Documentatie proiect ISP

Moise Victor-Stefan

Paliciuc Cosmin-Constantin

Pleanta Mihai-Alexandru

333A

1. **Tema**

Aplicația are ca obiectiv facilitarea colaborării între studenți prin crearea unui mediu digital unde aceștia pot găsi colegi cu interese academice comune, pot organiza sesiuni de învățare și pot partaja resurse utile.

Funcționalități principale:

* Administratorul are rolul de a:
  + crea și gestiona grupuri de studiu (ex. pe discipline sau teme comune);
* Studenții pot:
  + vizualiza și adera la grupuri existente;
  + crea sesiuni de învățare în cadrul grupurilor;
  + încărca și accesa materiale de studiu partajate (ex. fișiere PDF, linkuri, notițe);
  + colabora cu alți membri ai grupului.

1. **Diagrama de clase**

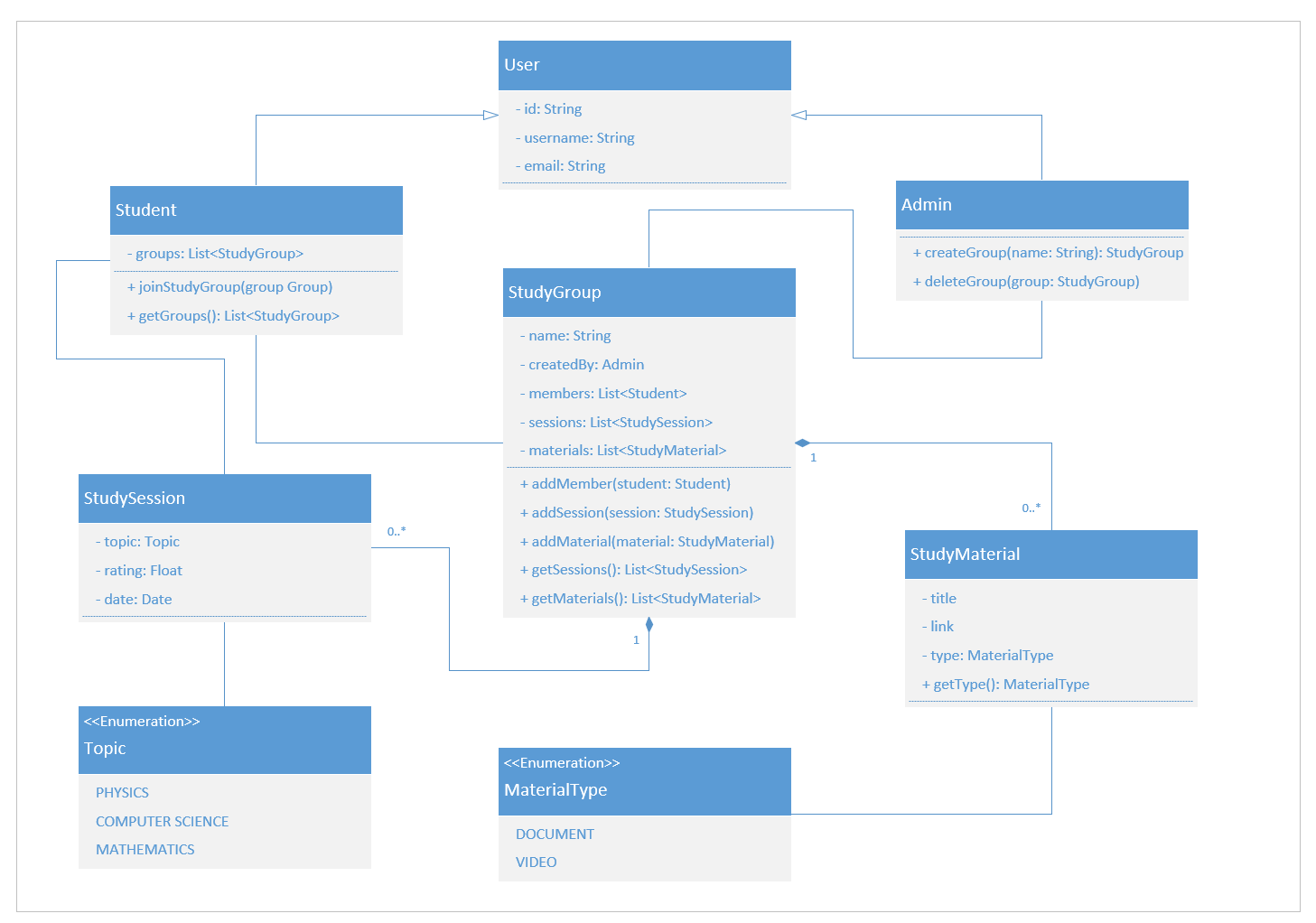
****

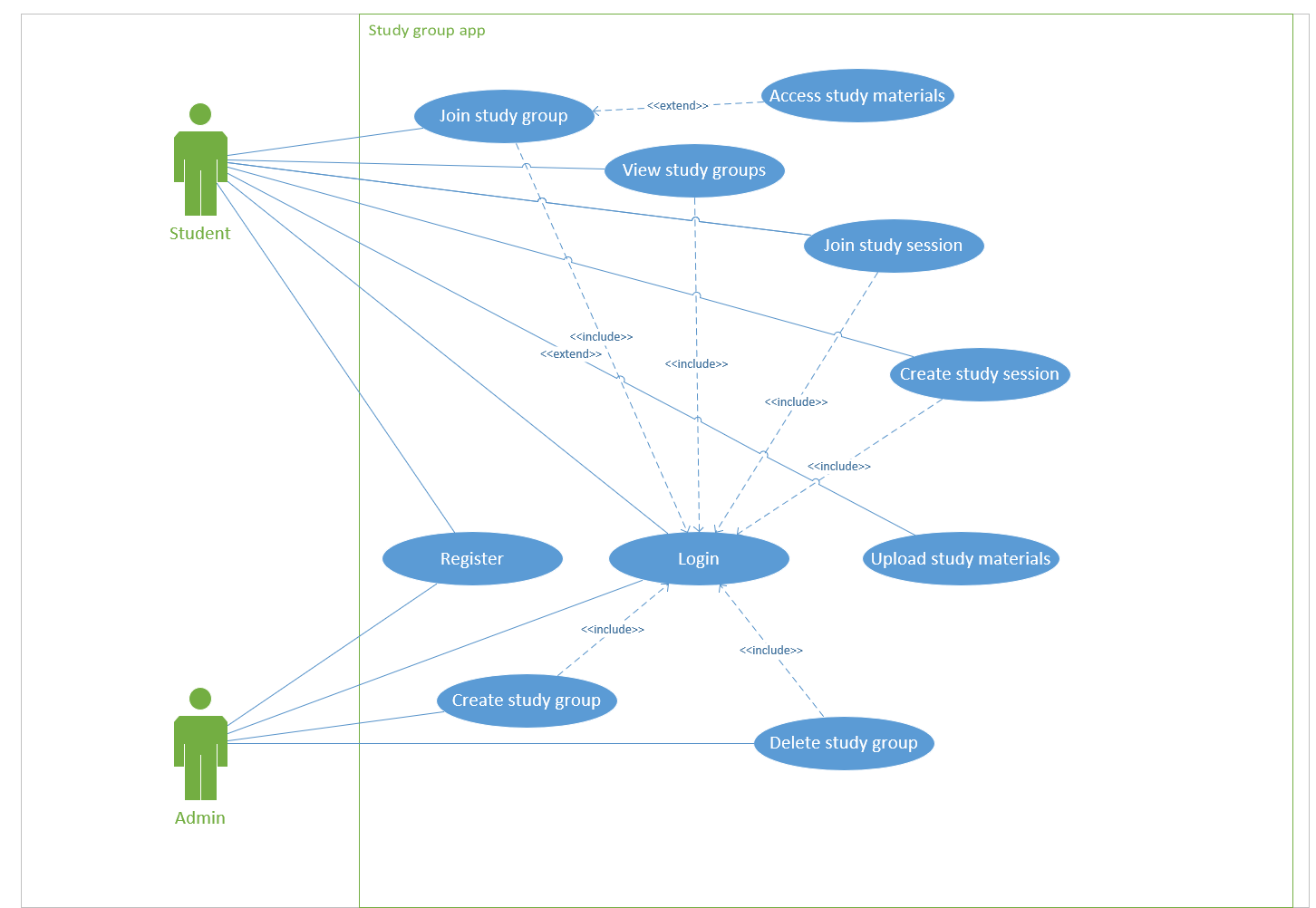
Diagrama prezintă modelul de domeniu pentru aplicația de organizare a grupurilor de studiu astfel:

* **User** (abstract) definește câmpurile comune (id, name, email) și metoda joinStudyGroup(...).
* **Student** extinde User, are o listă de grupuri (groups) și implementează constructorul, joinStudyGroup(...) și getGroups().
* **Admin** extinde User și în plus poate crea și șterge grupuri prin metodele createGroup(...) și deleteGroup(...).
* **StudyGroup** este compus dintr-un nume, referința la Admin care l-a creat, lista de membri (List<Student>), de sesiuni (List<StudySession>) și de materiale (List<StudyMaterial>). Oferă metode pentru a adăuga membri, sesiuni și materiale și pentru a le accesa.
* **StudySession** înregistrează topicul, data și locația fiecărei sesiuni de învățare.
* **StudyMaterial** are un titlu, un link și un tip (MaterialType – DOCUMENT, VIDEO, LINK, PRESENTATION), plus metoda getType().

Relații cheie:

* Moștenire (User → Student/Admin)
* Compoziție (StudyGroup “deține” sesiuni și materiale)
* Asocieri: studenții sunt membri ai grupurilor și administratorii le creează.

1. **Diagrama cazurilor de utilizare**

****

Use Case Diagram-ul „Study Group System” descrie interacțiunile dintre actorii principali — **Student** și **Admin** — și funcționalitățile sistemului:

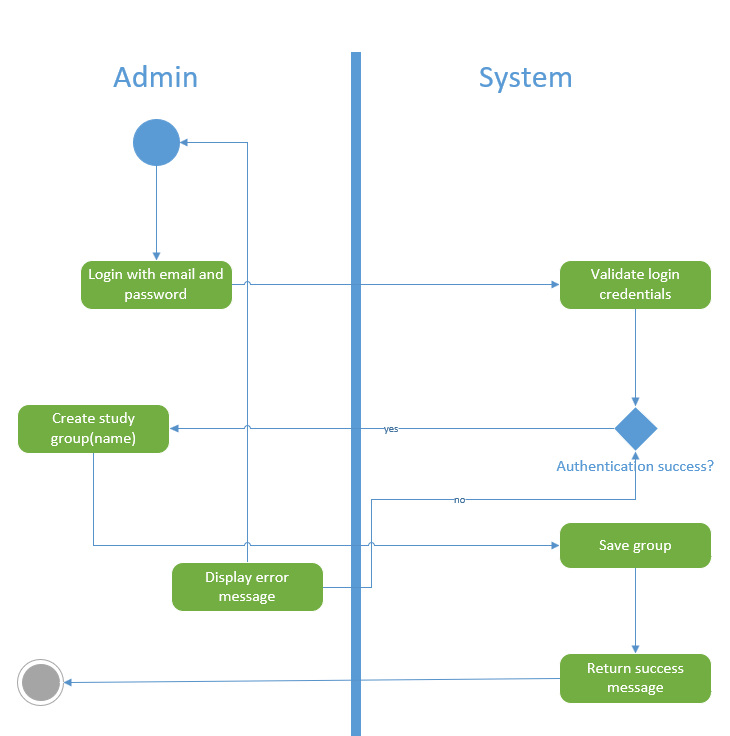
* **Student**
  + Se poate **autentifica** (Login) și **înregistra** (Register).
  + **Vizualizează** (View Study Groups) și **aderă** la grupuri de studiu (Join Study Group) — operațiune ce include autentificarea.
  + **Inițiază sesiuni de învățare** (Create Study Session), **participă** la ele (Join Study Session)
  + **Accesează** (Access Study Materials) și **încarcă** (Upload Study Materials) resurse de învățare.
* **Admin**
  + Se **autentifică** și se **înregistrează**.
  + **Creează** (Create Study Group), **gestionează** (Manage Study Groups) și **șterge** (Delete Study Group) grupuri de studiu — toate acestea incluzând autentificarea.

**Relații între use case-uri**

* Toate operațiunile cheie (aderare, creare/ștergere/grupuri, crearea și participarea la sesiuni) includ verificarea autentificării (Login).
* **Inițierea sesiunilor** se află în zona de responsabilitate a studentului, reflectând autonomia acestuia de a propune și organiza întâlniri de studiu.

1. **Lucru individual**

* **Moise Victor Stefan**
  1. **Diagrama de activități**



Scenariul de secvenţă ilustrează cum **Admin** creează un grup de studiu în trei paşi:

1. **Autentificare Admin**
   * Admin → System: login(email, password)
   * System → Admin: loginSuccess()
2. **Creare grup de studiu**
   * Admin → System: createStudyGroup(title, description)
   * System → StudyGroup: instanţierea obiectului StudyGroup(title, description)
   * StudyGroup → System: groupCreated()

Astfel, după validarea credenţialelor, sistemul alocă responsabilitatea de creare grupului către componenta StudyGroup şi confirmă administratorului faptul că grupul a fost creat cu succes.

* 1. **Diagrama secvențială**

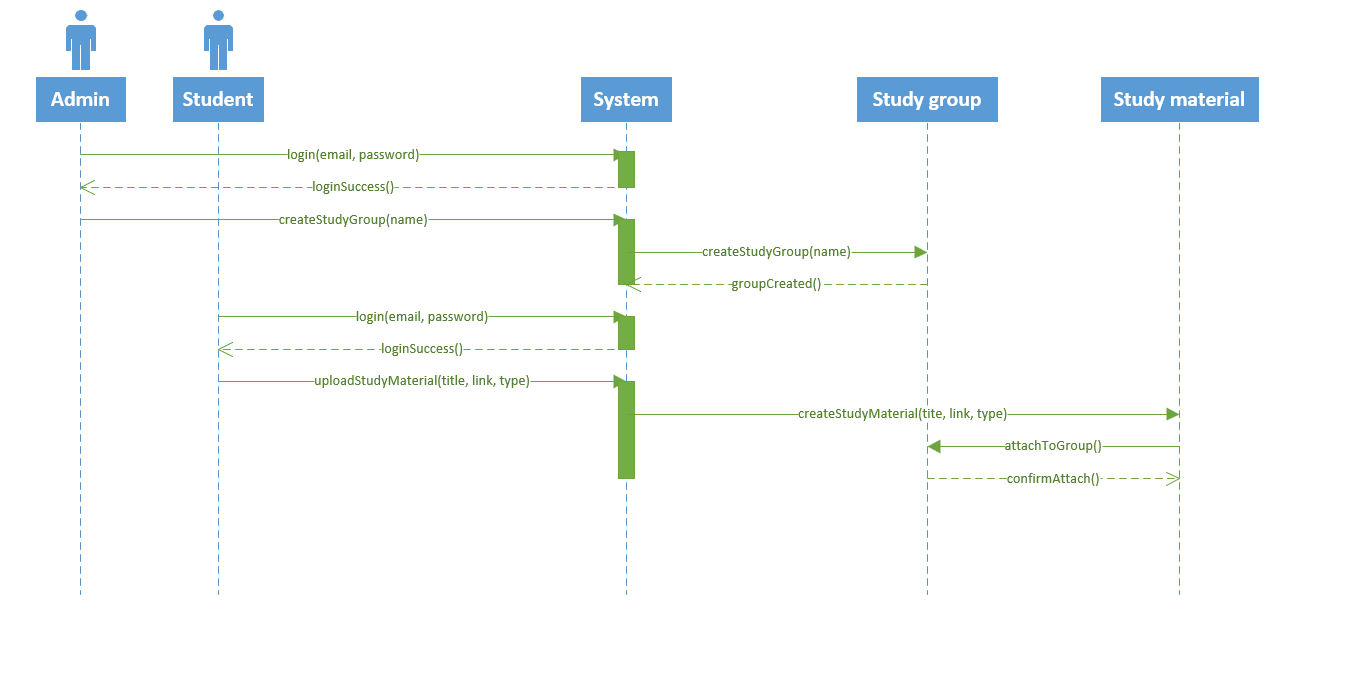
****

Diagrama ilustrează două fluxuri principale de secvență în „Study Group System”:

1. **Creare grup de studiu de către Admin**
   * **Admin** se autentifică (login(email, password) → loginSuccess()).
   * După autentificare, Admin solicită createStudyGroup(name) către **System**, care instanțiază un nou obiect **StudyGroup** și confirmă cu groupCreated().
2. **Încărcare material de studiu de către Student**
   * **Student** se autentifică similar (login(email, password) → loginSuccess()).
   * Apoi trimite uploadStudyMaterial(title, link, type) către **System**, care creează un obiect **StudyMaterial**, îl atașează la un **StudyGroup** (attachToGroup()) și primește confirmarea atașării.

În ambele cazuri, **System** acționează ca orchestrator: validează autentificarea și delegă crearea entităților către componentele **StudyGroup** și **StudyMaterial**, asigurând coerența și confirmarea operațiunilor.