



**UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" IAȘI**

**FACULTATEA AUTOMATICĂ ȘI CALCULATOARE**

**SPECIALIZAREA CALCULATOAREA ȘI TEHNOLOGIA  
INFORMAȚIEI**

**DISCIPLINA INGINERIA PROGRAMARII**



# **Aplicație de generare teste grilă**

***Stundetii:***

**COSMA PETRONELA**

**CRACEA PETRONELA-LUIZA**

**BUJOR ALEXANDRU-IONUT**

**GHIRVU ALEXANDRU-MADALIN**

**grupa 1305B**

## Cuprins

I. Software Requirements Specification(SRS) .....	3
1. Introducere .....	3
1.1. Obiectivul .....	3
1.2. Convențiile documentului .....	3
1.3. Audiența țintă/destinată.....	3
1.4. Scopul produsului .....	4
2. Descrieri generale .....	4
2.1. Perspectiva produsului .....	4
2.2. Funcțiile produsului .....	4
2.3. Clase de utilizatori și caracteristici .....	4
2.4. Mediu de operare .....	4
2.5. Constrângeri de desing și implementare .....	5
2.6. Documentația utilizatorului .....	5
2.7. Ipoteze și Dependente .....	5
3. Cerinte externe pentru interfață .....	5
3.1. Interfața utilizatorului .....	5
3.2. Interfața hardware .....	5
3.3. Interfața software .....	5
3.4. Interfața comunicației .....	5
4 . Caracteristicile sistemului .....	5
5. Alte cerințe nonfunctionale .....	6
5.1. Cerințe pentru performanță.....	6
5.2. Cerințe pentru siguranță .....	6
5.3. Cerințe pentru securitate.....	6
5.4. Atributele calității software-ului.....	6
5.5. Reguli de business .....	6
6. Alte cerințe .....	6
II. DIAGramele UML .....	7
III. Modul de utilizare a programului .....	10
IV. Capturi ecran din timpul executiei .....	10
V. Anexă .....	12

# **I. Software Requirements Specification(SRS)**

## **1. Introducere**

### **1.1. Obiectivul**

Aplicația de generare teste grilă (din diferite domenii) este un joc de cultură generală în care îți poți pune la încercare cunoștințele în domeniile propuse.

Jocul este simplu, de tip one player, care funcționează fără conexiune la internet, ce presupune la început alegerea domeniului din care acesta dorește să primească întrebări random. Testul poate fi oprit oricând prin apăsarea butonului STOP, moment în care veți primi pe loc rezultatul. Aplicația funcționează atâta timp cât detaliile și specificările funcționalității jocului sunt definite în secțiunile 4 și 5. În secțiunea 2 găsiți o prezentare generală iar în secțiunea 5 o lista de cerințe.

### **1.2. Convențiile documentului**

Acest document urmează formatul standard IEEE pentru dezvoltare software. Standardele definesc un format regulat pe care acest document le urmărește, incluzând să fie scris la persoana a 3-a, cu voce pasivă, citibil și cu un text gramatical corect.

### **1.3. Audiența țintă/destinată**

Documentul nu e dedicat end user-ilor deoarece conține specificații detaliate despre cum este implementată software aplicația. Atâta timp cât user-ul are nevoie de informații despre cum funcționează aplicația, în loc de cum e concepută, acest document este mai mult orientat către cei care testează și către majoritatea dezvoltatorilor de jocuri.

Începutul documentului conține o prezentare generală a funcțiilor și specificațiilor jocului în secțiunea 2, apoi descrie cererile pentru interfațare cu o secțiune externă hardware și software în secțiunea 3. În secțiunea 4 se descriu funcțiile jocului în detalii iar în secțiunea 5 se listează diferitele cereri pe care jocul trebuie să le respecte după finalizare.

E recomandat că la început să se citească secțiunea 2 pentru a vă putea face o idee generală despre cerințele software. Cei care testează trebuie să citească apoi secțiunile 5.1 cât și 5.4, pentru a-și face o idee cum va afecta acest joc atât pe ei cât și sistemul pe care rulează. Următoarea secțiune de citit este 3.1 urmată de secțiunea 4. Citind documentul în această ordine, va oferi tester-ului o idee asupra a ceea ce trebuie să se aștepte la prima vedere a interfeței.

După citirea celei de-a 2 secțiuni, dezvoltatorii trebuie să citească celelalte secțiuni rămase în ordine pentru că acest document a fost creat cu scopul de a dezvolta un joc. Dezvoltatorii trebuie să aibă o vedere de ansamblu pe care o găsesc în secțiunea 2 a jocului. Apoi, cum se interfațează în secțiunea 3. Secțiunea 4 e cea mai importantă pentru dezvoltatori pentru că descrie toate funcțiile jocului în detalii și îi ajută pe aceștia să ia decizii în scrierea codului pentru aplicație. Secțiunea 5 e considerată mai puțin importantă, dar dezvoltatorii trebuie să o citească că să fie siguri că jocul lor respectă idealurile.

## **1.4. Scopul produsului**

Aplicația dezvoltată e un joc de cultură generală dezvoltată pe o platformă Visual Studio. Această are o interfață simplă, ușor de utilizat, și e de tip single player. Obiectivul jocului este de a răspunde corect la cât mai multe întrebări. Controlul asupra opririi testului în orice moment aparține jucătorului.

Este dezvoltat în așa fel încât să ruleze pe orice computer care poate folosi VS și poate fi ușor rulat pe un computer cu o putere medie de procesare.

## **2. Descrieri generale**

### **2.1. Perspectiva produsului**

Aplicația dezvoltată este un joc de cultură generală. Este o aplicație de tip Window Form (.NET Framework) în mediul de dezvoltare Visual Studio. Organizația jocului va porni de la clasă de bază Form1 care este derivată din clasă Form, clasă ce se ocupă cu inițializarea obiectelor de pe interfață grafică și crearea callback-urilor pentru butoane. Clasa Test Factory se ocupă cu inițializarea obiectelor de tip test. Utilizatorul selectează ce tip de test vrea să dea iar metoda start din clasa Test Factory returnează acel test. Clasa Electronics, clasa Data Collection și ComputerScience sunt derivate din clasa abstractă test. În aceste clasă se deschide fisierul în care se află întrebările și răspunsurile pentru fiecare întrebare și sunt salvate în variabilele corespunzătoare.

### **2.2. Funcțiile produsului**

Funcțiile majore pe care Aplicația trebuie să le efectueze pentru end user-ii sunt :

- putere mediede procesare pentru PC - Aplicația trebuie să fie portabila
- jucători inteligenți - Aplicația trebuie să fie atrăgătoare, să poată fi atât pe placul persoanelor din domeniu cât și celor care nu sunt
- selecție unitară - Aplicația trebuie să permită selecția unităților din joc pentru a vedea infomatia atâta timp cât îndeplinește funcțiile bazate pe input

### **2.3. Clase de utilizatori și caracteristici**

Aplicația este concepută pentru orice tip de ușor, astfel orice clasă de ușor care acceptă ca timpul liber să fie petrecut jucându-te. Interfața și opțiunile trebuie să fie simple, descriptive și ușor de parcurs, de navigat. Simplitatea interfeței acestei aplicații e accesibilă oricărui tip de jucător.

### **2.4. Mediu de operare**

Având în vedere că aplicația este implementată în VS, această este multiplatforma, deci portabila și rulabila pe orice sistem de operare. Utilizatori de PC, MAC sau sisteme de operare Unix, pe care nu ar trebui să existe probleme în timpul rulării software-ului. A fost dezvoltată în Visual Studio 2017 versiunea C# 7.3 .NET core 2.1 SDK, astfel cei care folosesc aplicația trebuie să aibă o versiune compatibilă instalată pe sistemul lor, ce poate fi descărcată gratis în variantă comunității.

## **2.5. Constrângeri de desing și implementare**

Câtă vreme aplicația prezentată a fost concepută so implementată într-un singur semestru că și proiect pentru Facultatea de Automatică și Calculatoare,de o echipă software , e posibil că timpul să fie cel mai limitat factor în ciclul de dezvoltare.

## **2.6. Documentația utilizatorului**

În acest document veți găsi toate informațiile necesare unui uer: regulile jocului,cât și un tutorial despre rularea aplicațiilor dezvoltate în VS pe diferite platforme.

## **2.7. Ipoteze și Dependente**

Ne asumăm că VS va fi capabil să facă față update-urilor ce vor fi cerute. VS fva suporta toate acestea dar suplimentar testarea trebuie efectuată pentru a determina dacă suntem sau nu capabili să producem complexitatea pe care ne-o propunem.

# **3. Cerinte externe pentru interfață**

## **3.1. Interfața utilizatorului**

Interfața aplicației de teste grilă va apărea ca un chenar,cu o structură simplă,ușor de înțeles,cu butoane clare și concise. Întrebările vor fi distribuite random inr-un textbox de tip button. Utilizatorul va putea vedea foarte clar întrebările,va putea alege domeniul cu ajutorul unui comboBox, va putea opri testul oricând dorește cu ajutorul unui buton de stop,sau poate începe testul cu ajutorul butonului de start. Butonul Exit ,așa cum îi spune și numele ajută utilizatorul să iasă din aplicație.

## **3.2. Interfața hardware**

Interfața hadrware va include un mouse-o tastatură și un monitor pe care să ruleze interfată. Toate acestea sunt suficiente pentru a rulă această aplicație, iar butonul stânga al mouse-ului activează aceste butoane ale interfetei. Cu tastatură nu poți decât să alegi din opțiunile prezentate pentru domeniul din care să fie alese întrebările.

## **3.3. Interfața software**

Pentru a putea rula jocul, user-ul are nevoie de Visual Studio,versiunea 2015 până la versiunile actuale (2019) instalat pe computer. Este nevoie de acest program deoarece aplicația este scrisă în C#. Dacă user-ul nu are acest program,îl poate descărca de la adresa :<https://visualstudio.microsoft.com/vs/community/>.

## **3.4. Interfața comunicației**

Pentru a putea porni aplicația ,user nu are nevoie de o conexiune la internet. Trebuie doar să aibă un sistem de operare și aplicația necesară rulării, apoi se poate bucura de joc.

# **4 . Caracteristicile sistemului**

Caracteristicile sistemului este acoperit detaliat în apendice separate pentru scenariii și diagrame de clase.

## **5. Alte cerințe nonfunctionale**

### **5.1. Cerințe pentru performanță**

Aplicația trebuie să fie capabilă să ruleze pe un minim de Windows 7 ( cel folosit de noi este 10) ,Linux, MAC. Este recomandată să aveți pe puțin 256 Mb RAM. Pentru grafica interfeței avem nevoie de minim 20 Kb .

### **5.2. Cerințe pentru siguranță**

Ca pentru fiecare joc pe calculator, există un risc de epilepsie. Dacă vă știți cu asemenea probleme ,vorbiți cu doctorul înainte de a vă juca.

### **5.3. Cerințe pentru securitate**

Această aplicație nu va aduna informații personale de la persoanele care-l vor rula. Singura informație pe care o va obține va fi versiunea programului.

### **5.4. Atributele calității software-ului**

Software-ul trebuie să fie robust și să poată asigura utilizatorilor o experiență pozitivă. Aplicația trebuie să fie ușoară pentru un începător ,de deschis și de jucat. De asemenea trebuie să fie destul de flexibil,și nu în ultimul rând,cel mai important ,portabil pe sistemele specificate la secțiunea 5.1;

### **5.5. Reguli de business**

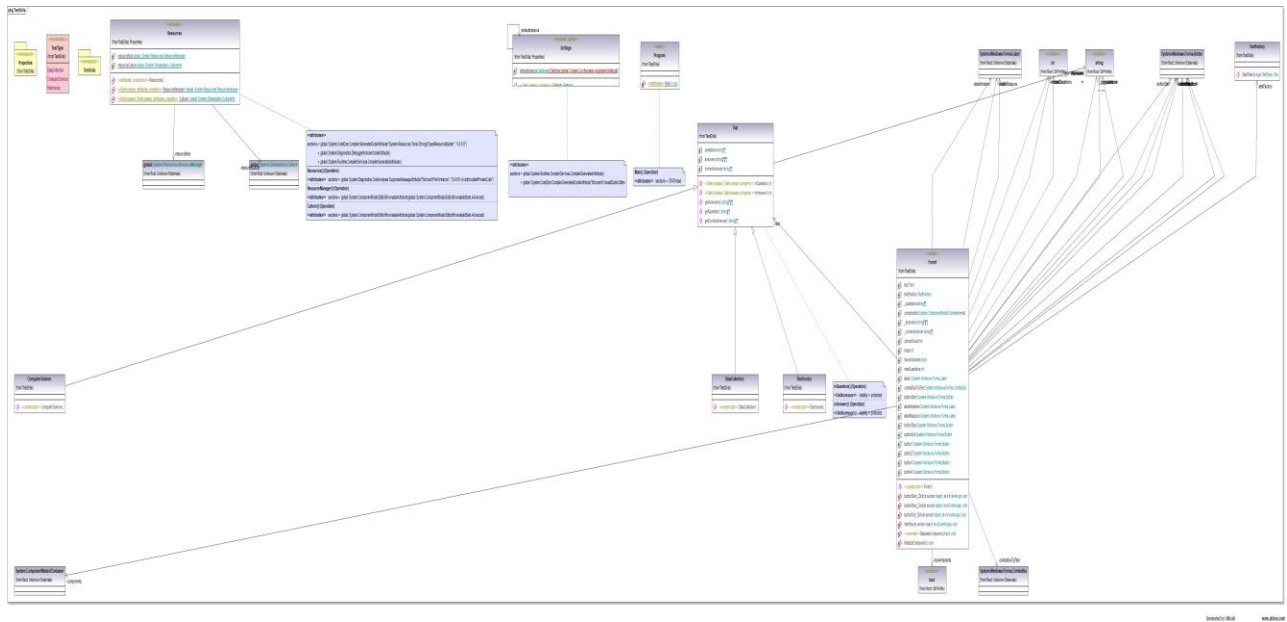
Este politica echipei de dezvoltare a aplicației să urmească codurile de conduită stabilite de către Universitatea Tehnică Gheorghe Asachi.

## **6. Alte cerințe**

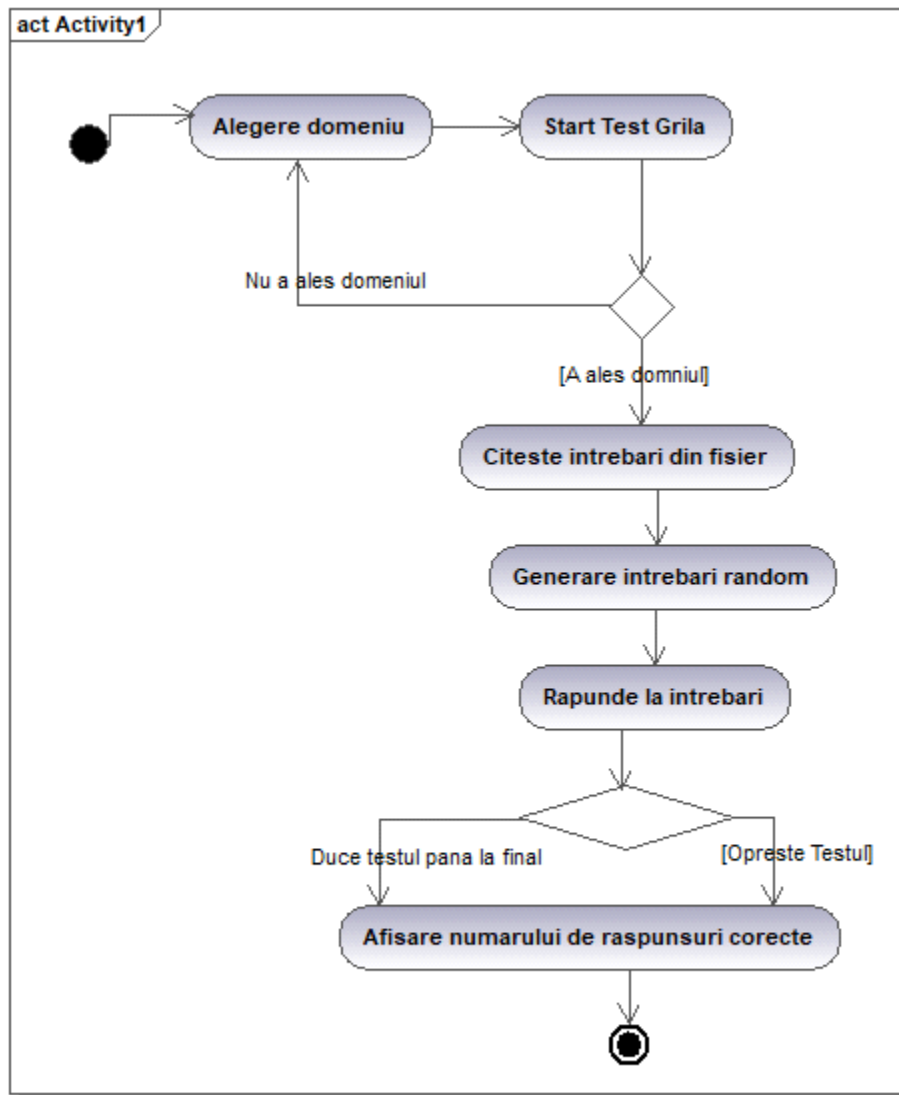
Clasa de diagrame, de activități, cazuri de utilizare și cea de secvențe va fi prezentată mai jos .

## II. DIAGRAMELE UML

- diagrama de clase ( obținuta cu ajutorul programului Altova UModel)



- diagrama de activități

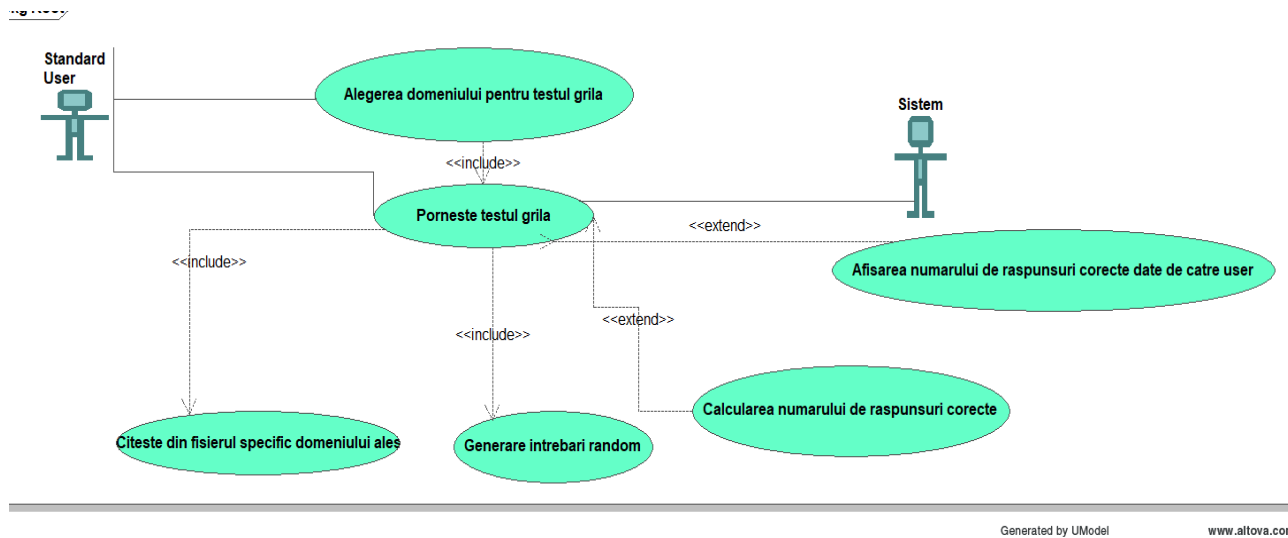


Generated by UModel

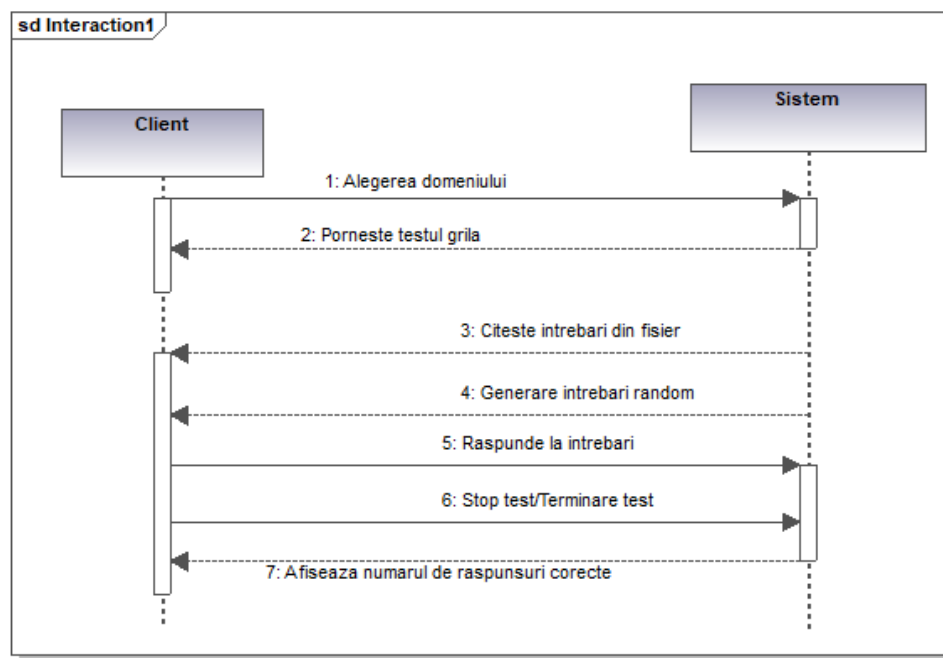
[www.altova.com](http://www.altova.com)

- diagrama cazuri de utilizare





- diagrama de secvențe



Generated by UModel

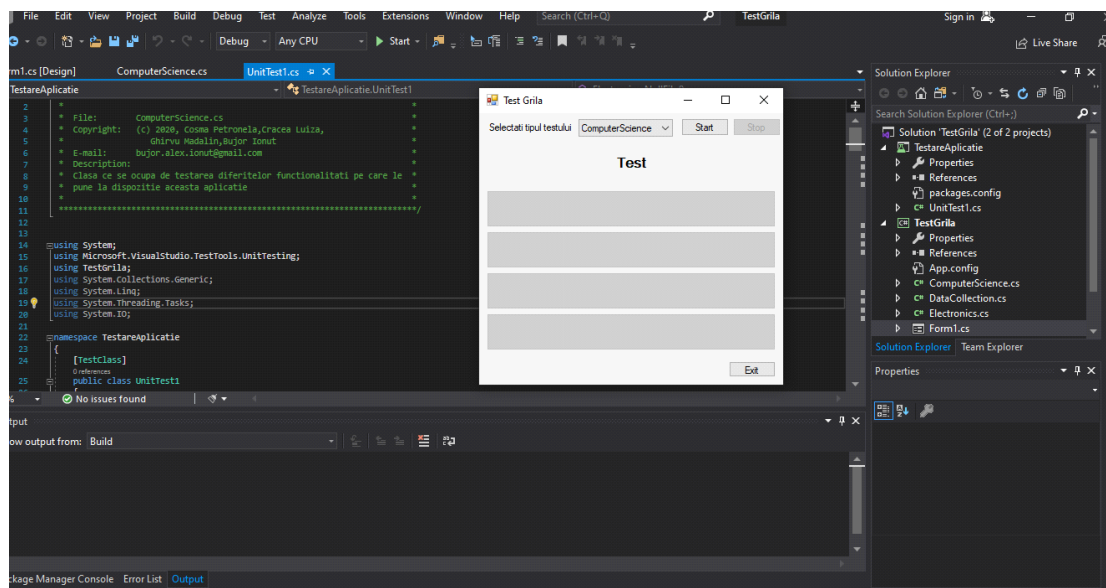
[www.altova.com](http://www.altova.com)

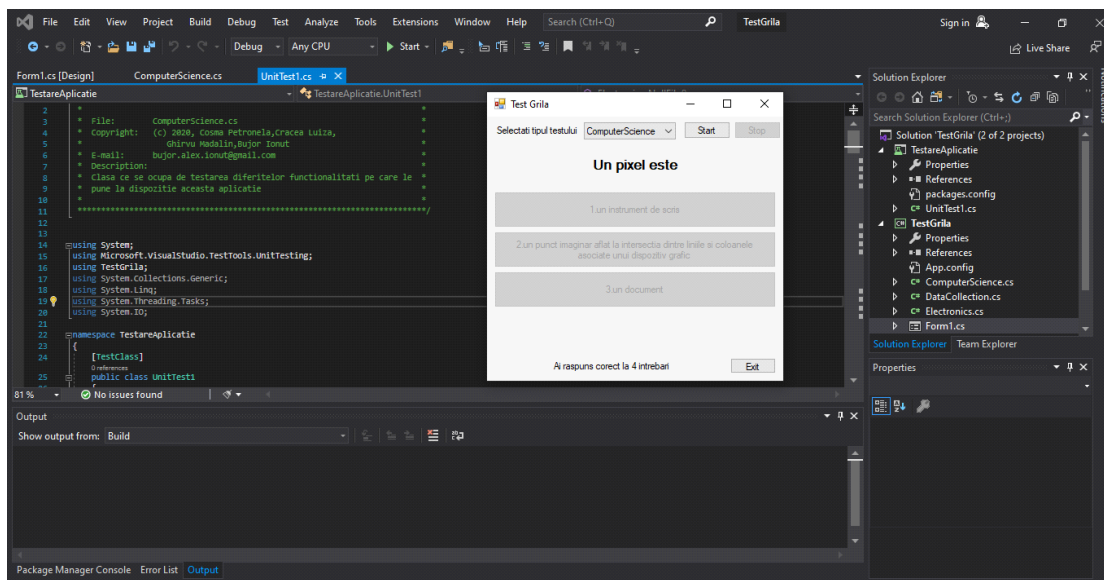
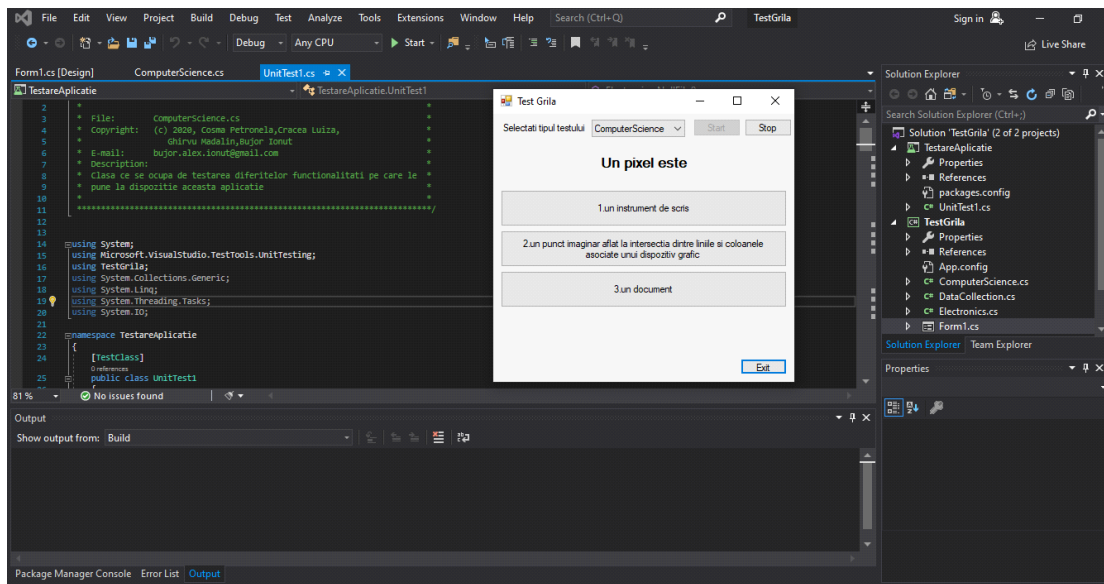
### III. Modul de utilizare a programului

Conform help-ului pus la dispoziție utilizatorilor, aplicația realizează generarea unor teste grilă cu 3 /4 variante. Corectarea testului se realizează fie la finalul pargurgerii, fie în momentul în care utilizatorul decide să-l oprească, afișându-se astfel numărul de întrebări la care ați răspuns corect.

Așa cum avem explicat și în help, alegerea domeniului de face prin apăsarea butonului pe care va apărea inițial scris "Data Colection". Testul începe la apăsarea butonului START. Alegerea răspunsului se face prin apăsarea click pe varianta pe care o considerați corectă ,din toate cele 4 variante. Apăsând butonul Stop putem opri testul. Cu ajutorul butonului EXIT putem ieși din aplicație.

### IV. Capturi ecran din timpul executiei





## V. Anexă

Proiectul este structurat în două module:

- Modulul de test în care se afla clasa **UnitTest1** care efectuează teste asupra aplicației
- Modulul TestGrila în care se afla clasele pentru creerea testelor și a interfeței:
  - Clasa **Test**- clasa de baza pentru construirea testului, ea este extensa de mai multe clase
  - Clasa **ComputerSciens** extinde clasa **Test**- Aceasta clasa creaza un test care are ca subiect informatica;
  - Clasa **DataCollection** extinde clasa **Test** - Aceasta clasa creaza un test care are ca subiect colectiile de date;
  - Clasa **Electronics** extinde clasa **Test** - Aceasta clasa creaza un test care are ca subiect electronica;
  - **TestFactory**- clasa care are ca scop determinarea tipului de test, ea conine un element de tipul **Test**;
  - **Form1**-clasa cu ajutorul careia am creat interfata cu utilizatorul;

Clasa **Test** este alcatuita din mai multe proprietati, questions (retine intrebarile), answers (retine raspunsurile), correctAnswer (retine raspunul corect), si metode, nr Question { get; protected set; }(retine numarul de intrebari) si nrAnswer { get; protected set; } (reine numarul de raspunsuri).

Clasa **ComputerSciens** contine un constructor in interiorul caruia se realizeaza operatiile de deschidere a fisierului afferent subiectului precum si salvarea intrebarile in variabila question si a raspunsurilor in variabila answers, urmand ca raspunsurile corecte sa fie stocate in variabila correctAnswer.

Clasa **DataCollection** are o functionalitate identica cu cea a clasei **ComputerSciens** ea continand tot un constructor cu aceasi rol.

Clasa **Electronics** la fel ca si clasele **ComputerSciens** si **DataCollection** contine aceasi structura si functionalitate.

Clasa **Form1** cotine numerosase metode fiecare cu rolul de a crea functionalitate butoanelor existente in interfata cu utilizatorul.

- a) Prima metoda din aceasta clasa este constructorul unde se initializeaza toate elementele interfetei.
- b) A doua metoda este buttonStart Click(object sender, EventArgs e), care are ca rol crearea functiei de callback pentru butonul "Start", preluarea tipul de test de la **coboBoxTipTest**, unde se creaza un nou test folosint variabila **testFactory** si generarea uni anumit numar de butoane in funcie de cate raspunsuri au fiecare intrebare.
- c) Cea de a treia metoda este buttonStop Click(object sender, EventArgs e), care are ca rol crearea functiei de callback pentru butonul "Stop", dezactiveaza toate butoanele si calculeaza numarul de raspunsuri corecte pana in acel moment.
- d) A patra metoda utilizata in aceasta clasa **Form1** este buttonExit Click(object sender, EventArgs e) care nu face altceva decat sa inchida interfata cu utikizatorul la apasarea butonului "Exit".

- e) Ultima metoda din aceasta clasa este Verifica(object sender, EventArgs e), ea are ca rol clacularea numarului de raspunsuri corecte la test.

Clasa **TestFactory** contine o variabila `TestType` ce retine tipurile de teste care pot fi generate dar si o metoda StartTest(TestType type) de tipul `Test` in care instantiem un obiect de tipul `Test` in care vom retine tipul testului ales dupa care acesta va fi returnat.

