

**BoomToys**  
**TP - Test Plan**  
**Versione 1.0**



Data: 13/12/16

**Partecipanti:**

Nome	Matricola
D'Amato Valentina	0512103052
Russo Daniele	0512103196
Cicchelli Marco	0512103292
Sergio Massimo	0512103070

<b>Scritto da:</b>	D'Amato, Russo, Cicchelli, Sergio
--------------------	-----------------------------------

## Revision History

Data	Versione	Descrizione	Autore
13/12/16	0.5	Creazione documento e stesura punti: 1, 2, 3, 4, 5, 6.	Cicchelli M.
14/12/16	0.6	Stesura punti: 7, 8, 10.	Russo D.
14/12/16	0.7	Stesura punto 9.	Massimo S
15/12/16	0.8	Revisione e modifica errori all'interno del testo del documento. Aggiornamento indice.	D'Amato V.
16/12/16	1.0	Revisioni finali documento.	Cicchelli, D'Amato, Russo, Sergio.

## Indice

<b>1. Introduzione</b>	<b>pag. 4</b>
<b>2. Documenti correlati</b>	<b>pag. 4</b>
<b>3. Overview di sistema</b>	<b>pag. 4</b>
<b>4. Funzionalità da testare</b>	<b>pag. 5</b>
<b>5. Criteri Pass/Failed</b>	<b>pag. 5</b>
<b>6. Approccio</b>	<b>pag. 5</b>
<b>7. Sospensione e ripresa</b>	<b>pag. 6</b>
<b>8. Materiale per il Testing</b>	<b>pag. 6</b>
<b>9. Test cases</b>	<b>pag. 6</b>
<b>10. Pianificazione del Testing</b>	<b>pag. 8</b>

## 1. Introduzione

Il test plan ha lo scopo di pianificare l'attività del sistema BoomToys al fine di verificare se differisce il comportamento atteso da quello osservato. In questa attività andremo a valutare gli eventuali errori prodotti dal codice, per evitare che si verifichino durante l'utilizzo dell'utente finale.

Le attività di test pianificate sono:

- Autenticazione/Registrazione
- Area Amministratore
- Area Utente
- Acquisto

Il nostro scopo è quello di definire i test case su cui verranno testate le funzionalità del sistema. Per ogni funzionalità avremo un determinato numero di istanze di input aventi dati corretti e non, in modo da valutarne l'impatto sul sistema.

## 2. Documenti Correlati

Questo documento ha una stretta relazione con i documenti precedentemente prodotti, quindi ora verranno riportate le relazioni tra il test plan e le documentazioni precedenti.

### Relazioni con il documento di analisi dei requisiti (RAD)

Questa relazione riguarda in particolar modo i requisiti funzionali e non, poiché i test eseguiti su ogni funzionalità terranno conto delle specifiche espresse nel RAD.

### Relazioni con il System Design Document (SDD)

Nell'SDD abbiamo suddiviso il sistema BoomToys in sottosistemi secondo l'architettura MVC. Il test che effettueremo dovrà rimanere fedele alle suddivisioni che abbiamo elaborato.

## 3. Overview del sistema

Le principali caratteristiche che dobbiamo testare per il controllo del corretto funzionamento del sistema, saranno:

- Robustezza: capacità del sistema di reagire fornendo input non valido per il dominio applicativo;
- Sicurezza: verrà testato che un utente col solo interfacciamento grafico non possa intaccare dati non inerenti alla specifica dell'operazione o accedere a dati non consentiti;
- Correttezza: verrà testato che le operazioni vengano eseguite correttamente così come dalla specifica dei requisiti sono stati designati.

## 4. Funzionalità da testare

Le funzionalità che andremo a testare sono:

- Autenticazione/Registrazione: inserimento di username e password da parte del cliente per registrarsi al sistema. Inserimento username e password da parte del cliente e amministratore al fine di loggarsi al sistema per accedere alle sue funzionalità.
- Area Amministratore: Dopo inserimento credenziali, cambiamento barra navigazionale (Utenti, Prodotti, Ordini, Logout) e controllo sezioni principali.
- Area Utente: cambiamento barra navigazionale (MieiAcquisti, Carrello, Logout) e controllo sezioni principali.
- Acquisto: Selezione di un prodotto e inserimento di dati per il pagamento.

## 5. Criteri Pass/Failed

I dati di input 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 quello atteso, cioè quello che è stato specificato dal membro di testing su tale test case, il membro di testing conosce quale dovrebbe essere l'output corretto.

## 6. Approccio

Le tecniche di testing adottate riguarderanno inizialmente il testing di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità. Seguirà il testing d'integrazione, 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. Quest'ultimo servirà soprattutto a verificare che il sistema soddisfi le richieste del committente.

### 6.1 Testing di Unità

Durante questa fase, verranno ricercate le condizioni di fallimento isolando i componenti. La strategia utilizzata per il testing si baserà sulla tecnica BlackBox, che si focalizza sul comportamento Input/Output, ignorando la struttura interna della componente. Gli stati erronei scovati in questa fase, come in qualsiasi altra fase di testing, che comporteranno un fallimento del sistema, dovranno essere tempestivamente comunicati agli implementatori al fine di correggerli e ripristinare il testing al più presto.

## 6.2 Testing d'integrazione

Dopo aver sottoposto ogni componente al testing di unità, ed aver corretto eventuali errori riscontrati, si procederà all'integrazione delle componenti di una funzionalità che verranno testate nel complesso attraverso una strategia di Bottom-Up. Si passerà, poi, alla funzionalità successiva fino ad esaurire le funzionalità implementate. Quest'approccio mira principalmente a facilitare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi.

## 6.3 Testing di Sistema

Lo scopo di questa fase del testing è quello di dimostrare che il sistema soddisfi effettivamente i requisiti richiesti e che sia, quindi, pronto all'uso. Si cercherà di testare le funzionalità più importanti per l'utente, quelle che saranno usate di frequente e con una maggiore probabilità di fallimento. Si procederà attraverso l'utilizzo del tool Selenium (poiché si tratta di un sistema web-based) per effettuare il testing di sistema. Selenium si occupa di simulare l'interazione con il sistema dal punto di vista dell'utente.

## 7. Sospensione e ripresa

### Criteri di sospensione

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

### Criteri di ripresa

In seguito ad ogni modifica o correzione delle componenti che genereranno errori o fallimenti, i test case verranno sottoposti nuovamente al sistema assicurandoci così di aver risolto effettivamente il problema.

## 8. Materiale per il Testing

Il materiale utilizzato si divide in hardware e software, come hardware per le attività di testing abbiamo utilizzato vari PC con gli opportuni software installati: DBMS MySQL 5.0 e Eclipse + Apache 8.0.

## 9. Test Cases

Le varie fasi di testing necessiteranno ognuna di test case utili ad individuare errori ed anomalie del codice. Si sono individuate varie classi di equivalenza per ogni input che possa essere immesso per l'utilizzo di una o più componenti. In tal modo, è possibile sviluppare test case con input delle tipologie identificate per testare una o più unità. Di seguito sono elencate le classi di equivalenza che saranno prese in considerazione durante i successivi documenti di Testing per sviluppare i test case.

**Autenticazione/Registrazione**

Input	Classe	Valido	Classe	Non Valido
Username	A01	Username ≠ " ". Username = formato stringa.	A02	Username = " ".
Password	A03	Password ≠ " ". Password = formato alfanumerico.	A04	Password = " ". Password = formato errato

**Pannello Amministratore**

Input	Classe	Valido	Classe	Non Valido
Nome Prodotto	A01	Nome Prodotto ≠ " ". Nome Prodotto = formato stringa.	A02	Nome Prodotto = " ". Nome Prodotto = formato interno.
Tipo	A03	Tipo ≠ " ". Tipo = formato stringa.	A04	Tipo = " ". Tipo = formato stringa.
Codice	A05	Costo = formato intero.	A06	Costo = formato float.
Inserimento Prezzo	A07	Inserimento Prezzo ≠ " ". Inserimento Prezzo = formato float.	A08	Inserimento Prezzo = " ". Inserimento Prezzo = formato errato.

**Acquisto**

Input	Classe	Valido	Classe	Non Valido
Selezione metodo di pagamento	A01	SMP ≠ " ". SMP = formato stringa.	A02	SMP = " ". SMP = formato alfanumerico.
Inserimento Dati Carta di Credito	A03	Inserimento Dati Carta di Credito ≠ " ". Inserimento Dati Carta di Credito = formato intero.	A04	Inserimento Dati Carta di Credito = " ". Inserimento Dati Carta di Credito = formato alfanumerico. Inserimento Dati Carta di Credito = formato errato.

**10. Pianificazione del Testing**

L'attività di testing è fondamentale nello sviluppo di un sistema software in quanto la mancanza di

tale attività o una cattiva interpretazione di essa può portare al completo fallimento del sistema.

### **Gestione dei rischi**

I possibili rischi generati dalle attività di testing sono stati minimizzati diminuendo le componenti del sistema da implementare e, quindi, testare. Il risultato atteso è di riscontrare al più un errore parziale per funzionalità: le componenti interagiscono fra loro e la funzionalità viene eseguita ma non in modo completamente corretto. Qualora la fase di testing evidenziasse un numero di errori maggiore rispetto alla media attesa, viene pianificato un impegno maggiore dei membri del team sulle attività di testing ed in casi estremi l'abbandono delle altre attività finché errori gravi (funzionalità non corretta, risultati errati, modifiche apportate in modo errato) non vengano risolti.

### **Organizzazione delle attività**

Le attività di testing devono svolgersi sulle singole funzionalità divise nei livelli di suddivisione del sistema, rispettando le direttive indicate dal documento di system design.

### **Schedulazione delle attività**

Le attività di testing per il sistema BoomToys sono previste per un periodo di tempo massimo di 10 giorni.