

Rute trenuri – 10 Februarie 2023

Se doreste o aplicatie cu o interfata grafica intuitiva, care sa permita vizualizarea rutelor trenurilor.

Cerinte functionale - 7p:

- 1) **1.5p** La pornirea aplicatiei, se va deschide o fereastra cu un buton **“Open New Client Window”**. La fiecare apasarea a butonului se va deschide cate o **noua fereastra** ce contine:
 - doua dropdown-uri: **“Departure City”** si **“Destination City”**; fiecare dintre aceste dropdown-uri contine **numele oraselor citite tabelul Cities**;
 - un checkbox **“Direct Routes Only”**;
 - un buton **“Search”**

Se va defini clasa:

City: { **id**: int/String, **name**: String }

Tabelul **Cities** va contine intrari de forma:

C1, Cluj-Napoca

C2, Apahida

C3, Campia Turzii

C4, Razboieni

- 2) **3p** La apasarea butonului **“Search”**, fereastra client va afisa **toate rutele/traseele posibile** dintre cele doua orase alese (backtracking), sub forma: [City] —[TrainId]—> [City] —[TrainId]—> [City] ...
 - Se va defini clasa: **TrainStation**: { **trainId**: int/String, **departureCityId**: int/String, **destinationCityId**: int/String }
 - Statiile rutelor se citesc din **tabelul TrainStations**, care va contine intrari de forma:
 - T1, C1, C2*
 - T1, C2, C3*
 - T2, C2, C3*
 - T2, C3, C4*
 - T3, C3, C4*
 - T4, C2, C4*
 - Daca checkbox-ul **“Direct Routes Only”** este bifat, se vor afisa doar rutele care nu presupun schimbarea trenului
1.5/jumate din punctaj pt o solutie
 - Daca checkbox-ul **“Direct Routes Only”** nu este bifat, se vor afisa si rutele care implica schimbări de trenuri
1.5/jumate din punctaj pt o solutie

Exemple:

Considerand datele din fisierele/tabelele de mai sus:

- pentru (Departure City = Cluj-Napoca, Destination City = Campia Turzii, Direct Routes Only = true), se va afisa:
Cluj-Napoca —T1—> Apahida —T1—> Campia Turzii
- pentru (Departure City = Cluj-Napoca, Destination City = Campia Turzii, Direct Routes Only = false), se va afisa:
Cluj-Napoca —T1—> Apahida —T1—> Campia Turzii
Cluj-Napoca —T1—> Apahida —T2—> Campia Turzii
- pentru (Departure City = Cluj-Napoca, Destination City = Razboieni, Direct Routes Only = true), nu se gaseste niciun rezultat.
- pentru (Departure City = Cluj-Napoca, Destination City = Razboieni, Direct Routes Only = false), se va afisa:
Cluj-Napoca —T1—> Apahida —T1—> Campia Turzii —T2—> Razboieni
Cluj-Napoca —T1—> Apahida —T1—> Campia Turzii —T3—> Razboieni
Cluj-Napoca —T1—> Apahida —T2—> Campia Turzii —T2—> Razboieni
Cluj-Napoca —T1—> Apahida —T2—> Campia Turzii —T3—> Razboieni
Cluj-Napoca —T1—> Apahida —T4—> Razboieni

- 3) **1.5p** Fiecare fereastra va afisa **in timp real** numarul de alti clienti care au aplicat aceleasi filtre (Departure City si Destination City) la momentul actual, printr-un mesaj **“[N] other user(s) are looking at the same route”**.

Exemplu:

Client Window 1: Departure City = X, Destination City = Y

Client Window 2: Departure City = X, Destination City = Y

Client Window 3: Departure City = X, Destination City = Z

=> Client Window 1 si 2 afiseaza mesajul **“1 other user(s) are looking at the same route”**;

Client Window 3 nu afiseaza niciun mesaj

Schimbam filtrele din Client Window 3: Departure City = X, Destination City = Y

=> Toate ferestrele afiseaza mesajul **“2 other user(s) are looking at the same route”**

Schimbam filtrele din Client Window 2 si Client Window 3: Departure City = X, Destination City = Z

=> Client Window 2 si 3 afiseaza mesajul **“1 other user(s) are looking at the same route”**;

Client Window 1 nu afiseaza niciun mesaj

Schimbam filtrele din Client Window 3: Departure City = Y, Destination City = Z

=> Nicio fereastra nu mai afiseaza niciun mesaj

- 4) **1p** Pentru fiecare ruta afisata, sa se afiseze si pretul acesteia.

Pretul unei rute se calculeaza dupa formula $PRET_PER_STATIE * NUMBER_OF_STATIONS$, unde:

- $PRICE_PER_STATION$ este o constanta de tip float;
- $NUMBER_OF_STATIONS$ este numarul de statii din cadrul rutei in cauza.

Exemplu:

Daca $PRICE_PER_STATION$ este 10.0;

Departure City = Cluj-Napoca, Destination City = Razboieni, Direct Routes Only = false,

se va afisa:

Cluj-Napoca —T1—> Apahida —T1—> Campia Turzii —T2—> Razboieni, price: 30.0

Cluj-Napoca —T1—> Apahida —T1—> Campia Turzii —T3—> Razboieni, price: 30.0

Cluj-Napoca —T1—> Apahida —T2—> Campia Turzii —T2—> Razboieni, price: 30.0

Cluj-Napoca —T1—> Apahida —T2—> Campia Turzii —T3—> Razboieni, price: 30.0

Cluj-Napoca —T1—> Apahida —T4—> Razboieni, price: 20.

Cerinte non-funcționale: (3p)

- In proiectare va trebui sa tineti cont de urmatoarele constrângeri:
 - 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. Dacă această cerință nu este îndeplinită, nota de la cerințele funcționale corespunzătoare va fi diminuată cu 10%.
 - Datele se citesc din baza de date (postgresql) **2.5 p** (se acorda punctaje intermediare comf cu cerintele functionale corespunzatoare).
 - Interacțiunea cu utilizatorul va avea loc numai in UI (GUI) **0.25p**
 - Validarea datelor de intrare **0.25p**

Important!!!!!!

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

1 Punct din oficiu.

Timp de lucru: 150 min