**Documentul de specificare a cerințelor**

**Software Requirements Specification**

**(SRS) Document**

**Sistem inteligent pentru reducerea congestionării traficului**

**<Data><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 3](#_Toc190689617)

[1.1 Scopul 3](#_Toc190689618)

[1.2 Convenții ale documentului 3](#_Toc190689619)

[1.3 Audiență țintă 3](#_Toc190689620)

[1.4 Sfera de aplicare 3](#_Toc190689621)

[1.5 Referințe 3](#_Toc190689622)

**[2 Descriere generală 3](#_Toc190689623)**

**[2.1 Perspectiva produsului 3](#_Toc190689624)**

**[2.2 Caracteristici ale produsului 3](#_Toc190689625)**

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

**[2.4 Mediul de operare 4](#_Toc190689627)**

**[2.5 Constrângeri de proiectare și de implementare 4](#_Toc190689628)**

**[2.6 Presupuneri și dependențe 4](#_Toc190689629)**

**[3 Cerințele sistemului 4](#_Toc190689630)**

**[3.1 Funcționalitatea 1 Error! Bookmark not defined.](#_Toc190689631)**

**[3.1.1 Descriere generală 4](#_Toc190689632)**

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

**[3.1.3 Condiții prealabile și constrângeri 5](#_Toc190689634)**

**[3.1.4 Detaliere cerință 5](#_Toc190689635)**

**[3.1.5 Scenarii de eroare și gestionarea excepțiilor 5](#_Toc190689636)**

**[3.1.5 Dependențe și interacțiuni cu alte funcționalități 5](#_Toc190689637)**

**[3.2 Cerința funcțională 2 5](#_Toc190689638)**

[3.3 .... 5](#_Toc190689639)

[4 Cerințe pentru interfețe 7](#_Toc190689640)

[4.1 Interfețe cu utilizatorul 7](#_Toc190689641)

[4.2 Interfețe hardware 7](#_Toc190689642)

[4.2.1 Configurații Minime Recomandate 7](#_Toc190689643)

[4.2.2 Dispozitive Externe Compatibile 8](#_Toc190689644)

[4.3 Interfețe de comunicare 8](#_Toc190689645)

[4.3.1 Protocoale și Standarde de Comunicare 8](#_Toc190689646)

[4.3.2 Cerințe de Securitate în Comunicare 8](#_Toc190689647)

[4.4 Interfețe software 8](#_Toc190689648)

[4.4.1 Tehnologii Utilizate 8](#_Toc190689649)

[4.4.2 Servicii Externe și API-uri 8](#_Toc190689650)

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

[5.1 Cerințe de performanță 9](#_Toc190689652)

[5.2 Cerințe de siguranță 9](#_Toc190689653)

[5.3 Cerințe de securitate 9](#_Toc190689654)

[5.4 Atribute de calitate ale software-ului 9](#_Toc190689655)

[6 Alte cerințe 9](#_Toc190689656)

[7 Anexe 9](#_Toc190689657)

[7.1 Anexa A: Glosar 9](#_Toc190689658)

[7.2 Anexa B: Modele de Analiză 9](#_Toc190689659)

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

# Introducere

## Scopul

Scopul documentului.

## 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

Descrie care parte a documentului este destinată fiecărui cititor. Include o listă a tuturor părților interesate ale proiectului, dezvoltatorilor, managerilor de proiect și testerilor pentru o mai bună claritate.

## Sfera de aplicare

Specifică cum se aliniază obiectivele produsului cu obiectivele generale ale sistemului în care se integrează produsul și conturează beneficiile proiectului pentru afacere.

## 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

Produsul este o aplicatie de tip simulator, in care utilizatorul poate vizualiza modul in care traficul se dezvolta intr-o intersectie semaforizata, in care semafoarele pot fi configurate pentru a se comporta in mai multe moduri, de exemplu: clasic in care fiecare culoare are un timer prestabilit sau modul “semafor inteligent” in care in functie de fluxul traficului pe fiecare sens.

## Caracteristici ale produsului

Simulatorul va contine urmatoarele functionalitati :

- Posibilitatea de adaugare a diferite tipuri de strazi, selectarea numarului de benzi pentru fiecare strada, modificarea limitei de viteza pe fiecare sectiune.

-Posibilitatea de a adauga diferite semne de circulatie (stop, cedeaza trecerea, trecere de pietoni)

-Posibilitatea de a adauga diferite tipuri de semafoarea si modificarea comportamentului acestora.

-Pornirea/oprirea simularii, prin pornirea simularii se vor genera un numar de vehicule ce vor circula prin intersectia creata anterior, acestea vor respecta legile de circulatie implicit semnele pozitionate de utilizator in cea de a doua caracteristica.

## Clase și caracteristici ale utilizatorilor

Va exista o singura clasa de utilizator, care va avea acces la toate functionalitatile aplicatiei.

## Mediul de operare

Aplicatia principala in care proiectul se realizeaza este Unity

Limbajul de programarea ales este C#

## Constrângeri de proiectare și de implementare

Unul dintre factorii de limitarea ar fi performanta sistemului pe care este rulata aplicatia, dat fiind faptul ca la pornirea simularii comportamentul fiecarui vehicul este gestionat de un script lipsa de resurse ale calculatorului vor duce la o performanta mai slaba a aplicatiei.

## Presupuneri și dependențe

Deoarece Unity este o platforma foarte populara folosita la nivel global am presupus ca nu voi duce lipsa de resurse externe, totusi nu voi putea implementa functionalitati pe care Unity nu le are.

# 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.*

## Generarea strazilor / intersectiilor unde traficul se va desfasura

### Descriere generală

* Permite utilizatorului sa genereze un sistem de drumuri si intersectii.
* Utilizatorul principal va genera strazi si intersectii ca prim pas in pornirea simularii
* Este functionalitatea principala, toate celelalte functionalitati depind de aceasta fara generarea strazilor/intersectiilor simularea nu va putea fi pornita.

### Flux de interacțiune (scenarii de utilizare)

* Utilizatorul va genera mai multe strazi printr-un sistem care are la baza Splines Unity Package, astfel acesta poate genera drumuri cat mai similare cu cele din realitate.
* In momentul in care utilzatorul alege sa genereze un drum acesta va da click in locul de unde doreste sa porneasca iar apoi click in locul unde drumul se va termina. Pe parcursul drumului vor exista noduri care permit schimbarea curburii in functie de dorinitele utilizatorului.

### Condiții prealabile și constrângeri

* Nu exista drumuri ce trec unul prin altul, in momentul in care doua drumuri se intalnesc intr-un punct atunci in acel loc se va genera o intersectie. Daca intersectia nu este generata utilizatorul nu va putea sa configureze un semafor implicit nu va putea porni simularea.

### Detaliere cerință

|  |  |  |  |
| --- | --- | --- | --- |
| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| **REQ-1** | Utilizatorul trebuie sa genereze minim doua strazi. | Ridicată | Mesaj de eroare daca nu se ajunge la numarul minim |
| **REQ-2** | Utilizatorul trebuie sa creeze o intersectie | Ridicata | Mesaj de eroare deoarece fara intesectie nu exista un semafor in simulare |

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

* Daca utilizatorul nu respecta conditiile de mai sus atunci programul il informeaza in functie de caz de ce este nevoie pentru a porni.

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

* Functionalitatile depind una de celalta “in scara” aceasta este prima deci toate celalte functionalitati depind de aceasta.

## Configurarea unui semafor in toate intersectiile generate

### Descriere generală

* Pentru fiecare intersectie se poate alege un tip de semafor care va fi respectat de traficul simulat
* Utilizatorul va putea alege tipul de semafor o data ce are cel putin o intersectie generata
* Este cea de a doua functionalitate dupa generarea strazilor/intersectiilor si depinde de aceasta.

### Flux de interacțiune (scenarii de utilizare)

* Prin selectarea unei intersectii se va afisa un pop up in care se va putea alege tipul de semafor.

### Condiții prealabile și constrângeri

* Nu se poate porni simularea daca una sau mai multe intersectii nu au semafor.

### Detaliere cerință

|  |  |  |  |
| --- | --- | --- | --- |
| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| **REQ-1** | Utilizatorul trebuie sa aleaga un tip de semafor pentru fiecare intersectie | Ridicată | Mesaj de eroare daca exista intersectie fara semafor configurat |
| **REQ-2** | Se vor alege directiile posibile pentru fiecare banda | Mica | Daca nu sunt alese atunci se vor lua directiile clasice (banda stanga - inainte/stanga ; banda dreapta - inainte/dreapta) |

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

* Daca exista intersectii fara semafor atunci nu se va putea porni simularea.

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

* Functionalitatea depinde de generarea strazilor/intersectiilor iar pornire a simularii depinde de aceasta.

## Alegerea punctelor de start/stop si pornirea simularii

### Descriere generală

* Fiecare vehicul din simulare se va orienta dupa punctele de start stop pe care utilizatorul le va putea alege la orice capat de strada. Dupa ce sunt alese punctele simularea poate incepe. Rolul acestora este de a replica directia traficului intre doua zone de exemplu centru si periferie.
* Utilizatorul alege punctele de start/stop dupa ce strazile si intersectiile au fost configurate corespunzator.
* Este funtionalitatea finala si depinde de celelate doua.

### Flux de interacțiune (scenarii de utilizare)

* Se aleg perechi de puncte care vor fi locul din care vehiculele vor pleca/ajunge, intre cele doua puncte se poate alege numarul de vehicule pe minut, astfel se pot simula diferite scenarii.

### Condiții prealabile și constrângeri

* Vor putea exista mai multe puncte de start/stop la un capat de strada.

### Detaliere cerință

|  |  |  |  |
| --- | --- | --- | --- |
| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| **REQ-1** | Utilizatorul trebuie sa aleaga punctele de start/stop. | Ridicată | Nu exista punct de start fara un punct de stop, in cazul acesta se primeste un mesaj de eroare. |
| **REQ-2** | Sistemul verifica automat daca este posibila deplasarea intre perechile de puncte | Medie | Daca deplasarea vehiculelor nu este posibila intre puncte atunci se primeste un mesaj de eroare. |

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

* Toate conditiile de mai sus trebuie sa fie respectate pentru pornirea simularii.

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

* Functionalitatile depinde de toate celelate functionalitati pentru a putea fi folosita.

# 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

Această secțiune descrie interfața grafică sau text-based a sistemului, incluzând **aspecte de design și cerințe de accesibilitate**. Aici se pot include ecrane demonstrative (funcționalitățile care ar apărea pe fiecare ecran, mesajele care urmează să fie afișate pe fiecare ecran și ghidurile de stil care urmează să fie utilizate). Detaliile designului interfeței utilizatorului ar trebui să fie documentate într-o specificație separată a interfeței utilizatorului.

## Interfețe hardware

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

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

* **PC/Laptop:** [Specificații minime – CPU, RAM, spațiu de stocare]
* **Dispozitive mobile:** [Specificații minime – procesor, versiune OS, RAM]

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

* **[Dispozitiv 1]** – [Ex. Scanner de coduri de bare, cititor NFC]
* **[Dispozitiv 2]** – [Ex. Cameră foto]

## Interfețe de comunicare

Această secțiune descrie protocoalele și metodele de **comunicare a sistemului.**

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

* **[Protocol 1]** – [Ex. HTTP/HTTPS pentru interacțiunea client-server]
* **[Protocol 2]** – [Ex. WebSockets pentru notificări în timp real]
* **[Protocol 3]** – [Ex. MQTT pentru integrare cu dispozitive IoT]

### ****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

**Această secțiune descrie interacțiunea sistemului cu alte aplicații sau servicii externe.**

### Tehnologii Utilizate

* **Backend:** [Ex. Django, Node.js, .NET Core]
* **Frontend:** [Ex. React, Angular, Vue.js]
* **Bază de date:** [Ex. PostgreSQL, MySQL, MongoDB]

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

| **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

<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