



Renato Milano 0512103100

Ciro Valerio Cerchia 0512103698

Francesco Buonomo 0512103794

T.P

Data	Versione	Descrizione	Autori
13/06/19	0.1	Stesura punti 1 e 2	Buonomo, Cerchia, Milano
14/06/19	0.2	Stesura punto 3	Buonomo, Cerchia, Milano
15/06/19	0.3	Stesura punto 4	Buonomo, Cerchia, Milano
16/06/19	0.4	Stesura punto 5	Buonomo, Cerchia, Milano
17/06/19	0.5	Stesura punto 6	Buonomo, Cerchia, Milano
18/06/19	0.6	Stesura punto 7 e 8	Buonomo, Cerchia, Milano
20/06/19	1.0	Stesura punto 9	Buonomo, Cerchia, Milano

1. Introduzione

Il Test Plan ha lo scopo di analizzare e gestire le attività di testing riguardanti il progetto CollegaMenti, per verificare il corretto funzionamento della piattaforma. In questo documento vengono riportate le strategie di testing da adottare, le funzionalità da testare e gli strumenti utilizzati per rilevare errori all'interno del codice prodotto, in maniera pianificata per evitare che essi si presentino ai clienti finali durante l'utilizzo del sistema. I risultati di questi test saranno utilizzati per capire dove bisognerà intervenire per correggere gli eventuali errori presenti all'interno del codice. Le attività di testing sono state pianificate per le seguenti gestioni:

- Gestione Registrazione
- Gestione Autenticazione
- Gestione Account
- Gestione Comunicazione
- Gestione Forum

2. Relazione con gli altri documenti

Per la corretta individuazione dei test case abbiamo fatto riferimento ai documenti prodotti precedentemente. La fase di testing, difatti, è strettamente legata alle fasi ad essa precedenti; ogni documento sarà un punto di partenza indispensabile per poter effettuare un testing corretto ed adeguato e per verificare che il sistema proposto sia fedele alle richieste del cliente.

2.1 Relazioni con il Requirements Analysis Document (RAD)

La relazione tra test plan e RAD riguarda in particolare i requisiti funzionali e non funzionali del sistema poiché i test che saranno eseguiti su ogni funzionalità terranno conto delle specifiche espresse nel RAD.

2.2 Relazioni con il System Design Document (SDD)

Nel System Design Document abbiamo suddiviso il nostro sistema in sottosistemi e l'architettura in tre livelli: Presentation Layer, Application Layer e Storage Layer. Il test dei vari componenti deve rimanere fedele a queste suddivisioni il più possibile.

2.3 Relazioni con l'Object Design Document (ODD)

Il test d'integrazione farà riferimento alle interfacce delle classi e i package definiti nell'ODD.

3. Panoramica del sistema

Come stabilito nel System Design Document la struttura del nostro sistema è divisa secondo una architettura "Three Layers" cioè a tre livelli: Presentation Layer, Application Layer, Storage Layer. In questo caso il livello più alto interagisce con il livello applicativo che a sua volta si occuperà di eseguire le operazioni nel database di Collega-Menti, cercando di garantire il più possibile basso accoppiamento e alta coesione tra le varie classi. Il sistema inoltre è stato suddiviso in sottosistemi più piccoli, in particolare è stato diviso per gestioni, definite al paragrafo 1 di questo documento.

4. Funzionalità da testare e non testare

Nella fase di testing saranno testate le funzionalità principali per ogni gestione. Sono elencate di seguito le funzionalità scelte:

- Gestione Registrazione:
 - Registrazione Account Coach
 - Registrazione Account Studente
- Gestione Autenticazione:
 - Login
- Gestione Account:
 - Modifica Password
 - Richiesta Coach
 - Sottomissione Abbonamento
- Gestione Forum:
 - Aprire Thread
 - Partecipare ad un Thread aperto
- Gestione Comunicazione
 - Rispondere ai Messaggi:

5. Pass/fail criteria

Lo scopo del testing è quello di dimostrare la presenza di faults (errori) all'interno del sistema. Le attività di testing, infatti, saranno mirate all'identificazione di questi faults e ad un successivo intervento per eliminarne la presenza.

Un test avrà successo se, dato un input al sistema, l'output osservato sarà diverso dall'output atteso dall'oracolo, quindi è stata identificata una failure. In tal caso questa verrà analizzata e si procederà alla correzione.

Al contrario, un test fallirà quando l'output osservato sarà uguale a quello presente nell'oracolo. Per oracolo si intende il risultato atteso, in base ai requisiti, dall'esecuzione di un caso di test.

La fase di testing non sarà considerata valida se una delle seguenti condizioni non verrà rispettata:

- Effettuare test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti;
- Raggiungere un branch coverage non inferiore al 75%.

6. Approccio

Per il sistema Collegamenti il testing si compone di tre fasi. Nella prima fase verranno eseguiti i test di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità andando a constatare il corretto funzionamento di tutte le singole unità di codice. Questa fase verrà effettuata al completamento di ogni unità realizzata per poter individuare tempestivamente gli errori presenti nel codice.

Nella seconda fase verrà fatto il testing di integrazione in cui si andrà a testare l'integrazione dei vari sottosistemi. Questa fase verrà effettuata automaticamente ad ogni commit dal tool di supporto scelto per questa tipologia di testing. Nella sezione dedicata al testing di integrazione verrà presentato il tool scelto.

Infine verrà eseguito il testing di sistema, che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti. Quest'ultimo servirà soprattutto a verificare che il sistema soddisfi le richieste del committente.

Le attività di testing saranno eseguite successivamente all'implementazione del sistema.

6.1 Testing di unità

Per effettuare il testing di ogni singola componente del sistema, verrà utilizzata la tecnica "Black-Box testing" attraverso il framework JUnit. In questa fase saranno analizzate le funzionalità dell'applicazione ed il comportamento delle singole componenti senza tener conto della loro struttura interna. Al fine di minimizzare il numero di test case, gli input saranno partizionati in classi di equivalenza. Tali classi serviranno a definire le categorie per poter utilizzare, come testing funzionale, il Category Partition quindi per ogni classe di equivalenza verranno individuati dei vincoli a cui saranno associate delle scelte specifiche. I risultati del testing verranno analizzati e usati per correggere gli errori che causano il fallimento del sistema. Se si verifica un errore con dei risultati inattesi si interviene in

maniera tempestiva sulla componente in modo da renderla correttamente funzionante e procedere con le fasi di testing successive.

Per effettuare testing in questa fase le componenti vengono isolate tramite l'utilizzo di Test Stub (implementazione completa o parziale di una componente dalla quale dipende un'altra componente sotto testing) e Test Driver (implementazione completa o parziale di una componente che fa uso di un'altra componente sotto testing) che servono a sostituire le parti mancanti del sistema.

Dal momento che abbiamo scelto di utilizzare una architettura SOA (Service Oriented Architecture), per unità intendiamo un singolo servizio offerto dalle classi che compongono il back-end.

6.2 Testing di integrazione

In seguito al test di unità di ogni componente, si procederà con il test di integrazione utilizzando Travis, un servizio di continuous integration. Il tool utilizza un approccio Big Bang per l'integrazione delle componenti.

6.3 Testing di sistema

Lo scopo principale di questa fase di testing è quello di dimostrare che l'intero sistema funzioni correttamente e che soddisfi effettivamente i requisiti funzionali e non funzionali descritti nel documento di analisi dei requisiti (RAD). È da considerare l'attività più critica, in quanto può risultare molto complesso andare alla ricerca di eventuali errori, essendo impegnati tutti i sottosistemi. Per questa fase sarà utilizzato il tool Selenium, che simula l'interazione dell'utente con il sistema.

7. Sospensione e Ripristino

La fase di testing occupa una parte importante nell'intera attività inerente il progetto, essa è anche molto delicata e può essere causa di imprevisti e slittamenti dei tempi a causa di errori e malfunzionamento del sistema.

7.1 Criteri di sospensione

Il testing sarà sospeso quando verrà raggiunta una branch coverage di almeno il 75%. Nel caso in cui la schedulazione del progetto fosse rispettata si cercherà di testare tutte le classi selezionate per ogni possibile input e il risultato dovrà essere quello atteso.

Nel momento in cui il testing rivelerà un errore si dovrà passare alla fase di correzione che dovrà essere eseguita da una ripetizione dell'intero processo di testing per rilevare la presenza di eventuali errori introdotti dalle correzioni stesse.

7.2 Criteri di ripresa

La ripresa del test avviene soltanto quando tutti i problemi relativi alla sospensione dello stesso sono stati risolti. L'attività di testing riprenderà a partire dal test case che ha causato la sospensione.

8. Strumenti per il testing

Il sistema verrà testato su un computer, configurato dal team stesso. Non sono richieste particolari specifiche hardware, se non quelle adatte a far funzionare il DBMS e il Server Tomcat.

9. Test Cases

9.1 Gestione Registrazioni

9.1.1 Registrazione Account Coach

Parametro: nome Formato: ^[A-Za-z ']+\$	
Lunghezza[LN]	1. <1 OR >15 [error] 2. >=1 AND <=15 [property LN_OK]
Formato [FN]	1. Non rispetta il formato [if FN_OK] 2. Rispetta il formato [if FN_OK] [property FN_OK]

Parametro: cognome

Formato: `^[A-Za-z ']+$`

Lunghezza[LC]	<ol style="list-style-type: none"> <1 OR >15 [error] >=1 AND <=15 [property LC_OK]
Formato [FC]	<ol style="list-style-type: none"> Non rispetta il formato [if FC_OK] Rispetta il formato [if FC_OK] [property FC_OK]

Parametro: titolo di studio

Formato: `^[A-Za-z ']+$`

Lunghezza[LTS]	<ol style="list-style-type: none"> <1 OR >25 [error] >=1 AND <=25 [property LTS_OK]
Formato [FTS]	<ol style="list-style-type: none"> Non rispetta il formato [if FTS_OK] Rispetta il formato [if FTS_OK] [property FTS_OK]

Parametro: email

Formato: `[A-Za-z0-9.]@[a-z.]`

Lunghezza[LE]	<ol style="list-style-type: none"> <1 OR >25 [error] >=1 AND <=25 [property LE_OK]
Formato [FE]	<ol style="list-style-type: none"> Non rispetta il formato [if FE_OK] Rispetta il formato [if FE_OK] [property FE_OK]

Parametro: password

Formato: `^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])([a-zA-Z0-9_.,\-\+*!#@?]{8,16})$`

Lunghezza[LP]	<ol style="list-style-type: none"> <1 OR >10 [error] >=1 AND <=10 [property LP_OK]
Formato [FP]	<ol style="list-style-type: none"> Non rispetta il formato [if LP_OK] Rispetta il formato [if FP_OK] [property FP_OK]

Codice	Combinazione	Esito
--------	--------------	-------

TC_1.1	LN2,FN2,LC2,FC2,LTS2,FTS2,LE2,FE2,LP2,FP2	Errato
TC_1.2	LN2,FN2,LC2,FC2,LTS2,FTS2,LE2,FE2,LP2,FP2	Errato
TC_1.3	LN2, FN2, LC1,FC2,LTS2,FTS2,LE2,FE2,LP2,FP2	Errato
TC_1.4	LN2, FN2, LC2, FC1,LTS2,FTS2,LE2,FE2,LP2,FP2	Errato
TC_1.5	LN2, FN2, LC2, FC2, LTS1,FTS2,LE2,FE2,LP2,FP2	Errato
TC_1.6	LN2, FN2, LC2, FC2, LTS2,FTS1,LE2,FE2,LP2,FP2	Errato
TC_1.7	LN2, FN2, LC2, FC2, LTS2,FTS2,LE1,FE2,LP2,FP2	Errato
TC_1.8	LN2, FN2, LC2, FC2, LTS2,FTS2,LE2,FE1,LP2,FP2	Errato
TC_1.9	LN2, FN2, LC2, FC2, LTS2,FTS1,LE2,FE2,LP1,FP2	Errato
TC_1.10	LN2, FN2, LC2, FC2, LTS2,FTS1,LE2,FE2,LP2,FP1	Errato
TC_1.12	LN2, FN2, LC2, FC2, LTS2,FTS2,LE2,FE2,LP2,FP2	Corretto

9.1.2 Registrazione Account Studente

Parametro: nome	
Formato: ^[A-Za-z ']+\$	
Lunghezza[LN]	<ol style="list-style-type: none"> <1 OR >15 [error] >=1 AND <=15 [property LN_OK]
Formato [FN]	<ol style="list-style-type: none"> Non rispetta il formato [if FN_OK] Rispetta il formato [if FN_OK] [property FN_OK]

Parametro: cognome	
Formato: ^[A-Za-z ']+\$	
Lunghezza[LC]	<ol style="list-style-type: none"> <1 OR >15 [error] >=1 AND <=15 [property LC_OK]
Formato [FC]	<ol style="list-style-type: none"> Non rispetta il formato [if FC_OK] Rispetta il formato [if FC_OK] [property FC_OK]

Parametro: email

Formato: [A-Za-z0-9.]@[a-z.]

Lunghezza[LE]	<ol style="list-style-type: none"> 1. <1 OR >25 [error] 2. >=1 AND <=25 [property LE_OK]
Formato [FE]	<ol style="list-style-type: none"> 1. Non rispetta il formato [if FE_OK] 2. Rispetta il formato [if FE_OK] [property FE_OK]

Parametro: password

Formato: ^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])([a-zA-Z0-9_,-\+*!#@?]{8,16})\$

Lunghezza[LP]	<ol style="list-style-type: none"> 1. <1 OR >16 [error] 2. >=1 AND <=16 [property LP_OK]
Formato [FP]	<ol style="list-style-type: none"> 1. Non rispetta il formato [if FP_OK] 2. Rispetta il formato [if FP_OK] [property FP_OK]

Codice	Combinazione	Esito
TC1.12	LN1, FN2, LC2,FC2,LE2,FE2,LP2,FP2	Errato
TC_1.13	LN2, FN1, LC2,FC2,LE2,FE2,LP2,FP2	Errato
TC_1.14	LN2, FN2, LC1,FC2,LE2,FE2,LP2,FP2	Errato
TC_1.15	LN2, FN2, LC2,FC1,LE2,FE2,LP2,FP2	Errato
TC_1.16	LN2, FN2, LC2,FC2,LE1,FE2,LP2,FP2	Errato
TC_1.17	LN2, FN2, LC2,FC2,LE2,FE1,LP2,FP2	Errato
TC_1.18	LN2, FN2, LC2,FC2,LE1,FE2,LP1,FP2	Errato
TC_1.19	LN2, FN2, LC2,FC2,LE2,FE1,LP2,FP1	Errato
TC_1.20	LN2, FN2, LC2,FC2,LE2,FE2,LP2,FP2	Corretto

9.2 Gestione Autenticazione

9.2.1 Login

Parametro: email

Formato: [A-Za-z0-9.]@[a-z.]

Lunghezza[LE]]	<ol style="list-style-type: none"> <1 OR >25 [error] >=1 AND <=25 [property LE_OK]
Formato [FE]	<ol style="list-style-type: none"> Non rispetta il formato [if FE_OK] Rispetta il formato [if FE_OK] [property FE_OK]

Parametro: password

Formato: ^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])([a-zA-Z0-9_.,\-\+*!#@?]{8,16})\$

Lunghezza[LP]]	<ol style="list-style-type: none"> <1 OR >16 [error] >=1 AND <=16 [property LP_OK]
Formato [FP]	<ol style="list-style-type: none"> Non rispetta il formato [if FP_OK] Rispetta il formato [if FP_OK] [property FP_OK]

Codice	Combinazione	Esito
TC_2.1	LE2, FE1, LP2,FP2	Errato
TC_2.2	LE2, FE2, LP2,FP2	Errato
TC_2.3	LE2, FE2, LP2,FP1	Errato
TC_2.4	LE1, FE2, LP2,FP2	Errato
TC_2.5	LE2, FE2, LP2, FP2	Corretto

9.3 Gestione Account

9.3.2 Modifica Password

Parametro: password

Formato: ^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])([a-zA-Z0-9_.,\-\+*!#@?]{8,16})\$

Lunghezza[LP]	<ol style="list-style-type: none"> <8 OR >16 [error]
---------------	---

	2. ≥ 8 AND ≤ 16 [property LP_OK]
Formato [FP]	1. Non rispetta il formato [if FP_OK] 2. Rispetta il formato [if FP_OK] [property FP_OK]

Codice	Combinazione	Esito
TC_5.1	LP1,FP2	Errato
TC_5.2	LP2,FP1	Errato
TC_5.3	LP2,FP2	Corretto

9.3.5 Richiesta Coach

Parametro: nome Formato: $^{[A-Za-z ']+}$	
Lunghezza[LN]	1. <1 OR >15 [error] 2. ≥ 1 AND ≤ 15 [property LN_OK]
Formato [FN]	1. Non rispetta il formato [if FN_OK] 2. Rispetta il formato [if FN_OK] [property FN_OK]

Parametro: cognome Formato: $^{[A-Za-z ']+}$	
Lunghezza[LC]	1. <1 OR >15 [error] 2. ≥ 1 AND ≤ 15 [property LC_OK]
Formato [FC]	1. Non rispetta il formato [if FC_OK] 2. Rispetta il formato [if FC_OK] [property FC_OK]

Parametro: argomento

Formato: ^[A-Za-z ']+\$

Lunghezza[LA]	<ol style="list-style-type: none"> <1 OR >15 [error] >=1 AND <=15 [property LA_OK]
Formato [FA]	<ol style="list-style-type: none"> Non rispetta il formato [if FA_OK] Rispetta il formato [if FA_OK] [property FA_OK]

Codice	Combinazione	Esito
TC_.5.3	LN2, FN1, LC2,FC2,LA2	Errato
TC_5.4	LN1, FN2, LC2,FC2,LA2	Errato
TC_5.5	LN2, FN2, LC2,FC1,LA2	Errato
TC_5.6	LN2, FN2, LC2,FC2,LA1	Errato
TC_5.7	LN2, FN2, LC1, FC2,LA2	Errato
TC_5.8	LN2, FN2, LC2, FC2,LA2	Corretto

9.3.8 Sottomissione Abbonamento

Parametro: nome

Formato: ^[A-Za-z ']+\$

Lunghezza[LN]	<ol style="list-style-type: none"> <1 OR >15 [error] >=1 AND <=15 [property LN_OK]
Formato [FN]	<ol style="list-style-type: none"> Non rispetta il formato [if FN_OK] Rispetta il formato [if FN_OK] [property FN_OK]

Parametro: cognome

Formato: ^[A-Za-z ']+\$

Lunghezza[LC]	<ol style="list-style-type: none"> <1 OR >15 [error] >=1 AND <=15 [property LC_OK]
---------------	--

Formato [FC]	<ol style="list-style-type: none"> 1. Non rispetta il formato [if FC_OK] 2. Rispetta il formato [if FC_OK] [property FC_OK]
--------------	---

Parametro: intestatario carta

Formato: `^[A-Za-z ']+$^`

Lunghezza[LIC]	<ol style="list-style-type: none"> 1. <1 OR >15 [error] 2. >=1 AND <=15 [property LIC_OK]
Formato [FIC]	<ol style="list-style-type: none"> 1. Non rispetta il formato [if FIC_OK] 2. Rispetta il formato [if FIC_OK] [property FIC_OK]

Parametro: numero carta

Formato: `^[0-9]{16}$`

Lunghezza[LNC]	<ol style="list-style-type: none"> 1. <1 OR >15 [error] 2. >=1 AND <=15 [property LNC_OK]
Formato [FNC]	<ol style="list-style-type: none"> 1. Non rispetta il formato [if FNC_OK] 2. Rispetta il formato [if FNC_OK] [property FNC_OK]

Codice	Combinazione	Esito
TC_3.3	LN1, FN2, LC2,FC2,LIC2,FIC2,LNC2,FNC2	Errato
TC_3.4	LN2, FN1, LC2,FC2,LIC2,FIC2,LNC2,FNC2	Errato
TC_3.5	LN2, FN1, LC2,FC1,LIC2,FIC2,LNC2,FNC2	Errato
TC_3.6	LN2, FN2, LC1,FC2,LIC2,FIC2,LNC2,FNC2	Errato
TC_3.7	LN2, FN2, LC2,FC2,LIC2,FIC1,LNC2,FNC2	Errato
TC_3.8	LN2, FN2, LC2,FC2,LIC1,FIC2,LNC2,FNC2	Errato
TC_3.9	LN2, FN2, LC2,FC2,LIC2,FIC2,LNC1,FNC2	Errato

TC_3.10	LN2, FN2, LC2,FC2,LIC2,FIC2,LNC2,FNC2	Errato
TC_3.11	LN2, FN2, LC2,FC2,LIC2,FIC2,LNC2,FNC2	Corretto

9.4 Gestione Forum

9.4.1 Aprire Thread

Parametro: titolo

Formato: `^[A-Za-z ']+$`

Lunghezza[LT]	<ol style="list-style-type: none"> <1 OR >20 [error] >=1 AND <=20 [property LT_OK]
---------------	--

Parametro: oggetto

Formato: `^[A-Za-z0-9 _]*[A-Za-z0-9][A-Za-z0-9 _]*{1,200}$`

Lunghezza[LO]	<ol style="list-style-type: none"> <1 OR >200 [error] >=1 AND <=200 [property LO_OK]
---------------	--

Codice	Combinazione	Esito
TC_4.1	LT1,LO2	Errato
TC_4.2	LT2,LO1	Errato
TC_4.3	LT2,LO2	Errato

9.4.2 Partecipare ad un Thread aperto

Parametro: oggetto

Formato: `^[A-Za-z0-9 _]*[A-Za-z0-9][A-Za-z0-9 _]*{1,200}$`

Lunghezza[LO]	<ol style="list-style-type: none"> <1 OR >200 [error] >=1 AND <=200 [property LO_OK]
---------------	--

9.5 Gestione Comunicazione

9.5.1 Rispondere ai messaggi

Parametro: messaggio

Formato: `^[A-Za-z0-9 _]*[A-Za-z0-9][A-Za-z0-9 _]*{1,200}$`

Lunghezza[LM]

1. <1 OR >200 [error]
2. >=1 AND <=200 [property LM_OK]

10. Testing Shedule

il Testing sarà effettuato non in modo incrementale allo sviluppo ma sarà effettuato interamente un volta sviluppato l'intero sistema.