



TP: Test Plan

HEALTH CARE

Riferimento	2022_TCS_C03_ver0.2 2022_MT_C03
Versione	0.4
Data	09/02/2023
Destinatario	Prof.ssa F. Ferrucci, Prof. F. Palomba
Presentato da	Giusy Castaldo, Mariarosaria Esposito, Francesca Perillo, Giovanni Borrelli, Gerardo di Muro, Agostino Andrea Mangia, Giovanni Mercurio, Luca Pastore, Angelo Zuottolo
Approvato da	Giusy Castaldo, Mariarosaria Esposito, Francesca Perillo



Revision History

Data	Versione	Descrizione	Autori
01/12/2022	0.1	Creazione	Castaldo Giusy Esposito Marianosaria Perillo Francesca
09/12/2022	0.2	Apertura di un Form da parte di un MMG	Angelo Zuottolo
09/12/2022	0.2	Registrazione MMG	Giovanni Mercurio
09/12/2022	0.2	Inserimento di un intervento da parte di un MMG ad un form aperto	Luca Pastore
09/12/2022	0.2	Modifica dati anagrafici dell'MMG	Gerardo Di Muro
10/12/2022	0.3	Inserimento sintomi del paziente e individuazione di una malattia rara	Agostino Andrea Mangia
10/12/2022	0.3	Ricerca di una Malattia Rara	Giovanni Borrelli
09/02/2023	0.4	Revisione documentazione	Tutti



Sommario

1.Introduzione	5
2.Riferimenti ai documenti correlati.....	5
2.1 Relazione con il Requirements Analysis Document (RAD)	5
2.2 Relazione con il System Design Document (SDD)	5
3. Panoramica del sistema.....	6
4. Dettagli per il testing	7
4.1 Funzionalità da testare	7
4.2 Funzionalità da non testare.....	8
4.3 Criteri di successo e fallimento dei test	8
4.4 Criteri di sospensione, ripristino e terminazione	8
4.4.1 Criteri di sospensione	8
4.4.2 Criteri di ripristino	8
4.4.3 Criteri di terminazione	9
5. Approccio	9
5.1 Testing di unità	9
5.2 Testing di sistema.....	9
6. Materiale di testing	10
7. Test cases	10
TP_1: Registrazione Medico di Medicina Generale	10
TP_8: Inserimento di un intervento da parte di un MMG ad un form aperto.....	16
TP_18: Ricerca di una Malattia Rara.....	17
TP_5: Apertura di un form da parte di un MMG	18
TP_17: Ricerca di una Malattia Rara tramite sintomi	20
TP_4: Modifica dei dati inseriti dall' MMG in fase di registrazione.....	21



8. Testing schedule	26
9. Riferimenti.....	26



1.Introduzione

L'obiettivo di **HealthCare** è quello di velocizzare la diagnosi per una Malattia Rara da parte di un Medico di Medicina Generale (MMG) per un dato paziente.[1] Data l'importanza che il prodotto ricopre, è d'obbligo analizzare e gestire lo sviluppo e le attività di testing riguardanti il software che si sta sviluppando in modo che questo risulti idealmente privo di tutti quegli errori che potrebbero essere prodotti in fase di implementazione. Lo scopo è quello di far sì che il prodotto diventi uno strumento fidato per un MMG, che possa fidarsi ciecamente senza paura di commettere sbagli irreversibili. Questo documento è volto a verificare il corretto funzionamento del prodotto in diversi casi al fine di confrontare il comportamento atteso con il comportamento osservato. Nel corso del documento si discuteranno i risultati ottenuti per comprendere dove intervenire al fine di rimuovere i fault presenti nel sistema.

2.Riferimenti ai documenti correlati

All'interno dei documenti precedenti è stato già definito il comportamento di alcuni servizi che ci aiuterà a verificare se il funzionamento del sistema progettato corrisponde al reale funzionamento del sistema implementato. Per tale motivo, l'individuazione dei test case è correlata ai documenti prodotti fino ad ora. Di seguito vediamo nello specifico le relazioni con ognuno di essi.

2.1 Relazione con il Requirements Analysis Document (RAD)

Per la pianificazione dei casi di test sono necessarie le specifiche, in particolare vi è una relazione con i requisiti funzionali e non funzionali discussi nel RAD.

Link alla documentazione: [RAD HealthCare v1.2.pdf](#)

2.2 Relazione con il System Design Document (SDD)

All'interno dell'SDD è presentata la suddivisione in sottosistemi, la quale deve essere rispettata dai testcases che verranno illustrati nel seguente documento.

Link alla documentazione: [SDD HealthCare v1.3.pdf](#)



3. Panoramica del sistema

Il sistema proposto è un software che si occupa di fornire un supporto ai Medici di Medicina Generale per diagnosticare patologie rare attraverso l'inserimento dei sintomi di un paziente determinando la patologia di cui esso è affetto, evitando tutti quei procedimenti onerosi che porterebbero una patologia a giungere in uno stadio avanzato e difficile da trattare. L'utilizzo di questo software determinerà un miglioramento in termini di risorse e tempo, garantendo una diagnosi più rapida in modo da poter fornire uno strumento di supporto per il rilevamento di Malattie Rare. Sfruttando Form dedicati, il software permetterà ai medici di interagire tra di loro creando un ambiente di confronto utile a risolvere dubbi in merito a patologie difficili da diagnosticare.

Dopo un'attenta analisi del nostro software, abbiamo ritenuto opportuno utilizzare un'architettura di tipo Three-tier, in cui l'interfaccia utente, i processi logico funzionali, l'archiviazione informatica dei dati e l'accesso ai dati sono sviluppate e mantenute come moduli indipendenti, la maggior parte delle volte su piattaforme separate. L'architettura three-tier è un'applicazione software ben consolidata che organizza applicazioni in tre tier di calcolo logici e fisici:

- il tier di presentazione, o interfaccia utente;
- il tier dell'applicazione, dove vengono elaborati i dati;
- il tier dei dati, dove vengono archiviati e gestiti i dati associati all'applicazione.

Il vantaggio principale dell'architettura three-tier è che, poiché ciascun tier viene eseguito sulla propria infrastruttura, può essere sviluppato contemporaneamente da un team di sviluppo separato e può essere aggiornato o scalato in base alle necessità senza effetti sugli altri tier. Il tier di presentazione è l'interfaccia utente ed il livello di comunicazione dell'applicazione, in cui l'utente finale interagisce con l'applicazione. Il suo scopo principale è quello di visualizzare le informazioni e raccogliere informazioni dall'utente. Il tier dell'applicazione, noto anche come tier logico o tier intermedio, è il cuore dell'applicazione. In questo tier, le informazioni raccolte nel tier di presentazione vengono elaborate, a volte rispetto ad altre informazioni nel tier dei dati, utilizzando la logica aziendale, una serie specifica di regole aziendali. Il tier dell'applicazione può anche aggiungere, eliminare o modificare i dati nel tier dei dati.

Il tier dei dati, in alcuni casi chiamato tier del database, tier di accesso ai dati o back-end, è il punto in cui vengono archiviate e gestite le informazioni elaborate dall'applicazione. Nel nostro sistema si prevede l'utilizzo un tipo di database non relazionale (MongoDB).



Abbiamo deciso di utilizzare un'architettura three-tier, inoltre, per i seguenti motivi:

- **Sviluppo più veloce:** poiché ciascun tier può essere sviluppato contemporaneamente da team differenti, un'organizzazione può immettere l'applicazione sul mercato più velocemente e i programmatori possono utilizzare i linguaggi e gli strumenti più recenti e migliori per ogni tier.
- **Scalabilità migliorata:** qualsiasi tier può essere scalato indipendentemente dagli altri in base alle necessità.
- **Massima affidabilità:** un'interruzione in un tier avrà meno probabilità di avere impatto sulla disponibilità o sulle prestazioni degli altri tier.
- **Sicurezza migliorata:** poiché il tier di presentazione e il tier dei dati non riescono a comunicare direttamente, un tier dell'applicazione ben progettato può funzionare come una sorta di firewall interno, impedendo SQL Injection e altri exploit dannosi.

4. Dettagli per il testing

4.1 Funzionalità da testare

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

- **Gestione Utente**

- o Registrazione di un MMG (Alta)

- o Login di un MMG (Alta)

- **Gestione dei Form**

- o Apertura di un Form (Alta)

- o Chiusura di un Form (Alta)

- o Modifica di un Form (Medio/Alta)

- o Intervento in un Form (Alta)

- o Modifica dell'intervento in un Form (Medio/Alta)



o Esplorazione dei form non propri del MMG in questione (Alta)

- **Gestione dei Sintomi e delle Malattie**

o Inserimento dei sintomi del paziente e individuazione della Malattia Rara (Alta)

o Ricerca di una Malattia Rara (Alta)

4.2 Funzionalità da non testare

Le funzionalità di cui non si andrà ad effettuare le attività di testing riguardano requisiti funzionali di Medio/Bassa priorità. Non saranno testate le funzionalità relative alla visualizzazione.

4.3 Criteri di successo e fallimento dei test

L'attività di testing ha l'obiettivo di verificare la correttezza nel funzionamento del prodotto software realizzato. Ciò avviene individuando ed eliminando quanti più errori possibili all'interno del sistema prima che quest'ultimo venga rilasciato.

L'individuazione degli errori necessita di un oracolo che rispecchia il risultato atteso dell'esecuzione. A quest'ultimo verrà confrontato l'esito effettivo dell'operazione effettuata. Se i due corrispondono allora parliamo di successo, in caso contrario di fallimento.

Il criterio che useremo per individuare un successo sarà **PASS**, mentre per il fallimento useremo **FAIL**.

4.4 Criteri di sospensione, ripristino e terminazione

In questa sezione verranno descritti i criteri di sospensione, ripresa e terminazione del testing e le attività di test che dovranno essere ripetute quando si riprenderà il testing.

4.4.1 Criteri di sospensione

La fase di testing verrà sospesa nel caso in cui verranno riscontrati errori che non permetteranno la riuscita del test. I membri del team si occuperanno prima a risolvere tali errori.

4.4.2 Criteri di ripristino

L'approccio per lo sviluppo del software è di tipo incrementale; pertanto, la fase di testing riprenderà nel momento in cui verranno introdotti i cambiamenti e/o inserite nuove componenti.



4.4.3 Criteri di terminazione

Il test si considera concluso nel momento in cui verranno soddisfatti i vincoli tecnici per il testing stabiliti all'interno del documento SOW – Statement Of Work [2].

5. Approccio

L'approccio nella fase di testing avviene mediante tre fasi cardine. Prima di tutto verrà effettuato il testing di unità per individuare gli errori di una singola componente; successivamente verrà effettuato il test di integrazione, nel quale saranno testate le funzionalità nate dall'integrazione delle componenti; infine verrà effettuato il testing di sistema per verificare che le caratteristiche richieste da **OMAR**, nostro committente, siano rispettate o meno.

5.1 Testing di unità

In questa prima fase si effettuerà il controllo delle varie classi e metodi del sistema. Il testing di unità si focalizza sostanzialmente sul comportamento di una componente e verrà svolto in modalità black-box per testare i componenti I/O delle singole componenti senza tenere conto della loro struttura interna. Dato l'onere nel testare tutti i possibili input questi verranno divisi in classi di equivalenza e ogni componente avrà un singolo caso di test per ogni classe di equivalenza strutturata. Sfruttando lo studio delle classi di equivalenza il test verrà reso più efficiente. Qualora sia presente un caso di errore la strategia sarà quella di correggerlo tempestivamente per poi ri-eseguire una verifica della correzione effettuata.

5.2 Testing di sistema

Prima di essere pronto all'uso, il sistema concluso dovrà transitare per l'ultima fase di test che verrà effettuata tramite un apposito tool. Il testing di sistema concluderà la fase di test del prodotto ed il primo ciclo di sviluppo.



6. Materiale di testing

Il materiale di testing vede una definizione tra l'hardware e il software necessario. Per quanto concerne il primo, al fine di effettuare i testing sarà necessario un computer, non necessariamente connesso ad una rete internet. Per il secondo, invece, occorre far riferimento a framework di supporto per Java.

7. Test cases

TP_1: Registrazione Medico di Medicina Generale

Parametro	Tipo	Numeri caratteri
Nome	String	255
Cognome	String	255
Mail	String	255
Indirizzo	String	255
Provincia	String	255
Comune	String	255
Sesso	String	3
Telefono	String	10
Password	String	24
Conferma Password	String	24

Parametro: Nome Formato: <code>^[a-z A-Z àéèìòù]{2,255}\$</code>	
Categoria	Scelte
Lunghezza ln	1: lunghezza < 2 [errore] 2: lunghezza > 255 [errore] 3: lunghezza >= 2 && lunghezza <= 255 [property lunghezzaLNok]
Formato fn	1: non rispetta il formato [errore] 2: rispetta il formato [property formatoFNok]



Parametro: Cognome Formato: <code>^[a-z A-Z àèèìòù]{2,255}\$</code>	
Categoria	Scelte
Lunghezza lc	1: lunghezza < 2 [errore] 2: lunghezza > 255 [errore] 3: lunghezza >= 2 && lunghezza <= 255 [property lunghezzaLCok]
Formato fc	1: non rispetta il formato [errore] 2: rispetta il formato [property formatoFCok]

Parametro: Mail Formato: <code>^(?!.{256,})[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z0-9]+\$</code>	
Categoria	Scelte
Formato fe	1: non rispecchia il formato [errore] 2: rispecchia il formato [property Lunghezza FEok]
Esiste nel database ee	1: Non esiste nel database [if formatoFEok] [errore] 2: Esiste nel database [if formatoFEok] [property esisteEEok]
Lunghezza le	1: lunghezza < 2 [errore] 2: lunghezza > 255 [errore] 3: lunghezza >= 2 && lunghezza <= 255 [property lunghezzaLEok]



Parametro: Indirizzo	
Formato: <code>^[a-z A-Z 0-9 àéèìòù ',-]{2,255}\$</code>	
Categoria	Scelte
Lunghezza li	1: lunghezza < 2 [errore] 2: lunghezza > 255 [errore] 3: lunghezza >= 2 && lunghezza <= 255[property lunghezzaLIok]
Formato fi	1: non rispetta il formato [if lunghezzaLIok] [errore] 2: rispetta il formato [if lunghezzaLIok] [property formatoFIok]

Parametro: Provincia	
Formato: <code>^[a-z A-Z àéèìòù]{2,255}\$</code>	
Categoria	Scelte
Lunghezza pv	1: lunghezza < 2 [errore] 2: lunghezza > 255 [errore] 3: lunghezza >= 2 && lunghezza <= 255[property lunghezzaPVok]
Formato fp	1: non rispetta il formato [if lunghezzaPVok] [errore] 2: rispetta il formato [if lunghezzaPVok] [property formatoFPok]



Parametro: Comune	
Formato: <code>^[a-z A-Z àéèìòù]{2,255}\$</code>	
Categoria	Scelte
Lunghezza cl	1: lunghezza < 2 [errore] 2: lunghezza > 255 [errore] 3: lunghezza >= 2 && lunghezza <= 255 [property lunghezzaCLok]
Formato cf	1: non rispetta il formato [if lunghezzaCLok] [errore] 2: rispetta il formato [if lunghezzaCLok] [property formatoCFok]

Parametro: Sesso	
Formato: <code>^[a-z A-Z àéèìòù]{2,255}\$</code>	
Categoria	Scelte
Formato sf	1: non rispetta il formato [errore] 2: rispetta il formato [property formatoSFok]

Parametro: Telefono	
Formato: <code>^(\\ + \\d{1,2} \\s)?1? \\-? \\-? \\s? \\(? \\d{3} \\) ?[\\s.-]? \\d{3} [\\s.-]? \\d{4}\$</code>	
Categoria	Scelte
Lunghezza lf	1: lunghezza > 11 [errore] 2: lunghezza < 10 [errore] 3: lunghezza >=10 & <=11 [property formatoLFok]
Formato tf	1: non rispetta il formato [errore] 2: rispetta il formato [property formatoTFok]



Parametro: Password Formato: $^{\wedge}([^\wedge \backslash s])^*[0-9])([^\wedge \backslash s])^*[a-zA-Z])([^\wedge \backslash s])\{8,24\}\$$	
Categoria	Scelte
Lunghezza lp	1: lunghezza < 8 [errore] 2: lunghezza > 24 [errore] 3: lunghezza >= 8 && lunghezza <= 24 [property lunghezzaLPok]
Formato tp	1: non rispetta il formato non rispetta il formato [if lunghezzaLP] [errore] 2: rispetta il formato [if lunghezzaLP] [property formatoTPok]

Parametro: Conferma Password Formato: $^{\wedge}([^\wedge \backslash s])^*[0-9])([^\wedge \backslash s])^*[a-zA-Z])([^\wedge \backslash s])\{8,24\}\$$	
Categoria	Scelte
Lunghezza clp	1: lunghezza < 8 [errore] 2: lunghezza > 24 [errore] 3: lunghezza >= 8 && lunghezza <= 24 [property lunghezzaCLPok]
Formato ctp	1: non rispetta il formato non rispetta il formato [if lunghezzaCLP] [errore] 2: rispetta il formato [if lunghezzaCLP] [property formatoCTPok]
Corrispondenza cp	1: Non corrisponde [if lunghezzaCLPok and formatoCTPok] [errore] 2: Corrisponde [if lunghezzaCLPok and formatoCTPok] [property corrispondenzaCPok]



Test Case ID	Test Frame	Esito
TP_1.1	ln1	errore
TP_1.2	ln2	errore
TP_1.3	ln3, fn1	errore
TP_1.4	ln3, fn2, lc1	errore
TP_1.5	ln3, fn2, lc2	errore
TP_1.6	ln3, fn2, lc3, fc1	errore
TP_1.7	ln3, fn2, lc3, fc2, fe1	errore
TP_1.8	ln3, fn2, lc3, fc2, fe2, ee1	errore
TP_1.9	ln3, fn2, lc3, fc2, fe2, ee2, le1	errore
TP_1.10	ln3, fn2, lc3, fc2, fe2, ee2, le2	errore
TP_1.11	ln3, fn2, lc3, fc2, fe2, ee2, le3, li1	errore
TP_1.12	ln3, fn2, lc3, fc2, fe2, ee2, le3, li2	errore
TP_1.13	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi1	errore
TP_1.14	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv1	errore
TP_1.15	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv2	errore
TP_1.16	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp1	errore
TP_1.17	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl1	errore
TP_1.18	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl2	errore
TP_1.19	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf1	errore
TP_1.20	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf1	errore
TP_1.21	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf1	errore
TP_1.22	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf2	errore
TP_1.23	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf3, tf1	errore
TP_1.24	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf3, tf2, lp1	errore
TP_1.25	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf3, tf2, lp2	errore
TP_1.26	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf3, tf2, lp3, tp1	errore
TP_1.27	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf3, tf2, lp3, tp2, clp1	errore
TP_1.28	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf3, tf2, lp3, tp2, clp2	errore
TP_1.29	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf3, tf2, lp3, tp2, clp3, ctp1	errore
TP_1.30	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf3, tf2, lp3, tp2, clp3, ctp2, cp1	errore
TP_1.31	ln3, fn2, lc3, fc2, fe2, ee2, le3, li3, fi2, pv3, fp2, cl3, cf2, sf2, lf3, tf2, lp3, tp2, clp3, ctp2, cp2	successo



TP_8: Inserimento di un intervento da parte di un MMG ad un form aperto

Parametro	Tipo	Numeri caratteri
Descrizione	String	800

Parametro: Descrizione	
Formato: <code>^[\\p{L}0-9\\s\\w\\d\\,\\.\\?\\!:-]{2,800}\$</code>	
Categoria	Scelte
Lunghezza ld	1: lunghezza < 2 [errore] 2: lunghezza > 800 [errore] 3: lunghezza > 2 && lunghezza <= 800 [property lunghezzaLDok]

Codice	Combinazione	Esito
TP_8.1	ld1	errore
TP_8.2	ld2	errore
TP_8.3	ld3	successo



TP_18: Ricerca di una Malattia Rara

Parametro: Malattia	
Formato: <code>^[a-zA-Z0-9.\s]{2,40}\$</code>	
Categoria	Scelte
Lunghezza[LS]	1. Lunghezza < 1 OR Lunghezza > 40 [errore] 2. Lunghezza >= 1 AND Lunghezza <= 40 [PROPERTY_LS_OK]
Categoria	Scelte
Esistenza[ES]	1. Malattia \notin DB [errore] 2. Malattia \in DB [PROPERTY_ES_OK]

Test Case ID	Test Frame	Esito
TCS_18.1	LS1	errato: lunghezza malattia
TCS_18.2	LS2, ES1	errato: malattia non valida
TCS_18.3	LS2, ES2	corretto



Test Case ID	Test Frame	Esito
TP_5.1	LTit1	errato: lunghezza titolo incorretta
TP_5.2	LTit2, FTit1	errato: formato titolo incorretto
TP_5.3	LTit2, FTit2, LD1	errato: descrizione incorretta
TP_5.4	LTit2, FTit2, LD2	corretto



TP_17: Ricerca di una Malattia Rara tramite sintomi

Parametro: sintomi	
Formato: ^[\p{L}\s]{3,50}\$	
Categoria	Scelte
Lunghezza [LS]	1. Lunghezza < 1 OR Lunghezza > 50 [errore] 2. Lunghezza >= 1 AND Lunghezza <= 50 [PROPERTY_LS_OK]
Categoria	Scelte
Esistenza [ES]	1. Sintomi \notin DB [errore] 2. Sintomo[i] \in DB AND Sintomo[j] \notin DB [errore] 3. Sintomi \in DB [PROPERTY_ES_OK]

Test Case ID	Test Frame	Esito
TP_17.1	LS1	errato: lunghezza sintomi incorretta
TP_17.2	LS2, ES1	errato: sintomi non validi
TP_17.3	LS2, ES2	errato: le malattie risultanti sono associate ai soli sintomi contenuti nel DB
TP_17.4	LS2, ES3	corretto



TP_4: Modifica dei dati inseriti dall' MMG in fase di registrazione

Parametro	Tipo	Numeri caratteri
Mail	String	255
Indirizzo	String	255
Provincia	String	255
Comune	String	255
Telefono	String	10
Password	String	24
Conferma Password	String	24

Parametro: Mail Formato: <code>^(?!.{256,})[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-zA-Z0-9]+\$</code>	
Categoria	Scelte
Formato fe	1: non rispetta il formato [errore] 2: rispetta il formato [property Lunghezza FEok]
Esiste nel database ee	1: Non esiste nel database [errore] 2: Esiste nel database [property esisteEEok]



Parametro: Indirizzo	
Formato: <code>^[a-z A-Z 0-9 àéèìòù ',.-]{2,255}\$</code>	
Categoria	Scelte
Lunghezza li	1: lunghezza < 2 [errore] 2: lunghezza > 255 [errore] 3: lunghezza >= 2 && lunghezza <= 255 [property lunghezzaLIok]
Formato fi	1: non rispetta il formato [if lunghezzaLIok] [errore] 2: rispetta il formato [if lunghezzaLIok] [property formatoFIok]

Parametro: Provincia	
Formato: <code>^[a-z A-Z àéèìòù]{2,255}\$</code>	
Categoria	Scelte
Lunghezza pv	1: lunghezza < 2 [errore] 2: lunghezza > 255 [errore] 3: lunghezza >= 2 && lunghezza <= 255 [property lunghezzaPVok]
Formato fp	1: non rispetta il formato [if lunghezzaPVok] [errore] 2: rispetta il formato [if lunghezzaPVok] [property formatoFPok]



Parametro: Comune	
Formato: <code>^[a-z A-Z àéèìòù]{2,255}\$</code>	
Categoria	Scelte
Lunghezza cl	1: lunghezza < 2 [errore] 2: lunghezza > 255 [errore] 3: lunghezza >= 2 && lunghezza <= 255 [property lunghezzaCLok]
Formato cf	1: non rispetta il formato [if lunghezzaCLok] [errore] 2: rispetta il formato [if lunghezzaCLok] [property formatoCFok]

Parametro: Telefono	
Formato: <code>^(\\+\\d{1,2}\\s)?1?\\-?\\-?\\s?(?\\d{3}\\s)?[\\s.-]?\\d{3}[\\s.-]?\\d{4}\$</code>	
Categoria	Scelte
Lunghezza lf	1: lunghezza > 10 [errore] 2: lunghezza < 10 [errore] 3: lunghezza = 10 [property LunghezzaLfok]
Formato tf	1: non rispetta il formato [errore] 2: rispetta il formato [property formatoTFok]



Parametro: Password Formato: $^?(=[^{\wedge}\backslash s]^*[0-9])(?=[^{\wedge}\backslash s]^*[a-zA-Z])([^{\wedge}\backslash s])\{8,24\}\$$	
Categoria	Scelte
Lunghezza lp	1: lunghezza < 8 [errore] 2: lunghezza > 24 [errore] 3: lunghezza >= 8 && lunghezza <= 24 [property lunghezzaLPok]
Formato tp	1: non rispetta il formato non rispetta il formato [if lunghezzaLP] [errore] 2: rispetta il formato [if lunghezzaLP] [property formato TPok]

Parametro: Conferma Password Formato: $^?(=[^{\wedge}\backslash s]^*[0-9])(?=[^{\wedge}\backslash s]^*[a-zA-Z])([^{\wedge}\backslash s])\{8,24\}\$$	
Categoria	Scelte
Lunghezza clp	1: lunghezza < 8 [errore] 2: lunghezza > 24 [errore] 3: lunghezza >= 8 && lunghezza <= 24 [property lunghezzaCLPok]
Formato ctp	1: non rispetta il formato non rispetta il formato [if lunghezzaCLP] [errore] 2: rispetta il formato [if lunghezzaCLP] [property formato CTPok]
Corrispondenza cp	1: Non corrisponde [if lunghezzaCLPok and formatoCTPok] [errore] 2: Corrisponde [if lunghezzaCLPok and formatoCTPok] [property corrispondenzaCPok]



Test Case ID	Test Frame	Esito
TP_4.1	fe1	errore
TP_4.2	fe2, ee1	errore
TP_4.3	fe2, ee2, li1	errore
TP_4.4	fe2, ee2, li2	errore
TP_4.5	fe2, ee2, li3, fi1	errore
TP_4.6	fe2, ee2, li3, fi2, pv1	errore
TP_4.7	fe2, ee2, li3, fi2, pv2	errore
TP_4.8	fe2, ee2, li3, fi2, pv3, fp1	errore
TP_4.9	fe2, ee2, li3, fi2, pv3, fp2, cl1	errore
TP_4.10	fe2, ee2, li3, fi2, pv3, fp2, cl2	errore
TP_4.11	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf1	errore
TP_4.12	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf1	errore
TP_4.13	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf2	errore
TP_4.14	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf3, tf1	errore
TP_4.15	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf3, tf2, lp1	errore
TP_4.16	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf3, tf2, lp2	errore
TP_4.17	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf3, tf2, lp3, tp1	errore
TP_4.18	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf3, tf2, lp3, tp2, clp1	errore
TP_4.19	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf3, tf2, lp3, tp2, clp2	errore
TP_4.20	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf3, tf2, lp3, tp2, clp3, ctp1	errore
TP_4.21	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf3, tf2, lp3, tp2, clp3, ctp2, cp1	errore
TP_4.22	fe2, ee2, li3, fi2, pv3, fp2, cl3, cf2, lf3, tf2, lp3, tp2, clp3, ctp2, cp2	successo



8. Testing schedule

Relativamente ai test, i periodi temporali saranno i seguenti (le informazioni non riportate di seguito saranno contenute nei documenti relativi alle schedule):

- le attività di pianificazione del testing avverranno dopo la fase di design necessaria per la pianificazione;
- la scrittura dei casi di test avverrà con lo sviluppo del codice;
- l'esecuzione dei test avverrà sia durante che dopo l'implementazione del sistema.

Una volta concluso lo sviluppo, tutti i test saranno rieseguiti per garantirne il corretto funzionamento e produrre i report finali.

9. Riferimenti

- [1] Business Case Document. Autori: Castaldo Giusy, Esposito Mariarosaria, Perillo Francesca.
https://unisalerno-my.sharepoint.com/:b:/g/personal/f_perillo11_studenti_unisa_it/EQfK4rL39O5EuET1FrSXMQYBxweeZwSDQXGn02mV4F9Qog?e=2aG3oY
- [2] Statement Of Work Document. Autori: Prof.ssa Filomena Ferrucci, Castaldo Giusy, Esposito Mariarosaria, Perillo Francesca.
https://unisalerno-my.sharepoint.com/:b:/g/personal/f_perillo11_studenti_unisa_it/EcBhCex-TBxJgnqZ5S_hiEoBB_uUkTuAQBXk94JohkmY9w?e=lGTJKA