DigitalMenù

TEST PLAN

Presentato da:
Antonio Della Monica
Alberto Cuomo
Gerardo De Lorenzo

Sommario

1. INTRODUZIONE

Lo scopo di questo documento è l'analisi e la gestione dell'attività di testing riguardante il sistema DigitalMenù. Viene verificato il corretto funzionamento del sistema analizzando i singoli casi presi dai test, e comparando i loro output con il presunto oracolo dei risultati. Da questa comparazione sarà possibile riscontrare gli eventuali errori e problemi da dover correggere con delle opportune modifiche. Il processo di testing viene iterato fino a che non si soddisfano in maniera accettabile i requisiti funzionali e non funzionali del sistema.

2. RELAZIONI CON ALTRI DOCUMENTI

Per l'individuazione corretta dei test casi si è fatto riferimento ai documenti precedentemente stilati. Infatti, la fase di testing è legata alle fasi ad essa precedenti; ogni documento sarà un punto di partenza indispensabile per poter effettuare un testing corretto ed adeguato e per verificare che il sistema desiderato sia simile a quello proposto.

2.1.Relazioni con il Requirements Analysis Document (RAD)

Al fine di verificare la correttezza di DigitalMenù sono stati predisposti dei test basati sulle funzionalità individuate nel RAD, in particolare riguardanti i requisiti funzionali e non funzionali del sistema

2.2. Relazioni con il System Design Document (SDD)

Al fine di verificare la tracciabilità di DigitalMenù, sono stati predisposti dei test sulle singole decomposizioni del sistema individuate nel SDD.

2.3.Relazioni con l'Object Design Document (ODD)

Al fine di verificare le performance di DigitalMenù, sono stati predisposti dei test per la verificabilità del funzionamento delle interfacce specificate nell' ODD

3. PANORAMICA DEL SISTEMA

4. FUNZIONALITA' DA TESTARE E NON

REQUISITI FUNZIONALI CLIENTE: (TESTATO)

- deve poter aggiungere un prodotto al carrello
- deve poter rimuovere un prodotto dal carrello
- deve poter scegliere di rimuovere alcuni ingredienti del prodotto
- deve poter visualizzare il menu
- deve poter visualizzare il carrello
- deve poter visualizzare il conto
- deve poter pagare: con carta di credito o in contanti

REQUISITI FUNZIONALI AMMINISTRATORE: (NON TESTATO)

- deve poter aggiungere un prodotto
- deve poter eliminare un prodotto
- deve poter modificare un prodotto
- deve poter aggiungere una categoria
- deve poter eliminare una categoria
- deve poter modificare una categoria

REQUISITI FUNZIONALI ADDETTO ALLA CUCINA: (TESTATO)

- deve poter visualizzare gli ordini pagati
- deve poter finalizzare gli ordini

REQUISITI FUNZIONALI CASSIERE: (NON TESTATO)

- deve poter confermare il pagamento in contanti
- deve poter prendere un'ordinazione per il cliente
- deve poter eliminare un ordine che non è in preparazione
- deve poter effettuare un rimborso
- deve poter visualizzare l'elenco di tutti gli ordini

5. PASS/FALL CRITERIA

Lo scopo del testing è quello di dimostrare la presenza di faults (errori) all'interno del sistema. Le attività di testing, infatti, saranno mirate all'identificazione di questi faults
e ad un successivo intervento per eliminarne la presenza.
Un test avrà successo se, dato un input al sistema, l'output osservato sarà diverso dall'output atteso dall'oracolo, quindi è stata identificata una failure. In tal caso questa verrà analizzata e si procederà alla correzione.
Al contrario, un test fallirà quando l'output osservato sarà uguale a quello presente nell'oracolo.
Per oracolo si intende il risultato atteso, in base ai requisiti, dall'esecuzione di un caso di test.

Il testing riterrà il sistema funzionante quando la coverage di testing avrà raggiunto il 75% per la verifica delle funzionalità di controllo del sistema.

6. APPROCCIO

Per il sistema DigitalMenù, il testing si compone tre fasi. Nella prima fase,verranno eseguiti i test di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità andando a constatare il corretto funzionamento. Questa fase verrà effettuata al completamento di ogni unità realizzata per poter individuare tempestivamente gli errori presenti nel codice.
Nella seconda fase, verrà effettuato il testing di integrazione in cui si andrà a testare l'integrazione dei vari sottosistemi.
Infine, verrà eseguito il testing di sistema che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti.
Nella sessione di testing del sistema verrà utilizzato un approccio "BLACK BOX" ed una strategia top down: tale strategia prevede che i sottosistemi nel layer più in alto della gerarchia vengono testati individualmente, successivamente vengono integrati i layer più in basso. Si ripete questo passo finché tutti i sottosistemi sono testati. Tale metodologia fa uso dei test stub. Un test stub è una implementazione parziale di una componente chiamata dall'unità testata.

Questo processo sarà dunque scomposto in 3 fasi:

6.1 Testing di unità

Per effettuare il testing di ogni singola componente del sistema DigitalMenù,verrà utilizzata la tecnica "Black-box testing" attraverso il framework JUnit. In questa fase, saranno analizzate le funzionalità dell'applicazione ed il comportamento delle singole componenti senza tener conto della loro struttura interna. Al fine di minimizzare il numero di test case, agli input verranno individuati dei vincoli a cui saranno associate delle scelte specifiche. I risultati del testing verranno analizzati e usati per correggere gli errori che causano il fallimento del sistema. Se si verifica un errore con dei risultati inattesi, si interviene in maniera tempestiva sulla
componente in modo da renderla correttamente funzionante e procedere con le fasi di testing successive.

6.2 Testing di integrazione

Il Testing di Integrazione è una delle fasi di testing più importanti. Essa consiste nel trovare fault integrando differenti componenti insieme, fault che non sono stati trovati nello "Unit testing". Per verificare la corretta integrazione dei sottosistemi del sistema sono stati predisposti dei test case basati sulla tecnica bottom-up, testando ogni modulo a livelli inferiori con i moduli principali fino a quando non abbiamo testato tutti i moduli.

6.3 Testing di sistema

Lo scopo principale di questa fase di testing è quello di dimostrare che l'intero sistema funzioni correttamente e che soddisfi effettivamente i requisiti funzionali e non

funzionali descritti nel documento di analisi dei requisiti (RAD). È da considerare l'attività più critica, in quanto può risultare molto complesso andare alla ricerca di eventuali errori, essendo impegnati tutti i sottosistemi. Inoltre, abbiamo

6.4 Criteri di successo

Il testing ha successo quando il comportamento osservato dal sistema implementato è diverso comportamento atteso specificato attraverso il modello del sistema. Avremo, quindi, un successo se il test individuerà una failure. In tal caso, questa verrà analizzata e si procederà alla sua eventuale correzione, dopo una correzione la fase di testing verrà reiterata per verificare che tale modifica non abbia impattato su altre componenti del sistema. Se il test non riesce ad individuare un errore parliamo di fallimento.

7. Strumenti di testing

Le risorse che vengono utilizzate per individuare i casi di testing sono i documenti di progetto RAD, SDD ed ODD.

8. TEST CASES

TC_UC_C_1: Visualizzazione prodotti di una categoria

Parametro:idCat **Formato:[0-9]**	
Categoria	Scelte
Valore[VA_idC]	1. idCat>0 [VA_idC_OK] 2. idCat<0 [error]
Esistenza[ES_idC]	1.idCat della categoria presente nel database [ES_idC_OK] 2.IdCat della categoria non presente nel database [error]

Codice	Combinazione	Esito
TC_UC_C_1_1	[VA_idC_2]	[error]
TC_UC_C_1_2	[VA_idC_1] [ES_idC_2]	[error]
TC_UC_C_1_3	[VA_idC_1] [ES_idC_1]	[OK]

TC_UC_C_2: Visualizzazione dettagli prodotto

Parametro:idPro **Formato:[0-9]**	
Categoria	Scelte
Valore[VA_idPro]	1. idPro>0 [VA_idPro_OK] 2. idPro<0 [error]
Esistenza[ES_idPro]	1.idPro del prodotto presente nel database [ES_idPro_OK] 2.idPro del prodotto non presente nel database [error]

Codice	Combinazione	Esito
TC_UC_C_2_1	[VA_idPro_2]	[error]
TC_UC_C_2_2	[VA_idPro_1] [ER_idPro_2]	[error]
TC_UC_C_2_3	[VA_idPro_1] [ER_idPro_1]	[OK]

TC_UC_C_6: Conferma Ordine

Parametro:idO **Formato:[0-9]**	
Categoria	Scelte
Valore[VA_idO]	1. idO>0 [VA_idO_OK] 2. idO<0 [error]
Esistenza[ES_idO]	1. idO dell'ordine non presente nel database [ES_idO_OK] 2. idO dell'ordine presente nel database [error]

Parametro:stato **Formato:[0-1]**	
Categoria	Scelte
Valore [VA_S]	1. stato>=0 AND stato<=1 [VA_S_OK] 2. stato<0 OR stato>1 [error]

Parametro:DetOrdine **Formato:[List<DettagliOrdineBean>]**	
Categoria	Scelte
Valore [VA_DO]	1. DetOrdine!=null [VA_DO_OK] 2. DetOrdine==null [error]

Parametro:DetOrdine.IdP **Formato:[0-9]**	
Categoria	Scelte
Valore [VA_idP]	1. DetOrdine.idP>0 [VA_idP_OK] 2. DetOrdine.idP<0 [error]
Esistenza[ES_idP]	1. DetOrdine.idP del prodotto è presente nel database [ES_idP_OK] 2. DetOrdine.idP del prodotto non è presente nel database [error]

Parametro:DetOrdine.quantita **Formato:[0-9]**	
Categoria	Scelte
Valore[VA_Q]	1. DetOrdine.quantita>0 [VA_Q_OK] 2. DetOrdine.quantita<1 [error]

Parametro:DetOrdine.DeleteIng **Formato:[A-Z a-z]**	
Categoria	Scelte
Formato[FR_DI]	1. DetOrdine.DeleteIng [a-z A-Z] [FR_DI_OK] 2. DetOrdine.DeleteIng [0-9] [error]
Esistenza[ES_DI]	1. DetOrdine.DeleteIng dell'ingrediente è presente nel database [ES_DI_OK] 2. DetOrdine.DeleteIng dell'ingrediente non è presente nel database [error]

Codice	Combinazione	Esito
TC_UC_C_6_1	[VA_idO_2]	[error]
TC_UC_C_6_2	[VA_idO_1] [ER_idO_2]	[error]

TC_UC_C_6_3	[VA_idO_1] [ER_idO_1] [VA_S_2]	[error]
TC_UC_C_6_4	[VA_idO_1] [ER_idO_1] [VA_S_1] [VA_DO_2]	[error]
TC_UC_C_6_5	[VA_idO_1] [ER_idO_1] [VA_S_1] [VA_DO_1] [VA_idP_2]	[error]
TC_UC_C_6_6	[VA_idO_1] [ER_idO_1] [VA_S_1] [VA_DO_1] [VA_idP_1] [ER_idP_2]	[error]
TC_UC_C_6_7	[VA_idO_1] [ER_idO_1] [VA_S_1] [VA_DO_1] [VA_idP_1] [ER_idP_1] [VA_Q_2]	[error]
TC_UC_C_6_8	[VA_idO_1] [ER_idO_1] [VA_S_1] [VA_DO_1] [VA_idP_1] [ER_idP_1] [VA_Q_1] [FR_DI_2]	[error]
TC_UC_C_6_9	[VA_idO_1] [ER_idO_1] [VA_S_1] [VA_DO_1] [VA_idP_1] [ER_idP_1] [VA_Q_1] [FR_DI_1] [ES_DI_2]	[error]
TC_UC_C_6_10	[VA_idO_1] [ER_idO_1] [VA_S_1] [VA_DO_1] [VA_idP_1] [ER_idP_1] [VA_Q_1] [FR_DI_1] [ES_DI_1]	[OK]

TC_UC_AC_1 Visualizzazione ordini pagati

Parametro:idO **Formato:[0-9]**	
Categoria	Scelte
Valore[VA_idO]	1. idO>0 [VA_idO_OK] 2. idO<0 [error]
Esistenza[ES_idO]	1. idO dell'ordine presente nel database [ES_idO_OK] 2. idO dell'ordine non presente nel database [error]

Parametro:stato **Formato:[0-1]**	
Categoria	Scelte
Valore [VA_S]	1. stato> 0 AND stato<=4 [VA_S_OK] 2. stato<= 0 OR stato>4 [error]

Codice	Combinazione	Esito
TC_UC_AC_1_1	[VA_idO_2]	[error]

TC_UC_AC_1_2	[VA_idO_1] [ES_idO_2]	[error]
TC_UC_AC_1_3	[VA_idO_1] [ES_idO_2] [VA_S_2]	[error]
TC_UC_AC_1_4	[VA_idO_1] [ES_idO_2] [VA_S_1]	[OK]

TC_UC_AC_2 Cambio stato dell'ordine

Parametro:idO **Formato:[0-9]**	
Categoria	Scelte
Valore[VA_idO]	1. idO>0 [VA_OK] 2. idO<0 [error]
Esistenza[ES_idO]	1. idO dell'ordine non presente nel database [ES_OK] 2. idO dell'ordine presente nel database [error]

Parametro:stato **Formato:[0-1]**	
Categoria	Scelte
Valore [VA_S]	1. stato>0 AND stato<3 AND this.stato == stato+1 [VA_OK] 2. stato<=0 OR stato>4 OR this.stato != stato+1 [error]

Codice	Combinazione	Esito
TC_UC_AC_2_1	[VA_idO_2]	[error]
TC_UC_AC_2_2	[VA_idO_1] [ES_idO_1]	[error]
TC_UC_AC_2_3	[VA_idO_1] [ES_idO_1] [VA_S_2]	[error]
TC_UC_AC_2_4	[VA_idO_1] [ES_idO_1] [VA_S_1]	[OK]