

# **Documentație Tehnică**

## Platformă E-learning

Autori:

Ivan Cosmin-Gabriel

Maloș Mihai

## **Cuprins**

<b>1</b>	<b>Introducere</b>	<b>2</b>
1.1	Descrierea temei . . . . .	2
1.2	Justificarea alegerii tehnice . . . . .	2
<b>2</b>	<b>Analiza cerințelor</b>	<b>2</b>
2.1	Cerințe funcționale . . . . .	2
2.2	Cerințe non-funcționale . . . . .	3
<b>3</b>	<b>Implementarea server-ului</b>	<b>3</b>
3.1	Nivelul de interfață API . . . . .	3
3.2	Nivelul de servicii . . . . .	3
3.3	Nivelul de acces la date . . . . .	4

# 1 Introducere

## 1.1 Descrierea temei

Proiectul constă în realizarea unui sistem software distribuit dedicat procesului de învățare. Arhitectura sistemului este compusă dintr-o aplicație server care gestionează logica de business și datele, și o aplicație mobilă Android care servește drept interfață pentru utilizatori. Sistemul permite interacțiunea dintre profesori și studenți prin funcționalități precum crearea de cursuri, înscrierea studenților și distribuirea materialelor didactice. O funcționalitate centrală este reprezentată de un agent conversațional capabil să răspundă la întrebări pe baza conținutului cursurilor.

## 1.2 Justificarea alegării tehnice

Soluția tehnică a fost aleasă pentru a răspunde necesității de a avea un sistem modular și ușor de întreținut. S-a optat pentru o arhitectură bazată pe servicii web REST deoarece acest standard asigură o comunicare eficientă între server și clientul Android, independent de platformă. Utilizarea unui limbaj de programare modern pentru server permite procesarea rapidă a cererilor și integrarea facilă a bibliotecilor de inteligență artificială. Pentru stocarea datelor s-a ales o bază de date relațională, întrucât structura informațiilor academice este bine definită și necesită consistență strictă a relațiilor dintre utilizatori și cursuri.

# 2 Analiza cerințelor

## 2.1 Cerințe funcționale

Sistemul implementează un set de funcționalități menite să asigure desfășurarea activității didactice pe o perioadă determinată, precum un modul de zece săptămâni.

Principalele funcționalități includ:

- **Gestiunea utilizatorilor:** Sistemul permite crearea conturilor și autentificarea securizată. Există roluri distincte pentru profesori și studenți, fiecare având drepturi specifice de acces.
- **Administrarea cursurilor:** Profesorii au posibilitatea de a iniția cursuri noi, de a adăuga descrieri și de a modifica conținutul acestora.
- **Înrolarea la cursuri:** Studenții pot vizualiza lista cursurilor disponibile și pot solicita înscrierea la acestea.

- **Managementul resurselor:** Platforma facilitează încărcarea documentelor și a fișierelor suport pentru fiecare curs în parte.
- **Asistență intelligentă:** Utilizatorii pot interoga sistemul printr-o interfață de chat pentru a obține informații specifice extrase automat din materialele de curs încărcate.

## 2.2 Cerințe non-funcționale

Arhitectura respectă standarde de calitate care asigură buna funcționare a aplicației:

- **Securitatea datelor:** Accesul la resurse este permis doar pe baza unor elemente de identificare verificate. Parolele sunt stocate într-o formă criptată ireversibilă.
- **Scalabilitatea:** Structura modulară permite adăugarea de noi funcționalități fără a afecta componente existente.
- **Disponibilitatea:** Serverul este proiectat să răspundă cererilor venite de la mai mulți utilizatori simultan fără blocaje.

# 3 Implementarea server-ului

Sistemul este organizat pe mai multe niveluri logice pentru a separa responsabilitățile și a simplifica procesul de dezvoltare.

## 3.1 Nivelul de interfață API

Acest nivel reprezintă punctul de intrare în aplicație. Aici sunt definite rutele care pot fi apelate de clientul mobil. Fiecare rută corespunde unei acțiuni specifice și primește datele într-un format standardizat JSON. Acest nivel validează datele primite și le trimit mai departe către nivelul de servicii.

## 3.2 Nivelul de servicii

Logica propriu-zisă a aplicației se află în acest nivel. Aici au loc procesările complexe, cum ar fi verificarea permisiunilor unui utilizator de a accesa un anumit fișier sau coordonarea algoritmilor de căutare intelligentă. Serviciile orchestrează fluxul de date între baza de date și interfața API.

### **3.3 Nivelul de acces la date**

Pentru a interacționa cu baza de date, sistemul folosește un strat intermediar de repository-uri. Aceste componente conțin codul necesar pentru executarea interogărilor de tip creare, citire, actualizare și ștergere. Această abordare izolează interogările SQL de restul codului și facilitează modificările ulterioare asupra structurii bazei de date.