<Aplicatie interactiva pentru reprezentarea fenomenelor fizice cu scop experimental si educativ>

# Documentul de proiectare

Cuprins

[1. Introducere 1](#_Toc164112175)

[1.1 Scopul documentului 1](#_Toc164112176)

[2. Prezentare generală și abordări de proiectare 2](#_Toc164112177)

[2.1 Prezentare generală 2](#_Toc164112178)

[2.2 Presupuneri/ Constrângeri/ Riscuri 2](#_Toc164112179)

[2.2.1 Presupuneri 2](#_Toc164112180)

[2.2.2 Constrângeri 3](#_Toc164112181)

[2.2.3 Riscuri 3](#_Toc164112182)

[3. Considerațiii de proiectare 4](#_Toc164112183)

[3.1 Obiective și linii directoare (ghiduri) 4](#_Toc164112184)

[3.2 Metode de dezvoltare 4](#_Toc164112185)

[3.3 Strategii de arhitectură 4](#_Toc164112186)

[4. Arhitectura Sistemului și Proiectarea Arhitecturii 6](#_Toc164112187)

[4.1 Arhitectură software 7](#_Toc164112188)

[4.2 Arhitectura informațiilor 7](#_Toc164112189)

[5. Proiectarea sistemului 9](#_Toc164112190)

[5.1 Interfețe utilizator 9](#_Toc164112191)

[5.1.1 Intrări 10](#_Toc164112192)

[5.1.2 Ieșiri 10](#_Toc164112193)

[5.2 Proiectarea interfețelor cu utilizatorul 11](#_Toc164112194)

[6. Scenarii de utilizare 13](#_Toc164112195)

[7. Proiectare de detaliu 14](#_Toc164112196)

[7.1 Proiectare software de detaliu 14](#_Toc164112197)

[Anexa A: Gestiunea modificărilor documentului 15](#_Toc164112198)

[Anexa B: Acronime 16](#_Toc164112199)

[Anexa C Documente la care se face referire 17](#_Toc164112200)

## Introducere

Documentul de Proiectare a Sistemului descrie modul în care cerințele funcționale și non-funcționale înregistrate în Documentul de Cerințe se transformă în specificații de proiectare a sistemului, mai tehnice, pe baza cărora se construiește sistemul. Se documentează proiectarea sistemului la un nivel înalt și specificațiile detaliate de proiectare.

Se descriu obiectivele și considerațiile de proiectare, se furnizează o prezentare generală la nivel înalt a arhitecturii sistemului și se descrie proiectarea datelor asociate cu sistemul, precum și interfața om-mașină și scenariile operaționale.

Proiectarea sistemului la nivel înalt este descompusă în continuare în specificații detaliate de proiectare pentru fiecare componentă a sistemului, inclusiv hardware, comunicațiile interne, software, controalele de integritate a sistemului și interfețele externe.

### Scopul documentului

Scopul SDD-ului (Software Design Document) este de a oferi o descriere detaliată a arhitecturii și funcționalităților sistemului de software în cadrul proiectului nostru de dezvoltare a unei aplicații de simulare a fenomenelor fizice. Acest document are rolul de a servi ca o hartă de navigare pentru întreaga echipă de dezvoltare, precum și pentru orice alte părți interesate, pentru a înțelege în mod clar cum va fi construită și implementată aplicația.

SDD-ul va cuprinde informații esențiale despre structura aplicației, inclusiv arhitectura software, modulele și componentele individuale, interacțiunile dintre acestea, precum și detaliile implementării tehnice. De asemenea, va furniza o descriere comprehensivă a funcționalităților sistemului, inclusiv scenariile de utilizare, interfața utilizatorului, modul de gestionare a datelor și orice alte aspecte relevante pentru proiectul nostru.

Scopul final al SDD-ului este de a asigura că întreaga echipă de dezvoltare are o înțelegere comună și clară a obiectivelor, cerințelor și așteptărilor pentru proiect, și că acestea sunt reflectate corespunzător în design-ul și implementarea aplicației.

## Prezentare generală și abordări de proiectare

Această secțiune descrie principiile și strategiile care vor fi utilizate ca ghiduri în momentul proiectării și implementării sistemului.

### Prezentare generală

Contextul sistemului nostru este dezvoltarea unei aplicații de simulare a fenomenelor fizice destinate copiilor care doresc să învețe și să exploreze conceptele de fizică într-un mod interactiv și captivant. Această aplicație va oferi utilizatorilor posibilitatea de a interacționa cu diferite simulări și experimente fizice într-o modalitate grafică și ușor de înțeles.

Abordarea noastră pentru proiectare constă în utilizarea limbajului de programare C++ împreună cu OpenGL pentru grafica 2D, ImGui pentru crearea interfeței utilizatorului și GLFW pentru gestionarea ferestrei și a input-ului de la tastatură și mouse. Această combinație de tehnologii ne va permite să dezvoltăm o aplicație robustă și interactivă, cu performanțe bune și o experiență de utilizare plăcută.

Arhitectura sistemului va fi structurată în jurul unui model modular, cu componente clare și bine definite. Vom avea module distincte pentru gestionarea simulărilor, interacțiunea cu utilizatorul, gestionarea parametrilor fizici și afișarea datelor grafice. Aceste module vor comunica între ele prin intermediul unor interfețe bine definite, facilitând extensibilitatea și întreținerea aplicației pe termen lung.

Obiectivele noastre de proiectare includ:

1. Asigurarea unei interfețe utilizator prietenoase și intuitive, adaptată nevoilor și capacităților copiilor.
2. Crearea unui sistem flexibil care să permită utilizatorilor să exploreze și să experimenteze diferite fenomene fizice într-un mod personalizat.
3. Optimizarea performanțelor pentru a asigura o experiență fluidă și fără întreruperi în timpul simulărilor.
4. Integrarea unui sistem robust de gestionare a datelor și parametrilor, pentru a facilita modificările și actualizările ulterioare ale aplicației.

### Presupuneri/ Constrângeri/ Riscuri

#### Presupuneri

1. **Sistem de operare**: Vom presupune că aplicația va fi dezvoltată pentru platforma Windows, deoarece vom utiliza tehnologii specifice acestui sistem de operare, cum ar fi GLFW pentru gestionarea ferestrei și a input-ului de la tastatură și mouse.
2. **Cerințe hardware**: Vom presupune că aplicația noastră va fi utilizată pe computere cu resurse hardware suficiente pentru a susține randarea graficii 2D și simulările fizice în timp real. Aceasta include o placă grafică compatibilă cu OpenGL și suficientă memorie RAM pentru a gestiona operațiunile aplicației fără probleme de performanță.
3. **Caracteristicile utilizatorilor finali**: Vom presupune că utilizatorii finali ai aplicației vor fi în principal copii sau adolescenți, interesați de învățarea și explorarea conceptelor de fizică. Prin urmare, interfața utilizatorului va trebui să fie prietenoasă și intuitivă, adaptată nivelului lor de înțelegere și experiență în utilizarea calculatorului.
4. **Posibile modificări ale funcționalității**: Vom lua în considerare posibilitatea ca funcționalitățile aplicației să fie extinse sau modificate în viitor pentru a răspunde nevoilor și feedback-ului utilizatorilor. Astfel, vom proiecta sistemul într-un mod modular și flexibil, astfel încât să permită adăugarea și modificarea funcționalităților fără a afecta întregul sistem în mod negativ.

#### Constrângeri

1. **Cerințe de performanță**: Limitările de performanță pot afecta capacitatea sistemului de a rula simulările fizice în timp real și de a gestiona afișarea grafică corespunzătoare. Astfel, trebuie să fim atenți la optimizarea codului și utilizarea eficientă a resurselor hardware pentru a asigura o experiență fluidă și fără întreruperi pentru utilizatori.
2. **Cerințe de memorie și resurse**: Capacitatea limitată de memorie și resurse hardware poate reprezenta o constrângere pentru dezvoltarea și implementarea aplicației. Trebuie să fim conștienți de aceste limitări și să proiectăm sistemul astfel încât să utilizeze eficient resursele disponibile, evitând supraîncărcarea și optimizând gestionarea memoriei.
3. **Securitate și reglementări**: Cerințele de securitate și reglementările privind protecția datelor pot afecta designul și implementarea aplicației, necesitând implementarea măsurilor adecvate pentru protejarea informațiilor sensibile și respectarea standardelor de securitate relevante.
4. **Interoperabilitate și standarde**: Este posibil să existe cerințe de interoperabilitate și conformitate cu standardele în industrie care trebuie respectate în proiectul nostru. Astfel, trebuie să ne asigurăm că aplicația noastră poate comunica eficient cu alte sisteme și să respecte standardele relevante pentru integrarea și interoperabilitatea cu alte produse sau platforme.
5. **Testare și validare**: Cerințele de verificare și validare pot afecta planificarea și implementarea procesului de testare a aplicației noastre. Trebuie să ne asigurăm că avem proceduri adecvate de testare și validare pentru a identifica și remedia problemele potențiale înainte de lansarea aplicației pe piață.

#### Riscuri

1. **Riscul de performanță insuficientă**: Există riscul ca aplicația să nu funcționeze suficient de rapid sau eficient pentru a oferi o experiență plăcută utilizatorilor. Pentru a reduce acest risc, vom efectua teste de performanță și optimizări continue ale codului și algoritmului pentru a asigura că aplicația rulează fluent chiar și pe hardware mai modest.
2. **Riscul de defecte de software**: Există riscul ca aplicația să conțină defecte de software care să afecteze funcționarea sau să producă rezultate incorecte în simulări. Pentru a reduce acest risc, vom implementa procese riguroase de testare și depanare, inclusiv testare automată și manuală, pentru a identifica și remedia problemele înainte de lansarea aplicației.

## Considerațiii de proiectare

### Obiective și linii directoare (ghiduri)

1. **Performanță și eficiență**: Accentul principal va fi pus pe optimizarea performanței și eficienței aplicației, pentru a asigura o experiență fluidă și fără întreruperi pentru utilizatori. Acest lucru implică minimizarea utilizării resurselor hardware și software, precum și optimizarea algoritmilor și a codului pentru a asigura o funcționare rapidă și eficientă a aplicației.
2. **Simplicitate și ușurință în utilizare**: Interfața utilizatorului va fi proiectată pentru a fi intuitivă și ușor de înțeles, cu accent pe simplificarea interacțiunii utilizatorului cu aplicația. Aceasta implică utilizarea unui design minimalist și clar, precum și furnizarea de instrucțiuni și indicații clare pentru utilizatori.
3. **Flexibilitate și personalizare**: Aplicația va oferi utilizatorilor posibilitatea de a personaliza și ajusta diferite aspecte ale simulărilor fizice, pentru a le permite să exploreze și să experimenteze în funcție de interesele și preferințele lor individuale. Acest lucru implică implementarea unui sistem modular și a unor opțiuni de configurare flexibile pentru parametrii simulării.
4. **Fiabilitate și stabilitate**: Prioritatea va fi acordată asigurării fiabilității și stabilității aplicației, pentru a evita defectele de software și pentru a asigura o experiență coerentă și fără probleme pentru utilizatori. Acest lucru implică efectuarea de teste riguroase și depanare a aplicației, precum și implementarea unor mecanisme robuste de gestionare a erorilor și a excepțiilor.

### Metode de dezvoltare

Pentru designul sistemului și al software-ului în acest proiect de dezvoltare a aplicației de simulare a fenomenelor fizice, am adoptat o abordare orientată pe obiecte, întrucât aceasta oferă o modalitate naturală și eficientă de a organiza și de a gestiona complexitatea aplicației noastre.

Abordarea orientată pe obiecte ne permite să împărțim sistemul în componente modulare și reutilizabile, fiecare cu responsabilități bine definite și interfețe clare de comunicare. Acest lucru ne ajută să gestionăm complexitatea și să îmbunătățim flexibilitatea și extensibilitatea sistemului, permițându-ne să adăugăm și să modificăm funcționalități fără a afecta întregul sistem.

În ceea ce privește metodele formale sau publicate, nu am adoptat în mod specific o anumită metodă, ci am integrat principii și practici din mai multe metodologii, inclusiv UML (Unified Modeling Language) pentru modelarea și proiectarea arhitecturii și a interacțiunilor sistemului, precum și practici Agile pentru dezvoltarea iterativă și gestionarea cerințelor în continuă schimbare.

### Strategii de arhitectură

1. **Utilizarea limbajului de programare C++ și a OpenGL pentru grafică 2D**: Am ales C++ datorită performanței și flexibilității sale, precum și a capacității de a gestiona eficient resursele hardware. OpenGL a fost selectat pentru grafica 2D datorită suportului său bogat și a performanței sale bune în randarea graficii.
2. **Utilizarea abordării orientate pe obiecte pentru arhitectură**: Această abordare ne permite să organizăm sistemul în jurul unor componente modulare și reutilizabile, cu responsabilități bine definite și interfețe clare de comunicare. Acest lucru îmbunătățește flexibilitatea și extensibilitatea sistemului și facilitează gestionarea complexității.
3. **Reutilizarea componentelor software existente**: Am planificat să reutilizăm biblioteci și framework-uri existente pentru implementarea diferitelor părți și caracteristici ale sistemului, pentru a reduce timpul de dezvoltare și pentru a beneficia de funcționalitățile și performanțele deja existente.
4. **Utilizarea unei arhitecturi modulară și ușor extensibilă**: Am planificat să proiectăm sistemul folosind o arhitectură modulară și ușor extensibilă, care să permită adăugarea și modificarea funcționalităților fără a afecta întregul sistem. Acest lucru ne va ajuta să răspundem rapid la schimbările cerințelor și să integrăm noi caracteristici cu ușurință.
5. **Implementarea unor politici stricte de gestionare a memoriei**: Pentru a evita scurgerile de memorie și supraîncărcarea, vom implementa politici stricte de gestionare a memoriei, inclusiv eliberarea corectă a resurselor și utilizarea inteligentă a structurilor de date pentru a minimiza utilizarea memoriei.

## Arhitectura Sistemului și Proiectarea Arhitecturii

Arhitectura sistemului este structurată în jurul a trei componente principale: interfața utilizatorului, motorul de simulare și motorul grafic. Aceste componente lucrează împreună pentru a oferi utilizatorilor o experiență interactivă și captivantă în simularea fenomenelor fizice.

1. **Interfața Utilizatorului (UI)**:
   * Componenta UI este responsabilă pentru interacțiunea directă cu utilizatorii. Ea afișează interfața grafică a aplicației și primește input-ul utilizatorului, precum și comenzi pentru modificarea parametrilor simulării.
   * UI utilizează modele de interfață grafică și componente de interfață pentru a afișa și a gestiona elementele UI, precum și pentru a comunica cu motorul de simulare și motorul grafic pentru a actualiza și a reflecta schimbările în simulare.
2. **Motorul de Simulare**:
   * Motorul de simulare este nucleul aplicației, responsabil pentru calcularea și gestionarea procesului de simulare a fenomenelor fizice. Acesta primește comenzi și parametri de la UI și produce datele de ieșire corespunzătoare pentru motorul grafic.
   * Motorul de simulare utilizează modele și algoritmi fizici pentru a simula diferite fenomene, precum și baze de date sau resurse externe pentru a obține informații suplimentare necesare pentru simulare.
3. **Motorul Grafic**:
   * Motorul grafic este responsabil pentru randarea și afișarea vizuală a simulării fenomenelor fizice. El primește datele de ieșire de la motorul de simulare și le afișează într-o formă vizuală pentru utilizatori.
   * Motorul grafic utilizează tehnologii grafice precum OpenGL pentru randare, precum și biblioteci și instrumente pentru gestionarea obiectelor grafice și interacțiunea cu utilizatorii.

Această decompunere a sistemului în componente/subsistemele sale a fost aleasă pentru a asigura o organizare clară și modulară a funcționalității și responsabilităților. Această arhitectură permite dezvoltarea, testarea și întreținerea fiecărei componente în mod independent, facilitând extinderea și modificarea sistemului în viitor.

În ceea ce privește tiparele de proiectare, vom utiliza probabil tipare precum Model-View-Controller (MVC) pentru structurarea interfeței utilizatorului și separarea logicii de afișare de logica aplicației. De asemenea, vom utiliza tipare de proiectare precum Factory pentru crearea obiectelor în mod dinamic și Observer pentru notificarea componentelor interesate despre schimbările în sistem.

Această arhitectură a fost aleasă pentru a asigura o structură clară și modulară a sistemului nostru, permițându-ne să implementăm funcționalitatea dorită în mod eficient și să gestionăm cu ușurință complexitatea.

### Arhitectură software

Componentele software ale sistemului sunt structurate în conformitate cu arhitectura descrisă anterior. Enumerăm mai jos principalele elemente logice și tehnologice ale aplicației noastre:

1. **Interfața Utilizatorului (UI)**:
   * **Biblioteca de interfață grafică**: Oferă componente grafice și funcționalități pentru construirea și gestionarea interfeței utilizatorului. Exemplu: ImGui.
   * **Motor de randare GUI**: Responsabil pentru randarea și afișarea interfeței utilizatorului. Exemplu: GLFW.
2. **Motorul de Simulare**:
   * **Motor fizic**: Implementează algoritmi și modele fizice pentru simularea fenomenelor. Exemplu: Bibliotecă fizică Box2D.
   * **Manager de simulare**: Gestionează parametrii și starea simulației. Implementează logica de control a simulării.
3. **Motorul Grafic**:
   * **Motor de randare grafică**: Responsabil pentru randarea și afișarea vizuală a simulării. Exemplu: OpenGL pentru randarea 2D.
4. **Limbaje de programare**:
   * **C++**: Limbajul principal de programare pentru implementarea logicii aplicației.
   * **OpenGL Shading Language (GLSL)**: Utilizat pentru programarea shaderelor în motorul grafic.
5. **Utilitare și instrumente**:
   * **CMake**: Utilizat pentru generarea fișierelor de proiect și gestionarea dependențelor.
   * **Git**: Sistem de control al versiunilor pentru gestionarea codului sursă.
6. **Sisteme de operare**:
   * **Windows**: Platforma principală pentru dezvoltare și implementare.
   * **Linux (opțional)**: Posibilitatea de a dezvolta și de a testa pe sisteme Linux.

### Arhitectura informațiilor

În sistemul de simulare a fenomenelor fizice, următoarele tipuri de informații vor fi stocate:

1. **Parametrii de simulare**: Acestea includ valorile parametrilor fizici precum masa, lungimea, accelerația gravitațională, coeficienții de frecare etc., care sunt utilizate pentru a configura și a controla simularea fenomenelor fizice.
2. **Starea simulării**: Informații despre starea actuală a simulației, cum ar fi pozițiile, vitezele și stările obiectelor simulate într-un anumit moment de timp.
3. **Configurări și preferințe ale utilizatorilor**: Acestea includ setările și preferințele selectate de utilizator în cadrul aplicației, cum ar fi valorile parametrilor simulației modificate de utilizator sau preferințele de afișare ale interfeței utilizatorului.
4. **Istoricul sesiunilor de utilizator**: Informații despre activitățile și interacțiunile utilizatorilor cu aplicația, cum ar fi logarea sesiunilor de utilizator, acțiunile întreprinse și opțiunile selectate.
5. **Date de configurare a aplicației**: Informații despre setările și configurările generale ale aplicației, cum ar fi preferințele de limbă, opțiunile de afișare și alte setări de sistem.

Aceste date vor fi stocate în format electronic în fișiere de configurare pe dispozitivul utilizatorului. Nu există informații cu caracter sensibil stocate în sistem, deoarece aplicația noastră nu colectează sau procesează date personale sau confidențiale ale utilizatorilor. Datele sunt furnizate și actualizate de către utilizatorii aplicației prin intermediul interfeței grafice, iar aplicația le prelucrează și le utilizează pentru simulare.

## Proiectarea sistemului

### Interfețe utilizator

În cadrul sistemului nostru de simulare a fenomenelor fizice, identificăm următoarele clase de utilizatori și roluri asociate:

1. **Utilizatorii Finali**:
   * **Descriere**: Utilizatorii finali sunt elevi sau studenți care folosesc aplicația pentru a învăța și a experimenta fenomenele fizice într-un mediu interactiv.
   * **Număr anticipat de utilizatori**: Sute sau chiar mii, în funcție de popularitatea aplicației.
   * **Număr maxim de utilizatori simultani**: Depinde de capacitatea serverului sau de puterea computațională a dispozitivelor utilizate.
   * **Număr de utilizatori externi**: Nu se aplică.
2. **Educatori**:
   * **Descriere**: Educatorii și instruitorii utilizează aplicația pentru a demonstra concepte și fenomene fizice în clase sau în medii educaționale.
   * **Număr anticipat de utilizatori**: Zeci sau sute, în funcție de numărul de educatori care integrează aplicația în curriculumul lor.
   * **Număr maxim de utilizatori simultani**: Depinde de numărul de clase sau de sesiunile de instruire care utilizează aplicația în același timp.
   * **Număr de utilizatori externi**: Nu se aplică.
3. **Dezvoltatori și Administratori**:
   * **Descriere**: Dezvoltatorii și administratorii sunt responsabili pentru dezvoltarea, întreținerea și gestionarea sistemului de simulare a fenomenelor fizice.
   * **Număr anticipat de utilizatori**: Unul sau mai mulți, în funcție de echipa de dezvoltare și de infrastructura IT a organizației.
   * **Număr maxim de utilizatori simultani**: Depinde de necesitățile și cerințele de lucru ale echipei de dezvoltare.
   * **Număr de utilizatori externi**: Nu se aplică.
4. **Personal de Suport Tehnic**:
   * **Descriere**: Personalul de suport tehnic oferă asistență și suport utilizatorilor în cazul problemelor tehnice sau întrebărilor legate de utilizarea aplicației.
   * **Număr anticipat de utilizatori**: Unul sau mai mulți, în funcție de mărimea și complexitatea organizației.
   * **Număr maxim de utilizatori simultani**: Depinde de volumul de solicitări de suport și de capacitatea echipei de suport tehnic.
   * **Număr de utilizatori externi**: Nu se aplică.

Aceste clase de utilizatori acoperă diverse roluri și moduri de interacțiune cu sistemul nostru de simulare a fenomenelor fizice, de la utilizatori finali care utilizează aplicația pentru educație și experimentare până la dezvoltatori și administratori care gestionează infrastructura și funcționarea generală a sistemului.

#### Intrări

În sistemul nostru de simulare a fenomenelor fizice, utilizatorii interacționează cu aplicația folosind următoarele mijloace de intrare:

1. **Interfața Utilizatorului (UI)**:
   * **Mouse-ul și Tastatura**: Utilizate pentru navigarea în interfața grafică și introducerea datelor.
   * **Butoane și Elemente de Interfață Grafică**: Pentru a selecta opțiuni, a modifica parametri și a interacționa cu elementele aplicației.
   * **Câmpuri de Text și Zone de Introducere**: Pentru introducerea manuală a valorilor parametrilor sau a altor date necesare.
2. **Ecrane de Intrare a Datelor**:
   * **Fereastra Principală a Aplicației**: Afișează simularea și oferă opțiuni pentru a modifica parametrii și setările.
   * **Ferestre de Configurare** **și Setări**: Pentru a permite utilizatorilor să configureze preferințele și setările aplicației.
   * **Ferestre de Selectare a Scenariilor de Simulare**: Pentru a permite utilizatorilor să aleagă scenariile și experimentele pe care doresc să le simuleze.

Aspectul și designul ecranelor de intrare a datelor vor fi proiectate pentru a fi intuitive și ușor de utilizat. Elementele de date asociate fiecărui ecran includ parametrii de simulare, setările aplicației și opțiunile de preferință. Acestea pot include următoarele criterii de editare:

* **Valori Specifice**: Pentru parametrii fizici, cum ar fi masă, lungime sau accelerație gravitațională, valorile trebuie să fie în limitele realiste și să respecte unitățile de măsură corespunzătoare.
* **Interval de Valori**: Pentru parametrii care sunt supuși anumitor limite, cum ar fi viteza maximă sau dimensiunea minimă a obiectelor, se vor specifica intervalul de valori acceptate.
* **Obligatoriu/Opțional**: Anumite câmpuri pot fi obligatorii, în timp ce altele pot fi opționale, în funcție de necesitățile aplicației.
* **Valori Alfanumerice și Lungime**: Pentru câmpurile de text, se va specifica tipul de date (numeric, textual etc.) și lungimea maximă a valorilor acceptate.

Pentru a preveni erorile și a oferi ghidare utilizatorilor, aplicația va furniza mesaje de avertizare și de confirmare pentru intrările utilizatorului, precum și validarea în timp real a datelor introduse. Interfețele grafice vor fi proiectate pentru a ghida utilizatorii în procesul de introducere a datelor și pentru a evita confuziile sau erorile.

#### Ieșiri

1. **Rapoarte de Simulare**:
   * **Cod/Nume**: Raportul de simulare este identificat prin numărul sau numele sesiunii de simulare.
   * **Conținut**: Raportul cuprinde rezultatele simularii, cum ar fi grafice, diagrame și rezumate numerice ale comportamentului sistemului fizic în funcție de parametrii selectați.
   * **Scopul Ieșirii**: Rapoartele sunt destinate utilizatorilor finali și educatorilor pentru a analiza și a înțelege rezultatele simularilor fizice.
   * **Restricții de Acces/Securitate**: Nu exista.
2. **Ecrane de Afisare a Datelor**:
   * **Cod/Nume**: Ecranele de afișare a datelor sunt identificate prin numele funcționalității sau a secțiunii relevante a aplicației.
   * **Conținut**: Aceste ecrane afișează informații despre starea simularii, parametrii configurabili, setările aplicației și alte detalii relevante pentru utilizatori.
   * **Scopul Ieșirii**: Ecranele de afișare sunt destinate utilizatorilor pentru a vizualiza și a interacționa cu datele și setările sistemului.
   * **Restricții de Acces/Securitate**: Nu exista.

Pentru fiecare raport și ecran de afișare a datelor, se vor defini clar elementele de date asociate și structurile acestora. Aceste elemente de date pot include rezultatele simularii, parametrii configurabili, informații despre obiectele simulate și alte detalii relevante. Interfețele grafice ale rapoartelor și ecranelor de afișare vor fi proiectate pentru a fi intuitive și ușor de înțeles de către utilizatori.

### Proiectarea interfețelor cu utilizatorul

1. **Interfața Principală a Aplicației**:
   * **Descriere**: Acesta este ecranul principal al aplicației, care oferă acces la toate funcționalitățile și opțiunile principale. Va conține elemente precum meniuri, butoane pentru pornirea și oprirea simulării, selecția scenariilor și modificarea parametrilor.
   * **Aspect și Funcționalități**:
     + Meniul principal pentru navigare și acces rapid la funcții.
     + Buton pentru pornirea și oprirea simulației.
     + Zone de selectare a scenariilor și modificare a parametrilor.
2. **Ecrane de Configurare și Setări**:
   * **Descriere**: Aceste ecrane permit utilizatorilor să configureze preferințele și setările aplicației, precum și să modifice parametrii simulației.
   * **Aspect și Funcționalități**:
     + Tab-uri sau panouri pentru diferite categorii de setări (de exemplu, setări generale, setări de simulare).
     + Câmpuri de introducere pentru parametrii configurabili.
     + Buton de salvare a setărilor.
3. **Ecrane de Afișare a Datelor și Rezultatelor**:
   * **Descriere**: Aceste ecrane prezintă rezultatele simulației și informațiile despre starea actuală a sistemului fizic simulat.
   * **Aspect și Funcționalități**:
     + Grafice și diagrame pentru vizualizarea rezultatelor simulației.
     + Tabele și listări pentru prezentarea datelor numerice.
     + Opțiuni pentru exportul datelor și a rapoartelor.
4. **Ferestre de Confirmare și Avertizare**:
   * **Descriere**: Aceste ferestre apar pentru a confirma acțiunile utilizatorului sau pentru a avertiza asupra operațiilor care pot afecta datele sau simulația.
   * **Aspect și Funcționalități**:
     + Mesaje clare și concise pentru a informa utilizatorii despre consecințele acțiunilor lor.
     + Buton pentru confirmarea sau anularea acțiunii.
5. **Ferestre de Ajutor și Ghidare**:
   * **Descriere**: Aceste ferestre oferă asistență și ghidare utilizatorilor în utilizarea aplicației.
   * **Aspect și Funcționalități**:
     + Întrebări frecvente și răspunsuri (FAQ-uri).
     + Tutoriale și instrucțiuni pas cu pas pentru funcționalitățile cheie ale aplicației.

Interfețele cu utilizatorul vor fi proiectate pentru a fi intuitive, accesibile și ușor de utilizat, cu accent pe experiența utilizatorului și pe navigarea eficientă în cadrul aplicației. Elementele grafice și funcționalitățile vor fi implementate pentru a oferi utilizatorilor un mediu interactiv și captivant pentru învățarea și explorarea fenomenelor fizice.

## Scenarii de utilizare

1. **Pornirea și Configurarea Simulării**:
   * Utilizatorul deschide aplicația și este întâmpinat de ecranul principal.
   * Utilizatorul selectează opțiunea de a începe o nouă simulare.
   * Pe ecranul de configurare, utilizatorul alege scenariul de simulare dorit și modifică parametrii opționali.
   * Utilizatorul apasă butonul de pornire a simulației și observă cum sistemul începe să simuleze fenomenul selectat.
2. **Vizualizarea Rezultatelor Simulării**:
   * După ce simularea este finalizată, utilizatorul este redirecționat către ecranul de afișare a datelor.
   * Utilizatorul examinează graficele și diagramele care prezintă rezultatele simulației.
   * Utilizatorul poate ajusta vizualizarea sau poate exporta datele pentru analiză ulterioară.
3. **Modificarea Parametrilor în Timpul Simulării**:
   * În timpul simulației, utilizatorul dorește să modifice un parametru specific pentru a observa efectele asupra rezultatelor.
   * Utilizatorul revine la ecranul principal și selectează opțiunea de a accesa parametrii simulației.
   * Utilizatorul ajustează valorile parametrului și revine la simulație pentru a observa schimbările.
4. **Oprirea și Salvarea Simulării**:
   * Utilizatorul dorește să oprească simulația și să salveze rezultatele.
   * Utilizatorul apasă butonul de oprire a simulației și este întrebat dacă dorește să salveze rezultatele.
   * Dacă utilizatorul confirmă, rezultatele sunt salvate într-un fișier specificat de utilizator.
5. **Gestionarea Setărilor și Preferințelor**:
   * Utilizatorul accesează ecranele de setări și preferințe pentru a personaliza experiența sa cu aplicația.
   * Utilizatorul modifică setările generale, cum ar fi limba sau unitățile de măsură.
   * Utilizatorul salvează setările și observă cum acestea sunt aplicate în întreaga aplicație.

Aceste scenarii acoperă interacțiunile de bază pe care utilizatorii le vor avea cu sistemul nostru de simulare a fenomenelor fizice, oferind o perspectivă detaliată asupra funcționalității și a modului de utilizare a aplicației.

## Proiectare de detaliu

### Proiectare software de detaliu

1. **Identificator Serviciu**: Motorul de Simulare Fizică
   * **Clasificare**: Serviciu de calcul
   * **Definiție**: Acest serviciu este responsabil pentru executarea efectivă a simulării fenomenelor fizice, aplicând legile fizice și modificând starea sistemului în funcție de parametrii specificați.
   * **Cerințe**: Satisfacerea cerințelor de performanță și acuratețe a simulării, gestionarea eficientă a resurselor de calcul.
   * **Structuri de Date Interne**: Matrice de stări ale obiectelor simulate, liste de obiecte și parametrii asociate.
   * **Constrangeri**: Limitările de memorie și de putere de calcul disponibile.
   * **Compoziție**: Utilizează componente pentru gestionarea fizicii fundamentale, cum ar fi legile lui Newton sau algoritmi pentru detecția coliziunilor.
   * **Utilizatori/Interacțiuni**: Interacționează cu motorul grafic pentru afișarea rezultatelor și cu interfața utilizatorului pentru a primi parametrii de intrare.
   * **Procesare**: Utilizează algoritmi de integrare numerică și de detecție a coliziunilor pentru a actualiza starea simulării la fiecare pas de timp.
   * **Interfețe/Exporturi**: Oferă funcții pentru inițializarea simulării, actualizarea stării și obținerea rezultatelor.
2. **Identificator Serviciu**: Interfața Utilizatorului
   * **Clasificare**: Aplicație
   * **Definiție**: Acest serviciu furnizează o interfață grafică utilizatorului pentru a interacționa cu sistemul, inclusiv pentru a introduce parametrii de simulare și pentru a vizualiza rezultatele.
   * **Cerințe**: Interfață intuitivă și ușor de utilizat, afișarea clară și concisă a rezultatelor simulării.
   * **Structuri de Date Interne**: Reprezentări ale ecranelor și controalelor interfeței grafice.
   * **Constrangeri**: Limitări legate de platforma de dezvoltare și de interfața utilizatorului, cum ar fi dimensiunea ecranului sau abilitățile hardware-ului.
   * **Compoziție**: Folosește biblioteci și framework-uri pentru dezvoltarea interfeței grafice.
   * **Utilizatori/Interacțiuni**: Interacționează direct cu utilizatorii pentru a primi input și a afișa output.
   * **Procesare**: Gestionează evenimente de la tastatură și mouse, actualizează interfața grafică în timp real.
   * **Interfețe/Exporturi**: Oferă funcții pentru a primi input de la utilizatori și pentru a afișa date și rezultate.

Anexa A: Gestiunea modificărilor documentului

Instrucțiuni: Furnizați informații despre modul în care dezvoltarea și distribuția documentului va fi controlată și urmărită. Utilizați tabelul de mai jos pentru a furniza numărul de versiune, data versiunii, autorul/deținătorul versiunii și o scurtă descriere a motivului pentru crearea versiunii revizuite.

Tabel 1 – Înregistrarea modificărilor asupreaa documentului curent

| versiune | Data | Autorul/Deținătorul | Descriere |
| --- | --- | --- | --- |
| <X.X> | <ZZ/LL/AAAA> | <nume autor> | <Descrierea modificării> |
| <X.X> | <ZZ/LL/AAAA> | <nume autor> | <Descrierea modificării> |
| <X.X> | <ZZ/LL/AAAA> | <nume autor> | <Descrierea modificării> |

Anexa B: Acronime

*Instrucțiuni: Furnizați o listă de acronime și traduceri literale asociate utilizate în cadrul documentului. Enumerați acronimele în ordine alfabetică folosind un format tabular, așa cum este ilustrat mai jos.*

Tabel 2 - Acronime

| Acronim | Forma completă |
| --- | --- |
| <Acronim> | <Forma completă> |
| <Acronim> | <Forma completă> |
| <Acronim> | <Forma completă> |

Anexa C Documente la care se face referire

*Instrucțiuni: Sintetizați relația acestui document cu alte documente relevante. Furnizați informații de identificare pentru toate documentele folosite pentru a ajunge la și/sau referite în acest document (de exemplu, documente conexe și/sau asociate, documente prealabile, documentație tehnică relevantă, etc.).*

Tabel 3 – Documente la care se facce referire

| Nume document | Locație sau URL | Dată emitere document |
| --- | --- | --- |
| < Nume document > | <Locație sau URL> | <ZZ/LL/AAAA> |
| < Nume document > | <Locație sau URL> | <ZZ/LL/AAAA> |
| < Nume document > | <Locație sau URL> | <ZZ/LL/AAAA> |