Universitatea Babeș-Bolyai

Temă Middleware

Student: Bărcan Florin-George

Specializare: Sisteme distribuite în Internet

Grupă: 244

Cuprins

[Prezentarea aplicației 1](#_Toc105275250)

[Implementare 2](#_Toc105275251)

[Diferențe 2](#_Toc105275252)

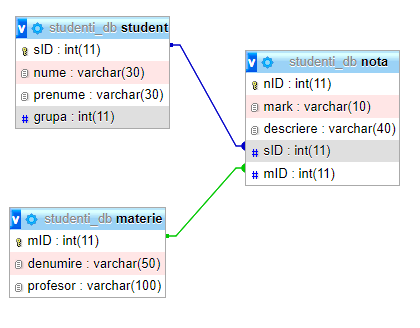
[Rularea 2](#_Toc105275253)

# Prezentarea aplicației

Enunțul problemei constă în crearea unei aplicații middleware care ține evidența rezultatelor la examene. Se cere posibilitatea de a adăuga/modifica/șterge studenți, adăuga/modifica/șterge materii și adăuga/modifica note.

Este necesar pentru ca la calculatorul prin care se rulează aplicația să fie pornit serverul MySQL care ascultă portul 3306 și să se poată porni serviciul ActiveMQ de la Apache. De asemenea, trebuie instalat serviciul STOMP pentru NodeJS și Python.

În baza de date, numită studenti\_db, avem tabelele Student, Materie și Nota. Relațiile din tabel sunt de many-to-many în care tabela Nota este tabela intermediară.



Atributele din clasele Student, Materie și Nota sunt aceleași ca și coloanele din baza de date de pe fiecare tabelă respectivă.

Clasa DBUtils conține metodele care folosesc limbaj de interogare pentru transmiterea de date de la și către server la baza de date folosită.

Clasa Controller preia mesajul transmis de client către server și folosește primul parametru pentru a apela funcționalitatea respectivă.

Clasele Server și Client se conectează la serviciul ActiveMQ. Clasa Server ascultă porturile și așteaptă mesajele de tip cerere de la client și transmite înapoi mesaje de tip răspuns înapoi către client.

## Implementare

Problema a fost implementată în limbajele Java, NodeJS și Python.

Pentru rezolvarea problemei se folosește o implementare orientată-obiect. Obiectele declarate sunt Client pentru client, iar pentru server sunt Server, Controller, DBUtils, Student, Materie și Nota.

## Diferențe

Pentru limbajul Java a fost nevoie de crearea unui fatJar care să permită rularea serverului și clientului. Serverul a fost nevoit de asemenea să ii fie introdus în arhivă pachetul de mysql-connector care să permită conectarea la baza de date.

Implementarea în limbajul Java folosește portul 61616 pe lângă celelalte 2 implementări care folosesc portul 61613.

## Rularea

Pentru rularea acestor aplicații, se folosesc fișierele mainSystem, fiecare având terminația în funcție de limbajul în care a fost scris. Pentru NodeJS se folosește comanda „node mainSystem.js”, iar pentru Python se folosește „py mainSystem.py”.

În cazul implementării în limbaj Java, se execută fișierul de tip gradle care crează fatjar-ul și se execută comanda „java JavaJmsSrv-1.0-SNAPSHOT-all.jar”.