

Test Summary Report

C04 – Esistere



Riferimento	C04_Esistere_TSR_V1.0
Versione	1.0
Data	19/01/2023
Destinatario	Prof.ssa Filomena Ferrucci, Prof.re Fabio Palomba
Presentato da	C04 Team Esistere: <ul style="list-style-type: none">• Antonio D'Auria (AA)• Luca Casillo (LC)• Maria Giovanna Della Pietra (MGP)• Ogham If Dell'Erba (OE)• Raffaele Forte (RF)• Rosa Carotenuto (RC)• Valentino Dragone (VD)
Approvato da	Alessandra Parziale, Saverio Napolitano



Esistere

Per poter abbracciare ogni istante.

Revision History

Data	Versione	Descrizione	Autori
19/01/2024	1.0.0	Creazione Test Summary	AA, LC, MGP, OE, RC, VD



Esistere

Per poter abbracciare ogni istante.

Team Members

Nome	Ruolo nel progetto	Acronimo	Informazione di controllo
Alessandra Parziale	Project Manager	AP	a.parziale8@studenti.unisa.it
Saverio Napolitano	Project Manager	SN	s.napolitano44@studenti.unisa.it
Antonio D'Auria	Team Member	AA	a.dauria123@studenti.unisa.it
Luca Casillo	Team Member	LC	l.casillo16@studenti.unisa.it
Maria Giovanna Della Pietra	Team Member	MGP	m.dellapietra10@studenti.unisa.it
Rosa Carotenuto	Team Member	RC	r.carotenuto16@studenti.unisa.it
Ogham If Dell'Erba	Team Member	OE	o.dellerba@studenti.unisa.it
Valentino Dragone	Team Member	VD	v.dragone5@studenti.unisa.it
Raffaele Forte	Team Member	RF	r.forte12@studenti.unisa.it



Esistere

Per poter abbracciare ogni istante.

Sommario

REVISION HISTORY	1
TEAM MEMBERS	2
SOMMARIO	3
1.INTRODUZIONE	4
2.RELAZIONE CON ALTRI DOCUMENTI	4
<i>Relazione con il Test Plan:</i>	<i>4</i>
<i>Relazione con il Test Case Specification:</i>	<i>4</i>
<i>Relazione con il Test Incident Report</i>	<i>4</i>
3. TESTING UNITARIO	4
4. TESTING DI SISTEMA	5



Esistere

Per poter abbracciare ogni istante.

1.Introduzione

Esistere costituisce una piattaforma che mira a semplificare l'interazione tra pazienti affetti da Alzheimer e professionisti medici, facilitando il monitoraggio della malattia attraverso un ambiente centralizzato in cui sia il medico che il caregiver possono operare con facilità.

Il Test Summary Report si occupa dell'analisi delle attività di testing condotte nel tempo per assicurare il corretto funzionamento della piattaforma. All'interno del documento saranno delineate le strategie di testing adottate, le funzionalità sottoposte a verifica e gli strumenti impiegati per garantire un funzionamento efficiente della piattaforma, riducendo al minimo possibile gli errori.

2.Relazione con altri documenti

Di seguito vengono elencate le relazioni tra il presente documento e gli altri documenti del testing:

Relazione con il Test Plan:

Il Test Summary Report fa riferimento alle attività di testing specificate nel Test Plan

Relazione con il Test Case Specification:

Il Test Summary Report contiene una cernita dei test specificati nel Test Case Specification

Relazione con il Test Incident Report

Il Test Summary Report contiene una cernita dei risultati dell'esecuzione specificati nel Test Incident Report

3. Testing unitario

L'obiettivo dei test di unità è isolare e verificare le singole unità di codice, focalizzandosi sulle interazioni tra i service e i dao (Data Access Object). Si mira a garantire la precisione e la coerenza delle chiamate dei metodi dei dao da parte dei service, assicurando la correttezza nelle operazioni.

Per raggiungere questo obiettivo, i test adottano la pratica del mocking dei dao.

Questa tecnica consente di simulare il comportamento dei dao, creando versioni semplificate e controllabili di tali componenti. Tale approccio assicura che i service siano testati in modo isolato, senza dipendenze dirette dagli strati inferiori dell'architettura, come i database o altri servizi esterni.

Utilizzando framework, nel caso di Esistere Jest, simuliamo il comportamento dei dao e controlliamo le interazioni con i service.

I risultati attesi dai test di unità comprendono la verifica che i service gestiscano correttamente le interazioni con i dao di cui è stato fatto il mocking, tramite l'utilizzo del framework, garantendo l'affidabilità delle operazioni logiche di business.



Esistere

Per poter abbracciare ogni istante.

4. Testing di sistema

Per il testing di sistema, è stato utilizzato Nightwatch.js, un framework open source integrato. Questo strumento facilita l'automazione del testing e supporta un approccio end-to-end, simulando le interazioni che un utente reale avrebbe con il sistema. L'obiettivo è assicurare che tutte le componenti collaborino come previsto.

Di seguito i risultati dei test:

Esecuzione	#Fallimenti	#Successi
Esecuzione 1 18/01/2024	4	39
Esecuzione 2 18/01/2024	0	43