

Università degli Studi di Salerno

Corso di Ingegneria del Software



Sandwich on web

Test plan

Versione 1.0

Data: 24/01/2016

Partecipanti:

Nome	Matricola
Sara Volpe	0512102434
Egidio Giacoia	0512102376
Nunzia Esposito	0512102328

Revision History

Data	Versione	Descrizione	Autore
21/01/2016	1	Prima stesura del documento	Nunzia Esposito
24/01/2016	1	Revisione	Sara Volpe

Indice

1. INTRODUZIONE
2. RELAZIONE CON ALTRI DOCUMENTI
3. PANORAMICA DEL SISTEMA
4. FUNZIONALITA' DA TESTARE E NON
 - Gestione Account
 - Gestione Menù
 - Gestione Carrello
 - Gestione Recensione
5. PASS/FAIL CRITERIA
6. APPROCCIO
 - 6.1 Testing di unità
 - 6.2 Testing di integrazione
 - 6.3 Testing di sistema
7. TEST CASES
8. SPECIFICA DEI TEST CASES
 - 8.1 Gestione Account
 - 8.1.1 Login

- 8.1.2 Registrazione Utente
- 8.1.3 Modifica dati utente
- 8.2 Gestione Recensione
 - 8.2.1 Inserisce recensione
- 8.3 Gestione Carrello
 - 8.3.1 Completa ordine
- 8.4 Gestione Menù
 - 8.4.1 Acquista menù

1. Introduzione

Lo scopo di questo documento è quello di analizzare e gestire lo sviluppo e le attività di testing riguardanti il software Sandwich On Web. Questa sessione di lavoro deve verificare il corretto funzionamento di SOW in diversi casi, studiati appositamente per mettere alla prova ogni singola funzionalità e caratteristica del sistema, al fine di ottenere un corretto funzionamento. I risultati di questi test saranno utilizzati per capire dove bisognerà intervenire, e quindi correggere eventuali errori o apportare modifiche per il miglioramento dei vari sottosistemi. Il processo verrà iterato fino a che non si otterranno i risultati attesi in accordo con i tempi di sviluppo previsti.

2. Relazione con gli altri documenti

Per verificare il corretto funzionamento del sistema SOW sono stati predisposti dei test cases basati sulle funzionalità individuate sia nel documento inerente gli scenari che su quello di analisi dei requisiti (RAD).

Il documento a cui facciamo riferimento è quindi:

- *Sandwich_on_web_RAD_3.0*;

3. Panoramica del sistema

Il sistema SOW ha come obiettivo la creazione di un servizio di consegne a domicilio di panini.

Il suo sviluppo è stato concepito in maniera modulare e pertanto è stato possibile dividerlo in sottosistemi (si veda System Design Document); questo approccio apporta vantaggi dal punto di vista della manutenzione e sarà fondamentale nella fase di testing, in quanto si potrà individuare un errore nel singolo sottosistema effettuando un numero limitato di test.

Una volta testati individualmente tutti i sottosistemi andremo a testare la loro integrazione utilizzando un approccio di tipo “Bottom-up”. Il sottosistema a livello gerarchico più basso sarà testato individualmente. Successivamente, sarà testata la sua integrazione con la componente a livello di gerarchia più alto. Questo procedimento sarà iterato finché non avremo integrato tutti i sottosistemi per verificare che l'intero sistema funzioni correttamente.

4. Funzionalità da testare e non

Le componenti prese in considerazione nella fase di testing rappresentano le funzionalità core di SOW, ovvero:

● Gestione Account

Questa funzionalità permette ad un utente di registrarsi, di effettuare l'accesso, di modificare i dati e di fare il logout dal sistema; permette agli admin, invece, di autenticarsi ed accedere alle funzionalità a loro consentite.

- Saranno testate le funzionalità, lato utente, di registrazione, del login e della modifica dati.

● Gestione Menù

Tale gestione consente di visualizzare, inserire, modificare ed eliminare dati relativi ai menù, di aggiungere un menù a quick menù e di acquistarlo.

- Sarà testata la funzionalità di inserimento delle quantità da parte di un utente di un determinato menù.

● Gestione Carrello

Tale gestione consente di visualizzare, inserire ed eliminare dati relativi ai menù presenti nel carrello e di completare l'ordine.

- Saranno testate le funzionalità di inserimento di un nuovo indirizzo e dell'orario di consegna.

● Gestione Recensione

Tale gestione consente di inserire e visualizzare le recensioni.

- Saranno testate le funzionalità di inserimento di una recensione.

Nonostante anche Gestione ordini (lato utente) e Gestione Quick Menù siano sottosistemi core non saranno effettuati test poichè non presentano campi di input.

5. Pass/Fail criteria

Il testing ha successo se l'output osservato è diverso dall'output atteso: ciò significa che la fase di testing avrà successo se individuerà una failure. In tal caso questa verrà analizzata e, se legata ad un fault, si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica non abbia impattato su altri componenti del sistema.

Al contrario, il testing fallirà se l'output osservato sarà uguale all'oracolo.

6. Approccio

Nella sessione di testing di SOW verrà utilizzato un approccio di tipo "BLACK BOX", che prevede che i test vengano effettuati in maniera da non scendere nei dettagli del codice, ma basandosi sulle specifiche delle funzionalità da testare.

L'approccio alla fase di testing si compone di tre fasi:

- Testing di unità, che controlla i singoli componenti (classi, metodi)
- Testing di integrazione, che va a testare l'integrazione dei vari sottosistemi
- Testing di sistema: test funzionale, che andrà a verificare la funzionalità dell'intero sistema assemblato.

6.1 Testing di unità

Con il testing di unità verrà effettuato un controllo delle varie classi e metodi del sistema, quindi saranno ricercate le condizioni di fallimento andando ad evidenziare gli errori. Il testing di unità, sarà eseguito dal team di sviluppo attraverso l'implementazione di classi di test utilizzando il framework PHPUnit. In particolare, per ogni classe che esegue operazioni complesse sarà sviluppata la relativa classe PHPUnit.

Segue la lista delle classi per le quali saranno sviluppate classi di test:

- Account.php
- Carrello.php
- Componente.php
- Gestore_Account_Control.php
- Gestore_Carrello_Control.php
- Gestore_Menu_Control.php
- Gestore_Ordine_Control.php
- Gestore_Recensione_Control.php
- Menu.php
- Menu_Scelto.php

- Ordine.php
- Recensione.php

6.2 Testing di integrazione

Con il testing di integrazione si effettuerà un controllo sull'integrazione delle varie componenti del sistema. Si adotterà una strategia di tipo "Bottom-up". Per effettuare questi test di integrazione, spesso saranno necessari l'utilizzo di driver dato che tale strategia va ad integrare passo passo i sottosistemi partendo dal layer che si trova più in basso nella scala gerarchica. La lista dei casi di test verrà fuori dall'applicazione del metodo del Category Partition (sezione 7).

6.3 Testing di sistema

Con il testing di sistema verrà effettuato un controllo della correttezza dell'intero sistema. E' da considerare il testing più critico, in quanto può risultare molto complesso andare alla ricerca di eventuali errori, essendo impegnati tutti i sottosistemi. Questo test sarà effettuato utilizzando il framework Selenium, che mette a disposizione strumenti per il controllo di sistemi web-based.

7. 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).

8. Specifica dei test cases

8.1 Gestione Account

8.1.1 Login

Parametro → e-mail	
formato: [A-Za-z0-9]@[A-Za-z.]	
Categorie	Scelte

lunghezza le	1: lunghezza < 15 [errore]
	2: lunghezza 15-25 [property lunghezzaLEok ,lunghezza e-mail da 15 a 25]
	3: lunghezza > 25 [errore]
formato fe	1: rispecchia il formato [if lunghezzaLEok][property formatoFEok ,rispecchia il formato [A-Za-z0-9]@[A-Za-z.]]
	2: non rispetta il formato [if lunghezzaLEok][errore]
esiste ee	1: esiste nel DB [if lunghezzaLEok and formatoFEok]
	2: non esiste nel DB [if lunghezzaLEok and formatoFEok][errore]
Parametro → Password	
Categorie	Scelte
lunghezza lp	1: lunghezza < 6 [errore]
	2: lunghezza 6-15[property lunghezzaLPok ,lunghezza password da 6 a 15]
	3: lunghezza > 15 [errore]
presenza pp	1: dato presente [if lunghezzaLEok and lunghezzaLPok]
	2: dato non presente [if lunghezzaLEok and lunghezzaLPok][errore]

CODICE	COMBINAZIONE	ESITO
TC_Login_01	le2fe1ee1p2pp1	accesso
TC_Login_02	le2fe1ee2	errore
TC_Login_03	le2fe1ee1p2pp2	errore
TC_Login_04	le2fe1ee2lp2pp2	errore

8.1.2- Registra Utente

Parametro → Nome	
formato: [A-Za-z]	
Categorie	Scelte
lunghezza ln	1: lunghezza < 2 [errore]
	2: lunghezza 2-20 [property lunghezzaLNok , lunghezza nome da 2 a 20]
	3: lunghezza > 20 [errore]
formato fn	1: rispetta il formato [if lunghezzaLNok] [property formatoFNok , rispetta il formato [A-Za-z]]
	2: non rispetta il formato [if lunghezzaLNok] [errore]
Parametro → Cognome	
formato: [A-Za-z]	
Categorie	Scelte
lunghezza lc	1: lunghezza < 2 [errore]
	2: lunghezza 2-20 [property lunghezzaLCok , lunghezza cognome da 2 a 20]
	3: lunghezza > 20 [errore]
formato fc	1: rispetta il formato [if lunghezzaLCok] [property formatoFCok , rispetta il formato [A-Za-z]]
	2: non rispetta il formato [if lunghezzaLCok] [errore]
Parametro → e-mail	
formato: [A-Za-z0-9]@[A-Za-z.]	
Categorie	Scelte
lunghezza le	1: lunghezza < 15 [errore]
	2: lunghezza 15-25 [property lunghezzaLEok , lunghezza e-mail da 15 a 25]
	3: lunghezza > 25 [errore]

formato fe	1: rispecchia il formato [if lunghezzaLEok][property formatoFEok ,rispecchia il formato [A-Za-z0-9]@[A-Za-z.]]
	2: non rispetta il formato [if lunghezzaLEok][errore]
esiste ee	1: esiste nel DB [if lunghezzaLEok and formatoFEok][errore]
	2.non esiste nel DB [if lunghezzaLEok and formatoFEok][property nonEsisteEok]
Parametro → Password	
Categorie	Scelte
lunghezza lp	1: lunghezza < 6 [errore]
	2: lunghezza 6-15[property lunghezzaLPok ,lunghezza password da 6 a 15]
	3: lunghezza > 15 [errore]
Parametro → Indirizzo	
Categorie	Scelte
lunghezza li	1: lunghezza < 7 [errore]
	2: lunghezza 7-30 [property lunghezzaLIok ,lunghezza indirizzo da 7 a 30]
	3: lunghezza > 30 [errore]
Parametro → CAP	
formato [0-9]	
Categorie	Scelte
lunghezza lca	1: lunghezza < 5 [errore]
	2: lunghezza = 5 [property lunghezzaLCAok , lunghezza CAP uguale a 5]
	3: lunghezza > 5 [errore]
formato fca	1: rispecchia il formato [if lunghezzaLCAok][property

	formatoFCAok ,rispecchia il formato [0-9]]
	2: non rispetta il formato [if lunghezzaLCAok][errore]
Parametro → Numero Telefono	
formato [0-9]	
Categorie	Scelte
lunghezza lnt	1: lunghezza < 10 [errore]
	2: lunghezza = 10 [property lunghezzaLNTok , lunghezza Numero Telefono uguale a 10]
	3: lunghezza > 10 [errore]
formato fnt	1: rispecchia il formato [if lunghezzaLNTok][property formatoFNTok ,rispecchia il formato [0-9]]
	2: non rispetta il formato [if lunghezzaLNTok][errore]

CODICE	COMBINAZIONE	ESITO
TC_Registrazione_01	ln2fn1lc2fc1le2fe1ee2lp2li2lca2fca1lnt2fnt1	registrazione
TC_Registrazione_02	ln2fn1lc2fc1le2fe2	errore
TC_Registrazione_03	ln2fn1lc2fc1le2fe1ee2lp1	errore
TC_Registrazione_04	ln2fn1lc2fc1le2fe1ee1	errore
TC_Registrazione_05	ln1	errore
TC_Registrazione_06	ln2fn1lc1	errore
TC_Registrazione_07	ln2fn1lc2fc1le2fe1ee2lp2li1	errore
TC_Registrazione_08	ln2fn1lc2fc1le2fe1ee2lp2li2lca1	errore

TC_Registrazione_09	ln2fn1lc2fc1le2fe1ee2lp2li2lca2fca1lnt1	errore
TC_Registrazione_10	ln2fn1lc2fc1le2fe1ee2lp3	errore
TC_Registrazione_11	ln3	errore
TC_Registrazione_12	ln2fn1lc3	errore
TC_Registrazione_13	ln2fn1lc2fc1le2fe1ee2lp2li3	errore
TC_Registrazione_14	ln2fn1lc2fc1le2fe1ee2lp2li2lca3	errore
TC_Registrazione_15	ln2fn1lc2fc1le2fe1ee2lp2li2lca2fca1lnt3	errore
TC_Registrazione_16	ln2fn2	errore
TC_Registrazione_17	ln2fn2lc2fc2	errore
TC_Registrazione_18	ln2fn1lc2fc1le2fe1ee2lp2li2lca2fca1lnt2fnt2	errore
TC_Registrazione_19	ln2fn1lc2fc1le1	errore
TC_Registrazione_20	ln2fn1lc2fc1le3	errore
TC_Registrazione_21	ln2fn1lc2fc1le2fe1ee2lp2li2lca2fca2	errore

8.1.3 Modifica dati utente

Parametro → Password	
Categorie	Scelte
lunghezza lp	1: lunghezza < 6 [errore]

	2: lunghezza 6-15[property lunghezzaLPok ,lunghezza password da 6 a 15]
	3: lunghezza > 15 [errore]
Parametro → Indirizzo	
Categorie	Scelte
lunghezza li	1: lunghezza < 7 [errore]
	2: lunghezza 7-30 [property lunghezzaLIok ,lunghezza indirizzo da 7 a 30]
	3: lunghezza > 30 [errore]
Parametro → CAP	
formato [0-9]	
Categorie	Scelte
lunghezza lca	1: lunghezza < 5 [errore]
	2: lunghezza = 5 [property lunghezzaLCAok , lunghezza CAP uguale a 5]
	3: lunghezza > 5 [errore]
formato fca	1: rispecchia il formato [if lunghezzaLCAok][property formatoFCAok ,rispecchia il formato [0-9]]
	2: non rispetta il formato [if lunghezzaLCAok][errore]
Parametro → Numero Telefono	
Formato [0-9]	
Categorie	Scelte
lunghezza lnt	1: lunghezza < 10 [errore]
	2: lunghezza = 10 [property lunghezzaLNTok , lunghezza Numero Telefono uguale a 10]
	3: lunghezza > 10 [errore]

formato fnt	1: rispecchia il formato [if lunghezzaLNTok][property formatoFNTok ,rispecchia il formato [0-9]]
	2: non rispetta il formato [if lunghezzaLNTok][errore]

CODICE	COMBINAZIONE	ESITO
TC_Modifica_dati_01	lp2li2lca2fca1lnt2fnt1	modificati
TC_Modifica_dati_02	lp1	errore
TC_Modifica_dati_03	lp2li1	errore
TC_Modifica_dati_04	lp2li2lca1	errore
TC_Modifica_dati_05	lp2li2lca2fca1lnt1	errore
TC_Modifica_dati_06	lp3	errore
TC_Modifica_dati_07	lp2li3	errore
TC_Modifica_dati_08	lp2li2lca3	errore
TC_Modifica_dati_09	lp2li2lca2fca1lnt3	errore
TC_Modifica_dati_10	lp2li2lca2fca1lnt2fnt2	errore
TC_Modifica_dati_11	lp2li2lca2fca2	errore

8.2 Gestione Recensioni

8.2.1 Inserisce recensione

Parametro → Nickname	
Formato [A-Za-z]	
Categorie	Scelte
lunghezza lni	1: lunghezza < 2 [errore]
	2: lunghezza 2-15[property lunghezzaLNIok ,lunghezza

	nickname da 2 a 15]
	3: lunghezza >15 [errore]
formato fni	1: rispecchia il formato [if lunghezzaLNIok][property formatoFNIok ,rispecchia il formato [0-9]]
	2: non rispetta il formato [if lunghezzaLNIok][errore]
esiste eni	1: esiste nel DB [if lunghezzaLNIok and formatoFNIok][errore]
	2: non esiste nel DB [if lunghezzaLNIok and formatoFNIok]
Parametro → Titolo	
Formato [A-Za-z]	
Categorie	Scelte
lunghezza lti	1: lunghezza < 10 [errore]
	2: lunghezza 10-30[property lunghezzaLTIok ,lunghezza titolo da 10 a 30]
	3: lunghezza > 30 [errore]
formato fti	1: rispecchia il formato [if lunghezzaLTIok][property formatoFTIok ,rispecchia il formato [0-9]]
	2: non rispetta il formato [if lunghezzaLTIok][errore]
Parametro → Testo	
Categorie	Scelte
lunghezza lte	1: lunghezza < 50 [errore]
	2: lunghezza 50-300 [property lunghezzaLTEok ,lunghezza testo da 50 a 300]
	3: lunghezza >300 [errore]

CODICE	COMBINAZIONE	ESITO
---------------	---------------------	--------------

TC_Inserisce_Recensione_01	lni2fni1eni21lti2fti1lte2	inserita
TC_Inserisce_Recensione_02	lni2fni1eni2lti1	errore
TC_Inserisce_Recensione_03	lni1	errore
TC_Inserisce_Recensione_04	lni2fni1eni2lti2fti1lte1	errore
TC_Inserisce_Recensione_05	lni2fni1eni2lti3	errore
TC_Inserisce_Recensione_06	lni1	errore
TC_Inserisce_Recensione_07	lni3	errore
TC_Inserisce_Recensione_08	lni2fni1eni2lti2fti1lte3	errore
TC_Inserisce_Recensione_09	lni2fni1eni2lti2fti2	errore
TC_Inserisce_Recensione_10	lni2fni2	errore
TC_Inserisce_Recensione_11	lni2fni2eni1	errore
TC_Inserisce_Recensione_12	lni2fni1eni21lti1	errore
TC_Inserisce_Recensione_13	lni2fni1eni21lti2fti1lte1	errore

8.3 Gestione Carrello

8.3.1 Completa ordine

Parametro → Indirizzo	
Categorie	Scelte
lunghezza li	1: lunghezza < 7 [errore]
	2: lunghezza 7-30 [property lunghezzaLiok , lunghezza indirizzo da 7 a 30]
	3: lunghezza > 30 [errore]
Parametro → Orario	
Formato [0-9]	
Categorie	

lunghezza lo	1: lunghezza < 2 [errore]
	2: lunghezza = 2 [property lunghezzaLook , lunghezza ora uguale a 2]
	3: lunghezza > 2 [errore]
formato fo	1: rispecchia il formato [if lunghezzaLook] [property formatoFOok , rispecchia il formato [0-9]]
	2: non rispetta il formato [if lunghezzaLook] [errore]

CODICE	COMBINAZIONE	ESITO
TC_Check_Out_01	li2lo2fo1	ordinato
TC_Check_Out_02	li2lo3	errore
TC_Check_Out_03	li2lo2fo1	ordinato
TC_Check_Out_04	li1	errore
TC_Check_Out_05	li2lo3	errore
TC_Check_Out_06	li2lo2fo2	errore
TC_Check_Out_07	li2lo1	errore
TC_Check_Out_08	li2lo1	errore

8.4 Gestione Menù

8.4.1 Acquista menù

Parametro → Quantità	
Formato [0-9]	
Categorie	Scelte
lunghezza lq	1: lunghezza < 1 [errore]

	2: lunghezza 1-2 [property lunghezzaLQok ,lunghezza quantità da 1 a 2]
	3: lunghezza > 2[errore]
formato fq	1: rispecchia il formato [if lunghezzaLQok][property formatoFQok ,rispecchia il formato [0-9]]
	2: non rispetta il formato [if lunghezzaLQok][errore]

CODICE	COMBINAZIONE	ESITO
TC_Aggiunge_Al_Carrello_01	lq2fq1	aggiunto
TC_Aggiunge_Al_Carrello_02	lq1	errore
TC_Aggiunge_Al_Carrello_03	lq3	errore
TC_Aggiunge_Al_Carrello_04	lq2fq2	errore