

## Subiect 11 iunie 2021.

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

Un joc cu trei jucatori. Doar utilizatorii autentificați pot juca acest joc. La începutul jocului, sistemul va trimite fiecărui participant 5 poziții cu valoarea asociată fiecărei poziții (ex. 30, 15, 5, 10, 20), iar toți jucătorii sunt poziționați înaintea primei poziții. Jucătorul care după 3 runde obține cei mai mulți bani (conform cerinței 2), câștigă jocul. Fiecare utilizator poate să facă următoarele:

1. *Login*. După autentificarea cu succes se deschide o nouă fereastră în care este afișat un buton de începere a jocului. Doar după ce trei jucători autentificați apasă butonul de pornire a jocului, începe jocul și toți jucătorii văd id-urile celorlalți participanți la joc, cele 5 poziții și valoarea fiecărei poziții. Dacă un alt utilizator autentificat apasă butonul de începere după începerea unui joc, acesta intră în starea de așteptare. La începerea jocului fiecare participant primește 50 lei de la server.
2. *Generare numar*. Fiecare jucător, când îi vine randul, generează aleator un număr  $n$  între 1 și 3 (1, 2 sau 3), prin apăsarea unui buton. După ce numărul este generat, acesta este trimis la server și jucătorul va înainta cu  $n$  poziții în lista pozițiilor afișate (e posibil să treacă iarași pe la start). Dacă pe noua poziție este deja un alt jucător, jucătorul curent trebuie să plătească celui alt jucător ocupant jumătate din valoarea poziției. Dacă noua poziție este liberă, jucătorul curent va plăti serverului valoarea poziției respective și va deveni ocupantul acelei poziții. Dacă noua poziție depășește cele 5 poziții și jucătorul trece din nou pe la start, acesta va primi 5 lei de la server. Toți jucătorii văd numărul  $n$  generat, jucătorul care l-a generat, noua poziție și suma plătită/obținută de el în urma acestei mutări. După ce un jucător a generat un număr, el nu mai poate genera alt număr până când nu generează și ceilalți jucători câte un număr. Cerința 2 se repetă de încă 2 ori.
3. *Logout*.
4. Un serviciu REST care pentru un anumit joc și jucător, permite vizualizarea pozițiilor ocupate de jucător și a sumelor primite sau plătite de acesta în timpul jocului.
5. Un serviciu REST care permite vizualizarea clasamentului final pentru anumit joc.

#### Cerințe:

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