Università degli Studi di Salerno
Corso di Ingegneria del Software

AniTour
Test Case Specification
Versione 1.2

Data: 18/02/2026

Progetto: AniTour	Versione: 1.2
Documento: Test Case Specification	Data: 18/02/2026

Indice

0. Test Case Derivation (Category Partition Method)

Questa sezione preliminare documenta l'analisi svolta per derivare i valori di input della sezione 3 ("Input specifications").

Per la funzionalità `processCheckout` del sottosistema Booking, sono state identificate le seguenti categorie e scelte:

Categoria	Scelta (Choice)	ID	Vincolo
Stato carrello	Vuoto	C1	[Error]
	Contiene tour disponibile	C2	[Valid]
	Contiene tour esaurito	C3	[Error]
Pagamento	Carta valida	P1	[Valid]
	Carta rifiutata	P2	[Error]
Autenticazione	Utente loggato	U1	[Valid]

1. Test Case 01: Acquisto con successo (Happy path)

1.1. Test case specification identifier TC-BOOK-01

Checkout di un Tour disponibile con pagamento valido.

1.2. Test items

- **Componente:** intero sottosistema `Booking` (JSP + Servlet + DB)
- **Feature:** `processCheckout(sessionToken, paymentData)`
- **Obiettivo:** verificare il flusso completo di acquisto dalla UI al Database.

1.3. Input specifications

Pre-condizioni: Tour “Persona 5” (ID 3) configurato nel DB con available_seats > 0.
Azioni (Selenium):

1. Utente loggato (U1)
2. Navigazione alla pagina /home.
3. Selezione del Tour "Persona 5: Tokyo Phantom Thieves".
4. Click su "PRENOTA ORA".
5. Compilazione Form Checkout:
    - Email: mariorossi@email.com.
    - Carta: 1234-5678-9012-3456 (P1).
    - Stato Carrello: Implicito (C2).
6. Click su "Conferma Ordine".

1.4. Output specifications

- **Interfaccia (frontend):**
    - Redirect a `success.jsp`.
    - Messaggio “Ordine completato” visibile.
- **Database (backend):**
    - Nuovo record in tabella bookings.
    - Decremento posti per tour ID 3.

1.5. Environmental needs

- **Hardware:** PC di sviluppo standard.
- **Software:** Java 17, JUnit 5.
- **Test drivers:** JUnit Test Runner per invocare il metodo `processCheckout`.
- **Browser:** Google Chrome controllato da Selenium WebDriver.
- **Database:** MySQL Server attivo con dati di seed.
- **Server:** Apache Tomcat in esecuzione.

1.6. Special procedural requirements

- **Setup:** prima del test, il database deve essere pulito e popolato con il record del Tour “Persona 5” (50 posti).
- **Teardown:** al termine, i dati di test devono essere rimossi per non influenzare i test successivi (rollback automatico).

1.7. Intercase dependencies

Nessuna.

2. Test Case 02: Fallimento per posti esauriti (Concurrency)

2.1. Test case specification identifier TC-BOOK-02

Tentativo di checkout su Tour divenuto sold out.

2.2. Test items

- **Componente:** `BookingControl`
- **Feature:** Gestione eccezioni in `processCheckout`
- **Obiettivo:** verificare che il sistema impedisca la creazione dell'ordine e non decrementi lo stock se il tour richiesto ha posti disponibili pari a zero (gestione concorrenza/sold Out).

2.3. Input specifications

Pre-condizioni: Tour “Sekiro” (ID 2) configurato nel DB con available_seats = 0.
Azioni (Selenium):

1. Utente loggato (U1)
2. Navigazione alla pagina /home.
3. Selezione del Tour "Sekiro".
4. Click su "PRENOTA ORA".
5. Compilazione Form Checkout:
    - Email: mariorossi@email.com.
    - Carta: 1234-5678-9012-3456 (P1).
    - Stato Carrello: Implicito (C3).
6. Click su "Conferma Ordine".

2.4. Output specifications

- **Interfaccia (frontend):**
    - Permanenza `checkout.jsp`.
    - Visualizzazione banner errore SoldOutException.
- **Database (backend):**
    - Nessun ordine creato.

2.5. Environmental needs

- Vedi specifica *TC-BOOK-01* (l’ambiente è condiviso).

2.6. Special procedural requirements

Nessuna.

2.7. Intercase dependencies

Nessuna.

3. Test Case 03: Pagamento rifiutato

3.1. Test case specification identifier TC-BOOK-03

Gestione del rifiuto transazione da parte della banca.

3.2. Test items

- **Componente:** `BookingControl`
- **Feature:** Integrazione con `PaymentGateway`.
- **Obiettivo:** assicurarsi che nessuna prenotazione venga salvata nel db se il servizio di pagamento esterno (`PaymentGateway`) restituisce un esito negativo (es. carta rifiutata).

3.3. Input specifications

Azioni (Selenium):

1. Utente loggato (U1)
2. Navigazione alla pagina /home.
3. Selezione del Tour "Persona 5".
4. Click su "PRENOTA ORA".
5. Compilazione Form Checkout:
    - Email: mariorossi@email.com.
    - Carta: 1234-1234-1234-9999 (P2).
    - Stato Carrello: Implicito (C2).
6. Click su "Conferma Ordine".

3.4. Output specifications

- **Interfaccia (frontend):**
    - Permanenza `checkout.jsp`.
    - Visualizzazione banner errore PaymentFailedException.
- **Database (backend):**
    - Nessun ordine creato.

3.5. Environmental needs

- Vedi specifica *TC-BOOK-01* (l'ambiente è condiviso).

3.6. Special procedural requirements

Nessuna.

3.7. Intercase dependencies

Nessuna.

4. Test Case 04: Carrello vuoto

4.1. Test case specification identifier TC-BOOK-04

Tentativo di checkout con carrello vuoto. Usiamo una email trigger per simulare la condizione C1.

4.2. Test items

- **Componente:** `BookingControl`
- **Feature:** validazione pre-condizioni in `processCheckout`.
- **Obiettivo:** verificare che il meccanismo di validazione intercetti le pre-condizioni invalide (lista articoli vuota) lanciando l'eccezione appropriata prima di tentare qualsiasi interazione con il db o il sistema di pagamento.

4.3. Input specifications

Azioni (Selenium):

1. Utente loggato (U1)
2. Navigazione alla pagina /prenota?tourId=3.
3. Compilazione Form Checkout:
    - Email: empty@test.com (trigger)
    - Carta: 1234-5678-9012-3456 (P1).
    - Stato Carrello: Implicito (C1).
4. Click su "Conferma Ordine".

4.4. Output specifications

- **Interfaccia (frontend):**
    - Permanenza `checkout.jsp`.
    - Visualizzazione banner errore “Carrello vuoto”.
- **Database (backend):**
    - Nessun ordine creato.

4.5. Environmental needs

- Vedi specifica *TC-BOOK-01* (l’ambiente è condiviso).

4.6. Special procedural requirements

Nessuna.

4.7. Intercase dependencies

Nessuna.

Revision History

Data	Versione	Descrizione	Autore
22/12/2025	1.0	Creazione documento Test Case Specification	Vincenzo Chiocca
22/12/2025	1.1	Aggiunto TC-BOOK-04 mancante al documento. Aggiunti obiettivi mancanti ai test case.	Vincenzo Chiocca
18/02/2026	1.2	Corrette tutte le sezioni per coerenza con testing Selenium	Vincenzo Chiocca

Partecipanti:

Nome	Matricola
Vincenzo Chiocca	**0512119182**
Salvatore Merola	**0512120979**