

# Test Plan

## C04 – Esistere



<b>Riferimento</b>	C04_Esistere_TP_V1.0
<b>Versione</b>	1.0
<b>Data</b>	11/12/2023
<b>Destinatario</b>	Prof.ssa Filomena Ferrucci, Prof.re Fabio Palomba
<b>Presentato da</b>	C04 Team Esistere: <ul style="list-style-type: none"><li>• Antonio D'Auria (AA)</li><li>• Luca Casillo (LC)</li><li>• Maria Giovanna Della Pietra (MGP)</li><li>• Ogham If Dell'Erba (OE)</li><li>• Raffaele Forte (RF)</li><li>• Rosa Carotenuto (RC)</li><li>• Valentino Dragone (VD)</li></ul>
<b>Approvato da</b>	Alessandra Parziale, Saverio Napolitano



## Revision History

Data	Versione	Descrizione	Autori
05/12/2023	0.1.0	Aggiunta Introduzione	OE
05/12/2023	0.2.0	Aggiunta Relazione con altri documenti	MGP
08/12/2023	0.3.0	Aggiunta Approccio	AA, LC, MGP, OE, RF, RC, VD
09/12/2023	0.4.0	Aggiunta Panoramica del Sistema	RF
09/12/2023	0.5.0	Aggiunte Features da Testare	VD
09/12/2023	0.6.0	Aggiunti Pass/Fail Criteria	AA
09/12/2023	0.7.0	Aggiunto Sospensione e Rispristino	LC
09/12/2023	0.8.0	Aggiunti Test Deliverables	LC
09/12/2023	0.9.0	Aggiunto Materiale di Testing	RC
09/12/2023	0.10.0	Aggiunte Responsabilità	RC
09/12/2023	0.11.0	Aggiunti Test Cases	AA, LC, MGP, OE, RF, RC, VD
10/12/2023	0.11.1	Revisione Architetture	LC, RC
10/12/2023	0.11.2	Revisione Strutturale	AA, RF
11/12/2023	1.0.0	Test Plan	AA, LC, MGP, OE, RF, 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

<b>REVISION HISTORY</b>	<b>1</b>
<b>TEAM MEMBERS</b>	<b>2</b>
<b>SOMMARIO</b>	<b>3</b>
<b>1. INTRODUZIONE</b>	<b>4</b>
<b>2. RELAZIONE CON ALTRI DOCUMENTI</b>	<b>4</b>
<b>3. PANORAMICA DEL SISTEMA</b>	<b>4</b>
<b>4. FEATURES DA TESTARE/DA NON TESTARE</b>	<b>5</b>
<b>5. PASS/FAIL CRITERIA</b>	<b>5</b>
<b>6. APPROCCIO</b>	<b>6</b>
<b>TESTING DI SISTEMA</b>	<b>6</b>
<i>Functional Testing</i>	<i>6</i>
<i>Performance Testing</i>	<i>6</i>
<i>Pilot Testing</i>	<i>6</i>
<i>Acceptance Testing</i>	<i>6</i>
<i>Installation Testing</i>	<i>6</i>
<b>7. SOSPENSIONE E RIPRISTINO</b>	<b>7</b>
<b>8. TEST DELIVERABLES</b>	<b>8</b>
<b>9. MATERIALE DI TESTING</b>	<b>8</b>
<b>10. RESPONSABILITÀ</b>	<b>8</b>
<b>11. TEST CASES</b>	<b>8</b>
<b>11.1 GESTIONE UTENTE</b>	<b>8</b>
<i>11.1.1 Registrazione</i>	<i>9</i>
<i>11.1.2 Login</i>	<i>14</i>
<i>11.1.3 Inserimento Dati Paziente</i>	<i>14</i>
<b>11.2 GESTIONE FILASTROCCA</b>	<b>15</b>
<i>11.2.1 Creazione Filastrocca</i>	<i>15</i>
<b>11.3 GESTIONE STORIE</b>	<b>16</b>
<i>11.3.1 Creazione Storia (Foto)</i>	<i>16</i>
<b>11.4 GESTIONE QUIZ DI ALLENAMENTO</b>	<b>17</b>
<i>11.4.1 Creazione Quiz Di Allenamento Giornaliero</i>	<i>17</i>
<b>11.5 GESTIONE TODOLIST</b>	<b>18</b>
<i>11.5.1 Creazione ToDoList</i>	<i>18</i>



# Esistere

Per poter abbracciare ogni istante.

## 1. Introduzione

---

Esistere è una piattaforma che ha come obiettivo quello di semplificare il rapporto tra pazienti affetti da Alzheimer e medici, fornendo gli strumenti necessari a favorire il monitoraggio della malattia tramite un ambiente centralizzato in cui sia il medico che il caregiver del paziente possano agevolmente operare. Il documento di Test Plan si occupa di analizzare le attività di Testing effettuate nel tempo per garantire il corretto funzionamento della piattaforma.

All'interno del documento saranno presenti le strategie di testing che verranno adottate, le funzionalità testate e gli strumenti che saranno utilizzati per garantire una piattaforma con il minor numero possibile di errori e che funzioni in maniera efficace.

## 2. Relazione con altri documenti

---

Questo documento fa riferimento a tutti i documenti prodotti fino al rilascio del sistema, di conseguenza verrà modificato in futuro dopo il rilascio di altri documenti non ancora prodotti. I test case sono basati sulle funzionalità del sistema, individuate e raccolte nei seguenti documenti realizzati:

### **Relazioni con il Requirements Analysis document (RAD)**

I test case pianificati nel Test Plan sono elaborati in relazione ai requisiti funzionali e non funzionali presenti nel RAD.

### **Relazioni con il System Design Document (SDD)**

I test case pianificati nel Test Plan devono rispettare la suddivisione in sottosistemi presentata nell'SDD.

## 3. Panoramica del sistema

---

Il sistema proposto si basa su un'architettura di tipo three-tier, il quale consente una separazione chiara e organizzata tra le varie componenti.

Durante lo sviluppo del sistema sarà utilizzato React in quanto garantisce una serie di vantaggi, tra cui una gestione efficiente dell'interfaccia utente, facilità di utilizzo e la creazione organizzata di componenti riutilizzabili, garantendo codice mantenibile e scalabile.

Per la parte di front-end e la generazione delle view verranno usati:

- HTML5
- CSS3

Per quanto riguarda invece la logica applicativa di back-end saranno utilizzati:

- NodeJS
- TypeScript

Per la gestione della persistenza dei dati verrà utilizzato il DBMS PostgreSQL, una soluzione robusta e scalabile che gestisce in modo affidabile i dati dell'applicazione.



## 4. Features da testare/da non testare

---

Di seguito la lista delle feature di cui si effettuerà il testing per le varie gestioni:

- Gestione Utente
  - Registrazione
  - Login
  - Inserimento Dati Paziente
- Gestione Filastrocca
  - Creazione Filastrocca
- Gestione Storie
  - Creazione Storia (Foto)
- Gestione Quiz Allenamento Giornaliero
  - Creazione Quiz Allenamento Giornaliero
- Gestione ToDoList
  - Creazione ToDoList

Le attività di testing non coinvolgeranno le funzionalità correlate a requisiti funzionali di bassa o media priorità. Saranno inoltre escluse le funzionalità che non richiedono l'input manuale dell'utente, come ad esempio le attività limitate alla sola visualizzazione di dati.

Il requisito funzionale “Consultazione Modulo Intelligenza Artificiale” (che rappresenta il modulo di intelligenza artificiale) che permettere al MED di visualizzare l'analisi dell'immagine della tac, facente parte della Gestione Tac, verrà testato in futuro con differenti tecniche e sarà quindi trattato in una diversa documentazione.

## 5. Pass/Fail criteria

---

L'obiettivo del testing è quello di identificare la presenza di errori (faults) all'interno del sistema, procedendo con la risoluzione degli stessi.

Per valutare l'esito di un test case viene utilizzato un oracolo: per oracolo si intende il risultato atteso dall'esecuzione, basandosi sui requisiti definiti.

Un test ha successo (pass) se: dato un input al sistema l'output restituito non corrisponde all'output atteso dall'oracolo.

Un test ha fallito (fail) se: dato un input al sistema, l'output restituito corrisponde all'output atteso dall'oracolo.

L'intera fase di testing sarà considerata valida se e solo se i seguenti vincoli saranno rispettati:

- Tutti i requisiti funzionali ad alta priorità devono essere testati.
- Deve essere effettuato un test di regressione ogni qualvolta vengono aggiunte nuove caratteristiche al sistema o vengono modificate caratteristiche già esistenti.
- Deve essere raggiunta una branch coverage non inferiore al 75%.



# Esistere

Per poter abbracciare ogni istante.

## 6. Approccio

---

Il testing dell'intero sistema si comporrà di tre fasi:

- Testing di sistema
- Testing di integrazione
- Testing di unità

Le tre fasi verranno progettate nell'ordine appena descritto, ma verranno eseguite in ordine inverso.

La progettazione dei casi di sistema, svolta prima dell'implementazione del sistema, avverrà nella sezione nove del medesimo documento, verrà perfezionata nel documento di Test Plan Specification e in seguito nella fase di esecuzione; durante la fase di implementazione verranno progettati, poi, i test di unità.

### Testing di Sistema

Il testing di unità e il testing di integrazione mirano alla ricerca di faults all'interno del sistema. Il testing di sistema verrà effettuato per assicurarsi che l'intero sistema garantisca i requisiti funzionali e non funzionali definiti nel documento Requirements Analysis Document.

Per il testing di sistema verrà utilizzato il tool Nightwatch.js, un framework integrato open source: permette l'automazione del testing e consente un testing end-to-end simulando le interazioni che un utente reale avrebbe con il sistema per garantire che tutte le componenti lavorino insieme come previsto.

Esso si compone di varie attività:

#### Functional Testing

Attuo a trovare le differenze tra i requisiti funzionali e il sistema; si focalizzerà sul testing delle attività che sono più rilevanti per l'utente e hanno una più alta probabilità di rilevare criticità.

#### Performance Testing

Attuo a trovare le differenze tra i Design Goals, definiti nel System Design Document, e il sistema; non se ne assicura l'esecuzione a causa del basso budget.

#### Pilot Testing

Il sistema verrà installato e utilizzato da un numero limitato di utenti; non se ne assicura l'esecuzione a causa del basso budget.

#### Acceptance Testing

Verrà effettuato un benchmark test solo sul functional testing, dove i Project Manager simuleranno la figura del cliente.

#### Installation Testing

Il sistema verrà installato nell'ambiente di destinazione; non se ne assicura l'esecuzione a causa del basso budget.



# Esistere

Per poter abbracciare ogni istante.

## Testing di Unità

Nel paragrafo dedicato ai Test di Unità del Test Plan, l'accento è posto sull'importanza di verificare l'interazione tra i componenti service e dao (Data Access Object). Questi test sono fondamentali per garantire che i service interagiscano correttamente con i dao, una componente chiave per la gestione dei dati all'interno dell'architettura software.

I test di unità hanno lo scopo di isolare e verificare le singole unità di codice, ovvero le interazioni tra i service e i dao. Il focus è garantire che i service chiamino i metodi dei dao in modo preciso e coerente, assicurando l'accuratezza nelle operazioni di manipolazione e accesso ai dati.

Per realizzare questo, i test utilizzano il mocking dei dao. Il mocking è una tecnica che permette di simulare il comportamento dei dao, creando versioni semplificate e controllabili di questi componenti. Questo approccio garantisce che i service siano testati in isolamento, senza dipendenze dirette dagli strati inferiori dell'architettura, come i database o altri servizi esterni. Utilizzando framework come Jest, simuliamo il comportamento dei dao e controlliamo le interazioni con i service.

Struttura dei test:

- Importazioni: vengono importati i moduli necessari, incluso il service da testare e il dao da simulare
- Inizializzazione: Utilizzando `beforeEach`, configuriamo l'ambiente di test, istanziando il service e creando i mock dei dao
- Esecuzione dei Test: Ogni caso di test (test o it) verifica specifiche interazioni tra il service e il dao mockato. Utilizziamo asserzioni per confermare che i metodi dei dao siano chiamati correttamente
- Pulizia Post-Test: Con `afterEach`, ripuliamo l'ambiente di test per prevenire interferenze tra test successivi.

I risultati attesi dai test di unità includono la conferma che ogni service gestisca le interazioni con i suoi dao mockati correttamente, assicurando così l'affidabilità delle operazioni di business logiche.

## 7. Sospensione e ripristino

---

Nella seguente sezione verranno specificati i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprenderà il test.

### Criteri di sospensione

Il testing non verrà sospeso fino alla sua terminazione, anche in caso di rilevazione di una failure. L'unico caso in cui il testing potrà essere momentaneamente sospeso sarà nel caso in cui venga restituito un errore relativo alla definizione di uno dei test stessi al momento dell'esecuzione.

### Criteri di ripristino

Il test verrà ripreso solo dopo aver risolto i fault individuati.





## 8. Test deliverables

---

I documenti che saranno prodotti durante questa fase sono i seguenti:

- Test Plan;
- Test Case Specification.

Ulteriori documenti saranno prodotti durante le prossime fasi.

## 9. Materiale di Testing

---

Le risorse che verranno utilizzate per l'esecuzione del testing si compongono di materiale redatto in precedenza, risorse hardware e risorse software.

Per quanto concerne il materiale redatto, le risorse utilizzate saranno il Requirements Analysis Document, il System Design Document e l'Object Design Document che identificheranno le componenti da testare.

Relativamente alla componente hardware, è necessario il semplice utilizzo di un computer.

Invece, a proposito dei tools di software, verranno utilizzati, come già detto in precedenza nella [Sez. 6](#), il tool Nightwatch.js per il testing di sistema; verranno inoltre integrati ad esso i due framework di Jest e, in combinazione con quest'ultimo, verrà utilizzata la libreria React Testing Library per il testing di integrazione e il testing di unità.

Per l'effettiva esecuzione, sarà richiesta l'installazione di Node.js come ambiente runtime e npm come gestore di pacchetti per l'installazione delle dipendenze del sistema.

Come accennato, se necessario, verrà utilizzato Jest anche in caso di mocking delle componenti per il test di integrazione.

Infine, altre risorse che verranno adoperate, saranno le Github Actions, fornite da Github, per l'integrazione continua (CI/CD).

## 10. Responsabilità

---

I team members verranno divisi in due sottogruppi che si occuperanno rispettivamente dello sviluppo del lato backend e del lato frontend.

Dei sette team members che compongono il team, i project manager assegneranno quattro team members al lato backend e tre team members al frontend.

Ogni team member conoscerà tutto il lato di testing del sistema realizzato.

## 11. Test cases

---

### 11.1 Gestione utente



# Esistere

Per poter abbracciare ogni istante.

## 11.1.1 Registrazione

### 11.1.1.1 Registrazione Medico

Parametro: Email	
Formato	
$^{\wedge}[A-z0-9.\_ \% + -] + @ [A-z0-9.\_ + \.][A-z] \{2,50\} \$$	
Nome Categoria	Scelta per la categoria
Formato[FE]	<ol style="list-style-type: none"><li>1. Rispetta il formato = false [error]</li><li>2. Rispetta il formato = true [PROPERTY FE_OK]</li></ol>
Parametro: Password	
Formato	
$^{\wedge} ( ? = . * [a-z] ) ( ? = . * [A-Z] ) ( ? = . * \backslash d ) ( ? = . * [ @ \$ ! \% * ? \& ] ) [A-Za-z \backslash d @ \$ ! \% * ? \& ] \{8,50\} \$$	
Nome Categoria	Scelta per la categoria
Formato [FP]	<ol style="list-style-type: none"><li>1. Rispetta il formato = false [error]</li><li>2. Rispetta il formato = true [PROPERTY FP_OK]</li></ol>
Parametro: Conferma Password	
Nome Categoria	Scelta per la categoria
Match[MP]	<ol style="list-style-type: none"><li>1. Rispetta il formato = false [error]</li><li>2. Rispetta il formato = true [PROPERTY MP_OK]</li></ol>



# Esistere

Per poter abbracciare ogni istante.

Parametro: Nome	
Nome Categoria	Scelta per la categoria
Lunghezza[LN]	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 0</math> OR Lunghezza <math>&gt; 30</math> = false [error]</li> <li>2. <math>0 &lt; \text{Lunghezza} \leq 30</math> = true [PROPERTY LN_OK]</li> </ol>
Parametro: Cognome	
Nome Categoria	Scelta per la categoria
Lunghezza[LC]	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 0</math> OR Lunghezza <math>&gt; 30</math> = false [error]</li> <li>2. <math>0 &lt; \text{Lunghezza} \leq 30</math> = true [PROPERTY LC_OK]</li> </ol>
Parametro: Indirizzo Studio	
Formato	
$^{\wedge}[A-z\grave{A}-\grave{u}\grave{'}]{2,30}\$$	
Nome Categoria	Scelta per la categoria
Formato [FIS]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FIS_OK]</li> </ol>
Parametro: Numero Civico Studio	
Nome Categoria	Scelta per la categoria
Lunghezza[LNCS]	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 0</math> OR Lunghezza <math>&gt; 6</math> = false [error]</li> <li>2. <math>0 &lt; \text{Lunghezza} \leq 6</math> = true [PROPERTY LNCS_OK]</li> </ol>
Parametro: Città Studio	
Nome Categoria	Scelta per la categoria
Lunghezza[LCS]	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 0</math> OR Lunghezza <math>&gt; 30</math> = false [error]</li> <li>2. <math>0 &lt; \text{Lunghezza} \leq 30</math> = true [PROPERTY LCS_OK]</li> </ol>
Parametro: Numero telefono studio	
Formato	
$^{\wedge}\backslash d\{10\}\$$	
Nome Categoria	Scelta per la categoria
Formato [FNFS]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FNFS_OK]</li> </ol>



# Esistere

Per poter abbracciare ogni istante.

Test Case ID	Test frame	Esito
TC_1.1.1_1	FE1	Errato: E-mail non corretta
TC_1.1.1_2	FE2, FP1	Errato: Password non corretta
TC_1.1.1_3	FE2, FP2, MP1	Errato: Conferma password errata
TC_1.1.1_4	FE2, FP2, MP2, LN1	Errore: Nome non rientra nei limiti
TC_1.1.1_5	FE2, FP2, MP2, LN2, LC1	Errore: Cognome non rientra nei limiti
TC_1.1.1_6	FE2, FP2, MP2, LN2, LC2, FIS1	Errato: Indirizzo Studio non corretto
TC_1.1.1_7	FE2, FP2, MP2, LN2, LC2, FIS2, LNCS1	Errato: Numero Civico non rientra nei limiti
TC_1.1.1_8	FE2, FP2, MP2, LN2, LC2, FIS2, LNCS2, LCS1	Errato: Città Studio non rientra nei limiti
TC_1.1.1_9	FE2, FP2, MP2, LN2, LC2, FIS2, LNCS2, LCS2, FNTS1	Errato: Numero telefono studio non corretto
TC_1.1.1_10	FE2, FP2, MP2, LN2, LC2, FIS2, LNCS2, LCS2, FNTS2	Corretto



# Esistere

Per poter abbracciare ogni istante.

## 11.1.1.2 Registrazione Caregiver/Familiare

Parametro: Email	
Formato	
$^{\wedge}[A-z0-9._\%+-]+\@[A-z0-9.-]+\.[A-z]{2,50}\$$	
Nome Categoria	Scelta per la categoria
Formato[FE]	<ol style="list-style-type: none"><li>3. Rispetta il formato = false [error]</li><li>4. Rispetta il formato = true [PROPERTY FE_OK]</li></ol>
Parametro: Password	
Formato	
$^{\wedge}(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@\$!\%*\?&])[A-Za-z\d@!\%*\?&]{8,50}\$$	
Nome Categoria	Scelta per la categoria
Formato [FP]	<ol style="list-style-type: none"><li>3. Rispetta il formato = false [error]</li><li>4. Rispetta il formato = true [PROPERTY FP_OK]</li></ol>
Parametro: Conferma Password	
Nome Categoria	Scelta per la categoria
Match[MP]	<ol style="list-style-type: none"><li>3. Rispetta il formato = false [error]</li><li>4. Rispetta il formato = true [PROPERTY MP_OK]</li></ol>
Parametro: Nome	
Nome Categoria	Scelta per la categoria
Lunghezza[LN]	<ol style="list-style-type: none"><li>3. Lunghezza &lt;= 0 OR Lunghezza &gt; 30 = false [error]</li><li>4. 0 &lt; Lunghezza &lt;= 30 = true [PROPERTY LN_OK]</li></ol>
Parametro: Cognome	
Nome Categoria	Scelta per la categoria
Lunghezza[LC]	<ol style="list-style-type: none"><li>3. Lunghezza &lt;= 0 OR Lunghezza &gt; 30 = false [error]</li><li>4. 0 &lt; Lunghezza &lt;= 30 = true [PROPERTY LC_OK]</li></ol>



# Esistere

Per poter abbracciare ogni istante.

Parametro: Indirizzo	
Formato	
$^{\wedge}[A-z\grave{A}-\grave{u}\grave{'}-]\{2,30\}\$$	
Nome Categoria	Scelta per la categoria
Formato [FI]	3. Rispetta il formato = false [error] 4. Rispetta il formato = true [PROPERTY FI_OK]
Parametro: Numero Civico	
Nome Categoria	Scelta per la categoria
Lunghezza[LNC]	3. Lunghezza $\leq 0$ OR Lunghezza $> 6$ = false [error] 4. $0 < \text{Lunghezza} \leq 6$ = true [PROPERTY LNC_OK]
Parametro: Città	
Nome Categoria	Scelta per la categoria
Lunghezza[LC]	3. Lunghezza $\leq 0$ OR Lunghezza $> 30$ = false [error] 4. $0 < \text{Lunghezza} \leq 30$ = true [PROPERTY LC_OK]
Parametro: Numero telefono	
Formato	
$^{\wedge}((00 \backslash+ )39[\backslash. ]??)??3\backslash d\{2\}[\backslash. ]??\backslash d\{6,7\}\$$	
Nome Categoria	Scelta per la categoria
Formato [FNT]	3. Rispetta il formato = false [error] 4. Rispetta il formato = true [PROPERTY FNT_OK]

Test Case ID	Test frame	Esito
TC_1.1.2_1	FE1	Errato: E-mail non corretta
TC_1.1.2_2	FE2, FP1	Errato: Password non corretta
TC_1.1.2_3	FE2, FP2, MP1	Errato: Conferma password errata
TC_1.1.2_4	FE2, FP2, MP2, LN1	Errore: Nome non rientra nei limiti
TC_1.1.2_5	FE2, FP2, MP2, LN2, LC1	Errore: Cognome non rientra nei limiti
TC_1.1.2_6	FE2, FP2, MP2, LN2, LC2, FI1	Errato: Indirizzo Studio non corretto
TC_1.1.2_7	FE2, FP2, MP2, LN2, LC2, FI2, LNC1	Errato: Numero Civico non rientra nei limiti



# Esistere

Per poter abbracciare ogni istante.

TC_1.1.2_8	FE2, FP2, MP2, LN2, LC2, FI2, LNC2, LC1	Errato: Città Studio non rientra nei limiti
TC_1.1.2_9	FE2, FP2, MP2, LN2, LC2, FI2, LNC2, LC2, FNT1	Errato: Numero telefono studio non corretto
TC_1.1.2_10	FE2, FP2, MP2, LN2, LC2, FIS2, LNC2, LC2, FNT2	Corretto

## 11.1.2 Login

Parametro: E-mail	
FORMATO:	
^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,10}\$	
Nome Categoria	Scelta per la categoria
Match [ME]	<ol style="list-style-type: none"> <li>Match con E-mail = false [error]</li> <li>Match con E-mail = true [PROPERTY ME_OK]</li> </ol>
Parametro: Password	
Nome Categoria	Scelta per la categoria
Match [MP]	<ol style="list-style-type: none"> <li>Match con Password = false [error]</li> <li>Match con Password = true [PROPERTY MP_OK]</li> </ol>

Test Case ID	Test frame	Esito
TC_1.2_1	ME1	Errato: E-mail errata
TC_1.2_2	ME2, MP1	Errato: Password errata
TC_1.2_3	ME2, MP2	Corretto

## 11.1.3 Inserimento Dati Paziente

Parametro: Nome	
Nome Categoria	Scelta per la categoria
Lunghezza [LN]	<ol style="list-style-type: none"> <li>Lunghezza &gt; 30 OR Lunghezza &lt; 0 = false [error]</li> <li>Lunghezza &lt;= 30 = true [PROPERTY LN_OK]</li> </ol>
Parametro: Cognome	
Nome Categoria	Scelta per la categoria



Lunghezza [LCN]	<ol style="list-style-type: none"><li>1. Lunghezza &lt; 0 OR Lunghezza &gt; 60 = false [error]</li><li>2. 0 &lt; Lunghezza &lt;= 60 = true [PROPERTY LCN_OK]</li></ol>
Parametro: Codice Fiscale	
FORMATO:  /^[A-Z]{6}\d{2}[A-Z]\d{2}[A-Z]\d{3}[A-Z]\$/i	
Nome Categoria	Scelta per la categoria
Lunghezza [LCF]	<ol style="list-style-type: none"><li>1. Lunghezza != 16 = false [error]</li><li>2. Lunghezza == 16 = true [PROPERTY LCF_OK]</li></ol>
Parametro: Data di nascita	
FORMATO:  "^(0[1-9]   1[012])[-./](0[1-9]   [12][0-9]   3[01])[-./](19   20)\d\d\$"	
Nome Categoria	Scelta per categoria
Valore [DT]	<ol style="list-style-type: none"><li>1. Valore &gt; Data corrente = false [error]</li><li>2. Valore &lt; Data corrente = true [PROPERTY DT_OK]</li></ol>

Test Case ID	Test frame	Esito
TC_1.3_1	LN1	Errato: Lunghezza nome non rientra nei limiti
TC_1.3_2	LN2, LCN1	Errato: Lunghezza cognome non rientra nei limiti
TC_1.3_3	LN2, LCN2, LCF1	Errore: Lunghezza codice fiscale non rientra nei limiti
TC_1.3_4	LN2, LCN2, LCF2, DT1	Errore: Valore della data non ammesso
TC_1.3_5	LN2, LCN2, LCF2, DT2	Corretto

## 11.2 Gestione Filastrocca

### 11.2.1 Creazione Filastrocca

Parametro: Titolo	
Nome Categoria	Scelta per la categoria
Lunghezza [LT]	<ol style="list-style-type: none"><li>3. Lunghezza &lt;= 0 OR Lunghezza &gt; 30 = false [error]</li><li>4. Lunghezza &lt;= 30 = true [PROPERTY LT_OK]</li></ol>
Parametro: Testo	





Nome Categoria	Scelta per la categoria
Lunghezza [LTX]	3. Lunghezza $\leq 0$ OR Lunghezza $> 300$ = false [error] 4. $0 < \text{Lunghezza} \leq 300$ = true [PROPERTY LTX_OK]
Parametro: Autore	
Nome Categoria	Scelta per la categoria
Lunghezza [LA]	3. Lunghezza $\leq 0$ OR Lunghezza $> 30$ = false [error] 4. $0 < \text{Lunghezza} \leq 30$ = true [PROPERTY LA_OK]

Test Case ID	Test frame	Esito
TC_2.1_1	LT1	Errato: Titolo non rientra nei limiti
TC_2.1_2	LT2, LTX1	Errato: Testo non rientra nei limiti
TC_2.1_3	LT2, LTX2, LA1	Errore: Autore non rientra nei limiti
TC_2.1_4	LT2, LTX2, LA2	Corretto

## 11.3 Gestione Storie

### 11.3.1 Creazione Storia (Foto)

Parametro: Testo della Storia	
Nome Categoria	Scelta per la categoria
Lunghezza [LNT]	1. Lunghezza $\leq 0$ OR Lunghezza $> 300$ = false [error] 2. Lunghezza $\leq 300$ = true [PROPERTY LNT_OK]
Parametro: Foto	
FORMATO: /\\. (png   jpg   jpeg   bmp   tif   tiff) \$/i	
Nome Categoria	Scelta per la categoria
Formato [FF]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FF_OK]
Parametro: Descrizione Foto	
Nome Categoria	Scelta per la categoria
Lunghezza [LND]	1. Lunghezza $\leq 0$ OR Lunghezza $> 300$ = false [error] 2. Lunghezza $\leq 300$ = true [PROPERTY LND_OK]



# Esistere

Per poter abbracciare ogni istante.

Test Case ID	Test frame	Esito
TC_3.1_1	LNT1	Errato: Testo della Storia non rientra nei limiti
TC_3.1_2	LNT2, FF1	Errato: Formato File non supportato
TC_3.1_3	LNT2, FF2, LND1	Errore: Descrizione Foto non rientra nei limiti
TC_3.1_4	LNT2, FF2, LND2	Corretto

## 11.4 Gestione Quiz di Allenamento

### 11.4.1 Creazione Quiz Di Allenamento Giornaliero

Parametro: Numero Domande	
Nome Categoria	Scelta per la categoria
Valore [VND]	<ol style="list-style-type: none"><li>Valore <math>\leq 0</math> = false [error]</li><li>Valore <math>&gt; 0</math> = true [PROPERTY VND_OK]</li></ol>
Parametro: Lunghezza Domanda	
Nome Categoria	Scelta per la categoria
Lunghezza [LD]	<ol style="list-style-type: none"><li>Lunghezza <math>\leq 0</math> OR Lunghezza <math>&gt; 300</math> = false [error]</li><li>Lunghezza <math>&gt; 0</math> AND Lunghezza <math>\leq 300</math> = true [PROPERTY LD_OK]</li></ol>
Parametro: Lunghezza Risposta	
Nome Categoria	Scelta per la categoria
Lunghezza [LR]	<ol style="list-style-type: none"><li>Lunghezza <math>\leq 0</math> OR Lunghezza <math>&gt; 300</math> = false [error]</li><li>Lunghezza <math>&gt; 0</math> AND Lunghezza <math>\leq 300</math> = true [PROPERTY LR_OK]</li></ol>
Parametro: Risposta corretta selezionata	
Nome Categoria	Scelta per la categoria
Selezione [SDC]	<ol style="list-style-type: none"><li>La selezione non è stata fatta = false [error]</li><li>La selezione è stata fatta = true [PROPERTY SDC_OK]</li></ol>



# Esistere

Per poter abbracciare ogni istante.

Test Case ID	Test frame	Esito
TC_4.1_1	VND1	Errore: Il valore del numero di domande non è consentito
TC_4.1_2	VND2, LD1	Errore: Una o più domande presentano una lunghezza non consentita
TC_4.1_3	VND2, LD2, LR1	Errore: Una o più risposte presentano una lunghezza non consentita
TC_4.1_4	VND2, LD2, LR2, SDC1	Errore: Per una o più domande non è stata selezionata la risposta corretta
TC_4.1_5	VND2, LD2, LR2, SDC2	Corretto

## 11.5 Gestione ToDoList

### 11.5.1 Creazione ToDoList

Parametro: Numero Attività	
Nome Categoria	Scelta per la categoria
Valore [VN]	<ol style="list-style-type: none"><li>1. Valore <math>\leq 0</math> = false [error]</li><li>2. Valore <math>&gt; 0</math> = true [PROPERTY VN_OK]</li></ol>
Parametro: Descrizione Attività	
Nome Categoria	Scelta per la categoria
Lunghezza [LNA]	<ol style="list-style-type: none"><li>1. Lunghezza <math>&gt; 300</math> = false [error]</li><li>2. Lunghezza <math>\leq 300</math> = true [PROPERTY LNA_OK]</li></ol>

Test Case ID	Test frame	Esito
TC_5.1_1	VN1	Errato: Non sono presenti attività sufficienti
TC_5.1_2	VN2, LNA1	Errato: Descrizione Attività non rientra nei limiti
TC_5.1_3	VN2, LNA2	Corretto