



TP HAPPY FIELDS

Sommario

1. Introduzione.....	2
2. Relazione con altri documenti.....	2
3. Panoramica del sistema	2
4. Features da testare/non testare	3
5. Pass/Fail criteria	3
6. Sospensione e Ripristino	3
7. Materiale di Testing	3
8. Test Cases	3
9. Testing Schedule	9

Team members

Nome	Matricola
Carlo Tammaro	0512112173
Antonio Renzullo	0512111906
Domenico Mattia Garofalo	0512110994
Annachiara Giugliano	0512111849

1. Introduzione

Il documento che segue ha l'obiettivo di descrivere e analizzare l'attività del testing per la piattaforma HappyFields. All'interno del documento sono riportate le strategie di testing e le funzionalità che verranno testate.

2. Relazione con altri documenti

Per la corretta individuazione dei test case, si fa riferimento ad altri documenti prodotti.

Relazioni con il Requirements Analysis Document (RAD)

I test case pianificati nel Test Plan sono elaborati in relazione ai requisiti funzionali e non funzionali presentati nel RAD.

Relazioni con il System Design Document (SDD)

I test case pianificati nel Test Plan devono rispettare la suddivisione in sottosistemi presentata nell'SDD.

Relazioni con il Object Design Document (ODD)

Per ciò che concerne il test di unità e di integrazione, maggiormente legati allo ODD e alla divisione in package del sistema, essi saranno scritti e documentati unicamente all'interno del codice dell'applicativo. Per tale motivo, nel presente documento, non vi saranno riferimenti al loro design.

3. Panoramica del sistema

Il sistema proposto è basato sullo stile architetturale Three Tier, implementato utilizzando MVC. Il motivo è che tale architettura è perfetta per lo sviluppo di Web App, come visto e testato nel corso di TSW. In particolare, la separazione dalla logica di presentazione da quella di elaborazione, migliora alcuni aspetti implementativi andando a favorire lo sviluppo del sistema.

Verranno utilizzati HTML5, CSS3, ed eventuali altri Framework per la parte di front-end.

Per la logica applicativa e back-end sarà utilizzato **Java**

Per la gestione del database verrà usato **SQL**

4. Features da testare/non testare

Di seguito la lista dei requisiti funzionali che sono stati testati:

- Registrazione Utente (Gestione Utente);
- Ricerca per orario e disponibilità (Gestione Evento);
- Creazione Evento (Gestione Evento);
- Modifica Dati (Gestione Gestore);

5. Pass/Fail criteria

Le attività di testing sono mirate a identificare la presenza di errori all'interno del sistema. L'esito di un test case è valutato mediante un oracolo, che esplicita se il test è andato a buon fine o meno.

Un test ha successo se, dato un input al sistema, l'output ottenuto è diverso dall'output dell'oracolo.

Un test fallisce se, dato un input al sistema, l'output ottenuto è uguale all'output atteso dall'oracolo.

Il test sarà considerato valido se saranno testati almeno **quattro** requisiti funzionali.

6. Sospensione e Ripristino

In questa sezione verranno specificati i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprende il test.

Criteri di sospensione: Il testing non verrà sospeso fino alla sua terminazione, anche in caso di rilevazione di una failure. Il testing potrà essere momentaneamente sospeso nel caso venga restituito, al momento dell'esecuzione, un errore nella definizione di uno dei test stessi.

Criteri di ripristino: Il testing verrà ripreso dopo aver risolto i fault individuati.

7. Materiale di Testing

L'hardware necessario per l'attività di testing è un semplice computer, che non necessita nemmeno di una connessione a internet.

8. Test Cases

L'approccio per la definizione dei test frame sarà il category partition. Per definire l'output atteso si userà un oracolo umano, per via dell'assenza di specifiche formali/semi-formali.

GESTIONE UTENTE

Registrazione Utente (RF_GU_1)

Parametro: Username	
Nome Categoria	Scelta per Categoria
Lunghezza [LU]	1. Lunghezza > 30 OR Lunghezza < 1 [errore] 2. Lunghezza <= 30 AND Lunghezza >= 1 [ok]
Formato [FU]	1. Nome.contains(!; ”; ”; £; \$; %; &; /; (;); =; ?; ^; ¢; _; -; @; °; §; [;]; {; }; #) [errore] 2. Nome [ok]
Parametro: Città	
Verifica [VC]	1. CittàInserita == NULL [errore] 2. Città [ok]
Parametro: Provincia	
Verifica [LP]	1. Provincia > 2 AND Provincia < 2 [errore] 2. Provincia == 2 [ok]
Parametro: Via	
Lunghezza [LV]	1. Lunghezza > 30 OR Lunghezza < 1 [errore] 2. Lunghezza <= 30 AND Lunghezza >= 1 [ok]
Parametro: Numero di Telefono	
Lunghezza [LNT]	1. Lunghezza < 10 OR Lunghezza > 10 [errore] 2. Lunghezza == 10 [ok]
Parametro: Password	
Lunghezza [LP]	1. Lunghezza < 8 [errore] 2. Lunghezza >= 8 [ok]
Maiuscola [MP]	1. !Password.contains(Maiuscola) [errore] 2. Password [ok]



TEST_CASE_ID	Test Frame	Esito
TC 1.1_1	LU1	Errato: username troppo lungo
TC 1.1_2	LU2, FU2	Errato: formato username errato
TC 1.1_3	LU2, FU2, VC1	Errato: il campo città è nullo
TC 1.1_4	LU2, FU2, VC2, LP1	Errato: provincia troppo lunga
TC 1.1_5	LU2, FU2, VC2, LP2, LV1	Errato: via troppo lunga
TC 1.1_6	LU2, FU2, VC2, LP2, LV2, LNT1	Errato: numero di telefono troppo corto
TC 1.1_7	LN2, FN2, VC2, LP2, LV2, LNT2, LP1,	Errato: password troppo breve
TC 1.1_8	LN2, FN2, VC2, LP2, LV2, LNT2, LP2, MP1	Errato: la password deve contenere almeno una lettera maiuscola
TC 1.1_9	LN2, FN2, VC2, LP2, LV2, LNT2, LP2, MP2, CP2	Corretto: registrazione va a buon fine.

Ricerca per data e provincia (RF_GE_5)

Parametro: Data	
Errore data [ED]	1. Data attuale > data specificata [Errore] 2. Data attuale <= data specificata [ok]
Parametro: provincia	
Lunghezza [LP]	1. Lunghezza > 2 OR Lunghezza < 2 [errore] 2. Lunghezza == 2 [OK]

TEST_CASE_ID	Test Frame	Esito
TC 2.1_1	ED1	Errato: data già trascorsa
TC 2.1_2	ED2, LP1	Errato: provincia troppo lunga
TC 2.1_3	ED2, LP2	Corretto: ricerca va a buon fine

Creazione Evento (RF_GE_1)

Parametro: Titolo Evento	
Lunghezza [LT]	Lunghezza > 50 OR Lunghezza < 1 [errore] Lunghezza <= 50 AND Lunghezza >= 1 [ok]
Parametro: Data	
Errore data [ED]	1.Data attuale> data specificata [Errore] 2. Data attuale<= data specificata [ok]
Parametro: Ora	
Scelta Obbligatoria [SOO]	1. Campo inserito = false [errore] 2. Campo inserito = true [ok]

TEST_CASE_ID	Test Frame	Esito
TC 2.2_1	LT1	Errato: titolo troppo lungo
TC 2.2_2	LT2, ED1	Errato: campo data precedente alla data odierna
TC 2.2_3	LT2, ED2, SOO1	Errato: campo ora è nullo
TC 2.2_4	LT2,ED2, SOO2	Corretto: la creazione dell'evento va a buon fine

Modifica dati campetti (RF_GG_3)

Parametro: Numero Giocatori	
Nome Categoria	Scelta per Categoria
Massimo [MNG]	1. NumeroGiocatori <= 2 AND NumeroGiocatori >= 11 [errore] 2. NumeroGiocatori >= 2 AND NumeroGiocatori <= 11 [ok]
Parametro: Città	
Lunghezza [LC]	1. Lunghezza <= 2 OR Lunghezza > 20 [errore] 2. Lunghezza > 2 AND Lunghezza <= 20 [ok]
Parametro: Provincia	
Lunghezza [LP]	3. Lunghezza > 2 OR Lunghezza < 2 [errore] 4. Lunghezza == 2 [OK]
Parametro: Via	
Lunghezza [LV]	1. Lunghezza <= 3 OR Lunghezza > 30 [errore] 2. Lunghezza > 3 AND Lunghezza <= 30 [ok]
Parametro: Costo	
Verifica [VC]	1. Costo <= 1 AND Costo >= 300 [errore] 2. Costo >= 1 AND Costo <= 300 [ok]
Parametro: Lunghezza	
Lunghezza [LL]	1. Lunghezza >= 150 mt AND Lunghezza <= 10mt [errore] 2. Lunghezza <= 150 mt AND Lunghezza >= 10mt [ok]
Parametro: Larghezza	
Lunghezza [LLA]	1. Lunghezza >= 50 mt AND Lunghezza <= 5mt [errore] 2. Lunghezza <= 50 mt AND Lunghezza >= 5mt [ok]
Parametro: Nome	
Lunghezza [LN]	1. Lunghezza < 1 OR Lunghezza > 30 [Errore] 2. Lunghezza >= 1 AND Lunghezza <= 30 [ok]
Formato [FU]	1. Nome.contains(!; ", "; £; \$; %; &; /; (;); =; ?; ^; ¢; _; -; @; °; §; [;]; {; }; #) [errore] 2. Nome [ok]

TEST_CASE_ID	Test Frame	Esito
TC 3.1_1	MNG1	Errato: il numero di giocatori è troppo alto.
TC 3.1_2	MNG2, LC1	Errato: la lunghezza della città è sbagliata
TC 3.1_3	MNG2, LC2, LP1	Errato: la lunghezza della provincia è sbagliata
TC 3.1_4	MNG2, LC2, LP2, LV1	Errato: la lunghezza della via è sbagliata
TC 3.1_5	MNG2, LC2, LP2, LV2, VC1	Errato: il costo è troppo elevato.
TC 3.1_6	MNG2, LC2, LP2, LV2, VC1, LL1	Errato: la lunghezza è troppo elevata.
TC 3.1_7	MNG2, LC2, LP2, LV2, VC2, LL2, LLA1	Errato: la larghezza è troppo elevata.
TC 3.1_8	MNG2, LC2, LP2, LV2, VC2, LL2, LLA2, LN1	Errato: la lunghezza del nome è sbagliata.
TC 3.1_9	MNG2, LC2, LP2, LV2, VC2, LL2, LLA2, LN2, FN1	Errato: il formato del nome è sbagliato.
TC 3.1_10	MNG2, LC2, LP2, LV2, VC2, LL2, LLA2, LN2, FN2	Corretto: la modifica va a buon fine.

9. Testing Schedule

Le attività di pianificazione dei testing avverranno dopo la fase di design necessaria per la pianificazione.



La scrittura dei casi di test avverrà durante la scrittura del codice.

L'esecuzione dei casi di test avverrà durante e dopo l'implementazione del sistema.