Università degli Studi di Salerno
Corso di Ingegneria del Software

AniTour
Test Plan
Versione 1.2

Data: 18/02/2026

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

Indice

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).
- Simulazione Pagamenti End-to-End:** Verifica del comportamento del sistema con carte di credito che generano esiti diversi (Successo vs Fallimento) tramite l'implementazione `RealPaymentGateway`.

Features not to be tested:

- **Registrazione Utente:** Appartiene allo User Subsystem, fuori dallo scope di questa consegna.

5. Pass/Fail criteria

- **Unit Tests:** Il 100% dei test JUnit deve passare (Green bar).
- **Integration Tests:** Le transazioni devono garantire l'atomicità e la persistenza corretta sul DB MySQL.

- **System Tests (Selenium):**
    - I test automatizzati devono completare l'esecuzione senza timeout o eccezioni `NoSuchElementException`.
    - L'interfaccia deve mostrare gli elementi visivi attesi (es. Banner di Errore in caso di Sold Out, Icona di Successo in caso di acquisto).

6. Approach

Unit Testing Strategy:

- **Approccio: White Box Testing**
- **Obbiettivo:** Verificare la correttezza della logica interna delle classi di dominio (Model) e delle classi di utilità, isolandole dalle dipendenze esterne.
- **Tecnica:** Sono stati sviluppati test automatizzati con **JUnit 5** per coprire i percorsi di esecuzione dei metodi nelle classi Entity (es. `Cart`, `Booking`, `PaymentDTO`).
- **Criterio:** Copertura delle istruzioni (Statement Coverage) e verifica dei valori limite (Boundary Value Analysis) per i campi dati.

Integration Testing Strategy:

- **Approccio: Bottom-Up Integration.**
- **Obbiettivo:** Verificare la corretta interazione tra il livello applicativo (Java) e il livello di persistenza (Database Relazionale).
- **Tecnica:** Invece di utilizzare Mock Objects per il database, è stata scelta una strategia di **Integrazione reale** con un'istanza MySQL di test. I test (es. `BookingDAOIntegrationTest`) verificano che le query SQL (CRUD) e le transazioni (es. decremento posti) vengano eseguite correttamente e persistano i dati come atteso.
- **Justification:** Questo approccio garantisce che eventuali disallineamenti tra codice Java e schema SQL vengano rilevati immediatamente.

System Testing Strategy:

- **Approccio: Automated End-To-End (E2E) Testing.**
- **Obbiettivo:** Validare i flussi funzionali completi (dal Frontend al Backend) simulando il comportamento di un utente reale.
- **Metodologia di Selezione:** I casi di test sono stati derivati utilizzando il **Category Partition Method**, identificando le classi di equivalenza per gli input (es. **Tour Disponibile vs Esaurito**, **Carta Valida vs Rifiutata**).

- **Tecnica di Esecuzione:** L'esecuzione è stata automatizzata tramite il framework **Selenium WebDriver.** Gli script di test interagiscono direttamente con l'interfaccia Web (JSP) su server Tomcat attivo, verificando:
    - **Output Visivo:** Presenza di elementi UI attesi (Redirect, Banner di errore, Conferme).
    - **Stato del Sistema:** Verifica della coerenza dei dati nel Database post-esecuzione (State Verification).

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.
    - **Automazione Browser:** Selenium WebDriver 4.x
    - **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. Responsabilities

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 (Automation)	Sviluppo ed esecuzione script **Selenium** per i casi d'uso (TC-BOOK-01...04). Verifica allineamento Frontend /Backend /Database.	09/01/2026	10/01/2026	Bug log e fix
T05	Reporting	Raccolta evidenze (screenshot, log), redazione del Test Execution Report finale.	11/01/2026	12/01/2026	Documento esecuzione test
T06	Consegna finale	Revisione e sottomissione degli	13/01/2026	14/01/2026	Consegna progetto

		elaborati.			

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
18/02/2025	1.2	Aggiornamento Test Plan (TP) per strategia Vertical Slice	Salvatore Merola

Partecipanti:

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