



TEST PLAN

SOMMARIO

1. Introduzione.....	3
2. Relazione con altri documenti.....	3
2.1 Relazioni con il documento di analisi dei requisiti (RAD)	3
2.2 Relazioni con il System Design Document (SDD)	3
2.3 Relazioni con l'Object Design Document (ODD)	3
3. Panoramica del sistema	4
4. Funzionalità da testare/non testare	4
5. Criteri di successo/insuccesso	4
6. Approccio.....	5
6.1 Testing di unità	5
6.2 Testing di integrazione	5
6.3 Testing di sistema	5
7. Sospensione e ripresa.....	5
7.2 Criteri di ripresa	5
8. Materiali di prova (requisiti hardware/software)	5
9. Casi di test.....	6
9.1 CP_TM: Titolare Manager	6
TC_TM_1: aggiungiPiatto	6
TC_TM_2: aggiungiIngrediente	7
TC_TM_3: popolaritaPiattiMensile	8
9.2 CP_OM: Ordine Manager	9
TC_OM_1: salvaConto	9
TC_OM_2: addOrdine.....	10
TC_OM_3: addRichiestaConto.....	11

1. INTRODUZIONE

Lo scopo di questo documento è quello di gestire lo sviluppo e le attività di test riguardanti il sistema software MenùMaxi. Saranno identificati: gli elementi da testare, le funzionalità da testare, le strategie di testing usate e gli strumenti utilizzati, il personale responsabile dei test, le risorse e le attività richieste per completare i test e i rischi associati al piano. Lo scopo del testing è quello di rilevare errori in maniera pianificata all'interno del codice realizzato. Quindi, l'obiettivo consiste nell'evitare che gli errori si presentino durante l'utilizzo del sistema dell'utente finale. I risultati prodotti dai test saranno utilizzati per comprendere dove intervenire per correggere gli errori o apportare modifiche per migliorare il sistema.

In questo documento verranno analizzate, in particolar modo, le seguenti attività:

- Titolare Manager
- Ordine Manager

Per ogni attività saranno testate solo le funzionalità contenente i requisiti con priorità **Medio/Alta**.

2. RELAZIONE CON ALTRI DOCUMENTI

Il Test Plan ha una stretta relazione con i documenti prodotti finora, dato che il sistema è stato pianificato nelle precedenti documentazioni. Per verificare il corretto funzionamento del sistema saranno usati i test cases individuati e documentati precedentemente nel processo di sviluppo del sistema. I test cases sono basati sulle funzionalità individuate nel documento di raccolta ed analisi dei requisiti (RAD).

2.1 RELAZIONI CON IL DOCUMENTO DI ANALISI DEI REQUISITI (RAD)

La relazione tra test plan e RAD (Requirement Analysis Document) riguarda in particolare i requisiti funzionali e non funzionali del sistema dato che i test saranno eseguiti su quelle funzionalità tenendo conto delle specifiche espresse nel documento precedente. In particolare, il RAD contiene lo scopo del sistema, l'ambito del sistema e gli obiettivi, evidenziando una panoramica di requisiti funzionali, requisiti non funzionali, scenari, casi d'uso, diagrammi e mock-up del sistema.

Il documento a cui si fa riferimento è:

- [RAD MENU MAXI](#)

2.2 RELAZIONI CON IL SYSTEM DESIGN DOCUMENT (SDD)

L'SDD (System Design Document) rappresenta l'architettura del sistema. In particolare, l'SDD contiene l'architettura del software proposto e i servizi dei sottosistemi.

Il documento a cui si fa riferimento è:

- [SSD MENU MAXI](#)

2.3 RELAZIONI CON L'OBJECT DESIGN DOCUMENT (ODD)

Il test si baserà sulle class Interfaces definite nell'ODD (Object Design Document). In particolare, l'ODD contiene i package e i class interfaces del sistema.

Il documento a cui si fa riferimento è:

-[ODD_MENUMAXI](#)

3. PANORAMICA DEL SISTEMA

Come analizzato e deciso nel System Design Document, la struttura del nostro sistema segue il pattern architetturale Client/Server basata sui servizi.

I sottosistemi individuati sono:

- AuthenticationManager
- TitolareManager
- OrdineManager
- MenuManager
- AndroidClientApp

4. FUNZIONALITÀ DA TESTARE/NON TESTARE

Abbiamo deciso di non testare i metodi get e set, e i metodi con priorità bassa.

Di seguito saranno riportati le funzionalità introdotte nel sistema che saranno sottoposte a test e suddivise in sottosistemi:

-Titolare Manager:

- aggiungiPiatto
- aggiungiIngrediente
- popolaritaPiattiMensile

-Ordine Manager:

- salvaConto
- addOrdine
- addRichiestaConto

5. CRITERI DI SUCCESSO/INSUCCESSO

I dati di input del test saranno raggruppati in insiemi dalle caratteristiche comuni in modo da effettuare un test su di un unico elemento rappresentativo.

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.

La failure quindi è qualsiasi variazione del comportamento del sistema rispetto al comportamento atteso specificato in analisi dei requisiti.

6. APPROCCIO

L'approccio alla fase di testing si compone di 3 fasi, la prima servirà a testare le componenti una ad una, poi si passerà a testare le funzionalità delle integrazioni dei vari sottosistemi, infine si testerà l'intero sistema assemblato per verificare soprattutto che esso soddisfi le richieste del cliente.

6.1 TESTING DI UNITÀ

Per realizzare il testing di ogni singola componente verrà utilizzata la tecnica "Black-Box testing".

Così facendo andremo ad esaminare le funzionalità dell'applicazione ed il comportamento input/output delle singole componenti senza tener conto della loro struttura interna. Essendo quasi impossibile generare tutti i possibili input, verrà partizionato l'input utilizzando la tecnica del Category Partition. I risultati del testing verranno analizzati e usati per correggere gli errori che causano il fallimento del sistema.

6.2 TESTING DI INTEGRAZIONE

Dopo aver sottoposto ogni componente al testing di unità, ed aver corretto gli eventuali errori scaturiti dal test, essi verranno integrati in sottosistemi più grandi per sottoporli ad un test di integrazione. Il testing verrà effettuato seguendo la strategia "Bottom-Up" che prevede l'integrazione dal basso verso l'alto.

6.3 TESTING DI SISTEMA

Prima di essere pronto all'uso, il sistema affronterà l'ultima fase di testing, quello di sistema, per dimostrare che siano soddisfatti tutti i requisiti richiesti. Lo scopo di questa fase è testare le funzionalità più importanti, usate maggiormente e con maggior probabilità di fallimento. Per la parte del sistema web-based verrà utilizzato, per effettuare il testing di sistema, il tool Selenium, che si occupa di simulare l'interazione con il sistema dal punto di vista dell'utente.

7. SOSPENSIONE E RIPRESA

7.1 CRITERI DI SOSPENSIONE

La fase di testing del sistema verrà sospesa quando si otterranno i risultati attesi in accordo con i tempi di sviluppo previsti, tenendo sempre conto dei costi dell'attività di testing.

Questo processo verrà quindi portato avanti quanto più possibile nel tempo senza però rischiare di ritardare la consegna finale del progetto.

7.2 CRITERI DI RIPRESA

La fase di testing potrà riprendere in seguito a modifiche o correzioni che generano errori o fallimenti, i test case verranno, quindi, sottoposti nuovamente al sistema assicurandosi così di aver risolto effettivamente il problema.

8. MATERIALI DI PROVA (REQUISITI HARDWARE/SOFTWARE)

Gli strumenti necessari per l'attività di test sono: un software MySQL per il database, un ide per java EE (per la parte server) e Android Studio (per la parte client).

9. CASI DI TEST

9.1 CP_TM: TITOLARE MANAGER

TC_TM_1: aggiungiPiatto

Parametro: piatto	
CATEGORIE	SCELTE
Esiste [pe]	1) Null [errore] 2) !=Null [property piattoEsiste]

Parametro: piatto.nome	
CATEGORIE	SCELTE
Lunghezza [ln]	1) Lunghezza > 50 [errore] 2) Lunghezza <= <u>50</u> [property nomeOK]

Parametro: piatto.foto	
CATEGORIE	SCELTE
Lunghezza [lf]	1) Lunghezza > 50 [errore] 2) Lunghezza <= 50 [property fotoOK]

Parametro: piatto.categoria	
CATEGORIE	SCELTE
Lunghezza [lc]	1) Lunghezza > 30 [errore] 2) Lunghezza <= 30 [property categoriaOK]

Parametro: piatto.prezzoCent Formato: >10	
CATEGORIE	SCELTE

Formato [fpc]	1) Non Rispetta formato [errore] 2) Rispetta formato [property prezzoCentOK]
----------------------	---

Codice	Combinazione	Esito
TC_TM_1.1	pe1	Errore
TC_TM_1.2	pe2, ln1	Errore
TC_TM_1.3	pe2, ln2, lf1	Errore
TC_TM_1.4	pe2, ln2, lf2, lc1	Errore
TC_TM_1.5	pe2, ln2, lf2, lc2, fpc1	Errore
TC_TM_1.6	pe2, ln2, lf2, lc2, fpc2	Successo

TC_TM_2: aggiungiIngrediente

Parametro: ingrediente	
CATEGORIE	SCELTE
Esiste [ie]	1) Null [errore] 2) !=Null [OK]

Parametro: ingrediente.nome	
CATEGORIE	SCELTE
Lunghezza [ln]	1) Lunghezza >50 [errore] 2) Lunghezza <=50 [property nomeOK]

Parametro: ingrediente.categoria	
CATEGORIE	SCELTE
Lunghezza [lc]	1) Lunghezza >30 [errore] 2) Lunghezza <=30 [property categoriaOK]

Parametro: ingrediente.sovrapprezzoCent Formato: >0	
CATEGORIE	SCELTE
Formato [fsc]	1) Non rispetta il formato [errore] 2) Rispetta il formato [property sovrapprezzoCentOK]

Codice	Combinazione	Esito
TC_TM_2.1	ie1	Errore
TC_TM_2.2	ie2, ln1	Errore
TC_TM_2.3	ie2, ln2, lc1	Errore
TC_TM_2.4	ie2, ln2, lc2, fsc1	Errore
TC_TM_2.5	ie2, ln2, lc2, fsc2	Successo

TC_TM_3: popolaritaPiattiMensile

Parametro: mese Variabile ambientale: mese_corrente Formato: 1-12	
CATEGORIE	SCELTE
Formato [fm]	1) Non rispetta il formato [errore] 2) Rispetta il formato [property formatoFmOK]

mese corrente [mc]	1) >mese_corrente [if formatoFmOK] 2) <=mese_corrente [if formatoFmOK]
---------------------------	---

Parametro: anno Variabile ambientale: anno_corrente Formato: 2019-anno_corrente	
CATEGORIE	SCELTE
Formato [fa]	1) Non rispetta il formato [errore] 2) Rispetta il formato [property formatoFaOK]
anno corrente [ac]	1) =anno_corrente [if formatoFaOK and formatoFmOK] 2) !=anno_corrente [if formatoFaOK and formatoFmOK]

Codice	Combinazione	Esito
TC_TM_3.1	fm1	Errore
TC_TM_3.2	fm2, mc1, fa1	Errore
TC_TM_3.3	fm2, mc1, fa2, ac1	Errore
TC_TM_3.4	fm2, mc1, fa2, ac2	Successo
TC_TM_3.5	fm2, mc2, fa2, ac1	Successo
TC_TM_3.6	fm2, mc2, fa2, ac2	Successo

9.2 CP_OM: ORDINE MANAGER

TC_OM_1: salvaConto

Parametro: tavolo	
CATEGORIE	SCELTE

Esiste [te]	1) =Null [errore] 2) !=Null [property tavoloEsiste]
Ha effettuato 1 o più ordini [eo]	1) No [errore] 2) Si

Codice	Combinazione	Esito
TC_OM_1.1	te1	Errore
TC_OM_1.2	te2, eo1	Errore
TC_OM_1.3	te2, eo2	Successo

TC_OM_2: addOrdine

Parametro: ordine	
CATEGORIE	SCELTE
Esiste [oe]	1) =Null [errore] 2) !=Null [property ordineEsiste]

Parametro: ordine.tavolo	
CATEGORIE	SCELTE
Esiste [te]	1) =Null [errore] 2) !=Null [property tavoloEsiste]

Parametro: ordine.piatti	
CATEGORIE	SCELTE
Esiste [pe]	1) =Null [errore] 2) !=Null [property piattiEsiste]

Contenuto [cp]	1) Vuoto [if piattiEsiste] [errore] 2) Non vuoto [if piattiEsiste]
-----------------------	---

Parametro: ordine.totaleCent Formato: >10	
CATEGORIE	SCELTE
Esiste [tce]	1) =Null [errore] 2) !=Null [property totaleCentEsiste]
Formato [ftc]	1) Non rispetta il formato [if totaleCentEsiste] [errore] 2) Rispetta il formato [if totaleCentEsiste]

Codice	Combinazione	Esito
TC_OM_2.1	oe1	Errore
TC_OM_2.2	oe2, te1	Errore
TC_OM_2.3	oe2, te2, pe1	Errore
TC_OM_2.4	oe2, te2, pe2, cp1	Errore
TC_OM_2.5	oe2, te2, pe2, cp2, tce1	Errore
TC_OM_2.6	oe2, te2, pe2, cp2, tce2, ftc1	Errore
TC_OM_2.7	oe2, te2, pe2, cp2, tce2, ftc2	Successo

TC_OM_3: addRichiestaConto

Parametro: conto	
CATEGORIE	SCELTE
Esiste [ce]	1) =Null [errore] 2) !=Null [property contoEsiste]

Parametro: conto.tavolo	
CATEGORIE	SCELTE
Esiste [te]	1) =Null [errore] 2) !=Null [property tavoloEsiste]
Il tavolo ha effettuato uno o più ordini [oe]	1) No [errore] 2) Si [if tavoloEsiste]

Codice	Combinazione	Esito
TC_OM_3.1	ce1	Errore
TC_OM_3.2	ce2, te1	Errore
TC_OM_3.3	ce2, te2, oe1	Errore
TC_OM_3.4	ce2, te2, oe2	Successo