DODO.NET

TIR

Test Incident Report

|  |  |
| --- | --- |
| Riferimento |  |
| Versione | 1.0 |
| Data | 5/01/2022 |
| Destinatario | Prof.ssa F. Ferrucci |
| Presentato da | Alfonso Cuomo  Simone Farina |
| Approvato da |  |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 07/12/2021 | 1.0 | Stesura documento | [Simone Farina] |

Sommario

1. Introduzione
2. Relazione con altri documenti

2.1 Test Case Specification

3. Test Incidents

3.1

11. Glossario

1. Introduzione

All’interno di questo documento verranno riportate tutte le problematiche relative alla fase di testing. Di seguito saranno descritti tutti i test case, presenti e descritti nel Test Case Specification (TCS) non sono andati a buon fine.

2.Relazione con altri documenti

I seguenti documenti saranno in relazione con il documento di Test Incident Report:

**2.1 Test Case Specification (TCS)**

Nel TCS sono descritti gli oracoli relativi ad ogni singolo test case in comunione con le funzionalità di sistema da testare.

3.Test incidents

|  |  |
| --- | --- |
| Test Incidence ID |  |
| Input |  |
| Tester |  |
| Risultati attesi |  |
| Risultati attuali |  |
| Animalie |  |
| Risoluzione |  |
| Tentativi da ripetere |  |

11.Glossario e definizioni

**Test Case Name:** test case univoco per distinguere i test case attuali dagli altri. ( la convenzione usata è TC\_[N].[M]\_[O], dove n è il numero dello use case, M quello per lo scenario e O indica il numero progressivo in funzione di N.

**Test Path:** la path del file di test che verrà effettuato.

**Input:** rappresenta la classe di equivalenza che è stata testata.

**Oracle:** sono i dati o comportamenti attesi dal sistema ( lo stato finale della situazione ).

**Log:** il comportamento osservato in seguito all’esecuzione del test.

**Pre-Condition:** è la precondizione dello use case.

**Flow of events:** è il flusso delle interazioni tra sistema e utente. Il flusso non è strettamente collegato allo use case. Bisogna specificare i dati effettivi che si usano e che vengono restituiti.

**Junit:** Framework di unit testing per il linguaggio di programmazione Java.

**Fault:** La causa meccanica o algoritmica di un errore.

**Failure:** Qualsiasi deviazione del comportamento osservato dal comportamento specificato.

**Tool:** Strumento software usato per ottenere un dato risultato.