Progetto **MYTHOS**

Test Plan

Partecipanti
Egidio Mario 0512105122
Pagliaro Vincenzo 0512105546

Project Manager
De Lucia Andrea
Pecorelli Fabiano

Sommario

1. Introduzione

Lo scopo di questo documento è quello di pianificare l'attività di test del sistema Mythos con lo scopo di verificare eventuali differenze tra comportamento atteso e comportamento osservato. In questa attività andremo a rilevare quanti più possibili errori prodotti dal nostro sistema, per evitare che questi si presentino nel momento in cui il sistema verrà utilizzato dall'utente finale.
Si noti, tuttavia, che di seguito saranno specificate anche le funzionalità che non verranno implementate nella prima versione del sistema, questo per facilitare la fase di testing delle versioni successive quando verranno implementate anche le funzionalità escluse nella prima versione del sistema.

2. Relazioni con altri documenti

Di seguito verranno riportate le relazioni tra il test plan e la documentazione precedente.

2.1. Relazioni con il documento di analisi dei requisiti (RAD)

La relazione tra test plan e RAD riguarda principalmente i requisiti funzionali del sistema e in piccola parte quelli non funzionali.
I requisiti funzionali poiché, per ognuno di essi, verranno progettati uno o più test i quali serviranno a testare ogni singola funzionalità.

2.2. Relazioni con il System Design Document (SDD)

Nel System Design Document il sistema è stato suddiviso in sottosistemi organizzati su un'architettura a tre livelli: Presentation Layer, Application Layer e Storage Layer. Il test dei vari componenti deve rimanere fedele a queste suddivisioni il più possibile.

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

Il test d'integrazione farà quanto più riferimento possibile alle interfacce delle classi definite nell'ODD. Questo perché per effettuare una parte della fase di test ci si deve basare specialmente sulla precondizione delle interfacce affinché si possano testare tutti i possibili casi (o per lo meno quelli per i quali c'è più probabilità di essere eseguiti).

3. Panoramica del sistema

Il sistema è stato suddiviso in sottosistemi più piccoli, in particolare è stato diviso per gestioni:

- Gestione Utenti
- Gestione Prenotazioni Tavoli
- Gestione Liste Clienti p.r.
- Gestione p.r.
- Gestione Magazzino
- Gestione Ordinazioni

Quasi ognuna di queste gestioni prevede principalmente operazioni di inserimento, modifica, cancellazione, visualizzazione e ricerca e saranno proprio queste funzionalità ad essere testate nel corso della fase di testing del sistema oltre a quelle che riguardano la restante parte di logica di business del sistema.

4. Funzionalità da testare

4.1. Gestione Utenti
- AutenticazioneLocale
- AutenticazioneOnline

4.2. Gestione Prenotazioni Tavoli
- ModificaCostoPrenotazioni
- ModificaPrenotazione
- NuovaPrenotazione

4.3. Gestione Liste Clienti p.r.
- InserimentoClienteInLista

4.4. Gestione p.r.
- InseirmentoNuovoPr

4.5. Gestione Magazzino
- ModificaProdotto
- InserimentoNuovoProdotto

4.6. Gestione Ordinazioni
- AggiungiProdottoAlCarrello

Si noti che la funzionalità "InserimentoNuovoProdotto" non verrà implementata nella prima versione del sistema, ma si è scelto comunque di progettare il category partition per questa funzionalità.

5. Criteri Pass/Failed

I dati di input del test saranno suddivisi in classi di equivalenza, ovvero verranno raggruppati in insiemi dalle caratteristiche comuni, per i quali sarà sufficiente testare un solo elemento rappresentativo. Un input avrà superato un test se l'output risultante sarà concettualmente uguale a quello atteso, cioè quello che è stato specificato nell'oracolo del category partition specification (il responsabile del testing conosce quale dovrebbe essere l'output corretto).

6. Approccio

Le tecniche di testing adottate riguarderanno inizialmente il testing di unità dei singoli componenti (DAO), in modo da testare nello specifico la correttezza di ciascuna unità. Seguirà il testing di integrazione(CONTROL), che focalizzerà l'attenzione principalmente sul test delle interfacce delle suddette unità. Infine, verrà eseguito il testing di sistema, che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti.

6.1. Testing di Unità

Durante questa fase, verranno ricercate le condizioni di fallimento, isolando i componenti ed usando test driver e stub, cioè implementazioni parziali di componenti che dipendono o da cui dipendono le componenti da testare.
Gli errori scovati in qualsiasi fase di testing, che comporteranno un fallimento del sistema dovranno essere tempestivamente corretti per poi ripristinare il testing al più presto.
In questa fase verranno testate tutte le classi DAO e le entity che hanno una particolare logica di controllo al suo interno.

6.2. Testing di Integrazione

In questa fase si procederà all'integrazione delle singole componenti che compongono una funzionalità le quali verranno testate nel complesso attraverso una strategia Bottom-Up. Si itera questo procedimento per tutte le funzionalità.
Quest'approccio mira principalmente a facilitare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi.
La strategia utilizzata per il testing si baserà esclusivamente sulla tecnica Black-Box, che si focalizza sul comportamento Input/Output, ignorando la struttura interna della componente. Al fine di minimizzare il numero di test cases, i possibili input verranno partizionati in categorie e ogni categoria sarà divisa ulteriormente in delle scelte. Tutte queste scelte verranno poi combinate tra loro affinché si possano ricoprire quanti più casi possibili (per quella singola funzionalità).

6.3. Testing di Sistema

Lo scopo di questa fase di testing è quello di dimostrare che il sistema soddisfi effettivamente i requisiti richiesti e sia, quindi, pronto all'uso. Come per il testing di unità, si cercherà di testare le funzionalità più importanti per l'utente e quelle maggiormente utilizzate.
Si noti che, come per il testing d'integrazione, si procederà attraverso tecnica Black-Box.

7. Sospensione e ripresa

7.1. Criteri di sospensione

La fase di testing del sistema (per un fattore legato ai ristretti tempi di consegna) verrà sospesa dai progettisti quando saranno testate almeno una volta tutte le funzionalità meno rilevanti, e testate maggiormente con più casi di test le funzionalità cruciali del sistema Mythos (es: nuovo ordine). Ulteriori test saranno effettuati direttamente dall'utente finale come da accordi.

7.2. Criteri di ripresa

In seguito alle modifiche o correzioni delle componenti che introdurranno errori o fallimenti, i test case verranno sottoposti nuovamente al sistema assicurandosi così di aver risolto effettivamente il problema.

8. Materiale per il testing

L'hardware necessario per l'attività di test è un pc con i seguenti requisiti:

- connessione ad Internet (per testare il sistema online)
- plugin per ambiente di sviluppo per il sistema in locale (in questo caso Eclipse): Mockito e JUnit
- framework per il testing del sistema online: PHPUnit
- plugin per il broswer: Selenium

9. Test Cases

Si rimanda al documento "Category Partition Specification.pdf".