

NowIWant
Test Plan Document
Versione 1.0

LOGO PROGETTO



Data: 18/12/2017

Progetto: NowIWant	Versione: 2.0
Documento: test plan document	Data: 18/12/2017

Coordinatore del progetto:

Nome	Matricola

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
17/12/2017	0.1	Abbozzo di stesura	FPA
17/12/2017	0.5	Tipi di testing	CA, LG
28/12/2017	1.0	Correzione errori	FPA, CA, RGA, LG

Indice

1. INTRODUZIONE4

2. RELAZIONE CON GLI ALTRI DOCUMENTI4

3. PANORAMICA DEL SISTEMA4

4. FUNZIONALITA' DA TESTARE E NON4

5. PASS/FAIL CRITERIA5

6. APPROCCIO5

 6.1 Testing di unità5

 6.2 Testing di integrazione.....5

 6.3 Testing funzionali.....6

1. Introduzione

Il documento Test Plan è stato creato per analizzare, gestire lo sviluppo e le attività di testing riguardante la piattaforma per verificarne il corretto funzionamento. Verranno provate delle 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.

3. Panoramica del sistema

Il sistema NowIWant è una piattaforma e-commerce per la vendita di prodotti di abbigliamento.

4. Funzionalità da testare

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

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

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 precedente 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:

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

6.1 *Testing di unità*

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 *Testing di integrazione*

Verrà effettuato tramite una strategia bottom-up. Per effettuare questi test verrà effettuata dal datalayer al business layer, dal business layer al view layer ed infine tra le tre.

6.3 Testing funzionali

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.

