Test Summary Report

Progetto:

CleanDesk

CleanDesk
Powered by AI

Riferimento:	
Versione:	0.9
Data:	09/03/2023
Destinatario:	Esame di Ingegneria del Software 2023/24
Presentato da:	Ambrosio Gennaro, Camoia Andrea
Approvato da:	

UNIVERSITÀ DEGLI STUDI
DI SALERNO

Revision Hystory

Data	Versione	Descrizione	Autori
15/02/2024	0.1	Stesura iniziale del documento	Ambrosio Gennaro Camoia Andrea
04/03/2024	0.5	Specifica e descrizione del testing.	Ambrosio Gennaro Camoia Andrea
09/03/2024	0.9	Revisione finale.	Ambrosio Gennaro Camoia Andrea

Team Member

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Andrea Camoia	Team Member	AC	a.camoia@studenti.unisa.it
Gennaro Ambrosio	Team Member	AG	g.ambrosio35@studenti.unisa.it

Indice

1 Introduzione

1.1 Scopo del Sistema

CleanDesk è un software punta a migliorare l'esperienza informatica e la produttività degli utenti. Utilizza l'intelligenza artificiale per **organizzare i file di uno spazio digitale** in modo intuitivo ed accessibile, fornendo un ambiente di utilizzo quanto più semplice ed efficace.

Inoltre, il sistema consente anche di **gestire in maniera consapevole la memoria** utilizzata da una cartella: analizzandone nel dettaglio l'utilizzo fisico oppure individuando file duplicati che sprecano spazio aumentando al tempo stesso la confusione

1.2 Scopo del Documento

Il seguente documento fornisce una specifica di tutte le **attività di testing svolte** per garantire il corretto funzionamento del sistema software CleanDesk. Saranno elencati i test incident risolti e non, valutando anche le tecniche di test adottate, descrivendone eventuali limitazioni.

1.3 Riferimenti

Di seguito vengono riportate le risorse utilizzate per la stesura della presente documentazione ed utili inoltre alla sua lettura:

- CleanDesk Statement of Work (SOW)
- CleanDesk Requirements Analysis Document (RAD)
- CleanDesk System Design Document (SDD)
- CleanDesk Test Plan (TP)
- CleanDesk Test Case Specification (TCS)
- CleanDesk Object Design Document (ODD)
- CleanDesk Test Incident Report (TIR)
- Dispense del prof. Carmine Gravino, fornite mediante la pagina del corso "Ingegneria del Software - Resto 2 - 2023/2024" sulla Piattaforma E-learning del Corso di Laurea in Informatica dell'Università degli Studi di Salerno;
- Libro di testo "Object Oriented Software Engineering Using UML Patterns and Java Prentice Hall 2010 Bernd Bruegge Allen H.Dutoit"
- Libro di testo "C. GHEZZI, D. MANDRIOLI, M. JAZAYERI, INGEGNERIA DEL SOFTWARE – FONDAMENTI E PRINCIPI, PRENTICE HALL, 2004"

2 Testing di Unità

2.1 Obiettivi

In questa fase, si testano i singoli sottosistemi, con l'obiettivo di verificare se questi sono stati correttamente codificati per svolgere la funzionalità per la quale sono stati individuati.

2.2 Svolgimento

Lo **Unit Testing** è stato eseguito dai membri interni del Team di CleanDesk utilizzando apposite **librerie di testing** automatico e sistematico. In particolare, come già specificato nel documento di *Test Plan*, le operazioni di testing si sono concentrate sulle funzionalità che coinvolgono l'interazione e l'input da parte dell'utente.

A questo scopo è stata utilizzata la libreria JavaScript `mocha`[1], ovvero un framework di testing che gira su `Node.js`. Grazie a quest'ultima è stato possibile strutturare il testing delle funzionalità in maniera rapida ed efficiente, e soprattutto separando logicamente i vari Test Cases.

I risultati ricevuti in base all'input, sono stati verificati e controllati rispetto a quelli attesi utilizzando la libreria `expect`.

```
> cleandesk@1.0.0 test
> mocha test/pathGlobal.test.js



  Insert folder Path to organize
    Cateogory 1) Check Lenght Category
      ✔ The lenght of the path should be between 4 and 256 [/Users/andrea/Desktop/NC23_CleanDesk/SDD]
      ✔ The lenght of the path should be between 4 and 256 [/Users/andrea/Desktop/NC23_CleanDesk/RAD]
    Category 2) Check Format Category
      ✔ The path should match the respective regular expression [/Users/andrea/Desktop/NC23_CleanDesk/SDD]
      ✔ The path should match the respective regular expression [/Users/andrea/Desktop/NC23_CleanDesk/RAD]
    Category 4) Check Admissibility Category
      ✔ The Path must refer to an eligible folder [/Users/andrea/Desktop/NC23_CleanDesk/SDD]
      ✔ The Path must refer to an eligible folder [/Users/andrea/Desktop/NC23_CleanDesk/RAD]


  6 passing (4ms)
```

Figure 2.1: Esempio esecuzione di un test tramite Mocha.

[1] `https://mochajs.org/`

3 Test di Integrazione

3.1 Obiettivi

Una volta testati i singoli sottosistemi, si può procedere a verificare se il **sistema funziona nella sua interezza.**

Questo viene fatto tramite il **test di integrazione (Integration Testing)**, in cui ci si focalizza su come funziona il sistema nel suo insieme e se le componenti precedentemente testate interagiscono correttamente tra loro.

3.2 Svolgimento

Il Test di integrazione è stato eseguito manualmente dai componenti del team, eseguendo ed effettuando prove pragmatiche delle funzionalità fornite da CleanDesk.

Particolare attenzione è stata posta nel verificare il corretto funzionamento del software durante l'interazione con il **modulo esterno di Machine Learning**, assicurando una corretta comunicazione e lavoro tra i sistemi.

4 Test di Sistema

4.1 Obiettivi

Una volta ottenuto il software CleanDesk nella sua interezza, è stato avviato il Testing di Sistema con l'obiettivo di verificare se il sistema completo è conforme ai requisiti funzionali e non funzionali.

4.2 Svolgimento

I Test di Sistema sono stati condotti manualmente **eseguendo l'applicazione e testando ogni funzionalità** con input differenti, simulando diversi scenari per ogni tipo di utente e su entrambe le tipologie di sistemi operativi disponibili.

- **I Requisiti Funzionali sono stati verificati** avviando manualmente le funzionalità del software, e verificando che il software restituisca correttamente i risultati attesi, secondo le modalità e preferenze indicate.
- **I Requisiti NON Funzionali**, allo stesso modo, sono stati verificati praticamente tramite esecuzione ed utilizzo del Software da parte di utenti Interni ed Esterni al team. In particolare:
  - Tutte le categorie di requisiti non funzionali sono stati testati dai membri del team, su entrambi i sistemi operativi, verificando che questi fossero effettivamente rispettati.
  - I Requisiti di Usability, Reliability e Performance sono stati testati e convalidati anche da **utilizzatori esterni al team**, su macchine proprietarie. Particolare attenzione è stata rivolta all'efficacia e comprensione dell'interfaccia grafica e al corretto funzionamento del software su sistemi operativi diversi.

Di seguito sono riportati i risultati dei test:

Data esecuzione	Numero di Successi	Numero di Fallimenti
10/02/2024	10	3
11/02/2024	25	1
13/02/2024	14	0