

Examen MAP 18/02/2021

Intr-o lume in care putem sa fim orice, e important sa nu uitam sa fim buni. Pentru a sarabatori saptamana internationala a faptelor bune, in urmatoarele 3 ore, va invitam sa construiti o aplicatie care sa ne incurajeze sa ne dezvoltam simtul etic si amabilitatea. Astfel, se cere sa implementati urmatoarele cerinte:

Cerinte functionale - 10p:

La pornirea aplicatiei, se va deschide o fereasta care o sa permita unui user sa se inregistreze sau sa intre in contul sau.

- 1) Pentru **inregistrare**, utilizatorul trebuie sa completeze campurile corespunzatoare atributelor din clasa *Persoana* – **vezi Figura 1. (1p)**
Orasul va trebui sa fie ales dintr-un *combobox*. Pentru atributul *oras* se cere folosirea unui *enum*. **(0.5p)**
Side note: Va fi punctat integral daca inregistrarea va fi facuta cu succes (noul utilizator trebuie adaugat in fisierul txt/csv).
- 2) Pentru a **accesa contul deja existent (login)**, este necesara selectarea *username* dintr-o lista care contine toti utilizatorii deja inregistrati in aplicatie. **(1p)** Selectand printr-un simplu click utilizatorul, se poate trece la pasul 3).
- 3) Odata ce utilizatorul a reusit sa se inregistreze sau sa isi acceseze contul cu succes, este redirectionat catre un *nou view*, care contine 2 *tab-uri* in partea de sus a ferestrei. Titlurile taburilor sunt: **Doresc sa ajut!** si **Doresc sa fiu ajutat!** **(1p)**
- 4) In tab-ul **Doresc sa ajut!** sunt prezente doua tabele. Primul tabel prezinta lista de nevoi pentru orasul corespunzator utilizatorului inregistrat (mai putin cererile acestuia). **(1p)**
Side note: **Vezi Figura 1**. Atributul *omSalvator* din clasa *Nevoie* o sa fie initial *null*. Atributul *status* o sa fie initial *Caut erou!*
- 5) User-ul poate alege un anunt pentru o nevoie pe care doreste sa o rezolve, primind un *mesaj de confirmare* care specifica faptul ca nevoia aleasa i-a fost atribuita. **(1p)**
Side note: Atributul *omSalvator* din clasa *Nevoie* o sa fie actualizat cu id-ul utilizatorului care preia nevoia, iar *statusul* o sa fie schimbat in *Erou gasit!*
- 6) In momentul in care o nevoie si-a gasit un erou, nevoia nu mai poate fi selectata. **(1p)**
- 7) A doua tabela contine lista de fapte buna a utilizatorului inregistrat, care o sa fie updatata in timp real, pe masura ce user-ul selecteaza nevoi pe care doreste sa le rezolve. **(1p)**
- 8) In tab-ul **Doresc sa fiu ajutat!**, prin completarea unui formular, utilizatorul poate adauga o nevoie/problema proprie. **(1p)** *Deadline-ul* va fi specificat folosind un *DatePicker* **(0.5p)**
- 9) Orice schimbare care poate afecta tabelul din tabul **Doresc sa ajut!** care prezinta lista nevoilor trebuie sa fie vizibila in timp real tuturor utilizatorilor. **(1p)**

Sunt necesare urmatoarele clase in domain:

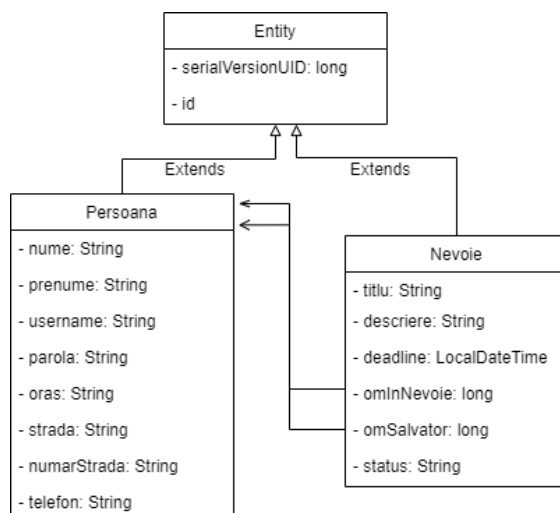


Figura 1

Cerinte non-functionale - 3p:

- Validarea datelor de intrare

Exemplu: Pentru inregistrarea unui nou user, validarea parolei prin adugarea unui nou input de verificare.

- Procesarea va avea loc numai la nivel de service sau de controller; interactiunea cu sursa de date se va face numai prin intermediul repository-ului (puteti folosi baze de date sau fisiere text)

- Interactiunea cu utilizatorul va avea loc numai in UI (GUI)

- Se cere eliminarea codului care nu este folosit precum si a functionalitatilor care nu s-au cerut (daca ati lucrat cu ceva template de la lab)

- Clasele, attributele și metodele lor vor avea exact numele cerute in problema sau nume sugestiv daca nu s-a specificat explicit numele lor.

IMPORTANT

- Se puncteaza doar cerintele functionale care ruleaza;
- Orice cod care nu poate fi explicat, atrage dupa sine nepunctarea cerintei/cerintelor din care face parte.

- Nu aveti voie sa comunicati in timpul examenului, in nicio modalitate posibila de a face acest lucru (chat, mail,etc)

1p din oficiu.

Timp de lucru: 1.5 ore.

Multa bafta! 😊