**Sistem de navigare**

**Coordonator,**

**Prof. Florin Leon**

**Studenți,**

**Chilimon Ana-Maria, Grupa 1312A**

**Chiriac Raluca-Ștefania, Grupa 1312A**

**Gălusca Mihnea-Ioan, Grupa 1312A**

**Stoian Mario-Daniel, Grupa 1312A**

**Iasi 2025**

**Cuprins**

**Cuprins................................................................................................................................ ii**

**1. Introducere.......................................................................................................................3**

1.1 Scopul Documentului..........................................................................................3

1.2 Convenții Utilizate în Document.........................................................................3

1.3 Audiența Vizată și Sugestii de Citire...................................................................3

1.4 Domeniul Produsului...........................................................................................3

1.5 Referințe...............................................................................................................3

**2. Descriere Generală...........................................................................................................4**

2.1 Perspectiva Produsului.........................................................................................4

2.2 Funcțiile Produsului.............................................................................................4

2.3 Clase de Utilizatori și Caracteristici.....................................................................4

2.4 Mediul de Operare................................................................................................4

2.5 Constrângeri de Proiectare și Implementare.........................................................4

2.6 Documentația Utilizatorului..................................................................................5

2.7 Ipoteze și Dependențe...........................................................................................5

**3. Cerințe pentru Interfețe Externe.....................................................................................5**

3.1 Interfețe Utilizator.................................................................................................5

3.2 Interfețe Hardware.................................................................................................8

3.3 Interfețe Software..................................................................................................8

3.4 Interfețe de Comunicații........................................................................................9

**4. Caracteristicile Sistemului ...............................................................................................9**

4.1 Caracteristica 1: Introducerea și Geocodarea Locațiilor.......................................9

4.2 Caracteristica 2: Calcularea și Afișarea Rutelor...................................................9

4.3 Caracteristica 3: Selecția și Evidențierea Rutei.....................................................9

4.4 Caracteristica 4: Calcularea și Afișarea Metricilor de Călătorie...........................9

4.5 Caracteristica 5: Managementul Stării Interfeței Utilizator..................................9

4.6 Caracteristica 6: Accesul la Sistemul de Ajutor (Help)........................................9

**5. Alte Cerințe Non-funcționale..........................................................................................10**

5.1 Cerințe de Performanță........................................................................................10

5.2 Cerințe de Siguranță.............................................................................................10

5.3 Cerințe de Securitate............................................................................................10

5.4 Atribute de Calitate Software...............................................................................10

5.5 Reguli de Business...............................................................................................10

**6. Alte Cerințe ......................................................................................................................10** Anexa A: Modele de Analiză (Diagrame UML)....................................................................10 Anexa B: Cod sursă important...............................................................................................17

**1. Introducere**

* 1. **Scopul Documentului**

Acest document specifică cerințele software pentru aplicația "Simulare Sistem de Navigație". Aplicația permite utilizatorilor să introducă o locație de plecare și una de sosire, să vizualizeze rute posibile pe o hartă interactivă și să obțină estimări privind timpul de parcurs și consumul de combustibil.

* 1. **Convenții Document**

Documentul utilizează fontul Times New Roman, mărimea 12, spațiere la un rând. Termenii specifici sau acronimele vor fi explicate în secțiunea Definiții sau în Glosar. Cerințele funcționale vor fi identificate cu un prefix "REQ-FUNC-" urmat de un număr unic. Cerințele non-funcționale vor folosi prefixul "REQ-NONFUNC-".

* 1. **Audiență Vizată și Sugestii de Citire**

Audiența acestui document include echipa de dezvoltare, coordonatorul proiectului și orice persoană interesată de funcționalitățile și constrângerile sistemului. Se recomandă citirea secțiunii 2 pentru o imagine de ansamblu, urmată de secțiunile 3 și 4 pentru detalii tehnice.

* 1. **Domeniul Produsului**

Aplicația "Simulare Sistem de Navigație" este un instrument desktop dezvoltat în C# ce oferă funcționalități de bază pentru planificarea rutelor. Utilizează servicii externe pentru geocodarea adreselor și pentru obținerea datelor de rutare. Principalul beneficiu este simularea procesului de navigație și calculul estimativ al parametrilor de călătorie.

* 1. **Referințe**
* Standard IEEE 830-1998 - Recommended Practice for Software Requirements Specifications.
* Documentația API OSRM (Open Source Routing Machine).
* Documentația API Nominatim (OpenStreetMap).
* Documentația bibliotecii GMap.NET.
* Librărie Newtonsoft.Json pentru parsare JSON.

1. **Descriere Generală**
   1. **Perspectiva Produsului**

Aplicația este un produs software independent, dezvoltat ca proiect în cadrul disciplinei Ingineria Programării. Interacționează cu servicii web externe (OSRM, Nominatim) pentru a prelua date geografice și de rutare. Nu înlocuiește un sistem de navigație comercial, ci simulează funcționalități de bază.

* 1. **Funcțiile Produsului**

Principalele funcții ale sistemului includ:

* Introducerea locațiilor de plecare și sosire sub formă de text.
* Geocodarea adreselor introduse în coordonate geografice.
* Calcularea și afișarea pe hartă a uneia sau mai multor rute alternative.
* Listarea rutelor alternative și permiterea selecției uneia dintre ele.
* Evidențierea rutei selectate pe hartă.
* Introducerea opțională a vitezei medii de deplasare.
* Calcularea și afișarea timpului estimat de parcurs și a consumului estimat de combustibil pentru ruta selectată.
* Furnizarea unui sistem de ajutor (Help) pentru utilizator.
  1. **Clase de Utilizatori și Caracteristici**

Există o singură clasă principală de utilizatori:

Utilizator General: O persoană care dorește să simuleze planificarea unei rute, fără a necesita cunoștințe tehnice avansate. Utilizatorul interacționează cu interfața grafică pentru a introduce date și a vizualiza rezultate.

* 1. **Mediul de Operare**
* Aplicația va rula pe sisteme de operare Microsoft Windows care suportă .NET Framewor - versiunea 4.8
* Necesită o conexiune activă la internet pentru interogarea serviciilor OSRM și Nominatim.
* Necesită instalarea bibliotecii GMap.NET și Newtonsoft.Json (sau acestea sunt incluse/distribuite cu aplicația).
  1. **Constrângeri de Proiectare și Implementare**
* Limbajul de programare: C#.
* Platforma: .NET Framework.
* Interfața grafică: Windows Forms.
* Control hartă: GMap.NET.
* Serviciu de rutare: OSRM
* Serviciu de geocodare: Nominatim
* Formatul de date pentru API-uri: JSON.
* Design Pattern: Aplicația utilizează State Design Pattern pentru gestionarea stărilor interfeței grafice (ex: `InitialState`, `LoadingState`, `RoutesDisplayedState`, `RouteSelectedState` definite în `ApplicationState.cs`). Acest pattern permite o mai bună organizare a logicii UI și tranzițiilor între diferite moduri de operare.
* Arhitectură modulară: Funcționalitățile specifice (serviciu de rutare, calcul metrici, nume culori) sunt implementate în DLL-uri separate (`RoutingServiceDLL.dll`, `RouteCalculatorDLL.dll`, `ColorNameDLL.dll`) conform cerințelor proiectului.
  1. **Documentație Utilizator**
* Acest document SRS.
* Un fișier de ajutor în format CHM (`Sistem-Simulare-de-Navigare\_Help.chm`).
  1. **Ipoteze și Dependențe**
* Serviciile OSRM și Nominatim sunt disponibile și funcționale.
* Datele returnate de aceste servicii sunt corecte și într-un format așteptat.
* Utilizatorul are o conexiune la internet stabilă.
* Biblioteca GMap.NET funcționează corect pe sistemul utilizatorului.
* Limitările serviciilor gratuite OSRM/Nominatim nu sunt depășite în timpul utilizării normale.

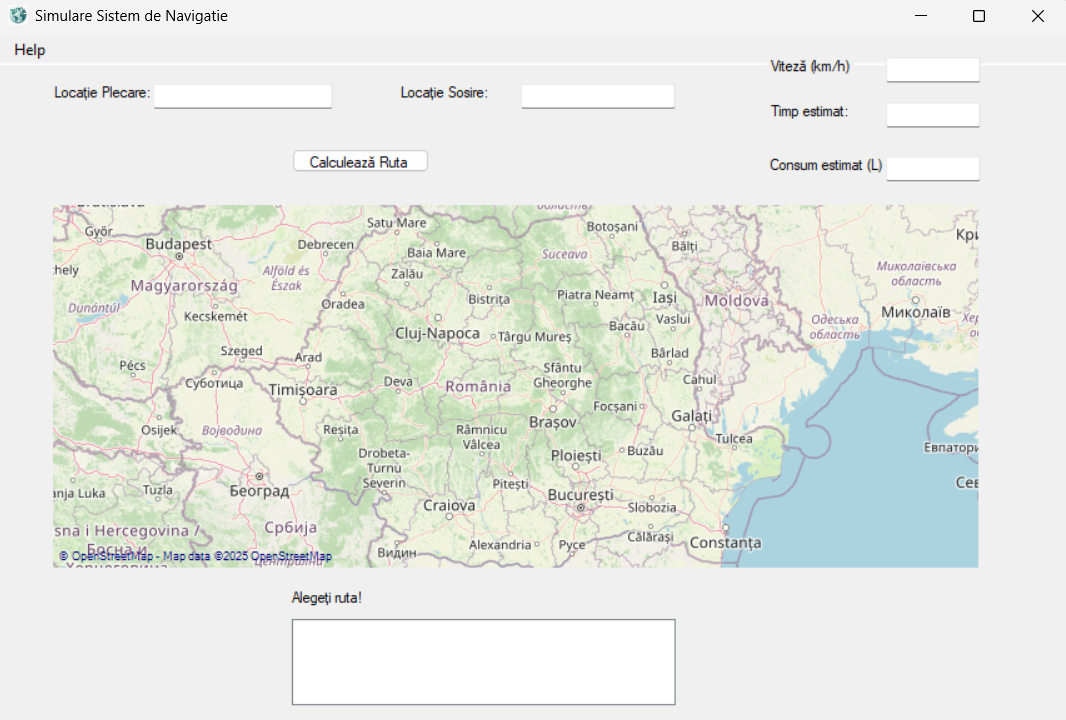
1. **Cerințe pentru Interfețe Externe**

**3.1. Interfețe Utilizator**

Interfața grafică principală (`Form1`) permite utilizatorului să specifice locațiile de plecare și sosire, să vizualizeze rute pe o hartă interactivă și să obțină estimări de călătorie.

3.1.1 Prezentarea Generală a Interfeței

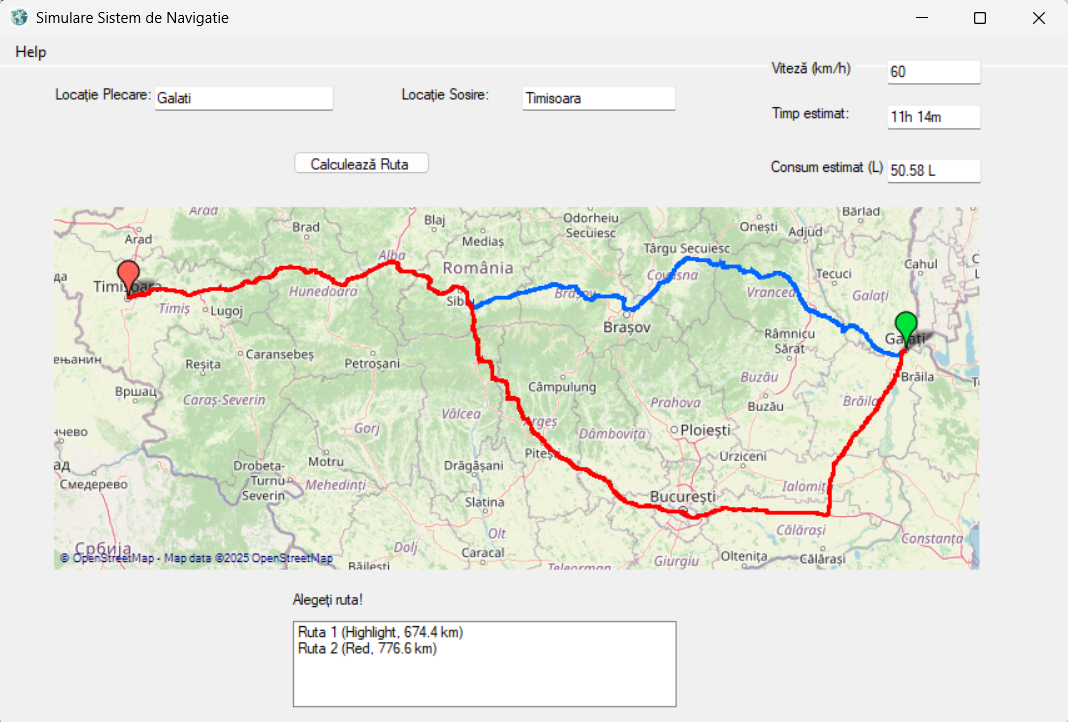
* Interfața inițială oferă câmpuri pentru introducerea datelor de rutare și controlul hărții.



* Componentele principale sunt:
  + - Câmpuri text pentru "Locație Plecare", "Locație Sosire" și "Viteză (km/h)".
    - Butonul "Calculează Ruta".
    - Controlul de hartă `gmapControl`.
    - Lista `listBox1` pentru rutele alternative.
    - Câmpuri text (doar citire) pentru "Timp estimat" și "Consum estimat". \
    - Meniu "Help".

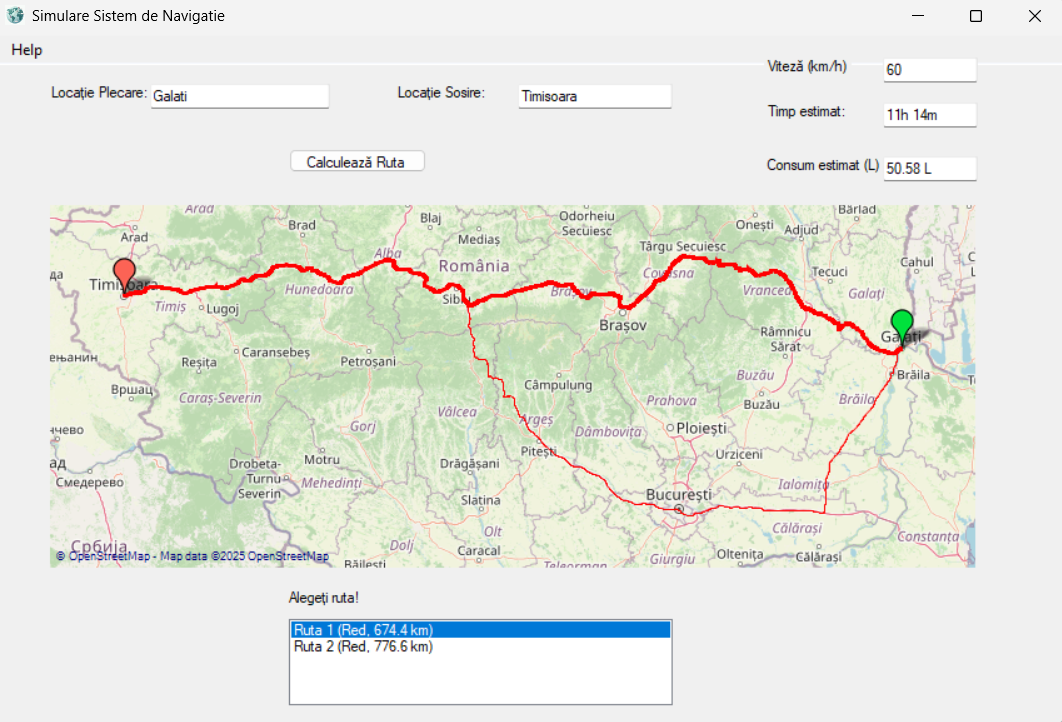
3.1.2 Afișarea Rutelor și a Metricilor

* După calcul, rutele sunt afișate pe hartă și în listă, iar metricile pentru ruta implicită/selectată sunt actualizate.



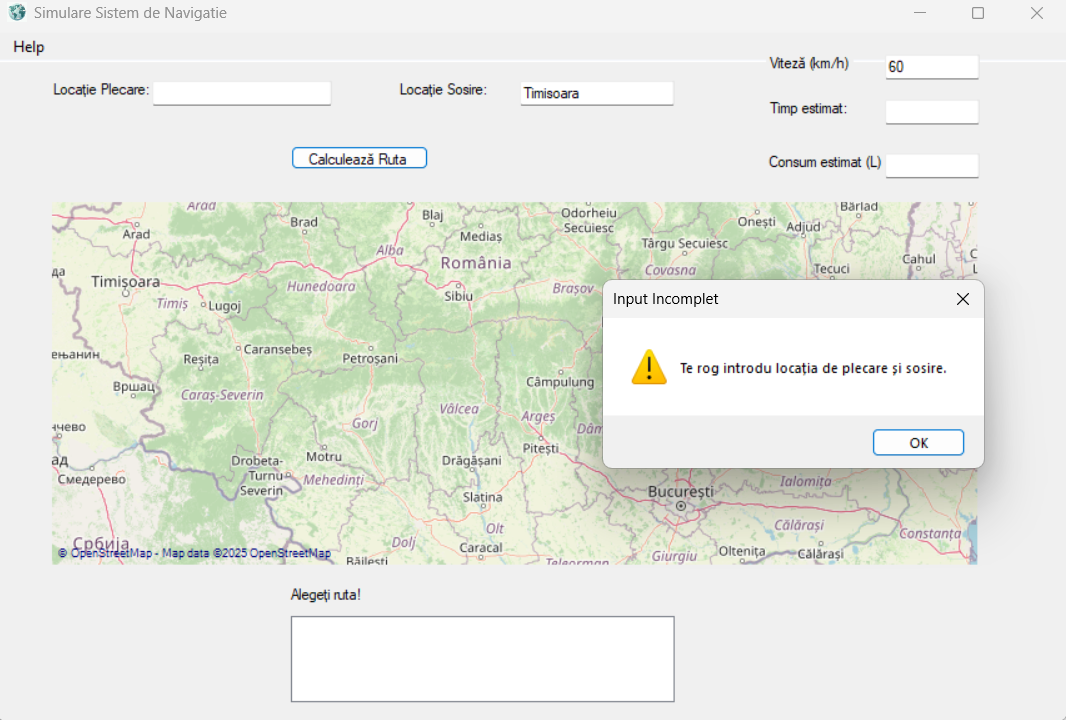
3.1.3 Selecția unei Rute Alternative

* Utilizatorul poate selecta o rută din listă, aceasta fiind evidențiată pe hartă, iar metricile corespunzătoare sunt actualizate.



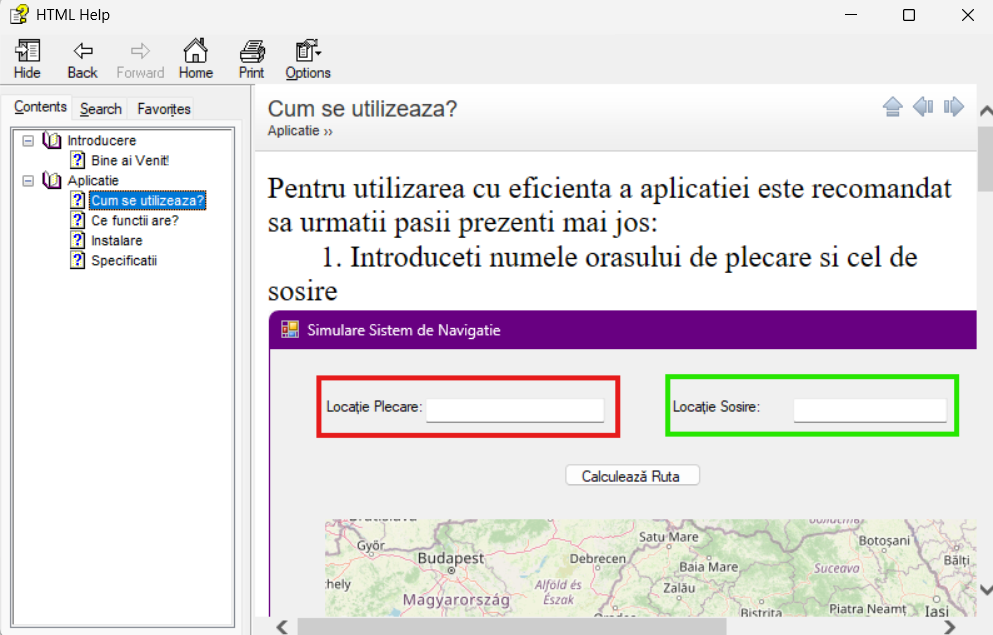
3.1.4 Notificări și Mesaje către Utilizator

* Aplicația utilizează ferestre de dialog (`MessageBox`) pentru a informa utilizatorul despre erori, avertismente sau stări ale sistemului



3.1.5 Accesul la Sistemul de Ajutor

* Meniul "Help" permite deschiderea fișierului de ajutor CHM.



**3.2. Interfețe Hardware**

* Tastatură standard pentru introducerea textului.
* Mouse standard pentru interacțiunea cu butoanele, harta și lista.
* Monitor standard pentru afișarea interfeței.

3.3. **Interfețe Software**

* API OSRM:
* Protocol: HTTP GET.
* URL: `http://router.project-osrm.org/route/v1/driving/{startLng},{startLat};{endLng},{endLat}?overview=full&alternatives=true&geometries=polyline`
* Format date: JSON.  
   Date trimise: Coordonate start (Lat, Lng), Coordonate end (Lat, Lng).
* Date primite: Geometria rutei (polyline), distanța, durata. \* Gestionat de: `RoutingService.cs` (`RoutingServiceDLL.dll`).
* API Nominatim (OpenStreetMap):
* Protocol: HTTP GET. \* URL: `https://nominatim.openstreetmap.org/search?q={address}&format=json&limit=1`
* Format date: JSON.
* Date trimise: Adresa textuală.
* Date primite: Coordonate (Lat, Lng).
* Gestionat de: `Form1.cs` (metoda `GetCoordinatesForAddressAsync`).
* Biblioteca GMap.NET:\*\* Utilizată pentru afișarea hărților, markerelor și rutelor.
* Biblioteca Newtonsoft.Json:\*\* Utilizată pentru parsarea răspunsurilor JSON de la API-uri (dacă nu se folosește `System.Text.Json` direct).
* .NET Framework:\*\* (specifică versiunea).

3.4. **Interfețe de Comunicații**

* Protocol TCP/IP pentru conexiunea la internet.
* Protocoale HTTP/HTTPS pentru interacțiunea cu API-urile OSRM și Nominatim.

1. **Caracteristicile Sistemului**

4.1. Introducerea și Geocodarea Locațiilor

* + - REQ-FUNC-1.1: Sistemul va permite utilizatorului să introducă o locație de plecare sub formă de text în câmpul `textBox2`.
    - REQ-FUNC-1.2: Sistemul va permite utilizatorului să introducă o locație de sosire sub formă de text în câmpul `textBox1`.
    - REQ-FUNC-1.3: La apăsarea butonului "Calculează Ruta", sistemul va trimite adresa de plecare către serviciul Nominatim pentru a obține coordonatele geografice.
    - REQ-FUNC-1.4: La apăsarea butonului "Calculează Ruta", sistemul va trimite adresa de sosire către serviciul Nominatim pentru a obține coordonatele geografice.
    - REQ-FUNC-1.5: Dacă serviciul Nominatim nu poate geocoda o adresă, sistemul va afișa un mesaj de eroare corespunzător utilizatorului.

4.2. Calcularea și Afișarea Rutelor

* + REQ-FUNC-2.1: După obținerea coordonatelor, sistemul va interoga serviciul OSRM pentru a obține una sau mai multe rute între punctul de plecare și cel de sosire.
  + REQ-FUNC-2.2: Sistemul va afișa pe `gmapControl` un marker pentru locația de plecare
  + REQ-FUNC-2.3: Sistemul va afișa pe `gmapControl` un marker pentru locația de sosire
  + REQ-FUNC-2.4: Sistemul va desena pe `gmapControl` toate rutele alternative returnate de OSRM, folosind culori distincte (utilizând `ColorNameDLL.dll` pentru a genera nume de culori descriptive).
  + REQ-FUNC-2.5: Sistemul va popula `listBox1` cu descrieri ale rutelor alternative
  + REQ-FUNC-2.6: Dacă OSRM nu returnează nicio rută, sistemul va afișa un mesaj corespunzător.

4.3. Selecția și Evidențierea Rutei

* REQ-FUNC-3.1: Utilizatorul va putea selecta o rută din `listBox1`.
* REQ-FUNC-3.2: La selecția unei rute, sistemul va evidenția ruta corespunzătoare pe `gmapControl`. Celelalte rute vor fi afișate cu o vizibilitate redusă sau o culoare neutră.
* REQ-FUNC-3.3: La selecția unei rute, denumirea din `listBox1` pentru ruta selectată poate fi actualizată pentru a reflecta starea de "selectat" .

4.4. Calcularea și Afișarea Metricilor de Călătorie

* REQ-FUNC-4.1: Utilizatorul va putea introduce o viteză medie de deplasare (km/h) în `textBox3`.
* REQ-FUNC-4.2: Dacă utilizatorul nu introduce o viteză, sistemul va utiliza o valoare implicită .
* REQ-FUNC-4.3: La selecția unei rute (sau pentru prima rută, dacă nu s-a făcut o selecție), sistemul va calcula timpul estimat de parcurs folosind distanța rutei și viteza medie. Rezultatul va fi afișat în `textBox4`.
* REQ-FUNC-4.4: La selecția unei rute, sistemul va calcula consumul estimat de combustibil folosind distanța rutei și o rată de consum implicită (ex: 7.5 L/100km). Rezultatul va fi afișat în `textBox5`.
* REQ-FUNC-4.5: Dacă viteza introdusă nu este validă (ex: non-numerică, negativă), sistemul va afișa un mesaj de eroare și nu va calcula metricile.

4.5. Managementul Stării Interfeței

* REQ-FUNC-5.1: Sistemul va intra într-o stare de "Încărcare" după ce utilizatorul apasă "Calculează Ruta", timp în care controalele de input și butonul vor fi dezactivate, iar cursorul se va schimba în "WaitCursor".
* REQ-FUNC-5.2: După finalizarea încărcării (cu succes sau eroare), sistemul va reveni la o stare corespunzătoare (`RoutesDisplayedState` sau `InitialState`), reactivând controalele.  
   REQ-FUNC-5.3: La selectarea unei rute, sistemul va trece în starea `RouteSelectedState`.

4.6. Sistemul de Ajutor (Help)

* REQ-FUNC-6.1: Utilizatorul va putea accesa un fișier de ajutor (CHM) prin meniul "Help" .

1. **Alte Cerințe Non-funcționale**

5.1. Cerințe de Performanță

* + REQ-NONFUNC-1.1: Geocodarea unei adrese prin Nominatim ar trebui să se finalizeze în mai puțin de 3 secunde, cu o conexiune la internet standard.
  + REQ-NONFUNC-1.2: Calcularea rutelor prin OSRM ar trebui să se finalizeze în mai puțin de 5 secunde pentru distanțe rezonabile, cu o conexiune la internet standard.
  + REQ-NONFUNC-1.3: Interfața grafică ar trebui să rămână responsivă în timpul operațiunilor de lungă durată

5.2. Cerințe de Siguranță

* N/A (nu este o aplicație critică din punct de vedere al siguranței).

5.3. Cerințe de Securitate

* N/A (aplicația nu gestionează date sensibile ale utilizatorului; comunică cu API-uri publice).

5.4. Atribute de Calitate Software

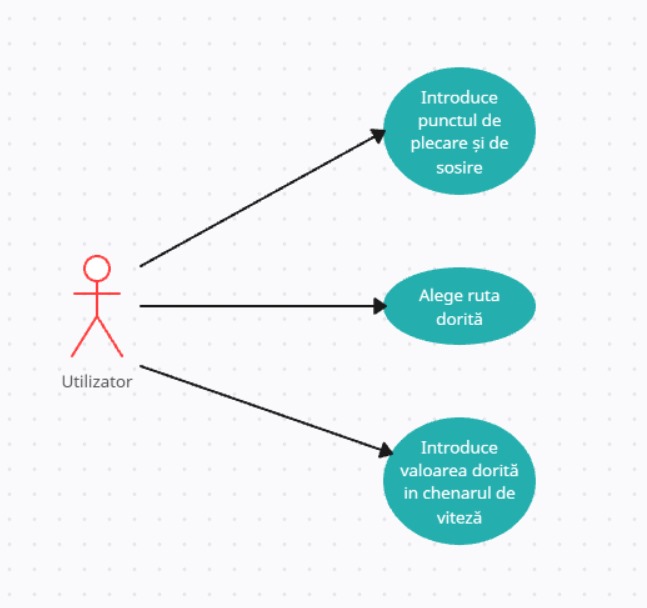
* Utilizabilitate: Interfața va fi simplă și intuitivă. Mesajele de eroare vor fi clare.
* Fiabilitate: Aplicația va gestiona erorile provenite de la serviciile externe (ex: lipsă conexiune, API indisponibil) și va informa utilizatorul corespunzător. Excepțiile interne vor fi prinse și gestionate (conform cerințelor proiectului).
* Mentenabilitate: Codul va fi comentat. Utilizarea State Pattern și separarea în DLL-uri contribuie la o mai bună structurare și mentenabilitate.
* Portabilitate:Limitată la sistemele Windows ce suportă .NET Framework specificat.

5.5. Reguli de Business

* + Rata de consum implicită a combustibilului este setată la 7.5 L/100km.
  + Viteza medie implicită este setată la 60 km/h.

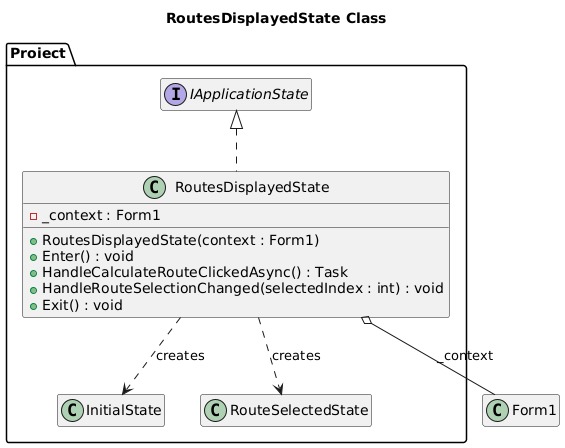
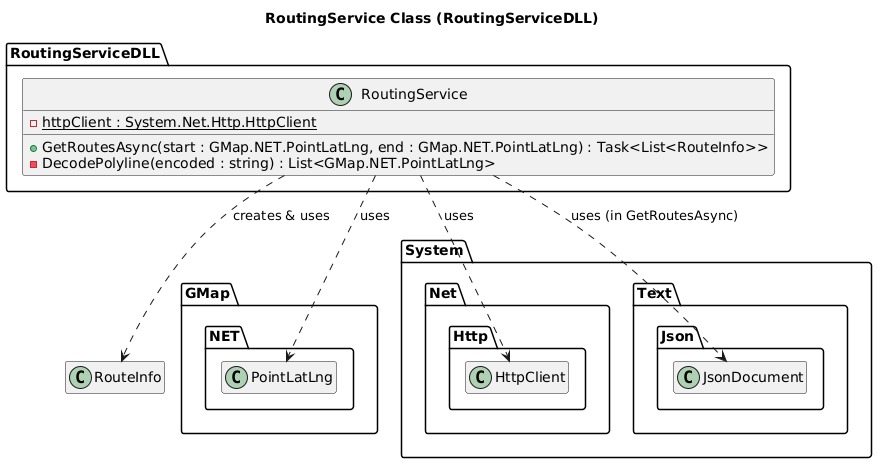
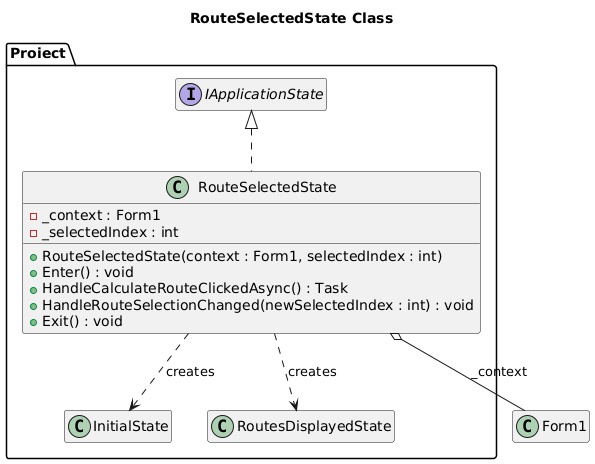
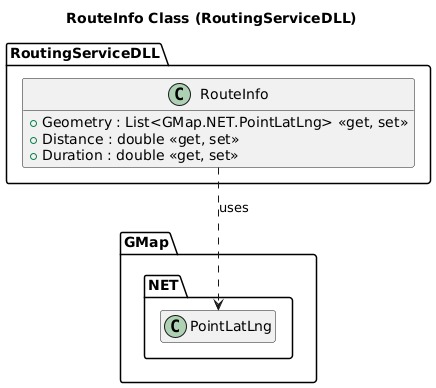
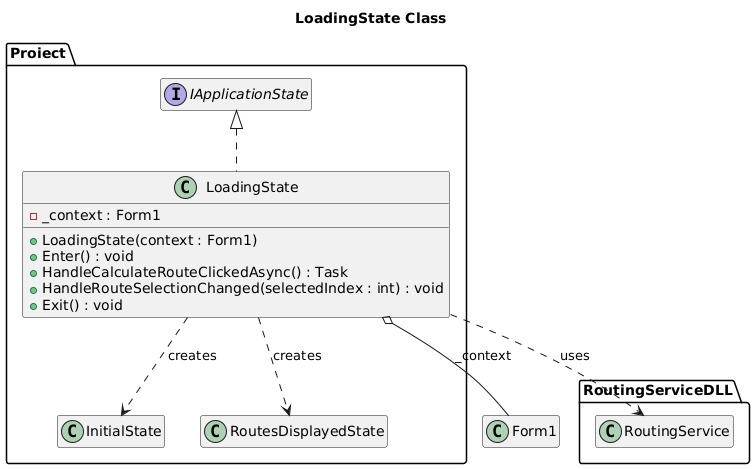
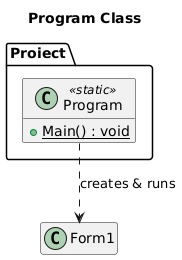
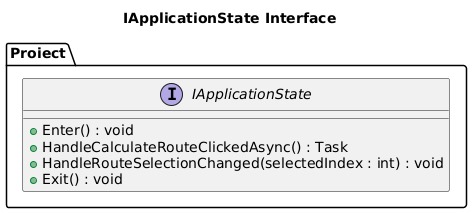
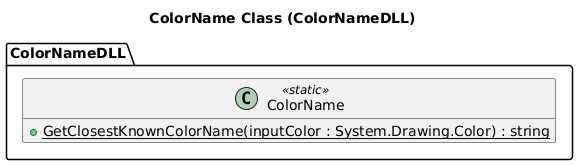
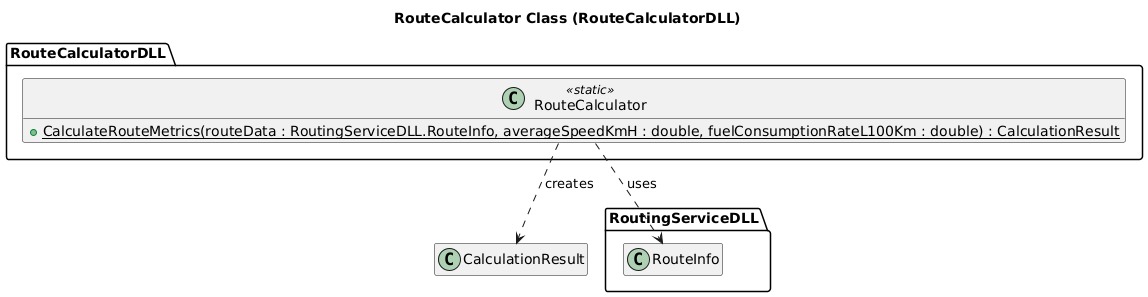
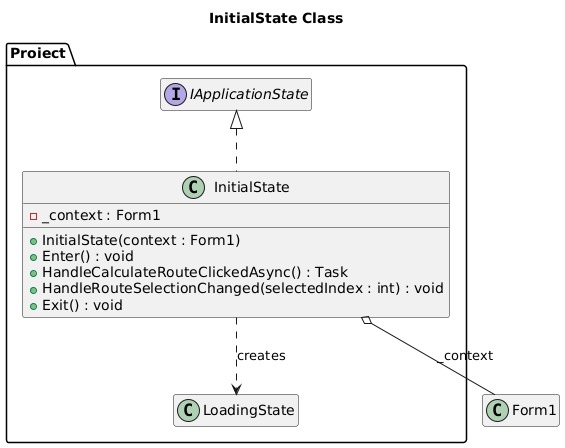
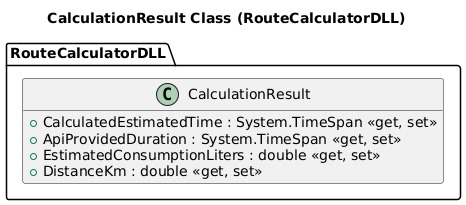
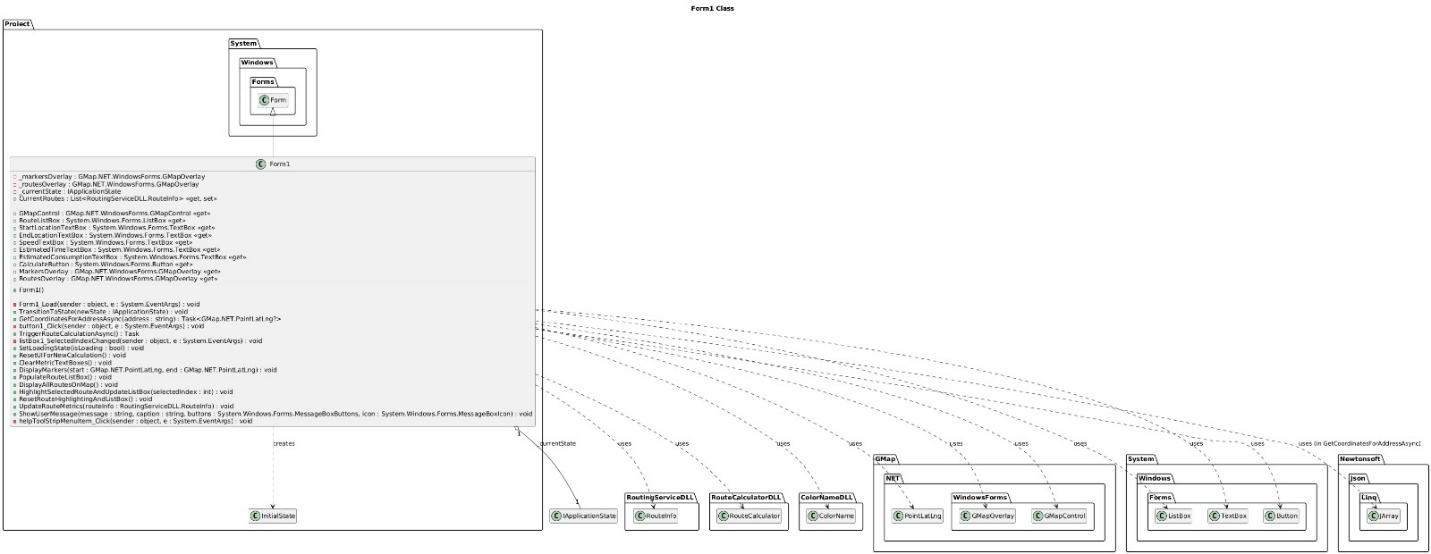
1. **Alte Cerințe**

Anexa A:  
  
 **Diagrama Use Case**

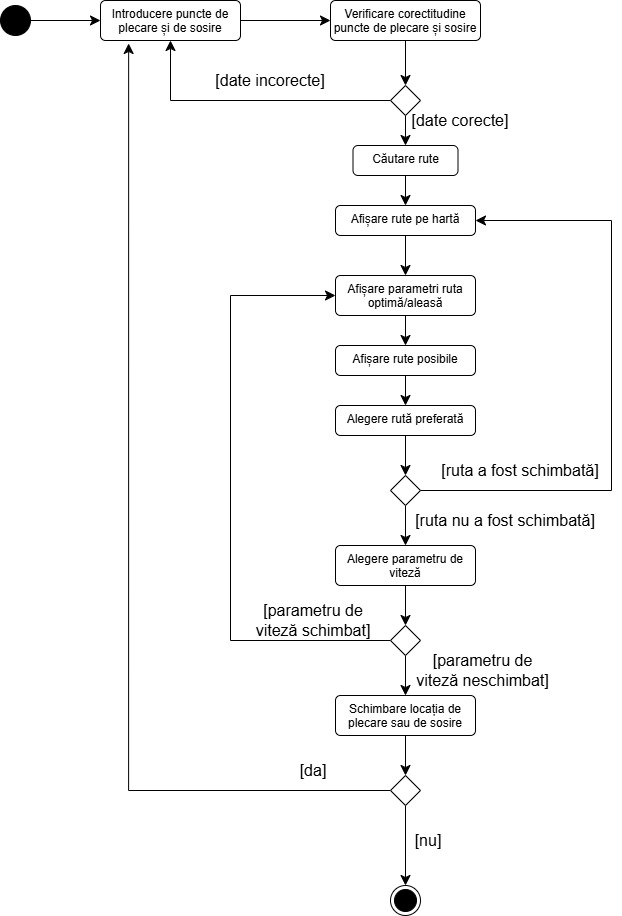


**Diagrama de clase**

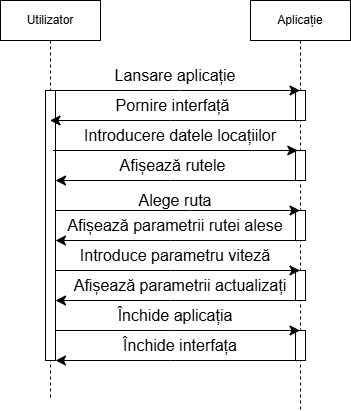
****

****

**Diagrama de secvențe pentru utilizator**

****

**Diagrama de secvențe pentru administrator**

****

Anexa B:

// Fragment din Form1.cs - Metoda Form1\_Load

// Inițializează controlul GMap.NET și setează starea inițială a aplicației.

private void Form1\_Load(object sender, EventArgs e)

{

try

{

gmapControl.MapProvider = GMapProviders.OpenStreetMap; // Setează providerul de hărți

gmapControl.MinZoom = 1;

gmapControl.MaxZoom = 18;

gmapControl.Zoom = 6; // Zoom inițial

gmapControl.Position = new PointLatLng(45.9432, 24.9668); // Centrează pe România

gmapControl.DragButton = MouseButtons.Left; // Permite drag cu butonul stâng

\_markersOverlay = new GMapOverlay("markers"); // Overlay pentru markeri

gmapControl.Overlays.Add(\_markersOverlay);

\_routesOverlay = new GMapOverlay("routes"); // Overlay pentru rute

gmapControl.Overlays.Add(\_routesOverlay);

}

catch (Exception ex)

{

ShowUserMessage($"Eroare la inițializarea hărții: {ex.Message}", "Eroare GMap.NET", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

TransitionToState(new InitialState(this)); // Aplică State Pattern - pornește în starea inițială

}

// Fragment din Form1.cs - Metoda button1\_Click

// Deleagă logica de calculare a rutei către starea curentă a aplicației.

private async void button1\_Click(object sender, EventArgs e)

{

if (\_currentState != null) // Verifică dacă există o stare curentă

{

// Apelează metoda specifică stării pentru a gestiona evenimentul

await \_currentState.HandleCalculateRouteClickedAsync();

}

}

// Fragment din Form1.cs - Metoda GetCoordinatesForAddressAsync

// Interoghează API-ul Nominatim pentru a obține coordonatele unei adrese textuale.

public async Task<PointLatLng?> GetCoordinatesForAddressAsync(string address)

{

if (string.IsNullOrWhiteSpace(address)) return null;

string url = $"https://nominatim.openstreetmap.org/search?q={Uri.EscapeDataString(address)}&format=json&limit=1";

using (var httpClient = new HttpClient())

{

httpClient.DefaultRequestHeaders.UserAgent.ParseAdd("SimulareNavigatieApp/1.0"); // User-Agent necesar

try

{

var response = await httpClient.GetAsync(url);

if (!response.IsSuccessStatusCode) return null;

var json = await response.Content.ReadAsStringAsync();

var results = JArray.Parse(json); // Parsează răspunsul JSON

if (results.Count == 0 || results[0]["lat"] == null || results[0]["lon"] == null) return null;

// Extrage și parsează latitudinea și longitudinea

double.TryParse((string)results[0]["lat"], System.Globalization.NumberStyles.Any, System.Globalization.CultureInfo.InvariantCulture, out double lat);

double.TryParse((string)results[0]["lon"], System.Globalization.NumberStyles.Any, System.Globalization.CultureInfo.InvariantCulture, out double lon);

return new PointLatLng(lat, lon);

}

catch { return null; /\* Gestionare simplificată a excepțiilor pentru concizie \*/ }

}

}

// Fragment din ApplicationState.cs - Clasa LoadingState, metoda Enter

// Logica executată la intrarea în starea de încărcare: obține coordonate și rute.

public class LoadingState : IApplicationState

{

private readonly Form1 \_context;

public LoadingState(Form1 context) { \_context = context; }

public async void Enter()

{

\_context.SetLoadingState(true); // Actualizează UI pentru starea de încărcare

var coordPlecare = await \_context.GetCoordinatesForAddressAsync(\_context.StartLocationTextBox.Text);

var coordSosire = await \_context.GetCoordinatesForAddressAsync(\_context.EndLocationTextBox.Text);

if (coordPlecare == null || coordSosire == null) { /\* ... gestionare eroare ... \*/ \_context.TransitionToState(new InitialState(\_context)); return; }

\_context.DisplayMarkers(coordPlecare.Value, coordSosire.Value); // Afișează markerii

var routingService = new RoutingService(); // Instanțiază serviciul de rutare (din DLL)

try

{

// Obține rutele de la API-ul OSRM

\_context.CurrentRoutes = await routingService.GetRoutesAsync(coordPlecare.Value, coordSosire.Value);

if (\_context.CurrentRoutes == null || \_context.CurrentRoutes.Count == 0) { /\* ... gestionare eroare ... \*/ }

\_context.TransitionToState(new RoutesDisplayedState(\_context)); // Tranziționează la afișarea rutelor

}

catch (Exception ex) { /\* ... gestionare eroare ... \*/ \_context.TransitionToState(new InitialState(\_context)); }

}

// ... restul metodelor interfeței ...

}

// Fragment din RoutingService.cs - Metoda GetRoutesAsync

// Obține rute de la API-ul OSRM și parsează răspunsul JSON.

public async Task<List<RouteInfo>> GetRoutesAsync(PointLatLng start, PointLatLng end)

{

// Formatează coordonatele pentru URL folosind CultureInfo.InvariantCulture pentru a evita probleme cu separatorul zecimal

var startLng = start.Lng.ToString(System.Globalization.CultureInfo.InvariantCulture);

var startLat = start.Lat.ToString(System.Globalization.CultureInfo.InvariantCulture);

var endLng = end.Lng.ToString(System.Globalization.CultureInfo.InvariantCulture);

var endLat = end.Lat.ToString(System.Globalization.CultureInfo.InvariantCulture);

var url = $"http://router.project-osrm.org/route/v1/driving/{startLng},{startLat};{endLng},{endLat}?overview=full&alternatives=true&geometries=polyline";

var response = await httpClient.GetAsync(url);

if (!response.IsSuccessStatusCode) { throw new Exception($"Eroare OSRM: {response.StatusCode}"); }

var json = await response.Content.ReadAsStringAsync();

var doc = JsonDocument.Parse(json);

var root = doc.RootElement;

if (root.GetProperty("code").GetString() != "Ok") { throw new Exception($"OSRM: {root.GetProperty("message").GetString()}"); }

var routes = new List<RouteInfo>();

foreach (var routeElement in root.GetProperty("routes").EnumerateArray())

{

routes.Add(new RouteInfo {

Distance = routeElement.GetProperty("distance").GetDouble(),

Duration = routeElement.GetProperty("duration").GetDouble(),

Geometry = DecodePolyline(routeElement.GetProperty("geometry").GetString()) // Apel metodă decodare

});

}

return routes;

}