

# Proiect final

Dima Ioana Nicoleta

04.09.2024

# Cuprins

## ❑ Partea I.

- Partea teoretica. Raspuns la intrebari.

## ❑ Partea II.

- Partea practica. Interogare baze de date. SQL
- Link catre Github
- [Nicoleta-24/proiect\\_testare \(github.com\)](https://github.com/Nicoleta-24/proiect_testare)

# Partea I

1. Explicati pe scurt ce sunt cerintele de business, la ce ne folosesc si cine le creeaza?

- Cerintele de business reprezinta documente care cuprind felul in care produsul trebuie sa functioneze conform cerintelor clientului, si folosesc la dezvoltarea si implementarea produsului, ele fiind create de echipa de business.

2. Explicati diferenta intre un test condition si un test case.

- Diferenta intre test condition si test case este aceea ca test condition reprezinta ceea ce testam si contine mai multe test case-uri, iar test case-ul se scrie in functie de cerintele din test condition.

3. Enumerati si explicati pe scurt etapele procesului de testare .

❑ Etapele procesului de testare sunt :

- etapa de planificare cuprinde ce parte a aplicației se dorește să fie testată;
  - se alocă roluri pentru persoanele care vor fi implicate în proiect;
  - se definesc criteriile de intrare și criteriile de ieșire;
  - se identifică riscurile de proiect inițiale și resursele necesare;
  - se creează un plan de testare care va conține informații generale legate de cum se va desfășura procesul de testare
  - se evaluează criteriile de intrare ( condiții care trebuie îndeplinite pentru a putea începe procesul de testare ).



- - etapa de analiza cuprinde : - Ce urmeaza sa se testeze;(test condition)
  - Se analizează documentația primită de la client (cerinte de bussines, specificații de design etc) pentru a ne asigura că le înțelegem, că nu exista greșeli, ambiguități, neconcordanțe, contradicții etc.
- - etapa de design:
  - Se creează cazurile de testare și se identifică datele de testare (informațiile pe care trebuie să le pregătim și care trebuie să fie supuse testării);test case-urile
- etapa de implementare :
  - Se creează datele de testare identificate în etapa anterioara;
  - Se validează mediul de test prin intermediul smoke testing (se face de regulă la începutul testării pentru a verifica dacă sistemul este suficient de stabil pentru a începe testarea);
  - Se prioritizeaza testele (pe baza importanței de business și a riscurilor) și se crează datele de testare;
  - Se grupează testele pe baza obiectivelor lor (testare funcțională, testare de regresie, testare de acceptanță etc)
  - Ne asigurăm că avem tot ce ne trebuie pentru a începe testarea propriu-zisă (mediu de testare, permisiuni, date de testare, documentație etc)
- etapa de executie :
  - cazurile de testare sunt executate (adică verificam comportamentul produsului software pe baza instrucțiunilor scrise în cazurile de testare);
  - rezultatele sunt raportate în tool-ul în care au fost scrise testele (Passed/Failed/Blocked etc);
  - bug-urile / Defectele / Fault-urile ,sunt raportate atunci când rezultatele așteptate nu coincid cu rezultatele actuale;
  - atunci când bug-urile sunt fixate, se face retestarea lor pentru a ne asigura că au fost într-adevăr fixate;
  - atunci când codul a fost schimbat (fie pentru fixarea unui bug fie pentru introducerea unei noi funcționalități), se va face si testare de regresie, pentru a ne asigura că schimbările făcute nu au avut un impact negativ asupra funcționalităților existente.

➤ etapa de inchidere :

- Se evaluează criteriile de ieșire pentru a ne asigura că putem să închidem procesul de testare în siguranță;
- Orice taskuri rămase deschise și buguri sunt reevaluate și ulterior închise;
- Materialele de testare sunt predate și arhivate ;
- Este generat un raport de închidere a testării;
- Se identifică riscurile de produs (dacă există) și se raportează către client.

➤ monitorizare si control, incepe odata cu etapa de planificare si se termina cu etapa de inchidere, este o activitate continuă care se desfășoară cu scopul de a compara progresul actual cu planul de testare (monitorizare), in cazul în care se observă riscul de a nu ne îndeplini obiectivele, se iau măsuri de control.

**4. Explicati diferenta intre retesting si regresion testing.**

**Retesting** se refera, la a retesta o functionalitate care anterior a fost evaluata ca fiind incorecta , pentru a ne asigura ca acum functioneaza corect, iar **testarea de regresie** se refera la a revalida functionalitatea care anterior a fost validata ca fiind corecta si in continuare functioneaza corect.

**5. Explicati diferentele intre functional testing si nonfunctional testing.**

- Testare funcțională : Testele funcționale sunt teste scrise pe baza specificațiilor și arată ce trebuie să facă produsul, reprezentand acțiuni făcute de către system.
- Testarea non-funcțională - Verifică atribute care descriu cât de bine își îndeplinește sistemul funcțiile. e.g. reliability, eficiență, mentenabilitate, transferabilitate, performanță, recuperare, localizare, conformitate etc

**6. Explicati diferentele intre blackbox testing si whitebox testing.**

Blackbox testing inseamna testare fara acces la cod si whitebox inseamna testare cu acces la cod.



## 7. Enumerati tehnicile de testare si grupatile in functie de categorie.

### ➤ Tehnici de testare :

- -testare statica (cea mai cunoscuta este testarea prin review- formal sau informal);
- -testare dinamica
- - whitebox testing
- - blackbox testing ( equivalence partitioning, BVA (boundary value analysis), state transition testing, decision table);
- - experience based testing

## 8. Explicati diferenta intre verification testing si validation testing.

- Verification - Se creează produsul cum trebuie?(se face cu scopul de a evalua materialele care stau la baza testării );
- Validation - Este produsul corect? Construiesc produsul care trebuie? ( se face cu scopul de a evalua produsul finit și a ne asigura că acesta îndeplinește cerintele de business si nevoile clientului ).

## 9. Explicati diferenta intre positive testing si negative testing si dati cate un exemplu din fiecare .

- Testare pozitivă înseamnă testarea sistemului cu valori pe care ar trebui să le poată procesa.( Verificam daca putem introduce valori tip litere, numere, caractere speciale pe campul de creare cont )
- Testare Negativă înseamnă testare cu valori pe care sistemul nu ar trebui să le poată procesa în mod normal pentru a ne asigura că aceste valori sunt într-adevăr respinse și că nu cauzează un crash al sistemului.( verificam ca nu putem introduce un numar mai mare de x caractere pentru crearea unui cont).

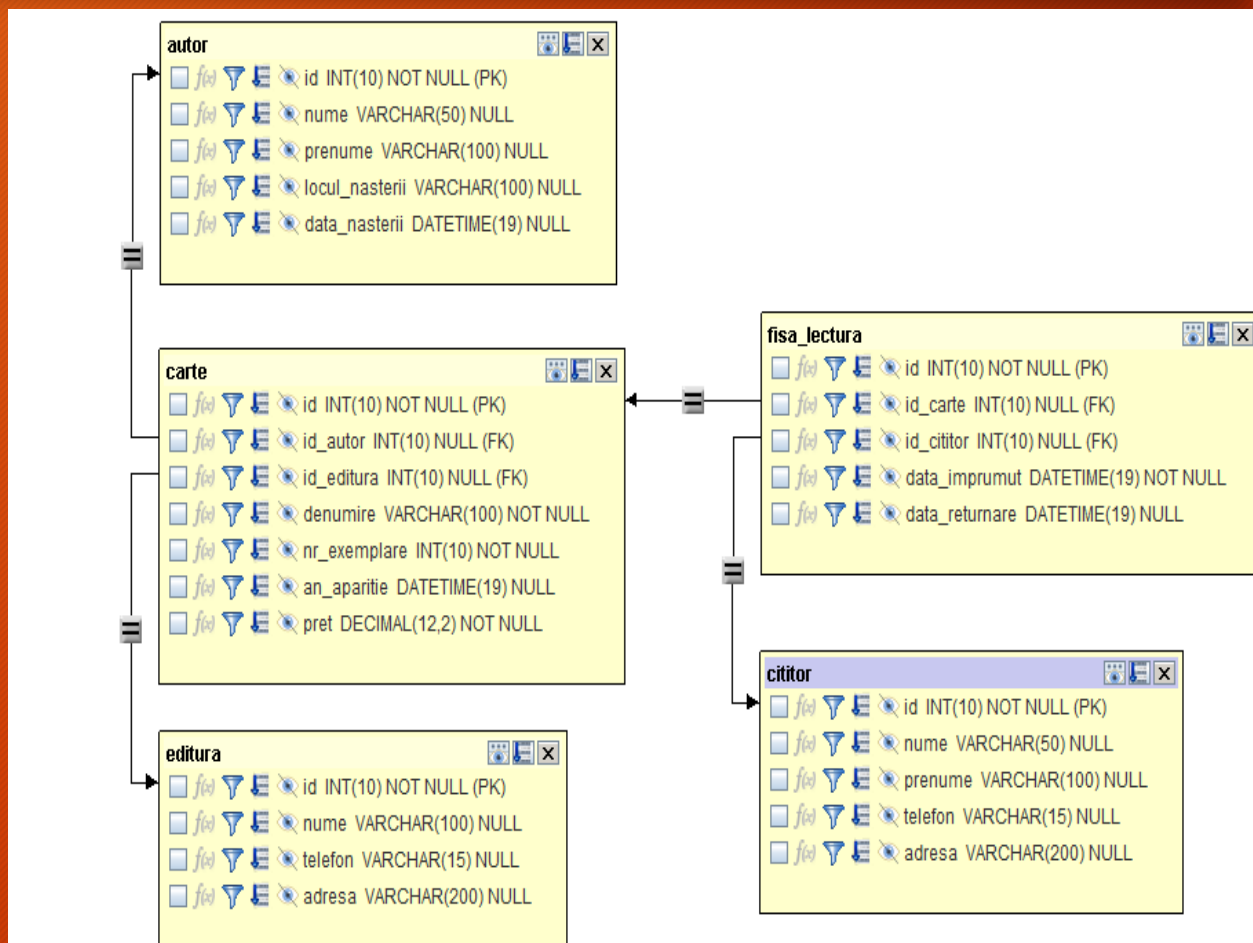
## 10. Enumerati si explicati pe scurt nivelurile de testare .

### ➤ Niveluri de testare :

- ❑ - Testare unitară - Un test unitar reprezintă testarea celei mai mici bucăți funcționale dintr-o aplicație ;
- ❑ - Testarea de Integrare - Se concentrează pe interacțiunile dintre componente și sisteme.
  - 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 ( Evaluează comportamentul și capacitatea sistemului ca un tot unitar)
- ❑ - Testarea de acceptanta ( verifică felul în care acesta îndeplinește nevoile clientului / utilizatorului )-
- - alpha testing ( reprezintă testarea unei aplicații atunci când dezvoltarea este completă sau aproape completa );
- - beta testing ( Scopul testării beta este să pună aplicația în mâinile unor utilizatori reali, oameni ce nu fac parte din echipa de dezvoltatori, pentru a descoperi defecte din perspectiva utilizatorului).

# Partea II. Baze de date

Creare baze de date si legaturi intr-e tabele .  
Tabele de autori, carte, editura, cititor si fisa de lectura





✓ Am creat baza de date folosind sintaxele:

- Create database;
- Use;

✓ Am creat tabele ( autor, editura, carte, cititor, fisa de lectura )

```
-- Creare tabele proiect "Imprumuturi carti biblioteca"
```

```
drop database Imprumuturi_carti_biblioteca;  
create database Imprumuturi_carti_biblioteca;  
use Imprumuturi_carti_biblioteca;
```

```
create table autor (  
    id int primary key AUTO_INCREMENT,  
    nume varchar(50),  
    prenume varchar(100),  
    locul_nasterii varchar(100),  
    data_nasterii datetime  
);
```

```
create table editura (  
    id int primary key AUTO_INCREMENT,  
    nume varchar(100),  
    telefon varchar(15),  
    adresa varchar(200)  
);
```

```
create table carte (  
    id int primary key AUTO_INCREMENT,  
    id_autor int,  
    id_editura int,  
    denumire varchar(100) not null,  
    nr_exemplare int not null,  
    an_aparitie datetime,  
    pret decimal(12,2) not null,  
    FOREIGN KEY (id_autor) REFERENCES autor(id),  
    FOREIGN KEY (id_editura) REFERENCES editura(id)  
);
```

```
create table cititor (  
    id int primary key AUTO_INCREMENT,  
    nume varchar(50),  
    prenume varchar(100),  
    telefon varchar(15),  
    adresa varchar(200)  
);
```

```
create table fisa_lectura (  
    id int primary key AUTO_INCREMENT,  
    id_carte int,  
    id_cititor int,  
    data_imprumut datetime not null,  
    data_returnare datetime,  
    FOREIGN KEY (id_carte) REFERENCES carte(id),  
    FOREIGN KEY (id_cititor) REFERENCES cititor(id)  
);
```



Pentru tabele, am folosit instructiunea “insert” pentru a introduce date.




```
-- autor
INSERT INTO autor (nume,prenume,locul_nasterii,data_nasterii) VALUES ('Eminescu','Mihai', 'Botosani', '1850-01-15');
INSERT INTO autor (nume,prenume,locul_nasterii,data_nasterii) VALUES ('Creanga','Ion', 'Iasi', '1889-12-31');
INSERT INTO autor (nume,prenume,locul_nasterii,data_nasterii) VALUES ('Sadoveanu','Mihail', 'Bucuresti', '1961-10-19');
```

```
- editura
INSERT INTO editura (nume,telefon,adresa) VALUES ('NEMIRA','0721747464','Str. Iani Buzoiani, nr. 14, sector 1, Bucuresti, România');
INSERT INTO editura (nume,telefon,adresa) VALUES ('Editura Gramar','0770104175','Bucuresti, Strada Ion Baiulescu, nr. 75');

- cititor
INSERT INTO cititor (nume,prenume,telefon,adresa) VALUES ('Vasile','Ion','0721111111','Bd. Constantin Brancoveanu nr 12');
INSERT INTO cititor (nume,prenume,telefon,adresa) VALUES ('Teodor','Critian','0721222222','Bd. Constantin Brancoveanu nr 13');
```

Am folosit instructiunea “select” pentru afisarea tabelelor create.

```
349 • select * from autor;  
350 • select id_autor, nume_autor, prenume_autor  
351 • select * from carte;  
352 • select * from cititor;  
353 • select * from editura;  
354 • select * from fisa_lectura;  
355
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:   					
	id	nume	prenume	locul_nasterii	data_nasterii
▶	1	Eminescu	Mihai	Botosani	1850-01-15 00:00:00
	2	Creanga	Ion	Iasi	1889-12-31 00:00:00
	3	Sadoveanu	Mihail	Bucuresti	1961-10-19 00:00:00
*	NULL	NULL	NULL	NULL	NULL

Am afisat numarul total de carti dupa autorul Eminescu.

```
384 • select sum(nr_exemplare) as total_exemplare from carte where id_autor= (select id from autor where nume= 'Eminescu');  
385
```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

total_exemplare	
▶	15



Am afisat cate carti are fiecare autor in ordine alfabetica., folosind sintaxele “inner join”, “group by”, “order by”.

```
385
386 • select a.id, a.nume, a.prenume, sum(c.nr_exemplare) as total_exemplare
387 from carte c
388 inner join autor a on a.id = c.id_autor
389 group by a.id, a.nume, a.prenume
390 order by a.nume, a.prenume;
391
392
393
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
id	nume	prenume	total_exemplare
2	Creanga	Ion	4
1	Eminescu	Mihai	15
3	Sadoveanu	Mihail	14

In tabela “autor” am afisat in ordine descrescatoare autorii, dupa id.





```
64 -- aranjare ordine descrescatoare
65 • select*from autor
66 order by id desc
67 limit 2 offset 1;
68
```

Result Grid

id	nume	prenume	locul_nasterii	data_nasterii
2	Creanga	Ion	Iasi	1889-12-31 00:00:00
1	Eminescu	Mihai	Botosani	1850-01-15 00:00:00
NULL	NULL	NULL	NULL	NULL

Am folosit instructiunea “inner join” ,pentru listarea cartilor  
fiecarui autor si a editurilor aferente.

```
372 • select a.id, e.id, a.numa as autor_nume, a.prenume as autor_prenume, c.denumire as denumire_carte, e.numa as editura_nume
373 from autor a
374 inner join carte c on c.id_autor = a.id
375 inner join editura e on a.id = c.id_editura;
```

Result Grid		 Filter Rows:			Export:		Wrap Cell Content:	
	id	id	autor_nume	autor_prenume	denumire_carte	editura_nume		
	2	1	Creanga	Ion	Amintiri din copilarie	NEMIRA		
	1	1	Eminescu	Mihai	La mormântul lui Aron Pumnul	NEMIRA		
	1	1	Eminescu	Mihai	Luceafarul	NEMIRA		
	2	2	Creanga	Ion	Amintiri din copilarie	Editura Gramar		
	1	2	Eminescu	Mihai	La mormântul lui Aron Pumnul	Editura Gramar		
	1	2	Eminescu	Mihai	Luceafarul	Editura Gramar		
	2	3	Creanga	Ion	Amintiri din copilarie	Editura Carturesti		
	1	3	Eminescu	Mihai	La mormântul lui Aron Pumnul	Editura Carturesti		
▶	1	3	Eminescu	Mihai	Luceafarul	Editura Carturesti		



Am folosit instructiunea “right join” pentru listarea autorilor, cartilor si a editurilor ,din tabela din dreapta si a celor commune din celelalte tabele .

```
379 • select a.id, e.id, a.numa as autor_numa, a.prenume as autor_prenume, c.denumire as denumire_carte, e.numa as editura_numa
380 from autor a
381 right join carte c on c.id_autor = a.id
382 right join editura e on a.id = c.id_editura;
383
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	id	id	autor_numa	autor_prenume	denumire_carte	editura_numa
▶	1	1	Eminescu	Mihai	Luceafarul	NEMIRA
	1	1	Eminescu	Mihai	La mormantul lui Aron Pumnul	NEMIRA
	2	1	Creanga	Ion	Amintiri din copilarie	NEMIRA
	1	2	Eminescu	Mihai	Luceafarul	Editura Gramar
	1	2	Eminescu	Mihai	La mormantul lui Aron Pumnul	Editura Gramar
	2	2	Creanga	Ion	Amintiri din copilarie	Editura Gramar
	1	3	Eminescu	Mihai	Luceafarul	Editura Carturesti
	1	3	Eminescu	Mihai	La mormantul lui Aron Pumnul	Editura Carturesti
	2	3	Creanga	Ion	Amintiri din copilarie	Editura Carturesti

Va multumesc pentru atentie!