

Software Requirements Specification

Version 1.0

29 Mai, 2023

Aplicație de simulare teste de admitere

Echipa 1310A-1310B-1311A

Hriscu Cornelia-Ștefana

Hușman Carla Gabriela

Pîslariu Andreea

Facultatea de Automatică și Calculatoare
Universitatea Tehnică „Gheorghe Asachi” din Iași

Cuprins

1. Introducere

- 1.1 Scop
- 1.2 Convenții ale documentului
- 1.3 Public vizat și conținutul documentului
- 1.4 Domeniul de aplicare
- 1.5 Referințe

2. Descriere generală

- 2.1 Perspectiva proiectului
- 2.2 Funcționalități
- 2.3 Caracteristicile utilizatorului
- 2.4 Mediul de operare
- 2.5 Constrângeri de proiectare și implementare
- 2.6 Documentația pentru user
- 2.7 Ipoteze și dependențe

3. Cerințe specifice

- 3.1 Interfața cu utilizatorul
- 3.2 Interfața hardware
- 3.3 Interfața software
- 3.4 Interfețe de comunicare

4. Alte cerințe nefuncționale

- 4.1 Cerințe de performanță
- 4.2 Cerințe de siguranță
- 4.3 Cerințe de securitate
- 4.4 Atribute de calitate software
- 4.5 Reguli de afaceri

1. Introducere

1.1. Scop

Aplicația *Teste Simulare Admitere* reprezintă un set de teste asemănătoare cu cele de admitere la facultatea noastră pe care oamenii le pot susține oricând doresc să își testeze nivelul. Aplicația pune la dispoziție 10 întrebări în funcție de materia aleasă de utilizator, acesta având două opțiuni: matematică sau informatică. Înainte de începerea testului, utilizatorul poate selecta timpul de care are nevoie pentru a finaliza testul. O întrebare are 4 variante de răspuns, și doar una reprezintă răspunsul corect. La finalul testului este afișat numărul de întrebări la care s-a răspuns corect, mesajul “Respins” sau “Admis”, posibilitatea verificării și întoarcerea la pagina cu materii pentru a da un nou test.

Documentul de specificarea cerințelor software prezent furnizează o descriere completă a proiectului intitulat „*Teste Simulare Admitere*” în cadrul Facultății de Automatică și Calculatoare, din Universitatea Tehnică "Gheorghe Asachi", Iași.

1.2. Convenții ale documentului

Acest document urmează formatarea standard IEEE pentru dezvoltarea software. Acest standard definește o formatare obișnuită pe care o urmează acest document care include scrierea care trebuie făcută la persoana a treia, voce pasivă, cât și text lizibil și corect din punct de vedere gramatical.

1.3. Public vizat și conținutul documentului

Acest document nu este destinat utilizatorului final, deoarece oferă o analiză detaliată a modului în care software-ul urmează să fie implementat. Din moment ce un utilizator are nevoie de informații despre cum poate parcurge testul, în loc de cum să creeze testul, acest document este mai mult orientat către testeri și mai ales dezvoltatorii testului.

În primul capitol se va găsi o introducere în care se specifică convențiile de scriere a documentului, dar și scopul acestuia și publicul vizat.

Capitolul al doilea își propune să facă o descriere generală a proiectului, care cuprinde funcționalitățile disponibile, caracteristicile unui utilizator, cât și constrângeri generale.

În al treilea capitol este ilustrat modul în care utilizatorul interacționează cu interfețele aplicației. De asemenea, sunt specificate și cerințele hardware(3.2).

Ultimul capitol, al patrulea, enumeră cerințele nefuncționale ale proiectului, cum ar fi cerințele de performanță(4.1) în care sunt specificate cerințele necesare rulării aplicației, cele de siguranță(4.2), de securitate(4.3), dar și atribute de calitate software(4.4) și reguli de afaceri(4.5).

1.4. Domeniul de aplicare

Teste Simulare Admitere este o aplicație creată să fie atât de intuitivă și de ușor de utilizat pe cât este de distractivă pentru utilizator. Este concepută să ruleze pe orice computer, indiferent de tipul procesorului.

1.5. Referințe

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements

2. Descriere generală a proiectului

2.1. Perspectiva proiectului

Aplicația *Teste Simulare Admitere* reprezintă un set de teste de matematică/informatică dezvoltat în IDE-ul Visual Studio 2019. Obiectivul testelor este de a testa cunoștințele utilizatorului în vederea susținerii examenului de admitere din cadrul Facultății de Automatică și Calculatoare.

Teste Simulare Admitere este o aplicație scrisă în C# pentru sistemul de operare Windows și va fi implementată ca o aplicație de tip Windows Forms.

2.2. Funcționalități

Funcțiile majore pe care aplicația *Teste Simulare Admitere* trebuie să le îndeplinească pentru utilizatorul său final sunt următoarele:

- Înregistrarea utilizatorului – aplicația permite utilizatorului să își poată crea un cont prin completarea câmpurilor nume, prenume, nume utilizator și parola
- Autentificarea utilizatorului – aplicația permite utilizatorului să se autentifice introducând numele de utilizator și parola
- Alegerea materiei – aplicația permite utilizatorului să își aleagă materia la care dorește să susțină testul
- Alegerea timpului – aplicația permite utilizatorului să poată alege intervalul de timp pe care îl dorește pentru susținerea testului

2.3. Caracteristicile utilizatorului

Aplicația *Teste Simulare Admitere* ar trebui să fie concepută pentru a fi ușor de utilizat, astfel încât orice categorie de utilizatori care dorește să-și testeze cunoștințele în timpul său liber să poată susține testul.

Simplitatea testului va permite oricărui utilizator generic să aibă parte de un test care să îi pună mintea la contribuție, indiferent de intervalul de timp pe care îl va alege.

2.4. Mediul de operare

Din moment ce aplicația este implementată ca o aplicație de tip Windows Forms, scrisă în C#, aceasta va putea fi compatibilă pe mai multe platforme. Utilizatorii de PC sau Mac nu ar trebui să întâmpine probleme în rularea programului.

2.5. Constrângeri de proiectare și implementare

Proiectarea și implementarea aplicației într-un singur semestru ca parte a proiectului pentru Facultatea de Automatică și Calculatoare, în cadrul materiei Ingineria Programării, impune constrângeri de timp considerabile în ciclul de dezvoltare. De asemenea, este necesar să dezvoltăm întreaga aplicație pe platforma .NET, ceea ce devine o provocare suplimentară din cauza lipsei noastre de experiență în limbajul C# și în proiectarea de software și documentație.

Costul dezvoltării acestui proiect nu reprezintă un factor limitativ, deoarece nu există implicații financiare asociate. Principalele provocări sunt reprezentate de gestionarea timpului și de acomodarea cu tehnologiile și conceptele specifice proiectului, având în vedere resursele limitate disponibile.

2.6.Documentația pentru user

Aplicația este însoțită de o documentație pentru utilizator (HelpnDoc).

2.7.Ipoteze și dependențe

-

3. Cerințe de interfață externă

3.1. Interfața cu utilizatorul

În această secțiune sunt descrise caracteristicile și constrângerile interfeței grafice (GUI) cu utilizatorul.

1. Fereastra principală a aplicației reprezintă un Form cu cele 4 butoane cu care se poate interacționa (*AUTENTIFICARE*, *ÎNREGISTRARE*, *AJUTOR*, *IEȘIRE*)
2. Prin apăsarea butonului *AUTENTIFICARE* se va închide fereastra principală și se va deschide o fereastră nouă care va permite autentificarea utilizatorului. După ce câmpurile nume utilizator și parolă au fost completate se apasă pe butonul de *INTRĂ*

ÎN CONT. Dacă contul există utilizatorul este redirecționat către o altă fereastră, însă dacă acesta este inexistent se va afișa mesajul “Nume utilizator sau parolă incorecte!” fiind recomandat ca utilizatorul să apese pe butonul *ÎNAPOI* pentru a se putea întoarce la pagina inițială și să își creeze un cont.

3. Prin apăsarea butonului *ÎNREGISTRARE* se va închide fereastra principală și se va deschide o fereastră nouă care va permite utilizatorului să își creeze un cont prin completarea câmpurilor Nume, Prenume, Nume Utilizator, Parolă și apăsând ulterior pe butonul de *CREAZĂ CONT*. Dacă numele de utilizator sau parola nu respectă condițiile impuse se va afișa mesajul “Nume de utilizator sau parole prea scurte. Minim 5 caractere.”, iar dacă condițiile sunt respectate utilizatorul va fi redirecționat către pagina de autentificare. Dacă acesta se răzgândește pe parcursul creării contului poate apăsa pe butonul *ÎNAPOI* pentru a se putea întoarce la pagina inițială.
4. Prin apăsarea butonului *AJUTOR* se va deschide o fereastră ce conține instrucțiuni pentru utilizator.
5. Prin apăsarea butonului *IEȘIRE* se va închide fereastra principală.
6. După autentificarea cu succes se va deschide o nouă fereastră care va conține mesajul “Înainte de a începe, alege timpul pe care îl dorești pentru parcurgerea simulării.”, utilizatorul fiind nevoit să apese pe butonul *SETĂRI*. Odată setat timpul, se vor activa butoanele *MATEMATICĂ* și *INFORMATICĂ* pentru a putea alege materia la care se dorește a da simulare. Pe interfață se mai regăsește și butonul Deconectare care permite utilizatorului ieșirea din cont și revenirea la pagina principală.
7. Prin apăsarea butonului *SETĂRI* se va deschide o fereastră care va permite utilizatorului să își seteze intervalul de timp dorit pentru efectuarea simulării. După alegerea opțiunii se va apăsa butonului *APLICĂ* ceea ce va avea ca efect închiderea ferestrei actuale.
8. Prin apăsarea butonului *MATEMATICĂ* sau *INFORMATICĂ* se va închide fereastra cu opțiuni și se va deschide una nouă care conține întrebarea și cele 4 variante de răspuns. În această fereastră există butonul care permite trecerea la următoare întrebare de abia după ce una din variante este selectată. În colț dreapta-sus utilizatorul poate verifica cât timp mai are al dispoziție pentru completarea testului.
9. La ultima întrebare butonul *URMĂTOAREA ÎNTREBARE* devine *FINALIZEAZĂ TESTUL*. Prin apăsarea butonului se va închide fereastra cu Întrebări și se va deschide o nouă fereastră în care va fi afișat rezultatul obținut, mesajul “Admis” sau “Respins”, posibilitatea utilizatorului de a se verifica prin parcurgerea întrebărilor cu ajutorul butoanelor *ÎNAINTE/ÎNAPOI* și observarea concordanței dintre varianta de răspuns corectă și cea aleasă de acesta. Totodată mai există și un buton de întoarcere la fereastra cu materii pentru a susține un nou test.

3.2. Interfața hardware

Interfețele hardware vor include un mouse și monitorul de afișare. Nu este nevoie de mult hardware pentru a putea rula aplicația, altul decât un simplu computer și un monitor. Toată interacțiunea se realizează prin apăsarea click stânga pe mouse și completarea câmpurilor (acolo unde este necesar) cu ajutorul tastaturii.

3.3. Interfața software

Pentru a putea rula cu succes aplicația, utilizatorul trebuie să aibă minim versiunea .NET Framework 4.5.

3.4. Interfețe de comunicare

-

4. Cerințe nefuncționale

4.1. Cerințe de performanță

Testul este conceput pentru a putea fi rulat pe orice versiune de Windows. Este recomandat de avut măcar 32 Mb RAM, totuși 128 Mb sunt preferați. Placa grafică integrată ar trebui să fie suficientă pentru a avea o experiență optimă.

4.2. Cerințe de siguranță

Testul nu ar trebui să conțină niciun risc pentru utilizator.

4.3. Cerințe de securitate

Testul nu se folosește de nicio informație privată din partea utilizatorilor.

4.4. Atribute de calitate software

Acest software trebuie să fie robust și cât mai lipsit de erori posibil pentru a se asigura că utilizatorii au o experiență pozitivă. Testul ar trebui să fie ușor de parcurs și cât se poate de intuitiv pentru un începător, având și o latură de învățare.

4.5. Reguli de afaceri

Politica echipei de dezvoltare este să respecte toate codurile de conduită și de scriere a codului stabilite în cadrul materiei Ingineria Programării.

Software Design Document

Version 1.0

29 Mai, 2023

Aplicație de simulare teste de admitere

Echipa 1310A-1310B-1311A

Hriscu Cornelia-Ștefana

Hușman Carla Gabriela

Pîslariu Andreea

Facultatea de Automatică și Calculatoare
Universitatea Tehnică „Gheorghe Asachi” din Iași

Cuprins

Cuprins	2
1. Scopul documentului	3
2. Conținutul documentului	3
3. Modelul datelor	3
3.1. Structuri de date globale	3
3.2. Structuri de date de legătură	3
3.3. Structuri de date temporare	4
3.4. Formatul fișierelor utilizate	4
3.5. Descrierea bazei de date	4
3.5.1. Diagrama schemei bazei de date	4
3.5.2. Descrierea tabelelor	5
4. Modelul arhitectural și modelul componentelor	7
4.1. Arhitectura sistemului	7
4.1.1. Șabloane arhitecturale folosite	7
4.1.2. Diagrama de arhitectură	7
4.2. Descrierea componentelor	8
4.3. Restricțiile de implementare	9
4.4. Interacțiunea dintre componente	9
5. Modelul interfeței cu utilizatorul	11
5.1. Succesiunea interfețelor	11
5.2. Ferestrele aplicației	12
5.2.1. Fereastra Intro	12
5.2.2. Fereastra de autentificare	13
5.2.3. Fereastra de înregistrare	14
5.2.4. Fereastra de setări	15
5.2.5. Fereastra de parcurgere a testului	
5.2.6. Fereastra de vizualizare a rezultatelor	16
6. Elemente de testare	18
6.1. Componente critice	18
6.2. Alternative	18

1. Scopul documentului

Acest document descrie proiectarea soluției pentru o aplicație de simulare a testelor de admitere la Facultatea de Automatică și Calculatoare. Această aplicație va fi utilizată de elevii care doresc să se pregătească pentru testele de admitere și va permite utilizatorilor să practice probleme de matematică și informatică.

2. Conținutul documentului

Documentul este format din patru secțiuni esențiale:

- *Modelul datelor* – prezintă principalele structuri de date folosite, precum și schema bazei de date
- *Modelul arhitectural și modelul componentelor* – prezintă șabloanele arhitecturale folosite, arhitectura sistemului și descrie componentele arhitecturii
- *Modelul interfeței cu utilizatorul* – prezintă interfața cu utilizatorul și succesiunea ferestrelor acesteia
- *Elemente de testare* – prezintă componentele critice și alternative de proiectare a acestora.

3. Modelul datelor

3.1. Structuri de date globale

Aplicația are o instanță globală a clasei de conexiune la baza de date, implementată folosind șablonul de proiectare Singleton. Această instanță permite tuturor modulelor din aplicație să aibă acces la conexiunea la baza de date și să efectueze operațiuni CRUD pe tablele respective. De asemenea, aplicația utilizează o structură de date globală pentru gestionarea stărilor quiz-ului, implementată folosind șablonul de proiectare Stare.

3.2. Structuri de date de legătură

Aplicația se conectează la o bază de date SQLite pentru a stoca și recupera date despre utilizatori și imagini pentru întrebările din test. Modulul GUI al aplicației comunica cu modulul de baze de date prin intermediul argumentelor ce conțin cheile primare din schema bazei de date (de exemplu, identificatorii de utilizatori sau de întrebări de test). Aplicația utilizează structuri de date de legătură pentru a gestiona aceste comunicări între module.

3.3. Structuri de date temporare

Aplicația nu utilizează structuri de date temporare care să presupună un consum semnificativ de resurse de memorie.

3.4. Formatul fișierelor utilizate

În ceea ce privește formatul fișierelor utilizate, aplicația de teste de simulare a admiterii utilizează imagini pentru a afișa întrebările de test, care sunt stocate direct în baza de date SQLite utilizând tipul de date BLOB. Astfel, imaginile sunt convertite și salvate în baza de date ca obiecte BLOB, ceea ce permite gestionarea lor eficientă prin intermediul sistemului de gestiune a bazei de date.

3.5. Descrierea bazei de date

3.5.1. Diagrama schemei bazei de date

Modelul bazei de date este format din următoarele tabele prezentate în figura de mai jos:

Utilizatori		IntrebariInformatica		IntrebariMatematica	
Nume	varchar	Intrebare	blob	Intrebare	blob
Prenume	varchar	Raspuns	integer	Raspuns	integer
NumeUtilizator	varchar				
Parola	varchar				

3.5.2. Descrierea tabelor

Schema bazei de date cuprinde următoarele tabele:

- **Utilizatori** – conține informații despre utilizatori, precum numele, prenumele, numele de utilizator, și parola. Tipul de date al coloanelor Nume, Prenume, NumeUtilizator, Parola este de tip varchar și poate stoca până la 255 de caractere.
- **IntrebariMatematica & IntrebariInformatica** – conțin informații despre întrebările pentru testul de informatică și matematică, respectiv. Acestea conțin două coloane: **Intrebare** și **Raspuns**. Coloana **Intrebare** este de tip BLOB, care poate stoca date binare, cum ar fi imagini, iar coloana **Raspuns** este de tip integer și conține răspunsul corect la întrebare.

4. Modelul arhitectural și modelul componentelor

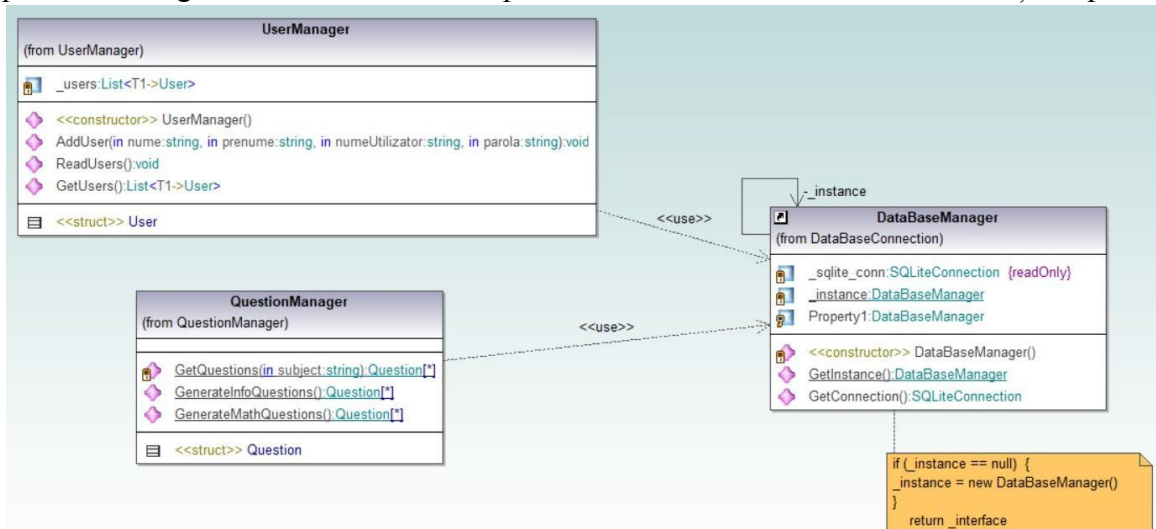
4.1. Arhitectura sistemului

4.1.1. Șabloane arhitecturale folosite

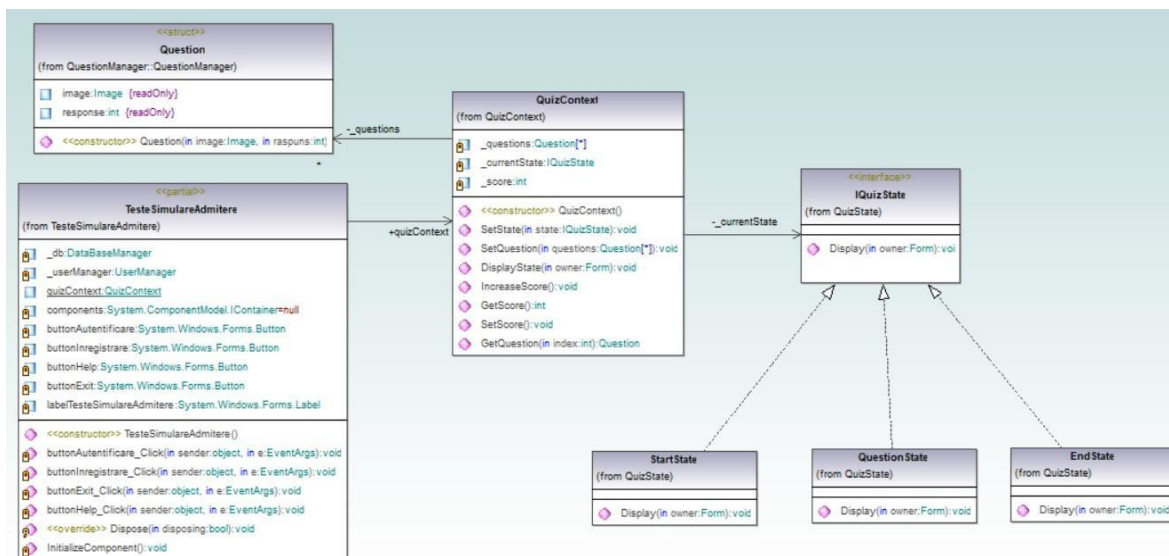
Aplicația este proiectată folosind modelul arhitectural Model-View-Controller (MVC) Această arhitectură este utilizată pentru a separa modelul de date, logica de afaceri și interfața cu utilizatorul în componente distincte, ușor de gestionat și de modificat. Modelul este responsabil pentru stocarea datelor, logica de afaceri și validarea datelor. View-ul este responsabil pentru afișarea datelor utilizatorului, precum și pentru colectarea și transmiterea datelor către controller. Controller-ul este responsabil pentru procesarea datelor, interacțiunea cu modelul și cu view-ul, precum și pentru gestionarea fluxului de date în aplicație.

În implementarea aplicației, s-au utilizat următoarele șabloane de proiectare:

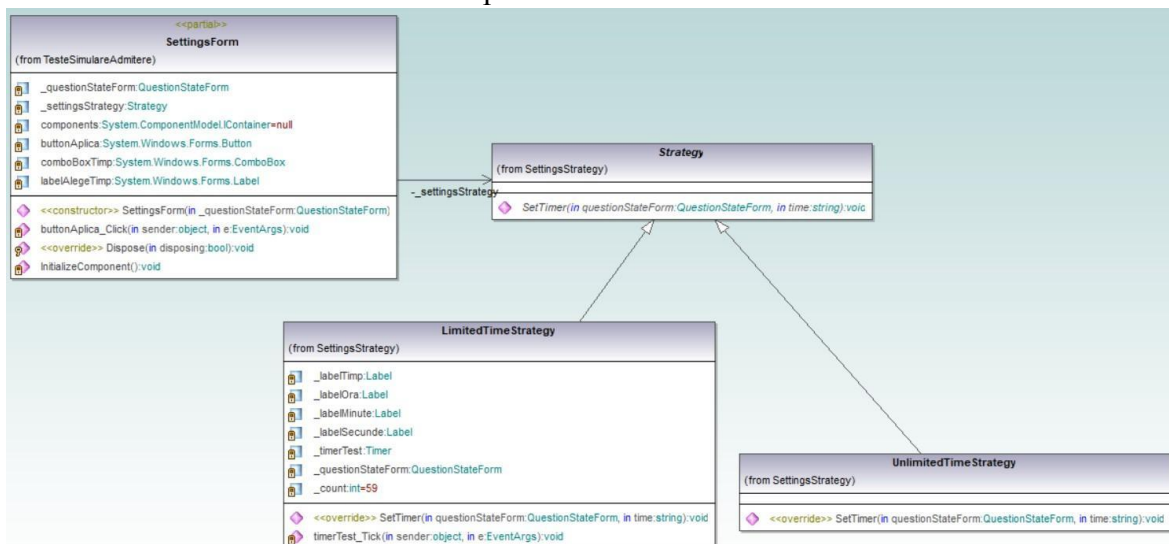
- **Singleton:** Acest șablon arhitectural este utilizat pentru a asigura că există o singură instanță a unei clase în aplicație. În cazul acestei aplicații, Singleton este utilizat pentru a garanta că există o singură conexiune la baza de date, indiferent de numărul de utilizatori care accesează aplicația. Astfel, conexiunea la baza de date va fi mai eficientă și se evită problemele legate de deschiderea multiplelor conexiuni la baza de date în același timp.



- **Șablonul de stare:** Acest șablon arhitectural este utilizat pentru a gestiona stările diferite ale quiz-ului. În cazul acestei aplicații, stările pot fi începutul quiz-ului, momentul în care se răspund la întrebări și încheierea quiz-ului. Fiecare stare poate avea comportamente specifice asociate cu ea, cum ar fi încărcarea întrebărilor în cazul stării de început, evaluarea răspunsurilor în cazul stării de quiz și afișarea rezultatelor în cazul stării de încheiere a quiz-ului.



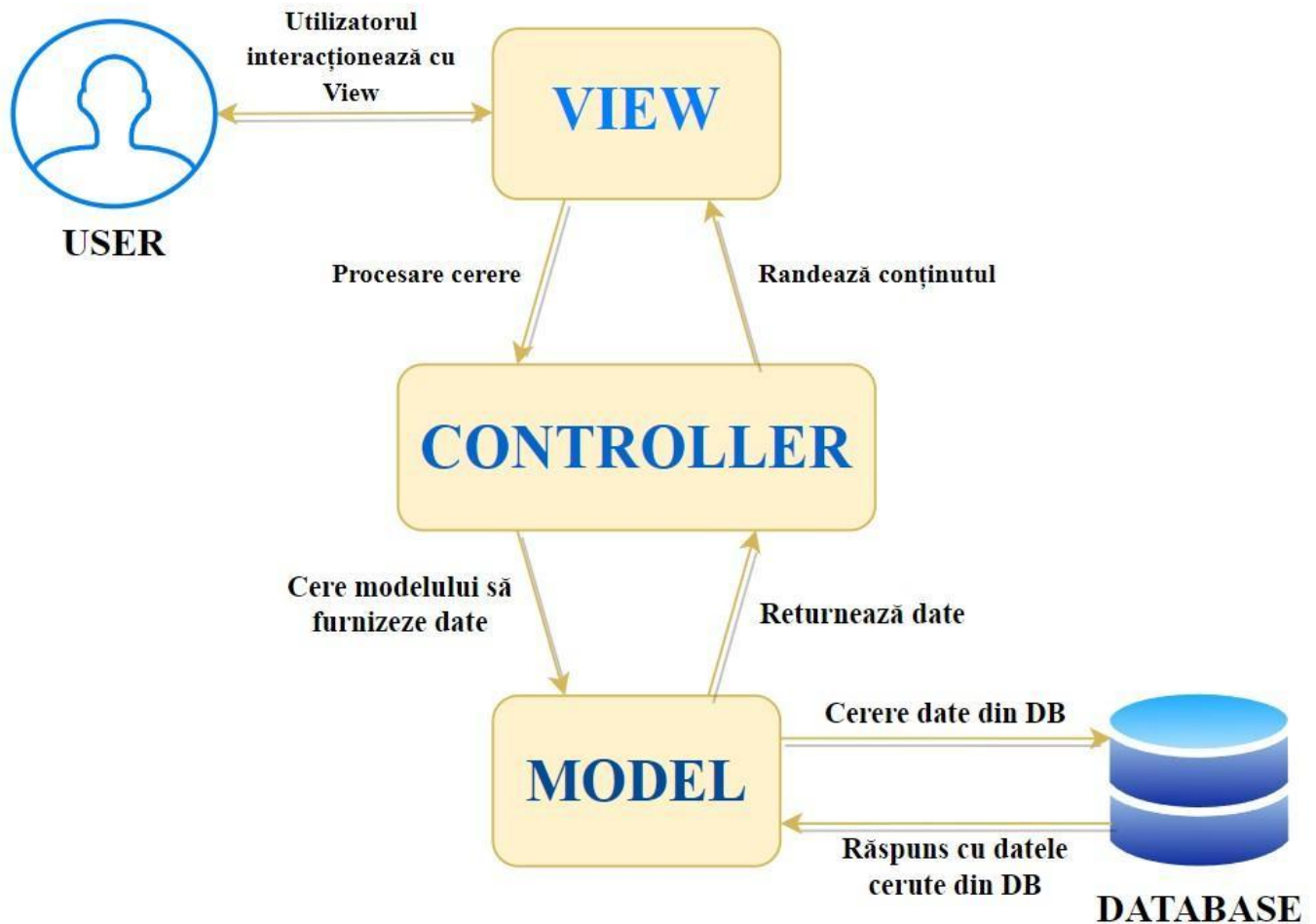
- **Șablonul de strategie:** Acest șablon arhitectural este utilizat pentru a permite utilizatorului să aleagă modul în care se va desfășura testul. În cazul acestei aplicații, utilizatorii pot alege între un test cu timp nelimitat sau un test cu timp limitat și pot selecta durata testului. Această implementare ar putea implica crearea a două strategii diferite pentru desfășurarea testului, una cu timp limitat și una fără limită de timp, și o metodă pentru a permite utilizatorului să selecteze una dintre acestea și să introducă durata testului în cazul testului cu timp limitat.



Aceste șabloane arhitecturale sunt utilizate pentru a îmbunătăți modularitatea, scalabilitatea și extensibilitatea aplicației, făcând-o mai ușor de dezvoltat și de întreținut pe termen lung.

4.1.2. Diagrama de arhitectură

Diagrama de arhitectură de mai jos descrie componentele arhitecturii aplicației și relațiile de interacțiune dintre acestea.



4.2. Descrierea componentelor

Aplicația constă din următoarele module interconectate. Aceste module lucrează împreună pentru a implementa arhitectura MVC și pentru a asigura o separare clară între logica de business, interfața utilizatorului și manipularea datelor.

- o **Modulul GUI** (Graphical User Interface)

Acesta reprezintă componenta View din arhitectura MVC. Modulul GUI este responsabil de interfața grafică cu care utilizatorul interacționează. Acesta afișează informațiile către utilizator și primește acțiunile utilizatorului prin intermediul elementelor de interfață, cum ar fi butoane, câmpuri de text, etc. Modulul GUI nu conține logica de afaceri, ci doar se ocupă de prezentarea și interacțiunea cu utilizatorul.

o **Modulul de Logică**

Acesta reprezintă componenta Controller din arhitectura MVC. Modulul de Logică conține logica de afaceri a aplicației. Acesta se ocupă de procesarea datelor și implementarea regulilor de business specifice. Modulul de Logică primește acțiunile utilizatorului de la modulul GUI și gestionează interacțiunea cu modulul de Baze de date și cu modulul GUI pentru a realiza operațiile necesare. În aplicația de simulare a testelor de admitere, modulul de Logică include logica pentru generarea întrebărilor de test, evaluarea răspunsurilor și calcularea rezultatelor.

o **Modulul de Baze de date**

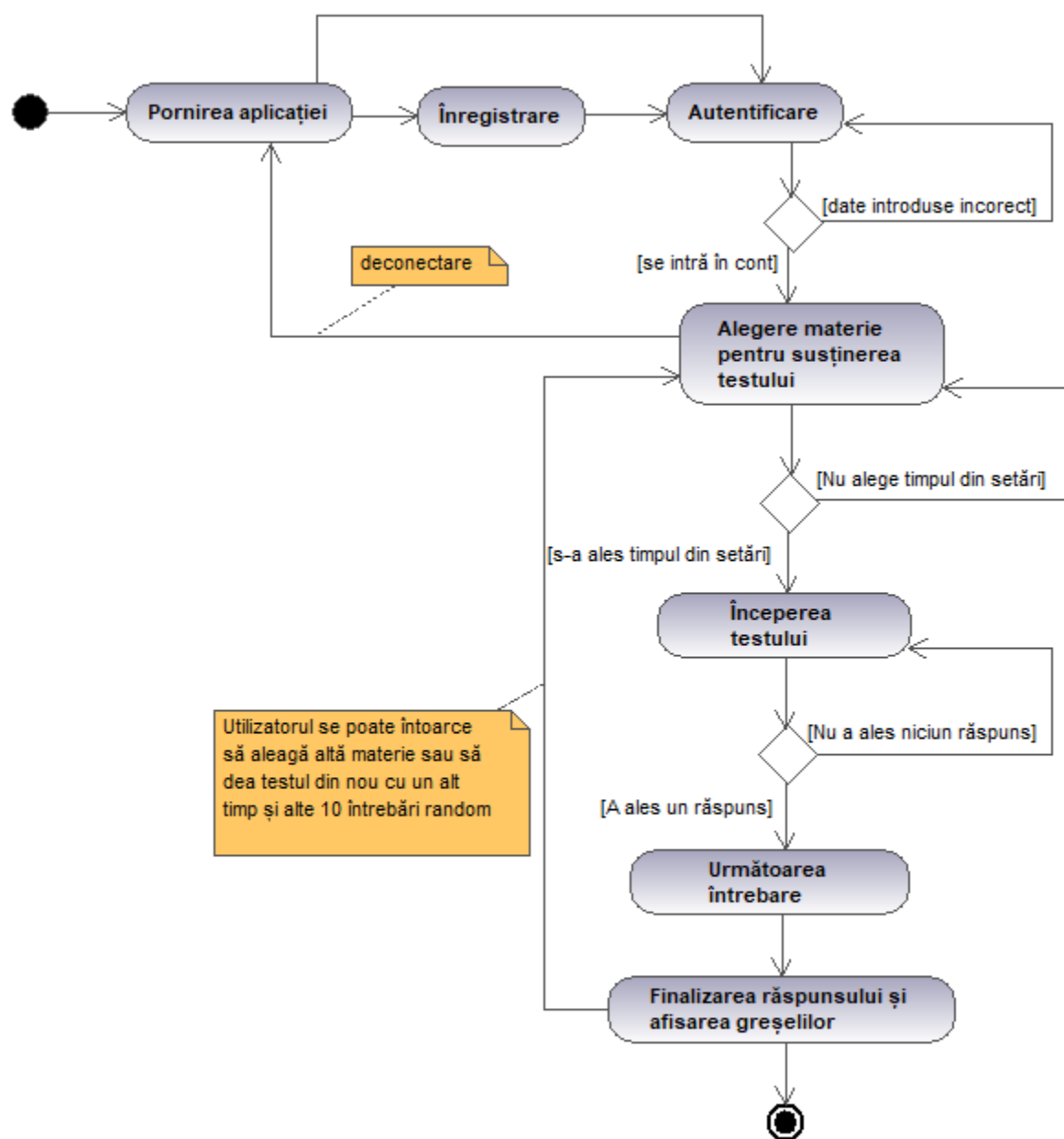
Acesta reprezintă componenta Model din arhitectura MVC. Modulul de Baze de date gestionează stocarea și accesul la date. Acesta se ocupă de crearea, actualizarea și interogarea bazei de date utilizate în aplicație. Modulul de Baze de date poate include clase și metode care facilitează manipularea datelor și comunicarea cu sistemul de gestiune a bazei de date. În aplicația de simulare a testelor de admitere, modulul de Baze de date ar gestiona stocarea și recuperarea întrebărilor și răspunsurilor din baza de date SQLite.

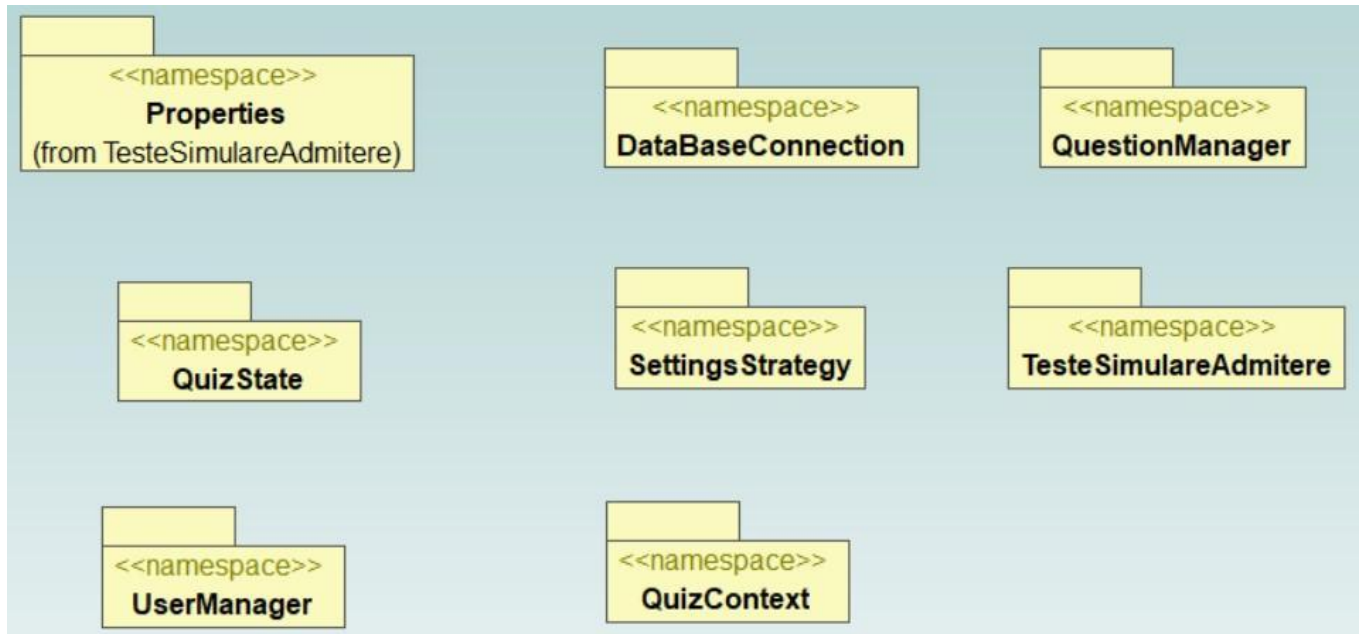
4.3. Restricțiile de implementare

- Va fi dezvoltat un modul de Baze de date care să faciliteze comunicarea cu baza de date SQLite utilizând DB Browser SQLite. Acest modul va include funcționalități pentru crearea și administrarea bazei de date, precum și pentru stocarea și recuperarea datelor relevante pentru simularea testelor de admitere.
- Modulul GUI, modulul de Logică și modulul de Baze de date vor fi implementate în limbajul de programare C#.
- Modulele vor fi dezvoltate ca proiecte de tip class library, generând DLL-uri separate.
- Se vor respecta principiile și standardele de programare specifice limbajului C# în implementarea modulelor.
- Se vor utiliza bibliotecile și framework-urile disponibile în C# pentru a obține funcționalitățile dorite în fiecare modul.

4.4. Interacțiunea dintre componente

Interacțiunea dintre componentele aplicației se bazează pe separarea responsabilităților și utilizarea șabloanelor de proiectare pentru a crea o arhitectură modulară, ușor de întreținut și extensibilă. DLL-urile separat implementate ca proiecte de tip class library oferă posibilitatea de a organiza și împărți codul în funcție de funcționalități și responsabilități specifice.



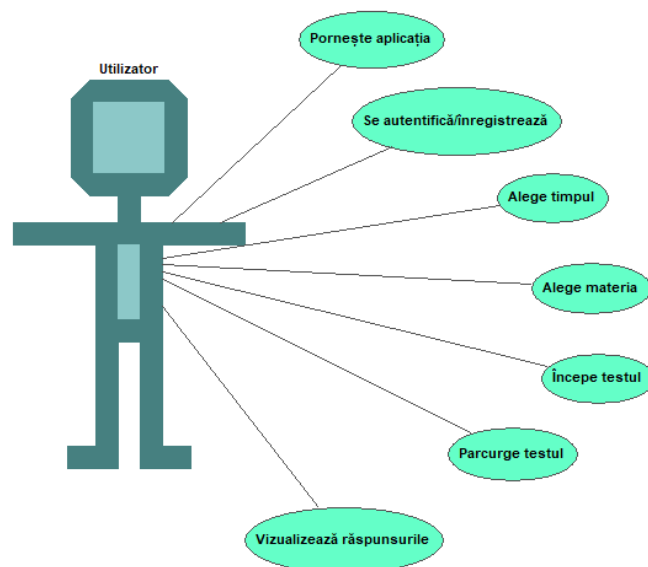
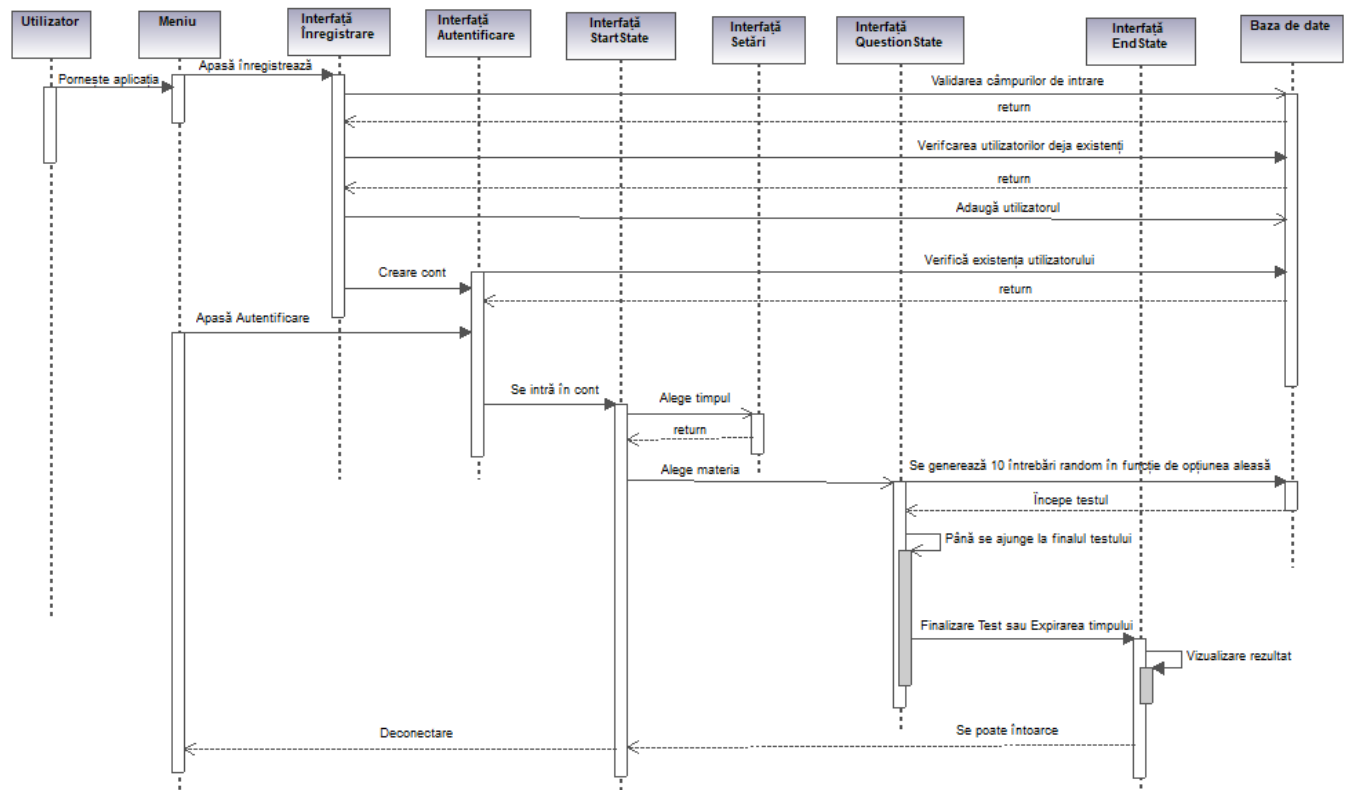


- Interacțiunea între client și modulul GUI:
 - Utilizatorul interacționează cu aplicația prin intermediul modulului GUI. Modulul GUI, implementat într-un proiect de tip Windows Form furnizează interfața grafică și elementele vizuale pentru interacțiunea cu utilizatorul.
- Interacțiunea dintre modulul GUI și modulul de Logică:
 - Modulul GUI, comunică cu modulul de Logică pentru a transmite acțiunile utilizatorului și a primi rezultatele procesării.
- Interacțiunea dintre modulul de Logică și modulul de Baze de date:
 - Modulul de Logică utilizează metode și funcționalități din modulul de Baze de date pentru a efectua operațiile de stocare, interogare și manipulare a datelor necesare pentru simularea testelor de admitere.
- Interacțiunea dintre modulele aplicației și DLL-urile separate:
 - DLL-urile separate, implementate ca proiecte de tip class library, conțin implementări ale unor șabloane de proiectare specifice, cum ar fi Singleton, State și Strategy.
 - Modulele aplicației utilizează DLL-urile separate prin includerea acestora ca referințe în proiectele respective.
 - Utilizarea DLL-urilor separate permite împărțirea funcționalităților și a logicii comune între module, facilitând reutilizarea și modularizarea codului.

5. Modelul interfeței cu utilizatorul

5.1. Succesiunea interfețelor

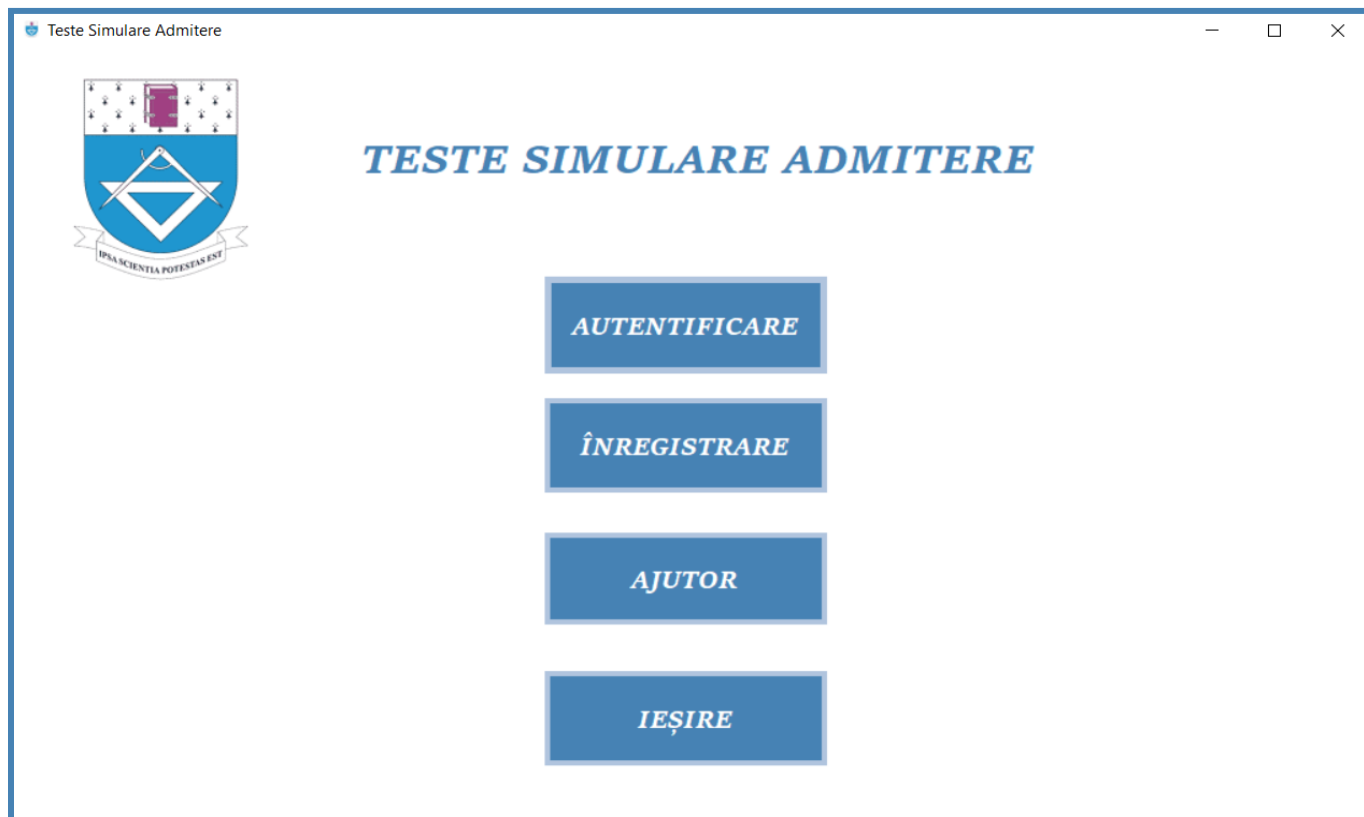
În cadrul modului GUI, ferestrele aplicației sunt afișate respectând un flux stabilit în conformitate cu standardele academice. Acest flux este descris grafic în figura de mai jos:



5.2. Ferestrele aplicației

5.2.1. Fereastra Intro

Fereastra principală a aplicației corespunde clasei TesteSimulareAdmitere.cs și arată astfel:

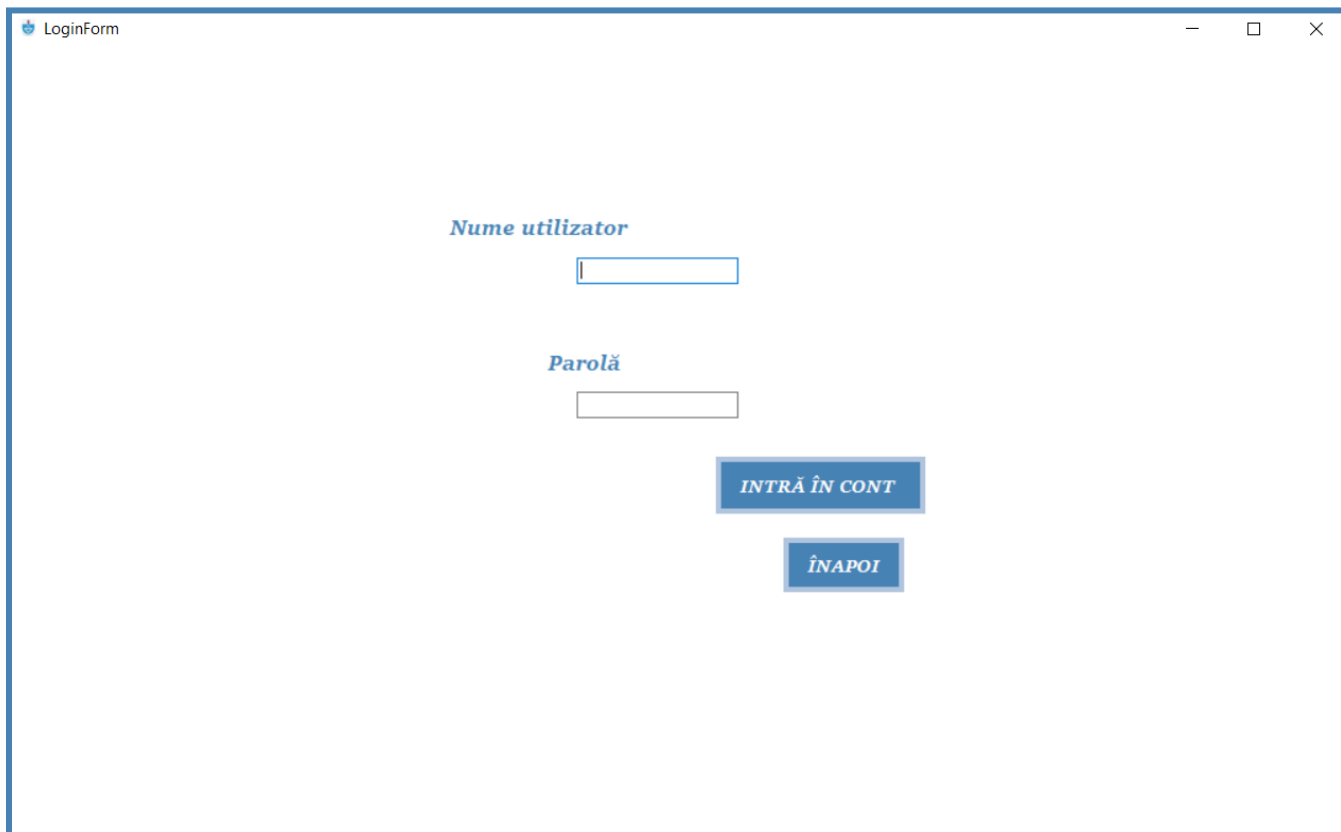


Această fereastră este panoul de comandă al utilizatorului în sensul că, de aici, utilizatorul alege ce acțiune dorește, determinând în felul acesta executarea unui flux specific.

- Butonul **AUTENTIFICARE**
 - Permite autentificarea în aplicație pentru a accesa funcționalitățile disponibile.
- Butonul **ÎNREGISTRARE**
 - Oferă posibilitatea de creare a unui cont nou pentru a beneficia de toate funcționalitățile disponibile.
- Butonul **HELP**
 - Furnizează acces la documentația aplicației, care oferă instrucțiuni detaliate despre modul de utilizare a aplicației.
- Butonul **EXIT**
 - Permite închiderea aplicației

5.2.2. Fereastra de autentificare

Prin apăsarea butonului *AUTENTIFICARE* se va închide fereastra principală și se va deschide o fereastră nouă care va permite autentificarea utilizatorului. După ce câmpurile nume utilizator și parolă au fost completate se apasă pe butonul de *INTRĂ ÎN CONT*. Dacă contul există utilizatorul este redirecționat către o altă fereastră, însă dacă acesta este inexistent se va afișa mesajul “Nume utilizator sau parolă incorecte!” fiind recomandat ca utilizatorul să apese pe butonul *ÎNAPOI* pentru a se putea întoarce la pagina inițială și să își creeze un cont.



The image shows a window titled "LoginForm" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a login form with the following elements:

- A label "Nume utilizator" in blue text above a text input field.
- A label "Parolă" in blue text above a text input field.
- A blue button labeled "INTRĂ ÎN CONT" positioned below the password field.
- A blue button labeled "ÎNAPOI" positioned below the login button.

5.2.3. Fereastra de înregistrare

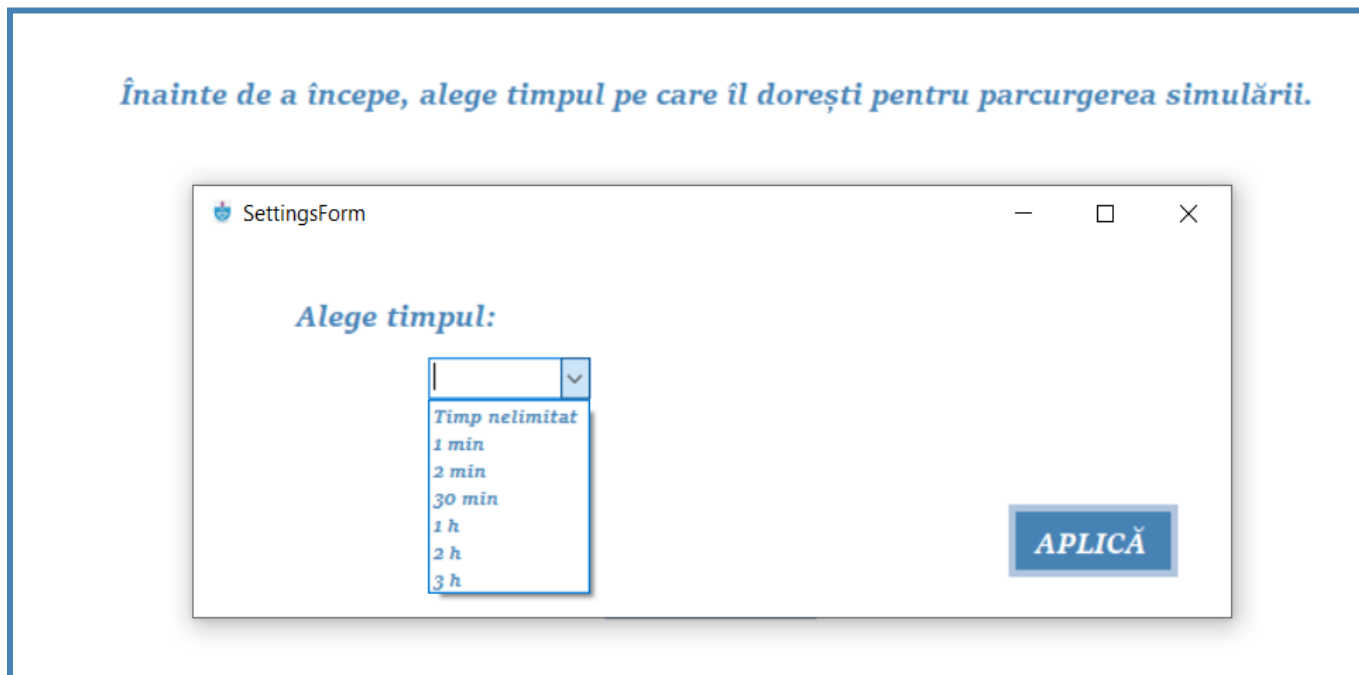
Prin apăsarea butonului **ÎNREGISTRARE** se va închide fereastra principală și se va deschide o fereastră nouă care va permite utilizatorului să își creeze un cont prin completarea câmpurilor Nume, Prenume, Nume Utilizator, Parolă și apăsând ulterior pe butonul de **CREAZĂ CONT**. Dacă numele de utilizator sau parola nu respectă condițiile impuse se va afișa mesajul “Nume de utilizator sau parole prea scurte. Minim 5 caractere.”, iar dacă condițiile sunt respectate utilizatorul va fi redirecționat către pagina de autentificare. Dacă acesta se răzgândește pe parcursul creării contului poate apăsa pe butonul **ÎNAPOI** pentru a se putea întoarce la pagina inițială.

The image shows a window titled "RegisterForm" with standard Windows window controls (minimize, maximize, close). The window contains a registration form with the following elements:

- Two input fields for "Nume" (Name) and "Prenume" (First Name), each with a blue label above it.
- Two input fields for "NumeUtilizator" (Username) and "Parolă" (Password), each with a blue label above it.
- A blue button labeled "CREAZĂ CONT" (Create Account) positioned below the input fields.
- A blue button labeled "ÎNAPOI" (Back) positioned below the "CREAZĂ CONT" button.

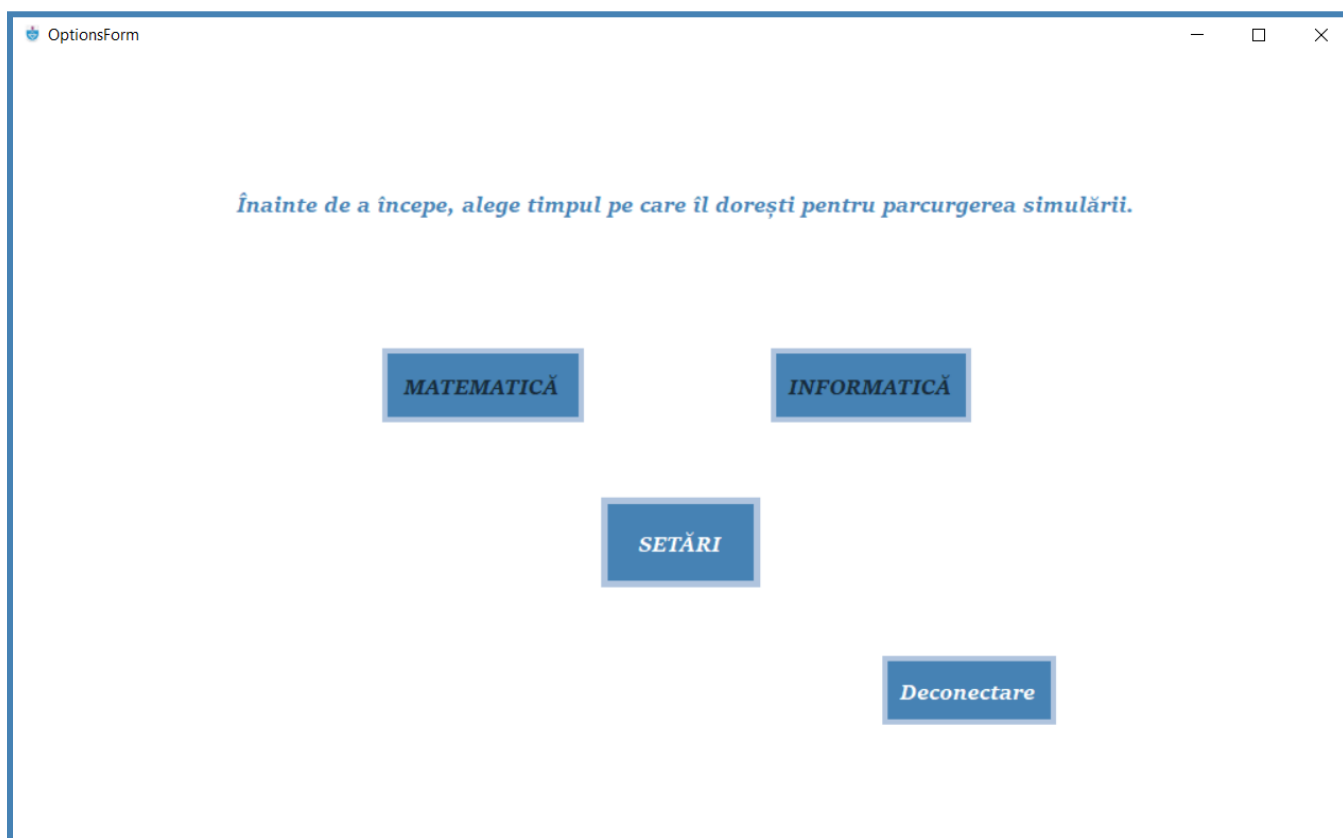
5.2.4. Fereastra de setări

Prin apăsarea butonului **SETĂRI** se va deschide o fereastră care va permite utilizatorului să își seteze intervalul de timp dorit pentru efectuarea simulării. După alegerea opțiunii se va apăsa butonului **APLICĂ** ceea ce va avea ca efect închiderea ferestrei actuale.



5.2.5. Fereastra de parcurgere a testului

După autentificarea cu succes se va deschide o nouă fereastră care va conține mesajul “Înainte de a începe, alege timpul pe care îl dorești pentru parcurgerea simulării.”, utilizatorul fiind nevoit să apese pe butonul **SETĂRI**. Odată setat timpul, se vor activa butoanele **MATEMATICĂ** și **INFORMATICĂ** pentru a putea alege materia la care se dorește a da simulare. Pe interfață se mai regăsește și butonul Deconectare care permite utilizatorului ieșirea din cont și revenirea la pagina principală.



Prin apăsarea butonului **MATEMATICĂ** sau **INFORMATICĂ** se va închide fereastra cu opțiuni și se va deschide una nouă care conține întrebarea și cele 4 variante de răspuns. În această fereastră există butonul care permite trecerea la următoare întrebare de abia după ce una din variante este selectată. În colț dreapta-sus utilizatorul poate verifica cât timp mai are al dispoziție pentru completarea testului.

TestForm

Matematică

Timp ramas: 0 h 1 m 51 s

Fie $F : (-1, \infty) \rightarrow \mathbb{R}$ primitiva funcției $f : (-1, \infty) \rightarrow \mathbb{R}$, $f(x) = \frac{3(x-1)}{x^3+1}$ care satisface $F(1) = 0$. Atunci $F(0)$ este:

(a) 0; (b) 1; (c) $-2 \ln 2$; (d) $2 \ln 2$.

☐ A

☐ B

☐ C

☐ D

Următoarea întrebare

5.2.6. Fereastra de vizualizare a rezultatelor

La ultima întrebare butonul **URMĂTOAREA ÎNTREBARE** devine **FINALIZEAZĂ TESTUL!**. Prin apăsarea butonului se va închide fereastra cu Întrebări și se va deschide o nouă fereastră în care va fi afișat rezultatul obținut, mesajul “Admis” sau “Respins”, posibilitatea utilizatorului de a se verifica prin parcurgerea întrebărilor cu ajutorul butoanelor **ÎNAINTE/ÎNAPOI** și observarea concordanței dintre varianta de răspuns corectă și cea aleasă de acesta. Totodată mai există și un buton de întoarcere la fereastra cu materii pentru a susține un nou test.

The screenshot shows a window titled "EndStateForm" with standard Windows window controls (minimize, maximize, close). The main content area displays the following information:

- Result: **Rezultatul: 5/10**
- Status: **Admis**
- Question text: **În $\triangle ABC$ avem $m(\hat{A}) = \frac{\pi}{3}$, $|AB| = 3$, $|AC| = 4$. Lungimea laturii BC este:**
- Options: **(a) 5; (b) $\sqrt{13}$; (c) $\frac{\sqrt{13}}{13}$; (d) 7.**
- Navigation buttons: **<** and **>** (blue buttons with white text).
- Return button: **Întoarcere** (blue button with white text).
- Feedback text at the bottom: **Răspuns corect: B** and **Ai răspuns: A** (both in red text).

6. Elemente de testare

6.1. Componente critice

Componentele critice pentru aplicația de teste simulate admitere ar putea fi:

- Modulul de generare a întrebărilor de test și evaluarea răspunsurilor: Acest modul este esențial pentru a asigura corectitudinea și relevanța întrebărilor de test, precum și evaluarea corectă a răspunsurilor furnizate de utilizatori.
- Modulul de interacțiune cu baza de date: Acest modul gestionează stocarea și recuperarea întrebărilor, răspunsurilor și altor date relevante utilizate în aplicație. O funcționare corectă și eficientă a acestui modul este crucială pentru a asigura integritatea datelor și accesul rapid la informațiile necesare.

6.2. Alternative

Alternativele pentru a optimiza aceste componente critice în cadrul aplicației de teste simulate admitere ar putea fi:

- Utilizarea unui algoritm eficient de generare a întrebărilor de test: Integrarea unui algoritm specializat pentru generarea întrebărilor de test, care să asigure diversitate și relevanță în selecția întrebărilor. Acest lucru ar contribui la îmbunătățirea calității testelor și la creșterea gradului de acuratețe în evaluarea cunoștințelor utilizatorilor.
- Optimizarea interacțiunii cu baza de date: Utilizarea unor tehnici și strategii eficiente pentru a accelera interogările și manipularea datelor în baza de date. Aceasta poate include indexarea corectă a tabelelor, utilizarea cache-ului de date sau optimizarea interogărilor SQL. În plus, ar putea fi luate în considerare metode de denormalizare a datelor pentru a reduce complexitatea interogărilor și timpul de răspuns.