Smart City  
Aplicație pentru gestiunea  
orașelor inteligente

Croitoru Traian

Lupșan Tudor Adrian  
C-112-B

# Pagina principala - Academia Tehnica Militara „Ferdinand I”

**Cuprins**

[Capitolul 1 - Introducere 3](#_Toc193783454)

[1.1 Scopul proiectului 3](#_Toc193783455)

[1.2 Lista definițiilor 3](#_Toc193783456)

[1.3 Structura DCS 4](#_Toc193783457)

[Capitolul 2 - Descrierea generală a produsului 4](#_Toc193783458)

[2.1 Descrierea produsului software 4](#_Toc193783459)

[2.2 Detalierea platformei HW/SW 4](#_Toc193783460)

[Capitolul 3 - Detalierea cerințelor specifice 5](#_Toc193783461)

[3.1 Cerințele funcționale 5](#_Toc193783462)

[3.2 Cerințele nefuncționale 5](#_Toc193783463)

# **Capitolul 1 - Introducere**

## **1.1 Scopul proiectului**

Smart-City este o aplicație care facilitează administrarea unui oraș printr-o platformă inteligentă bazată pe dispozitive echipate cu senzori. Sistemul colectează, analizează și stochează date pentru a genera statistici relevante, permițând astfel luarea unor decizii informate pentru îmbunătățirea calității vieții cetățenilor. Totodată, automatizările integrate vor reduce timpul de reacție al autorităților în situații de urgență, asigurând o intervenție rapidă și eficientă.

## **1.2 Lista definițiilor**

* **Client / Utilizator** – Se referă la primărie, autoritate publică sau instituție care utilizează aplicația Smart-City pentru administrarea orașului.
* **Admin** – Utilizator cu drepturi depline asupra platformei, care poate gestiona harta orașului, adăuga instituții, aloca permisiuni și monitoriza activitatea generală.
* **Instituție** – O entitate (ex: secție de poliție, unitate de pompieri, spital) care utilizează platforma pentru a gestiona senzori și a primi notificări în timp real.
* **Senzor** – Dispozitiv inteligent care colectează și transmite date despre diverse aspecte ale orașului (ex: poluare, trafic, temperatură, securitate).
* **Harta digitală a orașului** – Reprezentarea grafică a orașului, împărțită în zone administrative, utilizată pentru poziționarea instituțiilor și senzorilor.
* **Autentificare cu doi factori (2FA)** – Metodă de securitate care necesită două etape pentru accesarea contului (ex: parolă + cod unic generat de o aplicație).
* **Notificare** – Alertă transmisă utilizatorilor autorizați în cazul unei activități detectate de senzori.
* **Permisiuni** – Set de drepturi acordate unui utilizator pentru a accesa și gestiona anumite funcționalități din aplicație.
* **Timp real** – Procesarea și afișarea imediată a datelor colectate de senzori fără întârzieri semnificative.
* **Raport de analiză** – Document sau vizualizare grafică generată automat pe baza datelor colectate de senzori pentru a facilita luarea deciziilor.

## **1.3 Structura DCS**

Documentul este împărțit în două capitole:

* Capitolul 2 prezintă o descriere generală a aplicației Smart-City și a hardware-ului necesar.
* Capitolul 3 prezintă cerințele funcționale și nefuncționale ale produsului software.

# **Capitolul 2 - Descrierea generală a produsului**

## **2.1 Descrierea produsului software**

Smart-City oferă primăriilor posibilitatea de a digitaliza orașele pe care le gestionează. Astfel, fiecare client, în momentul creării contului, va încărca harta orașului pe care îl va administra. Aceasta va fi împărțită în mai multe zone în funcție de dimensiune, precum o matrice.

După autentificare, utilizatorii vor putea adăuga, în fiecare regiune, autorități/instituții (secție de poliție, unitate de pompieri etc.) prin drag and drop. În mod automat, fiecărei instituții i se va crea un cont cu restricții limitate. Așadar, în momentul în care o persoană se va conecta pe un astfel de cont, va avea acces să adauge doar senzorii specifici domeniului de activitate (poliție – radar, senzor efracție; pompieri – senzor fum etc.). Totodată, va putea specifica ce alte conturi vor avea acces să vizualizeze informațiile despre senzorii respectivi.

Senzorii vor oferi date utilizatorilor în timp real. Mai mult decât atât, în cazul în care se vor declanșa, o notificare va fi transmisă către autoritatea care a adăugat senzorul, dar și către conturile care au primit permisiune.

## **2.2 Detalierea platformei HW/SW**

Software-ul este dezvoltat pentru Windows 11, dar este compatibil și cu alte versiuni ale sistemului de operare. Interfața grafică este realizată folosind Qt, un framework multi-platformă, iar dezvoltarea are loc în mediul Microsoft Visual Studio.

Cerințele minime includ:

* Procesor de minim 2 GHz
* Cel puțin 4 GB de RAM

# **Capitolul 3 - Detalierea cerințelor specifice**

## **3.1 Cerințele funcționale**

1. **Autentificare și autorizare** – Sistemul trebuie să permită autentificarea utilizatorilor și atribuirea rolurilor (admin, instituție, utilizator standard).
2. **Gestionarea hărții orașului** – Primăriile trebuie să poată încărca și edita harta orașului, împărțind-o în zone.
3. **Adăugarea instituțiilor** – Utilizatorii trebuie să poată adăuga instituții prin drag and drop în diferite zone ale orașului.
4. **Crearea și gestionarea conturilor instituțiilor** – Sistemul trebuie să genereze automat conturi pentru fiecare instituție adăugată.
5. **Adăugarea și configurarea senzorilor** – Instituțiile trebuie să poată adăuga senzori specifici domeniului lor de activitate.
6. **Monitorizarea în timp real** – Datele colectate de senzori trebuie să fie afișate în timp real pe platformă.
7. **Notificări și alerte** – În cazul declanșării unui senzor, notificările trebuie trimise către instituția responsabilă și utilizatorii autorizați.
8. **Raportare și analiză** – Sistemul trebuie să genereze statistici și rapoarte bazate pe datele colectate.
9. **Gestionarea permisiunilor** – Utilizatorii trebuie să poată defini cine are acces la datele senzorilor.

## **3.2 Cerințele nefuncționale**

1. **Performanță** – Aplicația trebuie să răspundă la interogări în mai puțin de 2 secunde pentru 90% dintre cereri.
2. **Scalabilitate** – Sistemul trebuie să poată gestiona orașe de dimensiuni diferite, de la localități mici la metropole.
3. **Securitate** – Datele trebuie să fie criptate, iar autentificarea să folosească autentificare cu doi factori (2FA).
4. **Compatibilitate** – Aplicația trebuie să ruleze pe Windows 11 și versiunile mai vechi compatibile.
5. **Interfață prietenoasă** – UI-ul trebuie să fie intuitiv și ușor de utilizat chiar și pentru persoane fără cunoștințe tehnice avansate.
6. **Fiabilitate** – Sistemul trebuie să aibă un uptime de cel puțin 99,5%.
7. **Logare și audit** – Toate acțiunile utilizatorilor trebuie să fie înregistrate pentru audit și securitate.
8. **Modularitate** – Aplicația trebuie să fie dezvoltată astfel încât noi funcționalități să poată fi adăugate fără a afecta structura existentă.
9. **Toleranță la erori** – În caz de eroare, sistemul trebuie să poată relua funcționarea fără pierderi majore de date.