



Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare

Proiect - Ingineria Programării

Quiz Game

Documentație

Proiect realizat de:

Dumitrașcu Dragoș – Teodor

Hodea Paul

Marian Bogdan Costel

Marin Vlad

Grupa 1308A



Documentul specificațiilor cerințelor software (Software Requirements Specification)

Pentru

Quiz Game

Versiunea 1.0

Autori:

Dumitrașcu Dragoș – Teodor

Hodea Paul - Emanuel

Marian Bogdan - Costel

Marin Vlad

26 mai 2022



Curpins

1. Introducere

- 1.1 Scop
- 1.2 Convenții ale documentului
- 1.3 Public vizat
- 1.4 Conținutul documentului
- 1.5 Domeniul de aplicare
- 1.6 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

3. Cerințe specifice

- 3.1 Interfața cu utilizatorul
- 3.2 Interfața hardware
- 3.3 Interfața software

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

Quiz Game este un test de cultură generală pe care oamenii îl pot susține oricând doresc să își testeze nivelul de cunoștințe. Aplicația pune la dispoziție întrebări în funcție de nivelul de dificultate ales de utilizator. 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, iar utilizatorul are posibilitatea de a vedea și întrebările la care a greșit.

Documentul de specificare a cerințelor software prezent furnizează o descriere completă a proiectului intitulat „**Quiz Game**” 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

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.

1.4. Conținutul documentului

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).



1.5. Domeniul de aplicare

Quiz Game 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.

1.6. Referințe

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

2. Descriere generală a proiectului

2.1. Perspectiva proiectului

Quiz Game este un test de cultură generală dezvoltat în IDE-ul Visual Studio 2019. Obiectivul testului este de a testa cunoștințele utilizatorului.

Quiz Game 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 Quiz Game trebuie să le îndeplinească pentru utilizatorul său final sunt următoarele:

- Alegerea dificultății – Quiz Game permite utilizatorului să poată alege nivelul de dificultate pe care îl dorește pentru susținerea testului
- Alegerea rezoluției – Quiz Game suportă alegerea rezoluției în funcție de preferințele utilizatorului
- Vizualizarea răspunsurilor – Quiz Game permite vizualizarea răspunsurilor greșite la finalul susținerii testului

2.3. Caracteristicile utilizatorului

Quiz Game ar trebui să fie conceput 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 nivelul de dificultate pe care îl va alege. Acest test poate fi și solicitant pentru acei utilizatori care doresc să aibă parte de întrebări foarte dificile.



2.4. Mediul de operare

Din moment ce Quiz Game este implementat 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

Din cauză că Quiz Game este proiectat și implementat într-un singur semestru ca și proiect pentru Facultatea de Automatică și Calculatoare, în cadrul materiei Ingineria Programării, este posibil ca timpul să fie cel mai limitativ factor în ciclul de dezvoltare. Precum și constrângerea tehnologică de a dezvolta totul pentru platforma .NET, ceea ce va fi amplificat de lipsa noastră de experiență cu limbajul C# și neexperiența în proiectarea de software și documentație.

Nu există alți factori, deoarece nu există niciun cost în dezvoltarea acestui proiect.

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 5 butoane cu care se poate interacționa (Play, Difficulty, Settings, Help, Exit)
2. Prin apăsarea butonul Settings se va deschide o fereastră care va permite alegerea rezoluției.
3. Prin apăsarea butonului Difficulty se va deschide o fereastră din care se poate alege nivelul de dificultate al întrebărilor: începător, mediu și avansat.
4. Prin apăsarea butonul Help se va deschide o fereastră ce conține instrucțiuni pentru utilizator.
5. Prin apăsarea butonului Play se va închide fereastra principală și se va deschide una nouă care conține întrebarea și cele 4 variante de răspuns. În această fereastră există 3 butoane: două care permit trecerea la următoarea întrebare sau întoarcerea la una din întrebări; iar un buton de finalizare a testului care devine valabil atunci când se răspunde la toate întrebările.



6. Prin apăsarea butonului Finalizează test se va deschide o nouă fereastră în care va fi afișat punctajul obținut. În această fereastră există și un buton de afișare a răspunsurilor greșite, iar prin apăsarea acestuia vor fi afișate întrebările la care s-a răspuns greșit, unde cu roșu va fi varianta aleasă de utilizator, iar cu verde răspunsul corect.

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.

3.3. Interfața software

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

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.

Cum să folosești aplicația

Începerea testului

Înainte de a începe testul este indicat ca utilizatorul să aleagă dificultatea dorită. Pentru a realiza această acțiune se apasă click stânga pe butonul Difficulty. După care, se poate apăsa pe butonul de Play pentru începerea testului.

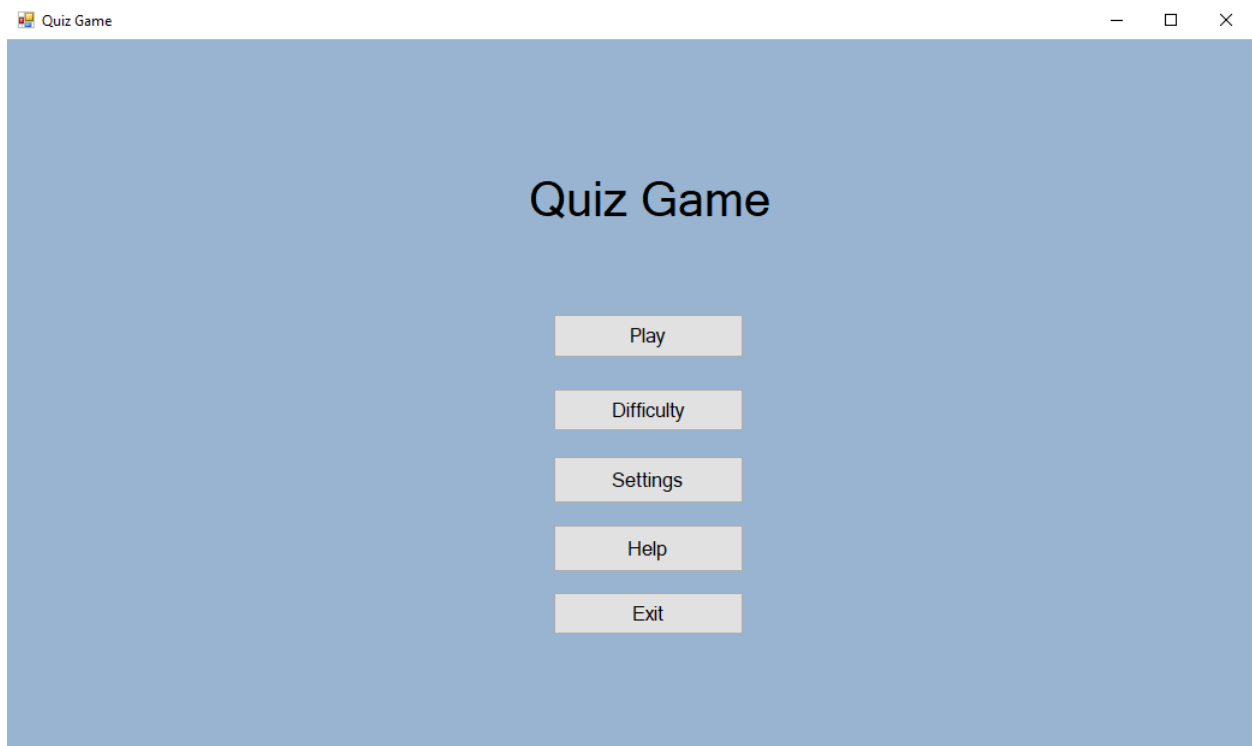


Figura 1: Start test

Parcurgerea testului

După începerea testului se va afișa câte o întrebare în fereastră. Fiecare întrebare conține 4 variante de răspuns, și doar una este corectă. Pentru a putea trece la următoarea întrebare trebuie ales obligatoriu un răspuns la întrebarea curentă. După alegerea unui răspuns, se poate trece la întrebarea următoare prin apăsarea butonul „Întrebarea următoare”. Dacă se dorește întoarcerea la una din întrebările precedente se poate apăsa butonul „Întrebarea precedentă”.

De asemenea, există în partea de jos un progress bar care arată cât s-a parcurs din test.



Carui presedinte i-a fost adresat cantecele "Happy Birthday to you Mr. President", cantat de catre Marilyn Monroe?

☐ Richard Nixon

☐ Gerald Ford

☐ John F Kennedy

☐ Dwight Eisenhower

Intrebarea precedenta Finalizeaza quiz Intrebarea urmatoare

Progres

Figura 2: Parcurgerea testului

Finalul testului

După ce s-a ales câte un răspuns la fiecare întrebare se poate vedea rezultatul final prin apăsarea butonul „Finalizează quiz”.

Punctaj final: 7 / 10

Afiseaza raspunsurile gresite

Figura 3: Punctaj final

Prin apăsarea butonului „Afișează răspunsurile greșite” se pot vedea întrebările la care s-a răspuns greșit. La fiecare întrebare greșită este afișat cu roșu varianta greșită aleasă de utilizator, iar cu verde este răspunsul corect.

Punctaj final: 7 / 10

Care dintre următoarele animale își trage numele din greaca veche însemnând "cal de rau"

~~Rechinul~~
Testoasa
Hipopotamul
Broasca

Care sunt grupele principale de sange?

~~A,B,C & D~~
~~A,AB,CD & O~~
A,B,AB, & O
B,C,D & O

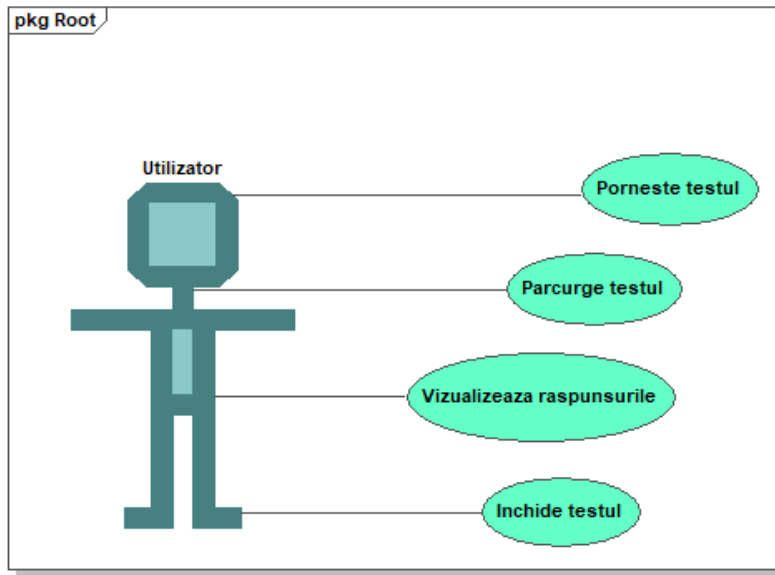
Cine a produs primul bec comercial în 1879?

~~Alexander Graham Bell~~
Guglielmo Marconi
Nikolai Tesla
Thomas Edison

Figura 4: Răspunsuri greșite

Diagrame UML

Cazuri de utilizare

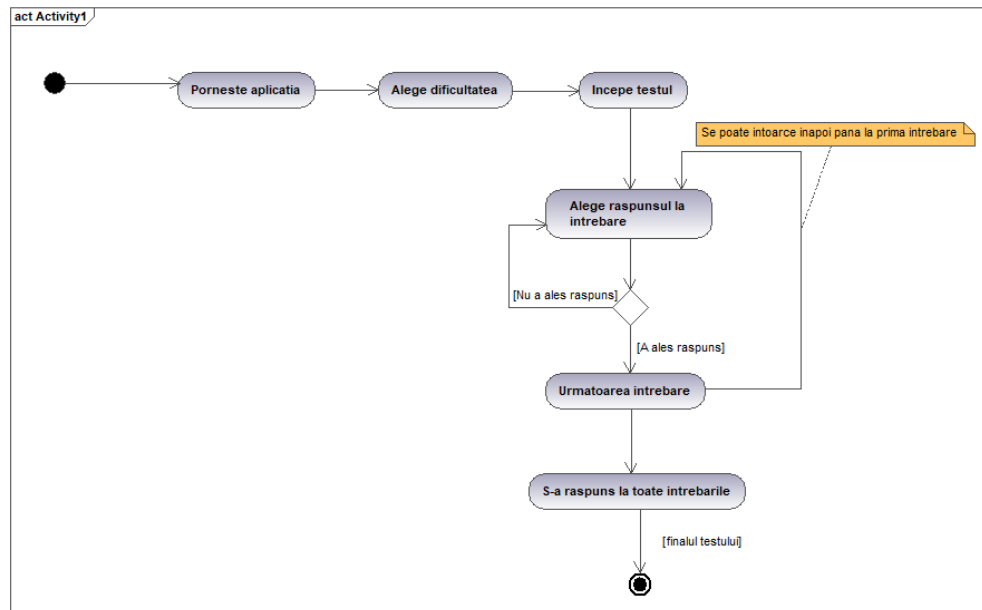


Generated by UModel

www.altova.com

Figura 5

Diagrama de activități



Generated by UModel

www.altova.com

Figura 6

Diagrama de secvențe

Quiz Game - Sequence Diagram

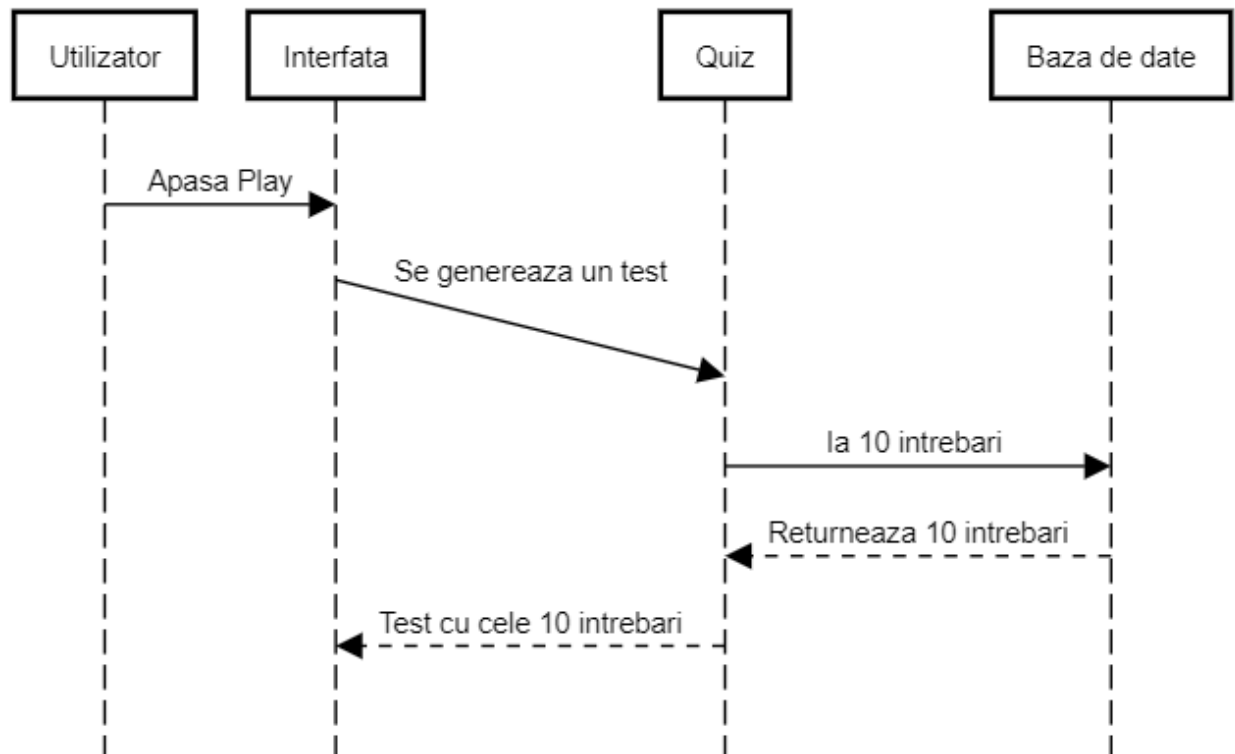


Figura 7

Diagrama de clase

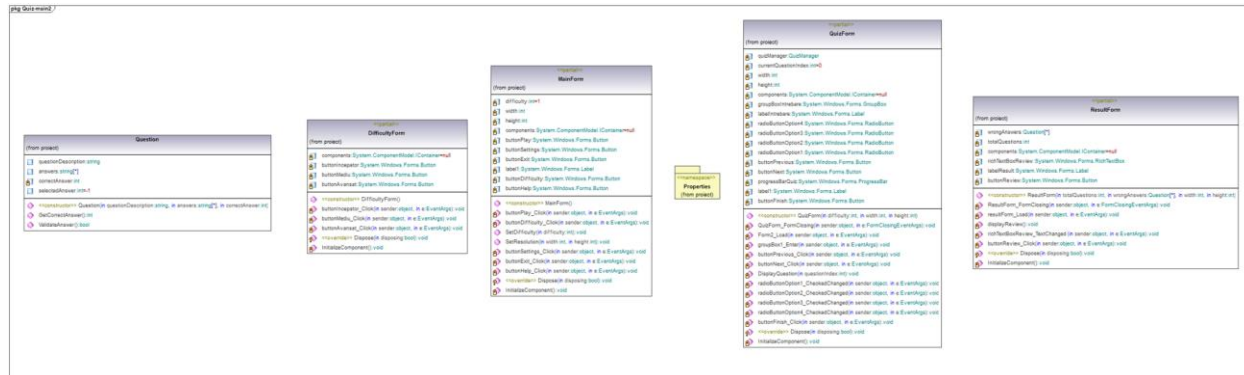


Figura 8

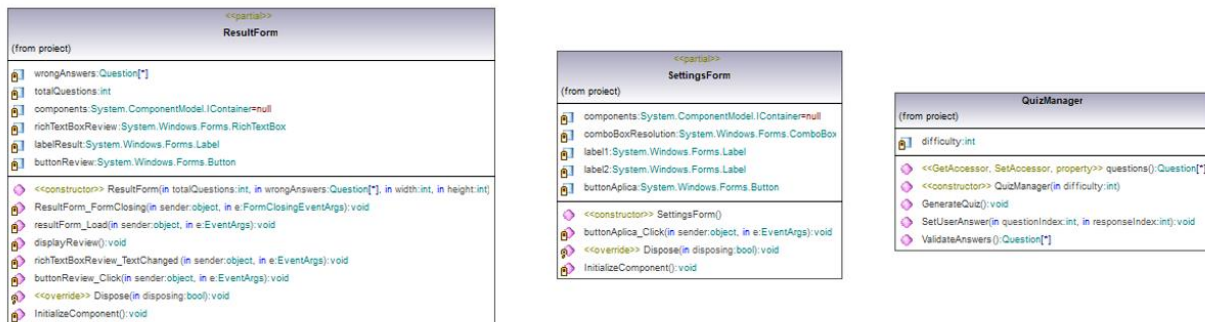


Figura 9

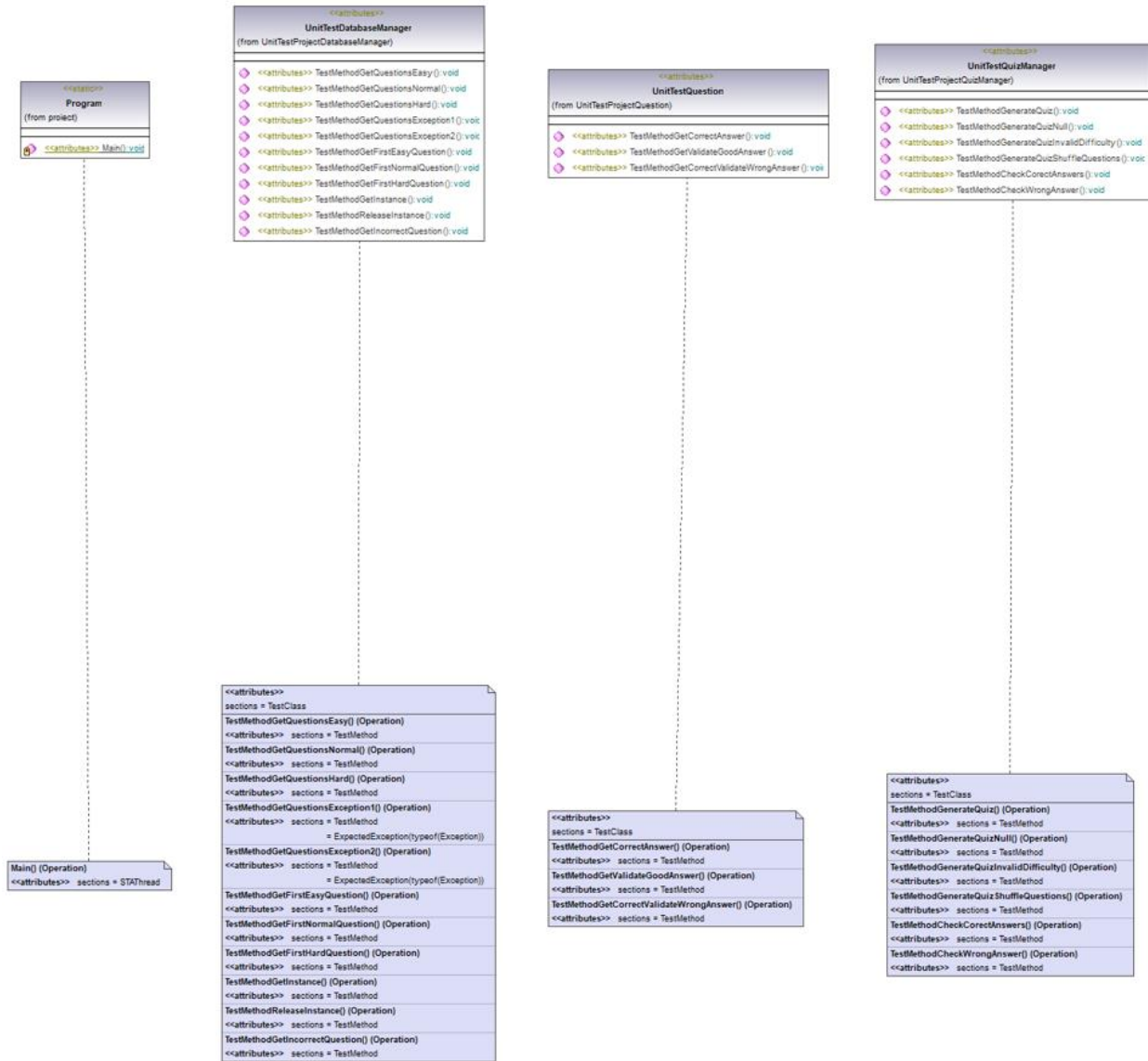


Figura 10



Exemple din codul sursă

Instanța bazei de date

```
/// <summary>
/// Metoda GetInstance apeleaza constructorul privat, si ne asigura ca putem avea
o singura instanta din aceasta clasa
/// </summary>
/// <returns>Instanta curenta din DatabaseManager, marcata cu keyword-ul
static</returns>
public static DatabaseManager GetInstance()
{
    lock (_lock)
    {
        if (dbInstance == null)
        {
            dbInstance = new DatabaseManager();
        }
        return dbInstance;
    }
}
```

Generarea testului cu cele 10 întrebări

```
/// <summary>
/// Metoda preia toate intrebarile din baza de date si selecteaza 10 dintre ele
in ordine aleatoare
/// </summary>
public void GenerateQuiz()
{
    try
    {
        Question[] allQuestions =
DatabaseManager.GetInstance().GetQuestions(this.difficulty);
        Random rand = new Random();
        allQuestions = allQuestions.OrderBy(x => rand.Next()).ToArray();
        this.questions = allQuestions.Take(10).ToArray();
    }
    catch (Exception e)
    {
        MessageBox.Show("Eroare preluare intrebari din baza de date: " +
e.Message);
    }
}
```



Extragerea de întrebări din baza de date în funcție de dificultate

```
/// <summary>
    /// Metoda GetQuestions face un query in baza de date prin care returneaza lista
    de intrebari de o anumita dificultate
    /// </summary>
    /// <param name="difficulty"></param>
    /// <returns>Lista de intrebari</returns>
    public Question[] GetQuestions(int difficulty)
    {
        if (difficulty > 2 || difficulty < 0)
            throw new Exception("Nivel de dificultate invalid");

        List<Question> questions = new List<Question>();

        SqlCommand cmd = new SqlCommand("select intrebare, raspuns1, raspuns2,
raspuns3, raspuns4, raspuns_corect from dbo.Intrebari where dificultate = '" + difficulty
+ "'", cnn);
        SqlDataReader reader = cmd.ExecuteReader();
        try
        {
            while (reader.Read())
            {
                questions.Add(new Question(reader["intrebare"].ToString(),
                    new[] { reader["raspuns1"].ToString(),
reader["raspuns2"].ToString(),
                    reader["raspuns3"].ToString(), reader["raspuns4"].ToString() },
                    int.Parse(reader["raspuns_corect"].ToString())));
            }
        }
        finally
        {
            reader.Close();
        }
        return questions.ToArray();
    }
}
```

Pentru a vedea mai multe despre proiect, dar și documentație puteți intra pe urmatorul link:

<https://github.com/phodea/Quiz>