



# **UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA**

---

**Sistem intelligent pentru management-ul task-urilor**

*Inginerie Software*

---

**Autori:**

Sîrbu Vlad  
Arduszădan Darius

**Grupa: 6**

**Îndrumător de laborator:**

Burzo Adrian Gabriel

FACULTATEA DE AUTOMATICĂ  
ȘI CALCULATOARE

## Cuprins

<b>1 Descriere generală a proiectului . . . . .</b>	<b>2</b>
<b>2 Limbaje și tehnologii utilizate . . . . .</b>	<b>2</b>
<b>3 Cerințe funcționale . . . . .</b>	<b>2</b>
3.1 Autentificare și autorizare . . . . .	2
3.2 Gestionarea task-urilor . . . . .	2
3.3 Gestionarea proiectelor . . . . .	2
3.4 Use-case diagram . . . . .	3
<b>4 Cerințe non-funcționale . . . . .</b>	<b>3</b>
<b>5 Diagrame UML . . . . .</b>	<b>4</b>
5.1 Diagrame realizate de Vlad . . . . .	4
5.2 Diagrame realizate de Darius . . . . .	5
<b>6 Design Patterns . . . . .</b>	<b>6</b>
6.1 Facade Pattern – Darius . . . . .	6
6.2 Adapter Pattern – Vlad . . . . .	7
<b>7 Concluzie . . . . .</b>	<b>7</b>

# 1 Descriere generală a proiectului

Captain's Log este o aplicație web modernă pentru gestionarea task-urilor și proiectelor, având o tematică pirat bine definită. Platforma permite crearea, organizarea și asignarea misiunilor către utilizatori, urmărirea progresului acestora și gestionarea proiectelor în cadrul unor echipe.

Aplicația este destinată atât utilizatorilor obișnuiți (Crew Members), cât și administratorilor (Captains), fiecare având drepturi și funcționalități diferite. Sistemul oferă notificări inteligente, prioritizare automată a task-urilor folosind AI și un design intuitiv, ușor de utilizat.

## 2 Limbaje și tehnologii utilizate

Pentru dezvoltarea aplicației au fost utilizate următoarele tehnologii:

- **Next.js 14** – framework React pentru aplicații web moderne
- **TypeScript** – limbaj pentru tipare statice și siguranță sporită
- **Prisma ORM** – gestionarea bazei de date
- **PostgreSQL** – bază de date relațională
- **NextAuth.js** – autentificare și gestionare sesiuni
- **Hugging Face API** – detecție automată a priorității task-urilor
- **HTML, CSS, TailwindCSS** – interfață grafică

## 3 Cerințe funcționale

### 3.1 Autentificare și autorizare

- Utilizatorii se pot înregistra și autentifica în aplicație
- Sistemul diferențiază între rolurile USER și ADMIN
- Accesul la pagini și API-uri este restricționat în funcție de rol

### 3.2 Gestionarea task-urilor

- Crearea, editarea și ștergerea task-urilor
- Asignarea task-urilor către utilizatori
- Setarea deadline-urilor și priorităților
- Marcare task-uri ca finalize

### 3.3 Gestionarea proiectelor

- Crearea și editarea proiectelor
- Adăugarea și eliminarea membrilor
- Vizualizarea task-urilor dintr-un proiect

### 3.4 Use-case diagram

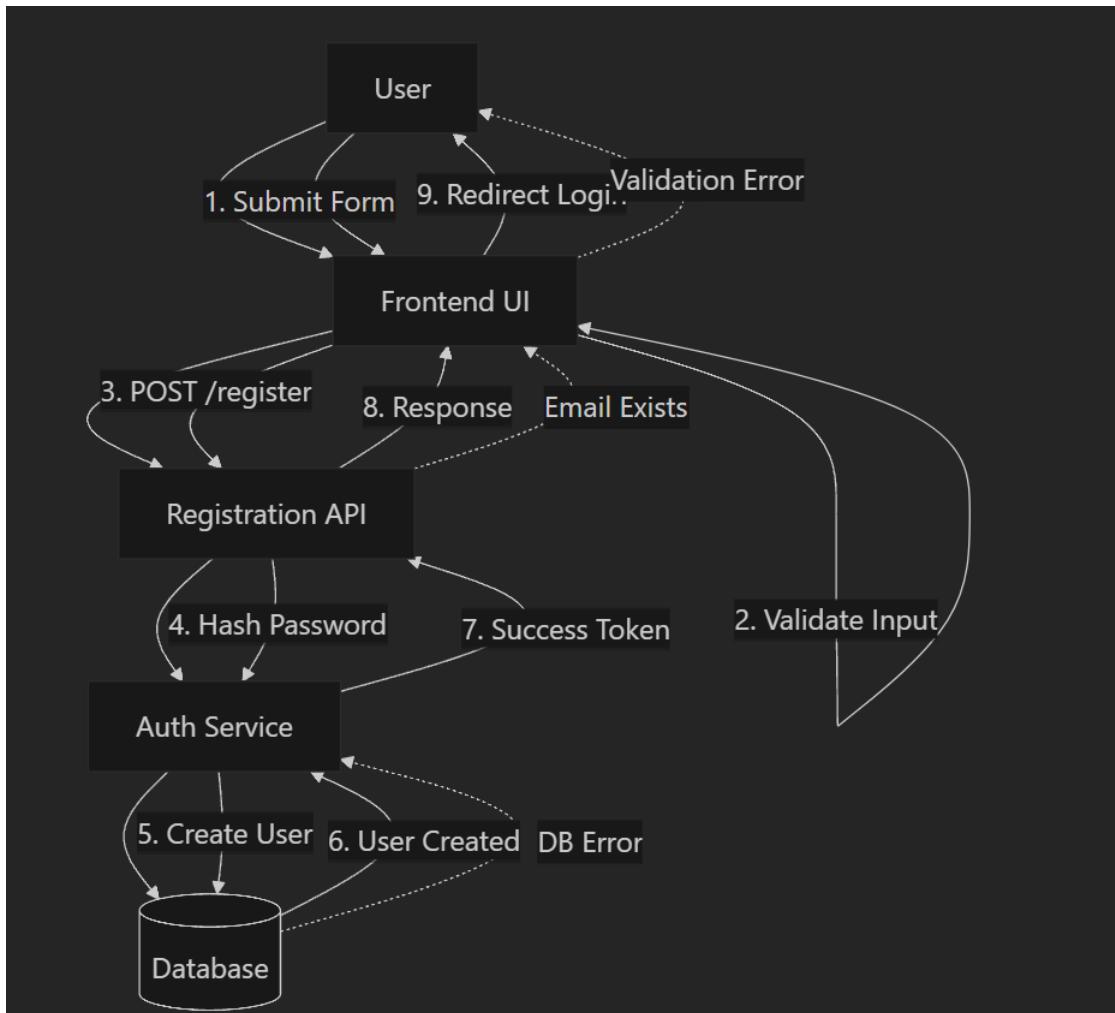


Figura 1: Diagramă Use-Case – User

## 4 Cerințe non-funcționale

- Timpul de răspuns pentru acțiuni uzuale sub 1 secundă
- Securitate ridicată pentru datele utilizatorilor
- Scalabilitate pentru un număr mare de utilizatori

## 5 Diagrame UML

### 5.1 Diagrame realizate de Vlad

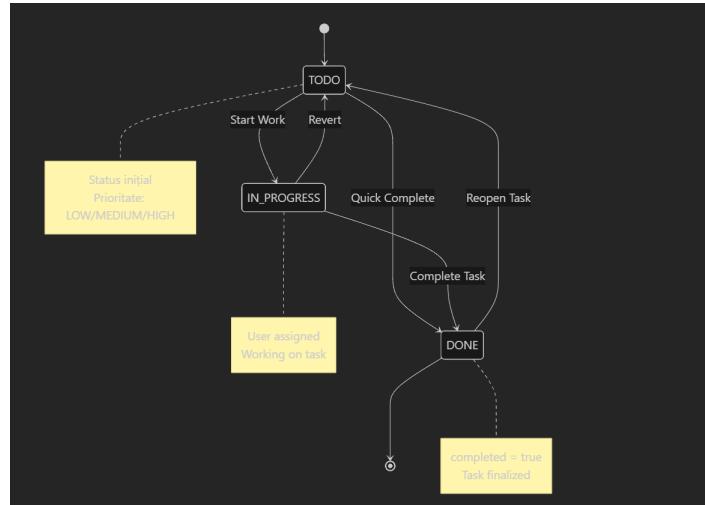


Figura 2: Diagramă de stare Task – Vlad

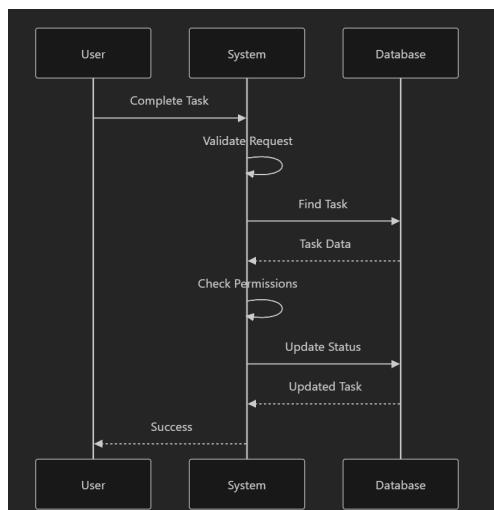


Figura 3: Diagramă de secvență Task Completion – Vlad

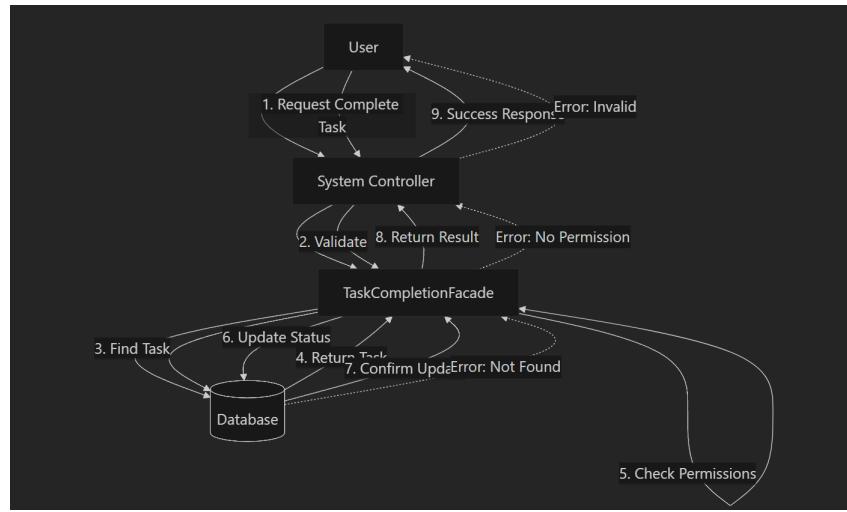


Figura 4: Diagramă de comunicare Task – Vlad

## 5.2 Diagrame realizate de Darius

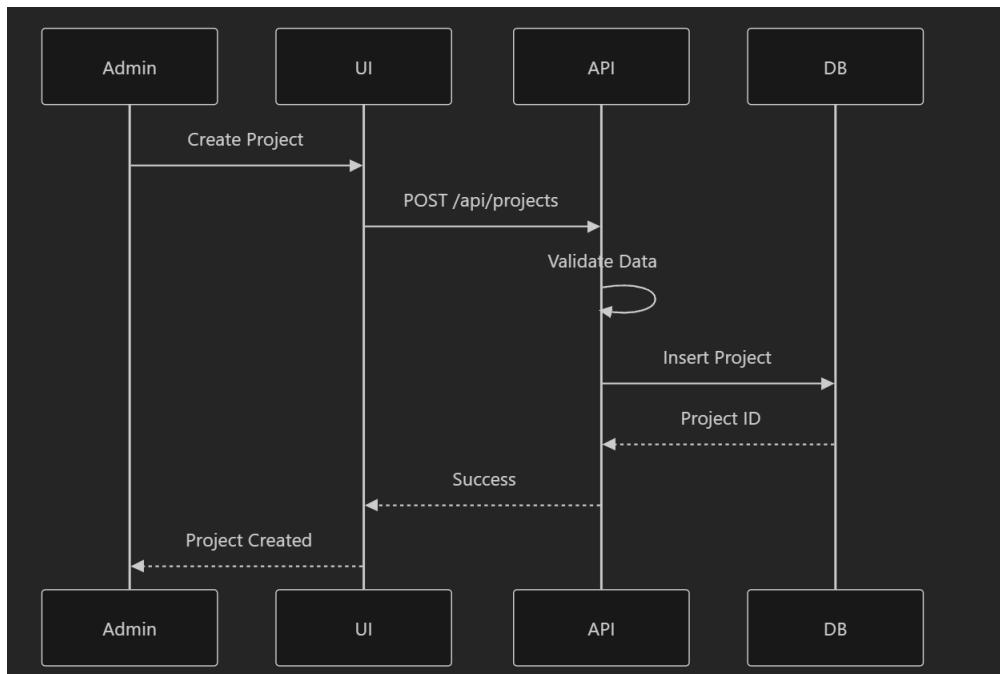


Figura 5: Diagramă de secvență Proiect – Darius

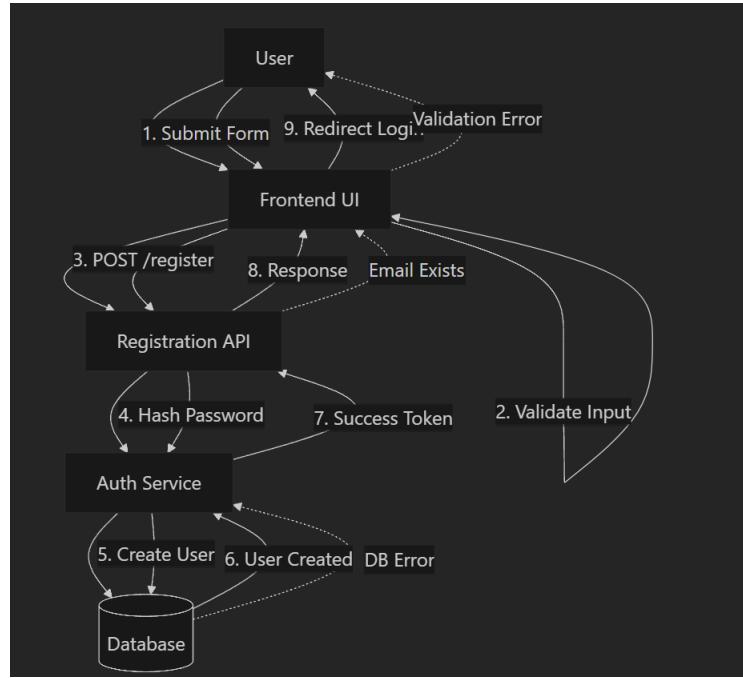


Figura 6: Diagramă de comunicare User – Darius

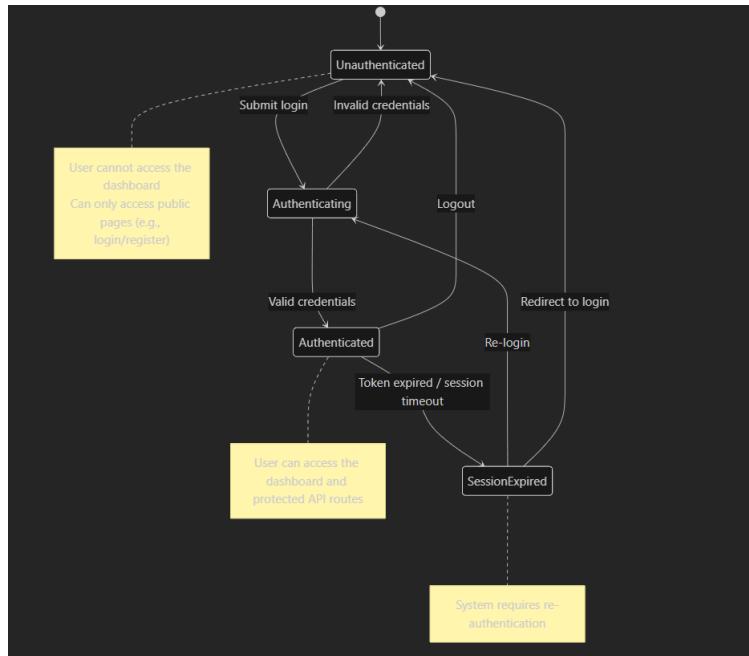


Figura 7: Diagramă de stare User – Darius

## 6 Design Patterns

### 6.1 Facade Pattern – Darius

**Problema identificată:** Logica de completare a task-urilor era complexă și distribuită în mai multe componente.

**Soluție:** A fost utilizat Facade Pattern pentru a oferi o interfață simplă care ascunde complexitatea internă.

**Implementare:** Clasa TaskCompletionFacade centralizează logica necesară completării task-urilor.

## 6.2 Adapter Pattern – Vlad

**Problema identificată:** Integrarea sistemului de autentificare cu Prisma ORM.

**Soluție:** Adapter Pattern permite adaptarea NextAuth la structura bazei de date Prisma.

**Implementare:** NextAuth PrismaAdapter gestionează automat relația dintre utilizatori și sesiuni. resources/

## 7 Concluzie

Captain's Log este o aplicație completă de task management, creată pentru a facilita organizarea muncii în echipă. Platforma oferă funcționalități avansate precum prioritizare automată, notificări inteligente și un sistem clar de roluri.

Proiectul este extensibil, bine structurat și poate fi adaptat ușor pentru utilizare reală în medii profesionale.