

Grupa A1



Gestionarea lucrarilor de licenta

Plan de asigurare a calitatii software

Cuprins

1. Introducere
2. Obiective
3. Scop
4. Strategia de testare
5. Referinte
6. Progresul testarii
7. Criterii pass/fail
8. Factori de risc
9. Aprobări

1. Introducere

Planul de testare al aplicatiei de gestionare a licentelor este un document ce descrie scopul, modul in care va fi abordata testarea si "programul" in care se va efectua testarea. Pe langa alti itemi de testare, se identifica functiile si functionalitatile ce vor fi sau nu testate, sarcinile de testare, asignari de persoane pentru fiecare sarcina, teste in diferite medii, testare a tehnicilor de design, criterii de pass/fail ce vor fi utilizate, motivul alegerii lor si estimarea riscurilor implementarii software(estimarea costurilor). Acesta reprezinta o inregistrare a procesului de testare.

2. Obiective

Obiectivele planului de test sunt acelea de a defini diverse strategii precum si uneltele necesare pentru a finaliza ciclul de testare al proiectului.

3.Scop

3.1 Caracteristici ce vor fi testate

- ❖ Functionalitatea paginii de inregistrare.
- ❖ Existenta datelor de logare in baza de date, precum si a functionalitatii paginii de logare.
- ❖ Functionalitatea paginii de Editare Profil.
- ❖ Existenta doar a datelor publicatiei.
- ❖ Inregistrarea unei noi citari.
- ❖ Inregistrarea unei noi conferinte.
- ❖ Inregistrarea unui proiect de ceretare.
- ❖ Vizibilitatea punctajelor/articolelor si drepturile fiecarui tip de utilizator.
- ❖ Posibilitatea modificarii normelor de punctare in cazul in care utilizatorul nu are acest drept.
- ❖ Generarea raportului.
- ❖ Functionalitatea barei de search.

3.2 Caracteristici ce nu vor fi testate

- ❖ Functiile de Get si Set din codul sursa.

4.Strategii de testare

4.1 Tipuri de testare:

- ❖ Testare manuala: Testam manual utilitatea si vizibilitatea fiecarui buton, label, formular existente in aplicatie, schema de culori, pozitionarea butoanelor, introducerea in mod corespunzator a textelor
- ❖ Testare automata: Testam automat functiile
- ❖ Testarea performantei: Testam robustetea aplicatiei, precum si consistenta, eficienta si completitudinea acesteia
- ❖ Testarea utilizabilitatii: Testam cat de usor este de folosit aplicatia, cati pasi sunt necesari pentru a face o anumita sarcina.
- ❖ Testarea Alpha: Va fi efectuat de catre client.

4.2 Unelte

Nume	Companie	Versiune
Junit	Java	junit4

5.Referinte

Locatia: <https://github.com/silvia95/ProiectIP> . Aici clientii pot vizualiza progresul dezvoltarii aplicatie.

6.Progresul testarii

6.1 Tasks

- ❖ Construirea fisei cerintelor
- ❖ Generarea de diagrame UML
- ❖ Generarea rapoartelor de testare manuala
- ❖ Stabilirea cazurilor de testare
- ❖ Efectuarea testarii automate
- ❖ Generarea planului de testarea
- ❖ Construirea raportului final de testare

6.2 Responsabilitati

- ❖ Partea vizuala este construita de echipa de front-end.
- ❖ Partea din spate(mecanismul aplicatiei) este realizata de echipa de back-end.
- ❖ Testarea manuala si testarea QA(Junit) este facuta de echipa de Testing.
- ❖ Integrarea este realizata in procentaj de 90% de echipa de back-end, celalat 10% fiind realizat de echipa de testare.
- ❖ Taskurile sunt distribuite, in fiecare echipa, intern de catre coordonatorul echipei.

6.3 Deliverable

Deliverable	Responsability
Test Design Document, Test plan document	Testare unit white-box – covers white testing criteria, methods and test cases
Test report document	System Test report – covers system test results, problems, summary and analysis

6.4 Orar

6.4.1 Faza de planificare

- ❖ A noua saptamana a semestrului 2
- ❖ Stabilirea cerintelor aplicatiei
- ❖ Distribuirea echipelor

6.4.2 Modul de planificare

- ❖ Organizare saptamanala a task-urilor si distribuirea lor per membru echipa.

6.4.3 Infrastructura

- ❖ Saptamana 7: Realizarea fisei cerintelor
- ❖ Saptamana 8: Generarea de diagrame si crearea raportului de test v1.0
- ❖ Saptamanile 9-10: Generearea planului de testare, cazurilor de testare si crearea raportului de test v1.1
- ❖ Saptamana 11: finalizarea planului de testare, Testarea automata a functiilor.

6.4.4 Test de regresie

❖

6.4.5 Beta

- ❖ Faza Beta trebuie sa fie gata cu 1-2 saptamani inainte de prezentare pentru descoperirea posibilelor erori omise in testarea alpha.

7. Criterii pass/fail

7.1 Criterii de intrare

- ❖ Resursele QA au inteles complet cerintele.
- ❖ Resursele de asigurare a calității au cunoștințe solide de funcționalitate în rapoarte.

7.2 Criterii de suspensie

- ❖ Constructia contine multe defecte grave sau posibil sa limiteze progresul testului
- ❖ Probleme Hardware/Software

7.3 Criterii de reluare

- ❖ Reluarea va avea loc numai atunci când problema care a cauzat suspendarea a fost rezolvata.

7.4 Criterii de iesire

- ❖ Functionalitatile principale trebuie sa fie disponibile si sa corespunda fisei cerintelor.
- ❖ Erorile majore au fost remediate.
- ❖ Documentatia este updatata.
- ❖ Raportul de test a fost inregistrat.

8. Factori de risc

- ❖ Neefectuarea proiectului la timp
- ❖ Nerespectarea cerintelor de test

9.Aprobari

Profesor responsabil: Alex Moruz

Echipa de testare:

- Bunea Daniel(coordonator echipa)
- Braescu Ana-Maria
- Dogaru Laurentiu
- Moniry Abianeh Daniel
- Pantaru Codrut
- Ungureanu Silvia-Ana-Maria