**Web based Document Processing Manager**

**Raport de Testare**

**Echipa: Batmanii**

**Titlu proiect: ocrWebManager**

**1. Scopul documentului**

Scopul acestui document este de a prezenta un sumar al rezultatelor obținute în urma rulării unor suite de teste.

**2. Sumar**

Nume proiect: ocrWebManager

Echipa: Batmanii

**2.1 Mod de testare**

În cadrul acestui proiect, s-a realizat o testare atât pe module a aplicației, cât și funcționarea sa per ansamblu, din interfața grafică.

În cazul testării pe module s-a utilizat framework-ul pentru testare unitară JUnit pentru Java.

Au fost utilizate cazuri de test pentru condiții de intrare atât valide, cât și invalide (aplicația să facă ce trebuie și să nu facă ce nu trebuie).

**2.2 Teste efectuate**

Pentru testarea pe module, s-a creat în proiectul ocrwebmanager pachetul com.mps.batmanii.ocrWebManager.tests, care conține câte o clasă de test pentru fiecare modul testat. Fișierele de test sunt disponibile în folderul resources din src/test.

1) TestClassXsd: clasa de test pentru fișierele Xsd.

S-au utilizat 30 de fișiere de test Xsd, dintre care 18 cu intrări valide și 12 cu intrări invalide. Testele au fost utilizate cu scopul de a verifica parsarea fișierelor Xsd. Se compară rezultatele generate în urma rulării programului cu fișierele originale.

S-au testat:

* lungimea listei de fișiere Xsd de parsat
  + programul generează rezultate corecte pentru toate testele rulate
* afișarea fișierelor Xsd parsate
  + rezultatele rulării în cazul intrărilor valide au arătat că fișierele Xsd sunt parsate corect, dar s-a remarcat următorul aspect:

|  |  |
| --- | --- |
| Rezultate generate la rularea programului | Fișiere originale |
| ComplexType => attribute =>  **use = null** | ComplexType => attribute =>  **use = ”required”** |

* + în cazul intrărilor invalide, programul a trecut testele: nu s-a realizat parsarea fișierelor de test, generându-se mesaje de eroare că schema nu este construită corect
  + mențiune: programul nu mai funcționează corect dacă în folderul destinat schemelor sunt fișiere cu alte extensii

2) TestClassXml: clasa de test pentru fișierele Xml.

S-au realizat 10 teste cu scopul de a verifica crearea de fișiere Xml. S-au comparat fișierele generate în urma rulării programului cu fișierele originale.

Aplicația a trecut toate testele, generând fișiere Xml corecte și conform parametrilor setați în test.

3) TestClassExec: clasa de test pentru fișierele executabile

S-au utilizat 6 fișiere de test.

Testele au verificat:

* lungimea listei de executabile
  + programul generează rezultate corecte pentru toate testele rulate
* afișarea numelor executabilelor și tipurilor lor
  + programul generează rezultate corecte pentru toate testele rulate
  + programul identifică corect fișierele executabile
    - fișierele cu alte extensii nu sunt luate în considerare
    - programul funcționează corect chiar dacă în folderul execs se află și fișiere neexecutabile

**2.3 Testarea per ansamblu a aplicației, din interfața grafică**

Aplicația a fost rulată în două moduri:

1) Folosind propriul server tomcat

* eroare în cazul accesării paginii de completare a parametrilor pentru executabilul selectat => problemă fixată
* Explicație: dwr-ul folosit este blocat de tomcat din cauza cookie-urilor. Astfel, se adaugă în fișierul context.xml, usehttponly = "false". Pentru skip la verificări în jsp-uri se procedează astfel: se adaugă în catalina.properties: org.apache.el.parser.SKIP\_IDENTIFIER\_CHECK = true.
* Aceste fișiere se găsesc în folderul conf din folderul în care a fost instalat tomcat.

2) Folosind tomcat-ul maven-ului

Pentru testarea aplicației, s-au utilizat fișierele .tif puse la dispoziție pe site-ul de cursuri.

Aplicația a fost testată pe mai multe tipuri de browser-e Web: Internet Explorer, Mozilla Firefox și Google Chrome.

S-au testat:

* funcționalitatea ”Browse”: aplicația funcționează corect pentru testele realizate
* funcționalitatea ”Upload”: aplicația funcționează corect pentru testele realizate
* funcționalitatea ”Proceed to selecting executables”: aplicația funcționează corect pentru testele realizate
* funcționalitatea ”Reinitialize”: aplicația funcționează corect pentru testele realizate
* funcționalitatea de selectare a executabilelor: aplicația funcționează corect pentru executabilele valabile pentru toate testele realizate
* funcționalitatea de completare a parametrilor executabilelor: aplicația funcționează corect pentru toate testele realizate
  + parametrii inputFile și outputFile nu se mai introduc, ci sunt adăugați automat de către aplicație
  + la completarea parametrilor în interfață, nu se poate selecta între valorile posibile ale enumerațiilor din schemele Xsd cu enumerații => problemă fixată
  + această funcționalitate nu funcționează corect în Internet Explorer: apare un câmp suplimentar ”task”
* funcționalități ”Back”: aplicația funcționează corect pentru testele realizate
  + această funcționalitate nu funcționează corect în Internet Explorer, pentru revenirea din pagina de completare a parametrilor executabilelor
* funcționalitatea ”Modify”: aplicația funcționează corect pentru testele realizate
* funcționalitatea ”Delete”: aplicația funcționează corect pentru testele realizate
* funcționalitatea ”Back”: aplicația funcționează corect pentru testele realizate
* funcționalitatea ”Process”: aplicația funcționează corect pentru testele realizate
  + este implementată funcționalitatea pentru cazul în care nu s-a selectat niciun executabil, însă este dezactivată din cauza lipsei de executabile
* funcționalitatea ”Click here to begin a new processing”: aplicația funcționează corect pentru testele realizate
* funcționalitatea de download imagine procesata: aplicația funcționează corect pentru testele realizate

În urma testării aplicației, s-au remarcat următoarele:

* pentru a funcționa corect aplicația, este necesar ca fișierele specificate in config.properties să fie create înainte de rulare
* dacă după pornirea server-ului, au fost adaugate noi scheme sau fișiere executabile, acesta trebuie repornit (din butonul Reinitialize)
* dacă în folderele corespunzătoare schemelor Xml se află și fișiere cu alte extensii, aplicația nu mai funcționează
* ordinea de rulare a executabilelor este următoarea: otsu, full\_layout, single\_paging, hierarchy

**3. Concluzie**

Aplicația ocrWebManager funcționează corect, atât timp cât se respect anumite observații, si anume ca folderele corespunzătoare să existe dinainte și ca în folderele corespunzătoare schemelor Xml să nu existe fișiere cu alte extensii.