**Documentul de specificare a cerințelor**

**Software Requirements Specification**

**(SRS) Document**

**<Sistem automatizat pentru controlul unui vehicul destinat monitorizarii parametrilor de mediu in regiuni greu accesibile>**

**<15.04.2025><Versiune>**

**<Companie>**

|  |
| --- |
| **Istoricul versiunilor** |

| Versiune | Autor(i) principali | Descriere versiune | Dată |
| --- | --- | --- | --- |
|  |  |  |  |

|  |
| --- |
| **Revizuiri și aprobări** |

Istoric aprobări

| Aprobă | Versiune | Semnătură | Dată |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Istoric revizuiri

| Revizor | Versiune | Semnătură | Dată |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **Cuprins** |

[1. Introducere 4](#_Toc190689617)

[1.1 Scopul 4](#_Toc190689618)

[1.2 Convenții ale documentului 4](#_Toc190689619)

[1.3 Audiență țintă 4](#_Toc190689620)

[1.4 Sfera de aplicare 4](#_Toc190689621)

[1.5 Referințe 4](#_Toc190689622)

[**2 Descriere generală 4**](#_Toc190689623)

[**2.1 Perspectiva produsului 4**](#_Toc190689624)

[**2.2 Caracteristici ale produsului 4**](#_Toc190689625)

[**2.3 Clase și caracteristici ale utilizatorilor 4**](#_Toc190689626)

[**2.4 Mediul de operare 5**](#_Toc190689627)

[**2.5 Constrângeri de proiectare și de implementare 5**](#_Toc190689628)

[**2.6 Presupuneri și dependențe 5**](#_Toc190689629)

[**3 Cerințele sistemului 5**](#_Toc190689630)

[**3.1 Funcționalitatea 1 5**](#_Toc190689631)

[**3.1.1 Descriere generală 5**](#_Toc190689632)

[**3.1.2 Flux de interacțiune (scenarii de utilizare) 5**](#_Toc190689633)

[**3.1.3 Condiții prealabile și constrângeri 6**](#_Toc190689634)

[**3.1.4 Detaliere cerință 6**](#_Toc190689635)

[**3.1.5 Scenarii de eroare și gestionarea excepțiilor 6**](#_Toc190689636)

[**3.1.5 Dependențe și interacțiuni cu alte funcționalități 6**](#_Toc190689637)

[**3.2 Cerința funcțională 2 6**](#_Toc190689638)

[3.3 .... 6](#_Toc190689639)

[4 Cerințe pentru interfețe 6](#_Toc190689640)

[4.1 Interfețe cu utilizatorul 6](#_Toc190689641)

[4.2 Interfețe hardware 6](#_Toc190689642)

[4.2.1 Configurații Minime Recomandate 7](#_Toc190689643)

[4.2.2 Dispozitive Externe Compatibile 7](#_Toc190689644)

[4.3 Interfețe de comunicare 7](#_Toc190689645)

[4.3.1 Protocoale și Standarde de Comunicare 7](#_Toc190689646)

[4.3.2 Cerințe de Securitate în Comunicare 7](#_Toc190689647)

[4.4 Interfețe software 7](#_Toc190689648)

[4.4.1 Tehnologii Utilizate 7](#_Toc190689649)

[4.4.2 Servicii Externe și API-uri 7](#_Toc190689650)

[5 Cerințe non-funcționale 8](#_Toc190689651)

[5.1 Cerințe de performanță 8](#_Toc190689652)

[5.2 Cerințe de siguranță 8](#_Toc190689653)

[5.3 Cerințe de securitate 8](#_Toc190689654)

[5.4 Atribute de calitate ale software-ului 8](#_Toc190689655)

[6 Alte cerințe 8](#_Toc190689656)

[7 Anexe 8](#_Toc190689657)

[7.1 Anexa A: Glosar 8](#_Toc190689658)

[7.2 Anexa B: Modele de Analiză 8](#_Toc190689659)

[7.3 Anexa C: Listă de Probleme 9](#_Toc190689660)

# Introducere

## Scopul

Această lucrare are ca scop descrierea detaliată a unui sistem automatizat care controlează un vehicul destinat colectării de informații despre mediul înconjurător din zone greu accesibile. Controlul se face cu ajutorul unei aplicații mobile, iar vehiculul este echipat cu o serie de senzori ce permit monitorizarea parametrilor de mediu în timp real.

## Convenții ale documentului

Metodologiile tipografice urmate în cadrul documentului. De exemplu, orice abrevieri, stilizare tipografică a conținutului sau schimbări de fonturi și semnificația acestora.

## Audiență țintă

Acest document este destinat profesorului coordonator pentru a putea identifica posibile erori, avand ca scop remedierea acestora intr-un timp favorabil. De asemenea este conceput pentru comisia de evaluare, dar si pentru studentii interesati de dezvoltarea acestui proiect.

## Sfera de aplicare

Proiectul se aliniază unui sistem mai larg de monitorizare ambientală, oferind o soluție mobilă, controlabilă de la distanță, pentru colectarea datelor din zone greu accesibile. Aceasta contribuie la creșterea eficienței, reduce riscurile pentru personal și oferă beneficii concrete în domenii precum agricultura, protecția mediului sau inspecțiile tehnice.

## Referințe

O listă a altor documente la care face referire documentul SRS, inclusiv surse precum site-uri web sau literatură scrisă.

# Descriere generală

## Perspectiva produsului

În ultimii ani, tehnologia sistemelor de control și monitorizare a devenit tot mai prezentă în proiectele educaționale și de cercetare, datorită avantajelor pe care le oferă în gestionarea parametrilor de mediu. Utilizarea senzorilor inteligenți pentru măsurarea în timp real a temperaturii, umidității, presiunii atmosferice sau calității aerului permite obținerea unor informații esențiale pentru evaluarea condițiilor din teren. Comunicația prin Bluetooth și integrarea acestor senzori cu o placă Arduino Mega fac posibilă dezvoltarea unor soluții eficiente, accesibile și ușor de implementat. Alegerea acestei teme este susținută de interesul personal pentru tehnologii moderne și protecția mediului, dorind să demonstrez cum electronica și programarea pot contribui activ la colectarea și interpretarea datelor de mediu într-un mod sustenabil și practic.

## Caracteristici ale produsului

Produsul propus va integra un set de funcții esențiale pentru monitorizarea mediului și controlul de la distanță al unei platforme mobile. Printre caracteristicile principale se numără: deplasarea controlată prin Bluetooth din aplicația mobilă, afișarea și stocarea datelor colectate de la senzori (precum temperatură, umiditate, presiune, nivel de gaz și sunet), evitarea automată a obstacolelor, transmiterea de alerte în cazul detectării unor condiții anormale și controlul manual sau automat al iluminării vehiculului. Aceste funcții vor fi gestionate printr-un sistem bazat pe Arduino Mega, oferind flexibilitate și eficiență în utilizare.

## Clase și caracteristici ale utilizatorilor

Produsul se adresează mai multor tipuri de utilizatori. Utilizatorul final interacționează cu aplicația mobilă pentru a controla vehiculul și a vizualiza datele, fără a necesita cunoștințe tehnice. Dezvoltatorul gestionează integrarea hardware-software și are acces complet la sistem. Testerul validează funcționalitățile în diverse condiții. Fiecare categorie are un nivel diferit de acces și privilegii, în funcție de rolul lor în utilizarea produsului.

## Mediul de operare

Produsul funcționează pe o platformă hardware bazată pe **Arduino Mega 2560**, care controlează componentele precum: **L298N, motoare DC, senzori DHT22, BMP180, MQ, ultrasonic, sunet, lumină**, alături de **LED-uri, buzzer** și **2 acumulatori de 3.7V**. Comunicarea cu utilizatorul se realizează prin **modul Bluetooth HC-05** și o aplicație Android dezvoltată în **MIT App Inventor**, compatibilă cu versiuni Android 6.0 sau mai noi. Codul Arduino este scris în **Arduino IDE**, iar sistemul este conceput să funcționeze stabil în medii externe variate.

## Constrângeri de proiectare și de implementare

Proiectul prezintă mai multe constrângeri tehnice care trebuie luate în considerare atât în faza de proiectare, cât și în implementare. Printre acestea se numără limitările hardware ale plăcii Arduino Mega, care are o capacitate redusă de memorie și procesare, ceea ce impune scrierea unui cod eficient și optimizat. De asemenea, sistemul funcționează la o tensiune totală de 7.4V, ceea ce restricționează alegerea și numărul de componente alimentate. Comunicarea se face exclusiv prin Bluetooth, ceea ce poate afecta stabilitatea conexiunii în anumite condiții. Aplicația mobilă este dezvoltată în MIT App Inventor, o platformă cu limitări în ceea ce privește designul interfeței și complexitatea funcțională. Alte constrângeri includ respectarea convențiilor de programare specifice Arduino și adaptarea sistemului pentru a funcționa fără conexiune permanentă la internet, în medii externe dificile.

## Presupuneri și dependențe

Pentru buna funcționare a sistemului se fac următoarele presupuneri: aplicația mobilă va putea stabili și menține o conexiune Bluetooth stabilă cu modulul HC-05; senzorii utilizați vor fi corect calibrați înainte de utilizare; terenul pe care se deplasează vehiculul va permite mișcarea acestuia fără obstacole imposibil de evitat. De asemenea, alimentarea vehiculului va fi suficientă pentru durata unei sesiuni de colectare de date. Proiectul depinde de buna funcționare a plăcii Arduino Mega, a componentelor electronice conectate și de compatibilitatea aplicației mobile cu dispozitivele Android. Orice întrerupere sau nefuncționare a acestor elemente poate afecta performanța generală a sistemului.

# Cerințele sistemului

Toate cerințele din cadrul sistemului sau sub-sistemului pentru a determina rezultatul pe care se așteaptă să-l ofere produsul în raport cu intrarea dată. Acestea constau în cerințe de design, cerințe grafice, cerințe de sistem de operare.

*Acest șablon ilustrează organizarea cerințelor funcționale pentru produs în funcție de caracteristicile sistemului, principalele servicii furnizate de produs. Poți prefera să organizezi această secțiune în funcție de cazul de utilizare, modul de operare, clasă de utilizatori, clasă de obiecte, ierarhie funcțională sau combinații ale acestora, orice ar face cel mai mult sens logic pentru produsul tău.*

## Controlul deplasarii vehiculului

### Descriere generală

* Această funcționalitate permite utilizatorului să controleze direcția de deplasare a vehiculului (înainte, înapoi, stânga, dreapta) și să oprească mișcarea la nevoie. Este folosită de utilizatorul final prin intermediul aplicației mobile, în contextul navigării vehiculului în teren. Se integrează direct cu sistemul de control al motoarelor, gestionat de placa Arduino Mega și driverul L298N, în funcție de comenzile primite prin Bluetooth.Cine o va utiliza și în ce context

### Flux de interacțiune (scenarii de utilizare)

* Utilizatorul deschide aplicatia mobila si se conecteaza la vehicul prin Bluetooth.Aceste alege, apoi, directia de deplasare(înainte, înapoi, stânga, dreapta) prin apasarea unui buton. Comanda este transmisa modulului Bluetooth si mai departe catre Arduino. Arduino interpreteaza comanda si activeaza motoarele corespunzatoare prin driver. Dupa primirea comenzii, motoarele se deplasează în direcția indicată. La apăsarea butonului de oprire, mișcarea este oprită imediat. Dacă conexiunea se întrerupe, vehiculul se oprește automat.

### Condiții prealabile și constrângeri

* Pentru functionarea optima a sistemului trebuie sa se tina cont de urmatoarele aspecte:

-Sistemul trebuie să fie alimentat corespunzător (minim 7.4V).

-Conexiunea Bluetooth trebuie să fie activă și stabilă.

-Aplicația mobilă trebuie să fie compatibilă cu dispozitivul utilizatorului.

-Comenzile trebuie să fie procesate în timp real (sub 0.5 secunde întârziere).

### Detaliere cerință

Aceasta este lista cerințelor care definesc exact comportamentul funcționalității. Fiecare cerință trebuie să fie:

* **Clară** (fără ambiguități)
* **Testabilă** (poate fi verificată printr-un test).

*<Fiecare cerință ar trebui identificată în mod unic printr-un număr de secvență sau o etichetă semnificativă:>* REQ-1, REQ-2, etc.

REQ-2:

*Exemplu:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| **REQ-1** | Comenzile de direcție trebuie să fie discrete (față, spate, stânga, dreapta). | Medie | Aplicatia ofera butoane separate pentru fiecare directie |
| **REQ-2** | La pierderea conexiunii Bluetooth, vehiculul trebuie să se oprească automat | Ridicată | Testarea întreruperii semnalului duce la oprirea completă a vehiculului |
| **REQ-3** | Utilizatorul trebuie să poată opri vehiculul în orice moment | Ridicată | La apăsarea butonului de oprire, vehiculul se oprește complet |
| **REQ-4** | Sistemul trebuie să răspundă în mai puțin de 0.5 secunde la comenzile de direcție | Ridicată | Comanda transmisă din aplicație determină reacția imediată a vehiculului |

### ****Scenarii de eroare și gestionarea excepțiilor****

* Dacă conexiunea Bluetooth este întreruptă, atunci aplicația pierde conexiunea cu vehiculul, iar Arduino oprește automat toate motoarele pentru a preveni deplasarea necontrolată.
* Dacă două comenzi sunt trimise aproape simultan (ex. înainte și înapoi), sistemul va executa doar ultima comandă primită.
* Dacă un buton din interfață nu trimite comanda corect, se va afișa un mesaj de eroare sau vehiculul nu va reacționa – caz detectabil prin testare manuală.

### ****Dependențe și interacțiuni cu alte funcționalități****

* Această funcționalitate depinde de conexiunea Bluetooth activă între aplicația mobilă și placa Arduino Mega. Ea interacționează direct cu driverul responsabil pentru controlul motoarelor. Controlul motoarelor este influențat indirect de funcționalitatea de evitare a obstacolelor, care poate întrerupe deplasarea dacă un obstacol este detectat. De asemenea, este important ca alimentarea electrică să fie stabilă și comenzile să fie trimise corect.

## Monitorizarea datelor de la senzori

### Descriere generală

* Această funcționalitate permite colectarea și afișarea datelor de mediu de la senzorii integrați în sistem. Utilizatorul poate vedea în aplicație valorile pentru temperatură, umiditate, presiune atmosferică, nivel de gaz, zgomot și lumină. Funcționalitatea este crucială pentru supravegherea condițiilor de mediu în zone inaccesibile.

### Flux de interacțiune (scenarii de utilizare)

* Vehiculul pornește și senzorii încep să colecteze date.Apoi, datele sunt citite de Arduino și transmise prin Bluetooth. Aplicația mobilă afișează valorile în timp real. Utilizatorul poate urmări continuu schimbările din mediu.

### Condiții prealabile și constrângeri

* Pentru functionarea optima a sistemului trebuie sa se tina cont de urmatoarele aspecte:

- Senzorii trebuie să fie funcționali și calibrați.

- Aplicația trebuie să fie conectată la vehicul.

- Actualizarea datelor să se facă la intervale de 5 secunde.

- Să nu existe interferențe care să afecteze măsurătorile.

### Detaliere cerință

|  |  |  |  |
| --- | --- | --- | --- |
| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| **REQ-1** | Sistemul trebuie să colecteze date de la senzorii de temperatură, umiditate, presiune, gaz, sunet și lumină | Ridicată | Toți senzorii oferă valori valide la fiecare ciclu de citire |
| **REQ-2** | Datele trebuie să fie afișate în aplicație la fiecare 5 secunde | Medie | Afișajul aplicației se actualizează automat cu noile valori din teren |
| **REQ-3** | Utilizatorul trebuie să poată vizualiza valorile în timp real | Ridicată | Aplicația prezintă datele imediat ce sunt primite prin Bluetooth |

### Scenarii de eroare si gestionarea exceptiilor

* Dacă un senzor returnează o valoare imposibilă (ex. -100 °C), sistemul o poate afișa cu o etichetă „eroare” sau o valoare implicită.
* Arduino poate semnala lipsa semnalului de la un senzor (dacă este implementată detecția) și aplicația va afișa un mesaj corespunzător.
* Dacă semnalul Bluetooth este întrerupt, datele nu mai ajung în aplicație, iar aceasta poate afișa o avertizare de pierdere a semnalului sau doar ultimele date disponibile.

### ****Dependențe și interacțiuni cu alte funcționalități****

* Această funcționalitate depinde în principal de comunicarea prin Bluetooth, pentru a trimite datele către aplicație, cât și de buna funcționare a plăcii Arduino Mega și a fiecărui senzor conectat.

## Evitarea obstacolelor

### Descriere generală

* Această funcționalitate permite vehiculului să detecteze automat obstacolele aflate în calea sa și să evite coliziunea. Este utilă în deplasarea autonomă sau în situații în care utilizatorul nu observă pericolele din teren. Senzorul ultrasonic este responsabil pentru detectarea obiectelor din proximitate.

### Flux de interacțiune (scenarii de utilizare)

În timpul deplasării vehiculului, senzorul ultrasonic scanează continuu zona din față pentru a detecta eventualele obstacole. Dacă un obiect este identificat la o distanță mai mică de 20 cm, sistemul oprește automat vehiculul pentru a evita coliziunea. Simultan, aplicația mobilă afișează o alertă pentru a informa utilizatorul despre pericolul detectat. Vehiculul rămâne oprit până când este primită o nouă comandă de la utilizator.

### Condiții prealabile și constrângeri

* Pentru functionarea optima a sistemului trebuie sa se tina cont de urmatoarele aspecte:

- Senzorul ultrasonic să funcționeze corect.

- Obiectele detectate să nu fie mai mici decât raza de acoperire a senzorului.

- Sistemul să reacționeze rapid pentru a evita coliziunea.

- Conexiunea între senzor și Arduino să fie stabilă.

### Detaliere cerință

|  |  |  |  |
| --- | --- | --- | --- |
| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| **REQ-1** | Sistemul trebuie să detecteze obstacole la cel puțin 20 cm distanță | Ridicată | Obstacolele apropiate sunt detectate constant de senzorul ultrasonic |
| **REQ-2** | La detectarea unui obstacol, vehiculul trebuie să se oprească automat | Ridicată | La apropierea unui obiect, motoarele se dezactivează imediat |

### Scenarii de eroare si gestionarea exceptiilor

O posibilă eroare poate fi un obstacol cu o dimensiune mică, care se poate afla sub unghiul de detecție al senzorului, deci poate fi ignorat. De asemenea, dacă senzorul este murdar sau blocat, poate returna distanțe incorecte.

### ****Dependențe și interacțiuni cu alte funcționalități****

* Această funcționalitate se bazeaza pe senzorul ultrasonic si pe Arduino Mega pentru interpretarea valorilor. Aceasta interacționează direct cu **funcționalitatea de deplasare**  în momentul detectării unui obstacol, poate forța oprirea motoarelor.

# Cerințe pentru interfețe

Această secțiune descrie modul în care sistemul interacționează cu utilizatorii, hardware-ul, alte sisteme software și rețele de comunicație.

## Interfețe cu utilizatorul

Interfața cu utilizatorul este dezvoltată în MIT App Inventor și este concepută pentru a fi simplă, intuitivă și ușor de utilizat pe orice dispozitiv mobil Android. Pe ecranul principal al aplicației sunt disponibile trei butoane mari, fiecare redirecționând către o pagină dedicată unei funcționalități importante: conectarea prin Bluetooth, controlul vehiculului și afișarea parametrilor de mediu.

Pagina de conexiune Bluetooth permite selectarea și conectarea la modulul vehiculului, oferind informații despre starea conexiunii. Pagina de control conține butoane pentru deplasarea vehiculului în direcțiile principale (înainte, înapoi, stânga, dreapta), precum și un buton pentru oprirea de urgență. În pagina dedicată parametrilor de mediu, utilizatorul poate vizualiza valorile în timp real pentru temperatură, umiditate, presiune atmosferică, gaz, sunet și lumină.

## Interfețe hardware

Această secțiune descrie **cerințele minime hardware** pentru funcționarea sistemului și **dispozitivele externe compatibile**.

### ****Configurații Minime Recomandate****

* **PC/Laptop:** [Procesor: minim Intel Core i3, Memorie RAM: 4 GB, Spațiu de stocare: minim 500 MB disponibili, Sistem de operare: Windows, Port USB pentru conectarea plăcii Arduino]
* **Dispozitive mobile:** [Procesor: minim Quad-core 1.2 GHz, RAM: minim 2 GB, Sistem de operare: Android 6.0 (Marshmallow) sau mai nou, Modul Bluetooth activ]

### ****Dispozitive Externe Compatibile****

* **Placă Arduino Mega 2560** – utilizată pentru controlul general al sistemului
* **Modul Bluetooth HC-05** – pentru comunicarea între vehicul și aplicația mobilă
* **Modul Bluetooth HC-05** – pentru comunicarea între vehicul și aplicația mobilă
* **Driver motoare L298N** – controlează motoarele vehiculului
* **Motoare DC cu roți** – pentru mișcare
* Senzori:
  + DHT22 – temperatură și umiditate
  + BMP180 – presiune atmosferică
  + MQ -05– gaz
  + Senzor ultrasonic – pentru detecția obstacolelor
  + Senzor de sunet și senzor de lumină
  + 4 Senzori Infrared-pentru evitarea obstacolelor
* **LED-uri și buzzer** – semnalizare vizuală și auditivă
* **2 acumulatori Li-Ion 3.7V** – sursă de alimentare portabilă

## Interfețe de comunicare

Sistemul utilizează metode de comunicare simple și eficiente pentru a asigura transmiterea corectă a comenzilor și a datelor între vehicul și aplicația mobilă. Comunicarea se realizează în principal prin conexiune Bluetooth, fiind potrivită pentru distanțe scurte și consum redus de energie.

### ****Protocoale și Standarde de Comunicare****

* **UART (Universal Asynchronous Receiver-Transmitter)** – utilizat pentru comunicarea serială între placa Arduino Mega și modulul Bluetooth HC-05.
* **Bluetooth SPP (Serial Port Profile)** – permite comunicarea între aplicația mobilă și modulul Bluetooth al vehiculului, simulând un port serial clasic.

### ****Cerințe de Securitate în Comunicare****

Modulul Bluetooth HC-05 este configurat cu o parolă personalizată pentru a preveni accesul neautorizat. De asemenea, aplicația mobilă poate fi configurată să recunoască doar adresa MAC a modulului Bluetooth al vehiculului. Întrucât Arduino Mega nu suportă criptare avansată, implementarea unor protocoale precum TLS sau OAuth nu este fezabilă în versiunea actuală, dar poate fi luată în considerare în dezvoltări viitoare, folosind module mai avansate precum ESP32.

## Interfețe software

Sistemul nu interacționează cu aplicații externe sau servicii web în timp real, însă utilizează o serie de tehnologii software locale pentru a gestiona funcționarea întregului sistem și comunicarea dintre componente. Aplicația mobilă și firmware-ul plăcii Arduino colaborează printr-un protocol simplificat, fără a necesita infrastructură server.

### Tehnologii Utilizate

În cadrul acestui proiect nu este utilizat un sistem clasic de tip frontend-backend și nu există baze de date sau servicii externe conectate. Singura componentă software responsabilă de interfața cu utilizatorul și de logica aplicației este aplicația mobilă, dezvoltată în MIT App Inventor. Aceasta gestionează atât partea de afișare (interfața grafică), cât și comunicarea cu vehiculul prin Bluetooth. Pentru programarea plăcii Arduino Mega s-a folosit Arduino IDE, unde codul este scris în C/C++ și utilizează biblioteci specifice pentru senzori și comunicație (precum DHT, BMP180, NewPing, SoftwareSerial). Astfel, întregul sistem este autonom și funcționează local, fără a necesita conexiune la internet sau integrare cu servicii web externe.

### ****Servicii Externe și API-uri****

Acest proiect nu utilizează servicii externe sau API-uri din cloud, deoarece întreaga funcționare este locală, bazată pe comunicația Bluetooth dintre aplicația mobilă și placa Arduino Mega. Toate datele sunt procesate și afișate direct în aplicație, fără a fi necesară conectarea la internet sau integrarea cu platforme terțe.

# Cerințe non-funcționale

<Această secțiune descrie cerințele care definesc **calitatea, performanța, siguranța și securitatea** sistemului, fără a se concentra pe funcționalitatea directă.>

## Cerințe de performanță

Sistemul este proiectat să funcționeze în timp real, motiv pentru care este esențial ca timpul de răspuns la comenzile trimise din aplicația mobilă către vehicul să nu depășească 0,5 secunde. Această viteză este necesară pentru a permite un control precis al deplasării, mai ales în medii cu obstacole sau teren dificil. De asemenea, valorile măsurate de senzorii de mediu trebuie să fie transmise și afișate în aplicație la un interval de cel mult 5 secunde, pentru a asigura o monitorizare constantă și relevantă. În cazul detectării unui obstacol, sistemul trebuie să reacționeze aproape instant, oprind vehiculul într-un timp ideal sub 0,3 secunde. Conexiunea Bluetooth dintre aplicație și vehicul trebuie să fie stabilă pe o distanță minimă de 5 metri, fără întreruperi sau întârzieri semnificative. Aceste cerințe de performanță sunt fundamentale pentru funcționarea sigură și eficientă a sistemului și trebuie respectate pe tot parcursul dezvoltării și testării.

## Cerințe de siguranță

Pentru a preveni orice situație care ar putea pune în pericol integritatea dispozitivului sau a utilizatorului, sistemul trebuie să includă mai multe măsuri de siguranță esențiale. În primul rând, vehiculul trebuie să se oprească automat în cazul pierderii conexiunii Bluetooth, pentru a evita deplasarea necontrolată. De asemenea, dacă este detectat un obstacol în fața vehiculului, mișcarea trebuie întreruptă imediat pentru a preveni coliziunea. Sistemul de alimentare trebuie să includă protecții împotriva scurtcircuitelor și a supraîncărcării acumulatorilor, astfel încât să nu existe riscul de deteriorare a componentelor sau de generare a unui pericol termic. Aplicația mobilă trebuie să fie concepută astfel încât comenzile critice, cum ar fi oprirea de urgență, să fie ușor accesibile și imposibil de declanșat accidental. Toate aceste măsuri au ca scop protejarea atât a echipamentului, cât și a persoanelor care îl utilizează, în condiții de siguranță maximă.

## Cerințe de securitate

Deși sistemul nu stochează date personale și nu este conectat la internet sau la baze de date externe, este important ca accesul la dispozitiv să fie controlat pentru a preveni intervențiile neautorizate. Comunicarea Bluetooth dintre aplicația mobilă și vehicul trebuie să fie permisă doar după o asociere manuală între dispozitive, folosind o parolă personalizată. În aplicație, se recomandă verificarea identității dispozitivului Bluetooth prin adresa MAC, pentru a permite conectarea doar cu vehiculul autorizat. În cazul în care, în versiunile viitoare, se vor adăuga funcții de stocare sau transmitere a datelor către servere externe, va fi necesară respectarea reglementărilor privind confidențialitatea și protecția datelor (cum ar fi Regulamentul GDPR), prin criptarea informațiilor și autentificarea utilizatorilor. În stadiul actual, securitatea este asigurată prin restricționarea accesului fizic și prin configurarea atentă a conexiunii Bluetooth.

## Atribute de calitate ale software-ului

Software-ul dezvoltat în cadrul acestui proiect trebuie să respecte o serie de caracteristici esențiale de calitate pentru a asigura o utilizare eficientă și o posibilă extindere ulterioară. Aplicația mobilă trebuie să fie ușor de utilizat, cu o interfață intuitivă, astfel încât și utilizatorii fără experiență tehnică să o poată folosi fără dificultăți. Codul Arduino trebuie scris clar și modular, pentru a permite întreținerea facilă și modificările ulterioare în cazul în care se adaugă noi funcționalități sau senzori.

Fiabilitatea este un aspect important – sistemul trebuie să funcționeze corect pe durate îndelungate, chiar și în medii externe mai dificile.

Flexibilitatea este susținută de arhitectura modulară, care permite integrarea unor componente suplimentare fără a afecta funcționarea de bază. Portabilitatea este asigurată prin compatibilitatea aplicației cu majoritatea dispozitivelor Android moderne, iar simplitatea sistemului embedded face ca soluția să fie replicabilă ușor și în alte proiecte similare. În ansamblu, software-ul trebuie să fie robust, adaptabil și ușor de întreținut.

# Alte cerințe

Deși sistemul nu utilizează baze de date sau servicii externe, există câteva cerințe suplimentare relevante pentru dezvoltarea și extinderea proiectului. În primul rând, arhitectura software și hardware trebuie concepută astfel încât să permită **extinderea ușoară cu noi senzori** sau funcționalități, fără a fi necesare modificări majore în structura de bază. De asemenea, proiectul poate fi reutilizat în scopuri educaționale sau de cercetare, cu adaptări minime, ceea ce îl face un exemplu bun de **soluție portabilă și scalabilă**.

# Anexe

## Anexa A: Glosar

<Definiți toți termenii necesari pentru a interpreta corect SRS-ul, inclusiv acronime și abrevieri. Puteți dori să construiți un glosar separat care să cuprindă mai multe proiecte sau întreaga organizație și să includeți doar termeni specifici unui singur proiect în fiecare SRS.>

## Anexa B: Modele de Analiză

<Opțional, includeți orice modele de analiză relevante, cum ar fi diagramele de flux de date, diagramele de clasă, diagramele de tranziție de stare sau diagramele de relații entitate-asociere.>

## Anexa C: Listă de Probleme

<Aceasta este o listă dinamică a problemelor de cerințe deschise care rămân de rezolvat, incluzând aspecte care urmează a fi decise în viitor - decizii în așteptare, informații necesare, conflicte așteptând rezolvare și altele asemenea.>

# Exemplu (parțial) de completare

## ****Funcționalitate: Autentificare Utilizator****

### ****3.1.1 Descriere Generală****

Această funcționalitate permite utilizatorilor să se autentifice în sistem folosind o adresă de email și o parolă. Este o funcționalitate esențială pentru gestionarea accesului utilizatorilor.

### ****3.1.2 Flux de Interacțiune****

1. Utilizatorul introduce email-ul și parola
2. Sistemul validează datele
3. Dacă datele sunt corecte → utilizatorul este autentificat
4. Dacă datele sunt greșite → apare un mesaj de eroare

### ****3.1.3 Condiții Prealabile și Constrângeri****

* Utilizatorul trebuie să aibă un cont activ
* Sistemul trebuie să fie conectat la baza de date pentru validarea acreditărilor

### ****3.1.4 Cerințe Funcționale Detaliate****

| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| --- | --- | --- | --- |
| **REQ-1** | Constrangere caractere speciale destinate introducerii unui format corect de email | Mediu | Campul accepta doar date introduce ce respecta formatul email@test.ro |
| **REQ-2** | Verificarea in baza de date a existentei datelor introduse pentru autentificare | Ridicata | Se verifica daca in baza de date exista inregistrari specifice numelui si parolei din campurile de logare |

### ****3.1.5 Scenarii de Eroare și Gestionare a Excepțiilor****

* Dacă utilizatorul introduce o parolă greșită de 5 ori → cont blocat temporar
* Dacă baza de date nu este accesibilă → mesaj de eroare „Serviciu indisponibil”

### ****3.1.6 Dependențe și Interacțiuni cu Alte Funcționalități****

* Depinde de sistemul de gestionare a utilizatorilor
* Se integrează cu sistemul de recuperare a parolei

**4.1 Interfețe cu utilizatorul**

* Sistemul trebuie să ofere o interfață grafică intuitivă, optimizată pentru utilizare pe desktop și mobil.
* Ecranele principale trebuie să includă:
  + **Ecran de autentificare:** câmp pentru email/parolă, buton „Login”
  + **Ecran principal:** navigare prin meniul aplicației
  + **Ecran de setări:** opțiuni de personalizare a contului

**4.2 Interfețe Hardware**

* Aplicația trebuie să ruleze pe următoarele configurații minime:
  + **PC/Laptop:** Procesor minim Intel i5, 8GB RAM, Windows 10/Linux
  + **Smartphone:** Android 9+ sau iOS 14+
* Dispozitive externe compatibile: cititoare de amprente, module NFC

**4.3 Interfețe de Comunicare**

* Aplicația va folosi următoarele protocoale:
  + **HTTP/HTTPS** pentru interacțiunea client-server
  + **WebSockets** pentru notificări în timp real
  + **MQTT** pentru transmiterea datelor IoT
* Datele trebuie să fie criptate utilizând **TLS 1.2+** pentru a asigura securitatea comunicațiilor.

**4.4 Interfețe Software**

* Backend-ul aplicației se va baza pe **Django + MySQL**.
* Aplicația va interacționa cu următoarele API-uri externe:
  + **Google Maps API** pentru localizare
  + **Stripe API** pentru procesarea plăților
  + **Firebase Authentication** pentru gestionarea conturilor de utilizatori