

FISA CERINTELOR

-Test-

Cuprins

1. Descriere
2. Domenii
3. Actionari & Interese
4. Actori & Obiective
5. Scenarii de utilizare
 - 5.1 Testarea butoanelor aplicației
 - 5.2 Testarea salvarii informatiilor
 - 5.3 Testarea corectitudinii fisei
 - 5.4 Testare functionalitate upload
 - 5.5 Testare redimensionare diagrama
 - 5.6 Testare salvare diagrama

1. Descriere

Proiectul consta in crearea unei aplicatii web care pune la dispozitie utilizatorilor unelte necesare obtinerii unei fise a cerintelor.

Echipa de testare trebuie sa verifice daca exista bug-uri in aplicatie. Ei trebuie sa testeze codul aplicatiei si sa verifice fiecare modul de functionare a aplicatiei. Daca se intampina dificultati in utilizarea aplicatiei se atentioneaza ceilalti programatori care se ocupa cu acea parte din proiect pentru a remedia problema. Scopul echipei de testare este de a gasi fiecare problema care poate aparea in timpul utilizarii aplicatiei pentru a-i oferi clientului o aplicatie functionala.

2. Domenii

Se vor descrie scenariile de utilizare la nivelul modulului corespunzator.

3. Actionari & Interese

Echipa de testeri: preia informatiile oferite de celelalte echipe cu scopul de a verifica daca aplicatia functioneaza cum trebuie.

Celelalte echipe: au certitudinea ca ceea ce au facut pana acum functioneaza si pot continua dezvoltarea aplicatiei.

Utilizatorul aplicatiei: dispune de o aplicatie care functioneaza corect si pe care o poate folosi in vederea generarii fisei cerintelor.

4. Actori & Obiective

Echipa de testeri: sa verifice daca aplicatia functioneaza corect

Celelalte echipe: sa ofere clientului mijloacele necesare pentru a genera fisa cerintelor

Utilizatorul aplicatiei: sa poata genera fisa cerintelor dorita

5.Scenarii de utilizare

5.1 Testarea butoanelor aplicației

5.1.1 Obiectiv/Context

Testarea reactiei aplicatiei atunci cand se da click pe un buton.

5.1.2 Scenariu/Pasi

1. Se deschide aplicatia.
2. Se da un singur click pe un buton pentru vedea daca aplicatia reactioneaza conform cerintelor, pasul se repeta pentru fiecare buton in parte (exemplu: daca facem click pe butonul SUBMIT, se va deschide o noua pagina in care avem mai multe coloane ce trebuie completate).
3. Se fac mai multe click-uri pe un buton pentru vedea daca aplicatia reactioneaza conform cerintelor, incarcand pagina o singura data intr-un timp scurt, pasul se repeta pentru fiecare buton in parte (exemplu: daca facem click de butonul SUBMIT se va incarca o singura pagina).
4. Se fac mai multe click-uri pe butoane diferite pentru a se verifica daca se ia ultima comanda pe care o cerem; pasul se repeta de mai multe ori, cu combinatii de butoane diferite. (exemplu: se face click pe butonul de SUBMIT si imediat dupa pe butonul de REVIEW, aplicatia ne deschide o noua pagina pentru comanda de REVIEW).

5.1.3 Extensii

- 2.a. Daca se face click pe un buton si nu se deschide pagina dorita, atunci se completeaza în raport diferența dintre comportamentul dorit și comportamentul actual. (exemplu: daca facem click pe butonul SUBMIT si se efectueaza comanda de REVIEW).

2.b Dacă se face click pe un buton și nu se întâmplă nimic atunci atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual. (exemplu: dacă face click pe butonul de SUBMIT și nu se întâmplă nimic din cauza lipsei vitezei de responsivitate (serverul este suprasolicitat cu requesturi/ alt bug care împiedică funcționarea corectă).

2.c Dacă se face click pe un buton și nu se întâmplă nimic iar butonul devine inactiv (exemplu: se face click pe butonul de SUBMIT, nu suntem redirecționați la pagina corespunzătoare și butonul de SUBMIT devine inaccesibil) atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual.

3.a. Dacă se fac mai multe click-uri pe un buton și pagina nu începe să se încarce de fiecare dată când apasăm de buton, atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual.

3.b. Dacă se fac mai multe click-uri pe un buton și pagina începe să se încarce de fiecare dată când apasăm de buton după care butonul se blochează și programul da crash (rămâne într-un loop de refresh/ pagina rămâne neîncărcată/ se primește un cod de eroare gen 404), atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual.

4.a. Dacă se fac mai multe click-uri pe butoane diferite și se deschide prima comandă efectuată, atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual.

5.2 Testarea salvării informațiilor

5.2.1 Obiectiv/Context

Se testează dacă datele pe care le introducem sunt salvate corect.

5.2.2 Scenariu/Pasi

1. Se deschide aplicația
2. Se da click pe câmpul în care dorim să introducem textul.
3. Se introduce textul dorit și se da click pe butonul SAVE, astfel se verifică dacă datele sunt salvate.

4. După ce am introdus un text, se va deschide din nou același câmp și se va introduce un alt text, astfel se verifică dacă se salvează ultimul text introdus.
5. După ce am introdus un text, se va deschide din nou câmpul dorit și se va șterge textul, astfel testăm dacă informația va fi ștearsă.

5.2.3 Extensii

- 2.a Dacă se încearcă să se adauge un input în câmpul pe care s-a dat click și nu se pot introduce datele dorite de către utilizator atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual.
- 3.a. Dacă după ce facem click pe butonul SAVE nu se salvează textul introdus de noi, atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual.
- 3.b. Dacă introducem un text folosind alt alfabet în afara de cel englez (de exemplu alfabetul roman) și caracterele care nu aparțin alfabetului englez nu sunt recunoscute și nu sunt salvate (sau înlocuite cu "?"), atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual.
- 4.a. Dacă modificăm textul de pe un câmp deja completat și nu se salvează modificarea făcută, atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual.
- 5.a. Dacă dorim să ștergem textul de pe un câmp deja completat și textul rămâne încă salvat, atunci se completează în raport diferența dintre comportamentul dorit și comportamentul actual.

5.3 Testarea corectitudinii fișei

5.3.1 Obiectiv/Context

Se testează dacă fișa rezultată corespunde cu inputul din formulare

5.3.2 Scenariu/Pasi

1. Se deschide aplicația.
2. Se completează câmpurile dorite.
3. Se apasă butonul SUBMIT și se verifică dacă în fișa rezultată se regăsește tot ce am introdus în formulare

4. Se apasa butonul GENERATE DIAGRAM și se verifica dacă diagrama generata de aplicație corespunde cu datele introduse în formular.

5.3.3 Extensii

3.a. Dacă datele introduse nu corespund cu inputul dat de utilizator, atunci se completeaza în raport diferența dintre comportamentul dorit și comportamentul actual.

4.a. Dacă diagrama generata de aplicație nu corespunde cu inputul dat de utilizator, atunci se completeaza în raport diferența dintre comportamentul dorit și comportamentul actual.

5.4 Testare functionalitate upload

5.4.1 Obiectiv/Context

Se testează dacă aplicația genereaza fisa cerintelor corect daca datele sunt preluate dintr-un document XML

5.4.2 Scenariu/Pasi

1. Se deschide aplicația.
2. Se apasa butonul OPEN.
3. Se deschide o fereastră în care utilizatorul poate sa aleaga fișierul pe care vrea sa îl uploadeze.
4. Se uploadeaza fișierul.
5. Se verifica dacă datele din fisier au fost preluate corect în fisa generata.

5.4.3 Extensii

3.a. Dacă nu se deschide fereastră se apasa din nou OPEN, sau se completeaza în raport diferența dintre comportamentul dorit și comportamentul actual.

4.a. Dacă fișierul nu este primit de către aplicație, atunci se verifica dacă acesta este de tip XML.

4.b. Dacă fisierul este validat chiar dacă nu este de tip XML, atunci se completeaza în raport diferența dintre comportamentul dorit și comportamentul actual.

5.a Daca datele nu corespund, se incearca din nou uploadarea fisierului.

5.5 Testare redimensionare diagrama

5.5.1 Obiectiv/Context

Se verifica daca actiunile asupra diagramei SVG obtinuta ca output al aplicatiei functioneaza.

5.5.2 Scenariu/Pasi

1. Se obtine diagrama SVG din continutul XML livrat de componenta Controller.
2. Utilizatorul pozitioneaza cursorul in zona pe care vrea sa o vizualizeze mai detaliat.
3. Utilizatorul foloseste zoom in/zoom out.

5.5.3 Extensii

- 1.a Continutul XML nu este parsat corect, deci diagrama SVG nu corespunde cerintelor utilizatorului
- 3.a Zoom-ul este pus asupra altei portiuni decat cea aleasa.
- 3.b Zoom-ul nu functioneaza deloc.
- 3.c Zoom-ul functioneaza, insa diagrama nu se redeseneaza si astfel avem o imagine mult marita si pixelata.

5.6 Testare salvare diagrama

5.6.1 Obiectiv/Context

Se doreste salvarea diagramei la nivel local.

5.6.2 Scenariu/Pasi

1. Dupa crearea diagramei, utilizatorul apasa butonul Save pentru a salva diagrama.
2. Se introduce un nume.
3. Se alege locatia unde va fi salvata diagrama si se salveaza.

5.6.3 Extensii

2.a. Numele fisierului contine caractere invalide.

2.b Exista deja un fisier cu acelasi nume.

3.a Se salveaza eronat (de exemplu: nu se salveaza diagrama completa/ se salveaza doar portiunea in care este dat zoom in).

3.b Nu se salveaza diagrama.