

**Documentația conține: Enunț, lista de funcționalități, planul de iterații. Pentru fiecare funcționalitate: scenariu de rulare**

**Enunț: O aplicație ce gestionează o listă de studenți și lista de discipline. Se va avea în vedere lista de funcționalități care cuprinde: adăugarea, ștergerea, modificarea, căutarea, asignarea și statistica între cele 2 liste.**

## **Lista de funcționalități:**

### **Gestionarea listei de studenți și discipline**

Adăugarea unui student nou cu un ID și nume specific.

Ștergerea unui student existent din catalog, identificat după ID.

Modificarea numelui unui student existent, identificat după ID.

Adăugarea unei discipline noi cu un ID, nume și numele profesorului responsabil.

Ștergerea unei discipline existente din catalog, identificată după ID.

Modificarea numelui și profesorului unei discipline existente, identificate după ID.

### **Căutarea de student și disciplină**

Căutarea unui student în catalog după nume sau ID.

Căutarea unei discipline în catalog după nume sau ID.

### **Asignarea de note la un student și o disciplină**

Asignarea unei note unui student pentru o disciplină specifică.

## Crearea de statistici

Obținerea unei liste de studenți și a notelor lor la o disciplină specifică, ordonată alfabetic după nume sau după notă.

Obținerea unei liste cu cei mai buni 20% dintre studenți, ordonați după media notelor lor la toate disciplinele, inclusiv numele și nota medie.

## Planul de iterații:

### Iterația 1: Structura de bază și funcționalitățile de bază

Crearea structurii de bază a aplicației cu clasele STUDENT, DISCIPLINA, NOTA, REPOSITORY, UI SI CONTROLLER

Implementare funcționalităților de adăugare, ștergere și modificare a studenților și discipline.

Crearea de teste pentru functionalitatiile de baza

### Iterația 2: Extinderea funcționalităților

Cautarea unui student / disciplina.

Implementarea funcționalităților pentru asignarea de note la un student și o disciplină.

Teste necesare pentru fiecare functie + extindere repo/controller

### Iterația 3: Interfață utilizator și testare

Finzalire detalii ui

Implementarea funcționalităților pentru crearea de statistici avansate, inclusiv lista celor mai buni 20% dintre studenți.

## Scenariu de rulare pentru adăugarea unui student:

Utilizatorul introduce comanda pentru adăugarea unui student nou:

```
add_student (1 "Andrei")
```

Aplicația verifică dacă datele introduse sunt valide (ID unic și nume nevid).  
Dacă datele sunt valide, aplicația adaugă studentul în catalog și afișează un  
mesaj de confirmare: Student adaugat cu succes!!!

## Scenariu de rulare pentru actualizarea unui student:

Utilizatorul pornește aplicația.

Utilizatorul introduce comanda pentru actualizarea numelui unui student  
existent:

```
update_student(1, "Andrei")
```

Aplicația verifică dacă studentul cu ID-ul 1 există în catalog.  
Dacă studentul există, aplicația actualizează numele acestuia și afișează un  
mesaj de confirmare:

Modificat cu succes!!!

Dacă studentul nu există, aplicația afișează un mesaj de eroare corespunzător  
și cere din nou datele de intrare.

## Scenariu de rulare pentru ștergerea unui student:

Utilizatorul pornește aplicația.

Utilizatorul introduce comanda pentru ștergerea unui student existent:

```
Delete_student(1, "Andrei")
```

Aplicația verifică dacă studentul cu ID-ul 1 există în catalog.  
Dacă studentul există, aplicația șterge studentul din catalog și afișează un  
mesaj de confirmare:

Student sters cu succes!!! Dacă studentul nu există, aplicația afișează un mesaj de eroare corespunzător și cere din nou datele de intrare.

Adaugare student:

| cmd: add_student            | Rezultat            |
|-----------------------------|---------------------|
| add_student(1, "Andrei")    | {1:"Andrei"}        |
| add_student(2, " ")         | Nu poate fi vid     |
| add_student(1, "Alexandru") | Deja cu Id existent |

Stergere student:

| cmd: delete_student                                       | Rezultat        |
|---|-----------------|
| add_student(1,"Andrei")<br>delete_student(1,"Andrei")     |                 |
| delete_student(1, "Andrei")<br>delete_student(1,"Andrei") | id deja sters   |
| detele_student(1, "")                                     | nu poate fi vid |

Modificare student:

| cmd: update_student                       | Rezultat      |
|---|---------------|
| {1: "Alex"}<br>update_student(1,"Andrei") | {1: "Andrei"} |
| update_student(1)                         |               |