

PROIECT FINAL

FULGER IRINA

09.12.2023

Informații teoretice

Ce sunt cerințele de business, la ce ne folosesc și cine le creează

- Cerințele de business reprezintă descrierea nevoilor și așteptărilor clientului și utilizatorului pentru un produs/software.
- Aceste cerințe sunt create de client și ghidează dezvoltarea și testarea produsului.

Diferența dintre test condition și test case

Test condition reprezintă o descriere generală a unei situații ce trebuie verificată, iar **test case** este o specificație detaliată a pașilor și datelor de testare pentru a verifica o funcționalitate sau o caracteristică a produsului.

Informații teoretice

Etapele procesului de testare

Planificare	Stabilirea obiectivelor, alocarea rolurilor, definirea criteriilor de intrare și ieșire, identificarea riscurilor de proiect și a resurselor necesare, dezvoltarea planului de testare	Monitorizare și control (compararea progresului actual față de planul de testare pe tot parcursul etapelor procesului de testare)
Analiză	Analizarea documentației primite de la client și generarea condițiilor de testare	
Design	Crearea cazurilor de testare, stabilirea priorităților, identificarea datelor de testare de care este nevoie și designul mediului de testare	
Implementare	Crearea datelor de testare, crearea mediului de test, gruparea testelor pe baza obiectivelor, verificarea că există tot ce trebuie pentru a începe testarea propriu-zisă	
Execuție	Executarea cazurilor de testare, raportarea rezultatelor în tool-ul în care sunt scrise testele, raportarea bug-urilor, retestarea bug-urilor	
Închidere	Evaluarea criteriilor de ieșire, predarea și arhivarea materialelor de testare, generarea unui raport de închidere a testării, identificarea riscurilor de produs și raportarea lor către client	

Informații teoretice

Retesting vs Regression testing

Retestarea este un proces prin care se verifică dacă defectele marcate ca și remediate au fost într-adevăr remediate

Testarea de regresie verifică dacă schimbările aduse asupra software-ului nu au afectat funcționalitățile existente și că acestea continuă să funcționeze corect

Functional testing vs Non-functional testing

Testare funcțională verifică dacă produsul își îndeplinește funcțiile, arată ce trebuie să facă produsul

Testarea non-funcțională verifică attribute care descriu cât de bine își îndeplinește sistemul funcțiile

Blackbox testing vs Whitebox testing

Blackbox testing se concentrează pe testarea funcționalității fără a lua în considerare detaliile interne ale codului

Whitebox testing implică testarea specifică a structurii și a logicii codului sursă al aplicației

Informații teoretice

Tehnici de testare

Blackbox testing

Partiționare
echivalentă

Analiza
valorilor limită

Testarea
tranzițiilor de
stare

Whitebox testing

Statement
coverage

Decision
coverage

Experienced based testing

Testarea ad-
hoc

Testarea
exploratorie

Ghicirea
erorilor

Informații teoretice

Verification vs Validation

Verification este procesul de evaluare prin care se determină dacă specificațiile și cerințele stabilite sunt respectate

Validation este procesul de evaluare prin care verificăm că produsul răspunde nevoilor și așteptărilor utilizatorului.

Positive testing și Negative testing

Testare pozitivă înseamnă testarea sistemului cu valori pe care ar trebui să le poată procesa.
Exemplu: testarea opțiunii de dată de naștere cu o dată de naștere.

Testare Negativă înseamnă testare cu valori pe care sistemul nu ar trebui să le poată procesa în mod normal pentru a ne asigura ca aceste valori sunt într-adevăr respinse și că nu cauzează un crash al sistemului.
Exemplu: testarea opțiunii de dată de naștere cu o literă.

Informații teoretice

Niveluri de testare

Testarea unitară - reprezintă testarea fiecărei bucăți individuale de cod pentru a verifica dacă este pregătită pentru utilizare.

Testarea de Integrare – testează interacțiunile dintre componente și sisteme și există două tipuri de testare de integrare:

- ❑ Integrare între **componente** (când două sau mai multe module sunt legate între ele)
- ❑ Integrare între **sisteme** (când două sau mai multe sisteme sunt legate între ele)

Testarea de sistem - testează comportamentul și capacitatea sistemului ca un întreg.

Testarea de acceptanță - se concentrează pe comportamentul și capacitățile sistemului ca un tot unitar și are două etape

- ❑ Alfa testing – testarea se face la sediul companiei de către o echipă independentă de testare
- ❑ Beta testing – testarea aplicație live de către un anumit număr de utilizatori la sediul clientului

Aspecte practice – Crearea unei baze de date

- Am creat o bază de date cu elevii unei clase de a VIII-a

- ☐ Am folosit mai multe tipuri de variabile
- ☐ Am folosit un primary key cu auto increment
- ☐ Am adaugat fiecare copil cu opțiunea de insert

```
1 • create database scoala89;
2 • use scoala;
3
4 • create table copii8C(
5     id int not null primary key auto_increment,
6     child_surname Varchar (20),
7     child_name Varchar (20),
8     child_sex Varchar (6),
9     child_birth_year int,
10    child_home_address Varchar (50)
11 );
12
13 • insert into copii8C (child_surname, child_name, child_sex, child_birth_year, child_home_address) values
14 • insert into copii8C (child_surname, child_name, child_sex, child_birth_year, child_home_address) values
15 • insert into copii8C (child_surname, child_name, child_sex, child_birth_year, child_home_address) values
```

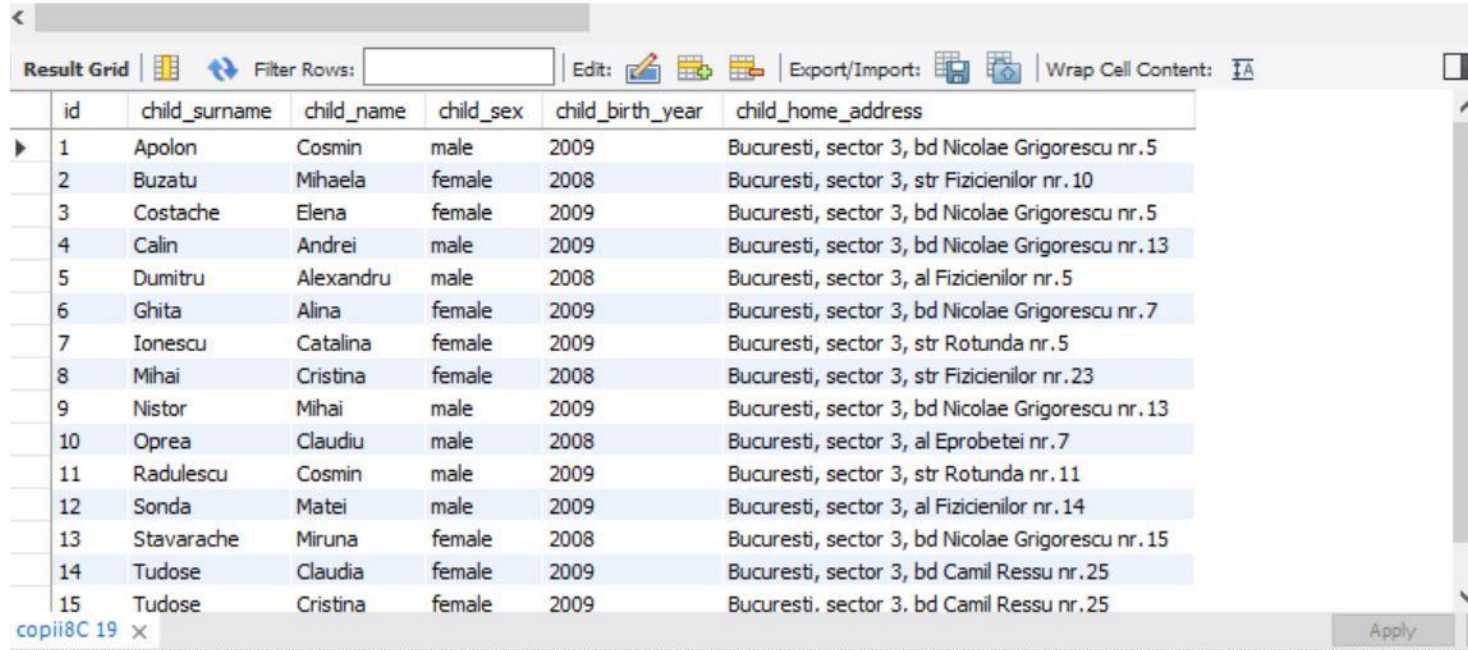

Aspecte practice – Folosirea unei baze de date

Tabelul final cu copiii clasei a 8-a C

28

29 • `select * from copii8C;`

30



The screenshot shows a database application interface. At the top, there's a toolbar with icons for grid view, filter, edit, export/import, and wrap cell content. Below the toolbar is a table with 6 columns: id, child_surname, child_name, child_sex, child_birth_year, and child_home_address. The table contains 15 rows of data. The bottom of the window shows a tab labeled 'copii8C 19' and an 'Apply' button.

	id	child_surname	child_name	child_sex	child_birth_year	child_home_address
▶	1	Apolon	Cosmin	male	2009	Bucuresti, sector 3, bd Nicolae Grigorescu nr. 5
	2	Buzatu	Mihaela	female	2008	Bucuresti, sector 3, str Fizicienilor nr. 10
	3	Costache	Elena	female	2009	Bucuresti, sector 3, bd Nicolae Grigorescu nr. 5
	4	Calin	Andrei	male	2009	Bucuresti, sector 3, bd Nicolae Grigorescu nr. 13
	5	Dumitru	Alexandru	male	2008	Bucuresti, sector 3, al Fizicienilor nr. 5
	6	Ghita	Alina	female	2009	Bucuresti, sector 3, bd Nicolae Grigorescu nr. 7
	7	Ionescu	Catalina	female	2009	Bucuresti, sector 3, str Rotunda nr. 5
	8	Mihai	Cristina	female	2008	Bucuresti, sector 3, str Fizicienilor nr. 23
	9	Nistor	Mihai	male	2009	Bucuresti, sector 3, bd Nicolae Grigorescu nr. 13
	10	Oprea	Claudiu	male	2008	Bucuresti, sector 3, al Eprobetei nr. 7
	11	Radulescu	Cosmin	male	2009	Bucuresti, sector 3, str Rotunda nr. 11
	12	Sonda	Matei	male	2009	Bucuresti, sector 3, al Fizicienilor nr. 14
	13	Stavarache	Miruna	female	2008	Bucuresti, sector 3, bd Nicolae Grigorescu nr. 15
	14	Tudose	Claudia	female	2009	Bucuresti, sector 3, bd Camil Ressu nr. 25
	15	Tudose	Cristina	female	2009	Bucuresti, sector 3, bd Camil Ressu nr. 25

Aspecte practice – Folosirea unei baze de date

Din tabelul denumit copii8C am selectat copii cu anul nașterii 2008 sau 2009 sau în funcție de sex

```
31 • select * from copii8C where child_sex='male';  
32 • select * from copii8C where child_sex='female';  
33 • select * from copii8C where child_birth_year='2008';  
34 • select * from copii8C where child_birth_year='2009';
```

```
33 • select * from copii8C where child_birth_year='2008';  
34 • select * from copii8C where child_birth_year='2009';  
35  
36 • create table note8C(
```

Result Grid Filter Rows: <input type="text"/> Edit: Export/Import: Wrap Cell Content:						
	id	child_surname	child_name	child_sex	child_birth_year	child_home_address
•	2	Buzatu	Mihaela	female	2008	Bucuresti, sector 3, str Fizicienilor nr. 10
	5	Dumitru	Alexandru	male	2008	Bucuresti, sector 3, al Fizicienilor nr. 5
	8	Mihai	Cristina	female	2008	Bucuresti, sector 3, str Fizicienilor nr. 23
	10	Oprea	Claudiu	male	2008	Bucuresti, sector 3, al Eprobetei nr. 7
	13	Stavarache	Miruna	female	2008	Bucuresti, sector 3, bd Nicolae Grigorescu nr. 15
•	NULL	NULL	NULL	NULL	NULL	NULL

Aspecte practice – Folosirea unei baze de date

Am creat un al doilea tabel, numit note8C, cu notele la limba română și la matematică

```
36 • ○ create table note8C(  
37     id int NOT NULL PRIMARY KEY auto_increment,  
38     child_surname Varchar (20),  
39     child_name Varchar (20),  
40     math_grade int,  
41     ro_grade int,  
42     situation Varchar(10)  
43 );  
44  
45 • insert into note8C (child_surname, child_name, math_grade, ro_grade) values ("Apolon", "Cosmin", 8, 9);  
46 • insert into note8C (child_surname, child_name, math_grade, ro_grade) values ("Buzatu", "Mihaela", 10, 10);  
47 • insert into note8C (child_surname, child_name, math_grade, ro_grade) values ("Costache", "Elena", 3, 7);  
48 • insert into note8C (child_surname, child_name, math_grade, ro_grade) values ("Calin", "Andrei", 5, 5);  
...
```




Aspecte practice – Folosirea unei baze de date

Am folosit un update pentru a actualiza coloana "situation" cu pass/failed în funcție de notele la limba română și matematică

```
61 • update note8C set situation='pass' where math_grade>=5 and ro_grade>=5;
```

```
--
```

```
63 • select * from note8C;
```

Result Grid						
Filter Rows: <input type="text"/>						
Edit:   						
	id	child_surname	child_name	math_grade	ro_grade	situation
▶	1	Apolon	Cosmin	8	9	pass
	2	Buzatu	Mihaela	10	10	pass
	3	Costache	Elena	3	7	failed
	4	Calin	Andrei	5	5	pass
	5	Dumitru	Alexandru	8	4	failed
	6	Ghita	Alina	5	6	pass
	7	Ionescu	Catalina	7	7	pass
	8	Mihai	Cristina	8	9	pass
	9	Nistor	Mihai	10	9	pass
	10	Oprea	Claudiu	9	9	pass
	11	Radulescu	Cosmin	6	8	pass
	12	Sonda	Matei	7	7	pass
	13	Stavarache	Miruna	2	4	failed
	14	Tudose	Claudia	7	9	pass
	15	Tudose	Cristina	9	7	pass

note8C 21

Aspecte practice – Folosirea unei baze de date

Am făcut mai multe încercări pentru a selecta anumite situații din baza de date, în funcție de doar nota la o materie sau la ambele materii

```
63 • select * from note8C;  
64 • select * from note8C where math_grade>=4;  
65 • select * from note8C where math_grade>=7;  
66 • select * from note8C where ro_grade>=7;  
67 • select * from note8C where ro_grade>=9 and math_grade>=9;  
68 • select * from note8C where ro_grade<5 and math_grade<5;  
69 • select * from note8C where ro_grade<5 or math_grade<5;
```

Aspecte practice – Folosirea unei baze de date

Scopul final era de a determina dacă putem afla situația de pass sau failed a unui elev în funcție de ambele note la limba română și la matematică

În condițiile în care elevul este "pass" cu minim nota 5 la ambele materii, situația de failed" s-a putut vedea cu filtrare cu OR. Astfel, o notă sub 5 la o materie înseamnă că situația elevului este "failed", chiar dacă la cealaltă materie are o notă mai mare de 5

```
69 • select * from note8C where ro_grade<5 or math_grade<5;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

	id	child_surname	child_name	math_grade	ro_grade	situation
▶	3	Costache	Elena	3	7	failed
	5	Dumitru	Alexandru	8	4	failed
	13	Stavarache	Miruna	2	4	failed
*	NULL	NULL	NULL	NULL	NULL	NULL

Proiectul poate fi găsit și aici: https://github.com/Fulger19/Proiect-examen-testare-manuala_SQL/tree/main

Vă mulțumesc pentru atenție!