



Test Plan

C08-Just Traditions

Riferimento	2022_TP_C08
Versione	1.1
Data	07/02/2023
Destinatario	Top Managment
Presentato da	G. Sgambato, L. Sorrentino, M. Spagnuolo, D. Avella, G. Miele
Approvato da	F. Cirillo, G. Brescia



Revision History

Data	Versione	Descrizione	Autori
24/11/2022	0.1	Prima stesura	Franco Cirillo
10/11/2022	0.2	Aggiunta Test Case	Tutti
11/12/2022	1.0	Revisione	Tutti
07/02/2023	1.1	Modifica test case sottomissione annuncio	LS



Sommario

1	Introduzione	4
2	Relazione con altri documenti	4
3	Panoramica del sistema	5
4	Feature da testare e feature da non testare	5
5	Pass/Fail Criteria	6
6	Approccio	6
7	Sospensione e ripristino	7
8	Materiale per il testing	7
9	Test Cases	7
9.1	Gestione Annunci	8
9.1.1	Modifica stato annuncio	8
9.1.2	Sottomissione annuncio	9
9.2	Gestione Prenotazioni	13
9.2.1	Visualizzazione prenotazioni annuncio	13
9.2.2	Effettua prenotazione	14
9.3	Gestione E-commerce	16
9.3.1	Ricerca annunci	16
10	Testing schedule	17



1 Introduzione

La piattaforma Just Traditions ha come scopo principale di mettere a disposizione dell'azienda Green Apple un sistema per valorizzare il turismo responsabile e sostenibile in Italia, favorendo la positiva interazione tra industria del turismo, comunità locali e viaggiatori.

In particolare, Just Traditions darà la possibilità di prenotare visite guidate presso le attività degli artigiani di tutta Italia. Inoltre, si pone l'obiettivo di essere una piattaforma altamente fruibile e tale scopo, per agevolare le ricerche degli utenti, sarà possibile restringere la ricerca degli annunci a quelli di una determinata provincia; in modo tale che ogni utente possa conoscere immediatamente le attività della propria zona.

Le attività di testing pianificate interesseranno i seguenti moduli:

- Gestione Annunci
- Gestione Prenotazioni
- Gestione E-commerce

2 Relazione con altri documenti

Per la pianificazione dei casi di test si fa riferimento anche ad altri documenti sviluppati, nello specifico:

- **Requirements Analysis Document (RAD)**

Il documento risulta essere il punto di riferimento per la selezione dei requisiti funzionali da testare.

- **System Design Document (SDD)**

Da tale documento risulta necessario l'apprendimento della suddivisione del sistema in sottosistemi in modo da poter procedere al testing in modo coerente con la suddivisione effettuata.



3 Panoramica del sistema

La piattaforma Just Traditions adotterà un'architettura Three Tier. Per la parte di front end verranno utilizzati i seguenti linguaggi:

- HTML5
- CSS3
- JavaScript

Per la logica applicativa e quindi il back-end sarà utilizzato Java SPRING.

Per la gestione del database saranno usati:

- Spring JPA per il collegamento al database.
- Per la gestione del DB verrà usato MySQL.

4 Feature da testare e feature da non testare

Feature da testare

Le attività di testing da effettuare nel periodo antecedente al primo rilascio interessano alcune delle funzionalità indicate con una priorità alta all'interno della specifica dei requisiti.

Nel dettaglio sono:

- Gestione Annunci
 - Modifica stato annuncio
 - Sottomissione annuncio
- Gestione Prenotazioni
 - Visualizzazione prenotazioni annuncio
 - Effettua prenotazione
- Gestione E-commerce
 - Ricerca annunci

Feature da non testare

Le attività di testing precedenti al primo rilascio non interesseranno i requisiti di media e bassa priorità individuati in fase di analisi.



5 Pass/Fail Criteria

Le attività di testing sono mirate a identificare la presenza di faults (errori) all'interno del sistema, per effettuare un successivo intervento di correzione.

L'esito di un test case è valutato mediante un oracolo, inteso come il risultato atteso dalla sua esecuzione, basandosi sui requisiti. Quest'ultimo verrà identificato durante le attività di Test Specification.

Pass

Un test case avrà successo se la sua esecuzione darà come risultato quello indicato dall'oracolo.

Fail

Un test case avrà esito di fallimento se la sua esecuzione darà un differente risultato da quello indicato dall'oracolo.

La fase di testing sarà validata soltanto con la soddisfazione dei seguenti vincoli:

- Tutte le feature indicate come da testare saranno testate,
- Verranno effettuati nuovamente i test in caso di modifiche al sistema.

6 Approccio

L'intera fase di testing ha come scopo principale la produzione di un prodotto di qualità che soddisfi i requisiti specificati. Le attività di testing saranno suddivise in:

- **Testing di unità** che mira all'individuazione di faults negli oggetti e/o sottosistemi coinvolti rispetto agli use cases dello use case model. Tale tipologia di testing viene svolta in modo parallelo alle attività di sviluppo.
- **Testing di sistema** che mira all'identificazione di faults rispetto agli scenari, i requisiti e i design goals identificati nell'analisi dei requisiti e nel system design, testando tutte le componenti insieme viste come un unico sistema.

Testing di unità

L'esecuzione dell'attività di testing unitario prevede il testing di un metodo delle classi controller per ciascun membro del team, verrà utilizzato per la maggior parte la strategia black-box, nel caso in cui si renderà necessario, ai fini del miglioramento della branch coverage, si adotterà anche la strategia white-box. Per l'esecuzione di tale attività verrà utilizzato il framework *JUnit*; per ogni classe sviluppata dovrà essere sviluppata una classe di test associata contenente i casi di test previsti.



Altre tecnologie usate in tale fase saranno:

- *Mockito*: per la costruzione degli stub e l'isolamento della componente testata.
- *Maven*: per la build e l'esecuzione automatica dei tests.

Testing di Sistema

L'esecuzione dell'attività di testing del sistema sarà effettuata mediante l'utilizzo di Selenium IDE. L'utilizzo di tale strumento permetterà di generare facilmente test automatizzati e permetterà inoltre di registrare ed archiviare ciò che è stato effettuato per massimizzare la riusabilità. Il testing di sistema sarà effettuato tramite category partition e riguarderà esattamente una funzionalità del sistema sviluppato.

7 Sospensione e ripristino

Di seguito sono riportati i criteri di sospensione e ripristino validi per tutte le attività di testing.

Criteri di sospensione

L'attività di testing viene sospesa se durante la sua esecuzione:

1. Vengono rilevati errori critici.
2. Sopraggiunge una Change Request che potrebbe portare alla modifica di ciò che si sta testando.

Criteri di ripristino

L'attività di testing verrà ripristinata quando:

1. Tutti i problemi relativi ai criteri di sospensione sono stati risolti.
2. Nel caso in cui la sospensione è dovuta al rilevamento di un errore critico, si riprenderà il testing solo se è stata eseguita di nuovo l'attività di testing sulla componente interessata dalla revisione, prima di poter procedere ai successivi test pianificati.

8 Materiale per il testing

I team member utilizzeranno hardware e software personale e sfrutteranno servizi di connettività di rete propria per le attività di testing. Il setup iniziale dell'ambiente avverrà in un incontro tra tutti i team members ed i project managers.



9 Test Cases

L'approccio per la definizione dei test frame sarà il category partition. Al fine di minimizzare il numero di test case, gli input saranno partizionati in classi di equivalenza.

9.1 Gestione Annunci

9.1.1 Modifica stato annuncio

Parametro: Nuovo stato annuncio	
Nome categoria	Scelte per la categoria
Correttezza [CE]	<pre>1.(StatoAttuale == "proposto" AND NuovoStato != "in revisione") OR (StatoAttuale == "in revisione" AND NuovoStato != "approvato" AND NuovoStato != "rifiutato") OR (StatoAttuale == "approvato" AND NuovoStato != "in revisione") OR (StatoAttuale == "rifiutato") [errore] 2.NOT((StatoAttuale == "proposto" AND NuovoStato != "in revisione") OR (StatoAttuale == "in revisione" AND NuovoStato != "approvato" AND NuovoStato != "rifiutato") OR (StatoAttuale == "approvato" AND NuovoStato != "in revisione") OR (StatoAttuale == "rifiutato")) [PROPERTY_CE_OK]</pre>



StatoAttuale corrisponde allo stato attuale dell'annuncio salvato nel sistema.

Parametro: Motivo del rifiuto	
Nome categoria	Scelte per la categoria
Esistenza [EI]	1.NuovoStato == "rifiutato" AND MotivoDelRifiuto BLANK [errore] 2.(NuovoStato == "rifiutato" AND MotivoDelRifiuto NOT BLANK) OR (NuovoStato != "rifiutato") [PROPERTY_EI_OK]
Lunghezza [LN]	1.Lunghezza > 255 [errore] 2.Lunghezza <= 255 [PROPERTY_LN_OK]

Test Case ID	Test frame	Esito
TC_1.1_1	CE1	Errato: nuovo stato "proposto" con stato attuale "in revisione"
TC_1.1_2	CE2, EI1	Errato: motivo del rifiuto vuoto con nuovo stato "rifiutato"
TC_1.1_3	CE2, EI2, LN1	Errato: motivo del rifiuto troppo lungo
TC_1.1_4	CE2, EI2, LN2	Corretto

9.1.2 Sottomissione annuncio

Parametro: Nome dell'attività	
Nome categoria	Scelte per la categoria
Lunghezza [LN]	1.Lunghezza < 2 or Lunghezza > 40 [error] 2. Lunghezza ≥ 2 and Lunghezza ≤ 40 [property LN OK]



Parametro: Descrizione	
Nome categoria	Scelte per la categoria
Lunghezza [LD]	1. Lunghezza < 2 or Lunghezza > 1024 [error] 2. Lunghezza \geq 2 and Lunghezza \leq 1024 [property LD_OK]

Parametro: Indirizzo	
Nome categoria	Scelte per la categoria
Lunghezza [LI]	1. Lunghezza < 2 or Lunghezza > 128 [error] 2. Lunghezza \geq 2 and Lunghezza \leq 128 [property LI OK]

Parametro: Provincia	
Nome categoria	Scelte per la categoria
Validità [VP]	1. Provincia Non Valida [error] 2. . Provincia Valida [property VP_OK]

Parametro: Servizi offerti	
Nome categoria	Scelte per la categoria
Lunghezza [LS]	1. Lunghezza < 2 or Lunghezza > 255 [error] 2. Lunghezza \geq 2 and Lunghezza \leq 255 [property LS_OK]

Parametro: Prezzo	
FORMATO: $\wedge \backslash d\{1,3\}(\backslash . \backslash d\{1,2\})? \$$	
Formato [FP]	1. Non rispetta il formato [error] 2. Rispetta il formato [property FP_OK]



Parametro: Giorno delle visita	
Nome categoria	Scelte per la categoria
Correttezza [CG]	1. Giorno delle visita != Giorno della settimana [error] 2. Giorno delle visita == Giorno della settimana [property CG_OK]

Si è scelto di riportare giorno della visita, invece di giorni delle visite, ai fini di semplificare il testing evitando input multipli

Parametro: Orario inizio	
FORMATO: ^([01][0-9] 2[0-3])(:([0-5][0-9]))\$ (HH:MM)	
Formato[FF]	1. Non rispetta il formato [error] 2. Rispetta il formato [property FF_OK]

Parametro: Orario Fine	
FORMATO: ^([01][0-9] 2[0-3])(:([0-5][0-9]))\$ (HH:MM)	
Formato[FK]	1. Non rispetta il formato [error] 2. Rispetta il formato [property FK_OK]

Si è scelto di riportare orario inizio e orario fine, al posto di fasce orarie, ai fini di semplificare il testing evitando input multipli ed essere conformi all'implementazione che prevede due input distinti per la fascia oraria



Parametro: Numero massimo persone	
Valore[VN]	1. Valore ≤ 0 [error] 2. Valore > 0 [property VN_OK]

Parametro: Foto	
Grandezza File [GFF]	1. Grandezza $> 15\text{MB}$ [errore] 2. Grandezza $\leq 15\text{MB}$ [PROPERTY GFF_OK]

Test Case ID	Test frame	Esito
TC_1.2_1	LN1	Errore: Lunghezza del nome invalida
TC_1.2_2	LN2.LD1	Errore: Lunghezza della descrizione invalida
TC_1.2_3	LN2.LD2.LI1	Errore: Lunghezza dell'indirizzo invalida
TC_1.2_4	LN2.LD2.LI2.VP1	Errore: Provincia invalida
TC_1.2_5	LN2.LD2.LI2.VP2.LS1	Errore: Lunghezza di servizi offerti invalida
TC_1.2_6	LN2.LD2.LI2.VP2.LS2.FP1	Errore: Formato Prezzo errato
TC_1.2_7	LN2.LD2.LI2.VP2.LS2.FP2.CG1	Errore: giorno della visita errato
TC_1.2_8	LN2.LD2.LI2.VP2.LS2.FP2.CG2.FF1	Errore: Formato orario inizio errato
TC_1.2_9	LN2.LD2.LI2.VP2.LS2.FP2.CG2.FF2.FK1	Errore: Formato orario fine errato
TC_1.2_10	LN2.LD2.LI2.VP2.LS2.FP2.CG2.FF2.FK2.VN1	Errore: Valore Numero massimo persone invalido
TC_1.2_11	LN2.LD2.LI2.VP2.LS2.FP2.CG2.FF2.FK2.VN2.GFF1	Errore: Dimensione file errata
TC_1.2_12	LN2.LD2.LI2.VP2.LS2.FP2.CG2.FF2.FK2.VN2.GFF2	Corretto



9.2 Gestione Prenotazioni

9.2.1 Visualizzazione prenotazioni annuncio

Parametro: Data	
Formato: ^2\d{3}-(0[1-9] 1[0-2])-(0[1-9] [12]\d 3[01])\$	
Nome Categoria	Scelte per la categoria
Formato [FNO]	<ol style="list-style-type: none">1. Non rispetta il formato [error]2. Rispetta il formato [PROPERTY_FNO_OK]

Test Case ID	Test frame	Esito
TC_2.1_1	FNO1	Errato: La data non rispetta il formato
TC_2.1_2	FNO2	Corretto



9.2.2 Effettua prenotazione

Parametro: Data della visita	
Formato: $^2\d{3}-(0[1-9] 1[0-2])-(0[1-9] 12)\d{3}[01]$\n\n(AAAA-MM-GG)$	
Nome categoria	Scelte per la categoria
Formato [FO]	1. Non rispetta il formato [errore] 2. Rispetta il formato [PROPERTY_FO_OK]
Correttezza [CD]	1. DataVisita < DataCorrente [errore] 2. DataVisita >= DataCorrente [property_CD_OK]
Validità [VD]	1.Data non valida [errore] 2.Data valida [PROPERTY_VD_OK]

La validità della data è controllata verificando che in quel giorno l'annuncio abbia visite disponibili.

Parametro: idVisita	
Nome categoria	Scelte per la categoria
Validità [VI]	1.idVisita non valido [errore] 2.idVisita valido [PROPERTY_VI_OK]



L'id per essere valido deve identificare una visita che si svolga nel giorno della settimana corrispondente alla data scelta.

Si è scelto di riportare idVisita al posto di fascia oraria per essere più conforme all'implementazione, nella quale si è scelto di mostrare la fascia oraria a livello di interfaccia, ma sottomettere al controller l'idVisita per motivi di performance.

Parametro: Numero di persone prenotate	
Nome categoria	Scelte per la categoria
Valore [VN]	1.Valore ≤ 0 [error] 2.Valore > 0 [property_VN_OK]
Correttezza [CN]	1. Numero di persone prenotate $>$ Numero prenotazioni disponibili [errore] 2. Numero di persone prenotate \leq Numero prenotazioni disponibili [property_CN_OK]

Numero prenotazioni disponibili non sarà altro che il risultato dato da:

Numero massimo di persone per visita - prenotazioni effettuate per la visita nella data inserita.

Test Case ID	Test Frame	Esito
TC_2.2_1	F01	Errore: formato data errato
TC_2.2_2	F02.CD1	Errore: data nel passato
TC_2.2_3	F02.CD2.VD1	Errore: data non valida
TC_2.2_4	F02.CD2.VD2.VI1	Errore: idVisita non valida
TC_2.2_5	F02.CD2.VD2.VI2.VN1	Errore: Numero di persone prenotate ≤ 0
TC_2.2_6	F02.CD2.VD2.VI2.VN2.CN1	Errore: Numero di persone prenotate non disponibili
TC_2.2_7	F02.CD2.VD2.VI2.VN2.CN2	Corretto



9.3 Gestione E-commerce

9.3.1 Ricerca annunci

Parametro: Nome	
Nome categoria	Scelte per la categoria
Lunghezza[LN]	1. Lunghezza > 40 [error] 2. Lunghezza ≤ 40 [property LN_OK]

Parametro: Provincia	
Nome categoria	Scelte per la categoria
Validità[VP]	1.Provincia non Valida [errore] 2. Provincia Valida [property VP_OK]

Test Case ID	Test frame	Esito
TC_3.1_1	LN1	Errato: lunghezza non corretta
TC_3.1_2	LN2.VP1	Errato: Provincia non valida
TC_3.1_3	LN2.VP2	Corretto



10 Testing schedule

Le attività di pianificazione del testing avverranno come definito nei capitoli precedenti, cioè subito dopo la fase di design necessaria per la pianificazione.

La scrittura dei casi di test avverrà in contemporanea con lo sviluppo del codice.

L'esecuzione dei test avverrà sia durante che dopo l'implementazione del sistema. Una volta concluso lo sviluppo, tutti i test saranno eseguiti nuovamente per garantire il corretto funzionamento e produrre i report finali.