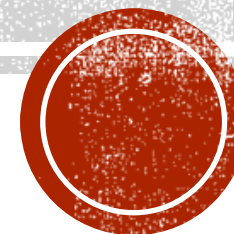


PROJECT FINAL

Rosian Alvin Ionut

04.09.2024



CERINTE DE BUSINESS

Ce sunt:

Cerinte in vederea dezvoltarii unei aplicatii, constituie o descriere completa a asteptarilor in vederea functionalitatii aplicatiei.

Utilitate:

Asigura aliniera tuturor partilor in vederea crearii produsului final

Cine le creeaza:

Echipa de business in colaborare cu echipa de dezvoltare si cea de testare.

TEST CONDITION VS TEST CASE

Test condition se refera la aspecte generale ce trebuiesc testate, in aditia acestuia un **test case** reprezinta o detaliere ampla a ce vom testa si contine: un set de pasi de urmat; conditii de testare; rezultate asteptate.

In concluzie putem privi relatia dintre test condition si test case ca fiind una complementara incepand de la identificarea test conditions pentru a dezvolta mai apoi un test case.



ETAPELE PROCESULUI DE TESTARE:

➤ Planificare

- Descriere: Etapa incipienta in care se definesc strategiile de testare, obiectivele testarii, resursele necesare, programul de testare cat si riscurile asociate.
- Activitati: Crearea planului de testare; identificarea obiectivelor testarii; stabilirea calendarului.

➤ Analiza

- Descriere: In aceasta etapa se analizeaza cerintele si specificatiile sistemului pentru a identifica ce anume se testeaza.
- Activitati: generarea conditiilor de testare.

➤ Design

- Descriere: Etapa in care se elaboreaza cazurile de testare si se identifica datele de testare.
- Activitati: crearea de test cases.



➤ **Implementare**

- Descriere: In aceasta etapa se pregatesc testele pe baza indicatiilor anterioare.
- Activitati: Configurarea mediului de testare; pregatirea datelor de testare; instalarea programelor si instrumentelor de testare.

➤ **Executie**

- Descriere: Etapa in care se executa testele conform planului de testare realizat anterior.
- Activitati: Rularea de teste; inregistrarea rezultatelor; raportare bug-uri.

➤ **Monitorizare si Control**

- Descriere: In aceasta etapa se monitorizeaza procesul iar dupa caz acesta se poate ajusta conform necesitatilor.
- Activitati: Monitorizare progres; ajustare plan de testare (daca este cazul); evaluarea rezultatelor obtinute vs rezultate asteptate.

➤ **Inchidere**

- Descriere: Aceasta este etapa finala a procesului de testare in care se finalizeaza testarea si se documenteaza rezultatele obtinute
- Activitati: Pregatire raport final de testare; evaluarea rezultatelor si a performantei echipei de testare



RETESTING VS REGRESSION TESTING

- **Retesting:**

Retesting-ul reprezinta repetarea procesului de testare asupra unei functionalitati care a fost depistata in trecut ca fiind nefunctionala si a fost corectata intre timp. Scopul retestari este acela de a ne asigura ca defectul a fost remediat.

- **Regression testing:**

Este procesul de testare a unei parti semnificative a unui sistem sau a intregului sistem care urmeaza dupa implementarea de noi modificari asupra sistemului iar scopul acesteia este de a ne asigura ca noile functionalitati nu au afectat functionalitatile deja existente. Regresion testing-ul se realizeaza in general dupa modificari importante ale softului dar nu numai.

Putem spune ca ambele procese de testare sunt esentiale pentru asigurarea calitatii produsului, retesting-ul verificand remedierea unui bug in timp ce regresion testing-ul se asigura ca modificarile precedente nu au afectat sistemul si nu au creat noi bug-uri.



FUNCTIONAL TESTING VS NON-FUNCTIONAL TESTING

- **functional testing**

Este procesul de testare ce se asigura de functionarea produsului

- **non-functional testing**

Este procesul de testare ce verifica specificatiile si caracteristicile softului non functionale de ex.: performanta, compatibilitatea etc.

BLACKBOX TESTING VS WHITEBOX TESTING

- **Blackbox testing**

Se refera la testarea produsului fara acces la codul acestuia

- **Whitebox testing**

Se refera la testarea produsului cu acces la codul acestuia



TEHNICI DE TESTARE

➤ **Blackbox testing**

- Equivalence partitioning
- Boundary value analysis
- Decision table testing
- State transition testing

➤ **Whitebox testing**

- Statement Coverage
- Decision Coverage

➤ **Experience based testing**

- Exploratory Testing
- Error Guessing
- Checklist-Based Testing



VERIFICATION VS VALIDATION

- **Verification**

Se refera la faptul ca softul este construit conform specificatiilor

- **Validation**

Face referire la faptul ca produsul satisface nevoile utilizatorului.

POSITIVE TESTING VS NEGATIVE TESTING

- **Positive testing**

Este testarea ce are ca scop demonstrarea faptului ca produsul functioneaza conform asteptarilor atunci cand este utilizat in conditii normale

Ex.: autentificarea pe un website cu date de login valide(username & password)

- **Negative testing**

Este testarea care are ca scop descoperirea de erori, bug-uri

Ex.: incercarea de autentificare pe un website folosind date de logare invalide



NIVELURILE DE TESTARE

- **Unit testing (testare unitara)**

Testeaza doar unitati de cod mici cum ar fi functii sau metode.

- **Integration testing (testare de integrare)**

Testeaza interactiunea dintre componentele softului pentru a asigura functionarea acestora impreuna.

- **System testing (testare de sistem)**

Verifica functionalitatea completa a sistemului testand intregul sistem software.

- **Acceptance testing (testare de acceptanta)**

Verifica daca sistemul indeplineste asteptarile utilizatorului final



BAZE DE DATE SQL

Pentru aceasta parte a proiectului am ales sa creez o baza de date pentru o biblioteca adaugand 5 tabele in aceasta. In cele ce urmeaza voi prezenta cateva instructiuni DDL, DML, DQL proiectul complet fiind accesibil pe Github in linkul de mai jos:

<https://github.com/Alvin-Rosian/Manual-Testing-Project>

DDL(Data Definition Language): Crearea tabelului Autori folosind functia CREATE

```
4 • ○ CREATE TABLE Autori (  
5     AutorID INT PRIMARY KEY AUTO_INCREMENT,  
6     Nume VARCHAR(100) NOT NULL,  
7     Prenume VARCHAR(100) NOT NULL,  
8     DataNasterii DATE  
9     );
```

Comanda de mai sus a creat tabela Autori cu coloanele aferente: AutorID; Nume; Prenume; DataNasterii

In mod asemanator au fost create un total de 5 astfel de tabele: Autori; Carti; Utilizatori; Imprumuturi; Recenzii.



- **DML (Data Manipulation Language)**

Inserarea datelor in tabelurile create anterior folosind functia INSERT.

Comanda alaturata a populat tabela Autori cu informatii introduse de la tastatura campurile Nume; Prenume si data nasterii, campul AutorID fiind populat automat cu ajutorul functiei AUTO_INCREMENT atribuita anterior in pasul de creare a tabelui

```
48 • INSERT INTO Autori (Nume, Prenume, DataNasterii) VALUES
49 ('Rowling', 'J.K.', '1965-07-31'),
50 ('Tolkien', 'J.R.R.', '1892-01-03'),
51 ('Martin', 'George R.R.', '1948-09-20'),
52 ('Austen', 'Jane', '1775-12-16'),
53 ('Shakespeare', 'William', '1564-04-26'),
54 ('Orwell', 'George', '1903-06-25'),
55 ('Hemingway', 'Ernest', '1899-07-21'),
56 ('Fitzgerald', 'F. Scott', '1896-09-24'),
57 ('Dickens', 'Charles', '1812-02-07'),
58 ('Huxley', 'Aldous', '1894-07-26'),
59 ('Verne', 'Jules', '1828-02-08'),
60 ('Dostoevsky', 'Fyodor', '1821-11-11'),
61 ('Kafka', 'Franz', '1883-07-03'),
62 ('Joyce', 'James', '1882-02-02'),
63 ('Poe', 'Edgar Allan', '1809-01-19'),
64 ('Homer', 'S', '800-01-01'),
65 ('Hugo', 'Victor', '1802-02-26'),
66 ('Steinbeck', 'John', '1902-02-27'),
67 ('Christie', 'Agatha', '1890-09-15'),
68 ('King', 'Stephen', '1947-09-21');
```



▪ DQL (Data Query Language):

Selectarea datelor din tabelele create si poulate anterior folosind functia SELECT:

Selectarea tuturor cartilor: 158 • `SELECT * FROM Carti;`

Returneaza toate cartile din tabela Carti:

	CarteID	Titlu	Gen	AnPublicare	AutorID
▶	161	Harry Potter and the Philosopher's Stone	Fantasy	1997	1
	162	Harry Potter and the Chamber of Secrets	Fantasy	1998	1
	163	The Hobbit	Fantasy	1937	2
	164	The Fellowship of the Ring	Fantasy	1954	2
	165	A Game of Thrones	Fantasy	1996	3
	166	A Clash of Kings	Fantasy	1998	3
	167	Pride and Prejudice	Romance	1951	4
	168	1984	Dystopian	1949	6
	169	Animal Farm	Satire	1945	6
	170	The Great Gatsby	Tragedy	1925	8
	171	Brave New World	Dystopian	1932	10
	172	Moby Dick	Adventure	1940	7
	173	War and Peace	Historical	1969	12
	174	The Odyssey	Epic	1920	16
	175	Les Misérables	Historical	1962	17
	176	The Grapes of Wrath	Historical	1939	18
	177	Murder on the Orient Express	Mystery	1934	19
	178	The Shining	Horror	1977	20
	179	It	Horror	1986	20
	180	The Catcher in the Rye	Literary	1951	14

Selectarea tuturor cartilor care fac parte din genul Fantasy: 160 • `SELECT * FROM Carti WHERE Gen = 'Fantasy';`

Returneaza toate cartile de genul Fantasy:

	CarteID	Titlu	Gen	AnPublicare	AutorID
▶	161	Harry Potter and the Philosopher's Stone	Fantasy	1997	1
	162	Harry Potter and the Chamber of Secrets	Fantasy	1998	1
	163	The Hobbit	Fantasy	1937	2
	164	The Fellowship of the Ring	Fantasy	1954	2
	165	A Game of Thrones	Fantasy	1996	3
	166	A Clash of Kings	Fantasy	1998	3



- Selectarea utilizatorilor cu numele Ionescu sau Popescu: 162 • `SELECT * FROM Utilizatori WHERE Nume = 'Ionescu' OR Nume = 'Popescu';`

Returneaza toti utilizatorii cu numele Ionescu sau Popescu:

	UtilizatorID	Nume	Prenume	Email
►	1	Popescu	Ion	ion.popescu@example.com
	2	Ionescu	Maria	maria.ionescu@example.com

Selectarea utilizatorilor care au facut imprumuturi si au returnat cartile in 2024:

```
164 • SELECT Utilizatori.Nume, Utilizatori.Prenume, Imprumuturi.DataReturnare
165 FROM Utilizatori
166 INNER JOIN Imprumuturi ON Utilizatori.UtilizatorID = Imprumuturi.UtilizatorID
167 WHERE YEAR(Imprumuturi.DataReturnare) = 2024;
```

returneaza →

	Nume	Prenume	DataReturnare
►	Popescu	Ion	2024-02-15
	Georgescu	Andrei	2024-02-21
	Dumitrescu	Elena	2024-03-01
	Mihai	Vasile	2024-03-25
	Radu	Alexandru	2024-04-20
	Stan	Ana	2024-04-30
	Marin	Daniel	2024-05-20
	Nicolae	Gabriel	2024-06-15
	Iliescu	Simona	2024-06-25
	Barbu	Catalin	2024-07-20
	Enache	Laura	2024-07-30
	Serban	Mihai	2024-08-20
	Nita	Raluca	2024-08-30

Selectarea genurilor care au mai mult de o carte:

```
171 • SELECT Gen, COUNT(*) AS NumarCarti FROM Carti GROUP BY Gen HAVING COUNT(*) > 1;
```

returneaza →

	Gen	NumarCarti
►	Fantasy	6
	Dystopian	2
	Historical	3
	Horror	2

Selectarea tuturor imprumuturilor si utilizatorii asociati acestora:

```
173 • SELECT Imprumuturi.ImprumutID, Utilizatori.Nume, Utilizatori.Prenume
174 FROM Imprumuturi
175 LEFT JOIN Utilizatori ON Imprumuturi.UtilizatorID = Utilizatori.UtilizatorID;
```

returneaza →

	ImprumutID	Nume	Prenume
►	21	Popescu	Ion
	22	Ionescu	Maria
	23	Georgescu	Andrei
	24	Dumitrescu	Elena
	25	Mihai	Vasile
	26	Constantinescu	Adriana
	27	Radu	Alexandru



▪ Subqueri-uri:

Selectarea utilizatorilor care au imprumutat carti de la un anumit autor:

```
183 • SELECT Utilizatori.Nume, Utilizatori.Prenume
184 FROM Utilizatori
185 WHERE UtilizatorID IN (
186     SELECT Imprumuturi.UtilizatorID
187 FROM Imprumuturi
188 INNER JOIN Carti ON Imprumuturi.CarteID = Carti.CarteID
189 WHERE Carti.AutorID = (SELECT AutorID FROM Autori WHERE Nume = 'Tolkien')
190 )
```

returneaza →

	Nume	Prenume
▶	Georgescu	Andrei
	Dumitrescu	Elena

Relatiile dintre tabelele create cu chei primare:

Autori si Carti au o relație de tip 1:1 (un autor poate avea mai multe carti, dar fiecare carte are un singur autor).

Utilizatori si Imprumuturi au o relație de tip 1:1 (un utilizator poate avea mai multe imprumuturi, dar fiecare imprumut este asociat unui singur utilizator).

Carti si Imprumuturi au o relație de tip 1:1 (o carte poate fi imprumutata de mai multe ori, dar fiecare imprumut este asociat unei singure carti).

Utilizatori si Recenzii au o relație de tip 1:1 (un utilizator poate face mai multe recenzii, dar fiecare recenzie este asociata unui singur utilizator).

Carti si Recenzii au o relație de tip 1:1 (o carte poate avea mai multe recenzii, dar fiecare recenzie este asociată unei singure carti).



VA MULTUMESC PENTRU ATENTIE

