



**Università degli Studi di Salerno**

**Corso di Ingegneria del Software**

**Mockbuster  
Test Case Specification  
Versione 1.0**

# **MOCKBUSTER**

Data: 17/01/2025

**Coordinatore del progetto:**

Nome	Matricola

**Partecipanti:**

Nome	Matricola
Roberto Ambrosino	0512117886

<b>Scritto da:</b>	Roberto Ambrosino
--------------------	-------------------

**Revision History**

Data	Versione	Descrizione	Autore
17/01/2025	1.0	Aggiunti casi di test e risultati dei test	Roberto Ambrosino

<b>1 – Casi di test</b>	<b>4</b>
<b>2 – Risultati dei test</b>	<b>10</b>

# 1 - Casi di test

---

## Accesso degli Utenti

### PARAMETRI

- username: indirizzo email dell'utente
- password: stringa per autenticazione

### OGGETTI DELL'AMBIENTE

- Database: contiene i dati degli utenti già registrati

Categorie	Scelte
Categoria 1: Email Presente	<ul style="list-style-type: none"><li>• EP1: Email presente nel database</li><li>• EP2: Email assente nel database</li></ul>
Categoria 2: Password	<ul style="list-style-type: none"><li>• PW1: Password corretta</li><li>• PW2: Password errata</li></ul>

TEST FRAME	
Combinazione scelte	Oracolo
EP1, PW1	Accesso effettuato con successo
EP1, PW2	Messaggio di errore: "Email e/o Password errata"
EP2, PW1	Messaggio di errore: " Email e/o Password errata"

### TEST CASE

- TC1: email: "pippo@example.com", password: "Pluto123" – in DB presente la coppia (email = "pippo@example.com", password = "Pluto123") – ORACOLO: Accesso effettuato con successo
- TC2: email: "pippo@example.com", password: "Topolino456" – in DB presente l'utente con email "pippo@example.com" ma password diversa – ORACOLO: Messaggio di errore " Email e/o Password errata"

- TC3: email: "nuovo@example.com", password: "Pluto123" – in DB assente l'utente con email "nuovo@example.com" – ORACOLO: Messaggio di errore "Email e/o Password errata"

## Registrazione degli Utenti

### PARAMETRI

- email: indirizzo email dell'utente
- password: stringa per autenticazione
- nome: nome dell'utente
- cognome: cognome dell'utente
- Indirizzo fatturazione: stringa che rappresenta l'indirizzo di fatturazione dell'utente

### OGGETTI DELL'AMBIENTE

- Database: contiene i dati degli utenti già registrati

Categorie	Scelte
Categoria 1: Formato Email	<ul style="list-style-type: none"> <li>• EM1: Email corretta</li> <li>• EM2: Email non corretta</li> </ul>
Categoria 2: Password	<ul style="list-style-type: none"> <li>• PW1: Password valida (lunghezza <math>\geq 8</math>)</li> <li>• PW2: Password NON valida</li> </ul>
Categoria 3: Nome e Cognome	<ul style="list-style-type: none"> <li>• NC1: Nome e cognome validi (non vuoti)</li> <li>• NC2: Nome e/o cognome non validi (vuoti)</li> </ul>
Categoria 4: Indirizzo di Fatturazione	<ul style="list-style-type: none"> <li>• IF1: Indirizzo valido (non vuoto)</li> <li>• IF2: Indirizzo non valido (vuoto)</li> </ul>

TEST FRAME	
Combinazione scelte	Oracolo
EM1, PW1, NC1, IF1	Messaggio di errore: "Email già registrata"
EM1, PW1, NC1, IF1	Registrazione effettuata con successo
EM2, PW1, NC1, IF1	Messaggio di errore: "Email non corretta"
EM1, PW2, NC1, IF1	Messaggio di errore: "Password non valida"

EM1, PW1, NC2, IF1	Messaggio di errore: "Nome non valido" o "Cognome non valido"
EM1, PW1, NC1, IF2	Messaggio di errore: "Indirizzo non valido"

## TEST CASE

- TC4: email: "pippo@example.com", password: "Pluto123", nome: "Pippo", cognome: "Rossi", indirizzo fatturazione: "Via Roma 12 40100 Bologna" – in DB presente l'utente con email "pippo@example.com" – ORACOLO: Messaggio di errore "Email già registrata"
- TC5: email: "nuovo@example.com", password: "Pluto123", nome: "Mario", cognome: "Bianchi", indirizzo fatturazione: "Via Roma 12 40100 Bologna" – in DB assente l'utente con email "nuovo@example.com" – ORACOLO: Registrazione effettuata con successo
- TC6: email: "errata", password: "Pluto123", nome: "Luigi", cognome: "Verdi", indirizzo fatturazione: "Via Roma 12 40100 Bologna" – ORACOLO: Messaggio di errore "Email non valida"
- TC7: email: "nuovo@example.com", password: "aa", nome: "Giulia", cognome: "Ferrari", indirizzo fatturazione: "Via Roma 12 40100 Bologna" – ORACOLO: Messaggio di errore "Password non valida"
- TC8: email: "nuovo@example.com", password: "Pluto123", nome: "", cognome: "Verdi", indirizzo fatturazione: "Via Roma 12 40100 Bologna" – ORACOLO: Messaggio di errore "Nome non valido"
- TC9: email: "nuovo@example.com", password: "Pluto123", nome: "Anna", cognome: "Bianchi", indirizzo fatturazione: "" – ORACOLO: Messaggio di errore "Indirizzo non valido"

## Visualizzazione e Ricerca nel Catalogo

### PARAMETRI

- query: stringa di ricerca per titolo

### OGGETTI DELL'AMBIENTE

- Database: contiene i dati del catalogo

Categorie	Scelte
Categoria 1: Query	<ul style="list-style-type: none"> <li>Q1: Query valida (titolo esistente)</li> <li>Q2: Query non valida (titolo inesistente)</li> <li>Q3: Query vuota</li> </ul>

TEST FRAME	
Combinazione scelte	Oracolo
Q1	Visualizzazione dei film corrispondenti
Q2	Messaggio di "Nessun risultato trovato"
Q3	Visualizzazione dell'intero catalogo

### TEST CASE

- TC10: query: "Inception" – presente nel catalogo – ORACOLO: Visualizzazione del film "Inception"
- TC11: query: "FilmNonEsistente" – assente nel catalogo – ORACOLO: Messaggio "Nessun risultato trovato"
- TC12: query: "" (vuota) – ORACOLO: Visualizzazione dell'intero catalogo

## Gestione del Carrello e Finalizzazione degli Ordini

### PARAMETRI

- carrello: memorizza i film selezionati per l'acquisto o il noleggio
- film: film da aggiungere o rimuovere

### OGGETTI DELL'AMBIENTE

- Carrello: memorizza i film selezionati per l'acquisto o il noleggio

Categorie	Scelte
Categoria 1: Stato del Carrello	<ul style="list-style-type: none"> <li>C1: Carrello vuoto</li> <li>C2: Carrello con articoli</li> </ul>
Categoria 2: Azione	<ul style="list-style-type: none"> <li>A1: Aggiungere un articolo</li> <li>A2: Rimuovere un articolo</li> <li>A3: Procedere con l'ordine</li> </ul>

TEST FRAME	
Combinazione scelte	Oracolo
C1, A1	Film aggiunto con successo
C2, A2	Film rimosso con successo
C1, A3	Messaggio di errore: "Carrello vuoto"
C2, A3	Ordine completato con successo

### TEST CASE

- TC13: carrello vuoto, azione: aggiungi film "Inception" – ORACOLO: Film aggiunto con successo
- TC14: carrello con "Inception", azione: rimuovi film "Inception" – ORACOLO: Film rimosso con successo
- TC15: carrello vuoto, azione: procedi all'ordine – ORACOLO: Messaggio di errore "Carrello vuoto"
- TC16: carrello con "Inception", azione: procedi all'ordine – ORACOLO: Ordine completato con successo

---

## Gestione del Catalogo da Parte dei Gestori

### PARAMETRI

- film: dettagli del film (titolo, descrizione, prezzo, numero licenze)

### OGGETTI DELL'AMBIENTE

- Database: contiene i dati del catalogo



<b>Categorie</b>	<b>Scelte</b>
Categoria 1: Azione	<ul style="list-style-type: none"> <li>• A1: Aggiungere un film</li> <li>• A2: Modificare un film</li> <li>• A3: Rimuovere un film</li> </ul>
Categoria 2: Validità Dati	<ul style="list-style-type: none"> <li>• VD1: Dati validi</li> <li>• VD2: Dati non validi</li> </ul>

<b>TEST FRAME</b>	
<b>Combinazione scelte</b>	<b>Oracolo</b>
A1, VD1	Film aggiunto con successo
A2, VD2	Messaggio di errore: "Dati non validi"
A3, VD1	Film rimosso con successo

## TEST CASE

- TC17: azione: aggiungi film "Inception" con dati validi – ORACOLO: Film aggiunto con successo
- TC18: azione: modifica film "Inception" con dati invalidi – ORACOLO: Messaggio di errore "Dati non validi"
- TC19: azione: rimuovi film "Inception" – ORACOLO: Film rimosso con successo

## 2 - Risultati dei test

---

```
-----
T E S T S
-----
Running control.filters.MovieUpdateFilterTest
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.586 s -- in control.filters.MovieUpdateFilterTest
Running control.filters.SignupFormFilterTest
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.041 s -- in control.filters.SignupFormFilterTest
Running control.filters.PlaceOrderFilterTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.022 s -- in control.filters.PlaceOrderFilterTest
Running control.admin.MovieUpdateServletTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.057 s -- in control.admin.MovieUpdateServletTest
Running control.common.SignupServletTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 s -- in control.common.SignupServletTest
Running control.common.LoginServletTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 s -- in control.common.LoginServletTest
Running control.browse.UpdateCartServletTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.058 s -- in control.browse.UpdateCartServletTest
Running control.browse.PlaceOrderServletTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.051 s -- in control.browse.PlaceOrderServletTest
Running control.browse.SearchMovieTitleServletTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.010 s -- in control.browse.SearchMovieTitleServletTest

Results:

Tests run: 17, Failures: 0, Errors: 0, Skipped: 0
```

I test sono stati eseguiti utilizzando JUnit e Mockito.

- JUnit è stato utilizzato per strutturare ed eseguire i test unitari, garantendo la copertura delle principali funzionalità dei metodi.
- Mockito è stato impiegato per simulare le dipendenze esterne, come servizi o persistenza, al fine di isolare il comportamento delle classi e verificare il corretto funzionamento del codice senza dipendere da implementazioni reali.

Tutti i test hanno fornito esiti positivi, senza rilevare particolari errori o malfunzionamenti. Le operazioni sono risultate conformi alle specifiche definite nei test case.