TirocinioSmart

TP
Test Plan

TirocinioSmart

Aniello Carotenuto
Giuseppe Dello Stretto
Rosanna Coccaro

Data	Versione	Cambiamenti	Autori
13/12/2017	1		RC
04/02/2018	1.1	Aggiunta combinazioni delle scelte del category partition, modifica descrizione documento e revisione	GDS
08/02/2018	1.1.1	Revisione finale	AC

Sommario

1. Introduzione

Al fine di consentire ad uno studente di maturare un bagaglio di esperienze e competenze realmente utili al suo ingresso nel mercato del lavoro, abbiamo messo in campo un sistema in gradi di offrire supporto non solo al tirocinante stesso ma a tutte le entità coinvolte nell'ambito della gestione dei tirocini esterni.

Nell'affrontare alcuni aspetti fondamentali del sistema TirocinioSmart ci siamo posti delle domande:

Come ottenere un buon prodotto? Può rimanere soltanto un ideale?

Beh, è difficile dare una risposta a tali domande ma sicuramente possiamo garantire all'utente una buona fruizione delle funzionalità offerte se mettiamo in campo un ottimo strumento in grado di migliorare la sua esperienza.

Nasce così la necessità di rilevare eventuali errori prodotti durante la fase di implementazione per evitare che essi si presentino nel momento in cui sistema verrà utilizzato dall'utente finale.

A tal proposito definiamo un piano di test il cui obiettivo principale è quello di analizzare e gestire lo sviluppo delle attività di testing relative al sistema proposto.

In questa fase occorre verificare il corretto funzionamento del sistema sotto determinate condizioni.

Abbiamo pensato a opportuni casi e dati di input specifici in grado di mettere alla prova ogni singola funzionalità e caratteristica offerta dalla piattaforma.

I risultati dei test che veranno eseguiti saranno il punto cruciale nell'analisi delle failure e delle loro cause (fault) per individuare dove bisognerà intervenire per correggere gli errori o apportare modifiche per il miglioramento dei vari sottosistemi.

Lo scopo è quindi stabilire la verità sulla corrispondenza tra comportamento atteso e comportamento osservato da TirocinioSmart.

2. Documenti correlati

2.1 Relazione con il documento di analisi

La progettazione dei casi di test avviene prescindendo dalla conoscenza della struttura interna del prodotto ed operando solo sulle specifiche.

Per questo motivo facciamo riferimento al contenuto del documento di analisi che descrive dettagliatamente le funzionalità del sistema attraverso scenari e use case.

2.2 Relazione con il System Design Document

Nel system design document abbiamo definito la suddivisione in sottosistemi relativamente al prodotto che intendiamo presentare.

Il sistema è suddiviso in tre livelli logici: presentazione, business e persistenza. Ogni livello è composta da vari sottosistemi.

In questa fase è importante focalizzare la nostra attenzione sul layer di business. Infatti, pianificheremo le attività di testing relative alle funzionalità garantite nei sottosistemi specificati all'interno nel System Design Document relativamente al livello business.

2.2 Relazione con l'Object Design Document

Nel documento di Object Design sono state definite le classi che compongono il sistema e le loro mansioni. Faremo riferimento ad esse nel corso del documento per associare i test al codice prodotto.

3. Panoramica del sistema

TirocinioSmart è una piattaforma web che mira all'ottimizzazione dei servizi fruibili dalle entità coinvolte nella gestione dei tirocini.

Il sistema che proponiamo prevede tre attori principali:

- Studente: ha la possibilità di inviare una domanda di tirocinio all'azienda che più rispecchia le caratteriste desiderate in base ad un'attenta analisi dei progetti formativi proposti dalla stessa
- Delegato Aziendale: può richiedere una convenzione con il Dipartimento di Informatica dell'Università degli Studi di Salerno: tale richiesta rappresenta l'impegno ad accogliere tirocinanti presso la struttura che rappresenta
- Impiegato dell'Ufficio Tirocini: ha la possibilità di accettare o rifiutare domande di tirocinio, richieste di iscrizione alla piattaforma da parte di studenti e richieste di convenzionamento pervenute dall'azienda

Nel System Design abbiamo definito l'architettura della piattaforma che si articola in tre layer: presentazione, business e persistenza. I sottosistemi che compongono i vari livelli logici collaborano tra loro cercando però di garantire il più possibile basso accoppiamento ed alta coesione.

I sottosistemi individuati nel business layer sono:

- Utenza: definisce l'utente generico del sistema ed offre tutti i servizi relativi all'autenticazione
- Convenzioni: modella tutto ciò che riguarda il processo di convenzionamento e le aziende convenzionate e ne definisce le operazioni
- Studenti: modella gli studenti ed il processo d'iscrizione alla piattaforma, definendo tutte le operazioni ad essi relative
- Impiegati: modella gli impiegati dell'ufficio tirocini
- Progetti formativi: modella i progetti formativi offerti dalle aziende e le relative operazioni
- Domande tirocinio: modella le domande di tirocinio e le operazioni ad esse associate

4. Funzionalità da testare

La fase di testing avrà come obbiettivo quello di testare interamente la comunicazione tra webapp e database sottostante, così come l'implementazione della logica di business e dei servizi offerti dai sottosistemi del livello centrale. Tale approccio potrebbe condurre all'esclusione dal testing (per motivi di budget) delle componenti che realizzano il layer web, ma errori a questo livello non arrecheranno danni al sistema per la robustezza (garantita) del livello del livello sottostante.

Il testing funzionale riguarderà nel dettaglio le funzionalità di seguito elencate (in base al sottosistema che le realizza):

- **Studente**
  - Registrazione alla piattaforma
- **Domande tirocinio**
  - Invio di una domanda di tirocinio
- **Convenzioni**
  - Invio di una richiesta di convenzionamento
- **Progetti formativi**
  - Aggiunta di un nuovo progetto formativo

Non saranno testate invece:

- Interfacce utente
- Sicurezza
- Performance

5. Criteri Pass/Failed

Abbiamo determinato un insieme di input possibili che possano aiutarci a scovare errori nel sistema.

Pertanto, il test ha successo se il comportamento osservato è diverso dal comportamento specificato nei requisiti funzionali. Ciò significa che raggiungiamo gli obiettivi che ci siamo posti durante questa fase se il test individuerà dei fault nel sistema.

In tal caso analizzeremo i sottosistemi coinvolti nell'errore, ed itereremo la fase di testing per verificare che le modifiche apportate agli stessi non abbiamo avuto impatti negativi su altre componenti del sistema.

Il testing fallirà se gli non saranno scovati errori nelle componenti.

6. Approccio

La fase di testing si compone di tre attività: una prima fase si occuperà di trovare errori in una singola componente; la seconda fase, invece, avrà come compito quello di testare le funzionalità nate dall'integrazione dei vari sottosistemi e per ultimo andremo a testare l'intero sistema assemblato al fine di verificare soprattutto che esso soddisfi i desideri del cliente.

Di seguito verranno descritte brevemente le strategie individuate per effettuare il test di unità, d'integrazione e di sistema.

6.1 Testing di unità

Durante la fase di testing è necessario testare singole componenti al fine di evidenziarne gli errori. Il testing di unità infatti si focalizza sul comportamento di una componente permettendo di eseguire testing in modalità black-box o white-box.

Le nostre componenti saranno testate secondo il metodo white-box. Infatti durante questa fase poseremo la nostra sulla struttura del codice che realizza le funzionalità fornite dalla componente al fine di individuare errori sia di logica che di implementazione.

6.2 Testing di integrazione

Una volta che sono stati rilevati i bug per una singola componente e riparati, le componenti sono pronte per essere integrate in sottosistemi più grandi. Pertanto dopo aver testato singolarmente le componenti del sistema, possiamo procedere a testarne le integrazioni.

Per effettuare l'integration testing abbiamo pensato di utilizzare la strategia bottom-up: ciò ci permette di garantire la presenza di fondamenta solide alla base del sistema ma richiede di mettere in campo test driver per simulare le componenti dei layer più in alto che non sono stati ancora integrati.

6.3 Testing di sistema

La verifica sulle funzionalità del sistema avviene testando i possibili input degli utenti. La riduzione dei casi di test è attuata tramite l'adozione del category partition.

Il testing di sistema concluderà la fase di test del prodotto ed il primo ciclo di sviluppo. Per questa tipologia di test, ci affidiamo all'utilizzo di un software ausiliario come Katalon Studio al fine di osservare il comportamento del sistema in presenza di combinazioni di input utente non ammesse.

6.4 Testing di usablità

Dopo aver realizzato la prima versione dei prototipi abbiamo deciso di proporre ad utenti reali una simulazione del sistema realizzata attraverso la traduzione del flusso dell'interazione tracciato con i mock-ups finali in interazioni simulate tramite animazioni.

Durante la fase di valutazione dell'usabilità ci soffermiamo, oltre che sugli aspetti funzionali del sistema, anche sul grado di intuizione delle interfacce utente, sulle azioni che si devono compiere per utilizzare una determinata funzionalità, sull'esperienza che oggetti ed icone dell'interfaccia permettono di vivere all'utente ed il senso che vi suggeriscono.

Abbiamo adottato la tecnica "task analysis" che consiste nell'assegnare agli utenti un insieme di compiti, di vario livello di difficoltà, per ricavare il grado di efficacia, efficienza e soddisfazione di una funzionalità offerta da sistema attraverso la misurazione di alcune variabili con cui un dato compito viene portato a termine:

- numero di errori
- tempo di esecuzione del task

7. Sospensione e ripresa

Tenuto conto delle risorse necessarie impiegate durante la fase di testing, abbiamo stabilito dei criteri in base ai quali le attività di test saranno sospese o riprese.

7.1 Criteri di sospensione

Il test è sospeso se almeno il 10% dei casi di test riportano errori: in queste condizioni, il team deve provvedere a correggere i fault prima di procedere con l'implementazione o il testing di nuove funzionalità.

7.2 Criteri di ripresa

Abbiamo previsto delle modifiche future al sistema dopo il rilascio. Pertanto sarà necessario, dopo aver introdotto cambiamenti, testare le nuove componenti: se esse introducono dei fault che impattano sulle componenti già esistenti, allora verranno testate nuovamente anche quest'ultime.

I test case, quindi, verranno ancora una volta somministrati al sistema per assicurarsi di aver risolto i problemi scorti precedentemente.

In questo senso evidenziamo la nostra volontà di effettuare test di regressione ad ogni modifica apportata al sistema, pertanto facciamo affidamento su un servizio che ci permette di lavorare in un ambiente di Continous Integration.

7.3 Criteri di terminazione

Il test si considera terminato quando la totalità dei casi di test somministrati al sistema riporta esito negativo. Come da indicazione del top management, la suddetta condizione sussiste solo se il 75% dei branch sviluppati viene ricoperto in questa fase.

8. Materiale per il testing

L'esecuzione dei test necessita di un server correttamente configurato su cui siano installati Java e MySQL. La configurazione deve avvenire come da manuale d'installazione.

Il testing è condotto utilizzando alcuni dei framework più famosi ed efficaci in ambienti Java: JUnit, Mockito e Hamcrest. Ad essi viene affiancata tutta la suite di test relativa a Spring, framework adottato per la realizzazione della webapp.

L'utilizzo di JUnit e Hamcrest riguarda sia il testing di unità che quello di integrazione, mentre Mockito utilizzato solo nel primo caso per "mascherare" le dipendenze.

Come detto al punto precedente, i test sono eseguiti ad ogni modifica apportata al sistema, in un ambiente di Continous Integration: ciò è possibile grazie all'utilizzo di Travis CI e Maven (per quest'ultimo fondamentali sono i plugin SureFire e FailSafe).

9. Test cases

Per ogni sottosistema mostriamo le funzionalità che andremo a testare. Inoltre, associamo ad ogni funzionalità una tabella in cui, per ogni parametro, vengono definite le relative catogorie, insieme alle possibili scelte.

Nella prima riga troviamo Parametro che sta ad indicare il valore di input e formato che viene indicato solitamente attraverso l'espressione regolare per indicare il pattern che il parametro deve seguire.
Nella prima colonna, invece, riportiamo le categorie selezionate, al loro fianco le scelte.

9.1 Studenti

Per il sottosistema studenti abbiamo previsto di testarne la ***funzionalità di iscrizione alla piattaforma***.
Tale funzionalità prevede la possibilità da parte degli studenti di compilare un modulo in cui risulta necessario l'inserimento dei dati personali quali:

- Nome: stringa con un numero di caratteri compreso tra i 2 ed i 255
- Cognome: stringa con un numero di caratteri compreso tra i 2 ed i 255
- Matricola: una stringa composta da 10 numeri
- Data di nascita
    - Giorno di nascita: intero compreso tra 1 e 31
    - Mese di nascita: intero compreso tra 1 e 12
    - Anno di nascita: intero compreso tra l'anno corrente -17 e l'anno corrente -130
- Indirizzo (via, civico, città, CAP): stringa di almeno 2 caratteri
- Sesso: stringa composta da un solo carattere ("M" o "F")

- Contatti
    - e-mail: stringa di lunghezza compresa tra i 2 e i 256 caratteri; nel mezzo contiene @.
    - telefono: stringa contenente 10 o 11 numeri
- Credenziali di accesso
    - username: stringa alfanumerica di lunghezza compresa tra 6 e 24
    - password: stringa di lunghezza compresa tra i 6 ed i 24 caratteri che non deve contenere spazi
    - conferma password: stringa che deve coincidere con quella del campo password

Parametro	Nome
LN – Lunghezza	1. < 2 or > 255 [error] 2. ≥ 2 and ≤ 255 [property LN_OK]

Parametro	Cognome
LC – Lunghezza	1. < 2 or > 255 [error] 2. ≥ 2 and ≤ 255 [property LC_OK]

Parametro	Matricola
Formato	[0-9]{10}
LM – Lunghezza	1. ≠ 10 [error] 2. = 10 [property LM_OK]
FM – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LM_OK] [property FM_OK]

Parametro	Data di nascita
DDN – Valore	1. Non valida or dista meno di 17 anni dalla data corrente or dista più di 130 anni dalla data corrente [error] 2. Valida and dista più di 17 anni dalla data corrente and dista meno di 130 anni dalla data corrente [property DDN_OK]

Parametro	Indirizzo
LI – Lunghezza	1. < 2 or > 255 [error] 2. ≥ 2 and ≤ 255 [property LI_OK]

Parametro	Sesso
S – Valore	1. ≠ “M” and ≠ “F” [error] 2. = “M” or = “F” [property S_OK]

Parametro	E-Mail
Formato	(?:[a-z0-9!#$%&'*+/=?^_`{\|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])*")@(?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?|\[(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])
LE – Lunghezza	1. < 2 or > 256 [error] 2. ≥ 2 and ≤ 256 [property LC_OK]
FE – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LE_OK] [property FE_OK]

Parametro	Telefono
Formato	[0-9]{10,11}
LT – Lunghezza	1. < 10 or > 11 [error] 2. ≥ 10 and ≤ 11 [property LT_OK]
FT – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LT_OK] [property FT_OK]

Parametro	Username
Formato	[a-zA-Z0-9]{6,24}
LU – Lunghezza	1. < 6 or > 24 [error] 2. ≥ 6 and ≤ 24 [property LU_OK]
FU – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LU_OK] [property FU_OK]
EU – Esistenza	1. Esiste nel sistema [if LU_OK] [if FU_OK] [error] 2. Non esiste nel database [property EU_OK]

Parametro	Password
Formato	[\S]{6,24}
LP – Lunghezza	1. < 6 or > 24 [error] 2. ≥ 6 and ≤ 24 [property LP_OK]
FP – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LP_OK] [property FP_OK]

Parametro	Conferma password
UCP – Uguaglianza	1. ≠ password [error] 2. = password [property UCP_OK]

Codice	Combinazione	Esito
TC_GU_2:1	LN1	✗
TC_GU_2:2	LN2.LC1	✗
TC_GU_2:3	LN2.LC2.LM1	✗
TC_GU_2:4	LN2.LC2.LM2.FM1	✗
TC_GU_2:5	LN2.LC2.LM2.FM2.DDN1	✗
TC_GU_2:6	LN2.LC2.LM2.FM2.DDN2.LI1	✗
TC_GU_2:7	LN2.LC2.LM2.FM2.DDN2.LI2.S1	✗
TC_GU_2:8	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE1	✗
TC_GU_2:9	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE1	✗
TC_GU_2:10	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE2.LT1	✗
TC_GU_2:11	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE2.LT2.FT1	✗
TC_GU_2:12	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE2.LT2.FT2.LU1	✗
TC_GU_2:13	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE2.LT2.FT2.LU2.FU1	✗
TC_GU_2:14	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE2.LT2.FT2.LU2.FU2.EU1	✗
TC_GU_2:15	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP1	✗
TC_GU_2:16	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP1	✗
TC_GU_2:17	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP1	✗
TC_GU_2:18	LN2.LC2.LM2.FM2.DDN2.LI2.S2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2	✓

9.2 Domande di Tirocinio

Per il sottosistema Domande di Tirocinio abbiamo previsto di testarne la ***funzionalità di invio di una domanda di tirocinio.***

Tale funzionalità prevede la possibilità da parte degli studenti di compilare un modulo in cui risulta necessario l'inserimento dei seguenti dati:

- Data di inizio tirocinio:
  - Giorno di inizio tirocinio: intero compreso tra 1 e 31
  - Mese di inizio tirocinio: intero compreso tra 1 e 12
  - Anno di inizio tirocinio: intero maggiore o uguale dell'anno corrente
- Data di fine tirocinio:
  - Giorno di fine tirocinio: intero compreso tra 1 e 31
  - Mese di fine tirocinio: intero compreso tra 1 e 12
  - Anno di fine tirocinio: intero maggiore o uguale dell'anno corrente
- Breve commento alla richiesta: una stringa composta da almeno 2 caratteri
- Numero di CFU da associare alla domanda: intero compreso tra 1 e 18

Parametro	Data d'inizio
DDI – Valore	1. Non valida or precede la data corrente [error] 2. Valida and segue la data corrente [property DDI_OK]

Parametro	Data di fine
DDF – Valore	1. Non valida or precede la data d'inizio [error] 2. Valida and segue la data d'inizio [property DDF_OK]

Parametro	Commento
LCOM – Lunghezza	1. < 2 [error] 2. ≥ 2 [property LCOM_OK]

Parametro	CFU
NCFU – Lunghezza	1. < 1 or > 18 [error] 2. ≥ 1 and ≤ 18 [property NCFU_OK]

Codice	Combinazione	Esito
TC_GDT_1:1	DD1	✗
TC_GDT_1:2	DD2.DDF1	✗
TC_GDT_1:3	DD2.DDF2.LCOM1	✗
TC_GDT_1:4	DD2.DDF2.LCOM2.NCFU1	✗
TC_GDT_1:5	DD2.DDF2.LCOM2.NCFU2	✓

9.3 Convenzioni

Per il sottosistema Convenzioni abbiamo previsto di testarne la ***funzionalità di invio di una richiesta di convenzionamento***. Ricordiamo che l'invio di una richiesta di convenzione coincide con la richiesta di iscrizione al sito da parte di un'azienda interessata.

Tale funzionalità prevede la possibilità da parte di un delegato aziendale di compilare un modulo in cui risulta necessario l'inserimento dei dati personali quali:

- Nome e cognome del delegato: entrambe stringhe di lunghezza compresa tra i 2 ed i 255 caratteri
- Contatti
    - e-mail: stringa di lunghezza compresa tra i 2 e i 256 caratteri; nel mezzo contiene @
    - telefono: stringa contenente 10 o 11 numeri
- Credenziali di accesso
    - username: stringa alfanumerica di lunghezza compresa tra 6 e 24 che non può contenere spazi
    - password: stringa di lunghezza compresa tra 6 e 24 che non può contenere spazi

    - conferma password: stringa di lunghezza compresa tra 6 e 24 che deve essere coincidere con la password
- Sesso del delegato: stringa composta dal singolo carattere "M" oppure "F"
- Nome dell'azienda: una stringa di lunghezza compresa tra 2 ed i 255 caratteri
- Sede dell'azienda (via, civico - CAP, città (provincia)): stringa di lunghezza compresa tra i 2 ed i 255 caratteri
- Partita IVA dell'azienda: stringa numerica di 11 cifre
- Id dell'azienda: stringa alfanumerica di lunghezza compresa tra i 6 e i 24 caratteri che non può contenere spazi

Accessibilità dell'azienda: booleano che è vero se l'azienda non ha barriere architettoniche, falso altrimenti

Parametro	Nome
LN – Lunghezza	1. < 2 or > 255 [error] 2. ≥ 2 and ≤ 255 [property LN_OK]

Parametro	Cognome
LC – Lunghezza	1. < 2 or > 255 [error] 2. ≥ 2 and ≤ 255 [property LC_OK]

Parametro	E-Mail
Formato	(?:[a-z0-9!#$%&'*+/=?^_`{\|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])*")@(?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?|\[(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])
LE – Lunghezza	1. < 2 or > 256 [error] 2. ≥ 2 and ≤ 256 [property LC_OK]
FE – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LE_OK] [property FE_OK]

Parametro	Telefono
Formato	[0-9]{10,11}
LT – Lunghezza	1. < 10 or > 11 [error] 2. ≥ 10 and ≤ 11 [property LT_OK]
FT – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LT_OK] [property FT_OK]

Parametro	Username
Formato	[a-zA-Z0-9]{6,24}
LU – Lunghezza	1. < 6 or > 24 [error] 2. ≥ 6 and ≤ 24 [property LU_OK]
FU – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LU_OK] [property FU_OK]
EU – Esistenza	1. Esiste nel sistema [if LU_OK] [if FU_OK] [error] 2. Non esiste nel database [property EU_OK]

Parametro	Password
Formato	[\S]{6,24}
LP – Lunghezza	1. < 6 or > 24 [error] 2. ≥ 6 and ≤ 24 [property LP_OK]
FP – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LP_OK] [property FP_OK]

Parametro	Conferma password
UCP – Uguaglianza	1. ≠ password [error] 2. = password [property UCP_OK]

Parametro	Sesso
S – Valore	1. ≠ “M” and ≠ “F” [error] 2. = “M” or = “F” [property S_OK]

Parametro	Nome dell’azienda
LNA – Lunghezza	1. < 2 or > 255 [error] 2. ≥ 2 and ≤ 255 [property LNA_OK]

Parametro	Indirizzo dell’azienda
LIA – Lunghezza	1. < 2 or > 255 [error] 2. ≥ 2 and ≤ 255 [property LIA_OK]

Parametro	Partita IVA
Formato	[0-9]{11}

LPIVA – Lunghezza	1. ≠ 11 [error] 2. = 11 [property LPIVA_OK]
FPIVA – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LPIVA_OK] [property FPIVA_OK]

Parametro	Id dell'azienda
Formato	[a-zA-Z0-9]{6,24}
LID – Lunghezza	1. < 6 or > 24 [error] 2. ≥ 6 and ≤ 24 [property LID_OK]
FID – Formato	1. Non rispetta il formato [error] 2. Rispetta il formato [if LID_OK] [property FID_OK]
EID – Esistenza	1. Esiste nel sistema [if LID_OK] [if FID_OK] [error] 2. Non esiste nel database [property EID_OK]

Codice	Combinazione	Esito
TC_GC_1:1	LN1	X
TC_GC_1:2	LN2.LC1	X
TC_GC_1:3	LN2.LC2.LE1	X
TC_GC_1:4	LN2.LC2.LE2.FE1	X
TC_GC_1:5	LN2.LC2.LE2.FE2.LT1	X
TC_GC_1:6	LN2.LC2.LE2.FE2.LT2.FT1	X
TC_GC_1:7	LN2.LC2.LE2.FE2.LT2.FT2.LU1	X
TC_GC_1:8	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU1	X
TC_GC_1:9	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU1	X
TC_GC_1:10	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP1	X
TC_GC_1:11	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP1	
TC_GC_1:12	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP1	X
TC_GC_1:13	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2.S1	X
TC_GC_1:14	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2.S2.LNA1	X
TC_GC_1:15	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2.S2.LNA2.LIA1	X
TC_GC_1:16	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2.S2.LNA2.LIA2.LPIVA1	X
TC_GC_1:17	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2.S2.LNA2.LIA2.LPIVA2.FPIVA1	X
TC_GC_1:18	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2.S2.LNA2.LIA2.LPIVA2.FPIVA2.LID1	X
TC_GC_1:19	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2.S2.LNA2.LIA2.LPIVA2.FPIVA2.LID2.FID1	X
TC_GC_1:20	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2.S2.LNA2.LIA2.LPIVA2.FPIVA2.LID2.FID2.EID1	X

TC_GC_1:21	LN2.LC2.LE2.FE2.LT2.FT2.LU2.FU2.EU2.LP2.FP2.UCP2.S2.LNA2.LIA2.LPIVA2.FPIVA2.LID2.FID2.EID2	✓

9.4 Progetti Formativi

Per il sottosistema Progetti Formativi abbiamo previsto di testarne la ***funzionalità per aggiungere un nuovo progetto formativo.***

Tale funzionalità prevede la possibilità da parte di un delegato aziendale di compilare un modulo in cui risulta necessario l'inserimento di alcuni dati quali:

- Nome del progetto formativo: stringa composta da 2 a 255 caratteri
- Descrizione: stringa di almeno 2 caratteri

Parametro	Nome progetto
LNP – Lunghezza	1. < 2 or > 255 [error] 2. ≥ 2 and ≤ 255 [property LNO_OK]

Parametro	Descrizione
LD – Lunghezza	1. < 2 [error] 2. ≥ 2 [property LD_OK]

Codice	Combinazione	Esito
TC_GPF_3:1	LNP1	✗
TC_GPF_3:2	LNP2.LD1	✗
TC_GPF_3:2	LNP2.LD2	✓

10. Riferimenti ad altri documenti di test

Le combinazioni di input che verranno somministrate al sistema sono definite nel documento di Test Case Specification, mentre sarà nel Test Execution Report che indicheremo i risultati del testing funzionale condotto tramite Katalon Studio.

Eventuali errori rilevati verranno riportati nel Test Incident Report, mentre il riassunto dei riscontri ottenuti in questa fase verrà proposto tramite il documento di Test Summary Report.