

LOGO PROGETTO



Data: 24/06/2018

Progetto: NowIWant	Versione: 1.0
Documento: test plan document	Data: 24/06/2018

Coordinatore del progetto:

Nome	Matricola
NA	NA

Partecipanti:

Nome	Matricola
Ferrazzano Pompeo Alessio	0512102928
Citro Antonio	0512102922
Giovanni Lembo	0512103252
Robertazzi Gennaro Alessio	0512103792

Scritto da:	FPA, CA, RGA, LG.
--------------------	-------------------

Revision History

Data	Versione	Descrizione	Autore
24/06/2018	0.1	Abbozzo di stesura	RGA
24/06/2018	0.5	Tipi di testing	RGA
25/06/2018	1.0	Revisione errori	FPA, CA, RGA, LG

Indice

1. INTRODUZIONE4

2. RELAZIONE CON ALTRI DOCUMENTI4

3. PANORAMICA DEL SISTEMA.....4

4. FUNZIONALITA’ DA TESTARE.....4

5. PASS/FAIL CRITERIA5

6. APPROCCIO5

 6.1 Unit Testing.....5

 6.2 Integration Testing5

 6.3 System Testing6

7. PIANIFICAZIONE DEL TESTING6

8. TEST CASE SPECIFICATION.....7

 8.1 TC_AUT_17

 1.1 TC_AUT_1.1: Login.....7

 1.2 TC_AUT_1.2: Registrazione8

 8.2 TC_PRO_2: Aggiungi Prodotto10

 8.3 TC_CAT_3: Aggiungi Categoria12

1. INTRODUZIONE

Il documento Test Plan è stato creato per analizzare, gestire lo sviluppo e le attività di testing riguardante *NowIWant* per verificarne il corretto funzionamento.

Verranno provate alcune funzionalità del sistema. I risultati di questo test saranno utilizzati per capire dove bisognerà intervenire per correggere eventuali errori o apportare modifiche per il miglioramento dei vari sottosistemi.

2. RELAZIONE CON GLI ALTRI DOCUMENTI

Per verificarne il corretto funzionamento del sistema, sono stati predisposti dei test case, basati sulla suddivisione in sottosistemi proposta in fase di System Design e raccolta dei requisiti.

3. PANORAMICA DEL SISTEMA

Il sistema NowIWant è una piattaforma e-commerce per la vendita di prodotti di abbigliamento, che punta sulla facilità di gestione del sistema e nel massimizzare l'usabilità dell'interfaccia da parte dell'utente finale.

4. FUNZIONALITA' DA TESTARE

Nella fase di testing verranno prese in considerazione le funzionalità fondamentali:

- *Login*: sarà testato il corretto funzionamento dell'accesso al sito, per permettere all'utente di sfruttare al massimo la nostra piattaforma.
- *Registrazione*: Sarà testato il corretto funzionamento della registrazione, che permetterà all'utente di accedere alla piattaforma.
- *Aggiungi Prodotto*: Verranno testate le funzionalità sull'aggiunta di un prodotto da parte dell'amministratore nella piattaforma.
- *Aggiungi Categoria*: Sarà testata la funzionalità da parte di un amministratore di aggiungere una nuova categoria (o sottocategoria) all'interno del sito.

5. PASS/FAIL CRITERIA

Il testing avrà esito positivo se l'output osservato risulterà essere diverso dall'output atteso, significa che il testing avrà successo se verrà individuata una failure. In tal caso questa verrà analizzata e si procederà alla correzione. Sarà infine iterata la fase di testing per verificare che la modifica effettuata per correggere il fault non abbia impattato su altri componenti del sistema. Al contrario, il testing fallirà se l'output sarà uguale all'oracolo.

6. APPROCCIO

Per la piattaforma *NowIWant* verrà utilizzato un approccio “*Black Box*” dove i test verranno effettuati senza scendere nel dettaglio del codice ma basandosi sulle specifiche funzionalità da testare. L'approccio si compone di tre fasi:

- ***Unit Testing***: testing che va a controllare i singoli componenti (es. classi, metodi);
- ***Integration Testing***: che va a testare l'integrazione dei vari sottosistemi;
- ***System Testing***: testerà l'intero sistema assemblato.

6.1 *Unit Testing*

Questo testing verrà effettuato sulle classi e sui metodi del sito, saranno ricercate le condizioni di fallimento andando ad evidenziare gli errori. Tutto questo verrà effettuato tramite il framework JUnit.

6.2 *Integration Testing*

Verrà effettuato tramite una strategia bottom-up. Questi test verranno effettuati dal datalayer al business layer, dal business layer al view layer ed infine tra le tre.

6.3 System Testing

Ci sarà un testing sulla correttezza dell'intero sistema. Esso è il testing più critico in quanto può risultare molto complesso andare alla ricerca di eventuali errori. Questo test sarà effettuato utilizzando il metodo del Weak Equivalence Class Testing.

7. PIANIFICAZIONE DEL TESTING

Il testing verrà effettuato utilizzando il metodo del Category Partition: è un tipo di test combinatorio che permette di identificare attributi, valori rilevanti e possibili combinazioni, il tutto permettendo la separazione dell'identificazione dei valori che caratterizzano lo spazio di input, dalla combinazione di valori diversi in casi di test completi; fornisce una stima del numero dei casi di test molto presto.

Sono previste 3 fasi diverse:

- **Decomporre le specifiche in feature testabili indipendentemente** : in questo passo, il test designer deve identificare le feature che devono essere testate in modo separato, identificando i parametri e qualunque altro elemento dell'ambiente di esecuzione da cui dipende (ad esempio un database). Per ciascun parametro e elemento dell'ambiente si identificano le caratteristiche (elementari) del parametro, dette categorie.
- **Identificare Valori Rappresentativi** : questo passo, prevede che il test designer identifichi un'insieme di valori rappresentativi (classe di valori, detta choices) per ciascuna caratteristica di ogni parametro definito nella fase precedente.
- **Generare Specifiche di Casi di Test** : Per ogni Test Case realizzato, verranno prodotti i relativi Test Case Specifications contenenti gli esempi di tutte le casistiche individuate per ogni funzionalità con i relativi risultati.

8. TEST CASE SPECIFICATION

Di seguito verranno riportati tutti i test cases realizzati raggruppati nel package “Test”.

8.1 TC_AUT_1

1.1 TC_AUT_1.1: Login

Parametro: Username	
Formato: [A-Z][a-z], [0-9]	
Categoria	Scelta
Lunghezza lu :	1. Lunghezza==0[errore] 2. Lunghezza>=1 [property esisteLUok]
Formato fu:	1. Rispetta il formato [if esisteLUok] [property esisteFUok] 2. Non rispetta il formato [if esisteLUok][errore]

Parametro: Password	
Formato: [A-Z][a-z], [0-9]	
Categoria	Scelta
Lunghezza lp :	1. Lunghezza<8[errore] 2. Lunghezza>=8 [property esisteLPok]
Formato fp:	1. Rispetta il formato [if esisteLPok] [property esisteFPok] 2. Non rispetta il formato [if esisteLNok][errore]

CODICE	COMBINAZIONE	ESITO
TC_1.1_1	lu1	ERRORE.
TC_1.1_2	lu2.fu2	ERRORE.
TC_1.1_3	lu2.fu1.lp1	ERRORE.
TC_1.1_4	lu2.fu1.lp2.fp2	ERRORE.
TC_1.1_5	lu2.fu1.lp2.fp2.fp1	LOGGATO.

1.2 TC_AUT_1.2: Registrazione

Parametro: Nome Formato: [A-Z][a-z], [0-9]	
<i>Categoria</i>	<i>Scelta</i>
Lunghezza ln :	1. Lunghezza==0[errore] 2. Lunghezza>=1 [property lunghezzaLNok]
Formato fn:	1. Rispetta il formato [if lunghezzaLNok] [property lunghezzaFNok] 2. Non rispetta il formato [if lunghezzaLNok][errore]

Parametro: Cognome Formato: [A-Z][a-z], [0-9]	
<i>Categoria</i>	<i>Scelta</i>
Lunghezza lc :	1. Lunghezza==0[errore] 2. Lunghezza>=1 [property lunghezzaLCok]
Formato fc:	1. Rispetta il formato [if lunghezzaLCok] [property lunghezzaLCok] 2. Non rispetta il formato [if lunghezzaLCok][errore]

Parametro: Username Formato: [A-Z][a-z], [0-9]	
<i>Categoria</i>	<i>Scelta</i>
Lunghezza lu :	1. Lunghezza==0[errore] 2. Lunghezza>=1 [property lunghezzaLUok]
Formato fu:	1. Rispetta il formato [if lunghezzaLUok] [property lunghezzaLUok] 2. Non rispetta il formato [if lunghezzaLUok][errore]

Parametro: Password Formato: [A-Z][a-z], [0-9]	
Categoria	Scelta
Lunghezza lp :	<ol style="list-style-type: none"> 1. Lunghezza<8[errore] 2. Lunghezza<=8 [property lunghezzaLPok]
Formato fp:	<ol style="list-style-type: none"> 1. Rispetta il formato [if lunghezzaLPok] [property lunghezzaFPok] 2. Non rispetta il formato [if lunghezzaLPok][errore]

CODICE		COMBINAZIONE	ESITO
TC_1.2_1	ln1		ERRORE.
TC_1.2_2	ln2.fn2		ERRORE.
TC_1.2_3	ln2.fn1.lc1		ERRORE.
TC_1.2_4	ln2.fn1.lc2.fc2		ERRORE.
TC_1.2_5	ln2.fn1.lc2.fc1.lu1		ERRORE.
TC_1.2_6	ln2.fn1.lc2.fc1.lu2.fu2		ERRORE.
TC_1.2_7	ln2.fn1.lc2.fc1.lu2.fu1.lp1		ERRORE.
TC_1.2_8	ln2.fn1.lc2.fc1.lu2.fu1.lp2.fp2		ERRORE.
TC_1.2_9	ln2.fn1.lc2.fc1.lu2.fu1.lp2.fp1		REGISTRATO.

8.2 TC_PRO_2: Aggiungi Prodotto

Parametro: Nome	
Formato: [A-Z][a-z], [0-9]	
<i>Categoria</i>	<i>Scelta</i>
Lunghezza ln :	<ol style="list-style-type: none">1. Lunghezza==0[errore]2. Lunghezza>=1 [property lunghezzaLNok]
Formato fn:	<ol style="list-style-type: none">1. Rispetta il formato [if lunghezzaLNok] [property lunghezzaFNok]2. Non rispetta il formato [if lunghezzaLNok][errore]

Parametro: Descrizione	
Formato: [A-Z][a-z], [0-9]	
<i>Categoria</i>	<i>Scelta</i>
Lunghezza ld :	<ol style="list-style-type: none">1. Lunghezza==0[errore]2. Lunghezza>=1 [property lunghezzaLDok]
Formato fd:	<ol style="list-style-type: none">1. Rispetta il formato [if lunghezzaLDok] [property lunghezzaFDok]2. Non rispetta il formato [if lunghezzaLDok][errore]

Parametro: Prezzo	
Formato: [0-9]	
<i>Categoria</i>	<i>Scelta</i>
Lunghezza lp :	<ol style="list-style-type: none">1. Lunghezza==0[errore]2. Lunghezza>=1 [property lunghezzaLPok]
Formato fp:	<ol style="list-style-type: none">1. Rispetta il formato [if lunghezzaLPok] [property lunghezzaFPok]2. Non rispetta il formato [if lunghezzaLPok][errore]

Parametro: idCategoria	
Formato: [0-9]	
<i>Categoria</i>	<i>Scelta</i>
Correttezza ci :	<ol style="list-style-type: none">1. idCategoria è un item presente nella drop-down list apposta [propety correttezzaCIok]

CODICE	COMBINAZIONE	ESITO
<i>TC_2_1</i>	ln1	ERRORE.
<i>TC_2_2</i>	ln2.fn2	ERRORE.
<i>TC_2_3</i>	ln2.fn1.ld1	ERRORE.
<i>TC_2_4</i>	ln2.fn1.ld2.fd2	ERRORE.
<i>TC_2_5</i>	ln2.fn1.ld2.fd1.lp1	ERRORE.
<i>TC_2_6</i>	ln2.fn1.ld2.fd1.lp2.fp2	ERRORE.
<i>TC_2_7</i>	ln2.fn1.ld2.fd1.lp2.fp1.ci1	PRODUCT ADDED.

8.3 TC_CAT_3: Aggiungi Categoria

Parametro: Nome	
Formato: [A-Z][a-z]	
<i>Categoria</i>	<i>Scelta</i>
Lunghezza ln :	1. Lunghezza==0[errore] 2. Lunghezza>=1 [property lunghezzaLNok]
Formato fn:	1. Rispetta il formato [if lunghezzaLNok] [property lunghezzaFNok] 2. Non rispetta il formato [if lunghezzaLNok][errore]

Parametro: IdCategoria Padre	
Formato: [0-9]	
<i>Categoria</i>	<i>Scelta</i>
Lunghezza li :	1. Lunghezza==0[errore] 2. Lunghezza>=1 [property lunghezzaLIok]
Formato fi:	1. Rispetta il formato [if lunghezzaLIok] [property lunghezzaFIok] 2. Non rispetta il formato [if lunghezzaLNok][errore]
esiste eIdCat	1. Esiste nel DB [property esisteEIDCATok] 2. Non esiste nel DB [errore]

CODICE		COMBINAZIONE	ESITO
TC_3_1	ln1		ERRORE.
TC_3_2	ln2.fn2		ERRORE.
TC_3_3	ln2.fn1.li1		ERRORE.
TC_3_4	ln2.fn1.li2.fi2		ERRORE.
TC_3_5	ln2.fn1.li2.fi1.eIdCat2		ERRORE.
TC_3_6	ln2.fn1.li2.fi1.eIdCat1		CATEGORY ADDED.