

Università degli Studi di Salerno
Corso di Ingegneria del Software

**AniTour
Test Plan
Versione 1.0**



Data: 19/12/2025

| | |
|----------------------|------------------|
| Progetto: AniTour | Versione: 1.0 |
| Documento: Test Plan | Data: 19/12/2025 |

Indice

1. Introduction
2. Relationship to other documents
3. System overview
4. Features to be tested/not to be tested
5. Pass/Fail criteria
6. Approach
7. Suspension and resumption
8. Testing materials (hardware/software requirements)
9. Test cases
10. Testing schedule

1. Introduction

Il presente Test Plan descrive la strategia di verifica per il sistema **AniTour**, con un focus specifico sulla validazione del **Booking Subsystem**. L'obiettivo principale è garantire che il flusso critico di acquisto sia robusto, gestisca correttamente le transazioni ACID e i fallimenti di pagamento, assicurando la qualità del software prima del rilascio.

2. Relationship to other documents

Questo piano di test è strettamente correlato ai seguenti documenti di progetto:

- **RAD (Requirement Analysis Document):** I test di sistema verificano il soddisfacimento dei Requisiti Funzionali (in particolare il Caso d'Uso "Acquisto Biglietto").
- **ODD (Object Design Document):** I test di unità e integrazione verificano le interfacce specificate nella Sezione 3 dell'ODD (es. BookingControl, IBookingRepository).
- **Test Case Specification:** I dettagli operativi di ogni test case elencato nella Sezione 9 sono definiti nel documento separato di specifica.

3. System Overview

Il focus dei test è il **Booking Subsystem**, composto dai seguenti componenti principali oggetto di Unit Test:

- **BookingControl:** Gestisce la logica di business e il coordinamento della transazione.
- **Cart:** Gestisce lo stato temporaneo della sessione utente.
- **IBookingRepository (DAO):** Gestisce la persistenza su database relazionale.
- **IPaymentGateway:** Adattatore per il sistema di pagamento esterno.

4. Features to be tested/not to be tested

Features to be tested:

- Processo di Checkout (processCheckout).
- Gestione della concorrenza (stock esaurito durante l'acquisto).
- Integrazione con il Database (salvataggio ordine).
- Gestione delle risposte del Gateway di Pagamento (Successo/Fallimento).

Features not to be tested:

- **Registrazione Utente:** Appartiene allo User Subsystem, fuori dallo scope di questa consegna.
- **Transazioni Bancarie Reali:** Verranno utilizzati oggetti Mock (Mockito) per simulare le risposte bancarie, per evitare addebiti reali durante i test.

5. Pass/Fail criteria

- **Unit Tests:** Il 100% dei test JUnit deve passare (Green bar). Copertura delle istruzioni (Statement Coverage) > 80%.
- **Integration Tests:** Le transazioni devono garantire l'atomicità (nessun ordine parziale salvato nel DB in caso di errore).

- **System Tests:** Il sistema deve rispondere entro 2 secondi (come da Requisiti Non Funzionali) e produrre l'output atteso descritto nei Test Cases.

6. Approach

Unit Testing Strategy: Verrà adottato un approccio **White Box** per testare i singoli metodi, utilizzando **Stub e Driver** per isolare le classi.

Integration Testing Strategy: Verrà adottato l'approccio **Sandwich Testing**, combinando:

- **Top-Down:** Per testare l'interazione tra i Controller e l'interfaccia utente (simulata).
- **Bottom-Up:** Per testare l'integrazione tra i Controller e il livello di persistenza (DAO/Database). Questa strategia è stata scelta per mitigare i rischi sia a livello di interfaccia che di persistenza dati in parallelo.

System Testing Strategy (Test Case Selection): Per la selezione dei casi di test funzionali è stato applicato il **Category Partition Method**. Sono state identificate le categorie di input (es. Stato Carrello, Validità Carta) e le relative scelte per generare combinazioni di test significative, riducendo la ridondanza e massimizzando la copertura.

7. Suspension and resumption

Il testing verrà sospeso se viene individuato un **Blocking Bug** nel metodo processCheckout che impedisce la finalizzazione di qualsiasi ordine (es. Crash del Database o NullPointerException sistematico). I test riprenderanno solo dopo il fix del bug critico e un successivo Smoke Test di verifica.

8. Testing materials (hardware/software requirements)

- **Hardware:** PC di sviluppo standard (Windows/Mac/Linux).
- **Software:**
 - Java JDK 17+.
 - IDE: IntelliJ IDEA o Eclipse.
 - Testing Framework: JUnit 5.
 - Mocking Framework: Mockito.
 - Database: MySQL 8.0 (Schema di Test dedicato).

9. Test cases

Di seguito l'elenco dei Test Case identificati (i dettagli completi di Input/Output sono nel documento *Test Case Specification*):

- **TC-BOOK-01:** Acquisto Confermato (Happy Path).
- **TC-BOOK-02:** Fallimento per Tour Esaurito (Test di Concorrenza).
- **TC-BOOK-03:** Pagamento Rifiutato dalla Banca.

- **TC-BOOK-04:** Tentativo Checkout con Carrello Vuoto.

10. Testing schedule

- **Responsabile Test:** [Tuo Nome/Gruppo]
- **Esecuzione Unit & Integration Tests:** Entro il 14 Dicembre 2025.
- **Esecuzione System Tests:** Entro il 15 Dicembre 2025.
- **Consegna Documentazione:** 16 Dicembre 2025.

Revision History

| Data | Versione | Descrizione | Autore |
|------------|----------|-------------------------------|------------------|
| 19/12/2025 | 1.0 | Creazione documento Test Plan | Salvatore Merola |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Partecipanti:

| Nome | Matricola |
|------------------|------------|
| Vincenzo Chiocca | 0512119182 |
| Salvatore Merola | 0512120979 |