



Test Plan Document



Riferimento	INDITEXReSkin_TP_2.0
Versione	2.0
Data	09/12/2023
Destinatario	C. Gravino
Presentato da	A. Ruggiero, G. Gurràli



Team composition

Ruolo	Nome	Acronimo	Contatti
Top Management	Carminè Gravino	CG	gravino@unisa.it
Team Member	Andrea Ruggiero	AR	a.ruggiero150@studenti.unisa.it
Team Member	Giovanni Gurràli	GG	g.gurràli@studenti.unisa.it



Sommario

Team composition	2
Revision History	4
1. Introduzione	5
2. Relazione con gli altri documenti	6
3. Panoramica del sistema	6
4. Funzionalità da testare/non testare	7
5. Pass/Fail criteria	7
6. Approccio	7
6.1. Testing di unità	7
6.2. Testing Funzionale	7
7. Strumenti per il testing (Hardware/Software)	8
8. Test cases	8
9. Specifica dei test case	9
9.2.TC_FRG -> Funzione registrazione	9
9.2. TC_FGI-> Funzione gestione inserzione	11
10. Pianificazione del testing ed Assegnazione dei ruoli	13



Revision History

Data	Versione	Descrizione	Autori
09/12/2023	0.1	Stesura Introduzione	[TEAM MEMBER]
10/12/2023	0.2	Relazione con gli altri documenti-Panoramica del sistema-Funzionalità da testare-Pass/Fail Criteria	[TEAM MEMBER]
11/12/2023	0.3	Approccio-Strumenti per il testing-Test cases	[TEAM MEMBER]
12/12/2023	0.4	TC_FRG	AG
12/12/2023	0.5	TC_FGI - Pianificazione del testing ed assegnazione dei ruoli	GG
17/12/2023	1.0	Revisione documento	[TEAM MEMBER]
14/02/2024	2.0	Revisione e modifiche post feedback	[TEAM MEMBER]

1. Introduzione

Nel corso dell'ultimo decennio, nonostante gli sforzi considerevoli compiuti da agenzie e governi per promuovere una vera e propria transizione ecologica attraverso ampi programmi di sviluppo, diversi settori continuano a lottare per adottare le pratiche comuni necessarie per lo smaltimento adeguato dei rifiuti. Uno dei settori più inquinanti che ha scalato posizioni, raggiungendo il secondo posto nella graduatoria tra quelli più dannosi per l'ambiente, è senz'altro quello della moda, in particolare il segmento noto come fast fashion, che si concentra sulla produzione in massa di capi d'abbigliamento. Diversi report hanno evidenziato la presenza di metano e petrolio, tra altre sostanze chimiche, in materiali ampiamente utilizzati dalle grandi aziende, come cotone, seta e fibre sintetiche. Queste sostanze, se non smaltite correttamente, possono contaminare le acque sotterranee o persistere nell'ambiente senza degradarsi, causando un inquinamento irreversibile del nostro pianeta. Inoltre, oggi, le aziende producono all'incirca 92 milioni di tonnellate di rifiuti tessili ogni anno, per ogni chilo di cotone consumano più di 20.000 litri d'acqua e quasi il 10% delle microplastiche disperse nell'oceano provengono dai tessuti non adeguatamente smaltiti. Problemi singoli ma che nell'insieme macchiano tutti i buoni propositi che da anni cerchiamo di portare avanti; propositi quali la sostenibilità ed il riciclo consapevole. Risolvere tutto e subito è irrealistico ma è bene iniziare a pensare a soluzioni alternative che possano modificare questa tendenza negativa. L'azienda INDITEX propone quindi un sistema innovativo chiamato INDITEX ReSkin. L'obiettivo principale di questo sistema è creare un canale di comunicazione che consenta a piccoli artigiani e innovative start-up italiane di acquistare a prezzi vantaggiosi il materiale di produzione in eccesso. L'azienda potrà dunque mettere in vendita sottoprodotti (ovvero uno scarto di lavorazione derivante da processi industriali, il quale viene riutilizzato in un altro processo produttivo come materia prima non vergine) derivanti dalla produzione di capi per i suoi principali brand di Fast Fashion. Questo approccio permette all'azienda di recuperare una parte delle spese sostenute per l'acquisto dei materiali, riducendo al contempo gli sprechi e promuovendo il riutilizzo delle stoffe e di altri materiali altrimenti destinati allo spreco. In particolare, l'azienda ha deciso di puntare sul mercato italiano proprio per la grande tradizione che l'Italia ha e ancora oggi mette in campo quando si parla di artigianato e innovazione nell'ambito della moda, puntando quindi a valorizzare questa eredità culturale ma al tempo stesso valorizzando i giovani talenti del territorio, che si mettono in gioco creando start-up innovative e all'insegna del green.

In particolare, andremo con questo documento a mostrare:

- Le funzionalità da testare
- L'approccio scelto per il testing
- I criteri scelti per il testing
- Test case

Tutto questo al fine di mettere in luce i punti deboli del nostro sistema software

2. Relazione con gli altri documenti

Questo documento è strettamente legato ai documenti precedentemente definiti, ma allo stesso tempo si pone come base anche per quelli che saranno i documenti futuri che dovranno essere redatti. Ogni test case che verrà analizzato è legato alle funzionalità individuate nel documento di raccolta ed analisi dei requisiti (RAD). In particolare:

- Relazione con il RAD
 - Dagli Use Case definiti nel RAD sono stati pianificati i relativi Test Case
- Relazione con l'SDD
 - Dati i sottosistemi definiti all'interno dell'SDD, ogni Test Case è legato a tale suddivisione

3. Panoramica del sistema

Dall'SDD: Lo sviluppo del sistema che stiamo proponendo segue lo stile architetturale "Three-tier". Questo tipo di approccio ci permetterà di dividere l'interfaccia dalla logica applicativa, consentendoci così di sviluppare e contestualmente di modificare diverse interfacce utente seguendo però la stessa logica applicativa.

Dal punto di vista delle tecnologie adottate per lo sviluppo abbiamo usato:

1. **Lato Server (Back-end):**
 - Java
2. **Interfaccia Utente (Front-end):**
 - HTML e CSS
 - JavaScript (JS)
 - Bootstrap 5.3
3. **Gestione delle Interazioni Client-Server:**
 - JQuery
4. **Gestione delle richieste asincrone:**
 - AJAX
5. **Persistenza dei Dati (Database):**
 - MySQL

In particolare, al fine di semplificarne lo sviluppo, l'applicazione web è stata suddivisa nei seguenti sottosistemi:

- Autenticazione e Gestione Utenti-> AGU
- Gestione delle Inserzioni-> GI
- Gestione degli Ordini-> GO
- Ricerca e Visualizzazione dei Prodotti-> RVP
- Gestione del Carrello-> GC
- Gestione delle Notifiche-> GN
- Piazzamento degli Ordini-> PO

4. Funzionalità da testare/non testare

Di tutte le funzionalità testabili per il nostro sistema ci siamo soffermati in particolare sulle funzionalità:

- **FRG.1: Registrazione OTP**
- **FGI.3: Modifica inserzione**

5. Pass/Fail criteria

La determinazione di un Pass o di un Fail per i nostri Test Case sono strettamente legati all'output osservato:

- **Pass:** Nel caso in cui l'output osservato sia lo stesso dell'output atteso, allora avremo un Pass
- **Fail:** Nel caso in cui l'output osservato sia diverso dall'output atteso, allora avremo un Fail

6. Approccio

Per il sistema che stiamo proponendo ed in particolare in base ai vincoli progettuali da seguire, è necessario effettuare due tipologie di Testing:

- **Testing di unità**
- **Testing di funzionale**

Entrambe le tipologie di testing verranno eseguite utilizzando un approccio **BlackBox**

6.1. Testing di unità

Nel testing di unità singoli sottosistemi o oggetti vengono testati separatamente. Questo comporta il vantaggio di ridurre il tempo di testing poiché vengono testate in parallelo piccole unità di sistema.

In particolare, nel nostro caso, andremo con questa tipologia di testing, tramite category partition, ad effettuare il test di esattamente quattro metodi del sistema sviuppato

6.2. Testing Funzionale

Vengono controllate le differenze tra i requisiti funzionali e il sistema. I test case vengono presi dai casi d'uso più importanti per l'utente finale e che hanno una buona probabilità di riscontrare un fallimento.

In particolare, nel nostro caso, andremo con questa tipologia di testing, tramite category partition, ad effettuare il test di esattamente due funzionalità del sistema sviluppato



7. Strumenti per il testing (Hardware/Software)

Le operazioni di testing per il nostro sistema verranno effettuate tramite:

- **Selenium:** Per il testing funzionale
- **JUnit e IntelliJ IDEA Code Coverage:** Per il testing di unità

8. Test cases

Per sviluppare i test cases sarà utilizzato il metodo del Category Partition. Questo metodo consiste nell'identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore. I test cases verranno definiti nel documento di Test Cases Specification (TCS).



9. Specifica dei test case

9.2.TC_FRG -> Funzione registrazione

Vincoli

Parametro: Email	
Formato:	
$^{\wedge}[a-zA-Z0-9.-]{1,29}[a-zA-Z0-9]@[a-zA-Z0-9.-]{1,29}[a-zA-Z0-9]\.[a-zA-Z]{1,5}\$$	
Nome categoria	Scelta per la categoria
FormatoEmail [FE]	1. Formato = true [PROPERTY_FE_OK] 2. Formato = false [ERROR]
Nome categoria	Scelta per la categoria
MatchEmail [ME]	1. Email Match in DB = true [ERROR] 2. Email Match in DB = false [PROPERTY_ME_OK]
Parametro: Password	
Formato:	
$^{\wedge}(?=[a-z])(?=[A-Z])(?=[\d])(?=[@!\%*\?&])[A-Za-z\d@!\%*\?&]{8,30}\$$	
Nome categoria	Scelta per la categoria
FormatoPassword [FP]	1. Formato = true [PROPERTY_FP_OK] 2. Formato = false [ERROR]
Parametro: Nome	
Nome categoria	Scelta per la categoria
LunghezzaNome [LN]	1. Lunghezza >= 2 AND Lunghezza <= 30 [PROPERTY_LN_OK] 2. Lunghezza < 2 OR Lunghezza > 30 [ERROR]
Parametro: Cognome	
Nome categoria	Scelta per la categoria
LunghezzaCognome [LC]	1. Lunghezza >= 2 AND Lunghezza <= 30 [PROPERTY_LC_OK] 2. Lunghezza < 2 OR Lunghezza > 30 [ERROR]



Parametro: OTP	
Formato: ^d{8}\$	
Nome categoria	Scelta per la categoria
FormatoOTP [FO]	1. Formato = true [PROPERTY_FO_OK] 2. Formato = false [ERROR]
Nome categoria	Scelta per la categoria
MatchOTP [MO]	1. OTP Match Software esterno = true [PROPERTY_MO_OK] 2. OTP Match Software esterno = false [ERROR]

Test frame

Testa case ID	Test frame	Esito
TC_FRG.1_1	FE2	Errore: Formato email non rispettato
TC_FRG.1_2	FE1, ME1	Errore: Email già registrata
TC_FRG.1_3	FE1, ME1, FP2	Errore: Password non valida
TC_FRG.1_4	FE1, ME2, FP1, LN2	Errore: Nome non valido
TC_FRG.1_5	FE1, ME2, FP1, LN1, LC2	Errore: Cognome non valido
TC_FRG.1_6	FE1, ME2, FP1, LN1, LC1, FO2	Errore: OTP inserito non valido
TC_FRG.1_7	FE1, ME2, FP1, LN1, LC1, FO1, MO2	Errore: OTP non corrispondente
TC_FRG.1_8	FE1, ME2, FP1, LN1, LC1, FO1, MO1	Registrazione eseguita

9.2. TC_FGI-> Funzione gestione inserzione

Vincoli

Parametro: Titolo	
Nome categoria	Scelta per la categoria
Lunghezza [LT]	1. Lunghezza >30 OR Lunghezza < 5 [ERROR] 2. Lunghezza <=30 AND Lunghezza >=5 [PROPERTY_LT_OK]
Parametro: Descrizione	
Nome categoria	Scelta per la categoria
Lunghezza [LD]	1. Lunghezza >150 OR Lunghezza < 20 [ERROR] 2. Lunghezza <=150 AND Lunghezza >=20 [PROPERTY_LD_OK]
Parametro: Categoria	
Nome categoria	Scelta per la categoria
Esistenza[EC]	1. Categoria \notin "Insieme categorie selezionabili" [ERROR] 2. Categoria \in "Insieme categorie selezionabili" [PROPERTY_EC_OK]
Parametro: Dimensione lunghezza	
Nome categoria	Scelta per la categoria
Valore [DLnV]	1. Valore <=0 [ERROR] 2. Valore >0 [PROPERTY_DLnV_OK]
Parametro: Dimensione larghezza	
Nome categoria	Scelta per la categoria
Valore [DLaV]	1. Valore <=0 [ERROR] 2. Valore >0 [PROPERTY_DLaV_OK]
Parametro: Quantità	
Nome categoria	Scelta per la categoria
Valore[QV]	1. Valore <0 [ERROR] 2. Valore >=0 [PROPERTY_QV_OK]
Parametro: Prezzo	
Nome categoria	Scelta per la categoria
Valore[PV]	1. Valore <=0 [ERROR] 2. Valore >0 [PROPERTY_PV_OK]

Test frame



Testa case ID	Test frame	Esito
TC_FGI.3_1	LT1	Errore: Lunghezza del titolo non rispettata
TC_FGI.3_2	LT2,LD1	Errore: Lunghezza descrizione non rispettata
TC_FGI.3_3	LT2,LD2,EC1	Errore: Categoria selezionata non esistente
TC_FGI.3_4	LT2,LD2,EC2, DLnV1	Errore: Valore inserito non valido
TC_FGI.3_5	LT2,LD2,EC2, DLnV2,DLaV1	Errore: Valore inserito non valido
TC_FGI.3_6	LT2,LD2,EC2, DLnV2,DLaV2,QV1	Errore: Valore inserito non valido
TC_FGI.3_7	LT2,LD2,EC2 ,DLnV2,DLaV2,QV2,PV1	Errore: Valore inserito non valido
TC_FGI.3_8	LT2,LD2,EC2,DLnV2,DLaV2,QV2,PV2	Modifica eseguita correttamente



10. Pianificazione del testing ed Assegnazione dei ruoli

Nome	Acronimo	Contatti	Test case da testare
Andrea Ruggiero	AR	a.ruggiero150@studenti.unisa.it	TC_FGI
Giovanni Gurrallì	GG	g.gurrallì@studenti.unisa.it	TC_FRG