

Università degli Studi di Salerno
Corso di Ingegneria del Software

**AniTou
Test Plan
Versione 1.1**

AniTou

Data: 22/12/2025

Progetto: AniTouR	Versione: 1.1
Documento: Test Plan	Data: 22/12/2025

Indice

1.	Introduction	4
2.	Relationship to other documents	4
3.	System Overview.....	4
4.	Features to be tested/not to be tested	4
5.	Pass/Fail criteria	4
6.	Approach	5
7.	Suspension and resumption	5
8.	Testing materials (hardware/software requirements).....	5
9.	Test cases	6
10.	Testing schedule	6
10.1.	Responsibilities	6
10.2.	Schedules and milestones	7
10.3.	Risks and contingencies.....	8

1. Introduction

Il presente Test Plan descrive la strategia di verifica per il sistema **AniTou**, 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 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.
 - 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

10.1. Responsibilities

I ruoli per la fase di testing sono così ripartiti all'interno del team di sviluppo:

- **Test Manager (Salvatore Merola):** responsabile della redazione del Test Plan, della configurazione dell'ambiente (JUnit/Mockito) e della verifica finale dei report.
- **Developer/Tester (tutto il team):** ogni membro del team è responsabile della scrittura dei Unit Test per le classi che implementa.
- **Integration Specialist (Vincenzo Chiocca):** responsabile dell'esecuzione dei test di integrazione tra Controller e Database (DAO).

10.2. Schedules and milestones

Le attività seguiranno un approccio incrementale, partendo dai test di unità fino ai test di sistema.

ID	Fase di Test	Attività Principali	Inizio	Fine	Deliverable (Output)
T01	Setup ambiente	Configurazione repository Git, JUnit 5, Mockito e Database di Test locale (MySQL).	22/12/2025	23/12/2025	Ambiente funzionante
T02	Unit Testing e sviluppo	Sviluppo classi BookingControl, Cart e relativi test JUnit (White Box). Mocking delle dipendenze esterne.	27/12/2025	04/01/2026	100% pass su Unit Tests
T03	Integration Testing	Implementazione DAO. Test "Sandwich" tra Controller e DB reale. Verifica transazioni ACID.	05/01/2026	08/01/2026	Integration Test report
T04	System Testing	Esecuzione dei Test Case funzionali (TC-BOOK-01...04) derivati col metodo Category Partition.	09/01/2026	10/01/2026	Bug log e fix
T05	Reporting	Raccolta evidenze (screenshot, log), redazione del <i>Test Execution Report</i> finale.	11/01/2026	12/01/2026	Documento esecuzione test
T06	Consegna finale	Revisione e sottomissione degli elaborati.	13/01/2026	14/01/2026	Consegna progetto

10.3. Risks and contingencies

- **Periodo festivo:** la pianificazione tiene conto della pausa natalizia, concentrando le attività intensive di testing dopo l'Epifania (dal 7 gennaio in poi) per garantire la disponibilità del team prima della scadenza del 12/01.

Revision History

Data	Versione	Descrizione	Autore
19/12/2025	1.0	Creazione documento Test Plan	Salvatore Merola
22/12/2025	1.1	Aggiunti maggiori dettagli alla sezione 10	Salvatore Merola

Partecipanti:

Nome	Matricola
Vincenzo Chiocca	0512119182
Salvatore Merola	0512120979