După ce toți jucătorii au trimis propunerile (jucător și litera) la server, acesta verifică propunerile primite astfel:

- dacă litera respectivă apare în cuvântul propus de acel jucător, jucătorul respectiv va primi câte 1 punct pentru fiecare apariție a literei în cuvântul respectiv;
- dacă litera nu apare deloc în cuvântul respectiv, jucătorul va primi 0 puncte;

**Exemplu:** Jucătorul ‘ion’ a propus la începerea jocului cuvântul ‘televizor’ și ‘ana’ propune la prima runda litera ‘e’ pentru cuvântul lui ‘ion’. ‘ana’ va primi 2 puncte deoarece litera ‘e’ apare de două ori în cuvântul ‘televizor’.

După verificarea soluțiilor, serverul trimite tuturor jucătorilor pozițiile literelor din cuvinte actualizate cu literele ghicite de toți jucătorii și punctajul obținut de fiecare jucător la runda respectivă. Aceste informații vor apărea automat pe interfața grafică a fiecărui jucător.

**Exemplu:** (ion, ‘ \_ e \_ e \_ \_ \_ \_ ’, 1), (ana, ‘ \_ \_ \_ \_ o \_ ’, 2), (vasile, ‘ \_ \_ \_ \_ \_ \_ \_ ’, 0)

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 cuvintelor propuse de jucători la începutul jocului pentru un anumit joc.
4. Un serviciu REST care pentru un anumit joc și un anumit jucător permite vizualizarea celor 3 propuneri și punctajul obținut la fiecare runda.

Observație:

1. Cuvintele propuse de jucători la pornirea jocului, propunerile de la fiecare rundă și punctajul obținut pentru fiecare propunere se păstrează 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 cuvintelor) - 0.5p
- Trimiterea unei propuneri(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