

Università degli Studi di Salerno
Corso di Ingegneria del Software

MyHotel
STP
Versione 1.1



Revision History

Data	Versione	Descrizione	Autore
15/12/2016	1.0	Introduzione, Riferimenti, Test di Unità, Test di Integrazione	Alessandro Criscuolo
16/12/2016	1.1	System testing, Test Criteria, Test Environment	Marco Avagliano

Indice

1 INTRODUZIONE.....	3
1.1 RIFERIMENTI.....	3
2 TEST DI UNITA'.....	4
3 TEST DI INTEGRAZIONE.....	4
4 SYSTEM TESTING.....	4
5 TEST CRITERIA.....	5
6 TEST ENVIRONMENT.....	5

1. Introduzione

Il Testing rappresenta una delle più importanti tecniche per verificare la qualità del software, in quanto consente di analizzare, valutare e promuovere il miglioramento della correttezza dell'implementazione con riferimento alle caratteristiche definite in particolare dal modello dei requisiti software. Lo scopo di tale attività è quello di provare il sistema e rilevare problemi; il principale scopo, quindi, sarà quello di eseguire quanti più test possibili per rilevare un ampio numero di errori. E' importante definire i casi d'uso su cui verranno testate le funzionalità del sistema. Tale attività va in contrasto con quelle svolte in precedenza: analisi, design, implementazione sono attività costruttive mentre il testing tenta di "rompere il sistema".

Questo è il test plan document dell'applicazione web "MyHotel". L'obiettivo del test è quello di provare il corretto funzionamento dei metodi delle classi di controllo e manager del sistema.

Si utilizzerà la modalità di test "black box" in quanto, data la semplicità dell'applicazione, si è interessati unicamente al comportamento delle singole classi.

Si procederà in prima fase con il test di unità, successivamente verrà eseguito anche un test di integrazione per verificare la corretta interdipendenza delle varie classi.

1.1. Riferimenti

I test dovranno tenere conto delle specifiche espresse nel documento "RADMyHotel".

Naturalmente, si darà una maggiore rilevanza ai "casi limite", cioè tutti quei casi che sono al limite del nostro dominio di dati d'ingresso. Si darà rilevanza anche ai requisiti non funzionali ed ai vari modelli prodotti in fase di analisi dei requisiti.

Il testing dovrà garantire la coerenza tra il software e gli obiettivi di design definiti in fase di System Design, specificati nel documento "SDDMyHotel". Si analizzeranno, quindi, i conflitti e le inconsistenze presenti tra le componenti del sistema testate e tali obiettivi. Per verificare la corretta integrazione dei sottosistemi del sistema MyHotel sono stati predisposti dei test case basati sulla suddivisione in sottosistemi proposta in fase di System Design. Il documento che riporta questi test case è il TCSMyHotel.

La fase di testing dovrà considerare il contenuto del documento di Object Design, in quanto quest'ultimo rappresenta la base per la realizzazione dell'implementazione, fondamentale per il testing. Pertanto, sarà necessario effettuare il testing delle unità per individuare le differenze tra ciò che è stato stabilito in fase di Object Design ed il sistema effettivo. Il documento di riferimento per questa fase sarà "ODDMyHotel".

Per verificare la corretta integrazione dei sottosistemi del sistema MyHotel sono stati predisposti dei test case basati sulla suddivisione in sottosistemi proposta in fase di System Design. Il documento che riporta questi test case è il TCSMyHotel.

2.Test di unità

Sarà effettuato un testing per ogni unità del software individuata. Per ogni componente verranno costruiti uno o più test case, di tipo “Black Box”. I test case “Black Box” saranno svolti analizzando unicamente il comportamento di I/O delle unità in varie situazioni, senza esaminare il codice.

3.Test di integrazione

3.1 Approccio di Integration Testing

Dopo aver testato le singole unità del software, ognuna di queste ultime sarà integrata insieme ad altre unità che interagiscono direttamente con essa. Si effettuerà, quindi, un testing che riguarderà gruppi di unità.

Anche nella fase di integration testing verrà utilizzato un approccio “Black box”.

3.2 Componenti da testare

La scelta delle componenti da testare segue la decisione di eseguire la strategia di testing non incrementale.

Per quanto riguarda il Control/Model, le componenti con le relative APIs da testare sono:

- GestoreUtente;
- GestoreCamere;
- VisualizzatoreCamere;
- GestorePrenotazioni;
- GestoreCarrello.

4.System testing

Dopo aver effettuato il testing d'integrazione, si svolgerà il testing dell'intero sistema software con l'uso di molteplici test case. Si analizzeranno, in particolare, le differenze tra il comportamento effettivo software e quello previsto, dettato dal cliente e descritto nel Documento di Analisi dei Requisiti. Sarà svolto, quindi, un testing funzionale, che riguarderà l'insieme delle funzionalità del software implementato. Anche in tal caso si costruiranno test case di tipologia “Black Box”.

5.Test Criteria

5.1 Suspension Criteria

Sei membri del team riportano che il 20% dei tests sono falliti, il testing sarà sospeso fino a che il team di sviluppo non ripari tutti i casi che hanno causato il fallimento.

5.2 Exit Criteria

Specifica i criteri che denotano il completamento positivo di una fase di test :

- La Run rate deve essere obbligatoriamente 100% a meno che non si diano ragione chiare;
- Pass rate è 80%, il raggiungimento della pass rate è obbligatorio in ogni caso.

5.3 Pass/Fail Criteria

Il testing ha successo se l'output osservato è diverso dall'output atteso: ciò significa che parliamo di SUCCESSO se il test 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. Viceversa parliamo di FALLIMENTO se il test non riesce ad individuare un errore.

6.Test Environment

Il test delle funzionalità sarà eseguito con Selenium sul browser Firefox.