

Documentul de specificare a cerințelor Software Requirements Specification (SRS) Document

***Masina autonoma cu sisteme de prevenire a accidentelor
rutiere***

05.05.2025

Istoricul versiunilor

Versiune	Autor(i) principali	Descriere versiune	Data

Revizuiți și aprobări

Istoric aprobări

Aprobă	Versiune	Semnătură	Data

Istoric revizuiți

Revizor	Versiune	Semnătură	Data

Cuprins

1.	Introducere	4
1.1	Scopul	4
1.2	Convenții ale documentului	4
1.3	Audiență țintă	4
1.4	Sfera de aplicare	4
1.5	Referințe	4
2	Descriere generală	4
2.1	Perspectiva produsului	4
2.2	Caracteristici ale produsului	4
2.3	Clase și caracteristici ale utilizatorilor	4
2.4	Mediul de operare	5
2.5	Constrângeri de proiectare și de implementare	5
2.6	Presupuneri și dependențe	5
3	Cerintele sistemului	5
3.1	Funcționalitatea 1	5
3.1.1	Descriere generală	5
3.1.2	Flux de interacțiune (scenarii de utilizare)	5

3.1.3	Condiții prealabile și constrângeri	6
3.1.4	Detaliere cerință	6
3.1.5	Scenarii de eroare și gestionarea excepțiilor	6
3.1.5	Dependente și interacțiuni cu alte funcționalități	6
3.2	Cerința funcțională 2	6
3.3	6
4	Cerințe pentru interfețe	6
4.1	Interfețe cu utilizatorul	6
4.2	Interfețe hardware	6
4.2.1	Configurații Minime Recomandate	7
4.2.2	Dispozitive Externe Compatibile	7
4.3	Interfețe de comunicare	7
4.3.1	Protocoale și Standarde de Comunicare	7
4.3.2	Cerințe de Securitate în Comunicare	7
4.4	Interfețe software	7
4.4.1	Tehnologii Utilizate	7
4.4.2	Servicii Externe și API-uri	7
5	Cerințe non-funcționale	8
5.1	Cerințe de performanță	8
5.2	Cerințe de siguranță	8
5.3	Cerințe de securitate	8
5.4	Atribute de calitate ale software-ului	8
6	Alte cerințe	8
7	Anexe	8
7.1	Anexa A: Glosar	8
7.2	Anexa B: Modele de Analiză	8
7.3	Anexa C: Listă de Probleme	9

• Introducere

• Scopul

Acest document descrie cerințele funcționale și non-funcționale pentru sistemul de conducere autonomă al unei mașini echipate cu senzori și o aplicație mobilă, proiectată pentru a preveni accidentele rutiere.

• Convenții ale documentului

REQ-xx: Identificatori pentru cerințe funcționale

NFR-xx: Identificatori pentru cerințe non-funcționale

• Audiență țintă

Studenti

Coordonatori/profesori

Dezvoltatori software

Evaluatori tehnici

- **Sfera de aplicare**

Proiectul constă în realizarea unei mașini autonome capabile să circule singură, să evite obstacole, să oprească la semafor și treceri de pietoni, să mențină banda de circulație și distanța față de alte vehicule. Sistemul trimite o alertă într-o aplicație mobilă în caz de accident.

- **Referințe**

Specificații Arduino Uno

Documentație LAFVIN

Documentație senzor ultrasonic HC-SR04

Documentație senzor IR

- **Descriere generală**

- **Perspectiva produsului**

Sistemul este format dintr-un vehicul fizic dotat cu senzori și un modul de comunicație cu aplicația mobilă. Comenzile sunt gestionate de un microcontroler Arduino Uno.

- **Caracteristici ale produsului**

Navigare autonomă

Evitarea obstacolelor

Oprire la semafor

Menținere bandă

Menținere distanță

Alertă în caz de accident

- **Clase și caracteristici ale utilizatorilor**

Utilizator tehnic: programatori/dezvoltatori

Utilizator final: student care operează vehiculul pentru demonstrații

- **Mediul de operare**

Arduino IDE pe Windows

Smartphone cu iOS pentru aplicația de alertă

Conectivitate prin Bluetooth/WiFi

- **Constrângeri de proiectare și de implementare**

Limitări de alimentare (două baterii)

Procesare limitată de Arduino Uno

- **Presupuneri și dependențe**

Senzorii sunt funcționali și calibrați corect

Conexiunea cu aplicația mobilă este stabilă

- **Cerințele sistemului**

- **Funcționalitatea 1**

1: Rulare autonomă

REQ-1: Sistemul trebuie să permită vehiculului să se deplaseze autonom pe o pistă prestabilită

REQ-2: Vehiculul trebuie să recunoască și să se oprească la culoarea roșie a semaforului

REQ-3: Vehiculul trebuie să mențină distanța de cel din față

REQ-4: Vehiculul trebuie să evite obstacole statice sau dinamice

REQ-5: Vehiculul trebuie să rămână pe banda de mers marcată vizual

Descriere generală

- Explică scopul general al funcționalității
- Cine o va utiliza și în ce context
- Cum se integrează în sistemul general

- **Flux de interacțiune (scenarii de utilizare)**

1. Inițializare sistem

2. Start rulare

3. Detectare obstacol/semafor

4. Decizie de frânare/opri/evitare

5. Continuare deplasare

Condiții prealabile și constrângeri

- **Detaliere cerință**

Pista trebuie să fie pregătită cu marcaje și semafoare

Conexiunea la baterii este activă

<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	Utilizatorul trebuie să poată introduce date în formularul X	Ridicăta	Câmpurile acceptă date valide
REQ-2	Sistemul trebuie să valideze automat datele introduse	Medie	Se afișează un mesaj de eroare pentru date incorecte

- **Scenarii de eroare și gestionarea excepțiilor**

Senzor defect: activare mod de siguranță

Lipsă comunicație: vehiculul se oprește

- **Dependențe și interacțiuni cu alte funcționalități**

Funcționalitatea de menținere a benzii (REQ-5) este necesară pentru rularea autonomă corectă (REQ-1)

Funcționalitatea de oprire la semafor (REQ-2) se bazează pe detectarea corectă a culorii de către cameră

Evitarea obstacolelor (REQ-4) este dependentă de citirile senzorului ultrasonic și poate influența deciziile din REQ-1 și REQ-3

- **Cerința funcțională 2**

-

- **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**

Buton de start/stop rulare pe aplicația mobilă

Notificare text în caz de accident

- **Interfețe hardware**

- Senzori IR (3)

- Senzor ultrasonic HC-SR04

- Cameră pentru recunoaștere culoare semafor

- Modul LAFVIN pentru comunicații

- **Interfețe de comunicare**

Bluetooth: comunicare între Arduino și aplicația mobilă

UART Serial: comunicare internă între modulele Arduino

HTTP/HTTPS: pentru trimitere de notificări în aplicația mobilă (prin API)

Potențial MQTT: pentru extensii IoT viitoare

- **Cerințe de Securitate în Comunicare**

- Toate datele transmise trebuie să fie criptate (ex. TLS 1.2+).
- Autentificarea utilizatorilor în sistem trebuie să respecte **OAuth 2.0** sau alt protocol standard.

- **Interfețe software**

Aplicație pe android pentru alertă

Cod C++ pentru Arduino

API de notificare (Firebase sau altul)

- **Tehnologii Utilizate**

Arduino Uno: microcontroler principal pentru procesarea senzorilor și controlul motorului

Arduino IDE: mediu de dezvoltare pentru codul C++

Senzori IR și ultrasonic HC-SR04: pentru detectarea benzii și obstacolelor

Cameră compatibilă Arduino: pentru detecția culorii semaforului

Modul LAFVIN (Bluetooth/WiFi): pentru comunicație cu aplicația mobilă

Aplicație iOS dezvoltată în Swift: pentru recepția notificărilor de alertă

Firebase (sau serviciu alternativ): pentru trimiterea notificărilor push

Compatibilitate Arduino IDE

Aplicația iOS trebuie să funcționeze offline pentru notificări locale

Compatibilitate Arduino IDE

Serviciu/Interfață	Utilizare
[Serviciu/API 1]	[Ex. Google Maps API pentru localizare]
[Serviciu/API 2]	[Ex. Stripe API pentru procesare plăți]
[Serviciu/API 3]	[Ex. Firebase pentru autentificare]

- **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**

Dacă există cerințe de performanță pentru produs în diverse circumstanțe, specificați-le aici și explicați raționamentul lor, pentru a ajuta dezvoltatorii să înțeleagă intenția și să facă alegeri de proiectare potrivite. Specificați relațiile de timp pentru sistemele în timp real. Faceți aceste

cerințe cât mai clare posibil. Este posibil să fie necesar să specificați cerințele de performanță pentru cerințele funcționale individuale sau caracteristici.

- **Cerințe de siguranță**

Enumerați toate măsurile de precauție care trebuie incluse pentru a preveni orice posibile prejudicii pe care utilizarea aplicației software le-ar putea cauza.

- **Cerințe de securitate**

Reglementările privind confidențialitatea și protecția datelor care trebuie respectate în timpul proiectării produsului.

- **Atribute de calitate ale software-ului**

Detalierea calităților suplimentare care trebuie incorporate în software, cum ar fi *ușurința în întreținere, adaptabilitatea, flexibilitatea, utilizabilitatea, fiabilitatea, portabilitatea* etc.

- **Alte cerințe**

Definiți orice alte cerințe care nu sunt acoperite în altă parte a SRS. Acestea ar putea include cerințe legate de baze de date, cerințe de internaționalizare, cerințe legale, obiective de reutilizare pentru proiect și altele asemenea. Adăugați orice secțiuni noi care sunt de interes pentru proiect.

- **Anexe**

- **Anexa A: Glosar**

IR: Infraroșu

HC-SR04: Senzor ultrasonic

- **Anexa B: Modele de Analiză**

- Diagrama de flux pentru evitarea obstacolului

- **Anexa C: Listă de Probleme**

Stabilitatea conexiunii Bluetooth

Calibrarea exactă a distanței pe senzorii IR

- **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**

- Utilizatorul introduce email-ul și parola
- Sistemul validează datele
- Dacă datele sunt corecte → utilizatorul este autentificat
- 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 Detaliat**

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 Dependente 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