Scenariu de Test: Modulul “StringMatching”

1. Introducere...............................................................

1.1 Abstract........................................................

1.2 Cerințe.........................................................

1.3 Module afectate...........................................

1.4 Pași de testare.............................................

2. Rezultatele testării..................................................

2.1 Bugs............................................................

1. **Introducere**

1.1 **Abstract**

Modulul se ocupă cu potrivirea textelor obținute din parsarea fișierelor date ca input.

* 1. **Cerințe**

Conținutul obținut după primul pas, cel de eliminare de taguri inutile, este procesat mai departe de modulul **String Matching**.

1. Prima cerință este legată de fișierele de intrare, și anume posibilitatea introducerii lor ca input.
2. A doua cerință este alinierea, formatarea, procesarea etc. a textului din fișier.
3. A treia cerință este găsirea procentajelor, formarea unei statistici asupra corespondențelor textelor.
   1. **Module afectate**

Aplicația va pune la dispoziția echipei de ?fișierul XML procesat.

1.4 **Pași de Testare**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | **Descriere** | Rezultate Asteptate | | Pass | Fail | Comentarii |
| Primul pachet: | | | | | | |
| 1 | * Clasa HammingDistance: testăm dacă metoda getHammingDistance() returnează rezultate corecte, conform unor calcule efectuate separat.  1. Am introdus ca input stringurile ”abcdefgh” respectiv ”abcdefhg”; 2. Apoi ”123456789” și ”123456798”; același rezultat; 3. Am testat dacă punctuația afectează în vreun fel compararea; rezultatul fiind negativ; 4. Am testat dacă se efectuează compararea pentru 2 stringuri de lungimi diferite; negativ; | | Un număr întreg pozitiv reprezentând distanța Hamming; | ✔ |  |  |
| 2 | Introducem un fișier vid în format xhtml | Generarea unui fișier vid in format xml | |  |  |  |
| 3 | Introducem un fișier xhtml invalid având un tag închis care nu a fost deschis | Mesaj de eroare | |  |  |  |
| 4. | Introducem un fisier xhtml invalid având un tag deschis care nu a fost închis | Mesaj de eroare | |  |  |  |
| 5 | Introducem un fișier xhtml cu taguri de bold/italic/sub/sup | Generarea unui fișier în format xml cu tagurile specificate păstrate | |  |  |  |
| 6 | Introducem un fișier xhtml cu taguri de bold/italic/sub/sup și taguri nedorite | Generarea unui fișier în format xml cu tagurile specificate păstrate și nedorite eliminate | |  |  |  |
| 7 | Introducem un fișier xml valid | Mesaj de informare că fișierul este deja în format xml | |  |  |  |
| 8 | Introducem un fișier xhtml cu text de tipul ‘<text>’ | Mesaj de eroare | |  |  |  |
| 9 | Introducem un fișier in format xml invalid | Mesaj de eroare | |  |  |  |

1. **Rezultatele Testarii**

2.1 Bugs