

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 fereastră care o să permită unui user să se înregistreze sau să intre în contul său.

- 1) Pentru **înregistrare**, utilizatorul trebuie să completeze câmpurile corespunzătoare atributelor din clasa Persoana – **vezi Figura 1. (1p)**
Orasul va trebui să fie ales dintr-un *combobox*. Pentru atributul *oras* se cere folosirea unui *enum*. **(0.5p)**
Side note: Va fi punctat integral dacă înregistrarea va fi făcută cu succes (noul utilizator trebuie adăugat în fisierul txt/csv).
- 2) Pentru **accesa contul deja existent (login)**, este necesară selectarea *username* dintr-o listă care conține toți utilizatorii deja înregistrați în aplicație. **(1p)** Selectând printr-un simplu click utilizatorul, se poate trece la pasul 3).
- 3) Odată ce utilizatorul a reușit să se înregistreze sau să își acceseze contul cu succes, este redirectionat către un *nou view*, care conține 2 *tab-uri* în partea de sus a ferestrei. Titlurile taburilor sunt: **Doresc să ajut!** și **Doresc să fiu ajutat!** **(1p)**
- 4) În tab-ul **Doresc să ajut!** sunt prezente două tabele. Primul tabel prezintă lista de nevoi pentru orașul corespunzător utilizatorului înregistrat (mai puțin cererile acestuia). **(1p)**
Side note: **Vezi Figura 1**. Atributul *omSalvator* din clasa Nevoie o să fie inițial *null*. Atributul *status* o să fie inițial *Caut erou!*
- 5) User-ul poate alege un anunț pentru o nevoie pe care dorește să o rezolve, primind un *mesaj de confirmare* care specifică faptul că nevoia aleasă i-a fost atribuită. **(1p)**
Side note: Atributul *omSalvator* din clasa Nevoie o să fie actualizat cu id-ul utilizatorului care preia nevoia, iar *statusul* o să fie schimbat în *Erou gasit!*
- 6) În momentul în care o nevoie și-a găsit un erou, nevoia nu mai poate fi selectată. **(1p)**
- 7) A doua tabelă conține lista de fapte bune a utilizatorului înregistrat, care o să fie updatată în timp real, pe măsura ce user-ul selectează nevoi pe care dorește să le rezolve. **(1p)**
- 8) În tab-ul **Doresc să fiu ajutat!**, prin completarea unui formular, utilizatorul poate adăuga 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 să ajut!** care prezintă lista nevoilor trebuie să fie vizibilă în timp real tuturor utilizatorilor. **(1p)**

Sunt necesare urmatoarele clase in domain:

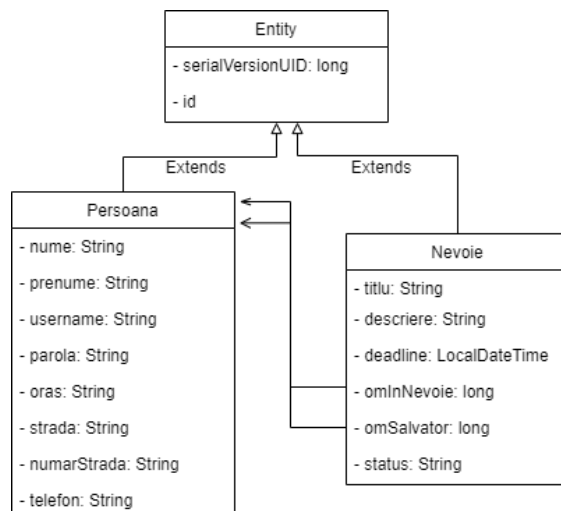


Figura 1

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! 😊