Università degli Studi di Salerno

Corso di Ingegneria del Software

Quick Service
Test Manual

Versione 2.0

Data <15/07/2008>

Revision History:

Version R0.1 9/15/98 Robin Loh. Created

Version R0.2 9/14/98 Joyce Johnstone. revised template

Version R1.0 07/05/2008 Pacifico Marta – aggiunta paragrafi 3.2, dal 3.4 al 5.2

Version R1.0 10/05/2008 Pizza Ernesto Luca – aggiunta paragrafi 3.3, dal 3.6 al 3.7.2

Version R1.0 12/06/2008 Iacoletti Alessandro – aggiunta paragrafi 1, 2, 3.1, dal 3.8 al 3.8.2

Version R1.0 14/07/2008 Vicidomini Vincenzo, Pizza Ernesto, Pacifico Marta, – aggiunta paragrafi 2.3, 3.9

Version R2.0 14/07/2008 Iacoletti Alessandro, Mercurio Antonio, Pizza Ernesto – aggiunta paragrafo 4

Version R2.0 15/07/2008 Iacoletti Alessandro – revisione del documento.

Paragrafo 5

Version R1.0 18/07/2008 Pizza Ernesto – aggiunta paragrafo 5

Preface:

This document addresses the requirements of the Quick Service system. The intended audience for this document are the designers and the clients of the project.

Target Audience:
Client, Developers

Quick Service Members:

Project Manager:

Iacoletti Alessandro

Team Members:

Vicidomini Vincenzo

Pacifico Marta

Pizza Ernesto

Mercurio Antonio

Premessa 5

1. Introduzione 6
   1.1 Organizzazione del documento 6

1.2 Scopo del documento 7

2. Test Plan 9
2.1 Premessa 9
2.2 Risorse 10
2.3 Identificazione dei Test Item 10
2.4 Metodologie 11
2.5 Attività di Testing 11
2.5.1 Visualizza ordine 12
2.5.2 Nuova ordinazione 12
2.5.3 Visualizza listino 12
2.5.4 Inserimento prodotto 13
2.5.5 Valida 13

3. Test 14
3.1 Caratteristiche generali 14
3.2 Descrizione dettagliata 14
3.3 Descrizione dei casi di Test 15
3.4 Visualizza ordinazioni 15
3.5 Nuova ordinazione 15
3.5.1 Test Design Specification 16
3.5.2 Test Case Specification 18
3.6 Visualizza listino 26
3.7 Inserimento prodotto 28
3.7.1 Test Design Specification 28
3.7.2 Test Case Specification 28
3.8 Valida 35
3.8.1 Test Design Specification 35
3.8.2 Test Case Specfication 37
3.9 Test degli oggetti remoti 44

4. Test Log 46
4.1 Descrizione dei test log 46
4.2 Test log nuova ordinazione 47
4.3 Test log inserimento prodotto 55
4.4 Test log valida 62

5. Descrizione classi di testo 69
5.1 Test della Collection 69
5.2 Test della Sessione Bar Gestore 71
5.3 Test della Sessione Cliente 75

Premessa

Il documento contiene la descrizione delle attività di testing effettuate sull'applicazione denominata *QUICK SERVICE* (***Documento di testing***).

Come tale, in base agli standard metodologici, il documento si concentra sulla descrizione della pianificazione di tali attività, delle tecniche utilizzate e dei risultati ottenuti in fase di esecuzione.

In questo senso, il documento mira a valutare l'aspetto qualitativo del software sviluppato attuando tecniche di analisi dinamica, cioè di tecniche utilizzate per osservare il comportamento del sistema in fase di esecuzione sulla base di particolari dati di ingresso.

1. Introduzione

1.1 Organizzazione del documento

Il documento può essere suddiviso essenzialmente in due parti fondamentali :

1. **documenti di pianificazione e specifica :** comprende il *Test Plan* (TP*),* il *Test Design Specification* (TDS), e il *Test Case Specification* (TCS).
2. **documenti di esecuzione :** costituito dal *Test Log* (TL).

Il *Capitolo 1* rappresenta un introduzione al documento di *testing.*

Il *Capitolo 2* rappresenta essenzialmente il *Test Plan* che descrive l'oggetto, l'approccio generale, le risorse e lo scheduling delle attività da realizzare e contiene una descrizione dei test item, le caratteristiche da testare, le attività di testing.

Il *Capitolo 3* è rappresentato dalle funzionalità del sistema che noi andremo a testare tramite lo sviluppo di due paragrafi : *Test Design Specification* (TDS), e il *Test Case Specification* (TCS).

Il *Capitolo 4,* infine, rappresenta il *Test Log*, ovvero la banca dati della memorizzazione sistematica, strutturata ed in ordine cronologico di tutti i dettagli rilevanti sulla esecuzione dei test. In particolare le informazioni fondamentali di tale documento sono il successo o l'insuccesso dei test, l'occorrenza e la descrizione di eventi anomali e di testincident.

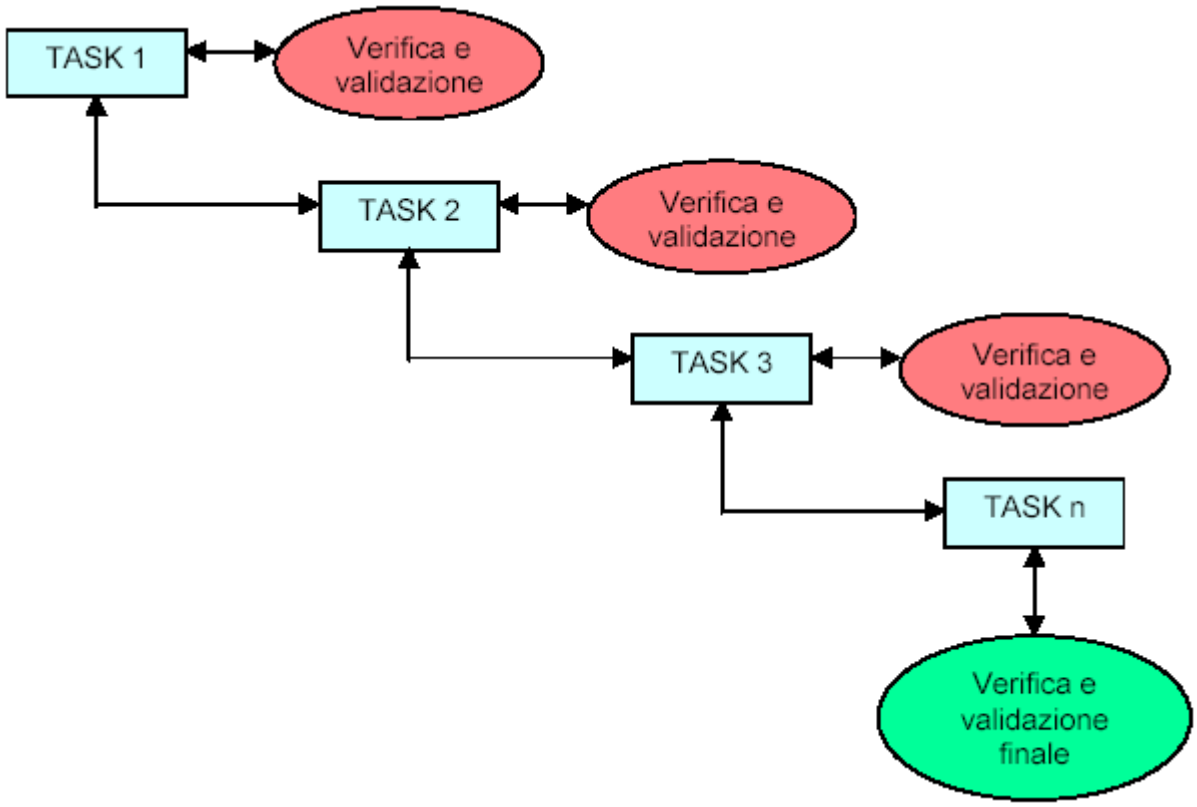
1.2. Scopo del documento

La realizzazione di un buon sistema software richiede che al centro dei processi di sviluppo, manutenzione ed evoluzione del software stesso vi sia il concetto di *qualità*. E' quindi necessario utilizzare metodi quantitativi che permettano di raggiungere tale obiettivo.

Il *Testing* rappresenta senza ombra di dubbio una delle più importanti tecniche per il problema della qualità del software, in quanto consente di analizzare, valutare e quindi promuovere il miglioramento della correttezza dell'implementazione con riferimento alle caratteristiche definite in particolare dal modello dei requisiti software.

Il sistema *QUICK SERVICE* ha come obiettivo la realizzazione di una serie di servizi che vanno ad incrementare ed a rendere più agevoli le funzionalità di un albergo (ordinazioni dalla propria camera tramite touch screen, etc…)

Il presente documento, nato dall'accurata analisi del modello dei requisiti software, contiene la specifica del documento di testing per l'applicazione denominata *QUICK SERVICE* (***Documento di testing***). Tale documento contiene la descrizione dettagliata della pianificazione e quindi della specifica delle attività di testing, la descrizione dei documenti di esecuzione e rappresenta quindi una essenziale sorgente di informazioni per la valutazione e la certificazione di qualità del sistema software in esame.



2. Test Plan

2.1 Premessa

Una pianificazione attenta ed accurata risulta necessaria per poter ottenere i risultati migliori dal processo di testing. E' ovvio che il raggiungimento di tale obiettivo richieda che la pianificazione parta presto nel processo di sviluppo del sistema software, ricordando però di non ridurre il test ad una semplice fase di tale processo e di far si che l'attività di analisi e di test siano presenti per tutta la durata del ciclo di vita. In questo modo sarà possibile pianificare il processo per garantire visibilità il prima possibile e continuamente. Il piano di testing progettato rispetta la suddivisione in moduli del sistema software *QUICK SERVICE* , in particolare verrà testato:

1. **modulo BAR:** insieme dei servizi offerti ai clienti per effettuare ordinazioni al bar dell'albergo.

Il piano di testing ha quindi l'obiettivo di testare il corretto funzionamento dei servizi nel modulo. In particolare, i servizi del *modulo BAR* che verranno testati saranno:

- *Visualizza ordine;*
- *nuova ordinazione;*
- *visualizza listino;*
- *inserimento prodotto;*
- *valida;*

2.2 Risorse

L' esecuzione delle attività di testing verrà effettuata in ambiente distribuito, cioè mediante l'interazione di due o più macchine collegate tra esse da uno switch e da alcuni cavi di rete e richiede su ciascuna di esse l'utilizzo delle seguenti risorse :

- JAVA Virtual Machine;
- Driver JDBC-ODBC

2.3 Identificazione dei test item

Gli oggetti fondamentali delle attività di testing pianificate saranno le seguenti applicazioni (codice oggetto) realizzate in *jbuilder X;*:

- *Quick Service - Server*;
- *Quick Service - Cliente*: interfaccia per l'accesso al S.I.O. da parte del cliente;
- *Quick Service – Addetto Bar*: interfaccia per l'accesso al S.I.O. da parte dell'addetto bar.

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 iterazione 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.

2.4 Metodologie di testing

Nella fase di testing abbiamo utilizzato due diverse metodologie :

- Per gli oggetti remoti sono state usate le classi di test implementate direttamente con il framework JUnit. La scelta di tale framework è dovuta a diverse ragioni:

  1. E' un progetto open-source quindi è liberamente utilizzabile.
  2. Permette di separare i test dal codice.
  3. E' facile da integrare nel processo di compilazione.
  4. È integrabile negli ambienti di sviluppo IDE.

- Per il test del reale comportamento del sistema abbiamo usato un tipo di test manuale, ovvero la semplice compilazione dei campi che vengono mostrati dall'interfaccia grafica

2.5 Attività di testing

Le attività di testing del modulo gestione prenotazioni che il piano progettato realizzerà sono:

- **testing di *visualizza ordine;***
- **testing di *nuova ordinazione;***
- **testing di *visualizza listino;***
- **testing di *inserimento prodotti;***
- **testing di *valida;***

2.5.1. Testing di *Visualizza ordine*

Questa attività di testing non richiede input quindi si testerà solamente che i dati relativi alla ordinazione effettuata dal cliente venga visualizzata correttamente.

2.5.2. Testing di *Nuova ordinazione*

L'*Operatore* seleziona il prodotto/i e conferma la scelta.
A tale attività verranno forniti i seguenti tipi di input:

- Nuova ordinazione mediante la selezione di un prodotto con i campi Quantità e Note non compilati;
- Campo Quantità non compilato;
- Campo Note non compilato;

- Errata compilazione del campo Quantità;
- Nuova ordinazione mediante la selezione di due prodotti, con i campi Quantità e note non compilati;
- Nuova ordinazione mediante la selezione di nessun prodotto;

2.5.3. Testing di *Visualizzazione listino*

Questa attività di testing non richiede input quindi si testerà solamente che i dati relativi al listino dei prodotti venga visualizzata correttamente dal cliente.

2.5.4. Testing di *Inserimento prodotto*

L'Addetto Bar inserisce un nuovo prodotto nel listino.
A tale attività verranno forniti i seguenti tipi di input:

- Nome Prezzo corretti riferiti al prodotto da inserire;
- Nome non corretto riferito al prodotto da inserire;
- Prezzo non corretto riferito al prodotto da inserire;
- Dati inseriti per un prodotto presente nel listino;

2.5.5. Testing di *Valida*

L'*Addetto Bar* accede alla maschera di login, inserisce Login e Password e conferma la validazione.
A tale attività verranno forniti i seguenti tipi di input:

- Valori corretti per Login e Password;
- Valori corretti e scorretti (alternamente) per Login e password.

Se la Login e la password sono corretti, verrà testato che il sistema abiliti l'accesso al S.I.O. Se la Login e la Password non sono corretti, verrà testato che il sistema visualizzi un messaggio di errore e consenta la ripetizione dell'inserimento dei dati.

3. Test

3.1. Caratteristiche generali

Il testing verrà applicato sul sistema software completo e pienamente integrato, con l'obiettivo di valutare l'adesione del sistema ai requisiti specificati: si provvederà quindi alla realizzazione del ***testing di sistema***.

La tecnica di testing che verrà utilizzata per testare il sistema *Quick Service* sarà il *testing funzionale*. Le caratteristiche di tale tecnica ci permetteranno di definire i casi di test e i relativi oracoli sulla base della sola conoscenza dei requisiti del sistema e dei suoi componenti specificati nel *Modello dei requisiti software.* I casi di test saranno suddivisi in classi di equivalenza utilizzando in modo da poter ridurre e semplificare il numero di test case necessari per la realizzazione del testing.

3.2 Descrizione dettagliata

Effettuiamo la descrizione dettagliata delle attività di testing, mettendo in evidenza la suddivisione del dominio di ingresso di ciascun test in *classi di equivalenza*. E' opportuno osservare che ogni funzionalità verrà eseguita almeno una volta e in particolare il numero di esecuzioni dipenderà dai dati di ingresso e di uscita, da precondizioni e postcondizioni.

3.3 Descrizione dei casi di test

Procediamo ora con la descrizione dei casi di test sulla base delle classi di equivalenza definite nel *Test Design Specification*.
Ovviamente verranno rispettati i seguenti vincoli :

- verranno individuati tanti casi di test da coprire tutte le classe di equivalenza valide in modo che ciascuno comprenda il maggior numero possibile di classi valide ancora scoperte;
- verranno individuati tanti casi di test da coprire tutte le classe di equivalenza non valide in modo che ciascuno copra una ed una sola delle classi non valide;
- durante la composizione dei casi di test, saranno prese in considerazione *solo le situazioni effettivamente verificabili*.

3.4 Testing di *Visualizza ordinazioni*

In questa attività di testing andiamo solamente a controllare che i dati visualizzati siano effettivamente quelli relativi alle ordinazioni effettuate dal cliente. Non è quindi necessario effettuare una classificazione formale delle classi di equivalenza.

3.5 Testing *Nuova Ordinazione*

Questa funzione permette al cliente di effettuare una nuova ordinazione al bar.
Per effettuare una ordinazione, il cliente, deve scegliere uno o più prodotti, tra quelli disponibili, definendone la quantità e le eventuali note. Una volta effettuata la scelta il tutto viene confermato con l'ordinazione.
Nell'ordinazione deve essere selezionato almeno un prodotto.
È possibile modificare o cancellare l'ordinazione fatta entro un "Tot" di tempo prestabilito, utile per un eventuale ripensamento del cliente.

3.5.1 *Test Design Specification*

Il Cliente per effettuare una nuova ordinazione deve:

- Selezionare la voce "nuova ordinazione".
- Scegliere almeno un prodotto dalla lista dei prodotti del bar.
- Indicare la quantità di prodotto desiderato.
- Inviare l'ordinazione.

Procediamo ora all'individuazione delle classi di equivalenza.

Selezione del prodotto desiderato

Viene selezionato con il semplice tocco dello schermo il prodotto che si desidera.

Condizione di ingresso: Deve essere visualizzato il listino dei prodotti disponibili (ciò è possibile invocando la funzione di “Visualizzazione Listino”)

- ✓ *Classi di equivalenza valide*

  CE1 : Almeno 1 Prodotto selezionato.
- ✓ *Classi di equivalenza non valide*

  CE2 : Nessun Prodotto selezionato.

Inserimento quantità del prodotto

Il campo corrispondente alla quantità di prodotto desiderata è settato di default ad 1.

La quantità del prodotto viene immessa utilizzando un tastierino numerico, è quindi impossibile specificare una quantità negativa.

- ✓ *Condizione di ingresso 1*: la quantità deve essere superiore a 0 ed inferiore ad X (disponibilità del prodotto).
- ✓ *Condizione di ingresso 2:* deve essere stato selezionato un prodotto dal listino.

- ✓ *Classi di equivalenza valide*

  CE3 : (quantità > 0)

  CE4 : (quantità < X)

  CE5 : (quantità = ‘’)

  CE10 : (quantità <= disponibilità)

- ✓ *Classi di equivalenza non valide*

  CE6 : (quantità = 0)

  CE7 : (quantità > X)

  CE11 : (quantità < disponiblità)

Inserimento note

Il campo note è riservato a quelle persone che desiderano avere delle piccole modifiche durante la preparazione del prodotto ordinato al bar.

Es. Il cliente desidera un caffè senza zucchero. Per fare ciò il cliente seleziona il prodotto e tra le note scrive “Senza zucchero”.

- ✓ *Condizione di ingresso:* Deve essere stato selezionato un prodotto dal listino.

- ✓ *Classi di equivalenza valide*

  CE8 : (note ≠ '')

  CE9 : (note = '')

3.5.2. *Test Case Specification*

Test Case C01	Nuova Ordinazione mediante la sola selezione di un prodotto. Non vengono compilati i campi di Quantità e Note.	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d’Uso: **NUO_ORD**	Permette al Cliente di effettuare una nuova ordinazione presso il bar.	
Priorità	Alta.	
Set Up	Il prodotto è memorizzato all’interno del listino del bar.	
Descrizione Test		
Input	**Prodotto :** Caffè **Quantità :** 1 **Note :** <<Null>>	
Oracolo	**Viene ordinato un Caffè zuccherato.**	

Copertura	*Classi valide:* **CE1, CE5, CE9, CE10** *Classi non valide:* **Nessuna.**

Test Case C02	Nuova Ordinazione mediante la selezione di un prodotto. Non viene compilato il campo Quantità. Nelle note viene scritto : "Senza Zucchero".	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'Uso: **NUO_ORD**	Permette al Cliente di effettuare una nuova ordinazione presso il bar.	
Priorità	Alta.	
Set Up	Il prodotto è memorizzato all'interno del listino del bar.	
Descrizione Test		
Input	**Prodotto :**Caffè **Quantità :** 1 **Note :** Senza zucchero	
Oracolo	**Viene ordinato un Caffè non zuccherato.**	

Copertura	*Classi valide:* **CE1, CE5, CE8, CE10** *Classi non valide:* **Nessuna.**

Test Case C03	Nuova Ordinazione mediante la selezione di un prodotto. Viene compilato il campo Quantità con 2. Il campo Note non viene compilato.	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'Uso: **NUO_ORD**	Permette al Cliente di effettuare una nuova ordinazione presso il bar.	
Priorità	Alta.	
Set Up	Il prodotto è memorizzato all'interno del listino del bar.	
Descrizione Test		
Input	**Prodotto :** Caffè **Quantità :** 2 **Note :** <<Null>>	
Oracolo	**Vengono ordinati due Caffè zuccherati.**	

Copertura	*Classi valide:* **CE1, CE3, CE4, CE9, CE10** *Classi non valide:* **Nessuna.**

Test Case C04	Nuova Ordinazione mediante la selezione di un prodotto. Viene compilato il campo Quantità con 0. Il campo Note non viene compilato.	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'Uso: **NUO_ORD**	Permette al Cliente di effettuare una nuova ordinazione presso il bar.	
Priorità	Alta.	
Set Up	Il prodotto è memorizzato all'interno del listino del bar.	
Descrizione Test		
Input	**Prodotto :** Caffè **Quantità :** 0 **Note :** <<Null>>	

Oracolo	**Dati inseriti errati. Quantità di prodotto inferiore a 1. Non viene effettuata l'ordinazione.**
Copertura	*Classi valide:* **CE1, CE9, CE10** *Classi non valide:* **CE6.**

Test Case C05	Nuova Ordinazione mediante la selezione di un prodotto. Viene compilato il campo Quantità con 100000. Il campo Note non viene compilato.	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'Uso: **NUO_ORD**	Permette al Cliente di effettuare una nuova ordinazione presso il bar.	
Priorità	Alta.	
Set Up	Il prodotto è memorizzato all'interno del listino del bar.	
Descrizione Test		
Input	**Prodotto :** Caffè **Quantità :** 100000 **Note :** <<Null>>	

Oracolo	**Dati inseriti errati. Quantità di prodotto richiesta eccessiva. Non viene effettuata l'ordinazione.**
Copertura	*Classi valide:* **CE1, CE9, CE10.** *Classi non valide:* **CE7.**

Test Case C06	Nuova Ordinazione mediante la selezione di due prodotti. Non vengono compilati i campi Quantità e Note.	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'Uso: **NUO_ORD**	Permette al Cliente di effettuare una nuova ordinazione presso il bar.	
Priorità	Alta.	
Set Up	Il prodotto è memorizzato all'interno del listino del bar.	
Descrizione Test		
Input	**Prodotto :** Caffè, Cornetto **Quantità :** 1,1 **Note :** <<Null>>	

Oracolo	**Vengono ordinati un Caffè Zuccherato e un Cornetto non farcito.**
Copertura	*Classi valide:* **CE1, CE5, CE9,CE10.** *Classi non valide:* **Nessuna.**

Test Case C07	Nuova Ordinazione mediante la selezione di nessun prodotto. Non vengono compilati i campi Quantità e Note.	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'Uso: **NUO_ORD**	Permette al Cliente di effettuare una nuova ordinazione presso il bar.	
Priorità	Alta.	
Set Up	Il prodotto è memorizzato all'interno del listino del bar.	
Descrizione Test		
Input	**Prodotto :** <<Null>> **Quantità :** <<Null>> **Note :** <<Null>>	

Oracolo	**Dati inseriti errati. Si deve selezionare almeno un prodotto. Non viene effettuata l'ordinazione.**
Copertura	*Classi valide:* **CE5, CE9, CE10.** *Classi non valide:* **CE2.**

Test Case C08	Nuova Ordinazione mediante la selezione di un prodotto. La quantità inserita è superiore alla disponibilità del bar.	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'Uso: **NUO_ORD**	Permette al Cliente di effettuare una nuova ordinazione presso il bar.	
Priorità	Alta.	
Set Up	Il prodotto è memorizzato all'interno del listino del bar.	
Descrizione Test		
Input	**Prodotto :** Coca Cola **Quantità :** 85 **Note :** <<Null>>	

Oracolo	**Ordinazione non effettuata, quantità superiore alla disponibilità.**
Copertura	*Classi valide:* **CE5, CE9.** *Classi non valide:* **CE2, CE11.**

3.6 Testing di *Visualizza listino*

In questa attività di testing andiamo solamente a controllare che i dati visualizzati siano effettivamente quelli relativi al listino dei prodotti. Non è quindi necessario effettuare una classificazione formale delle classi di equivalenza.

3.7 Testing di *Inserimento prodotto*

Questa funzionalità permette all'*Addetto bar* di inserire un nuovo prodotto nel listino

3.7.1 *Test Design Specification*

L'*Addetto bar* inserisce i seguenti dati: *Nome, Prezzo e Disponibilità* relativi al prodotto da inserire nel listino, mentre per quanto riguarda il *Tipo* viene utilizzata una "combo box" che visualizza le tipologie di prodotto esistenti e che quindi non richiede nessun controllo di validità in fase di testing e inoltre il *Codice*, essendo di tipo "contatore", non deve essere inserito dall'utente visto che sarà il numero successivo a quello del codice dell'ultimo prodotto immesso nel database.

Procediamo ora all'individuazione delle classi di equivalenza.

Inserimento nome

- ✓ *Condizione di ingresso*: il nome deve essere una stringa non vuota

- ✓ *Classi di equivalenza valide*

  CE1: (caratteri (nome) != "")

  CE2: (nome non presente nel database)

- *Classi di equivalenza non valide*

  CE3: (caratteri (nome) = "")

  CE4: (nome già presente nel database)

Inserimento prezzo

- *Condizione di ingresso:* il prezzo deve essere un numero maggiore o uguale a 0.

- *Classi di equivalenza valide*

  CE5: (numero (prezzo) >= 0)

- *Classi di equivalenza non valide*

CE6: (numero (prezzo) < 0)

Inserimento disponibilità

- *Condizione di ingresso:* la disponibilità deve essere maggiore o uguale di 0 eccetto il caso in cui è *illimitata*, tale condizione si specifica inserendo "-1".

- *Classi di equivalenza valide*

  CE7: ((numero(disponibilità) >= 0) || (numero(disponibilità) = -1))

- *Classi di equivalenza non valide*

  CE8: ((numero(disponibilità) < 0) && (numero(disponibilità) != -1))

3.7.2. Test Case Specification

Test Case C01	**Inserimento dei dati relativi a un prodotto non registrato nel database. I dati inseriti sono non nulli.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **INS_PRO**	Si occupa di inserire un nuovo prodotto nel listino.	
Priorità	Alta	
Set UP	Il prodotto non deve essere già memorizzato all'interno del listino bar.	
Descrizione Test		
Input	**Nome:** Aranciata **Prezzo:** 2,50 **Disponibilità:** 45	
Oracolo	**Il sistema consente la registrazione del prodotto nel listino.**	
Copertura	*Classi valide:* **CE1, CE2, CE5, CE8** *Classi non valide :* **Nessuna**	

Test Case C02	**Inserimento dei dati relativi a un prodotto non registrato nel S.I.O. Il prezzo è valido, il nome è nullo.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **INS_PRO**	Si occupa di inserire un nuovo prodotto nel listino.	
Priorità	Alta	
Set UP	Il prodotto non deve essere già memorizzato all'interno del listino bar.	
Descrizione Test		
Input	**Nome:** <<Null>> **Prezzo:** 2,50 **Disponibilità:** 45	
Oracolo	**Input non valido: Nome**	
Copertura	*Classi valide:* **CE2, CE5, CE7** *Classi non valide :* **CE3**	

Test Case C03	**Inserimento dei dati relativi a un prodotto non registrato nel S.I.O. Dati inseriti non nulli, il prezzo è non valido.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **INS_PRO**	Si occupa di inserire un nuovo prodotto nel listino.	
Priorità	Alta	
Set UP	Il prodotto non deve essere già memorizzato all'interno del listino bar.	
Descrizione Test		
Input	**Nome:** Aranciata **Prezzo:** -2,50 **Disponibilità**: 45	
Oracolo	**Input non valido: Prezzo.**	
Copertura	*Classi valide:* **CE1, CE2, CE7** *Classi non valide :* **CE6**	

Test Case C04	**Inserimento dei dati relativi a un prodotto il cui nome è già registrato nel S.I.O. Dati inseriti non nulli.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **INS_PRO**	Si occupa di inserire un nuovo prodotto nel listino.	
Priorità	Alta	
Set UP	Il prodotto non deve essere già memorizzato all'interno del listino bar.	
Descrizione Test		
Input	**Nome:** Coca Cola **Prezzo:** 3,00 **Disponibilità:** 45	
Oracolo	**Nome già presente nel database Il sistema non consente la memorizzazione dei dati.**	
Copertura	*Classi valide:* **CE1, CE5, CE7** *Classi non valide :* **CE4**	

Test Case C05	**Inserimento dei dati relativi a un prodotto, disponibilità illimitata. Dati inseriti non nulli.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **INS_PRO**	Si occupa di inserire un nuovo prodotto nel listino.	
Priorità	Alta	
Set UP	Il prodotto non deve essere già memorizzato all'interno del listino bar.	
Descrizione Test		
Input	**Nome:** Aranciata **Prezzo:** 3,00 **Disponibilità:** -1	
Oracolo	**Il sistema consente la memorizzazione dei dati.**	
Copertura	*Classi valide:* **CE1, CE5, CE7** *Classi non valide :* **nessuna**	

Test Case C06	**Inserimento dei dati relativi a un prodotto, disponibilità minore di 0. Dati inseriti non nulli.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **INS_PRO**	Si occupa di inserire un nuovo prodotto nel listino.	
Priorità	Alta	
Set UP	Il prodotto non deve essere già memorizzato all'interno del listino bar.	
Descrizione Test		
Input	**Nome:** Aranciata **Prezzo:** 3,00 **Disponibilità:** -1234	
Oracolo	**Il sistema non consente la memorizzazione dei dati.** **Disponibilità non valida.**	
Copertura	*Classi valide:* **CE1, CE5** *Classi non valide :* **CE8**	

Test Case C07	**Inserimento dei dati relativi a un prodotto.** **Dati inseriti non nulli tranne disponibilità.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **INS_PRO**	Si occupa di inserire un nuovo prodotto nel listino.	
Priorità	Alta	
Set UP	Il prodotto non deve essere già memorizzato all'interno del listino bar.	
Descrizione Test		
Input	**Nome:** Aranciata **Prezzo:** 3,00 **Disponibilità:** <<Null>>	
Oracolo	**Il sistema consente la memorizzazione dei dati.**	
Copertura	*Classi valide:* **CE1, CE5, CE7** *Classi non valide :* **CE8**	

3.8 Testing di *valida*

Questa funzionalità permete all'*Addetto bar* di compiere operazioni riservate, con l'accesso al sistema tramite Login e Password.

3.8.1 *Test Design Specification*

L'*Adetto bar* inserisce login e password nell'apposita sezione della maschera di login.

Procediamo ora all'individuazione delle classi di equivalenza.

Inserimento login

- ✓ *Condizione di ingresso 1*: la login è una stringa alfanumerica composta da 6 caratteri.
- ✓ *Condizione di ingresso 2*: la login deve essere presente all'interno del S.I.O.

- *Classi di equivalenza valide*

  CE1 : ((#caratteri(login) = 5)

  CE2 : (login presente nel S.I.O.)

- *Classi di equivalenza non valide*

  CE3 : (#caratteri(login) < 5)

  CE4 : (#caratteri(login) > 5)

  CE5 : (login non presente nel S.I.O.)

Inserimento password

- ✓ *Condizione di ingresso 1:* la password è una stringa alfanumerica composta da 6 caratteri.
- ✓ *Condizione di ingresso 2:* la password deve essere presente all'interno del S.I.O.

- *Classi di equivalenza valide*

  CE6 : (#caratteri(password) = 5)

  CE7 : (password presente nel S.I.O.)

- *Classi di equivalenza non valide*

  CE8 : (#caratteri(password) < 5)

  CE9 : (#caratteri(password) > 5)

  CE10 : (password non presente nel S.I.O.)

3.8.2 *Test Case Specification*

Test Case C01	**Valida addetto bar mediante login corretta, password corretta e la coppia login, password presente nel S.I.O.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **VALIDA**	Si occupa di eseguire le funzioni necessarie per l'autenticazione dell'addetto bar	
Priorità	Alta	
Set UP	La coppia (login, password) = (admin, admin) è registrata nel S.I.O.	
Descrizione Test		
Input	**Login:** admin **Password:** admin	
Oracolo	**Il sistema consente l'accesso all'addetto bar**	
Copertura	*Classi valide:* **CE1, CE2, CE6, CE7** *Classi non valide :* **Nessuna**	

Test Case C02	**Valida addetto bar mediante login non corretta (numero inferiore di caratteri), e password registrata nel S.I.O.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **VALIDA**	Si occupa di eseguire le funzioni necessarie per l'autenticazione dell'addetto bar	
Priorità	Alta	
Set UP	La password "admin" è registrata nel S.I.O.	
Descrizione Test		
Input	**Login:** adm **Password:** admin	
Oracolo	**Input non valido: login**	
Copertura	*Classi valide:* **CE6, CE7** *Classi non valide :***CE3, CE5**	

Test Case C03	**Valida addetto bar mediante login non corretta (numero superiore di caratteri), e password registrata nel S.I.O.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **VALIDA**	Si occupa di eseguire le funzioni necessarie per l'autenticazione dell'addetto bar	
Priorità	Alta	
Set UP	La password **admin** è registrata nel S.I.O.	
Descrizione Test		
Input	**Login:** vitolosalvatore **Password:** admin	
Oracolo	**Input non valido: login**	
Copertura	*Classi valide:* **CE6, CE7** *Classi non valide :***CE4, CE5**	

Test Case C04	**Valida addetto bar mediante login registrate nel S.I.O. e password non corretta (numero inferiore di caratteri).**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **VALIDA**	Si occupa di eseguire le funzioni necessarie per l'autenticazione dell'addetto bar	
Priorità	Alta	
Set UP	La login **admin** è registrata nel S.I.O.	
Descrizione Test		
Input	**Login:** admin **Password:** sal	
Oracolo	**Input non valido: password**	
Copertura	*Classi valide:* **CE1, CE2** *Classi non valide :***CE8, CE10**	

Test Case C05	**Valida addetto bar mediante login registrate nel S.I.O. e password non corretta (numero superiore di caratteri).**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **VALIDA**	Si occupa di eseguire le funzioni necessarie per l'autenticazione dell'addetto bar	
Priorità	Alta	
Set UP	La login **admin** è registrata nel S.I.O.	
Descrizione Test		
Input	**Login:** admin **Password:** salvatore1983	
Oracolo	**Input non valido: password**	
Copertura	*Classi valide:* **CE1, CE2** *Classi non valide :***CE9, CE10**	

Test Case C06	**Valida addetto bar mediante login registrata nel S.I.O. e password corretta ma non registrata nel S.I.O.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **VALIDA**	Si occupa di eseguire le funzioni necessarie per l'autenticazione dell'addetto bar	
Priorità	Alta	
Set UP	La login **admin** è registrata nel S.I.O. La password lello non è registrata nel S.I.O.	
Descrizione Test		
Input	**Login:** admin **Password:** lello	
Oracolo	**Il sistema non consente l'accesoo all'addetto bar**	
Copertura	*Classi valide:* **CE1, CE2, CE6** *Classi non valide :***CE10**	

Test Case C07	**Valida addetto bar mediante login corretta ma non registrata nel S.I.O. e password corretta registrata nel S.I.O.**	*Data : 11 / 06 / 2008* *Versione : 0.1*
Caso d'uso: **VALIDA**	Si occupa di eseguire le funzioni necessarie per l'autenticazione dell'addetto bar	
Priorità	Alta	
Set UP	La login **lello** non è registrata nel S.I.O. La password **admin** è registrata nel S.I.O.	
Descrizione Test		
Input	**Login:** lello **Password:** admin	
Oracolo	**Il sistema non consente l'accesso all'addetto bar**	
Copertura	*Classi valide:* **CE1, CE6, CE7** *Classi non valide :***CE5**	

3.9 Test degli oggetti remoti

Ogni test è composto da test case individuali. Ogni test case è una classe che estende TestCase.
Per convenzione la classe di test ha lo stesso nome della classe da testare cui viene aggiunto il prefisso “Test” (ad es. HelloWorld.java diventa TestHelloWorld.java) e si trova all'interno dello stesso package della classe cui viene aggiunto il suffisso “test” (ad es. se il package della classe da testare è hello il package della classe di test sarà hello.test). Analogamente i metodi della classe di test hanno lo stesso nome dei metodi della classe da testare ma inziano con il prefisso “test”.
La verifica dei test case è effettuata mediante le condizioni di assert (ad esempio: assertEquals, assertTrue, assertNotNull).
Supponiamo di voler scrivere un test case per la seguente classe:

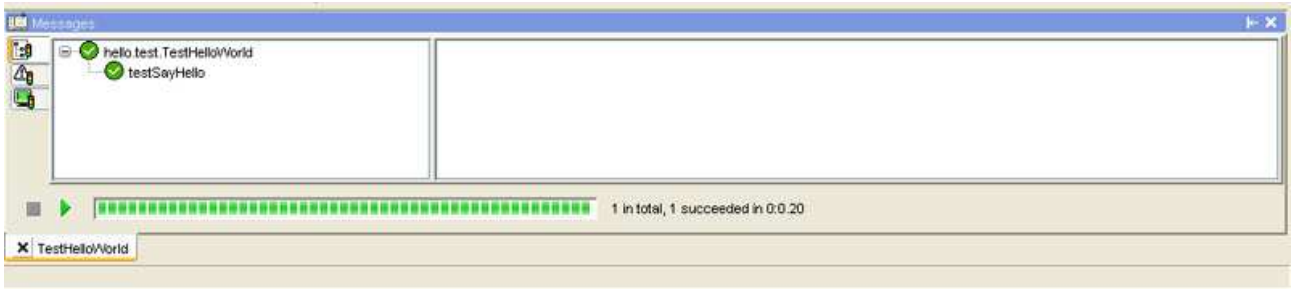
```
package hello;

class HelloWorld {

     public String sayHello(){
           return "Hello World";

     }


     public static void main(String[] args){
           HelloWorld world=new HelloWorld();
           System.out.println(world.sayHello());
     }
}
```

In base a quanto detto creiamo la nuova classe TestHelloWorld contenente un metodo per ogni metodo della classe da testare (nel nostro caso il solo metodo testSayHello). Il risultato è il seguente:

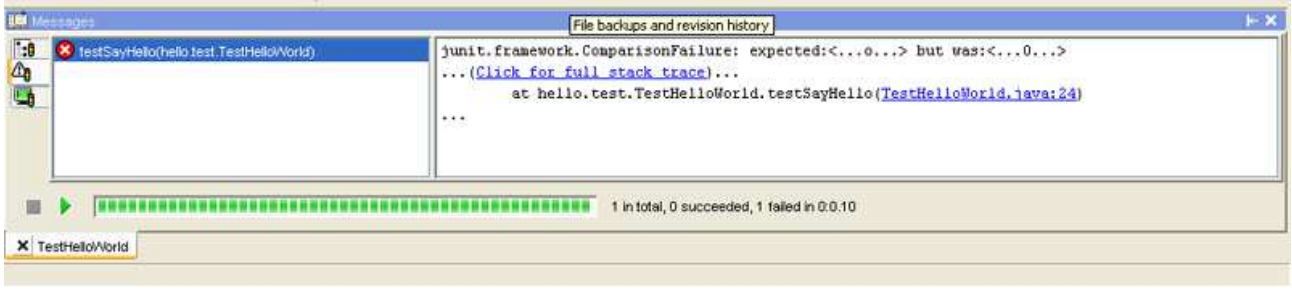
```
package hello.test;
```

```
import hello.HelloWorld;
import junit.framework.TestCase;
import junit.framework.AssertionFailedError;

public class TestHelloWorld extends TestCase{

       public TestHelloWorld(String name){
             super(name);

       }

       public static void main(String[] args){
            junit.textui.TestRunner.run(TestHelloWorld.class);
       }


       public void testSayHello(){
               HelloWorld world = new HelloWorld();
                assertEquals("Hello World",world.sayHello());
        }
}
```

Dal momento che il framework JUnit è integrato nell'ambiente di sviluppo da noi utilizzato, è possibile controllare l'esecuzione dei test direttamente da JBuilder. La seguente figura mostra l'esito positivo dell'esecuzione del test per la classe dell'esempio precedente:



Supponiamo ora di modificare la stringa restituita dal metodo sayHello della classe da testare in "Hello WOrld". Il testcase fallirà e la seguente figura mostra l'esito dell'esecuzione:



4. Test log

4.1 Descrizione del test log

In questa sezione descriviamo i risultati dell'esecuzione dei casi di test, riportando per ognuno la descrizione delle seguenti caratteristiche :

_ ***Stato iniziale***: situazione del sistema immediatamente prima dell'esecuzione del test;
_ ***Scenario***: descrizione dettagliata dei passi di esecuzione. Per ogni passo di esecuzione viene registrato il completamento o meno dell'operazione effettuata, e in caso di anomalie del sistema l'ID dell'errore che si è presentato ;
_ ***Stato finale***: situazione del sistema al termine dell'esecuzione del test ;
_ ***Conclusioni***: descrizione dell'esito del test. Viene stabilito se il test ha avuto successo o meno, e se il completamento è rimasto in sospeso ;

4.2 Test log Nuova Ordinazione

<table>
<tr><td rowspan="2">Test Case C01</td><td colspan="3" rowspan="2">Nuova Ordinazione mediante la sola selezione di un prodotto. Non vengono compilati i campi di Quantità e Note.</td><td>Data : 13 / 07 / 2008</td></tr>
<tr><td>Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto è memorizzato all’interno del listino del bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="3">Scenario principale
CASO D’USO: NUO_ORD</td><td>Scelta prodotto: Caffè</td><td>SI</td><td></td></tr>
<tr><td>Scelta quantità: 1</td><td>SI</td><td></td></tr>
<tr><td>Inserimento note: <<Null>></td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema informa che l’ordinazione è avvenuta con successo.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

Test Case C02	Nuova Ordinazione mediante la selezione di un prodotto. Non viene compilato il campo Quantità. Nelle note viene scritto : “Senza Zucchero”.	*Data : 13 / 07 / 2008*
		Versione : 0.1
Stato iniziale	Il prodotto è memorizzato all’interno del listino del bar.	

Descrizione esecuzione			
Scenario	**Descrizione passi**	**Positivo**	**IDErr**
Scenario alternativo CASO D’USO: **NUO_ORD**	Scelta prodotto: **Caffè**	**SI**	
	Scelta quantità: **1**	**SI**	
	Inserimento note: **Senza zucchero**	**SI**	

Rapporto dettagliato dei risultati	
Stato finale	**Il sistema informa che l’ordinazione è avvenuta con successo.**
Pending	NO
Passed	SI

<table>
<tr><td>Test Case C03</td><td colspan="3">Nuova Ordinazione mediante la selezione di un prodotto. Viene compilato il campo Quantità con 2. Il campo Note non viene compilato.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto è memorizzato all’interno del listino del bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="3">Scenario alternativo
CASO D’USO: NUO_ORD</td><td>Scelta prodotto: Caffè</td><td>SI</td><td></td></tr>
<tr><td>Scelta quantità: 2</td><td>SI</td><td></td></tr>
<tr><td>Inserimento note: <<Null>></td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema informa che l’ordinazione è avvenuta con successo.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

Test Case C04	Nuova Ordinazione mediante la selezione di un prodotto. Viene compilato il campo Quantità con 0. Il campo Note non viene compilato.	*Data : 13 / 07 / 2008*
		Versione : 0.1
Stato iniziale	Il prodotto è memorizzato all'interno del listino del bar.	

Descrizione esecuzione			
Scenario	**Descrizione passi**	**Positivo**	**IDErr**
Scenario alternativo CASO D'USO: **NUO_ORD**	Scelta prodotto: **Caffè**	**SI**	
	Scelta quantità: **0**	**SI**	
	Inserimento note: <<**Null**>>	**SI**	

Rapporto dettagliato dei risultati	
Stato finale	**Il sistema non consente di scegliere quantità "0" poichè il tasto "0" è disabilitato.**
Pending	NO
Passed	SI

Test Case C05	Nuova Ordinazione mediante la selezione di un prodotto. Viene compilato il campo Quantità con 100000. Il campo Note non viene compilato.	*Data : 13 / 07 / 2008*
		Versione : 0.1
Stato iniziale	Il prodotto è memorizzato all'interno del listino del bar.	

Descrizione esecuzione

Scenario	Descrizione passi	Positivo	IDErr
Scenario alternativo CASO D'USO: **NUO_ORD**	Scelta prodotto: **Caffè**	**SI**	
	Scelta quantità: **100000**	**SI**	
	Inserimento note: **<<Null>>**	**SI**	

Rapporto dettagliato dei risultati

Stato finale	**Il sistema non consente di scegliere quantità superiori a 99.**
Pending	NO
Passed	SI

Test Case C06	Nuova Ordinazione mediante la selezione di due prodotti. Non vengono compilati i campi Quantità e Note.	*Data : 13 / 07 / 2008*
		Versione : 0.1
Stato iniziale	Il prodotto è memorizzato all'interno del listino del bar.	

Descrizione esecuzione			
Scenario	**Descrizione passi**	**Positivo**	**IDErr**
Scenario alternativo CASO D'USO: **NUO_ORD**	Scelta prodotto: **Caffè, Cornetto**	**SI**	
	Scelta quantità: **1,1**	**SI**	
	Inserimento note: <<**Null**>>	**SI**	

Rapporto dettagliato dei risultati	
Stato finale	**Il sistema informa che l'ordinazione è avvenuta con successo.**
Pending	NO
Passed	SI

<table>
<tr><td>Test Case C07</td><td colspan="3">Nuova Ordinazione mediante la selezione di nessun prodotto. Non vengono compilati i campi Quantità e Note.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto è memorizzato all’interno del listino del bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="3"><i>Scenario alternativo</i>
CASO D’USO: NUO_ORD</td><td>Scelta prodotto: <<Null>></td><td>SI</td><td></td></tr>
<tr><td>Scelta quantità: <<Null>></td><td>SI</td><td></td></tr>
<tr><td>Inserimento note: <<Null>></td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema informa che è impossibile inviare un’ordinazione nulla.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

Test Case C08	Nuova Ordinazione mediante la selezione di un prodotto. La quantità inserita è superiore alla disponibilità del bar.	*Data : 13 / 07 / 2008*
		Versione : 0.1
Stato iniziale	Il prodotto è memorizzato all'interno del listino del bar.	

Descrizione esecuzione			
Scenario	**Descrizione passi**	**Positivo**	**IDErr**
Scenario alternativo CASO D'USO: **NUO_ORD**	Scelta prodotto: **Coca Cola**	**SI**	
	Scelta quantità: **85**	**SI**	
	Inserimento note: **<<Null>>**	**SI**	

Rapporto dettagliato dei risultati	
Stato finale	**Il sistema informa che è impossibile inviare l'ordinazione, perché la quantità richiesta è superiore alla disponibilità del prodotto.**
Pending	NO
Passed	SI

4.3 Test log Inserimento Prodotto

<table>
<tr><td>Test Case C01</td><td colspan="3">Inserimento dei dati relativi a un prodotto non registrato nel database. I dati inseriti sono non nulli.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto non deve essere già memorizzato all’interno del listino bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="3">Scenario principale
CASO D’USO: INS_PRO</td><td>Inserimento nome:
Aranciata</td><td>SI</td><td></td></tr>
<tr><td>Inserimento prezzo: 2,50</td><td>SI</td><td></td></tr>
<tr><td>Inserimento disponibilità: 45</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il prodotto è stato inserito nel database.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C02</td><td colspan="3">Inserimento dei dati relativi a un prodotto non registrato nel S.I.O. Il prezzo e la disponibilità sono validi, il nome è nullo.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto non deve essere già memorizzato all’interno del listino bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="3">Scenario alternativo
CASO D’USO: INS_PRO</td><td>Inserimento nome: <<Null>></td><td>SI</td><td></td></tr>
<tr><td>Inserimento prezzo: 2,50</td><td>SI</td><td></td></tr>
<tr><td>Inserimento disponibilità: 45</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il prodotto non viene inserito (nome nullo).</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C03</td><td colspan="3">Inserimento dei dati relativi a un prodotto non registrato nel S.I.O. Dati inseriti non nulli, il prezzo è non valido.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto non deve essere già memorizzato all’interno del listino bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="3">Scenario alternativo
CASO D’USO: INS_PRO</td><td>Inserimento nome:
Aranciata</td><td>SI</td><td></td></tr>
<tr><td>Inserimento prezzo: -2,50</td><td>SI</td><td></td></tr>
<tr><td>Inserimento disponibilità: 45</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema non permette l’inserimento (prezzo non valido).</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C04</td><td colspan="3">Inserimento dei dati relativi a un prodotto il cui nome è già registrato nel S.I.O.
Dati inseriti non nulli.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto non deve essere già memorizzato all’interno del listino bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="3">Scenario alternativo
CASO D’USO: INS_PRO</td><td>Inserimento nome: Coca Cola</td><td>SI</td><td></td></tr>
<tr><td>Inserimento prezzo: 3,00</td><td>SI</td><td></td></tr>
<tr><td>Inserimento disponibilità: 45</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema non permette l’inserimento (prodotto già presente).</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C05</td><td colspan="3">Inserimento dei dati relativi a un prodotto non registrato nel S.I.O. Dati inseriti non nulli, la disponibilità è illimitata.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto non deve essere già memorizzato all’interno del listino bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td colspan="2">IDErr</td></tr>
<tr><td rowspan="3">Scenario alternativo
CASO D’USO: INS_PRO</td><td>Inserimento nome:
Aranciata</td><td>SI</td><td colspan="2"></td></tr>
<tr><td>Inserimento prezzo: 2,50</td><td>SI</td><td colspan="2"></td></tr>
<tr><td>Inserimento disponibilità: -1</td><td>SI</td><td colspan="2"></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema permette l’inserimento (disponibilità illimitata).</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C06</td><td colspan="3">Inserimento dei dati relativi a un prodotto il cui nome non registrato nel S.I.O.
Dati inseriti non nulli, disponibilità non valida.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto non deve essere già memorizzato all’interno del listino bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="3">Scenario alternativo
CASO D’USO: INS_PRO</td><td>Inserimento nome:
Aranciata</td><td>SI</td><td></td></tr>
<tr><td>Inserimento prezzo: 2,50</td><td>SI</td><td></td></tr>
<tr><td>Inserimento disponibilità:
-1234</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema non permette l’inserimento (disponibilità negativa).</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C07</td><td colspan="3">Inserimento dei dati relativi a un prodotto il cui nome non è registrato nel S.I.O.
Dati inseriti non nulli, tranne disponibilità.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">Il prodotto non deve essere già memorizzato all’interno del listino bar.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="3">Scenario alternativo
CASO D’USO: INS_PRO</td><td>Inserimento nome:
Aranciata</td><td>SI</td><td></td></tr>
<tr><td>Inserimento prezzo: 2,50</td><td>SI</td><td></td></tr>
<tr><td>Inserimento disponibilità:
<<Null>></td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema permette l’inserimento con disponibilità 0.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

4.4 Test log Valida

Test Case C01	**Valida addetto bar mediante login corretta, password corretta e la coppia login, password presente nel S.I.O.**	*Data : 13 / 07 / 2008*
		Versione : 0.1
Stato iniziale	La coppia (login, password) = (admin, admin) è registrata nel S.I.O.	

Descrizione esecuzione			
Scenario	**Descrizione passi**	**Positivo**	**IDErr**
Scenario principale CASO D'USO: **LOG_PWD**	Inserimento nel campo login: **admin**	**SI**	
	Inserimento nel campo password: **admin**	**SI**	

Rapporto dettagliato dei risultati	
Stato finale	**Il sistema ha consentito l'accesso.**
Pending	NO
Passed	SI

<table>
<tr><td>Test Case C02</td><td colspan="3">Valida addetto bar mediante login non corretta (numero inferiore di caratteri), e password registrata nel S.I.O.</td><td>Data : 13 / 07 / 2008

Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">La password adm non è registrata nel S.I.O.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="2">Scenario alternativo
CASO D’USO: LOG_PWD</td><td>Inserimento nel campo login: adm</td><td>SI</td><td></td></tr>
<tr><td>Inserimento nel campo password: admin</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema non ha consentito l’accesso perché il numero di caratteri della password è inferiore a 5.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C03</td><td colspan="3">Valida addetto bar mediante login non corretta (numero superiore di caratteri), e password registrata nel S.I.O.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">La password admin è registrata nel S.I.O.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="2">Scenario alternativo
CASO D’USO: LOG_PWD</td><td>Inserimento nel campo login: vitolosalvatore</td><td>SI</td><td></td></tr>
<tr><td>Inserimento nel campo password: admin</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema non ha consentito l’accesso perché il numero di caratteri della login è superiore a 5.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C04</td><td colspan="3">Valida addetto bar mediante login registrate nel S.I.O. e password non corretta (numero inferiore di caratteri).</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">La login admin è registrata nel S.I.O.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="2">Scenario alternativo
CASO D’USO:LOG_PWD</td><td>Inserimento nel campo login: admin</td><td>SI</td><td></td></tr>
<tr><td>Inserimento nel campo password: sal</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema non ha consentito l’accesso perché il numero di caratteri della password è inferiore a 5.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C05</td><td colspan="3">Valida addetto bar mediante login registrate nel S.I.O. e password non corretta (numero superiore di caratteri).</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">La login admin è registrata nel S.I.O.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="2">Scenario alternativo
CASO D’USO:LOG_PWD</td><td>Inserimento nel campo login: admin</td><td>SI</td><td></td></tr>
<tr><td>Inserimento nel campo password: salvatore1983</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema non ha consentito l’accesso perché il numero di caratteri della password è superiore a 5.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C06</td><td colspan="3">Valida addetto bar mediante login registrata nel S.I.O. e password corretta ma non registrata nel S.I.O.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">La login admin è registrata nel S.I.O.

La password lello non è registrata nel S.I.O.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td colspan="2">Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="2">Scenario alternativo
CASO D’USO:LOG_PWD</td><td colspan="2">Inserimento nel campo login: admin</td><td>SI</td><td></td></tr>
<tr><td colspan="2">Inserimento nel campo password: lello</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema non ha consentito l’accesso perché la password è errata.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

<table>
<tr><td>Test Case C07</td><td colspan="3">Valida addetto bar mediante login corretta ma non registrata nel S.I.O. e password corretta registrata nel S.I.O.</td><td>Data : 13 / 07 / 2008
Versione : 0.1</td></tr>
<tr><td>Stato iniziale</td><td colspan="4">La login lello non è registrata nel S.I.O.
La password admin è registrata nel S.I.O.</td></tr>
<tr><td colspan="5">Descrizione esecuzione</td></tr>
<tr><td>Scenario</td><td>Descrizione passi</td><td>Positivo</td><td>IDErr</td></tr>
<tr><td rowspan="2">Scenario alternativo
CASO D’USO:LOG_PWD</td><td>Inserimento nel campo login: lello</td><td>SI</td><td></td></tr>
<tr><td>Inserimento nel campo password: admin</td><td>SI</td><td></td></tr>
<tr><td colspan="5">Rapporto dettagliato dei risultati</td></tr>
<tr><td>Stato finale</td><td colspan="4">Il sistema non ha consentito l’accesso perché la login è errata.</td></tr>
<tr><td>Pending</td><td colspan="4">NO</td></tr>
<tr><td>Passed</td><td colspan="4">SI</td></tr>
</table>

5. Descrizione classi di testo

5.1 Test della collection

```
package Qsclassicomuni.test;

import Qsclassicomuni.adapters.*;
import Qsclassicomuni.collezioni.*;
import junit.framework.*;

/**
 * <p>Title: TestQsCollection</p>
 * <p>Description: Testa le collection usate</p>
 * <p>Copyright: Copyright (c) 2008</p>
 * <p>Company: UNISA (Corso ingegneria del Software)</p>
 * @author Gruppo De Sio
 * @version 1.0
 */

public class TestQsCollection
    extends TestCase {
  private QsCollection qsCollection;

 /**
  * Inizializza le classi da testare
  * @throws Exception
  */

  protected void setUp() throws Exception {
    super.setUp();
    qsCollection = new QsCollectionAdapter();
    qsCollection.aggiungi("VOCE 0");
    qsCollection.aggiungi("VOCE 1");
  }

 /**
  * Il metodo testa l'aggiunta di un elemento alla collection
  */
  public void testAggiungi() {
    int i;
    for (i = 2; i < 10; i++) {
      qsCollection.aggiungi("VOCE " + i);
    }
    System.out.println(qsCollection.dimensione());
    /**@todo fill in the test code*/
  }


 /**
  * Il metodo testa il prelievo di un elemento da una collection.
  * Il valore della variabile oracolo deve essere lo stesso di output.
```

```
 */
  public void testPreleva() {
    int i = 0;
    String oracolo = "VOCE 0";
    String output = (String) qsCollection.preleva(i);
    assertEquals("return value", oracolo, output);
    /**@todo fill in the test code*/
  }

/**
 * Il metodo testa la modifica di un elemento in una collection.
 * Il test avrà sccesso se la variabile oracolo avrà lo stesso valore del
 * risultato di qsCollection.preleva(0).
 */
  public void testModifica() {
    int i = 0;
    String oracolo = "VOCE 00";
    qsCollection.modifica(0, oracolo);
    assertEquals("Modified value", oracolo,
                 (String) qsCollection.preleva(0));
  }

/**
 * Il metodo testa la richiesta della dimensione della collection.
 * Il test avrà sccesso se la variabile oracolo avrà lo stesso valore di output.
 */
 public void testDimensione() {
    int oracolo = 2;
    int output = qsCollection.dimensione();
    System.out.println(output);
    assertEquals("return value", oracolo, output);
    /**@todo fill in the test code*/
  }

/**
 * Il metodo testa la rimozione di un elemento da una collection.
 */
  public void testRimuovi() {
    int oldSize = qsCollection.dimensione();
    qsCollection.rimuovi(0);
    assertTrue("new size", qsCollection.dimensione() < oldSize);

  }
/**
 * Il metodo svuota le collection e libera la memoria.
 * @throws Exception
 */
  protected void tearDown() throws Exception {
    qsCollection = null;
    super.tearDown();
  }
}
```

5.2 Test Sessione Bar Gestore

```
package QsServer.test;

import java.rmi.*;
import java.util.*;

import QsServer.*;
import QsServer.AddettoBar.*;
import QsServer.DataLayer.*;
import Qsclassicomuni.collezioni.*;
import com.borland.dx.sql.dataset.*;
import junit.framework.*;

/**
 * <p>Title: TestSessioneBarGestoreImp</p>
 * <p>Description: Testa la sessione per la gestione del bar</p>
 * <p>Copyright: Copyright (c) 2008</p>
 * <p>Company: UNISA (Corso ingegneria del Software)</p>
 * @author Gruppo De Sio
 * @version 1.0
 */

public class TestSessioneBarGestoreImp
    extends TestCase {
  private SessioneBarGestoreImp sessioneBarGestoreImp = null;
  private Database database;
  private ResourceBundle sqlRes;

 /**
  * Inizializza le classi da testare
  * @throws Exception
  */

  protected void setUp() throws Exception {
    super.setUp();
    /**@todo verify the constructors*/
    database = new Database();
    database.setConnection(new
com.borland.dx.sql.dataset.ConnectionDescriptor(
        "jdbc:odbc:QuickService", "", "", false,
"sun.jdbc.odbc.JdbcOdbcDriver"));
    sqlRes = ResourceBundle.getBundle("QsServer.SqlRes");
    sessioneBarGestoreImp = new SessioneBarGestoreImp(database,
sqlRes);
  }

 /**
   * Il metodo svuota le collection e libera la memoria.
   * @throws Exception
   */
```

```
  protected void tearDown() throws Exception {
    sessioneBarGestoreImp = null;
    database.closeConnection();
    database = null;
    super.tearDown();
  }

 /**
  * Il metodo testa la creazione di una nuova sessione per il gestore del bar.
  * @throws RemoteException
  */

  public void testSessioneBarGestoreImp() throws RemoteException {
    Database db = database;
    ResourceBundle sqlR = sqlRes;
    sessioneBarGestoreImp = new SessioneBarGestoreImp(database,
sqlRes);
    /**@todo fill in the test code*/
  }


 /**
  * Il metodo testa il caricamento categorie.
  * Se la dimensione è maggiore di 0 allora esistono categorie.
  * @throws RemoteException
  */
 public void testCaricaCategorieDimensione() throws RemoteException
{
    QsCollection output = sessioneBarGestoreImp.caricaCategorie();
    assertTrue("ci sono categorie", output.dimensione() > 0);
    /**@todo fill in the test code*/
  }


 /**
  * Testa che il metodo restituisca correttamente i dati dell'utente.
  * Il test avrà sccesso se la variabile oracolo avrà lo stesso valore di output.
  */
  public void testGetDatiUtente() {
    Utente oracolo = null;
    Utente output = sessioneBarGestoreImp.getDatiUtente();
    assertEquals("return value", oracolo, output);
    /**@todo fill in the test code*/
  }


 /**
  * Il metodo testa l'inserimento di un nuovo prodotto nel database.
  * @throws RemoteException
  */
```

```
public void testInserisciProdotto() throws RemoteException {
    QueryDataSet qds = new QueryDataSet();
    qds.setQuery(new
com.borland.dx.sql.dataset.QueryDescriptor(database,
        sqlRes.getString("Prodotti"), null, true,
        Load.ALL));
    qds.open();
    int size = qds.rowCount();
    int oracolo = size + 1;
    ProdottoBar p = new ProdottoBar(1, "Prova", 5.00, 0);
    sessioneBarGestoreImp.inserisciProdotto(p);
    qds.refresh();
    assertEquals(oracolo, qds.rowCount());
    assertTrue(qds.getString("Descrizione") != null);
    qds.close();
    qds = null;
    /**@todo fill in the test code*/
  }
```

```
 /**
  * Il metodo testa il caso in cui nella valida nome utente e password siano vuoti.
  * Il test avrà sccesso se la variabile oracolo avrà lo stesso valore di output.
  * @throws validaException
  */
  public void testValidaNoUIDPWD() throws validaException {
    String UID = "";
    String PWD = "";
    boolean oracolo = false;
    boolean output;
    try {
      output = sessioneBarGestoreImp.valida(UID, PWD);
    }
    catch (validaException ex) {
      output = false;
    }
    assertEquals("return value", oracolo, output);
    /**@todo fill in the test code*/
  }
```

```
 /**
  * Il metodo testa la valida utente.
  * @throws validaException
  */

 public void testValidaUIDPWD() throws validaException {
    String UID = "admin";
    String PWD = "admin";
    boolean oracolo = true;
    boolean output;
    try {
      output = sessioneBarGestoreImp.valida(UID, PWD);
    }
```

```
    catch (validaException ex) {
      output = false;
    }
    assertEquals("return value", oracolo, output);
    /**@todo fill in the test code*/
  }

}
```

5.3 Test Sessione Cliente

```
package QsServer.test;

import java.rmi.*;
import java.util.*;

import QsServer.Cliente.*;
import QsServer.DataLayer.*;
import Qsclassicomuni.adapters.*;
import Qsclassicomuni.collezioni.*;
import com.borland.dx.sql.dataset.*;
import junit.framework.*;

/**
 * <p>Title: TestSessioneClienteImp</p>
 * <p>Description: Tesa la sessione del cliente</p>
 * <p>Copyright: Copyright (c) 2008</p>
 * <p>Company: UNISA (Corso ingegneria del Software)</p>
 * @author Gruppo De Sio
 * @version 1.0
 */

public class TestSessioneClienteImp
    extends TestCase {
  Database database;
  ResourceBundle sqlRes;
  Camera camera;
  private SessioneClienteImp sessioneClienteImp = null;
  private double tot;

 /**
   * Inizializza le classi da testare
   * @throws Exception
   */

  protected void setUp() throws Exception {
    super.setUp();
    /**@todo verify the constructors*/
    database = new Database();
    camera = new Camera(2, 0, true, false, 5, 5);
    database.setConnection(new
com.borland.dx.sql.dataset.ConnectionDescriptor(
        "jdbc:odbc:QuickService", "", "", false,
"sun.jdbc.odbc.JdbcOdbcDriver"));
    sqlRes = ResourceBundle.getBundle("QsServer.SqlRes");
    sessioneClienteImp = new SessioneClienteImp(database, sqlRes,
camera);
  }
```

```
 /**
   * Il metodo svuota le collection e libera la memoria.
   * @throws Exception
   */

  protected void tearDown() throws Exception {
    sessioneClienteImp = null;
    super.tearDown();
  }

 /**
  * Il metodo testa la creazione di una nuova sessione cliente.
  * @throws RemoteException
  */

 public void testSessioneClienteImp() throws RemoteException {
    sessioneClienteImp = new SessioneClienteImp(database, sqlRes,
camera);
    /**@todo fill in the test code*/
  }

 /**
  * Testa che il metodo restituica i dati di una camera correttamente.
  */
  public void testGetDatiCamera() {
    Camera actualReturn = sessioneClienteImp.getDatiCamera();
    assertTrue("is not null", actualReturn != null);
    /**@todo fill in the test code*/
  }

 /**
  * Il metodo testa l'inserimento di nuove ordinazioni.
  * @throws DisponibilitaException
  */
  public void testOrdinazioneBar() throws DisponibilitaException {
    int room = 2;
    QsCollection collect = new QsCollectionAdapter();
    QueryDataSet qds = new QueryDataSet();
    qds.setQuery(new
com.borland.dx.sql.dataset.QueryDescriptor(database,
        sqlRes.getString("Prodotti")+ " WHERE disponibilita >0 OR
disponibilita =-1", null, true,
        Load.ALL));
    qds.open();


   for (int i = 0; i < 10 && qds.next(); i++) {
      collect.aggiungi(new DettaglioPrenotazioneBar(qds.getInt(0),
qds.getString(2),1,
                                          qds.getDouble(3), ""));

      double tot = +qds.getDouble(3);
```

```
      qds.next();
    }
    qds.close();
    sessioneClienteImp.ordinazioneBar(room, collect, tot);
    /**@todo fill in the test code*/
  }

 /**
  * Il metodo testa il caso in cui l'inserimento di nuove prenotazioni fallisce.
  * @throws DisponibilitaException
  */
  public void testOrdinazioneBarFallita() throws
DisponibilitaException {
   int room = 2;
   boolean oracolo = true;
   QsCollection collect = new QsCollectionAdapter();
   QueryDataSet qds = new QueryDataSet();
   qds.setQuery(new
com.borland.dx.sql.dataset.QueryDescriptor(database,
       sqlRes.getString("Prodotti")+ " WHERE disponibilita <0",
null, true,
       Load.ALL));
   qds.open();

     for (int i = 0; i < 10 && qds.next(); i++) {
     collect.aggiungi(new DettaglioPrenotazioneBar(qds.getInt(0),
qds.getString(2),1,
                                          qds.getDouble(3), ""));

     double tot = +qds.getDouble(3);
     qds.next();
   }
   qds.close();
  try {
    sessioneClienteImp.ordinazioneBar(room, collect, tot);
      }
  catch (DisponibilitaException ex){
  oracolo = false;
  assertFalse(oracolo);
  }
   /**@todo fill in the test code*/
 }
/**
 * IL metodo testa la visualizzazione delle categorie.
 * Se la dimensione è maggiore di 0 allora vengono visualizzate le categorie.
 * @throws RemoteException
 */
 public void testVisualizzaListinoCategorie() throws
RemoteException {
    int codCat = -1;
    QsCollection actualReturn =
sessioneClienteImp.visualizzaListino(codCat);
```

```
    assertTrue("the size is >0", actualReturn.dimensione() > 0);
    /**@todo fill in the test code*/
  }

 /**
  * Il metodo testa la visualizzazione dei prodotti.
  * Se la dimensione è maggiore di 0 allora vengono visualizzati i prodotti.
  * @throws RemoteException
  */
 public void testVisualizzaListinoProdotti() throws RemoteException
{
    int codCat = 1; // categoria presente
    QsCollection actualReturn =
sessioneClienteImp.visualizzaListino(codCat);
    assertTrue("the size is >0", actualReturn.dimensione() > 0);
    /**@todo fill in the test code*/
  }

}
```