

# Progetto: **CleanDesk**

UNIVERSITÀ DEGLI STUDI  
DI SALERNO



## Revision History

Data	Versione	Descrizione	Autori
11/12/2023	0.1	Stesura iniziale del documento	Ambrosio Gennaro Camoia Andrea
12/12/2023	0.2	Descrizione dell'Introduzione, Documenti Correlati e Panoramica del Sistema	Ambrosio Gennaro Camoia Andrea
12/12/2023	0.3	Descrizione delle Funzionalità da Testare, descrizione dell'Approccio e criteri Pass/Fail	Ambrosio Gennaro Camoia Andrea
13/12/2023	0.4	Descrizione dei Criteri di Sospensione e Ripristino e Descrizione dei Test Case	Ambrosio Gennaro Camoia Andrea
15/12/2023	0.5	Decrizione degli Strumenti per il Testing	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

<b>1</b>	<b>Introduzione</b>	<b>4</b>
<b>2</b>	<b>Documenti Correlati</b>	<b>4</b>
<b>3</b>	<b>Panoramica del Sistema</b>	<b>4</b>
<b>4</b>	<b>Funzionalità da Testare</b>	<b>5</b>
4.1	Funzionalità da NON Testare . . . . .	5
<b>5</b>	<b>Approccio</b>	<b>5</b>
5.1	Testing di Unità (Unit Testing) . . . . .	5
5.2	Test di Integrazione (Integration Testing) . . . . .	5
5.3	Testing di Sistema (System Testing) . . . . .	6
<b>6</b>	<b>Criteri Pass/Fail</b>	<b>6</b>
<b>7</b>	<b>Criteri di Sospensione e Ripristino</b>	<b>6</b>
7.1	Criteri di Sospensione . . . . .	6
7.2	Criteri di Ripresa . . . . .	6
<b>8</b>	<b>Strumenti per il Testing</b>	<b>6</b>
<b>9</b>	<b>Test Cases</b>	<b>7</b>
9.1	Organizzazione File . . . . .	7
9.1.1	TC_OF.1: Inserimento Path da Organizzare. . . . .	7
9.2	Ricerca File . . . . .	9
9.2.1	TC_OF.2: Visualizzazione dettagli Report Organizzazione . . . . .	9

# 1 Introduzione

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.

Con l'obiettivo di garantire il corretto funzionamento di queste funzionalità, tramite questo documento si va a definire quelle che sono le **attività di Testing**, volte appunto ad individuare problemi di correttezza, completezza e affidabilità delle componenti software in corso di sviluppo, e soprattutto in modo da consegnare al committente un sistema privo di malfunzionamenti.

In particolare, questo documento di Test Plan si concentra sul definire l'obiettivo, l'approccio, le risorse e lo scheduling per le attività di testing.

# 2 Documenti Correlati

In questa sezione si specificano i documenti in forte relazione ed utilizzati come riferimento per le attività di testing, in particolare per la definizione dei Test Case.

- Relazione con il **Requirements Analysis Document (RAD)**  
Si individuano i Test Case basandosi sui requisiti funzionali descritti mediante Use Case e Scenari nel RAD.
- Relazione con il **System Design Document (SDD)**  
I Test Case individuati rispettando la suddivisione in sottosistemi dichiarata nell'SDD.

# 3 Panoramica del Sistema

Il sistema di CleanDesk è basato su un architettura software a "Three-Tier", con una stratificazione dei sottosistemi in appunto 3 livelli indipendenti: Interface Layer, Application Logic Layer e Storage Layer. I sottosistemi di cui è composto sono:

- Organizzazione File;
- Analisi Memoria;
- Ricerca Duplicati;

- Gestione File System;
- Storage;
- Storage Connection;

## 4 Funzionalità da Testare

La fase di testing del sistema si concentra sulle seguenti funzionalità:

1. Organizzazione File: **Avvia Nuova Organizzazione.**
2. Organizzazione File: **Visualizza Report Organizzazione**

### 4.1 Funzionalità da NON Testare

Non saranno incluse nell'attività di testing le feature con interazione con l'utente nulla oppure identica ad altre funzionalità già sottoposte a testing.

## 5 Approccio

La fase di testing, in accordo con i vincoli progettuali, sarà composta da tre attività fondamentali:

### 5.1 Testing di Unità (Unit Testing)

In cui si testano i singoli sottosistemi, con l'obiettivo di verificare se questi svolgono correttamente le attività per le quali sono stati implementati.

In questa fase si prosegue con una tecnica di **testing "Black Box"**

### 5.2 Test di Integrazione (Integration Testing)

In cui si verifica se i diversi sottosistemi funzionano correttamente quando interagiscono tra loro.

In particolare sarà seguita una **strategia bottom-up**, in cui si sottopongono al testing prima i sottosistemi di livello inferiore, che non hanno nessuna dipendenza da altre componenti.

### 5.3 Testing di Sistema (System Testing)

Dove si verifica il sistema nella sua interezza, assicurandosi che le richieste del cliente siano state soddisfatte correttamente.

## 6 Criteri Pass/Fail

Definiamo un Test come **Positivo**, che ha avuto cioè successo, quando il sistema produce un output/comportamento diverso da quello definito nell'oracolo, dato uno specifico insieme di input.

Al contrario, un Test avrà esito **Negativo**, cioè fallimentare, quando il comportamento del sistema sarà conforme a quello previsto.

## 7 Criteri di Sospensione e Ripristino

In questa sezione vengono illustrati i criteri che saranno seguiti durante lo svolgimento delle delicate attività di testing.

### 7.1 Criteri di Sospensione

Il testing non sarà interrotto fino al suo completamento, anche in caso di rilevamento di un fallimento. Tuttavia si prevedono due casi straordinari in cui il testing potrà essere momentaneamente sospeso:

- Quando si verifica un errore nel Test stesso che rischia di compromettere le attività. In questo caso è necessario correggere l'errore prima di poter riprendere il testing.
- Quando, a discrezione dell'addetto al Testing, si individuano troppi errori nell'implementazione delle funzionalità.

### 7.2 Criteri di Ripresa

In caso di modifiche apportate al sistema, la fase di testing sarà nuovamente eseguita per assicurare che il sistema mantenga la sua funzionalità anche in seguito a cambiamenti.

## 8 Strumenti per il Testing

Per effettuare il testing del sistema è necessario soltanto un computer con all'interno il software CleanDesk correttamente installato.

## 9 Test Cases

Per lo sviluppo dei Test Cases sarà utilizzata la tecnica del Category Partition, una strategia che permette di effettuare il Testing in maniera sistematica.

Per ogni funzionalità testata saranno individuati dei **Parametri**, e per ogni parametro verranno individuate delle **Categorie**, le quali poi saranno suddivise a loro volta in **Scelte**. Alle scelte verrà assegnato un valore sottoposto poi al sistema.

I test cases verranno definiti nel documento di **Test Cases Specification (TCS)**.

### 9.1 Organizzazione File

#### 9.1.1 TC\_OF.1: Inserimento Path da Organizzare.

In questo Test Case, sono specificati due formati di versi per il parametro "Path". A seconda del sistema operativo su cui viene eseguito e testato il software, sarà automaticamente utilizzato il formato rispettivo.

Parametro:	Path
Formato 1	(Windows)
$\wedge(?:[a-zA-Z\backslash s]:\\ \\\\\\)(?:[^\backslash s:]+\backslash)(?:[^\w/:<>"'?!]+)?\$$	
Formato 2	(MacOS)
$\wedge(?:\w/(?:[^\w/:]+\w/)*)(?:[^\w/\backslash s]+)?\$$	
Nome Categoria	Scelte per la Categoria
Lunghezza [LN]	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\geq 4</math> AND Lunghezza <math>\leq 256</math> [PROPERTY_LN_OK]</li> <li>2. Lunghezza <math>&lt; 4</math> OR Lunghezza <math>&gt; 256</math> [ERROR]</li> </ol>
Formato [PF]	<ol style="list-style-type: none"> <li>1. Il Path rispetta il relativo formato. [PROPERTY_LN_OK]</li> <li>2. Il Path NON rispetta il formato. [ERROR]</li> </ol>





Correttezza [CT]	<ol style="list-style-type: none"><li>1. Il Path fa riferimento ad una cartella. [PROPERTY_LN_OK]</li><li>2. Il path fa riferimento ad un file. [ERROR]</li></ol>
Ammissibilità [AM]	<ol style="list-style-type: none"><li>1. Il Path fa riferimento ad una cartella ammissibile. [PROPERTY_LN_OK]</li><li>2. Il Path fa riferimento ad una cartella NON ammissibile. [ERROR]</li></ol>

Test Case ID	Test Frame	Esito
TC_OF.1_1	LN2	<b>Errore:</b> lunghezza invalida
TC_OF.1_2	LN1, PF2	<b>Errore:</b> il path non rispetta il formato
TC_OF.1_3	LN1, PF1, CT2	<b>Errore:</b> il path non fa riferimento ad una cartella
TC_OF.1_4	LN1, PF1, CT1, AM2	<b>Errore:</b> il path riferisce ad una cartella non ammissibile.
TC_OF.1_5	LN1, PF1, CT1, AM1	<b>Successo</b>

## 9.2 Ricerca File

### 9.2.1 TC\_OF.2: Visualizzazione dettagli Report Organizzazione

Parametro:	Path
Nome Categoria	Scelte per la Categoria
MatchReport [MR]	1. Report Match nel DB = True [PROPERTY_MR_OK] 2. Report Match nel DB = False [ERROR]
Parametro:	Nome File
Nome Categoria	Scelte per la Categoria
MatchFile [MF]	1. File Match nel DB = True [PROPERTY_MF_OK] 2. File Match nel DB = False [ERROR]

Test Case ID	Test Frame	Esito
TC_OF.2_1	MR2	<b>Errore:</b> Report non trovato nel DB
TC_OF.2_2	MR1, MF2	<b>Errore:</b> File non trovato nel DB
TC_OF.2_3	MR1, MF1	<b>Successo</b>