

## Colectionar

Un colectionar foloseste o aplicatie mobila pentru a gestiona obiectele sale. Colectionarul gestioneaza obiectele iar prietenii si colegii pot sa le imprumute sau vizualizeze.

Pe server se vor memora urmatoarele detalii:

- Id - Identificatorul unui obiect. O valoare intreaga mai mare decat zero.
- Nume - Numele obiectului. O valoare de tip string.
- An - Anul achizitiei. O valoare de tip intreg.
- Luna - Luna achizitiei. O valoare de tip intreg.
- Zi - Ziua achizitiei. O valoare de tip intreg.
- Valoare - Valoarea obiectului. O valoare de tip intreg.
- Rezervari - Numar rezervari. O valoare de tip intreg.
- Status - O valoare booleana care specifica daca obiectul mai este disponibil.

Aplicatia ofera urmatoarele functionalitati (fara a fi nevoie sa restartam):

- Sectiune colectionar (ecran separat).
  - a. (1p) Vizualizare obiecte, intr-o lista. Folosind apelul **GET /obiecte**, un utilizator va accesa toate obiectele din colectie, si le va afisa doar pe cele disponibile. Elementele din lista trebuie sa prezinte: Id-ul obiectului, numele, anul, luna, ziua si statusul. Obiectele se ordoneaza crescator dupa an, luna, si zi. Daca aplicatia este offline, se va afisa un mesaj corespunzator si un buton care sa reincerce apelul. Odata ce un apel reuseste se vor folosi doar obiectele din baza de date offline/locala.
  - b. (1p) Adaugare obiect. Folosind apelul **POST /nou** specificand toate attributele unei obiect (mai putin status). Ca raspuns de la server se primeste un intreg reprezentant id-ul obiectului. Se afiseaza un mesaj de confirmare cu raspunsul de la server. Daca device-ul este offline, inregistrarea se persista in baza de date locala.
  - c. (1p) Stergere obiect. Selectand un obiect din lista de mai sus, si folosind apelul **DELETE /distruge** specificand doar id-ul obiectului. Ca raspuns de la server se primeste un intreg reprezentant id-ul obiectului sters. Se afiseaza un mesaj de confirmare cu raspunsul de la server. Daca device-ul este offline, cererea se persista in baza de date locala.
- Sectiune prieteni (ecran separat).
  - a. (1p) Vizualizare obiecte disponibile dupa an. Utilizatorul specifica un an, iar apoi folosind apelul **GET /an**, se prezinta doar obiectele avand anul specificat. Obiectele se ordoneaza descrescator dupa valoare. Elementele din lista trebuie sa prezinte: numele, anul, si valoarea. Odata ce un apel reuseste se vor folosi doar obiectele din baza de date offline/locala.
  - b. (1p) Rezervare obiect. Prin selectarea unui obiect, din lista precedenta, folosind apelul **POST /imprumuta**, specificand id-ul obiectului, utilizatorul va trimite o cerere de rezervare. Apel disponibil doar online.
  - c. (1p) Returnare obiect. Prin selectarea unui obiect, din lista precedenta, folosind apelul **POST /restituie**, specificand id-ul obiectului, utilizatorul va trimite o cerere de returnare. Apel disponibil doar online.
- Sectiune statistica (ecran separat).
  - a. (0.5p) Vizualizare obiecte valoroase. Folosind apelul **GET /valoare**, se prezinta doar top 5 obiecte disponibile ordonate descrescator dupa valoare. Elementele din lista trebuie sa prezinte: numele, anul, luna achizitiei si valoarea.

- b. (0.5p) Vizualizare obiecte ieftine. Folosind acelari apelul **GET /valoare**, se prezinta doar top 10 obiecte disponibile ordonate crescator dupa valoare. Elementele din lista trebuie sa prezinte: numele, anul, luna, ziua achizitiei si valoarea.

(1p) Pe server, odata ce se adauga o noua inregistrare, folosind o conexiune de tip WebSocket, se va trimite un mesaj tuturor aplicatiilor conectate. Aplicatiile conectate, vor afisa o notificare cu continutul mesajului primit (folosind un dialog, sau snackbar sau toast). Intr-un format uman, nu text-ul JSON sau toString.

(0.5p) Pentru fiecare apel catre server/DB se va afisa un indicator de progres.

(0.5p) Pentru fiecare interactiune server/DB, daca se produce o eroare, aplicatia va afisa mesajul folosind un toast, snackbar sau dialog. Indiferent de apel (server sau DB), se va inregistra un mesaj de debug/log.