



# Università degli Studi di Salerno

Corso di Ingegneria del Software



**Negozio-Online**  
**Test Plan**  
**Versione 1.7**

**Partecipanti:**

Nome	Matricola
Gaglione Rino	0512102476
Capobianco Marco	0512102462

**Revision History**

Data	Versione	Descrizione	Autore
10/12/2015	1.0	Introduzione	Gaglione Rino
11/12/2015	1.1	Overview-Approccio	Gaglione Rino
14/12/2015	1.2	Overview-Approccio	Gaglione Rino Capobianco Marco
15/12/2015	1.3	Materiale per il testing	Gaglione Rino Capobianco Marco
17/12/2015	1.4	Test cases	Gaglione Rino Capobianco Marco
18/12/2015	1.5	Test cases	Gaglione Rino Capobianco Marco
21/12/2015	1.6	Test cases	Gaglione Rino Capobianco Marco
22/12/2015	1.7	Pianificazione del testing	Gaglione Rino

# Indice

- 1. Introduzione**
- 2. Documenti correlati**
  - 2.1. Relazioni con il documento di analisi dei requisiti (RAD)
  - 2.2. Relazioni con il System Design Document (SDD)
- 3. Overview del sistema**
- 4. Funzionalità da testare**
- 5. Criteri Pass/Failed**
- 6. Approccio**
  - 6.1. Testing di Unità
  - 6.2. Testing d'integrazione
  - 6.3 Testing di Sistema
- 7. Sospensione e ripresa**
  - 7.1. Criteri di sospensione
  - 7.2. Criteri di ripresa
- 8. Materiale per il testing(Requisiti Hardware/software)**
- 9. Test cases**
  - 9.1. Classi di equivalenza
- 10. Pianificazione del testing**
  - 10.1. Gestione dei rischi
  - 10.2. Organizzazione delle attività
  - 10.3. Schedulazione delle attività

# 1.Introduzione

Lo scopo di questo documento è quello di pianificare l'attività di test del sistema Negozio-Online, al fine di verificare se esistono differenze tra il comportamento atteso e il comportamento osservato. In questa attività andremo a rilevare gli eventuali errori prodotti all'interno del codice, per evitare che essi si presentino nel momento in cui il sistema verrà utilizzato dall'utente finale.

Le attività di test sono state pianificate per le seguenti gestioni:

- GestionePersonale
- GestioneMagazzino
- GestionePrenotazioniCliente
- GestioneCliente

Lo scopo del documento è quello di definire i test case su cui verranno testate le funzionalità del sistema. Per ogni funzionalità saranno forniti un numero sufficiente di istanze di input in modo tale da fornire almeno un test case composto da dati corretti, quindi appartenenti a classi valide per ogni campo di input che rispettano le condizioni definite nel documento di test plan, almeno un test case per ogni input che prevede una classe di equivalenza non valida e almeno un test case che non soddisfa le condizioni definite nel test plan. La fase di testing è strettamente legata alle fasi ad essa precedenti; ogni documento, risultato delle differenti fasi di sviluppo, sarà punto di partenza indispensabile per poter effettuare un testing corretto e adeguato.

## 2. Documenti Correlati

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

La relazione tra test plan e RAD riguarda in particolare i requisiti funzionali e non funzionali del sistema poiché i test che saranno eseguiti su ogni funzionalità terranno conto delle specifiche espresse nel RAD.

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

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

### 3. Overview del sistema

Come stabilito nel System Design Document la struttura del nostro sistema è divisa secondo un' architettura "Three tier" cioè a tre livelli: presentation Layer, application Layer, Data Layer. In questo caso il livello più alto interagisce con il livello applicativo che a sua volta si occuperà di eseguire le operazioni nel database cercando di garantire il più possibile basso accoppiamento e alta coesione tra le varie classi.

Le caratteristiche da testare per il controllo del corretto funzionamento di ciascuna funzionalità saranno:

- robustezza: capacità del sistema di reagire fornendo input non valido per il dominio applicativo;
- usabilità: analisi di ogni forma di interazione corrisposta da messaggi di aiuto (in caso di errore) o di notifica (in caso di operazioni eseguite con successo) ;
- 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

Di seguito saranno elencate le funzionalità che saranno testate.

GestionePersonale: Inserimento, elenca ,seleziona (modifica, elimina) . La precedente gestione prevede principalmente operazioni di inserimento, modifica, cancellazione, e visualizzazione e saranno proprio queste funzionalità ad essere testate nel corso della fase di testing del sistema.

GestionePrenotazioniCliente: Elenca,Seleziona(Modifica,Elimina).

GestioneMagazzino:Inserimento,elenca,seleziona(modifica,elimina).

GestioneAcquisti:Elenca,dettagliProdotto,dettagliCarrello,Acquista.

GestioneAccount:Login,Iscrizione,Modifca.

## 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 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 ed usando test driver e test stub, cioè implementazioni parziali di componenti che dipendono o da cui dipendono le componenti da testare. La strategia utilizzata per il testing si baserà esclusivamente sulla tecnica BlackBox, che si focalizza sul comportamento Input/Output, ignorando la struttura interna della componente. Al fine di minimizzare il numero dei test cases, i possibili input verranno partizionati in classi di equivalenza e per ogni classe verrà selezionato un test case. Gli stati erronei scoperti in questa, 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 di Integrazione

In questa fase 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 ridurre le dipendenze tra funzionalità differenti e 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. Come per il testing di unità, si cercherà di testare le funzionalità più importanti per l'utente e quelle che hanno una maggiore probabilità di fallimento. Si noti che, come per il testing di unità, si procederà attraverso la tecnica BlackBox.

# **7.Sospensione e Ripresa**

## 7.1 Criteri di Sospensione

La fase di Testing del sistema verrà sospesa quando si raggiungerà un compromesso tra qualità del prodotto e costi dell'attività di Testing. Il Testing verrà quindi portato avanti quanto più possibile nel tempo senza però rischiare di ritardare la consegna finale del progetto .

## 7.2 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 (Requisiti Hardware/software)

L'hardware necessario per l'attività di test è un PC, o possibilmente uno per ogni oggetto incaricato del testing, su cui sia installato il DBMS MySQL 5.0 e Apache 2 + PHP 5.0

## 9. Test Cases

Le varie fasi di testing necessiteranno ognuna di test case utili ad individuare errori ed anomalie sia analizzando il codice che provando la sua esecuzione. 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.

### 9.1 Classi di equivalenza

InserisciPersonale

Input	Classe	Valido	Classe	Non Valido
Nome	CE01	Nome!=" " Nome=formato stringa	CE02	Nome=" " Nome=formato intero
Cognome	CE03	Cognome!=" " Cognome=formato stringa	CE04	Cognome=" " Cognome=formato intero
Data Nascita	CE05	Data!=" " Data=formato Data(gg/mm/aaaa)	CE06	Data=" "
			CE08	Data!=" " Data=formato sbagliato
Ruolo	CE09	1=<Ruolo<=3	CE10	Ruolo=" "
Username	CE11	Username!=" "	CE12	Username=""
Password	CE10	Password!=" "	CE11	Password=""



### ModificaPersonale

Input	Classe	Valido	Classe	Non Valido
Nome	CE01	Nome!=" " Nome=formato stringa	CE02	Nome=" " Nome=formato intero
Cognome	CE03	Cognome!=" " Cognome=formato stringa	CE04	Cognome=" " Cognome=formato intero
Data Nascita	CE05	Data!=" " Data=formato Data(gg/mm/aaaa)	CE06	Data=" "
			CE07	Data!=" " Data=formato sbagliato
Ruolo	CE08	1=<Ruolo<=3	CE09	Ruolo=" "
User	CE10	Username!=" "	CE11	Username=""

### RicercaPersonale

Input	Classe	Valido	Classe	Non Valido
Ruolo	CE01	1=<Ruolo<=3	CE02	Ruolo=" "

### InserisciProdotto

Input	Classe	Valido	Classe	Non Valido
IDProdotto	CE01	IDProdotto!=" " IDProdotto>0	CE02	IDProdotto =" " IDProdotto<=0
IDs	CE03	IDs!=" "	CE04	IDs=" "
Nome	CE05	Nome!=" " Nome=formato stringa	CE06	Nome=" " Nome=formato intero

Descrizione	CE07	Descrizione!=" " Descrizione=format o testo	CE08	Descrizione=" " Descrizione=formato intero
Scheda_Tecnica	CE09	SchedaTecnica!=" " SchedaTecnica=for mato testo	CE08	SchedaTecnica=" " SchedaTecnica=formato intero
Prezzo	CE10	Prezzo!=" " Prezzo >0	CE11	Prezzo=" "
			CE12	Prezzo!=" " Prezzo <=0
Immagine	CE13	Immagine!=""	CE14	Immagine=""

#### ModificaProdotto

Input	Classe	Valido	Classe	Non Valido
IDProdotto	CE01	IDProdotto!=" " IDProdotto>0	CE02	IDProdotto =" " IDProdotto<=0
IDSottocategoria	CE03	IDSottocategoria!=" "	CE04	IDSottocategoria=" "
Nome	CE05	Nome!=" " Nome=formato stringa	CE06	Nome=" " Nome=formato intero
Descrizione	CE07	Descrizione!=" " Descrizione=formato testo	CE08	Descrizione=" " Descrizione=formato intero
Scheda_Tecnica	CE09	SchedaTecnica!=" " SchedaTecnica=forma to testo	CE08	SchedaTecnica=" " SchedaTecnica=formato intero
Prezzo	CE10	Prezzo!=" " Prezzo >0	CE11	Prezzo=" "
			CE12	Prezzo!=" " Prezzo <=0

Giacenza	CE12	Giacenza!=" " Giacenza>0	CE13	Giacenza=" "
			CE14	Giacenza!=" " Giacenza<=0
Immagine	CE15	Immagine!=""	CE16	Immagine=""

#### RicercaProdotto

Input	Classe	Valido	Classe	Non Valido
Valore	CE01	4=<Valore<=0	CE02	Valore=""

#### RicercaPrenotazioni

Input	Classe	Valido	Classe	Non Valido
valore	CE01	2=<valore<=0	CE02	valore=""

#### ModificaStato

Input	Classe	Valido	Classe	Non Valido
Stato	CE03	1=<stato<=0	CE04	stato=""

## Acquista

Input	Classe	Valido	Classe	Non Valido
Quantita	CE01	Quantita!=" " Quantita>0	CE02	Quantita=" "
			CE03	Quantita!=" " Quantita<=0

## Login

Input	Classe	Valido	Classe	Non Valido
Username	CE01	Username!= " "	CE02	Username=""
Password	CE03	Password!= " "	CE04	Password=""

## Modifica Account

Input	Classe	Valido	Classe	Non Valido
Nome	CE01	Nome!=" " Nome=formato stringa	CE02	Nome!=" " Nome=formato intera
Cognome	CE03	Nome!=" " Nome=formato stringa	CE04	Nome!=" " Nome=formato intera
DataNascita	CE05	Data!=" " Data=formato Data(gg/mm/aaaa)	CE06	Data=" "
			CE07	Data!=" " Data=formato sbagliato
LuogoResidenza	CE08	LuogoResidenza!="" LuogoResidenza=for mato stringa	CE09	LuogoResidenza="" LuogoResidenza=formato intero

Provincia	CE10	Provincia!="" Provincia=formato stringa	CE11	Provincia="" Provincia=formato intero
Cap	CE12	Cap!=" " Cap=formato intero Cap=5 interi	CE13	Cap=" "
			CE14	CAP!="" Cap<5 interi
			CE15	CAP!="" Cap=formato stringa
Via	CE16	Via!="" Via=formato stringa	CE17	Via=""
			CE18	Via!=" " Via=formato intero
Email	CE19	Email!=" "	CE20	Email = " "
	CE21	Email=formato mail corretto	CE22	Email!=" " Email=formato mail sbagliato
Username	CE23	Username!= " "	CE24	Username=""

#### Iscrizione

Input	Classe	Valido	Classe	Non Valido
Nome	CE01	Nome!=" " Nome=formato stringa	CE02	Nome!=" " Nome=formato intera
Cognome	CE03	Nome!=" " Nome=formato stringa	CE04	Nome!=" " Nome=formato intera
Usurname	CE05	Username!= " "	CE06	Username=""
Password	CE07	Password!=" "	CE08	Password=""
DataNascita	CE09	Data!=" "	CE10	Data=" "

		Data=formato Data(gg/mm/aaaa)		
			CE11	Data!=" " Data=formato sbagliato
LuogoResidenza	CE12	LuogoResidenza!="" LuogoResidenza=formato stringa	CE13	LuogoResidenza="" LuogoResidenza=formato intero
Provincia	CE14	Provincia!="" Provincia=formato stringa	CE15	Provincia="" Provincia=formato intero
Cap	CE016	Cap!=" " Cap= 5 interi	CE17	Cap=" "
			CE18	CAP!="" Cap<5 interi
			CE19	CAP!="" Cap=formato stringa
Via	CE20	Via!="" Via=formato stringa	CE21	Via=""
			CE22	Via!=" " Via=formato intero
Email	CE23	Email!=" "	CE24	Email = " "
	CE25	Email=formato mail corretto	CE26	Email!=" " Email=formato mail sbagliato

## 10. Pianificazione del Testing

L'attività di testing è fondamentale nello sviluppo di un sistema software in quanto la mancanza di tale attività o un cattiva interpretazione di essa può portare al completo fallimento del sistema.

### 10.1 Gestione dei rischi

I possibili rischi generati dalle attività di testing sono stati minimizzati diminuendo le componenti del sistema da implementare e, quindi, testare. Inoltre, effettuando un testing di tipo funzionale viene limitato lo sviluppo di stub e driver per il testing delle singole componenti e quindi l'introduzione di nuovi errori nel nuovo codice di cui si sarebbero composti.

D'altro canto il testing di funzionalità rallenta l'individuazione di errori qualora un caso di test avesse esito positivo, poiché l'utilizzo di più componenti per il test di una singola funzionalità estranea dall'ipotesi di totale correttezza di ogni componente interessata.

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.

### 10.2 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 document.

### 10.3 Schedulazione delle attività

Le attività di testing per il sistema Negozio-Online sono previste per un periodo di tempo massimo di 5 giorni.

