**Documentul de specificare a cerintelor**

**Software Requirements Specification**

**(SRS) Document**

**Dezvoltarea unui sistem automat de colectarea datelor de mediu din zone greu accesibile**

|  |
| --- |
| **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 3](#_Toc159317952)

[1.1 Scopul 3](#_Toc159317953)

[1.2 Convenții ale documentului 3](#_Toc159317954)

[1.3 Audiență țintă 3](#_Toc159317955)

[1.4 Sfera de aplicare 3](#_Toc159317956)

[1.5 Referințe 3](#_Toc159317957)

[**2 Descriere generală 3**](#_Toc159317958)

[**2.1 Perspectiva produsului 3**](#_Toc159317959)

[**2.2 Caracteristici ale produsului 3**](#_Toc159317960)

[**2.3 Clase și caracteristici ale utilizatorilor 3**](#_Toc159317961)

[**2.4 Mediul de operare 4**](#_Toc159317962)

[**2.5 Constrângeri de proiectare și de implementare 4**](#_Toc159317963)

[**2.6 Presupuneri și dependențe 4**](#_Toc159317964)

[**3 Cerințele sistemului 4**](#_Toc159317965)

[**3.1 Funcționalitatea 1 4**](#_Toc159317966)

[**3.1.1 Descriere și Prioritate 4**](#_Toc159317967)

[**3.1.2 Secvențe Stimul/Răspuns 4**](#_Toc159317968)

[**3.1.3 Cerințe Funcționale 5**](#_Toc159317969)

[**3.2 Cerința funcțională 2 5**](#_Toc159317970)

[3.3 .... 5](#_Toc159317971)

[4 Cerințe pentru interfețe externe 5](#_Toc159317972)

[4.1 Interfețe ale Utilizatorului 5](#_Toc159317973)

[4.2 Interfețe Hardware 5](#_Toc159317974)

[4.3 Interfețe de Comunicare 5](#_Toc159317975)

[4.4 Interfețe Software 5](#_Toc159317976)

[5 Cerințe non-funcționale 6](#_Toc159317977)

[5.1 Cerințe de performanță 6](#_Toc159317978)

[5.2 Cerințe de siguranță 6](#_Toc159317979)

[5.3 Cerințe de securitate 6](#_Toc159317980)

[5.4 Atribute de calitate ale software-ului 6](#_Toc159317981)

[6 Alte cerințe 6](#_Toc159317982)

[7 Anexe 6](#_Toc159317983)

[7.1 Anexa A: Glosar 6](#_Toc159317984)

[7.2 Anexa B: Modele de Analiză 6](#_Toc159317985)

[7.3 Anexa C: Listă de Probleme 7](#_Toc159317986)

# Introducere

## Scopul

Scopul proiectului este de a dezvolta un sistem eficient de colectare a datelor de mediu din zone greu accesibile, folosind un RC car echipat cu senzori și controlat prin telecomandă. Prin intermediul acestui sistem, se urmărește monitorizarea parametrilor de mediu precum temperatura, umiditatea, presiunea atmosferică și nivelurile de gaz, oferind astfel informații utile pentru evaluarea și gestionarea calității mediului înconjurător. Scopul final este de a contribui la protejarea și conservarea mediului, oferind date precise și detaliate despre starea acestuia în zonele greu accesibile.

# Descriere generală

## Perspectiva produsului

Context:

Produsul a fost conceput ca răspuns la nevoia de a colecta date precise și detaliate despre mediul înconjurător în zone greu accesibile sau dificil de monitorizat manual. Aceste zone pot include terenuri montane, păduri dense, zone îndepărtate sau chiar zone periculoase pentru acces uman direct, cum ar fi zonele afectate de incendii sau deșeuri toxice.

Originea produsului:

Originea acestui produs poate fi atribuită preocupării crescânde față de mediul înconjurător și nevoii de a monitoriza și gestiona eficient resursele naturale. Dezvoltarea tehnologică rapidă și disponibilitatea unor componente hardware și software accesibile au făcut posibilă conceperea unor soluții inovatoare pentru colectarea datelor de mediu. Ideea de a folosi un RC car echipat cu senzori și controlat prin telecomandă pentru a colecta aceste date a apărut ca o soluție ingenioasă pentru a depăși obstacolele legate de accesibilitate și siguranță.

## Caracteristici ale produsului

Colectarea datelor de mediu: Software-ul va permite colectarea datelor de mediu din zone greu accesibile sau periculoase folosind un RC car echipat cu senzori precum temperatură, umiditate, presiune atmosferică, niveluri de gaz etc.

Transmiterea datelor către server: Software-ul va transmite datele colectate către un server central pentru stocare și analiză ulterioară.

Vizualizarea datelor în timp real: Utilizatorii vor putea vizualiza datele de mediu în timp real, inclusiv grafice și diagrame pentru a evalua evoluția parametrilor de mediu în timp.

Controlul RC car-ului: Software-ul va permite utilizatorilor să controleze mișcările și funcțiile RC car-ului prin intermediul unei telecomenzi.

## Mediul de operare

Platforme hardware:

* RC car echipat cu senzori și controlat prin telecomandă.
* Raspberry Pi , Arduino Nano , Esp8266 si alte dispozitive de calcul single-board pentru procesarea și transmiterea datelor.

Sisteme de operare:

* Raspberry Pi OS
* Sisteme de operare mobile: iOS și Android pentru dispozitivele mobile.
* Sisteme de operare desktop: Windows și Linux pentru computerele personale.

Alte componente software și aplicații:

* Server web cu suport pentru Node.js și MongoDB pentru stocarea și gestionarea datelor.
* Servicii de streaming video pentru transmiterea fluxului video în timp real.
* Framework-uri și biblioteci de dezvoltare pentru aplicații mobile și web, cum ar fi React Native pentru dezvoltarea aplicației client.

## Constrângeri de proiectare și de implementare

Cerințe de performanță: Software-ul trebuie să funcționeze eficient și să gestioneze volumul mare de date colectate și transmise în timp real fără întârzieri sau întreruperi semnificative. Limitările hardware ale dispozitivelor utilizate pentru procesare și transmitere pot reprezenta o constrângere în acest sens.

Compatibilitatea platformelor: Dezvoltarea software-ului pentru a funcționa pe o varietate de dispozitive și sisteme de operare poate fi dificilă din cauza diferențelor în cerințele hardware și software. Asigurarea compatibilității și interoperabilității software-ului cu diverse platforme este o provocare importantă în procesul de dezvoltare.

Interfețe către alte aplicații și servicii: Integrarea software-ului cu alte aplicații și servicii existente poate fi complicată din cauza diferențelor în formatele de date, protocoalele de comunicare și cerințele de securitate. Dezvoltarea interfețelor robuste și flexibile este crucială pentru asigurarea unei colaborări eficiente între diferitele componente ale sistemului.

Cerințe de memorie și procesare: Limitările hardware ale dispozitivelor utilizate pentru implementarea software-ului pot reprezenta o constrângere în ceea ce privește capacitatea de memorie și puterea de procesare disponibilă. Optimizarea codului și gestionarea eficientă a resurselor sunt importante pentru a asigura funcționarea corectă și performanța adecvată a software-ului.

# Cerințele sistemului

Cerințe de design:

Interfață utilizator prietenoasă și intuitivă pentru accesul și utilizarea facilă a funcționalităților software-ului.

Organizarea eficientă a funcțiilor și a informațiilor în aplicație pentru o navigare ușoară și o experiență utilizator plăcută.

Design responsiv și adaptabil pentru compatibilitate cu diverse dispozitive și dimensiuni de ecran.

Coerență și consistență în aspectul și comportamentul elementelor interfeței utilizator.

Cerințe grafice:

Utilizarea unor elemente grafice clare, relevante și ușor de înțeles pentru reprezentarea datelor și a informațiilor.

Grafice și diagrame eficiente pentru vizualizarea și interpretarea datelor de mediu într-un mod accesibil și comprehensibil.

Cerințe de sistem de operare:

Compatibilitate cu sistemele de operare specifice, inclusiv Raspberry Pi OS pentru dispozitivele Raspberry Pi și iOS/Android pentru dispozitivele mobile.

Utilizarea unor tehnologii și biblioteci specifice pentru a asigura funcționarea corectă și optimă a software-ului pe platformele hardware și sistemul de operare targetat.

## Funcționalitatea 1

## Controlul RC car-ului: Utilizatorii pot controla mișcarea și funcțiile RC car-ului prin intermediul unei telecomenzi, permițând explorarea și monitorizarea precisă a mediului înconjurător.

### Descriere și Prioritate

Controlul RC car-ului prin intermediul unei telecomenzi este o caracteristică cu prioritate medie în cadrul proiectului. Această funcționalitate permite utilizatorilor să controleze mișcarea și funcțiile RC car-ului în timpul operațiunilor de monitorizare a mediului înconjurător.

Evaluări specifice ale componentelor priorității:

* Beneficiu: 7 - Permite utilizatorilor să exploreze și să monitorizeze cu precizie mediul înconjurător, oferindu-le controlul asupra mișcării și direcției RC car-ului.
* Penalizare: 5 - Necesită o telecomandă dedicată și o conexiune stabilă pentru a asigura funcționarea corectă, ceea ce poate adăuga costuri și complexitate suplimentare.
* Cost: 6 - Implementarea unei telecomenzi și a tehnologiilor asociate poate implica costuri suplimentare pentru achiziționarea echipamentului și dezvoltarea software-ului necesar.
* Risc: 6 - Există un risc moderat de interferențe sau pierderi de semnal în timpul controlului RC car-ului, ceea ce ar putea afecta precizia și fiabilitatea operațiunilor de monitorizare.

### Secvențe Stimul/Răspuns

Utilizatorul dorește să înceapă monitorizarea mediului înconjurător utilizând RC car-ul:

Acțiunea utilizatorului: Pornirea telecomenzii și a RC car-ului.

Răspunsul sistemului: Inițierea conectării între telecomandă și RC car, confirmarea stării de conexiune și a disponibilității RC car-ului pentru control.

Utilizatorul dorește să direcționeze RC car-ul către o anumită zonă de interes:

Acțiunea utilizatorului: Apăsarea butoanelor direcționale de pe telecomandă pentru a schimba direcția RC car-ului.

Răspunsul sistemului: Mișcarea corespunzătoare a RC car-ului conform comenzilor primite de la telecomandă.

Utilizatorul dorește să controleze viteza RC car-ului:

Acțiunea utilizatorului: Utilizarea butoanelor de control al vitezei de pe telecomandă.

Răspunsul sistemului: Ajustarea vitezei RC car-ului în conformitate cu comenzile primite de la telecomandă.

Utilizatorul dorește să oprească sau să oprească RC car-ul:

Acțiunea utilizatorului: Apăsarea butonului de oprire sau inversare a mișcării pe telecomandă.

Răspunsul sistemului: Oprirea sau inversarea mișcării RC car-ului conform comenzilor primite de la telecomandă.

### Cerințe Funcționale

REQ-1: Sistemul trebuie să primească comenzile de la telecomandă pentru a controla mișcarea și funcțiile RC car-ului.

REQ-2: RC car-ul trebuie să execute comenzile primite de la telecomandă pentru a schimba direcția și viteza.

REQ-3: Monitorizarea datelor de mediu: Sistemul trebuie să poată colecta și să monitorizeze datele de mediu, cum ar fi temperatura, umiditatea, presiunea atmosferică, calitatea aerului și distanța.

REQ-4: Transmiterea datelor către aplicația mobilă: Datele colectate de senzori trebuie să fie transmise în timp real către aplicația mobilă a utilizatorului pentru vizualizare și monitorizare.

REQ-5: Funcționalități de avertizare și notificare: Sistemul trebuie să poată genera avertizări sau notificări către utilizator în cazul în care valorile măsurate depășesc anumite praguri prestabilite pentru parametrii de mediu.

REQ-6: Livestreaming video: Sistemul trebuie să permită utilizatorului să vizualizeze în timp real imagini de pe o cameră video montată pe mașina RC, prin intermediul aplicației mobile.

REQ-7: Comunicare bidirecțională între mașina RC și aplicația mobilă: Utilizatorul trebuie să poată trimite comenzi către mașina RC și să primească feedback în timp real prin intermediul aplicației mobile.

REQ-8: Integrarea cu serverul de pe Raspberry Pi: Mașina RC trebuie să fie capabilă să comunice cu un server localizat pe un Raspberry Pi, care poate gestiona procesarea datelor și alte funcționalități avansate.

# Cerințe pentru interfețe externe

## Interfețe cu utilizatorul

Aplicatia proiectului este realizata cu ajutorul React-Native si Typescript, care permit construirea aplicațiilor mobile native folosind JavaScript și React. Aplicatia este formata din 4 pagini principale: pagina Home , pagina Control , pagina Statistici si pagina pentru Camera.

Pagina Home este pagina unde putem sa gasim valorile in timp real ale senzorilor de mediu.

Pagina Control este pagina de unde putem controla anumiti senzori , cum ar fi led-uri , buzzere , si alti senzori. Tot aici se exista posibilitatea de adaugare a unui nou senzor , prin butonul verde cu + din stanga sus , care odata apasat ne va cere un nume pentru senzor , o comanda care trebuie sa fie sub forma /nume\_senzor , de exemplu /led , si apoi o imagine pentru acest senzor , apoi apasam save si o sa ne apara un nou senzor in lista noastra.

Urmatoarea iconita este folosita pentru state-ul conexiunii la WebSocket , daca este rosie inseamna ca nu este conectat , deci nu poate sa trimita date la server , daca e verde , inseamna ca s-a conectac cu succes.

Mai exista un button de setari prin care , putem modifica Socket Url , daca conexiunea se face pe alt ip , sa ne putem conecta direct din aplicatie , si sub acest input mai este un button prin care putem sa eliminam toate comenzile deja adaugate.

Pagina Statistici este pagina unde se gasesc grafice in timp real , valorile senzorilor de mediu , si statistici , cum ar fi temperatura medie , temperatura maxima , si temperatura minina , la fel si pentru ceilalti senzori.

Pagina Camera este pagina unde se gaseste livestream-ul camerei montate pe macheta si cele doua slidere care trimit datele la Raspberry pentru a putea controla cele doua servo-uri.

Interfața aplicației este concepută să ofere o experiență prietenoasă și ușor de folosit pentru toți utilizatorii. Meniurile și controalele sunt plasate strategic pentru a facilita accesul rapid la funcționalitățile cheie ale aplicației, iar fluxul de navigare este intuitiv și clar. Designul este atrăgător și modern, utilizând o paletă de culori plăcută și o iconografie sugestivă. Elementele vizuale sunt alese cu grijă pentru a crea coerență și consistență în întreaga aplicație.

A screenshot of a phone

Description automatically generated

Figura 1.0 Pagina Camera

Screens screenshot of a device

Description automatically generated

Figura 1.1 Pagina Stats

A screenshot of a phone

Description automatically generated

Figura 1.2 Pagina Control

A screenshot of a car

Description automatically generated

Figura 1.0 Pagina Home

## Interfețe hardware

Interacțiunile hardware-software și dispozitivele acceptate pentru software-ul destinat să ruleze includ:

* Raspberry Pi 3: Este utilizat ca platformă principală pentru software-ul de control al RC car-ului și pentru gestionarea datelor de mediu.
* Arduino Nano: Este utilizat pentru controlul direct al motoarelor și servoului RC car-ului.
* ESP8266: Este utilizat pentru conectivitatea WiFi și comunicația cu alte dispozitive sau servere.
* Senzorul de temperatură și umiditate DHT11: Utilizat pentru măsurarea temperaturii și umidității în mediul înconjurător.
* Senzorul de presiune BMP180: Utilizat pentru măsurarea presiunii atmosferice și a altitudinii.
* Senzorul de lumină: Utilizat pentru măsurarea intensității luminii în mediul înconjurător.
* Senzorul de gaz MQ4: Utilizat pentru detectarea concentrațiilor de gaz combustibil în aer.

Cerințele de rețea includ:

* Conectivitate WiFi: Software-ul trebuie să fie capabil să se conecteze la rețele WiFi disponibile pentru transmiterea datelor către server sau pentru controlul RC car-ului prin intermediul unei rețele locale.
* Protocolul de comunicare WebSocket: Este utilizat pentru comunicarea bidirecțională între aplicație și server pentru transmiterea datelor de mediu și controlul RC car-ului în timp real.

## Interfețe de comunicare

Standardul de comunicare principal utilizat în cadrul proiectului este WebSocket. Acesta permite comunicarea bidirecțională în timp real între aplicație și server pentru transmiterea datelor de mediu și controlul RC car-ului.

Protocolul de rețea WiFi (IEEE 802.11): Folosit pentru conectarea dispozitivelor la rețele WiFi disponibile pentru transmiterea datelor către server sau pentru controlul RC car-ului prin intermediul unei rețele locale.

Protocolul Serial (UART): Utilizat pentru comunicarea între Esp8266 și Arduino Nano pentru transmiterea comenzilor de control ale RC car-ului .

## Interfețe software

Software-ul care urmează să fie dezvoltat va interacționa cu mai multe componente software, inclusiv:

Frontend-ul: Acesta este interfața utilizatorului cu care utilizatorii interacționează direct. Poate fi dezvoltat folosind framework-uri precum React Native pentru aplicații mobile, sau ReactJS și HTML/CSS pentru aplicații web. Frontend-ul va afișa informații despre datele de mediu colectate și va permite utilizatorilor să controleze RC car-ul și să vizualizeze livestream-ul camerei.

Backend-ul: Este responsabil pentru gestionarea logicii de afaceri, procesarea datelor și comunicarea cu baza de date. Poate fi dezvoltat folosind framework-uri precum Node.js pentru JavaScript și TypeScript. Backend-ul va gestiona solicitările de la frontend, va comunica cu dispozitivele hardware și va gestiona transmiterea datelor către și de la baza de date.

Sistemul de gestionare a bazelor de date: Va stoca și organiza datele colectate de dispozitivele hardware și alte informații relevante pentru funcționarea sistemului. Un sistem de gestionare a bazelor de date cum ar fi MongoDB poate fi utilizat pentru a asigura stocarea eficientă și accesul rapid la date.

Bibliotecile asociate: Acestea includ biblioteci specifice pentru comunicarea cu dispozitivele hardware, cum ar fi bibliotecile pentru controlul servo-motorului și a senzorilor de mediu, precum și biblioteci pentru gestionarea comunicării WebSocket între frontend și backend.

# Cerințe non-funcționale

## Cerințe de performanță

Cerințele de performanță pentru produsul nostru sunt cruciale pentru a asigura o experiență fluidă și eficientă a utilizatorului în diverse circumstanțe. Acestea includ:

* Latența minimă pentru comenzi de la utilizator la acțiunea efectivă a RC car-ului: Este esențial ca timpul dintre trimiterea comenzilor de la utilizator prin intermediul aplicației și efectuarea corespunzătoare a acțiunilor de către RC car să fie cât mai mic posibil. O latență redusă asigură un control precis și în timp real al vehiculului, permițând utilizatorului să interacționeze cu mediul înconjurător fără întârzieri semnificative.
* Capacitatea de a gestiona volumul crescut de date de mediu în timp real: Având în vedere că sistemul nostru colectează și transmite date de mediu în timp real, este important ca software-ul să fie capabil să gestioneze eficient un volum mare de date fără a afecta performanța generală a sistemului. Acest lucru implică optimizarea procesării și transmiterii datelor pentru a asigura că sistemul poate funcționa fără întreruperi chiar și în condiții de încărcare ridicată.
* Fiabilitatea și stabilitatea conexiunii WebSocket: Conexiunea WebSocket între aplicație și server trebuie să fie stabilă și fiabilă pentru a asigura o comunicare continuă și fără întreruperi între utilizator și RC car. Este esențial ca sistemul să gestioneze eficient reconectarea în cazul unei pierderi temporare a conexiunii și să ofere mecanisme adecvate pentru gestionarea erorilor și recuperarea conexiunii.
* Timpul de răspuns al aplicației: Aplicația trebuie să ofere un timp de răspuns rapid la interacțiunile utilizatorului, inclusiv la trimiterea comenzilor și la afișarea actualizărilor de stare. Un timp de răspuns scăzut contribuie la o experiență utilizator mai plăcută și la creșterea eficienței în controlul RC car-ului și monitorizarea mediului înconjurător.

# Anexe

## Anexa B: Modele de Analiză

A diagram of a flowchart

Description automatically generated