UNIVERSITATEA POLITEHNICA BUCUREȘTI

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

DEPARTAMENTUL CALCULATOARE

|  |  |
| --- | --- |
| upb | cs |

PROIECT DE DIPLOMĂ

**Aplicație web pentru managementul învățării**

versiunea 2024

Panait Ștefan-Andrei

**Coordonator științific:**

Prof. dr. ing. Andrei Ionescu

BUCUREŞTI

2024

UNIVERSITY POLITEHNICA OF BUCHAREST

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

COMPUTER SCIENCE DEPARTMENT

|  |  |
| --- | --- |
| upb | cs |

DIPLOMA PROJECT

Aplicație web pentru managementul învățării

2024 version

Ioana Popescu

**Thesis advisor:**

Prof. dr. ing. Andrei Ionescu

BUCHAREST

2024

**CUPRINS**

[Sinopsis 2](#_Toc7074723)

[Abstract **Error! Bookmark not defined.**](#_Toc7074724)

[Mulțumiri 3](#_Toc7074725)

[1 Introducere 4](#_Toc7074726)

[1.1 Context 4](#_Toc7074727)

[1.2 Problema 4](#_Toc7074728)

[1.3 Obiective 5](#_Toc7074729)

[1.4 Structura lucrării 4](#_Toc7074730)

[2 Analiza și specificarea cerințelor 7](#_Toc7074731)

[3 Studiu de piață / Abordări existente 11](#_Toc7074732)

[3.1 Indicații formatare figuri **Error! Bookmark not defined.**](#_Toc7074733)

[4 Soluția propusă 17](#_Toc7074734)

[4.1 Indicații formatare formule **Error! Bookmark not defined.**](#_Toc7074735)

[5 Detalii de implementare 27](#_Toc7074736)

[5.1 Indicații formatare tabele 27](#_Toc7074737)

[6 Studiu de caz / Evaluarea rezultatelor 29](#_Toc7074738)

[7 Concluzii 31](#_Toc7074739)

[8 Bibliografie 32](#_Toc7074740)

[9 Anexe 34](#_Toc7074741)

# Sinopsis

Aplicația web pentru managementul învățării este o platformă ușor de utilizat pentru a facilitarea educației și a studiul suplimentar, online. Această are o interfață prietenoasă fiind foarte fiabila atat pentru profesori cat și pentru cursanți in vederea interacționarii, gestionarii cursurilor. Cursantii inscrisi pot s primimeasca recenzii personalizate și să acceseze diferite resurse materiale educative.

Aplicația ofera siguranta utilizatorilor prin autentificare securizata si controlul accesului pe baza de rol pentru modificarea permisiunilor la functionalitati. Printre alte optiuni, utilizatorul isi poate personaliza profilul, atat in rolul de student cat si de instructor. Profesorii pot edita modulele cursurilor s emblema, pot da teme suplimentare si materiale elevilor, precum si recenzie personalizata si nota dupa verificarea temelor efectuate de cursanti.

# Mulțumiri

(opțional) Aici puteți introduce o secțiunea specială de mulțumiri / acknowledgments.

# Introducere

In domeniul educatiei, nevoia de inovatie este constanta. Atat contextul mondial cat si tehnologia dezvoltata din ultimii ani au atins inevitabil si anii de temelie din viata fiecarui individ, mai exact, anii de scoala astfel, modernizarea invatamantului reprezinta o prioritate. Cu o tranzitie neprevazuta si foarte rapida in domeniul invatamatului, profesorii si elevii au fost si sunt nevoiti sa utilizeze diferite platforme online pentru continuarea predarii si a studiului. Este evident faptul ca nevoia de performanta, flexibilitate, cooperare si accesibilitate nu a fost niciodata la fel de mare.

Aplicatia de management al invatarii pe care o prezint se adreseaza aceastor nevoi, oferind un mediu online complexa pentru facilitarea si sprijinul educatiei si al dezvoltarii. Aceasta platforma este creata pentru imbunatatirea experientei educationale a studentilor, a profesorilor, dar si a unitatilor de invatamant.

La baza, aplicatia ofera o interfata centralizata, prietenoasa cu utilizatorii si clara, care favorizeaza comunicarea dintre profesori si cursanti. Cu capabilitati ce varieaza de la creeare cursului, gestionarea si editarea acestuia pana la livrarea de continut educativ suplimentar si evaluare detaliata si notarea temelor efectuate si trimise profesorului prin intermediul platformei, aplicatia este de un real ajutor in interactiunea profesor-cursant si facilitarea intelegerii notiunilor precum si suplimentarea orelor standard. Focalizandu-se pe accesibilitate si performanta la nivel inalt, aplicatia web asigura utilizatorilor navigarea cu facila pe platforma, accesarea resurse materiale. Astfel, aplicatia asigura elevilor siguranta ca educatia lor nu este afectata si pot continua calatoria lor de dezvoltare constant, indiferent de factorii care pot parea ca impiedica continuitatea educatiei.

De altfel, consider ca fiabilitatea si usurinta folosirii aplicatiei pot fi cele mai importante atuuri deoarece astfel poate fi la indemana oricui, fara a fi nevoie de timp suplimentar special pentru intelegerea si invatarea folosirii aplicatiei.

## Context

Situatiile neprevazute din ultimii ani au dus catre o schimbare drastica a organizarii invatamantului, de la metodele traditionale de educatie, la un nou tip de educatie, cel online, care inca tinde sa se dezvolte din ce in ce mai mult. Aceasta schimbare a evidentiat importanta si nevoia unei platforme digitale bine dezvolatate usor de folosit in vederea asigurarii continuitatii educatiei.

## Problema

Invatarea suplimentare poate deveni greu de realizat in contextul traditional in care profesorii si elevii sunt constransi si nu isi pot desfasura orele si materialele dupa cum doresc adaptandu-se in functie de fiecare elev in parte. In continuare voi enumera o parte din inconvenientele ce restrictioneaza invatarea suplimentara in contextul traditional:

* Limitarile de timp
* Limitarile de spatiu
* Numarul redus al resurselor materiale puse la dispozitie
* Lipsa interactivitatii si modernizarii
* Lipsa recenziilor detaliate si personalizate
* Lipsa flexibilitatii
* Lipsa temelor personalizate in functie de nevoile elevilor
* Constrangerea folosirii cartilor impuse de scoala

## Obiective

Obiectivul principal pe care mi l-am propus in crearea unei noi platforme de invatare, in scopul invatarii suplimentare, il constituie facilitarea, sprijinirea, inovarea si imbunatatirea calitatii invatarii suplimentare, atat pentru studenti cat si pentru profesori.

Consider ca invatarea suplimentara trebuie sa fie la indemana oricui si usor de accesat si folosit, de aceea vdoresc sa evidentiez o parte dintre beneficiile pe care platforma le ofera profesorilor si studentilor.

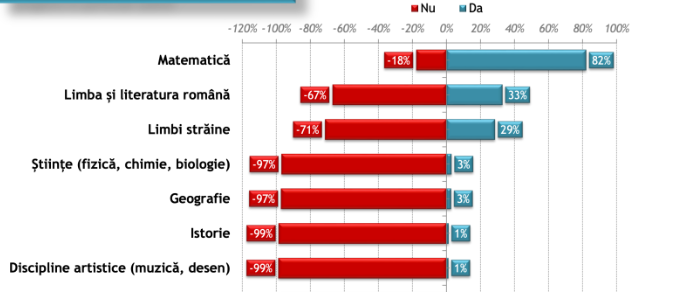
1. Beneficii pentru profesor:
2. Flexibilitate in adaugarea resurselor materiale educative: Profesorii pot incarca oricand materialele educative care le sunt disponibile cursantilor in orice moment. De asemenea profesorii pot adauga materiale selectate de ei, nu doar din cartile de la scoala.
3. Crearea si gestionarea temelor: Profesorii pot adauga diferite probleme in lista de exercitii, de asemenea acestia pot select orice problema din lista pentru a crea o tema pentru elevi. Temele pot fi personalizate pentru nevoile comune ale mai multor studenti sau personalizate pentru un anumit elev. Profesorul poate impune sau nu un anumit termen limita pentru realizarea si incarcarea temelor de catre elevi.
4. Evaluarea temelor: Dupa efectuarea si incarcarea temelor de catre elevi, profesorii le pot corecta oricand. Corectarea temelor se poate realizeaza prin citirea lor si oferirea de recenzii personalizate fiecarui elev, cu exemple specifice din tema sa, incluzand si nota.
5. Beneficii pentru studenti:
6. Flexibilitatea in accesarea materialelor educationale oferite de profesori: Studentii pot vizualiza si descarca resursele materiale oferite de profesori si disponibile in orice moment.
7. Incarcarea temelor: Elevii pot incarca temele efectuate asigurandu-se ca lucrarile lor urmeaza sa fie evaluate de profesori si sa primeasca comentarii clare si precise cu greselile facute si sugestiile primite de catre cadrele didactice. De asemenea elevii dispun de flexibilitatea si in efectuarea si incarcarea pe platforma a temelor rezolvate.
8. Evaluare: Dupa evaluare temelor, elevii pot citi notitele si sugestiile profesorilor in timp ce isi pot revedea lucrarea, cat si rezolvarea corecta a acesteia. De asemenea fiecare elev primeste nota si recenzia in mod privat.

## Structura lucrării

Un paragraf în care fiecare dintre secțiunile următoare este prezentată în 1-2 fraze, punând accentul pe elementele cele mai semnificative din fiecare secțiune.

# Analiza și specificarea cerințelor

Putem remarca faptul ca in Romania cultura meditatiilor este una foarte dezvoltata, fiecare parinte isi trimite copilul la meditatii macar in cei mai importanti ani din ciclul de invatamant al copiilor acestora. Un studiu la vinel national realizat de IRES si Societatea Academica din Romania spune ca „Conform studiului realizat de IRES și SAR în rândul părinților, o treime dintre elevi participă la sistemul de meditații, cu o frecvență considerabilă în clasele terminale, atunci când examenele naționale se apropie. În urma chestionării elevilor de gimnaziu și liceu, a rezultat faptul că un sfert dintre elevii din România accesează sistemul de meditații la cel puțin o materie.”[[1]](#footnote-1)



Comparand aceste informatii cu alte tari:

* In Italia 1 din 5 elevi fac meditatii conform unei statistici realizate pe 6000 de elevi de ripetizioni.it, platforma italiana pentru mediatii atat online cat si fizic.[[2]](#footnote-2)
* In germania la fel ca in italia 1 din 5 elevi se adreseaza profesorilor pentru meditatii in invatamantul liceal conform lucrarii stiintifice „Effectiveness of private tutoring during secondary schooling in Germany: Do the duration of private tutoring and tutor qualification affect school achievement?” [[3]](#footnote-3)

Din aceste statistici putem concluziona ca studentilor nu le sunt de ajuns orele de pregatire din cadrul scolilor. Astfel aceasta platforma ar actiona ca un pilon educational pentru student.

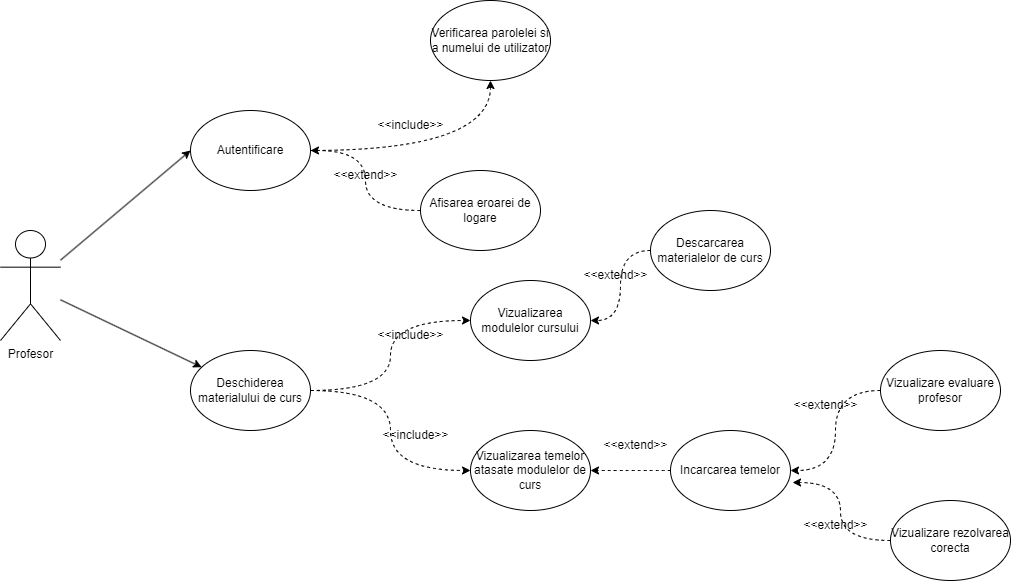
## Cerinte nonfunctionale

Cerintele nonfunctionale se refera la specificatiile care descriu cum ar trebui sa se comporte un sistem, acestea se concentreaza pe calitatea si performanta aplicatiei si sunt esentiale pentru a asigura o experienta optima pentru utilizatori.

1. Securitate: datele utilizatorilor sunt protejate la orice moment de timp
2. Performanta: timpii de raspuns si viteza aplicatiei sa nu fie un inpediment in folosirea acesteia
3. Viteza: Aplicatia trebuie sa raspunda cerintelor intr-un timp optim.
4. Fiabilitate: Aplicatia v-a functiona fara erori care sa afecteze disponabilitatea aplicatiei.

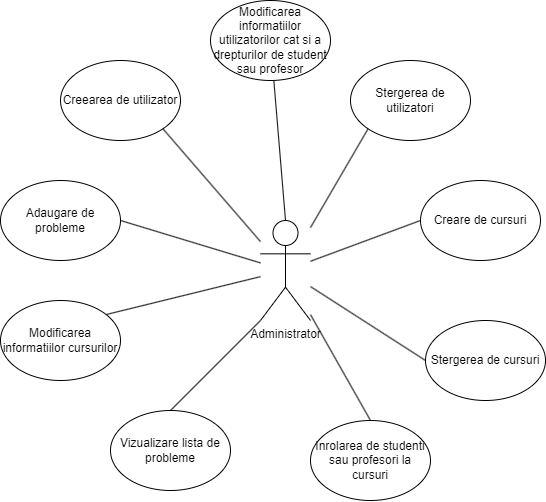
## Cerinte functionale

Cerintele functionale se axeaza pe cum aplicatia functioneaza daca atinge sau nu anumite caracteristici dorite. In aplicatia propusa sunt 3 roluri: student, profesor si administrator. In continuare voi prezenta procesul de folosire a aplicatiei atasand diagrame ale cazurilor de utilizare.



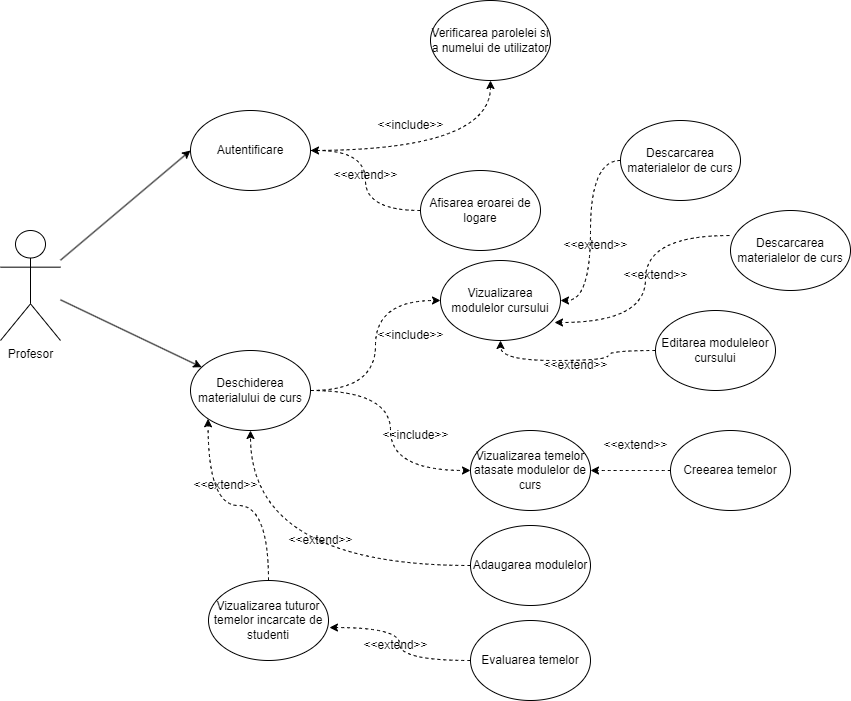
In rolul de student procesul este urmatorul:

1. Creearea contului: Oricarei creeare de cont nou ii este atribuit automat rolul de student.
2. Autentificarea: La autentificare studentul isi adauga credentialele urmand ca acestea sa fie verificate. Daca acestea sunt corecte utilizatorul este redirectat pe pagina unde sunt vizibile cursurile. Daca credentialele sunt incorecte acestuia ii va fi afisat un mesaj de eroare.
3. Inrolare la cursuri: Utilizatorul se poate inrola singur la cursurile la care crede ca are nevoie de studiu suplimentar.
4. Deschiderea materialului de curs: Dupa inrolarea la curs este accesibila deschiderea si privirea materialelor disponibile.
5. Vizualizarea modulelor cursului: Dupa deschiderea materialului de curs sunt vizibile modulele cursului. Un exemplu de modul: Titlul: „Polinoame”, Descriere: „Acest modul te va ajuta sa intelegi polinoamele” si un fisier atasat Suport curs: polinoame.pdf.
6. Descarcarea suportului de curs: Studentul poate descarca suportul de curs.
7. Vizualizarea temelor atasate modulelor de curs: Pentru fiecare modul cursantul poate avea atasat o tema.
8. Incarcarea temelor: Dupa ce studentul considera ca tema este rezolvata acesta poate incarca tema pentru evaluarea unui profesor.
9. Vizualizare evaluare profesor: Elevul poate observa comentariile profesorului asupra temei incarcate.
10. Vizualizare rezolvare corecta: Dupa evaluarea temei de catre profesor rezolvarea corecta a exercitiului este disponibila elevului.



In rolul de administrator procesul este urmatorul:

1. Creearea de utilizator: administratorul poate creea utilizatori
2. Modificarea informatiilor utilizatorilor cat si a drepturilor de student sau profesor: administratorul poate schimba informatiile utilizatorului de ex numele, rolul sau adresa de email.
3. Stergerea de utilizatori
4. Creearea de cursuri: administratorul poate creea cursuri
5. Modificarea informatiilor cursurilor: Administratorul poate schimba titlul, descrierea cat si categoria din care face parte cursul
6. Stergerea de cursuri
7. Inrolarea de studenti sau profesori la cursuri: Elevul se poate inrola si singur la cursuri dar profesorul se poate inrola doar daca este inrolat de administrator. Ratiunea din spatele acestei functionalitati este ca doar un administrator poate stii daca un profesor este bun la o materie. Astfel nu se v-a intampla de exemplu ca un profesor de Informatica sa corecteze si sa puna materiale la Geografie.
8. Vizualizare lista de probleme: Administratorul poate vizualiza lista de probleme, acesta poate edita problemele sau de asemenea sterge probleme.
9. Adaugare de probleme: Administratorul de asemenea poate adauga probleme.



In rolul de profesor procesul este urmatorul:

1. Editarea siglei de curs: profesorul poate schimba sigla la cursurile la care acesta este inrolat.
2. Adaugarea de probleme in lista de probleme: profesorul poate adauga probleme
3. Adaugarea modulelor cursurilor: acesta poate adauga module noi pentru curs
4. Editarea modulelor de curs: Acesta poate edita titlul modulului, descrierea, cat si poate sterge sau incarca suportul de curs aferent modulului.
5. Vizualizare lista de probleme: Profesorul poate vizualiza lista de probleme pentru a alege o problema din aceea lista pe care sa o dea ca tema.
6. Adaugare de probleme: Acesta poate sa adauge probleme.
7. Creare de tema: Profesorul poate creea teme alegand o problema din lista de probleme. Acesta poate de asemenea sa seteze un o zi limita pentru incarcarea acesteia.
8. Vizualizarea listei cu elevii care au incarcat temele: Profesorul poate vizualiza lista cu cei care au incarcat temele, apoi cu ajutorul unui modul de cautare acesta poate cauta dupa mai multe criterii cum ar fii titlul temei sau numele elevului.
9. Evaluarea temei: Acesta poate da o recenzie temelor incarcate cat si o nota.

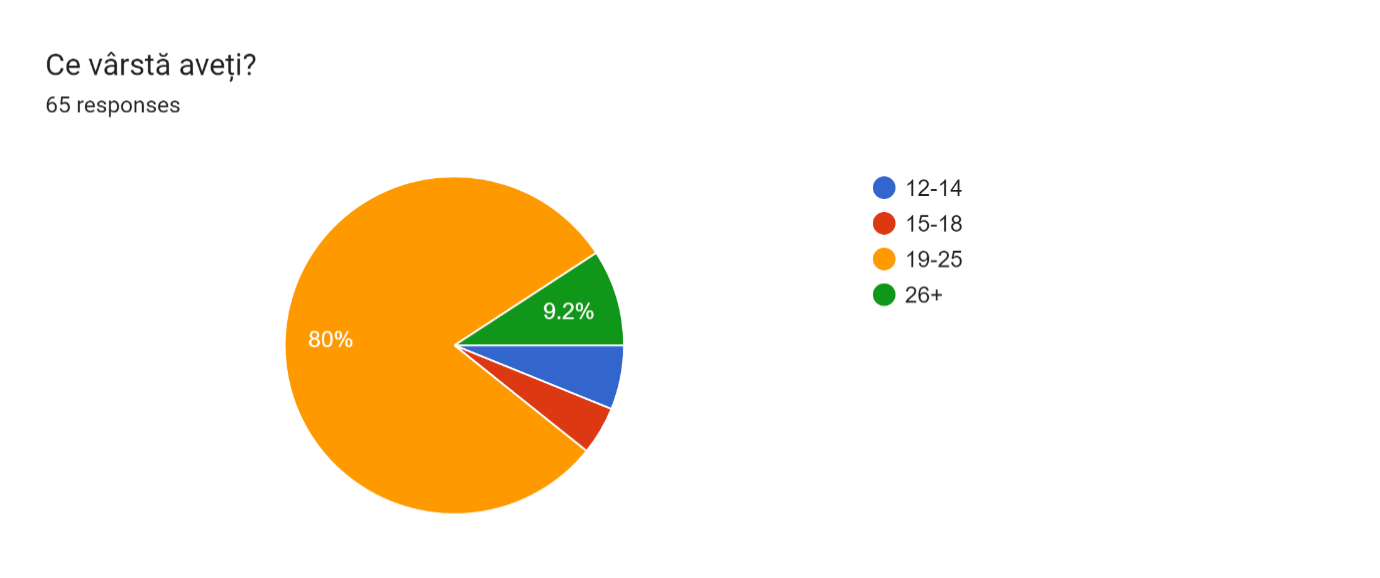
# Studiu de piață / Abordări existente

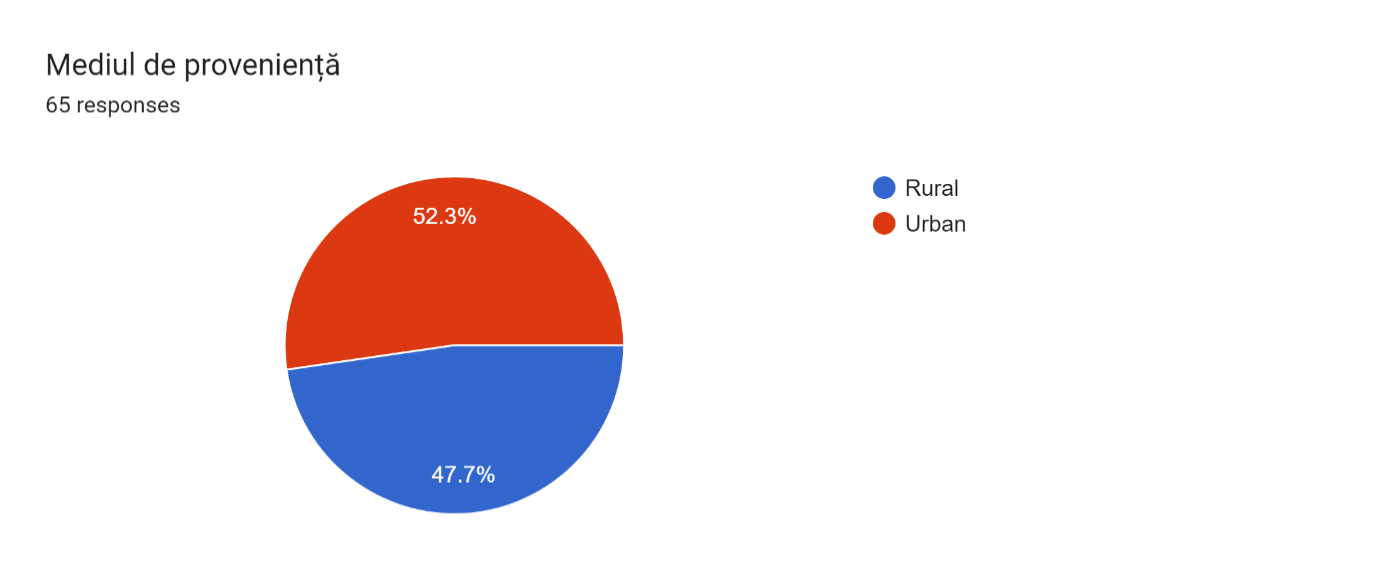
Obiectivul aparitiei unui produs nou pe piata trebuie sa se rezume in totalitate la rezolvarea problemelor potentialilor utilizatori, astfel incat fiecare aplicatie disponibila este orientata unui anumit grup tincta.

Aplicatia pe care o propun se orienteaza catre elevi si profesori, cu scopul facilitari procesului de invatare si de predare. Consider ca, la momentul actual, in special in Romania, ideea de meditatii este puternic raspandita printre elevi, de multi ori numarul cererilor depasind numarul ofertelor.

De altfel, din cauza contextelor globale din ultimii ani, profesorii au fost nevoiti sa se orienteze catre metode noi de invatare, mai exact platformele de invatare online, considerate poate chiar „nonconformiste” pana in acel moment. Pe parcurs, obisnuindu-se cu noile metode, din ce in ce mai multi profesori au inceput sa utilizeze platformele online de invatare, considerandu-le benefice.

In domeniul educatiei este nevoie tot timpul de metode noi si inovative care sa faciliteze invatarea. Astfel, pentru a realiza o platforma online de invatare suplimentara, utila si fiabila, am creat un chestionar orientat catre nevoile intampinate de elevi, afland de asemenea si parerea unui numar de 65 de persoane despre ideea de „platforma online de invatare”.

Figura 1

Figura 2

Cu ajutorul figurilor 1 și 2 putem observa fiabilitatea platformelor online de invatare, relevante si benefice persoanelor de diferite varste, fiind chestionate 65 de persoane cu varste cuprinde intre 12 si peste 26 de ani, provenite atat din mediul urban (52.3%) cat si din cel rural (47.7%)

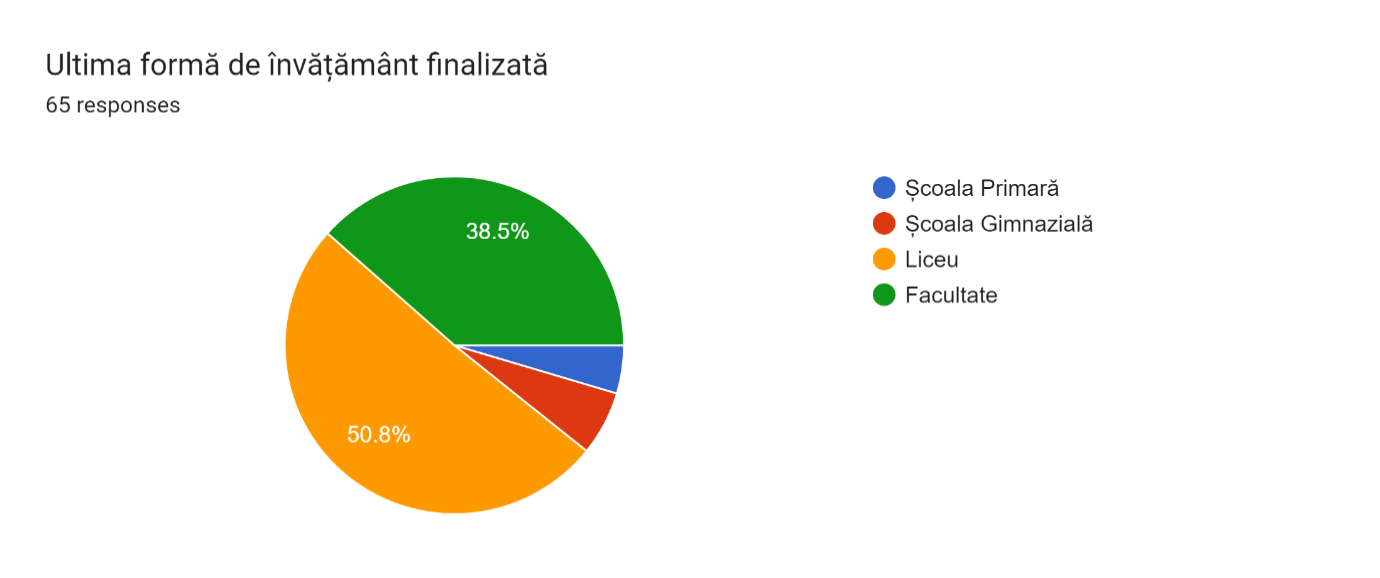


Figura 3

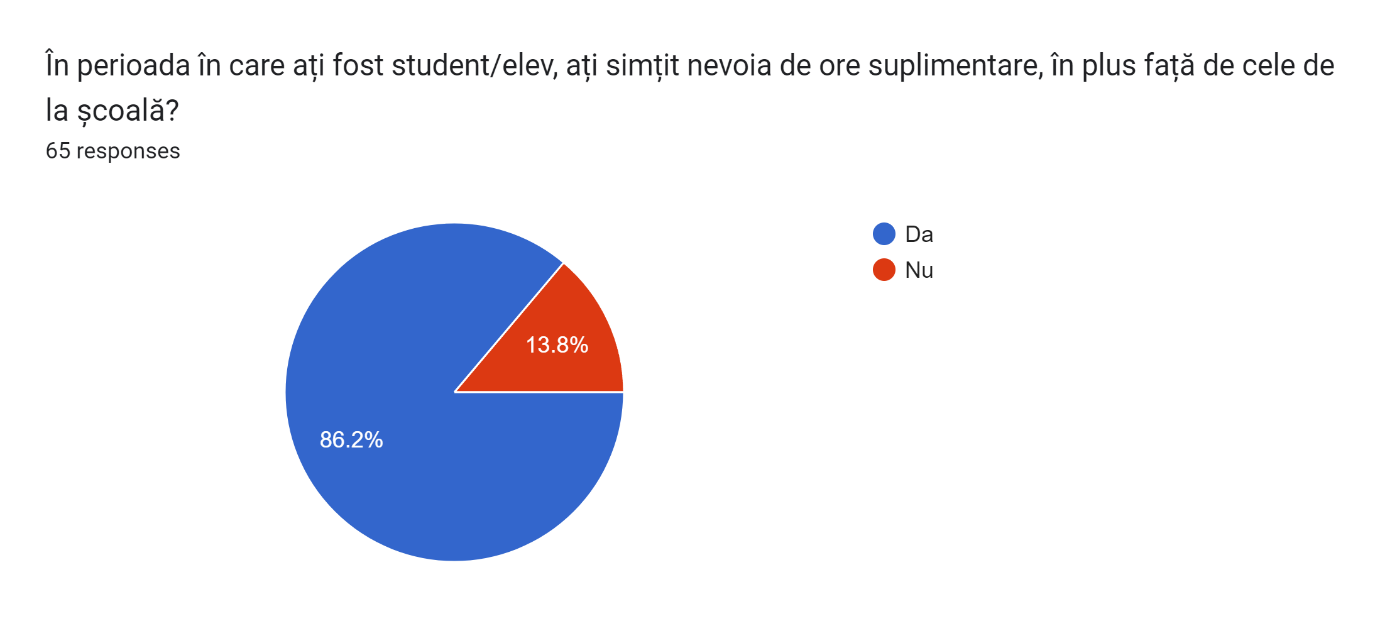


Figura 4

De altfel, cu ajutorul figurilor 3 si 4 observam faptul ca 86,2% dintre persoanele chestionate au simtit nevoia de ore suplimentare in perioada studiilor si doar 13,8% nu au simtit nevoia, persoanele intrebate avand ca ultima forma de invatamant finalizata scoala primara, scoala gimnaziala, liceu si facultate. Astfel, putem spune ca nevoia orelor suplimentare apare indiferent de forma de invatamant, indiferent daca urmeaza un examen important precum bacalaureatul sau nu.

Este evidentiata nevoia unei schimbari in randul elevilor si al studentilor, pentru facilitarea procesului de invatare al acestora si ajutorarea profesorilor carora le este greu ca intr-un curs sa ajunga la 100% numarul elevilor care au inteles si nu mai au nevoie de pregatire si explicatii in plus.

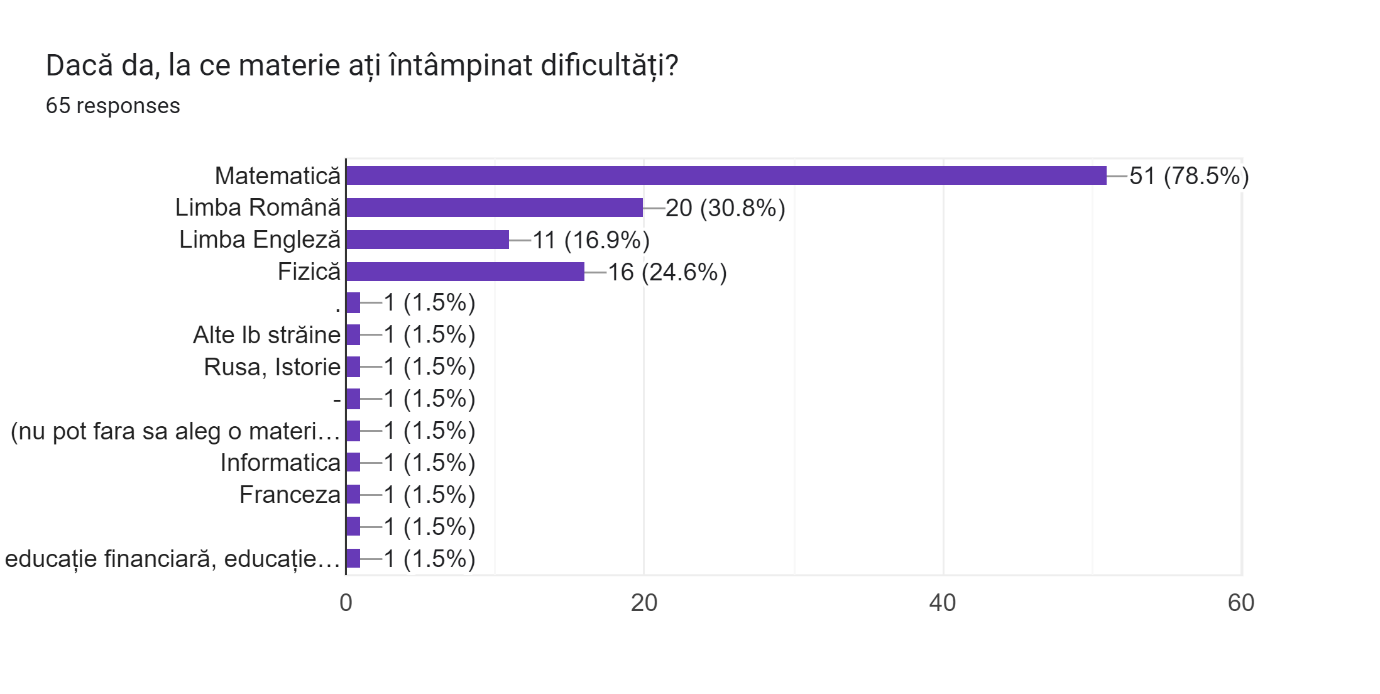


Figura 5

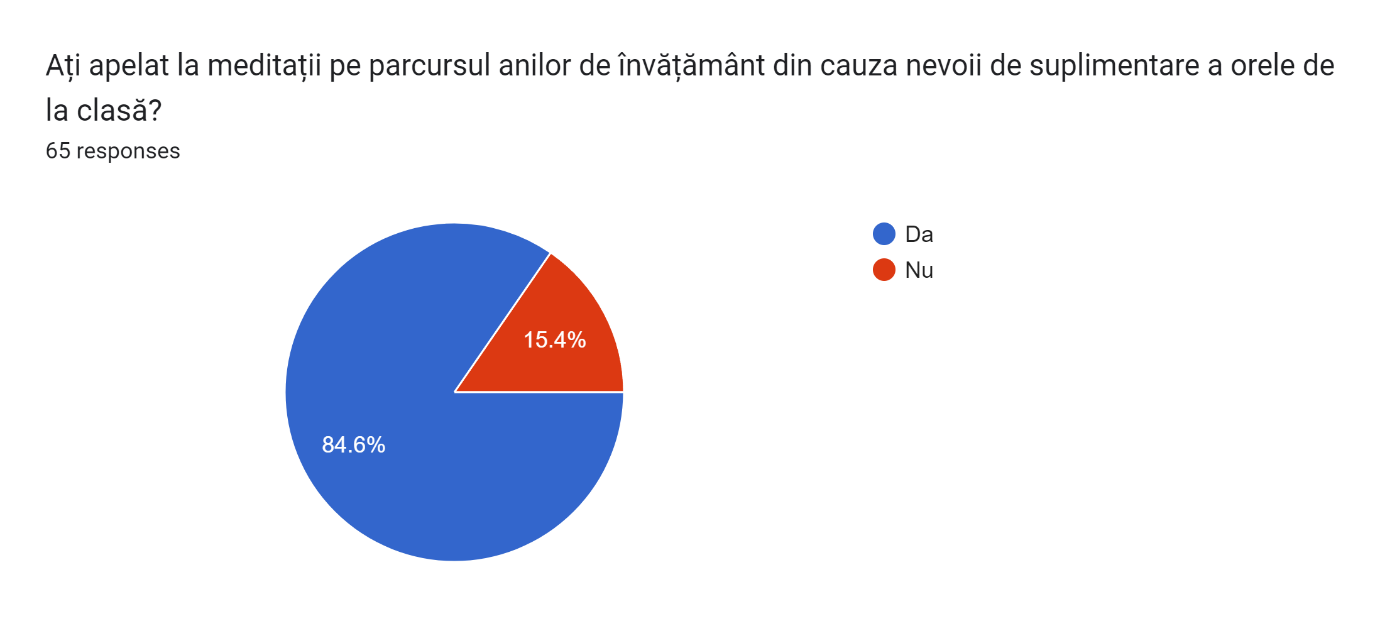


Figura 6

Din figura 5 reiese faptul ca doar 4 persoane din 65 nu au intampinat dificulatati in perioada scolii. Consider ca numarul persoanelor ce nu au avut probleme pe parcursul anilor de studii este unul extraordinar de mic si trebuie schimbat. Acest fapt m-a motivat in realizarea aplicatiei de invatare suplimentara si sper sa reusesc sa aduc o imbunatatire si un beneficiu atat elevilor cat si profesorilor.

De asemenea, privind figura 6 observam ca 84,6% dintre persoanele chestionate au apelat la meditatii din cauza nevoi de a suplimenta orele de la clasa si doar 15,4% nu au facut meditatii. Procentul persoanelor care au facut meditatii este unul foarte mare, de altfel si nevoia si de pregatire suplimentara este uriasa. Sustin faptul ca platforma de invatare suplimentara poate sprijini nevoile elevilor.

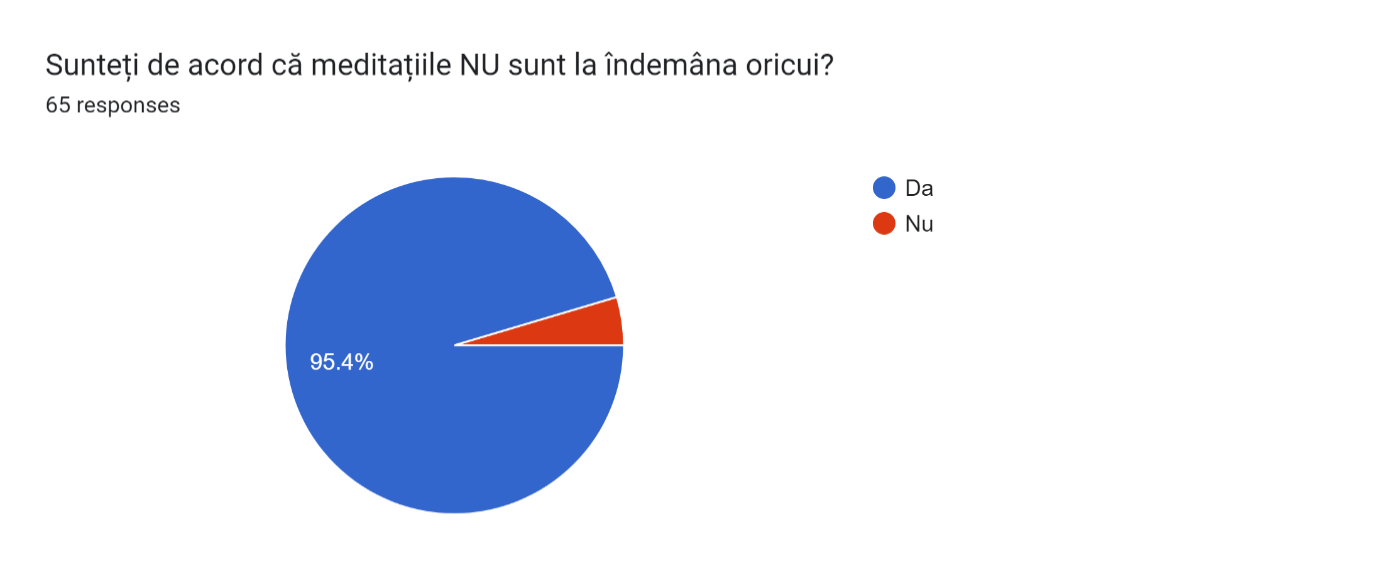


Figura 7

Figura 7 reprezinta un adevar foarte trist cu care, din pacate, multe persoane se confrunta. Meditatiile din nefericire nu sunt la indemana oricui, de altfel multi elevi chiar daca au nevoie de pregatire suplimentara nu pot avea parte de ea iar acest lucru ii poate impiedica in dezvoltarea lor.

Sunt de parere ca toata lumea ar trebui sa aiba acces la pregatire suplimentara in caz de nevoie si consider ca platforma prezentata poate ajuta prin scaderea numarului elevilor ce nu au acces la pregatire suplimentara si a celor care apeleaza la meditatii.

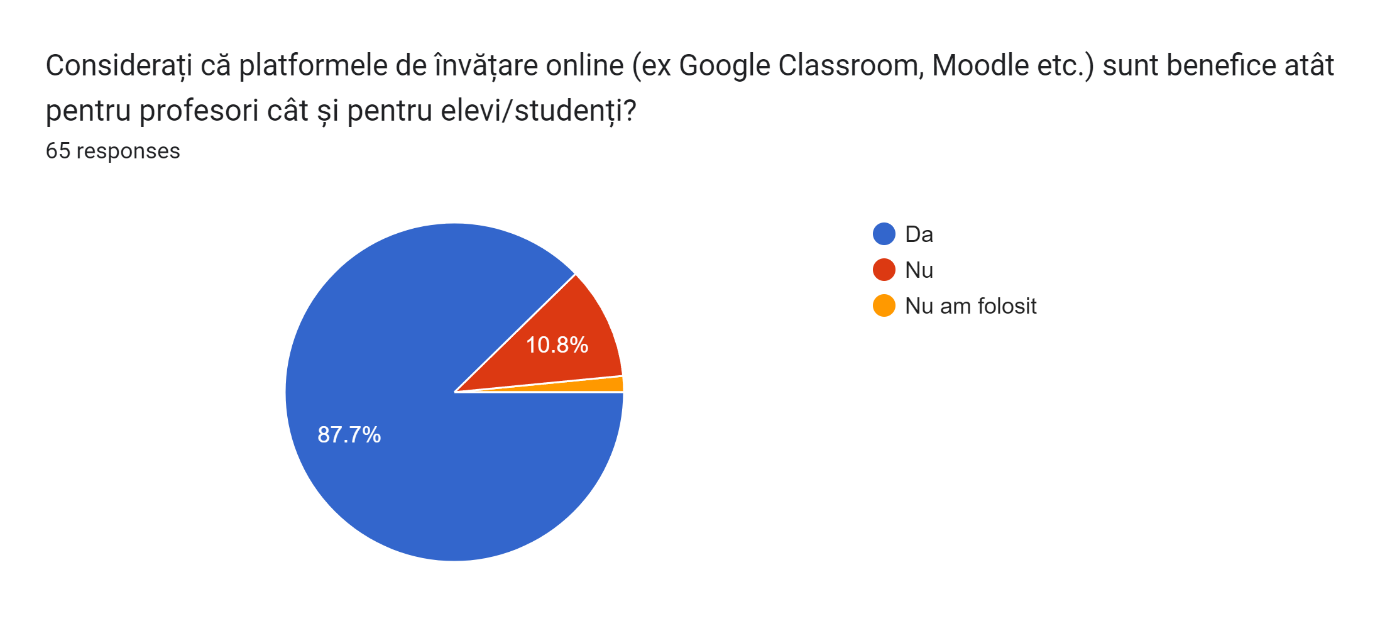


Figura 8

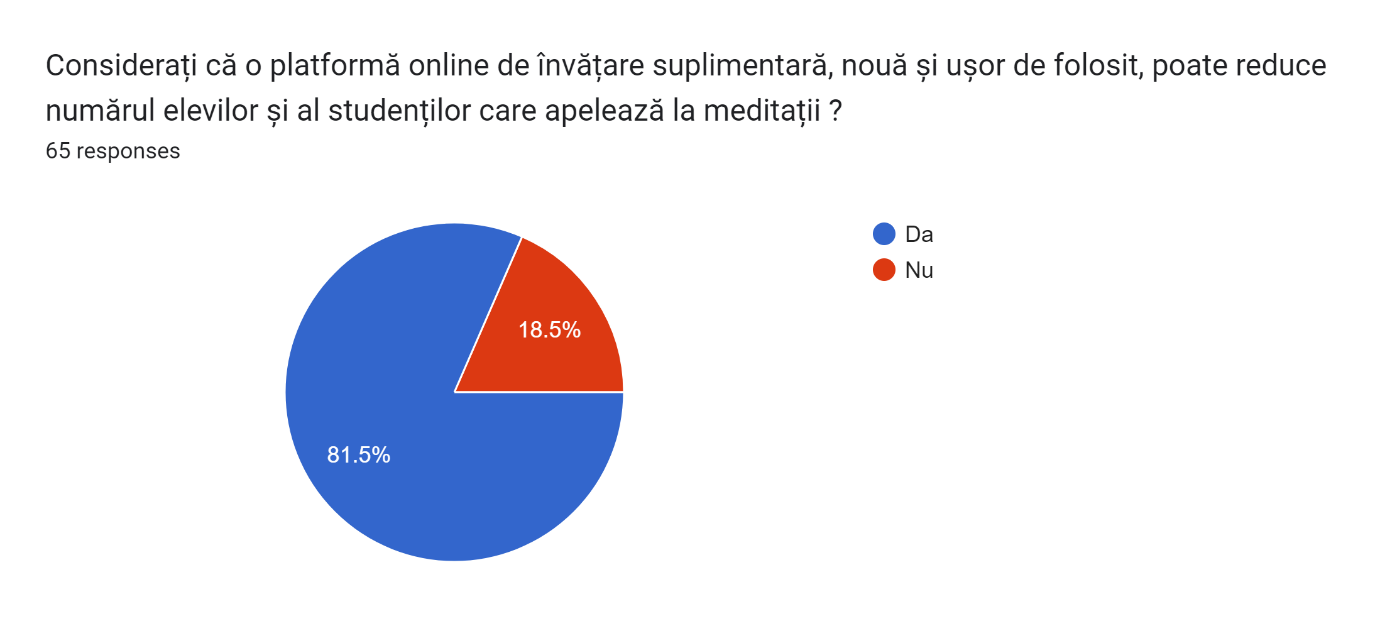


Figura 9

Analizand figurile 8 si 9 vedem parerea persoanelor chestionate referitor la platformele online si beneficiile lor. In figura 8 se observa ca doar 10,8% nu considera benefice aplicatiile de invatare online deja existente pe piata si 1,5% nu au folosit, ramanand un procentaj foarte mare de persoane ce considera benefice aplicatiile de educatie online.

De asemenea, in figura 9 este precizat faptul ca 81,5% dintre persoane sunt de parere ca o noua platforma online de invatare suplimentara poate reduce numarul elevilor care apeleaza la meditatii intarind ideea pe care o sustin si obiectivul pe care mi l-am propus in realizarea aplicatiei prezentate.

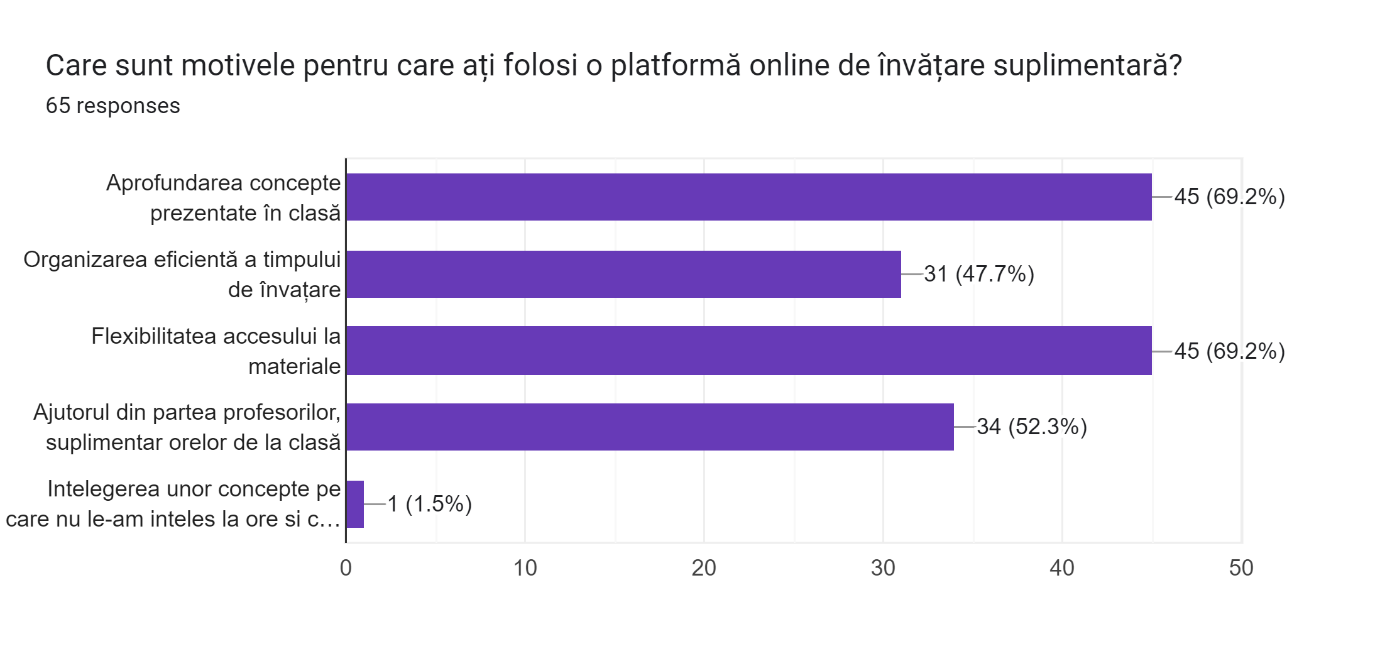


Figura 10

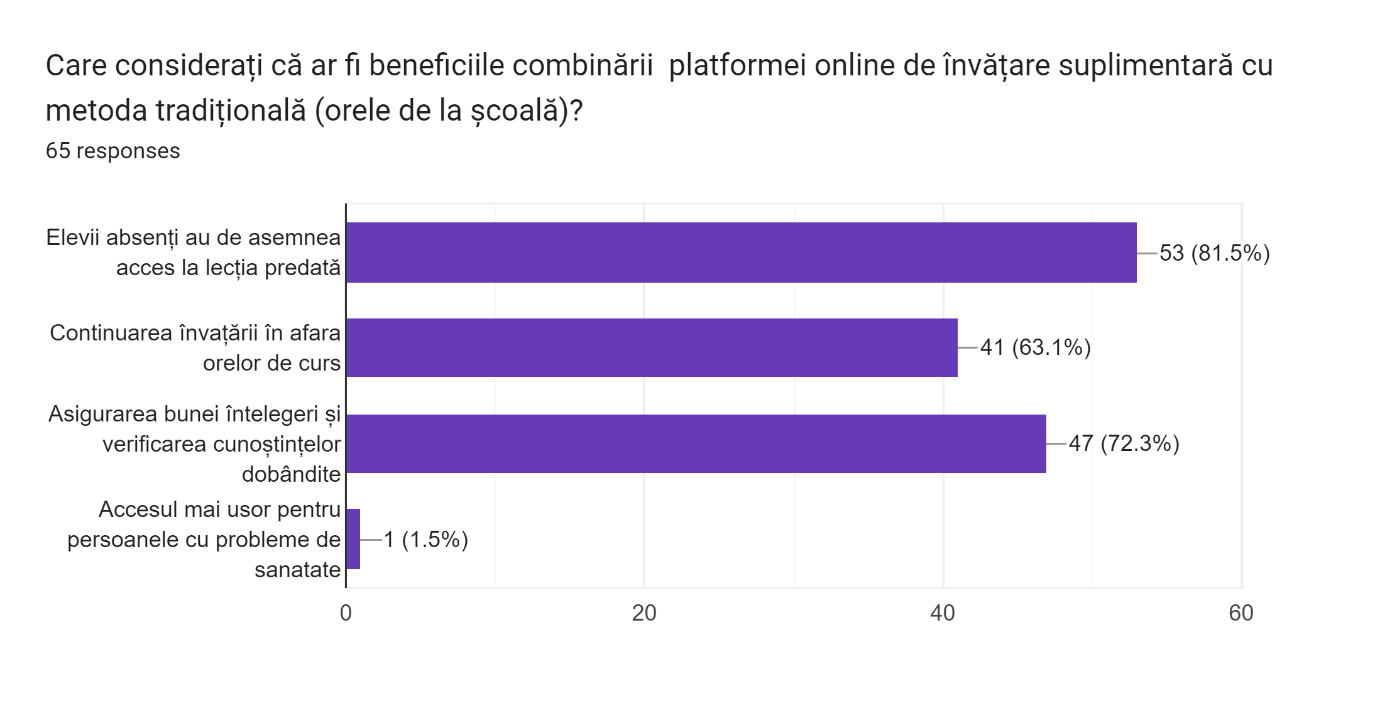


Figura 11

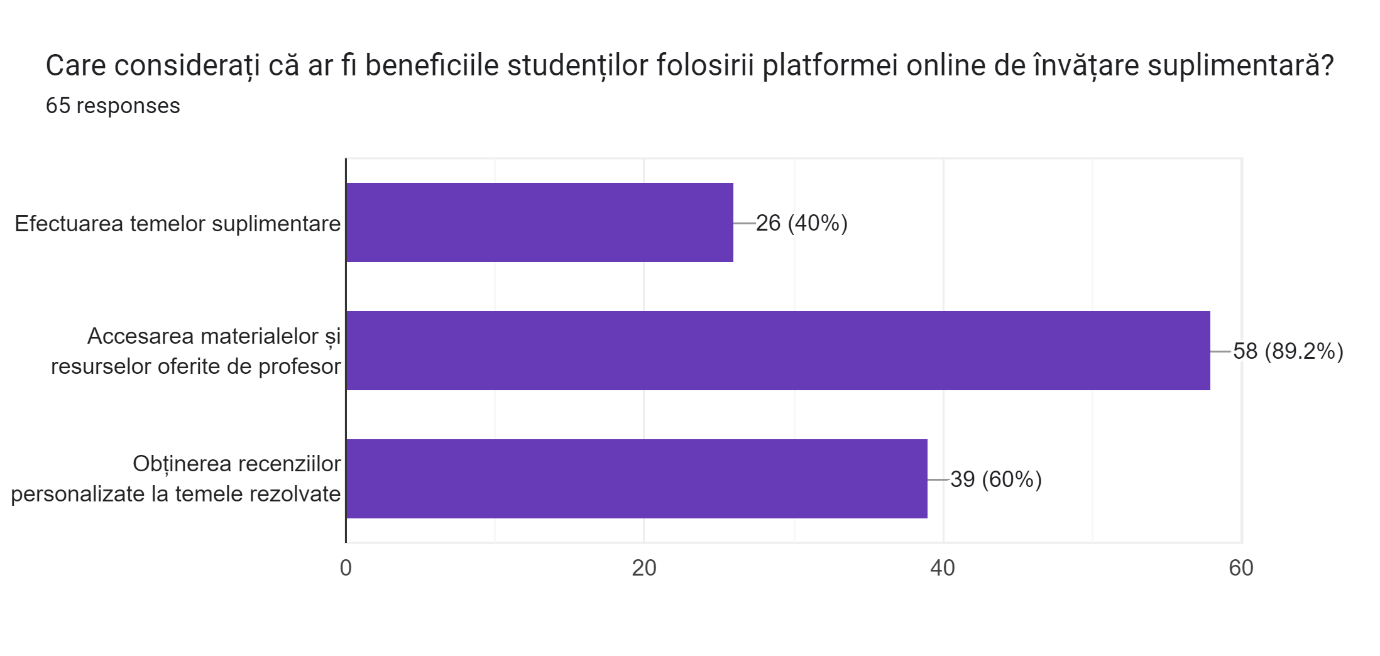


Figura 12

In prezent, in multe scoli si institutii de invatamant, profesorii folosesc platformele de invatare online pentru trimiterea temelor sau, mai rar, pentru postarea cursurilor din ziua respectiva. Insa, platforma pe care eu o propun vizeaza in special invatarea suplimentara si inlocuirea pe cat posibil a meditatiilor prin activitati, cursuri si explicatii suplimentare, ci nu strict pentru trimiterea temelor sau a lectiei de zi.

Consider ca o platforma online noua si usor de folosit poate fi benefica invatamantului si dezvoltarii persoanelor.

Observam in figura 10 ca niciuna dintre persoanele chestionate nu a mentionat faptul ca nu ar dori sa foloseasca o asemenea persoana, in plus raspunsurile primite sunt pro unei asemenea platforme deoarece nevoia invatarii suplimentare este mare.

De asemenea, figurile 11 si 12 se refera la beneficiile platformei, in persepctiva elevilor dar si a profesorilor. In plus, niciuna dintre persoanele chestionate nu a mentionat vreun dezavantaj in folosirea acestei noi platforme.

Pentru a exemplifica, analizam figura 11 care se refera la imaginea de ansablu si beneficiile pe care platforma le poate avea in invatamant, prin combinarea sa cu metoda de invatare traditionala, cea din salile de clasa. Persoanele chestionate considera un avantaj combinarea celor 2 metode de invatare, in vederea unei mai bune intelegeri a cursurilor, continuarea invatarii si comunicarea cu profesorii in afara programului, in plus, chiar sugestia anonima privind accesul persoanelor cu probleme de sanatate ofera o imagine de ansamblu asupra castigului utizilarii platformei. Se observa faptul ca verificarea cunostintelor dobandite are o mare importanta si este pe locul 2 in clasament, la mica diferenta de locul 1 care vizeaza accesul elevilor absenti la materialele si cursurile predate in ziua respectiva.

Ultima intrebare a chestionarului (figura 12) ii vizeaza strict pe elevi si avantajele pe care acestia le-ar avea utilizand platforma prezentata. De asemenea, fiecare persoana a ales drept raspuns minim 2 variante, rezultand beneficiile si suportul pe care cursantii le-ar avea prin folosirea platformei online de invatare suplimentara.

Consider ca, raspunsurile primite in acest chestionar imi intaresc si imi sprijina ideea de necesitatea a unei asemenea platforme, ce poate fi la indemana cat mai multor persoane si poate aduce ajutor invatamantului actual si elevilor ce se confrunta cu diferite probleme privind educatia, prin fiabilitatea sa, usurina folosirii din orice tip de profil si optiunile pe care aceasta le ofera profesorilor si elevilor.

# Soluția propusă

Abordarea pe care am ales sa merg provine din documentarea pietei la momentul actual. Astfel dupa o analiza amanuntita am observat ca o buna parte din marile companii precum JPMorgan Chase Bank, Intellibus, Deutsche Bank si multe altele folosesc Java Spring Boot pentru backend. Pentru frontend am ales sa merg pe Angular folosind typescript cu toate ca React este mai popular in randul aplicatiilor mai mici dar pentru proiectele mari este mai folosit Angular. Pentru baza de date am ales sa merg pe o baza de date relationala, respectiv MySql, aceasta fiind open-source, de asemenea fiind cunoscuta pentru performanta si scalabilitate.

## Spring Boot

Spring este cel mai popular framework de java. Titulatura de cel mai popular framework este obtinuta din viteza, simplitatea si productivitatea acestuia.

Spring este un framework pentru a construi aplicatii java, open-source, care ofera o infrastructura complexa si usor de mentinut pentru dezvoltarea aplicatiilor. Principalele avantaje in folosirea Spring:

1. Inversion of control – acesta este un principiu de proiectare care ofera modularitate programului, oferind frameworkului control asupra unor portiuni de cod sau a unor obiecte. Acest lucru este realizabil prin Dependency Injection. Dependency Injection este mecanismul prin care controlul este inversat de la programator la framework.
2. Accesul la date – simplificarea accesului la date folosind tehnologii precum Hibernate, JPA (Java persistence API), MyBatis, EclipseLink. Acestea reduc complexitatea interactiunii cu baza de date.
3. Ideal pentru construirea de RESTful API datorita adnotarilor simpliste: @RestController, @GetMapping, @PostMapping simplifică definiția și gestionarea. Aceste adnotari fac codul usor de citit si intretinut permitand o dezvoltare usor facila a enpointurilor de tip CRUD (create, read, update, delete).
4. Securitatea – securitatea este usor de facilitat datorita frameworkului Spring Security JWT (json-web token) care ajuta in autentificare si autorizare. De asemenea ofera protectie impotriva atacurilor de tip CSRF

### Acces la date

In solutia mea am ales sa folosesc JPA impreuna cu hibernate. JPA fiind o solutie de a controla accesul la baza de date iar hibernate implementarea acestuia. Aceasta abstractizeaza toata complexitatea implementarii in spatele unor interfete si reguli.

JPA si hibernate sunt un mod de programare object-relational mapping, acesta fiind un mod de a alinia codul cu baza de date. Se structureaza foarte bine datorita modului de programare orientat pe obiecte, fiecare clasa din codul java este mapata la un tabel.

Am ales sa folosesc acest model la acces de date fata de un model ca MyBatis pentru optimizarea si abstractizarea pe care o aduce jpa si hibernate acestea avand incorporat caching.

Cu toate ca MyBatis ofera mai mult control asupra interogarilor sql, am decis ca automatizarea ORM este de ajutor pentru mine datorita faptului ca mentinerea interogarilor este mult mai usor de facilitat.

### Securitate

Pentru partea de securitate folosesc Spring Security si json-web token pentru autentificare si autorizarea diferitelor roluri din aplicatie. Odata ce userul se autentifica acestuia ii este generat un jwt token care este trimis din frontend inapoi, in fiecare cerere pe care acesta o face, pentru a confirma ca acesta este insusi userul pentru care tokenul a fost generat.

Explicație Detaliată

Autentificare: Utilizatorul introduce credențialele (de exemplu, nume de utilizator și parolă) și trimite o cerere de autentificare către server.

Serverul verifică credențialele utilizatorului. Dacă autentificarea este reușită, se generează un token JWT care poate contine informatii despre utilizator si rolurile acestuia. Aceste informatii se numesc claims.

Tokenul JWT este semnat folosind o cheie secretă pentru a asigura integritatea și autenticitatea acestuia.

Tokenul JWT generat este trimis către frontend și stocat de obicei in memoria locala a browserului sau un cookies.

Autorizare:

La fiecare cerere ulterioara din frontend este trimis alaturi de informatiile cererii in antet tokenul. Serverul verifica tokenul asupra validitatii, daca tokenul este valid acesta permite cererea altfel nu trimitand o eroare.

## Angular

Angular este un framework open-source bazat pe TypeScript, care rulează pe Node.js și este utilizat pentru crearea aplicațiilor web.

De ce TypeScript și nu JavaScript?

Unul dintre cele mai mari avantaje pe care consider ca le are typescript asupra javascript este pentru ca acesta are erori de compilare datorita tipizarii statice in loc de tipizarea dinamica in javascript.

Caracteristici ale Angular

Angular este bazat pe componente. Componentele ofera utilizatorului abilitatea de a defini propriile componenete personalizate. Aceste pot fi folosite simultan cu elementele HTML standard, precum <div></div>. Un exemplu de folosire al acestei caracteristici este creearea unei componente <add-problem></add-problem> care este o pagina cu un formular pentru adaugare a exercitiilor. Aceste poate fi folosit ca un „obiect” in orice alta componenta a aplicatiei. Astfel acest stil de programare este similar cu programarea orientata pe obiecte, aducand toate beneficiile ale acestuia cum ar fi modularitatea, reutilizarea, intretinerea si posibilitatea de extindere.

Printre beneficiile majore ale utilizării componentelor se numără:

* Organizare și Claritate: Codul este mult mai bine organizat și clar, facilitând înțelegerea și navigarea.
* Reutilizarea Codului: Componentele pot fi reutilizate în diferite părți ale aplicației, reducând redundanța și economisind timp.
* Întreținere Facilă: Codul modular și componentizat este mai ușor de întreținut și actualizat, permițând implementarea mai rapidă a modificărilor și a noilor funcționalități.

## MySQL

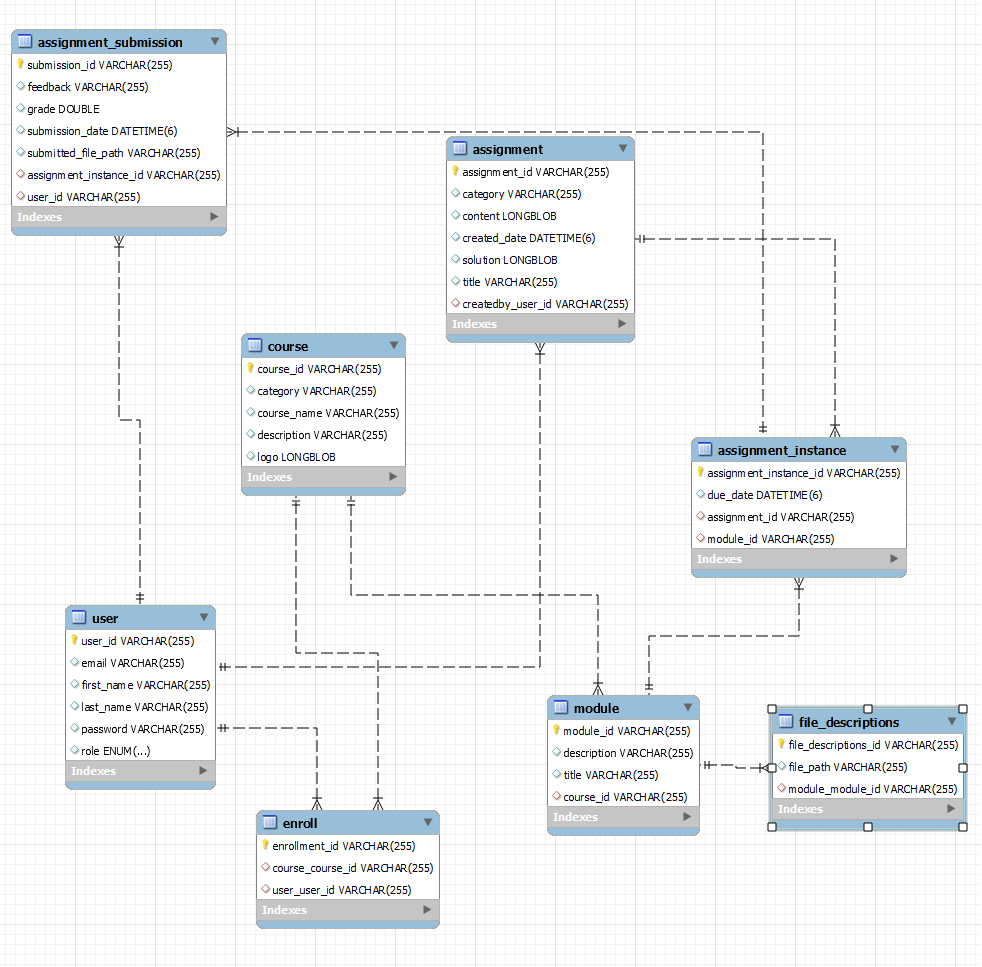
MySQL este un sistem de gestionare a bazelor de date relaționale (RDBMS) open-source. Este unul dintre cele mai populare și utilizate motoare de baze de date din lume, cunoscut pentru performanța sa, fiabilitatea și ușurința în utilizare. MySQL este dezvoltat, distribuit și susținut de Oracle.

## Open-Source

Am mentionat faptul ca alegerile mele in construirea aplicatiei sunt open-source. Astfel voi argumenta care sunt avantajele:

1. Transparenta: dezvoltatorii pot inspecta codul sursa pentru a observa daca acesta este functional, de asemenea acesta este un factor bun si din punct de vedere al securitatii.
2. Securitatea: datorita transparentei dezvoltatorii pot analiza codul de vulnerabilitati ascunse. Legea lui Linus Torvalds „Given enough eyeballs, all bugs are shallow" – aceasta inseamna cu cat mai multa lume lucreaza si testeaza aplicatia problemele aplicatiei dispar.
3. Valoare educationala
4. Suport si comunitate: datorita faptului ca este open-source suportul si comunitatea este pe masura fiind foarte usor sa gasesti informatii. De asemenea toti dezvoltatorii pot aduce contributii.
5. Inovare si dezvoltare rapida: datorita faptului ca toata lumea poate contribuii duce la dezvoltarea rapida si aducerea a noi inovatii in tehnologie.

## Arhitectura bazei de date



1. Tabela user: Aceasta contine informatiile despre utilizatori. Coloanele sunt: user\_id cheie primara, email first\_name last\_name, password si role.
2. Tabela course: Aceasta contine informatiile despre cursuri. Coloanele sunt course\_id cheie primara, category, course\_name, description si logo.
3. Tabela module: Aceasta contine informatii despre modulele din curs, fiecare modul fiind legat de un curs. Coloanele sunt module\_id cheie primara, description, title si course\_id cheie straina.
4. Tabela file\_description: Aceasta contine caile pentru fisierele din module. Coloanele sunt: file\_descriptions\_id cheie primara, file\_path, module\_module\_id cheie straina.
5. Tabela assignment: Aceasta contine informatii despre exercitiile care pot fi folosite ca teme. Coloane: assignment\_id cheie primara, category, content, created\_date, solution, title si creadedby\_user\_id cheie straina
6. Tabela assignment\_instance: Aceasta tabela contine informatii despre teme, fiind legata de module. Coloane: assignment\_instance\_id cheie primara, due\_date, assignment\_id cheie straina, module\_id cheie straina
7. Tabela assignment\_submission: Aceasta tabela contine informatii despre temele incarcate ale studentilor. Coloane: submission\_id cheie primara, feedback, grade, submission\_date, submitted\_file\_path, assignment\_instance\_id cheie straina si user\_id cheie straina.
8. Tabela enroll: In aceasta tabela este retinut ce utilizator este inregistrat la un anumit curs. Coloane enrollment\_id cheie primara, course\_course\_id cheie straina si user\_user\_id cheie straina.

## Docker

Docker este o platformă open-source care automatizează deployment-ul aplicațiilor, permițând rularea acestora într-un mediu izolat. Docker facilitează împachetarea unei aplicații într-un singur container, ceea ce contribuie la principiile de dezvoltare continuă și livrare continuă.

În aplicație vor fi trei servicii: frontend, backend și baza de date, orchestrate printr-un fișier docker-compose.yml. Fiecare serviciu rulează într-un container propriu și este interconectat printr-o rețea comună. Astfel, vor rula trei containere:

1. mysqldb: Containerul care rulează MySQL.
2. Spring Boot RESTful Web Services: Containerul care rulează backend-ul aplicației, configurat pentru a se conecta la baza de date și la frontend.
3. frontend: Containerul care rulează serverul Angular pentru partea de frontend.

De asemenea, utilizez o rețea comună, springboot-mysql-net, pentru a permite containerelor să comunice eficient între ele.

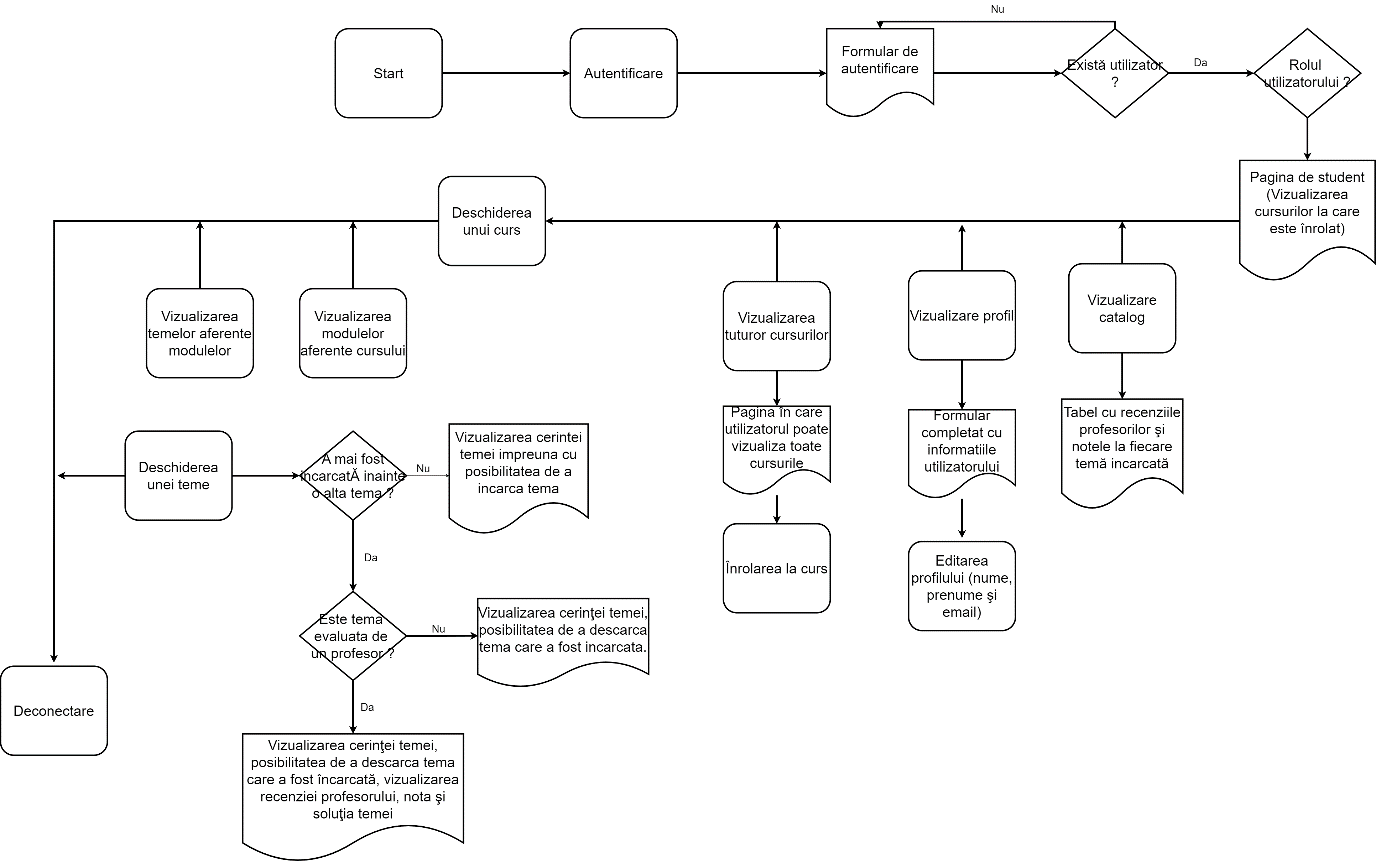
## Principiul dezvoltare continua si livrare continua

Obiectivul principiului dezvoltare continua si livrarea continua este de a creste rapiditatea cu care este dezvoltata o aplicație si de a îmbunătății calitatea produsului finit. Principalul atu al acestui principiu este faptul ca îmbunătățește experiența dezvoltatorului dar si a consumatorului aplicației, aplicația fiind livrata către acesta in cicluri stabilite de timp permițând o mai buna comunicare intre aceștia. Consumatorul poate propune schimbări in aplicație la orice moment de timp, iar dezvoltatorul poate implementa aceasta schimbare si o poate livra într-o noua iterație.

Astfel acest principiu se pliază cel mai bine pe un mod de dezvoltare iterativ.

## Descrierea fluxului funcționalităților

Am ales să folosesc o diagrama simpla pentru a evidenția punctele critice de decizie si pentru a ilustra fluxul funcționalităților. Aceste diagrame sunt vitale pentru înțelegerea modului în care diferitele părți ale sistemului lucrează împreună pentru a îndeplini anumite funcționalități.



In aceasta diagrama sunt evidentiate toate functionalitatile pe care le poate efectua un utilizator in rolul de student. La autentificare daca credentialele sunt corecte acesta este redirectionat pe o pagina pe care poate vizualiza toate cursurile la care este introlat. Descrierea fluxului funcţionalitaţilor din pagina initiala:

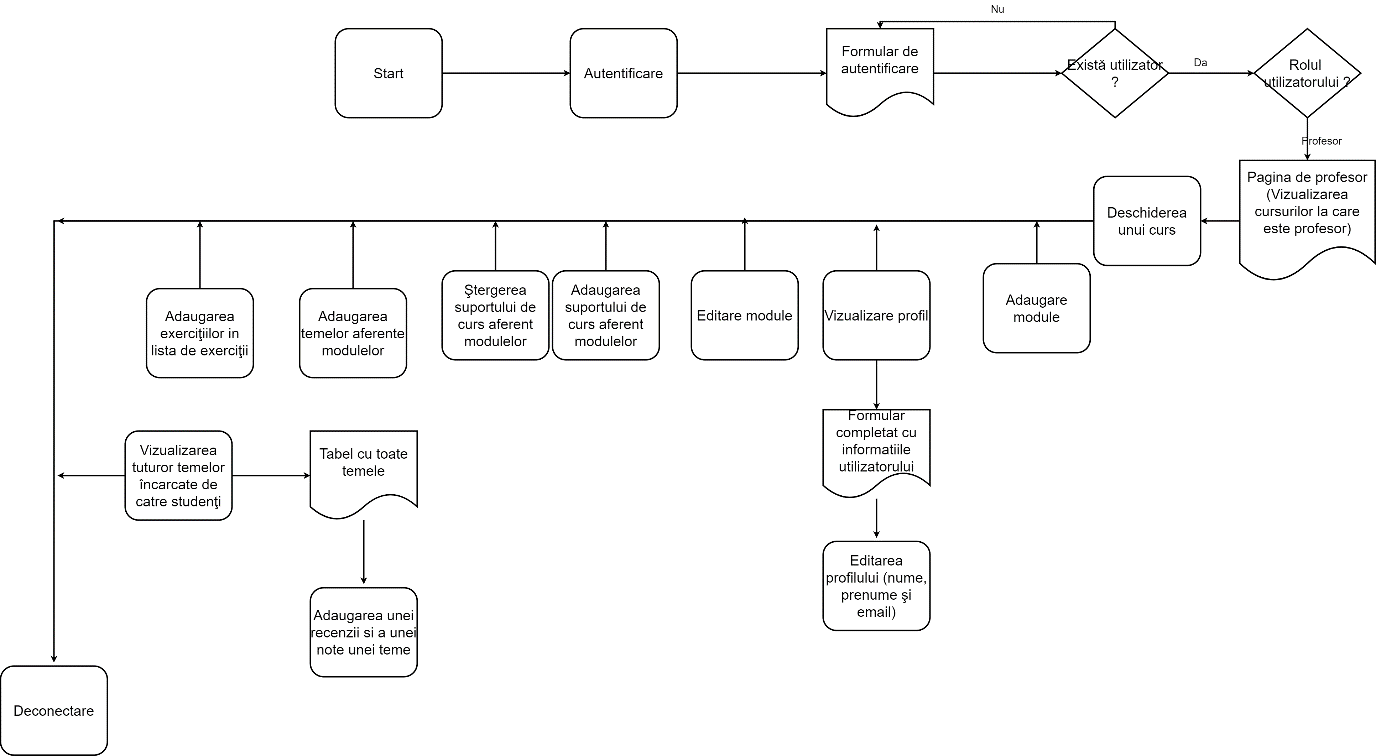
* Vizualizarea catalogului cu posibilitatea de a cauta recenziile profesorilor la un anumit curs
* Vizualizarea profilului cu posibilitatea de a modifica elemente ale profilului cum ar fi nume, prenume si email
* Vizualizarea tuturor cursurilor cu posibilitatea de introlare la orice curs care apare

Descrierea fluxului functionalitatilor dupa deschiderea unui curs la care studentul este inrolat:

* Vizualizarea modulelor de curs
* Vizualizarea temelor

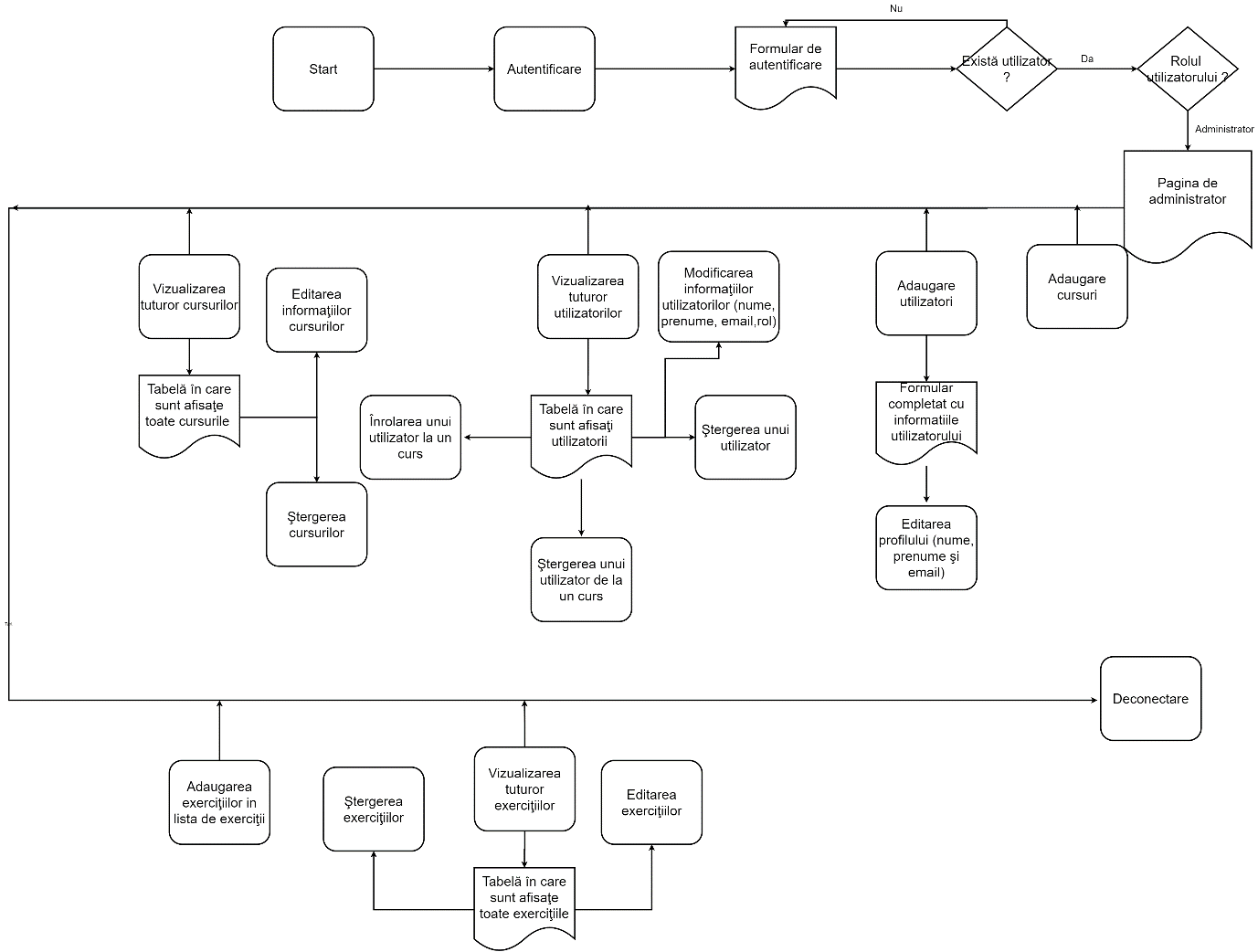
La deschiderea unei teme:

* Daca o tema nu este incarcata acesta vede cerinta si are posibilitatea de a incarca.
* Daca este deja o tema incarcata de catre care inca nu este evaluate de un profesor student acesta poate vedea cerinta temei, ultima incarcare, posibilitatea de a descarca ultima incarcare si posibilitatea de a incarca pentru a modifica ultima incarcare.
* Daca tema este evaluata de un profesor studentul poate observa recenzia profesoruli, nota cat si posibilitatea de a vizualiza cerinta, incarca o alta tema si descarca ultima incarcare.



In aceasta diagrama sunt evidentiate toate functionalitatile pe care le poate efectua un utilizator in rolul de profesor. Daca profesorul deschide un curs poate realiza urmatoarele actiuni:

* Adauga un modul
* Editarea unui modul
* Stergerea unui modul
* Adaugarea suportului de curs aferent modulelor
* Stergerea suportului de curs aferent modulelor
* Adaugarea temelor aferente modulelor
* Adaugarea exercitiilor in lista de exercitii
* Vizualizarea listei de exercitii, modificarea si stergerea exercitiilor din lista la care pentru o mai usoara folosinta exista un sistem de cautare specializat
* Vizualizarea tuturor temelor incarcate de catre studenti unde apare un tabel cu toate temele incarcate pentru cursul respectiv la care profesorul poate cauta daca doreste sa corecteze temele cu un anumit titlu



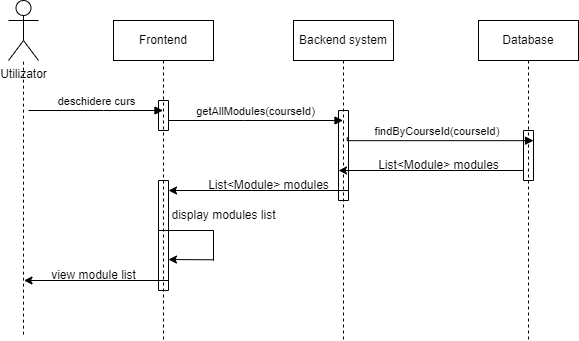
In aceasta diagrama sunt evidentiate toate functionalitatile pe care le poate efectua un utilizator in rolul de administrator.

* Adaugarea cursurilor
* Adaugarea utilizatorilor
* Vizualizarea tuturor utilizatorilor de unde administratorul poate edita informatiile utilizatorului(nume, prenume, email si rol), inrola profesori si utilizatori la anumite cursuri, sterge utilizatori de la anumite cursuri si sterge utilizatori
* Vizualizarea tuturor cursurilor de unde administratorul poate edita informatiile cursului si sterge cursul
* Vizualizarea listei de exercitii, modificarea si stergerea exercitiilor din lista la care pentru o mai usoara folosinta exista un sistem de cautare specializat
* Adaugarea exercitiilor in lista de exercitii

Interacțiunile dintre utilizator, front-end, back-end și managerul de autentificare sunt ilustrate în diagrama de secvență a fluxului de autentificare din figura X. Utilizatorul completează câmpurile (email și parolă) și apasă pe submit. Din front-end se trimite o cerere de tip POST către back-end. Back-end-ul folosește Authentication Manager pentru a verifica credențialele în baza de date.

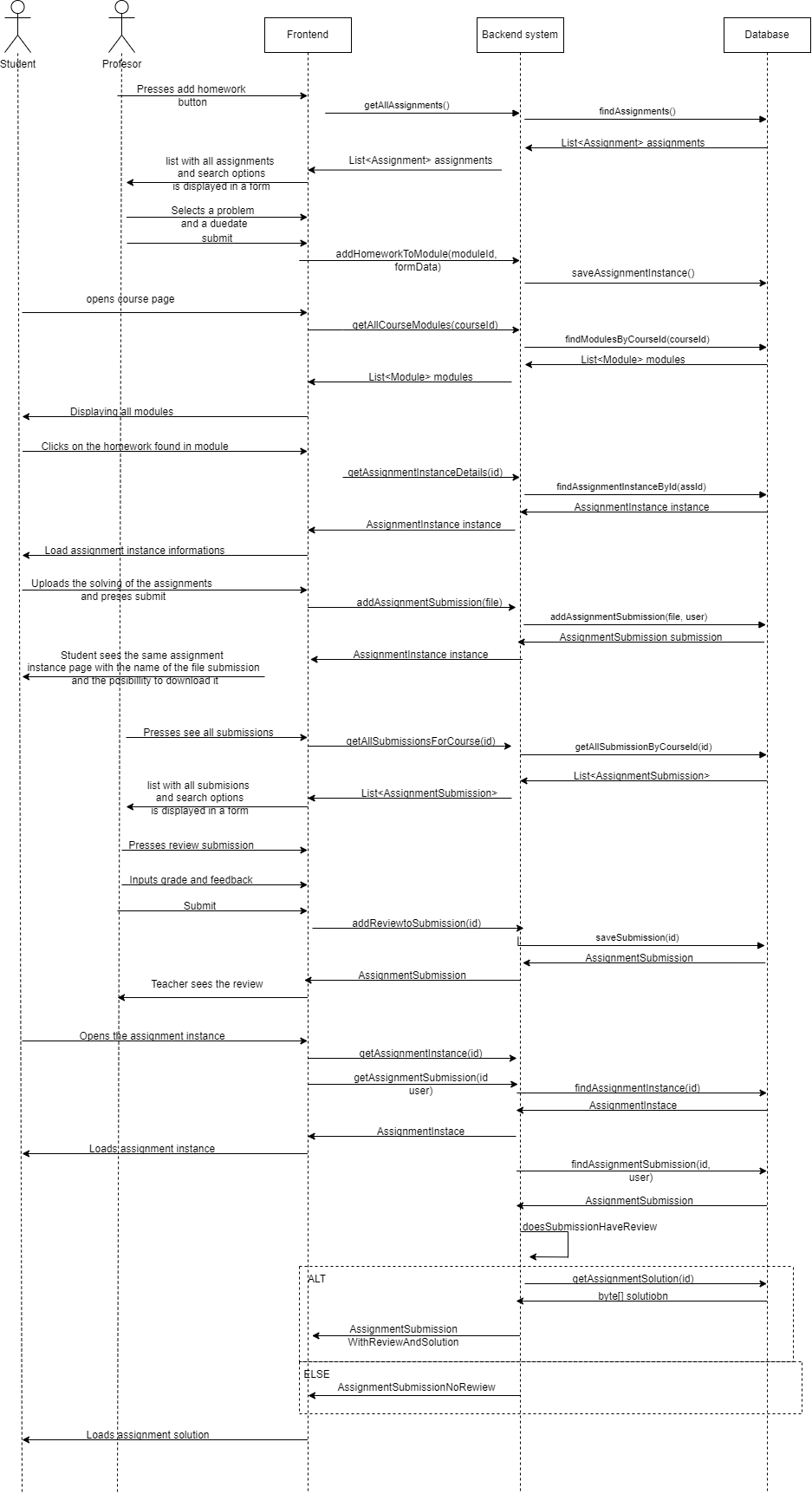
Dacă credențialele sunt corecte, se întoarce un obiect de autentificare. În back-end se generează un token JWT care conține proprietățile utilizatorului, cum ar fi rolul și emailul. Front-end-ul redirecționează utilizatorul pe pagina specifică rolului acestuia: pentru student și profesor pe pagina principală unde sunt afișate cursurile, iar pentru admin pe pagina specială pentru administrator si tokenul este salvat in local storage-ul browserului.

Dacă credențialele sunt incorecte, se întoarce un mesaj HTTP "Bad credentials", iar utilizatorului i se afișează un mesaj de eroare.



Interacțiunile dintre utilizator, front-end, back-end și baza de date sunt ilustrate în diagrama de secvență a fluxului de accesare a unui curs din figura X. Utilizatorul deschide cursul pentru a vizualiza lista de module disponibile. Din front-end se trimite o cerere de tip GET către back-end. Back-end-ul folosește baza de date pentru a găsi modulele asociate cursului specificat.

Dacă cererea este validă, back-end-ul interoghează baza de date pentru a găsi toate modulele asociate cursului cu ID-ul specificat. Baza de date returnează o listă de module către back-end. Back-end-ul trimite lista de module către front-end, care o afișează utilizatorului.

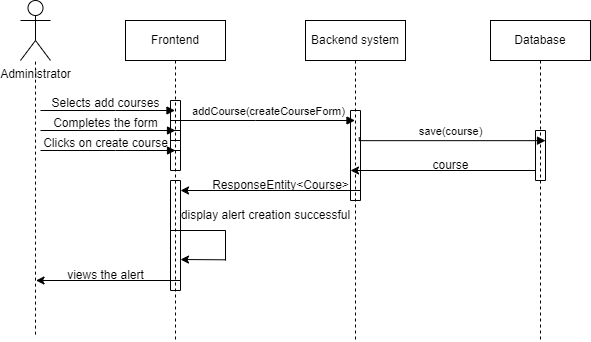


In diagrama din figura x este prezentata interactiunea a diferitelor roluri, student si profesor, cu sistemul. Profesorul este in pagina de curs, acesta apasa pe butonul add homework aflat in unul dintre module. Ii este afisat un formular in care are un modul de cautare si toate problemele existente in lista de exercitii. Acesta alege unul dintre exercitii dupa care selecteaza un termen limita pentru incarcarea temei si apasa pe butonul add homework, pagina de curs este updatata, fiind vizibila noua tema propusa.

Studentul deschide pagina de curs, vede tema propusa de profesor intr-un modul acesta apasa click pe tema. O noua pagina se deschide. Aceasta contine informatiile temei. Elevul termina tema si se decide sa o incarce. Apasa pe butonul upload homework alege fisierul si da submit.

Profesorul apasă pe butonul „See all submisions”, astfel acesta este redirectat pe o pagina care conține toate încărcările studenților. Apasă pe butonul de review, se deschide o pagina cu formular de unde acesta poate descarcă ceea ce a încărcat studentul. Acesta completează recenzie si apasă pe butonul de submit.

Studentul deschide tema, acum este disponibila recenzia profesorului, nota cat si rezolvarea corecta a temei. Daca nu ar fi fost încă adăugată o recenzie acesta avea vizibil decât ultima lui încărcare pe care acesta putea sa o descarce si sa vadă ce se afla.



Doar utilizatorii cu rol de administrator pot crea cursuri. Acestia cand se autentifica sunt redirectionati automat catre o pagina la care doar administratorii au acces. De altfel pentru adaugarea unui curs completeaza informatiile cursului cum ar fi denumirea, categoria si descrierea si apasa pe butonul de creeaza. Se face o cerere de tip POST catre backend prin care se trimit informatiile completate in formular. Se salveaza in baza de date, iar apoi se introarce un obiect de tip ResponseEntity cu STATUS 201 si cursul care a fost adaugat. Apoi utilizatorul vede in aplicatie un mesaj „Cursul a fost adaugat cu succes”.

# Detalii de implementare

In acest capitol vom explora in detaliu intrand amanuntit in provacarile intampinate pe parcursul dezvoltarii aplicatiei. Pentru a intra mai in amanunt si a oferi o perspectiva clara asupra problemelor ma voi folosi de prezentarea secventelor de cod esentiale, pseudo-cod, diagrame de secventa si de resursele gasite in timpul cercetarilor. Resursele prezinta diferite perspective incluzand articole, studii de caz si documentatii tehnice folosite pentru a privi din perspective diferite solutiile problemei.

În plus, capitolul va detalia modul în care studiul de caz si tehnologiile propuse au fost de ajutor in dezvoltarea solutiei finale, accentuand modul in care ele au contribuit la rezolvarea problemelor. De asemenea vom analiza strategiile de arhitectura abordate la diferite impedimente in structura aplicatiei.

Prin prezentarea detaliată a algoritmilor și structurilor utilizate, împreună cu etapele de dezvoltare și corectitudinea soluțiilor implementate, acest capitol va oferi o imagine completă și transparentă a procesului de dezvoltare tehnică.

## Autentificare si securitate

### Pentru a intelege metoda pe care am ales-o in autentificarea si autorizarea utilizatorilor in aplicatie, cat si elmente ce tin de securitate voi prezenta mai multe optiuni.

#### Autentificare pe baza cookie-urilor

Autentificarea pe baza cookie-urilor functioneaza in urmatoarea maniera:

1. Utilizatorul completeaza email-ul si parola si trimte o cerere catre server de tip post.
2. Serverul verifica validitatea credentialelor. Daca acestea sunt corecte creeaza o sesiune pe care o retine si trimite id-ul sesiunii alaturi de infromatiile aferente intr-un cookie catre client.
3. La fiecare cerere ulterioară, clientul trimite automat informațiile din cookie. Astfel, serverul poate verifica dacă utilizatorul este autorizat sau nu.

Avantajele Autentificării pe bază de Cookie-uri

* Gestionare Automată: Cookie-urile sunt administrate automat de browser, simplificând astfel codul pe partea de client.
* Validare și Invalidare Facilă: Deoarece serverul menține starea, acesta poate valida sau invalida sesiunile cu ușurință.

Dezavantajele Autentificării pe bază de Cookie-uri

* Scalabilitate: Serverul trebuie să păstreze starea sesiunilor, ceea ce face dificilă scalarea verticală.
* Blocaje la trafic mare: Administrarea sesiunilor pe server poate duce la blocaje în cazul unui trafic mare de cereri.

### OAuth 2.0

OAuth 2.0 este un protocol de autorizare care permite aplicațiilor terțe să obțină acces limitat la resursele utilizatorilor fără a necesita trimiterea de credențiale.

Pasii prin care se realizeaza autorizarea sunt urmatorii:

1. Clientul este redirecționat către o aplicație terță aleasă, unde utilizatorul introduce credențialele, dacă nu este deja autorizat pe acea aplicație.
2. Aplicatia terta autorizeaza autentificarea si utilizatorul este redirectionat catre aplicatia prinicipala impreuna cu un cod de autorizare.
3. Clientul creeaza o cerere de tip POST ce contine codul de autorizare de la aplicatia terta pentru a obtine un token de acces in aplicatie.
4. La fiecare cerere ulterioara tokenul de acces este trimis in antetul cererii, astfel se face autorizarea.

Avantajele autentificării pe bază de OAuth 2.0

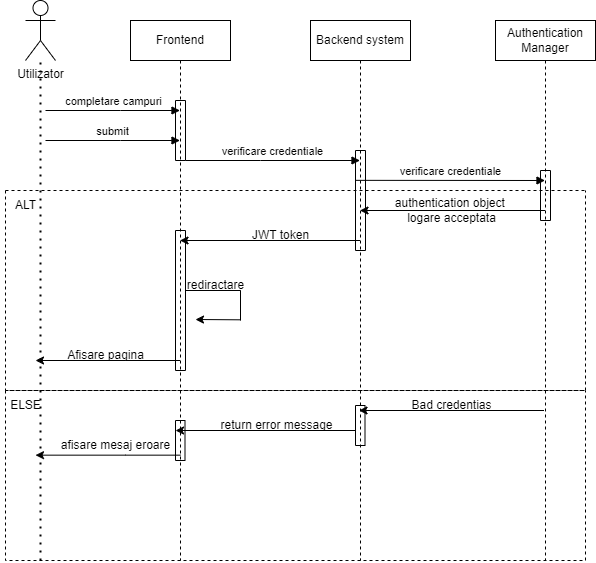
* Autentificarea numai este strans legata de credentialele utilizatorului
* Suportă diverse tipuri de fluxuri de autorizare si de tokeni cum ar fi opaque tokens, SAML assertions, JWT tokens.

Dezavantajele Autentificării pe bază de Cookie-uri

* Cea mai mare vulnerabilitate pe care o poate avea acest tip de autorizare este ca codul de autorizare din pasul 2 sa fie interceptat de un atacator.
* Se bazeaza pe o aplicatie externa pentru a gestiona autorizarea.

### JWT token

Autentificarea pe baza JSON Web Token este o metoda fără stare ce permite transmiterea de informații intre server si client intru-un mod securizat prin intermediului unui token semnat. Pentru a descrie pașii de autentificare și autorizare folosind JSON Web Token, voi utiliza o diagramă de secvență (FIGURA X).



Astfel utilizatorul completează câmpurile (email si parola) şi apasă autentificare. Clientul trimite o cerere de tip POST ce conține credenţialele utilizatorului către server. Serverul utilizează un Authentication Manager care facilitează verificarea credenţialelor in baza de date. Daca credenţialele sunt corecte, acesta returnează un obiect de autentificare, iar clientul primește un jwt token criptat ce conține proprietățile utilizatorului (email, rol, data expirarii tokenului). Clientul îl salvează în memoria locala a motorului de căutare si îl trimite ulterior cu fiecare cerere pe care clientul o face. Dacă credenţialele sunt incorecte Authentication managerul întoarce HTTP STATUS BAD CREDENTIALS, iar utilizatorului ii este afișat un mesaj de eroare.

Avantajele autentificării pe bază de OAuth 2.0

* Scalabilitate: Datorită faptului că este statless, nu necesită menţinerea informaţiilor de către server, ceea ce face aplicatia scalabila.
* Performanta: Tokenul poate conține informații (claims), eliminând astfel necesitatea unor interogări frecvente în baza de date, cum este cazul autentificării folosind sesiuni cookie.
* Interoperabilitate: Poate fi utilizat cu orice limbaj de programare, iar datorită faptului că este statless se pliază foarte bine pe principiile RESTful
* Securitate: Token-urile sunt semnate digital asigurând integritatea datelor

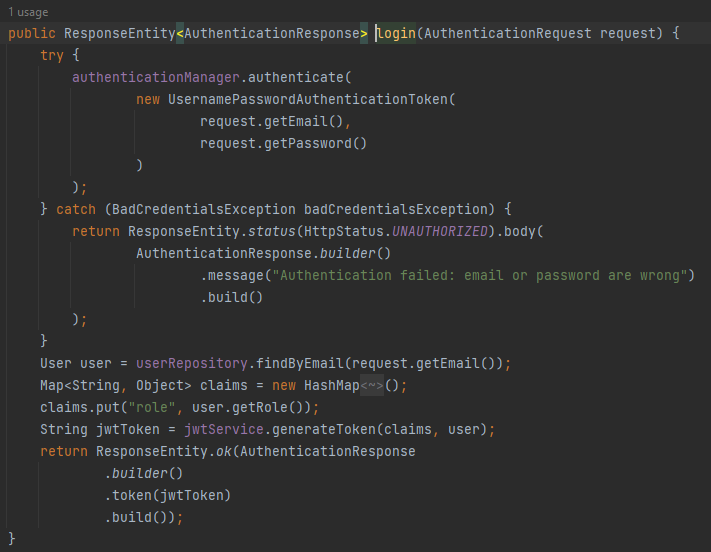
Dezavantajele Autentificării pe bază de Cookie-uri

* Revocarea token-ului înainte de expirare este complicată. Soluțiile includ utilizarea unei liste de revocare sau emiterea de token-uri cu durată scurtă de viață.
* Securitate: Deoarece payload-ul (claims) JWT nu este criptat, ci doar semnat, informațiile sensibile nu ar trebui incluse in token.

Rezultatul analizei mele au fost ca utilizarea JSON Web Token este cea mai buna alegere pentru această aplicație.



Principala configurație a securității se află în imaginea din figura x. Astfel metodă securityFilterChain configurează CORS, definește regulile de autorizare și adaugă filtrul de autentificare JWT înainte de UsernamePasswordAuthenticationFilter. Acest filtru este o clasă JwtAuthenticationFilter care extinde OncePerRequestFilter ce asigura faptul că filtrul este apelat o singură dată pe cerere și suprascrie metodă doFilterInternal care verifică antetul „Authorization” pentru a extrage și valida tokenul JWT.



În figura x este este prezentată metodă în care ajunge cererea de la client când utilizatorul încearcă să se autentifice. După verificarea credentialelor dacă sunt valide este trimis un token ce conține rolul și alte detalii cum ar fii dată expirării către client.

În serverul de Angular salvez tokenul în baza de date locală a browserului și am creat un interceptor prin care trece fiecare request și adaugă în antetul cererii tokenul.

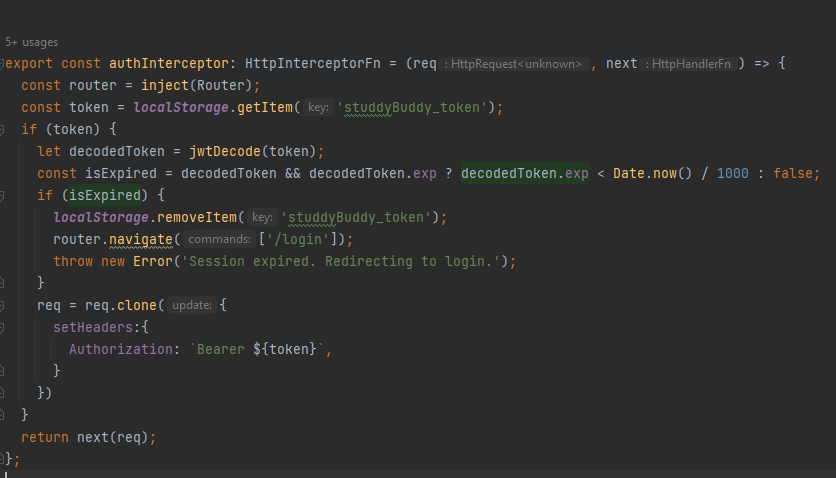


Figura x reprezintă codul sursă pentru adăugarea antetului ce conține tokenul JWT la fiecare cerere pe care o face către server. Tokenul este luat din local storage-ul browserului, se face verificarea dacă token-ul este expirat utilizatorul este trimis către pagină de login și îi apare mesajul de eroare aferent, iar dacă nu este expirat cererea este clonată și adăugat token-ul.

## Accesul la date

O alta problema pe care am întâmpinat-o este găsirea celui mai optim mod pentru aplicația mea in legătura cu accesul la date. In Spring Boot am găsit mai multe opțiuni pe care le voi prezenta si voi justifica alegerea făcută.

### JDBC Template

JDBC este o interfata care defineste cum clientii pot accesa baza de date. Aceasta metoda de a interactiona cu baze de date este orientata catre bazele de date relationale si ofera metode pentru a face cereri, actualizari, cereri si apeluri de proceduri in baza de date.

JdbcTemplate este o clasa centrala in pachetul JDBC care simplifica utilizarea acestuia si ajuta la evitarea erorilor comune cum ar fi gestionarea exceptiilor.

Avantaje:

* Mai performant pentru operațiuni de bază.
* Suporta cereri si procedure stocate

Dezavantaje:

* Scrierea a mult cod pana la executarea unei cereri cum ar fi, creeare conexiunii, crearea cereri, inchiderea conexiunii.
* Repetarea codului de la o baza de date la alta implica timp.

### MyBatis

MyBatis este un framework de persistenta care suporta SQL personalizat. Acesta ofera suport pentru maparea SQL si XML la obiecte java.

Avantaje:

* Flexibil pentru cereri complexe.
* Este mai flexibil cand vine vorba de arhitectura bazei de date.

Dezavantaje:

* MyBatis solicita maparea manuala a campurilor obiectelor la rezultatele cererilor SQL.
* Mai putin orientat pe obiecte.

### Spring Data JPA

Spring Data JPA este un modul al framework-ului Spring ce simplifică accesul la baze de date folosind API-ul Java Persistence (JPA). JPA este o specificație Java care descrie o metodologie de mapare a datelor între un model de obiecte și o bază de date relațională. Hibernate este una dintre cele mai populare implementări ale JPA și oferă un set complet de funcționalități pentru managementul persistenței datelor.

Avantaje:

* Simplificare și Reducerea Codului: Elimină necesitatea de a scrie manual codul de acces la date, reducând astfel complexitatea și cantitatea de cod necesar.
* Permite crearea de metode în repository care generează automat interogări pe baza numelor metodologice.
* Suport pentru Query-uri Personalizate: Permite scrierea de query-uri personalizate folosind JPQL (Java Persistence Query Language) sau SQL nativ.
* Caching: Hibernate, ca implementare JPA, oferă mecanisme de caching pentru îmbunătățirea performanței aplicațiilor prin reducerea numărului de interogări la baza de date.

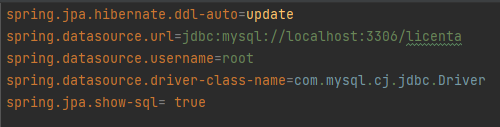
Dezavantaje:

* Abstracție Ridicată: Nivelului de abstractizare poate face dificila rezolvarea erorilor în comparație cu alte metode.
* Performanță: În unele cazuri, performanța poate fi afectată datorită mecanismelor de gestionarea entităților și a tranzacțiilor.

Astfel, am ales sa folosesc JPA in special datorita faptului ca face o mapare a datelor intre obiecte si baza de date relationala automat cu implementarea hibernate pentru a adauga si nivelul de caching pentru performanta. De asemenea, suportul pentru Lazy Loading a fost de ajutor în contextul în care unele tabele conțin BLOB-uri, aceste date fiind aduse doar când este necesar.

### Implementarea JPA

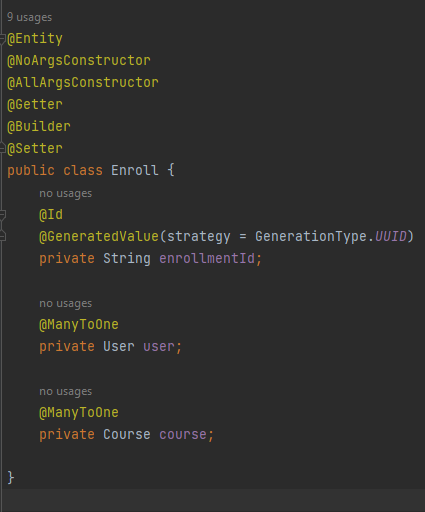
Pentru a configura o aplicație Spring Boot care folosește JPA și Hibernate cu o bază de date MySQL, este nevoie de câteva setări în fișierul application.properties.



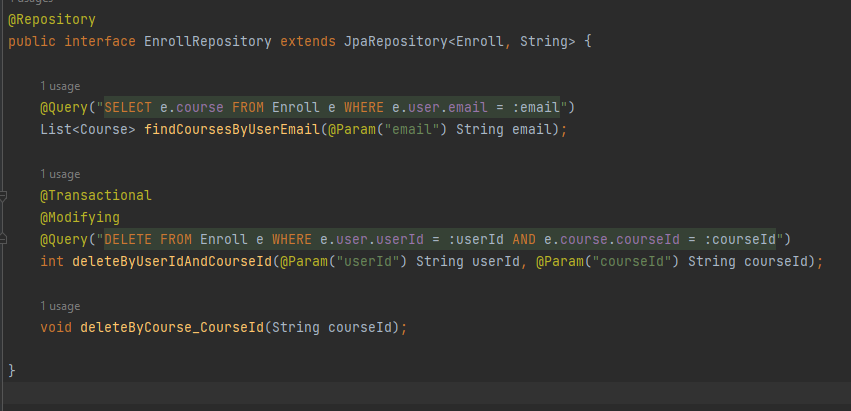
Explicația setărilor:

* Spring.jpa.hibernate.ddl-auto=update Aceasta setare specifica strategia Hibernate pentru actualizarea schemei bazei de date. Opțiunea „update” va actualiza schema bazei de date la pornirea aplicației, adăugând tabele noi sau modificând tabelele existente pentru a se potrivi cu modelul obiectului Java.
* Restul setărilor țin de conexiunea cu baza de date.

Un exemplu de entitate JPA care utilizează Hibernate pentru persistenta:



Un repository este o componenta ce oferă un mecanism pentru accesarea, manipularea si interogarea bazelor de date. In continuare voi prezenta un exemplu de repository din aplicație.



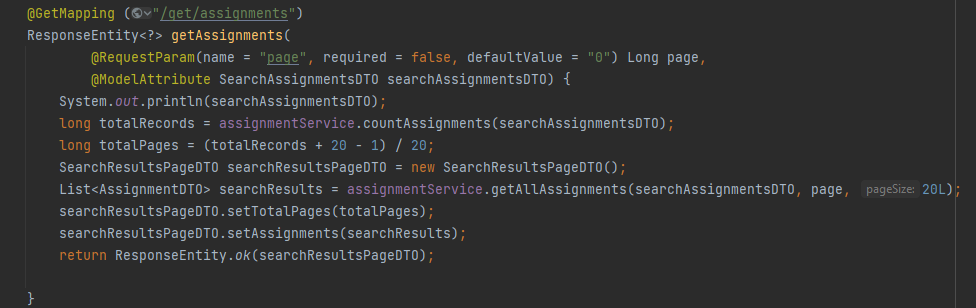
În repository-ul EnrollRepository observăm că, datorită utilizării Spring Data JPA, metoda deleteByCourse\_CourseId(String courseId) este generată automat de bibliotecă pe baza convențiilor de denumire. În schimb, pentru celelalte metode, definesc explicit interogările folosind adnotarea @Query.

## Paginare si cautare dupa diferite informatii

O alta problema pe care am intampinat-o in aplicatie a fost paginarea si cautarea. Am folosit paginare pentru lista de exercitii dar si pentru lista de incarcari a temelor studentilor.

Problema este ca voiam ca aplicatia sa faca cautarea direct prin cerere in baza de date, nu sa fac cerere apoi sa sortez elementele, pentru ca acest lucru ar fi redus din performanta, de asemenea voiam cautarea pe mai multe campuri in acelasi timp, ceea ce necesita o cerere complexa.

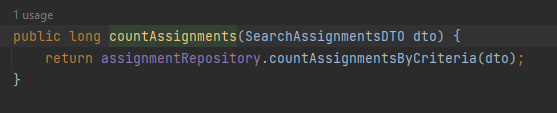
Pentru a rezolva aceasta problema, am implementat urmatoarea solutie:



1. Expunerea unui REST endpoint care primește de la client numărul paginii si un obiect de tipul SearchAssignmentsDTO care specifică criteriile de căutare.



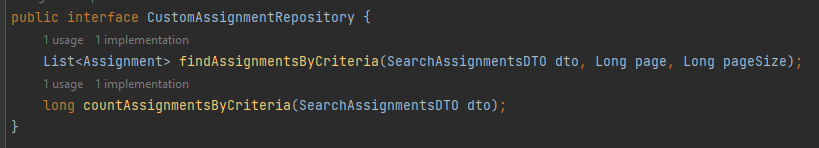
1. După cum observam ne folosim de assignmentService pentru a primi numărul de exerciții care există in baza de date cu aceste criterii.



1. Ne folosim de service pentru a obține numărul de exerciții care exista in baza de date conform criteriilor specificate

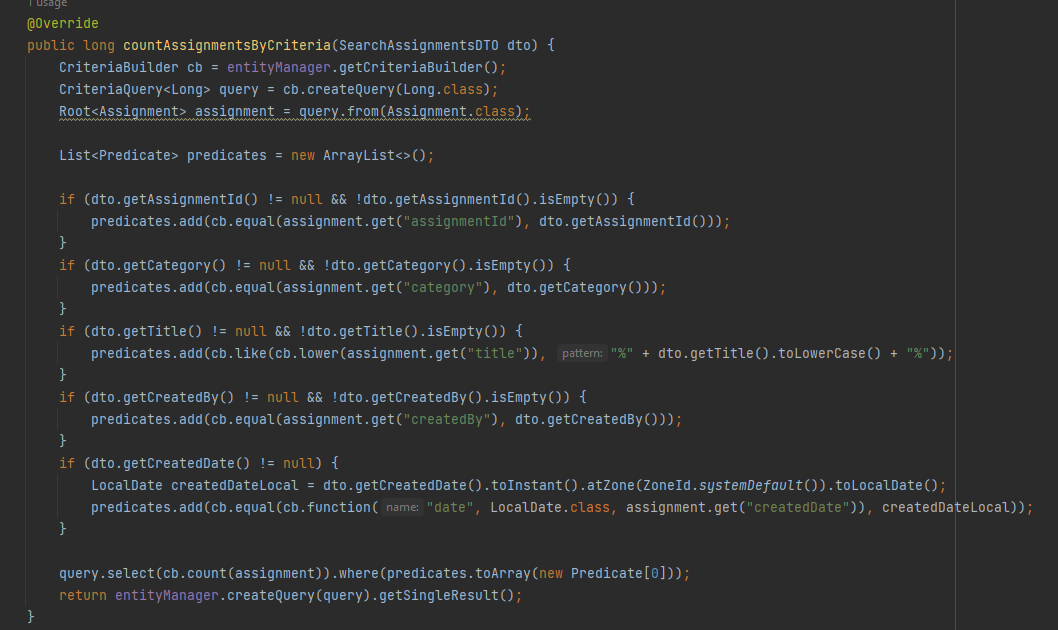


1. Metoda din repository este adusa de extinderea CustomAssignmentRepository.

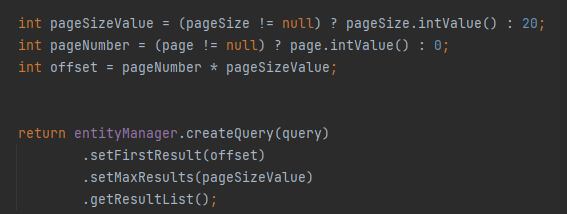


### Detalii de implementare

Astfel daca ne uitam in interfata CustomAssignmentRepositroy putem observa metoda noastra care are 1 implementare. Aceasta implementare duce catre o clasa CusstomAssignmentRepositoryImpl care implementeaza aceasta interfata si implementeaza cele 2 metode.



In aceaste functie countAssignmentsByCriteria(SearchAssignmentsDTO)’ se creeaza un CriteriaBuilder pentru a construi interogari dinamice, CriteriaQuery este configurat pentru a returna Long care reprezinta numarul de intregistrari ce se potrifvesc criteriilor, Root reprezinta tabelul de baza in acest caz Assignment, List<Predicate> este lista cu conditiile interogarii, apoi se face un query select care contorizeaza numarul de exercitii cu predicatele create si returneaza rezultatul.



Functia findAssignmentsByCriteria(SearchAssignmentsDTO, Long page, Long pagesize) este asemanatoare din punct de vedere al predicatelor, avand aceleasi criterii de cautare setate doar ca la acest pas este adaugata paginarea prezenta in figura x. Astfel pentru acest query trebuie calculat indexul de start pentru interogarea bazei de date si numarul de inregistrari care sunt returnate pe aceea pagina sunt date ca parametru in interogare de asemenea.

In final in controller dupa ce acestea sunt setate este construit un obiect de tipul SearchResultsPageDTO care contine numarul total de pagini si lista de exercitii, pentru a fi afisate pe partea de client.

# Studiu de caz / Evaluarea rezultatelor

Acest capitol trebuie să răspundă, în principiu, la **2 întrebări** și să se încheie cu **o discuție** a rezultatelor obținute. Cele doua întrebări la care trebuie sa se răspundă sunt:

1) **Merge corect**? (Conform specificațiilor extrase în capitolul 2);

Evaluarea dacă merge corect se face pe baza cerințelor identificate în capitolele anterioare.

2) Cât de bine merge / cum se compară cu soluțiile existente? (pe baza unor metrici clare).

Evaluarea cât de bine merge trebuie să fie bazată pe procente, timpi, cantitate, numere, **comparativ cu soluțiile prezentate în capitolul 3**. Poate fi vorba de performanță, overhead, resurse consumate, scalabilitate etc.

În realizarea discuției, se vor utiliza tabele cu procente, rezultate numerice și grafice. În mod obișnuit, aici se fac comparații și teste comparative cu alte proiecte similare (dacă există) și se extrag puncte tari și puncte slabe. Se ține cont de avantajele menționate și se demonstrează viabilitatea abordării / aplicației, de dorit prin comparație cu alte abordări (dacă acest lucru este posibil). Cuvântul cheie la evaluare este „metrică“: trebuie să aveți noțiuni măsurabile și cuantificabile. În cadrul procesului de evaluare, explicați datele, tabelele și graficele pe care le prezentați și insistați pe relevanța lor, în următorul stil: „este de preferat ... deoarece …“; explicați cititorului nu doar datele ci și semnificația lor și cum sunt acestea interpretate. Din această interpretare trebuie să rezulte poziționarea proiectului vostru printre alternativele existente, precum și cum poate fi acesta îmbunătățit în continuare.

Criterii pentru calificativul *Nesatisfăcător*:

* Aplicația este testată dar rulează pe calculatorul studentului, nu există posibilități de testare, nu a fost validată cu clienți / utilizatori;
* Nu au fost realizate comparații cu alte sisteme similare.

Criterii pentru calificativul *Satisfăcător*:

* [Dezvoltare de produs] Există teste unitare și de integrare, există o strategie de punere în funcțiune (*deployment*), există validare minimală cu clienții / utilizatorii.
* [Cercetare] Principalele componente și soluția în ansamblu au fost evaluate din punct de vedere al performanței, însă nu sunt folosite seturi de date standard, există unele erori de interpretare a datelor.
* [Ambele] Discuție minimală asupra relevanței rezultatelor prezentate, comparație minimală cu alte sisteme similare.

Criterii pentru calificativul *Bine*:

* [Dezvoltare de produs] Teste unitare și de integrare, instrumente de punere in funcțiune (*deployment*) utilizate și care arată lucru constant de-a lungul semestrului, lucrare validată cu clienții / utilizatorii, produs în producție.
* [Cercetare] Componentele și soluția în ansamblu au fost evaluate din punct de vedere al performanței, folosind seturi de date standard și cu o interpretare corectă a rezultatelor.
* [Ambele] Discuție cu prezentarea calitativă și cantitativă a rezultatelor, precum și a relevanței acestor rezultate printr-o comparație complexă cu alte sisteme similare.

# Concluzii

În acest capitol este sumarizat întreg proiectul, de la obiective, la implementare, si la relevanta rezultatelor obținute. În finalul capitolului poate exista o subsecțiune de „Dezvoltări ulterioare“.  
Criterii pentru calificativul *Nesatisfăcător*:

* Concluziile nu sunt corelate cu conținutul lucrării;

Criterii pentru calificativul *Satisfăcător*:

* Concluziile sunt corelate cu conținutul lucrării, însă nu se oferă o imagine asupra calității și relevantei rezultatelor obținute;

Criterii pentru calificativul *Bine*:

* Concluziile sunt corelate cu conținutul lucrării, și se oferă o imagine precisa asupra relevantei și calității rezultatelor obținute în cadrul proiectului.

# Bibliografie

* Trebuie respectat **un singur standard** de trimiteri bibliografice (citare), **dintre** următoarele alternative:
  + APA (<http://pitt.libguides.com/c.php?g=12108&p=64730>)
  + IEEE (<https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf>)
  + Harvard (<https://libweb.anglia.ac.uk/referencing/harvard.htm>)
  + Cu numerotarea referințelor în ordine alfabetică sau în ordinea apariției în text (de exemplu, stilul cu numere folosit de unele publicații ACM - <https://www.acm.org/publications/authors/reference-formatting>)
* Toate referințele din acest capitol trebuie să fie referite în text. Exemple:
  + [Articol jurnal]: [2];
  + [Articol conferință]: [3];
  + [Carte]: [4]
  + [Weblink]: [5]
  + [Application report] [6]

|  |  |
| --- | --- |
| [1] | J. Silva-Martinez, "ELEN-325. Introduction to Electronic Circuits: A Design Approach," 2008. [Online]. Available: http://www.ece.tamu.edu/~spalermo/ecen325/Section%20III.pdf. |
| [2] | H. Baali, H. Djelouat, A. Amira and F. Bensaali, "Empowering Technology Enabled Care Using IoT and Smart Devices: A Review," *IEEE Sensors Journal,* vol. 18, no. 5, pp. 1790-1809, 2018. |
| [3] | A. Haroon, S. Akram, M. A. Shah and A. Wahid, "E-Lithe: A Lightweight Secure DTLS for IoT," in *IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Toronto, 2017. |
| [4] | A. K. Jain and R. C. Dubes., Algorithms for Clustering Data, Upper Saddle River: Prentice-Hall, Inc., 1988. |
| [5] | "Kernel panic! What are Meltdown and Spectre, the bugs affecting nearly every computer and device?," techcrunch.com, 2018. [Online]. Available: https://techcrunch.com/2018/01/03/kernel-panic-what-are-meltdown-and-spectre-the-bugs-affecting-nearly-every-computer-and-device. [Accessed 14 02 2018]. |
| [6] | E. Rogers, "Understanding Buck-Boost Power Stages in Switch Mode Power Supplies," Texas Instruments, 2007. |

* NU utilizați referințe la Wikipedia sau alte surse fără autor asumat.
* Pentru referințe la articole relevante accesibile în web (descrise prin URL) se va nota la bibliografie și data accesării.
* Mai multe detalii despre citarea referințelor din internet se pot regăsi la:
  + <http://www.writinghelp-central.com/apa-citation-internet.html>
  + <http://www.webliminal.com/search/search-web13.html>
* Note de subsol se utilizează dacă referiți un link mai puțin semnificativ o singură dată; Dacă nota este citată de mai multe ori, atunci utilizați o referință bibliografică.
* Dacă o imagine este introdusă în text și nu este realizată de către autorul lucrării, trebuie citată sursa ei (ca notă de subsol sau referință - este de preferat utilizarea unei note de subsol).
* Referințele se pun direct legate de text (de exemplu „KVM [1] uses“, „as stated by Popescu and Ionescu [12]”, etc.). Nu este recomandat să folosiți formulări de tipul „[1] uses”, „as stated in [12]“, „as described in [11]“ etc.
* Afirmațiile de forma „are numerous“, „have grown exponentially“, „are among the most used“, „are an important topic“ trebuie să fie acoperite cu citări, date concrete si analize comparative.
  + Mai ales în capitolele de introducere, „state of the art“, „related work“ sau „background“ trebuie să vă argumentați afirmațiile prin citări. Fiți autocritici și gândiți-vă dacă afirmațiile au nevoie de citări, chiar și cele pe care le considerați evidente.
  + Cea mai mare parte dintre citări vor fi în capitolele de introducere „state of the art“, „related work“ sau „background“.
* Toate intrările bibliografice trebuie citate în text. Nu le adăugați pur și simplu la final.
* Nu copiați sau traduceți niciodată din surse de informație de orice tip (online, offline, cărți, etc.). Dacă totuși doriți să oferiți, prin excepție, un citat celebru - de maxim 1 frază- utilizați ghilimele și evident menționați sursa.
* Dacă reformulați idei sau creați un paragraf rezumat al unor idei folosind cuvintele voastre, precizați cu citare (referință bibliografică) sau cu notă de subsol sursa sau sursele de unde ați preluat ideile.

# Anexe

Anexele sunt opționale.

Ce poate intra în anexe:

* Exemplu de fișier de configurare sau compilare;
* Un tabel mai mare de ½ pagină;
* O figura mai mare mai mare de ½ pagină;
* O secvență de cod sursa mai mare de ½ pagină;
* Un set de capturi de ecran („screenshot”-uri);
* Un exemplu de rulare a unor comenzi plus rezultatul („output”-ul) acestora;
* În anexe intră lucruri care ocupă mai mult de o pagină ce ar întrerupe firul natural de parcurgere al textului.

1. https://www.edupedu.ro/piata-meditatiilor-estimata-la-peste-120-de-milioane-de-lei-pe-luna-pe-baza-datelor-publice-si-a-cercetarilor-recente-analiza/ [↑](#footnote-ref-1)
2. https://www.rainews.it/articoli/2024/04/scuola-1-studente-su-5-va-a-ripetizioni-cosi-le-famiglie-spendono-circa-450-euro-7a461864-a445-4a71-849c-ff05fbf738b6.html#:~:text=Il%20trend%20%C3%A8%20confortante%2C%20sia,si%20%C3%A8%20passati%20al%2024%25 [↑](#footnote-ref-2)
3. https://www.sciencedirect.com/science/article/pii/S0959475219303093 [↑](#footnote-ref-3)