

Inginerie Software II

Proiect IS2 - Serviciu WEB

Deadline : 16.01.2026

Responsabili temă: Silviu Pantelimon

Contents

Cerințe generale	2
Cerințe auxiliare	2
Notiuni teoretice	2
REST API	2
Docker	2
Gestiunea bazelor de date	2
Obiecte JSON	3
Descrierea soluție	3
Cerințe arhitecturale	3
Cerințe funcționare	3
Etape și notare	3

Cerințe generale

Obiectivul acestei teme este dezvoltarea unei aplicații WEB care utilizează:

- un serviciu REST implementat utilizând .NET;
- utilizarea unei baze de date pentru persistență.

Cerințe auxiliare

În scopul asigurării unui flux rapid de implementare și gestionare a sarcinilor între membrii echipelor, persoanele desemnate (din fiecare echipă), trebuie să pregătească următoarele:

- Crearea unui repository gol pe Gitlab/GitHub și adăugarea celorlalți membri ai echipei ca și colaboratori;
- Adăugarea coordonatorului și responsabilului (Bogdan Mocanu și Silviu Pantelimon) pe platformele anterior menționate (în cazul Gitlab, rol de developer).

În cadrul procesului de dezvoltare al produsului software, se vor avea în vedere utilizarea branch-urilor și a mecanismului de **Pull Request - Review - Approve**.

Elemente de identificare:

- Bogdan Mocanu: bogdan_costel.mocanu@upb.ro;
- Silviu Pantelimon: silviu.pantelimon@upb.ro.

Noțiuni teoretice

REST API

Prinț- un REST API se intenționează un server web de tip blackbox ce comunica pe baza unor contracte formulate folosind cereri HTTP. Serverul este considerat blackbox deoarece consumatorii acestui API nu stiu nimic despre implementarea serverului, ci sunt pusi în fața unui contract de comunicare definit de cereri și răspunsuri HTTP.

Docker

Docker este o tehnologie de containerizare bazată pe motorul containerd. Utilizând Docker, dezvoltatorii pot realiza aplicații fără să tina cont de sistemul de operare pe care acestea vor rula, fără să tina cont de specificații sau dependențe, codul rulându-se într-un container specializat și izolat.

Containerele pot fi rulate, atât utilizând comenzi individuale, cât și în bază pe o configurație centralizată, folosind Docker Compose. Docker Compose folosește fisiere de configurație .yml pentru a rula mai multe containere concomitent. Aceleasi principii care se aplică în cadrul rularii individuale de containere, se aplică și în cazul rularii Docker Compose, însă trebuie să fiți familiari cu formatul unui fisier .yml.

Gestiunea bazelor de date

Pentru această temă trebuie să aveți noțiuni privind configurația bazelor de date și execuțarea operațiilor de bază (CRUD -Create/Read/Update/Delete).

Obiecte JSON

JSON este un acronim în limba engleză pentru JavaScript Object Notation, și este un format de reprezentare și interschimb de date între aplicații informatici. Este un format text, inteligibil pentru oameni, utilizat pentru reprezentarea obiectelor și a altor structuri de date și este folosit în special pentru a transmite date structurate prin rețea, procesul purtând numele de serializare. Informații suplimentare privind utilizarea JSON se regăsesc în [The JavaScript Object Notation \(JSON\) Data Interchange Format](#).

Descrierea soluție

Cerințe arhitecturale

Arhitectura generală a soluției este prezentată în Figura 1.



Figure 1: Prezentare generală a soluției propuse.

Va trebui să implementați o configurație formată din cel puțin 2 microservicii astfel:

- **REST API** pentru operații CRUD în C#;
- **Baza de date locală** (de ex. SQLite).

Cerințe funcționare

Interfața utilizatorului trebuie să permită cel puțin următoarele funcționalități:

- Operațiile CRUD pentru baza de date.

Etape și notare

ETAPA 1: Stabilire echipe și descriere proiect (07.11 – 14.11.2025) (0.0p):

- Stabilirea echipe, repository public, descriere proiect (maxim 2-3 studenți) - [aici](#);

Neparcurgerea acestei etape va conduce automat la diminuarea notei proiectului cu 1p.

ETAPA II: Proiectare (14.11 – 28.11.2025)(2.0p):

- 2p - document de 1-2 pagini cu specificațiile aplicatiei (user stories) cu eventuale diagrame de activitate și schema de baza de date (SRS și SDD).

ETAPA III: Dezvoltare și deplyment (28.11 – 16.01.2026)(8.0p):

- 1p - controlere (trebuie sa acopere toate actiunile CRUD)
- 2p - servicii care implementeaza logica aplicatiei
- 2p - tabele/entitati in baza de date cu relatii intre ele
- 2p - folosirea de data transfer objects
- 1p - deployment in Docker cu docker compose

Prezentarea temelor se va face în perioada **16-22.01.2026**.

Documentul final care include SRS (Software Requirements Specification) și SDD (Software Design Document) va fi încărcat pe Moodle.