

Proiect final

Craciun Ioana

04.09.2024

Partea I

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

Cerintele de business sunt documente care conțin felul în care produsul ar trebui să funcționeze conform dorințelor clientului și care vor sta la baza felului în care produsul este dezvoltat și testat. Sunt create de către client.

2. Explicați diferența între un test condition și test case

Test condition-condiția pe care trebuie să o îndeplinească funcționalitatea testată pentru a putea fi considerată corectă (ce testăm?)

Test case-set de instrucțiuni care trebuie urmate pas cu pas pentru a putea valida o anumită funcționalitate (cum testăm?)

3. Enumerați și explicați pe scurt etapele procesului de testare

Planificare

- Include activități care definesc obiectivele și abordarea cu privire la testare
- În această etapă se decide care părți ale aplicației se dorește să fie testate
- 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 programează activitățile și etapele procesului de testare
- 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

Analiza

- În această etapă se analizează documentația primită de la client pentru a ne asigura că le înțelegem, că nu exista greșeli, ambiguități, inconsistențe, neconcordanțe, contradicții etc. Putem de asemenea să facem sugestii de îmbunătățire asupra cerințelor, iar clientul va decide dacă sugestiile noastre sunt utile pentru el sau nu. Defectele care sunt găsite în cerințe sunt mai ușor de fixat și mai ieftin decât dacă ar fi găsite în cod
- De asemenea, în aceasta etapă se generează condițiile de testare

Design

- În aceasta etapă se creează cazurile de testare și se identifică datele de testare
- De asemenea se identifică datele de testare de care avem nevoie și se face design-ul mediului de testare prin identificarea oricărui tool sau infrastructură de care avem nevoie pentru testare.

Implementare

- În aceasta etapă se creează datele de testare identificate în etapa anterioară
- Se validează mediul de test prin intermediul smoke testing
- Se prioritizează testele și se creează datele de testare
- Se grupează testele pe baza obiectivelor lor (testare funcțională, testare de regresie, testare de acceptanță etc)
- Tot în etapa aceasta ne asigurăm că avem tot ce ne trebuie pentru a începe testarea propriu-zisă

Executie

- În această etapă cazurile de testare sunt executate
- Rezultatele sunt raportate în tool-ul în care au fost scrise testele
- 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 se va face si testare de regresie, pentru a ne asigura că schimbările făcute nu au avut un impact negativ asupra funcționalităților existente

Inchiderea

- În această etapă se evaluează criteriile de ieșire pentru a ne asigura că putem sa închidem procesul de testare în siguranță
- Orice taskuri rămase deschise și buguri sunt reevaluate și ulterior închise
- Materialele de testare sunt predate și arhivate (handover of testware)
- Este generat un raport de închidere a testării care ulterior este trimis catre stakeholders
- Se analizează lecțiile învățate în urma procesului de testare pentru a evalua schimbările necesare într-un proiect similar viitor
- Se identifică riscurile de produs (dacă există) și se raporteaza către client

4. Explicați diferența între retesting și regression testing

Retesting este un tip de testare prin care se verifică dacă defectele marcate ca și remediate au fost într-adevăr remediate.

Regression testing e un tip de testare prin care se verifică programul sau o parte din program pentru a ne asigura că schimbările aduse asupra lui nu au cauzat alte defecte.

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

Testare funcțională – Ce trebuie să facă produsul? – Verifică dacă produsul își îndeplinește funcțiile.

Testarea non-funcțională – Cum trebuie să se comporte produsul? – Verifică attribute care descriu cât de bine își îndeplinește sistemul funcțiile.

6. Explicați diferența între blackbox testing și whitebox testing

Pentru testarea blackbox nu este necesara cunoasterea codului,pe cand testarea whitebox presupune cunoasterea codului sursa pe baza caruia actioneaza programul.

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

Tehnici de testare whitebox: Statement Coverage si Decision Coverage

Tehnici de testare blackbox: Equivalence Partitioning(EP), Boundary Value Analysis(BVA), State Transition Testing(STT), Decision Table(DT)

Tehnici de testare experience-based: Testare Ad-hoc, Ghicirea Erorilor, Testare Exploratorie.

8. Explicați diferența între verification și validation

Verificarea se face cu scopul de a evalua materialele care stau la baza testării pentru a ne asigura ca produsul e construit in mod corect,iar validarea se face cu scopul de a evalua produsul finit si a ne asigura ca acesta indeplineste cerintele de business si nevoile utilizatorului.

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

Positive testing-testarea sistemului cu valori pe care ar trebui sa le poata procesa.

Negative testing-testare cu valori pe care sistemul nu ar trebui sa le poata procesa in mod normal pentru a ne asigura ca aceste valori sunt respinse si nu cauzeaza un crash al sistemului.

De exemplu daca avem voie sa introducem valori numerice cuprinse intre 0-10, pentru testarea pozitiva vom introduce valoarea 2 si pentru testarea negativa valoarea 20,-5,a .

10. Enumerați și explicați pe scurt nivelurile de testare

- Testarea unitara-testarea celei mai mici bucati functionale dintr-o aplicatie
- Testare de integrare-se concentreaza pe interactiunile dintre componente si sisteme
- Testare de sistem-se concentreaza pe comportamentul si compatibilitatea sistemului ca un tot-unitar ,tinand cont de comportamentul end-to-end al functionalitatilor pe care sistemul trebuie sa le execute si de comportamentul non-functional asteptat al acelor taskuri
- Testare de acceptanta-se concentreaza pe comportamentul produsului si verifica felul in care acesta indeplineste nevoile clientului/utilizatorului.

Partea a II-a

Am creat o baza de date pentru un club sportiv ce cuprinde mai multe tabele:

- Antrenori
- Date_copii
- Date_parinti
- Echipamente
- Taxa

```
insert into Date_parinti values
(1, 'Craciun', 'Ioana', '0773974717', 1),
(2, 'Badea', 'Lucian', '0751127286', 2),
(3, 'Bucatariu', 'Madalina', '0747222111', 7),
(4, 'Neagu', 'Razvan', '0778555111', 5),
(5, 'Mihai', 'Razvan', '0747555444', 6),
(6, 'Craciun', 'Adrian', '0777125286', 1);
```

Instructiune pentru adaugarea valorilor in tabelul Date_parinti

```
create database Club_de_fotbal;
create table Date_copii(
ID int unsigned auto_increment primary key,
Nume varchar(20) not null,
Prenume varchar(20) not null,
Data_nasterii date not null
);
```

Instructiuni pentru crearea bazei de date si a unui tabel



Baza de date cu tabelele aferente

```
drop table Meciuri;  
Alter table Date_parinti add foreign key (ID_copil) references Date_copii(ID_copil);
```

Cu ajutorul instructiunii 'DROP' am sters tabelul 'Meciuri'.

Prin instructiunea 'ALTER TABLE' am modificat tabelul 'Date_parinti' si am adaugat o cheie secundara, facand astfel legatura intre cele 2 tabele: 'Date_parinti' si 'Date_copii'.

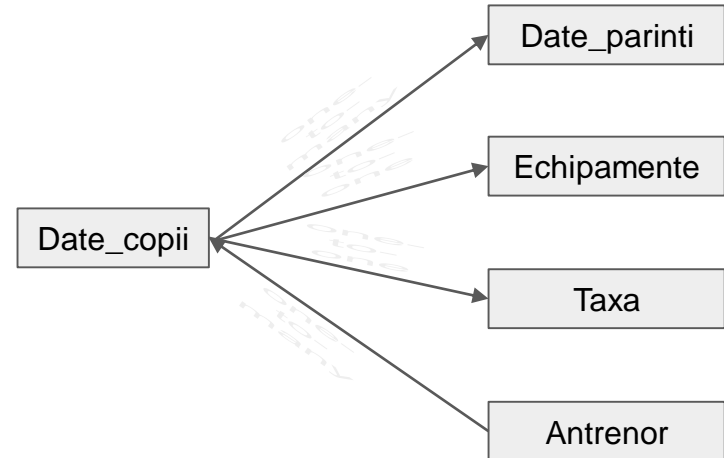
```
create table Echipamente  
(ID int primary key not null,  
  Marime varchar(10),  
  Nume_inscriptionat varchar(10),  
  Numar_inscriptionat int,  
  ID_copil int not null,  
  foreign key (ID_copil) references Date_copii(ID_copil));
```

Cheie primara

Cheie secundara

ID din tabelul Echipamente este cheie primara, deci este indicator unic si campul acesta nu poate fi lasat fara o valoare.

Cheia secundara ID_copii din tabelul Echipamente face legatura cu tabelul Date_copii. Astfel putem sa avem o evidenta a copiilor care au echipamente, ce nume si numar au inscriptionat pe ele.




```
select Nume,Prenume from Date_copii order by Nume;
```

Aceasta instructiune afiseaza numele si prenumele copiilor in ordine alfabetica.

Nume	Prenume
Babos	Patrick
Badea	Filip
Bucatariu	David
Craciun	Sebastian
Gache	Razvan
Mihai	Gabi
Muresan	Patrick
Neagu	Teodor
Pruteanu	Andrei

```
select * from Date_copii order by Date_copii.data_nasterii;
```

Prin aceasta secventa de cod se afiseaza toate datele din tabelul 'Date_copii' in ordine crescatoare a datei de nastere a copiilor.

ID_copil	Nume	Prenume	Data_nasterii	ID_antrenor
2	Badea	Filip	2019-01-17	3
3	Pruteanu	Andrei	2019-01-18	1
4	Gache	Razvan	2019-01-18	3
1	Craciun	Sebastian	2019-01-19	1
5	Neagu	Teodor	2019-02-25	1
6	Mihai	Gabi	2019-03-12	1
7	Bucatariu	David	2019-10-10	3
8	Muresan	Patrick	2019-10-14	3
9	Babos	Patrick	2019-12-12	2

```
select Nume,prenume,ID_antrenor from Date_copii where id_antrenor=2;
```

Nume	prenume	ID_antrenor
Babos	Patrick	2

Aceasta instructiune returneaza numele,prenumele copilului si id-ul antrenorului,daca id-ul antrenorului este 2

```
select Nume_antrenor,Prenume_antrenor from Antrenori where Nume_antrenor like '%an';
```

Nume_antrenor	Prenume_antrenor
Ciocan	Cristian

Prin aceasta secventa de cod se returneaza numele si prenumele antrenorului,pentru cei al caror nume se termina cu "-an".

```
select * from Date_copii cross join Date_parinti on Date_copii.ID_copil=Date_parinti.ID_copil;
```

ID_copil	Nume	Prenume	Data_nasterii	ID_antrenor	ID	Nume	Prenume	Numar_telefon	ID_copil
1	Craciun	Sebastian	2019-01-19	1	1	Craciun	Ioana	0773974717	1
2	Badea	Filip	2019-01-17	3	2	Badea	Lucian	0751127286	2
7	Bucatariu	David	2019-10-10	3	3	Bucatariu	Madalina	0747222111	7
5	Neagu	Teodor	2019-02-25	1	4	Neagu	Razvan	0778555111	5
6	Mihai	Gabi	2019-03-12	1	5	Mihai	Razvan	0747555444	6
1	Craciun	Sebastian	2019-01-19	1	6	Craciun	Adrian	0777125286	1

Cu ajutorul instructiunii 'CROSS JOIN' am facut legatura intre tabelul 'Date_copii' si 'Date_parinti'.

Am afisat toate datele cumulate din cele doua tabele.

```
select * from Date_copii inner join Echipamente on Date_copii.ID_copil=Echipamente.ID_copil;
```

ID_copil	Nume	Prenume	Data_nasterii	ID_antrenor	ID	Marime	Nume_inscriptionat	Numar_inscriptionat	ID_copil
1	Craciun	Sebastian	2019-01-19	1	1	s	Sebi	19	1
3	Pruteanu	Andrei	2019-01-18	1	2	XS	Andrei	12	3
7	Bucatariu	David	2019-10-10	3	3	M	David	11	7
6	Mihai	Gabi	2019-03-12	1	4	s	Gabi	15	6
8	Muresan	Patrick	2019-10-14	3	5	s	Patrick	10	8
5	Neagu	Teodor	2019-02-25	1	6	xs	Teo	1	5

Am folosit instructiunea 'INNER JOIN' pentru a returna toate datele din tabelu 'Date_copii' si 'Echipamente' pentru copiii care au echipamente.

```
select * from Date_copii left join Date_parinti on Date_copii.ID_copil=Date_parinti.ID_copil where Date_parinti.Nume='Craciun';
```

ID_copil	Nume	Prenume	Data_nasterii	ID_antrenor	ID	Nume	Prenume	Numar_telefon	ID_copil
1	Craciun	Sebastian	2019-01-19	1	1	Craciun	Ioana	0773974717	1
1	Craciun	Sebastian	2019-01-19	1	6	Craciun	Adrian	0777125286	1

Cu ajutorul functiei 'LEFT JOIN' am returnat toate datele din tabelul 'Date_copii' si 'Date_parinti' doar pentru copilul care are parinti al caror nume este 'Craciun'.

https://github.com/loanaC90/proiect_1/blob/main/Club_fotbal.md

https://github.com/loanaC90/proiect_1/blob/main/proiect%20club%20de%20fotbal_v3.sql

Va multumesc pentru atentie!