

Sprint 1 – Agile Board Manager (TAM3)

1. Tema aleasă

Aplicație Agile Board Manager pentru echipele care lucrează Agile. Utilizatorii pot crea proiecte, epics, stories și task-uri, pot urmări progresul acestora și pot actualiza statusurile într-un board Kanban (To Do, In Progress, Done).

2. Echipa

- **Product Owner:** Mamaishi Lucian
- **Tech Lead:** Alexandru Constantin Orzu
- **Developer:** Lazăr Alexandra

3. Backlog inițial

Epics:

- Management Task Board
- Management Stories
- Management Users
- Raportare și statusuri

Exemple User Stories (scrise în Gherkin):

- Feature: Create Task
Scenario: Adăugare task nou
Given că sunt utilizator logat
When adaug un task cu titlul 'Implementare Login'
Then task-ul apare în lista 'To Do'
- Feature: Update Task Status
Scenario: Schimbare status task
Given că există un task 'Implementare Login' cu status 'To Do'
When schimb statusul la 'In Progress'
Then task-ul apare sub coloana 'In Progress'
- Feature: Assign Task
Scenario: Atribuire task unui utilizator
Given că există un task 'Design API'
When îl atribui utilizatorului 'Maria Pop'
Then task-ul afișează 'Assignee: Maria Pop'

4. ADR 001 – Alegerea stack-ului tehnologic

Context: Trebuie ales un stack pentru dezvoltarea aplicației.

Opțiuni analizate:

- Java + Spring Boot + PostgreSQL + React
- .NET Core + SQL Server + Angular
- Node.js + MongoDB + Vue.js

Decizie: Se alege Java + Spring Boot + PostgreSQL + React deoarece:

- Este cunoscut de majoritatea echipei.
- Integrare ușoară cu testare automată și CI/CD.
- PostgreSQL este open-source și stabil.
- React permite interfețe moderne și flexibile.

Consecințe: Curba de învățare scăzută, suport bun, resurse educaționale disponibile.

ADR 002 – Stil arhitectural

Context: Trebuie ales un stil arhitectural potrivit pentru o aplicație Java de tip Agile.

Opțiuni analizate:

- Microservicii
- Monolit clasic
- Modular Monolith (DDD stratificat)

Decizie: Se alege Modular Monolith cu DDD stratificat deoarece:

- Simplifică dezvoltarea și livrarea.
- Menține separarea logicii pe straturi (controller, service, repository).
- Poate evolua ulterior către microservicii.

Consecințe: Implementare rapidă și clară, potrivită pentru proiecte de echipă mică.

5. C4 Context Diagram

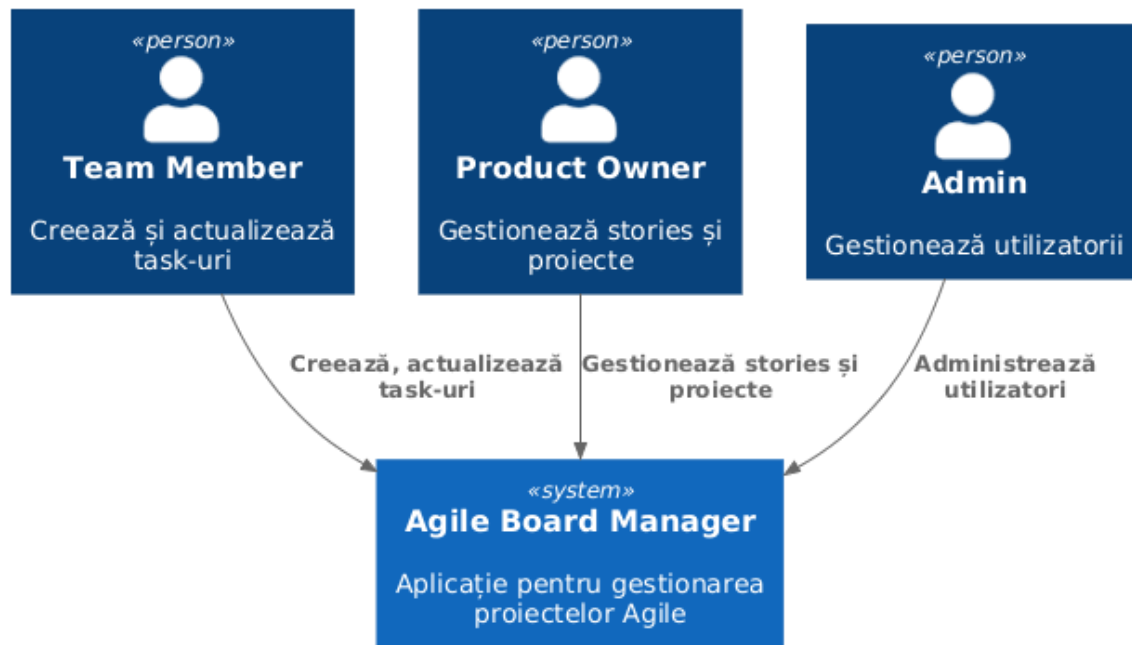
Actorii principali:

- **Team Member:** creează și actualizează task-uri.
- **Product Owner:** gestionează stories și proiecte.
- **Admin:** gestionează utilizatori (opțional).

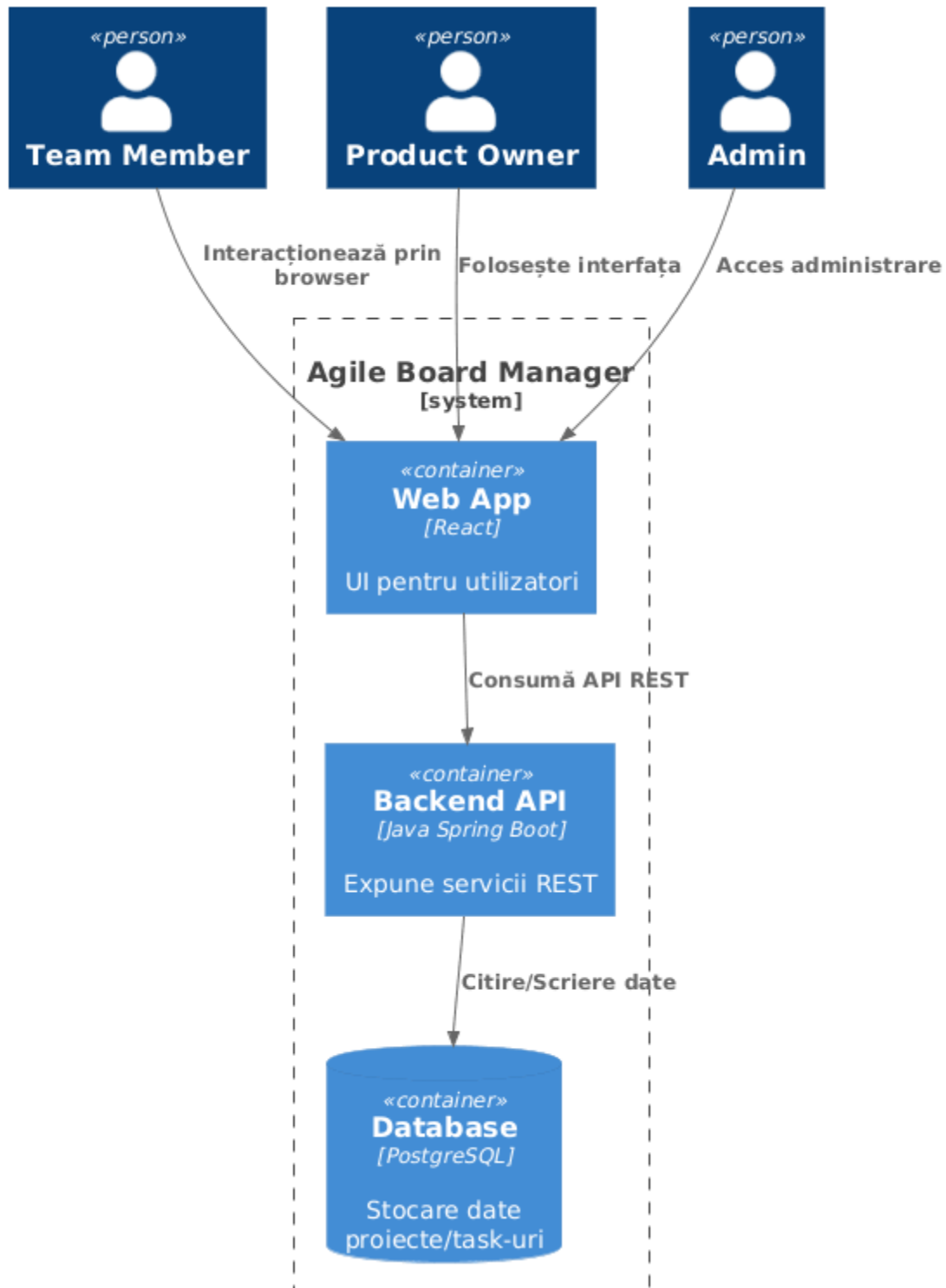
Sistem: Aplicație Agile Board Manager.

- Conține module pentru utilizatori, stories, task-uri și rapoarte.
- Se conectează la PostgreSQL pentru gestionarea datelor.
- Expune API REST consumat de interfața web React.

Context Diagram



Container Diagram



Components Diagram

