



PROIECT FINAL

STERIE IOLANDA MARIA



EXAMEN ACREDITARE

04.10.2024



PARTEA 1

1. Explicați pe scurt ce sunt cerințele de business, la ce ne folosesc și cine le creează.

Cerintele de business sunt un set de documente care trebuie sa contina felul in care un anumit produs ar trebui sa functioneze, conform dorintelor clientului. Acestea se intocmesc de catre client (business analyst) si stau la baza procesului de testare.

2. Explicați diferența între un test condition și test case.

Test condition este definit ca si o functionalitate ce poate fi validata la un anumit moment; se numeste conditie de testare pentru ca reprezinta conditia pe care trebuie sa o indeplineasca functionalitatea testata pentru a fi considerata corecta. Acesta raspunde la intrebarea :” **Ce testam?** ”

Test case este definit ca si un grup de pasi ce trebuie implementati pentru verificarea calitatii unei anumite functionalitati. Acesta raspunde la intrebarea :” **Cum testam?** ”

Ca si componente un **test case** are *summary, preconditionii, pasi de reproducere si rezultate asteptate*.

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

Acestea sunt:

a) Etapa de planificare- aceasta etapa include activitati care definesc obiectivele cu privire la testare. Se decide care parti ale aplicatiei se doreste sa fie testate. Se aloc rolurile in cadrul noii echipe.

Se definesc criteriile de intrare si criteriile de iesire. Se identifica riscurile de proiect initiale si resursele necesare. Se creeaza un plan de testare.

b) **Etapa de analiza**- raspunde la intrebarea “**ce urmeaza sa testam?**”, in aceasta etapa analizandu-se documentatia primita de la client, pentru a ne asigura ca intelegem si ca nu exista greseli. Tot in acesta etapa se genereaza si conditiile de testare (ce vom testa).

c) **Etapa de design**- raspunde la intrebarea “**cum vom testa?**”. Aici se creeaza cazurile de testare si se indentifica datele de testare.

d) **Etapa de implementare**- raspunde la intrebarea “**avem tot ce ne trebuie pentru a incepe executarea testelor?**” Aici se creeaza datele de testare , validandu-se mediul de testare prin intermediul smoke testing, se priorititeaza testetele si se grupeaza pe baza obiectivelor lor.

Ne asigura ca avem tot ce ne trebuie pentru a incepe testarea propriu-zisa.

e) **Etapa de executie**- Aici sunt executate cazurile de testare, rezultatele sunt raportate in tool-ul in care au fost scrise testele. Se raporteaza bug-urile/defectele/fault-urile. Dupa fixarea bug-urilor se face retestarea lor. Tot in aceasta etapa se face si testarea de regresie , pentru a ne asigura ca schimbarile facute, nu au avut un impact negativ asupra functionalitatilor existente.

f) **Etapa de inchidere**- In aceasta etapa se evalueaza criteriile de iesire; se reevalueaza task-urile ramase deschise si bug-urile; iar apoi se inhid , se predau si arhiveaza materialele de testare.

Se genereaza un raport de inchidere a testarii (test summary report) si se trimite catre stakeholders.

4. Explicați diferența între retesting și regression testing.

Atat **retesting** , cat si **regression testing** sunt tipuri de testare facute in urma schimbarilor. Insa diferenta este ca retesting se concentreaza pe a revalida o functionalitate care anterior a fost evaluata ca fiind incorecta (pentru a ne asigura ca acum are comportamentul asteptat) , iar regression testing se concentreaza pe functionalitati care anterior au fost confirmate ca si corecte si pentru a ne asigura ca acestea in continuare se ridica la nivelul asteptarilor clientului.

5. Explicați diferența între functional testing și non-functional testing.

Testarea functionala verifica daca produsul isi indeplineste testele, iar **testarea non-functionala** verifica attributele care descriu cat de bine isi indeplineste sistemul functiile (eficienta, mentenanta, transferabilitate, etc.)

Testarea functionala raspunde la intrebarea “ce trebuie sa faca produsul?” , iar cea **non-functionala** raspunde la intrebarea “cum trebuie sa se comporte produsul?” .

6. Explicați diferența între blackbox testing și whitebox testing

Blackbox testing reprezinta o testare fara acces la cod.

Tehnicile de testare Blackbox mai sunt numite si tehnici de testare bazate pe specificatii si sunt o modalitate prin care putem sa generam conditii de testare , cazuri de testare sau date de testare pe baza unei analize a documentatiei care sta la baza testarii. Asta include atat testarea functionala cat si testarea non-functionala.

Whitebox testing reprezinta o testare cu rulare de cod. Este o tehnica de testare dinamica prin intermediul careia se valideaza codul.

7. Enumerați tehnicile de testare și grupați-le în funcție de categorie (blackbox, whitebox, experience-based).

Tehnicile de testare se impart in: testare statica (testare fara rulare de cod) si testare dinamica (testare cu rulare de cod).

Tehnicile de testare dinamica se impart in:

- Black-box testing-Equivalence Partitioning
 - *Boundary Value Analysis (2 point and 3 point BVA)*
 - *State Transition Testing*
 - *Decision Table Testing*
- White-box testing
 - *Statement Coverage*
 - *Decision Coverage*
- Experience-based testing
 - *Testare ad hoc*
 - *Ghicirea erorilor*
 - *Testare exploratorie*

8. Explicați diferența între verification și validation.

Verificarea se concentreaza pe felul in care produsul este construit. Este un tip de **testare proactiva** care se face cu scopul de a evalua materialele care stau la baza testarii (cerinte, cod, teste,planuri de testare) pentru a ne asigura ca produsul este construit in mod corect. Raspunde la intrebarea “**Construiesc produsul asa cum trebuie?**”.

Validarea se concentreaza pe felul in care produsul functioneaza. Este un tip de testare reactiva care se face cu scopul de a evalua produsul finit si a ne asigura ca acesta indeplineste cerintele de business si nevoile utilizatorului. Raspunde la intrebarea “**Este produsul corect?**”.

9. Explicați diferența între positive testing și negative testing și dați câte un exemplu din fiecare

Testarea pozitivă înseamnă testarea sistemului cu valori pe care ar trebui să le poată procesa, iar **testarea negativă** înseamnă testarea 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 nu cauzează un crash al sistemului.

Exemplu testare pozitivă: când se introduc date de autentificare corecte, în cadrul unei aplicații

Exemplu testare negativă: când se introduc în mod voit date de autentificare incorecte pentru a verifica funcționalitatea sistemului (dacă acesta răspunde așa cum dorim în cazul de față).

10. Enumerați și explicați pe scurt nivelurile de testare

Cele 4 niveluri de testare sunt:

Acceptance Testing – evaluează comportamentul produsului și verifică felul în care acesta îndeplinește nevoile clientului/utilizatorului.

System Testing – evaluează comportamentul și capacitatea sistemului ca un tot unitar, ținând cont de comportamentul end-to-end al funcționalităților pe care sistemul trebuie să le execute și de comportamentul non-funcțional așteptat al acelor task-uri.

Integration Testing – testează interacțiunile dintre componente (component integration testing) și sistem (system integration testing).

Unit Testing – testarea celor mai mici bucăți funcționale dintr-o aplicație.

PARTEA II VARIANTA 3

Această variantă trebuie să conțină următoarele componente:

- Instrucțiuni DDL (cel puțin una dintre CREATE, ALTER, DROP, TRUNCATE)
- Instrucțiuni de DML (INSERT, DELETE, UPDATE)
- Instrucțiuni DQL (select all, select câteva coloane, filtrare cu where, filtrări cu like, filtrări cu AND și OR, funcții agregate, filtrări pe funcții agregate, joinuri - inner join, left join, right join, cross join, limite, order by, chei primare, chei secundare)
- OPTIONAL (ofera bonus la examen) - folosirea subquery-urilor

Fiecare instrucțiune va trebui să aibă asociată o explicație a scenariului pe care l-ați acoperit.

De asemenea, fiecare două tabele legate prin **chei primare** și **chei secundare** vor trebui să aibă atașată o explicație legată de felul în care acestea sunt legate (care e cheia primară, care e cheia secundară) și respectiv care e relația între cele două tabele (1:1, 1:n, n:n).

Creare **baza de date**

```
create database myshop;
```

Creare **tabela produse**

```
create table produse(  
    id int primary key auto_increment,  
    nume_produș varchar(30),  
    categorie_produș varchar(40),  
    pret_vanzare float(5),  
    stoc_disponibil int,  
    descriere_produș varchar(150));
```

Creare **tabela utilizatori**

```
create table utilizatori(  
    id int primary key auto_increment,  
    nume_utilizator varchar (30),  
    email_utilizator varchar (40),  
    adresa_livrare varchar (80));
```

Creare **tabela comenzi**

```
create table comenzi(  
    id int primary key auto_increment,  
    id_client int,  
    data_comanda varchar(12));
```

Creare tabela **detalii comenzi**

```
create table detalii_comenzi(  
    id int primary key auto_increment,  
    id_comanda int,  
    id_produc int,  
    cantitate int,  
    pret_total float(5));
```

Creare tabela **gestiune produse**

```
create table gestiune_produce(  
    id int primary key auto_increment,  
    id_produc int,  
    cantitate_intrari int,  
    cantitate_iesiri int,  
    stoc_depozit int);
```

Foreign key

Coloana **id_produc** din tabela **comenzi** face referinta catre coloana **id** din tabela **produse**.

Relatia intre cele doua tabele este 1:1

```
alter table detalii_comenzi  
add foreign key (id_produc) references produse(id);
```

Foreign key

Coloana **id_comanda** din tabela **detalii_comenzi** face referinta catre coloana **id** din tabela **comenzi**

Relatia intre cele doua tabele este 1:1

```
alter table detalii_comenzi  
add foreign key (id_comanda) references comenzi(id);
```

Foreign key

Coloana **id_produc** din tabela **gestiune_produce** face referinta catre coloana **id** din tabela **produse**

Relatia intre cele doua tabele este 1:1

```
alter table gestiune_produce  
add foreign key (id_produc) references produse(id);
```


Inserate date tabela produse

```
insert into produse (id, nume_produș, categorie_produș, pret_vanzare, stoc_disponibil, descriere_produș)
values (1, "Antemergator", "Jucarii Montessori", 420, 22, "Antemergator 6 in 1 cu ceas labirint, abac,
        forme geometrice, puzzle 3D, din lemn");
insert into produse (id, nume_produș, categorie_produș, pret_vanzare, stoc_disponibil, descriere_produș)
values (2, "Ceas copii", "Jucarii Montessori", 173, 41, "Ceas educativ Montessori, multicolor, lemn,
        15 cm diametru");
insert into produse (id, nume_produș, categorie_produș, pret_vanzare, stoc_disponibil, descriere_produș)
value (3, "Puzzle 3D", "Jucarii Montessori", 31, 73, "Puzzle educativ Montessori cu 6 fete, caracita,
        delfin, broasca testoasa, rac, crab, rechin. Jucarie din lemn cu dimensiuni 15x15x5, 9 piese");
insert into produse (id, nume_produș, categorie_produș, pret_vanzare, stoc_disponibil, descriere_produș)
value (4, "Joc educativ matematic", "Jucarii Montessori", 82, 20, "Joc educativ matematic Montessori abac,
        cu cercuri si coloane, lemne, multicolor. Dimensiuni 35x5x12 cm, 56 piese.");
insert into produse (id, nume_produș, categorie_produș, pret_vanzare, stoc_disponibil, descriere_produș)
value (5, "Balance Board", "Mobilier copii", 480, 17, "Placa de echilibru din lemn, ce ajuta imbunatatirea
        coordonarii motrice. Design minimalist. Adecvat pentru toate varstele. Sarcina maxima 200kg.");
insert into produse (id, nume_produș, categorie_produș, pret_vanzare, stoc_disponibil, descriere_produș)
value (6, "Pink Lemon", "Jucarii organice", 159, 36, "Jucarie muzicala din bumbac organic, ce ii permite
        copilului sa adoarma mai usor");
insert into produse (id, nume_produș, categorie_produș, pret_vanzare, stoc_disponibil, descriere_produș)
value (7, "Marsupiu ISARA", "Marsupii ergonomice", 859, 36, "Marsupiu ajustabil, marime unica, realizat
        din bumbac organic, potrivit inca de la nastere. Sarcina maxima: 15 kg");
```

Inserare date tabela utilizatori

```
insert into utilizatori (id, nume_utilizator, email_utilizator, adresa_livrare)
value (1, "IONESCU CALIN", "ionescucalin.10@gmail.com", "Bucuresti, str Ion Mincu, bl A3, ap 5");
insert into utilizatori (id, nume_utilizator, email_utilizator, adresa_livrare)
value (2, "POPESCU DIANA", "popescu.diana@yahoo.com", "Craiova, str Ion Creanga, nr 35");
insert into utilizatori (id, nume_utilizator, email_utilizator, adresa_livrare)
value (3, "STERIE ANTONIA", "sterieantonio.5@gmail.com", "Targoviste, str Mihai Eminescu bl 1A, ap 10");
insert into utilizatori (id, nume_utilizator, email_utilizator, adresa_livrare)
value (4, "STAN DAN", "stan.7dan@yahoo.com", "Cluj, str Craciunului, nr.72");
insert into utilizatori (id, nume_utilizator, email_utilizator, adresa_livrare)
value (5, "ION ANDREI", "ionandrei25@gmail.com", "Timisoara, str Scolii bl 32b sc A ap5 ");
insert into utilizatori (id, nume_utilizator, email_utilizator, adresa_livrare)
value (6, "DUMITRU SABIN", "dumitrusabin.2@yahoo.com", "Iasi, str Mihai Viteazu, nr 45");
```

Inserare date tabela comenzi

```
insert into comenzi (id, id_client, data_comanda)
value (1, 3, "2024-03-21");
insert into comenzi (id, id_client, data_comanda)
value (2, 5, "2024-03-22");
insert into comenzi (id, id_client, data_comanda)
value (3, 1, "2024-03-22");
insert into comenzi (id, id_client, data_comanda)
value (4, 6, "2024-03-26");
insert into comenzi (id, id_client, data_comanda)
value (5, 2, "2024-03-27");
insert into comenzi (id, id_client, data_comanda)
value (6, 4, "2024-03-27");
```

Inserare date tabela detalii_comenzi

```
insert into detalii_comenzi (id, id_comanda, id_produs, cantitate, pret_total)
value (1, 1, 3, 1, 73 );
insert into detalii_comenzi (id, id_comanda, id_produs, cantitate, pret_total)
value (2, 2, 4, 2, 164 );
insert into detalii_comenzi (id, id_comanda, id_produs, cantitate, pret_total)
value (3, 3, 5, 1, 480 );
insert into detalii_comenzi (id, id_comanda, id_produs, cantitate, pret_total)
value (4, 4, 1, 1, 420 );
insert into detalii_comenzi (id, id_comanda, id_produs, cantitate, pret_total)
value (5, 5, 2, 2, 346 );
insert into detalii_comenzi (id, id_comanda, id_produs, cantitate, pret_total)
value (6, 6, 7, 1, 859 );
```

Inserare date tabela gestiune_produce

```
insert into gestiune_produce (id, id_produs, cantitate_intrari, cantitate_iesiri, stoc_depozit)
value (1, 1, 23, 1, 22);
insert into gestiune_produce (id, id_produs, cantitate_intrari, cantitate_iesiri, stoc_depozit)
value (2, 2, 43, 2, 41);
insert into gestiune_produce (id, id_produs, cantitate_intrari, cantitate_iesiri, stoc_depozit)
value (3, 3, 74, 1, 73);
insert into gestiune_produce (id, id_produs, cantitate_intrari, cantitate_iesiri, stoc_depozit)
value (4, 4, 22, 2, 20);
insert into gestiune_produce (id, id_produs, cantitate_intrari, cantitate_iesiri, stoc_depozit)
value (5, 5, 18, 1, 17);
insert into gestiune_produce (id, id_produs, cantitate_intrari, cantitate_iesiri, stoc_depozit)
value (6, 6, 80, 0, 80);
insert into gestiune_produce (id, id_produs, cantitate_intrari, cantitate_iesiri, stoc_depozit)
value (7, 7, 37, 1, 36);
```

Schimbarea tipului variabilei din varchar in date

```
alter table comenzi  
change data_comanda data_comanda date;
```

Schimbarea dimensiunii maxime a denumirii variabilei descriere_produc

```
alter table produse  
change descriere_produc descriere_produc varchar(250);
```

Update al pretului de vanzare a produsului cu id 1 din tabela produse

```
update produse set pret_vanzare=450 where id=1;
```

Update al pretului comenzi cu id 4

```
update detalii_comenzi set pret_total=450 where id=4;
```

Afisare tabela produse

```
select * from produse;
```

Stergerea comenzii cu id 3 din tabela detalii comenzi

```
delete from detalii_comenzi where id=3;
```

Afisarea email-ului si adresa de livrare a utilizatorilor

```
select email_utilizator, adresa_livrare from utilizatori;
```

Afisarea numelui produselor cu pretul de vanzare mai mic decat 100 din tabela produse

```
select nume_produc from produse where pret_vanzare < 100;
```

Afisarea produselor cu pret de vanzare cuprins
intre 100 si 200

```
select * from produse where pret_vanzare > 100 and pret_vanzare < 200;
```

Afisarea produselor unde pretul de vanzare nu este mai mare de 450 sau stocul disponibil este mai mic decat 50 si fac parte
din categoria produselor ce incep cu "Jucarii"

```
select * from produse where not pret_vanzare > 450 or stoc_disponibil < 50 and categorie_produs like "Jucarii%";
```

Numaram cati utilizatori avem

```
select count(*) from utilizatori;
```

Afisarea datei ultimelor comenzi

```
select max(data_comanda) from comenzi;
```

Afisare pretului mediu/comanda din tabela detalii comenzi

```
select avg(pret_total) from detalii_comenzi;
```

Calcularea valorii totale a comenzilor

```
select sum(pret_total) from detalii_comenzi;
```

Afisarea id-ului comenzilor minime, medii si maxime din
tabela detalii comanda

```
select id_comanda,  
       avg(pret_total) as "Pret mediu",  
       min(pret_total) as "Pret minim",  
       max(pret_total) as "Pret maxim"  
from detalii_comenzi  
group by id_comanda;
```


SUBQUERI

Afisarea numelui si pretului de vanzare a produselor cu pretul de vanzare **peste medie**.

```
select nume_produc, pret_vanzare from produse  
where pret_vanzare > (select avg(pret_vanzare) from produse);
```

Afisarea produselor comune **dintre** tabela produse si tabela detalii comenzi

```
select nume_produc from produse inner join detalii_comenzi  
on produse.id=detalii_comenzi.id_produc;
```

Afisarea produselor din tabela produse ce **se regasesc** si in tabela detalii comenzi

```
select nume_produc from produse left join detalii_comenzi  
on produse.id=detalii_comenzi.id_produc;
```

Afisarea produselor din tabela detalii comenzi ce **se regasesc si** in tabela produse

```
select nume_produc from detalii_comenzi left join produse  
on produse.id=detalii_comenzi.id_produc;
```

Afisarea numelor produselor **si** pretul de vanzare **din** tabela produse **si** tabela gestiune produse

```
select nume_produc, pret_vanzare from produse cross join gestiune_produse;
```

MULTUMESC ECHIPEI *IT FACTORY* PENTRU INDRUMARE SI SUPORT!

link GitHub <https://github.com/IolandaSterie/Myshop>