



INFINITY GAMES

# Test Plan

## **Membri del Team**

Falciano Ciro	0512103596
Schiroso Alessandro	0512104514
Ruggiero Simone	0512106074
De Lucia Antonino	0512105366

## REVISION HISTORY

DATA	VERSIONE	DESCRIZIONE	AUTORE
02/01/21	1.0	Prima versione del documento.	C. Falciano, S. Ruggiero, A. Schirosa, A. De Lucia

## SOMMARIO

1. Introduzione
2. Relazione con gli altri documenti
  - 2.1. Relazioni con il Requirements Analysis Document (RAD)
  - 2.2. Relazioni con il System Design Document (SDD)
  - 2.3. Relazioni con l'Object Design Document (ODD)
3. Panoramica del sistema
4. Funzionalità da testare e non
5. Pass/Fall criteria
6. Approccio
  - 6.1. Testing di unità
  - 6.2. Testing di integrazione
    - 6.2.1. Componenti da Testare
  - 6.3. Testing di Sistema
  - 6.4. Criteri di successo
7. Strumenti per il testing
8. Test Cases
  - 8.1. Gestione Utente
  - 8.2. Gestione Amministratore
  - 8.3. Gestione Prodotto
  - 8.4. Gestione Vista Prodotto
  - 8.5. Gestione Categoria
  - 8.6. Gestione Ordine
9. Test schedule

# **1. Introduzione**

Lo scopo di questo documento è l'analisi e la gestione dell'attività di testing riguardante la piattaforma ***Infinity Games***.

Viene verificato il corretto funzionamento del sistema analizzando i singoli casi presi dai test, e comparando i loro output con il presunto oracolo dei risultati. Da questa comparazione sarà possibile riscontrare gli eventuali errori e problemi da dover correggere con delle opportune modifiche.

Il processo di testing viene iterato fino a che non si soddisfano in maniera accettabile i requisiti funzionali e non funzionali del sistema.

## **2. Relazione con gli altri documenti**

Per l'individuazione corretta dei test casi si è fatto riferimento ai documenti precedentemente stilati. Infatti, la fase di testing è legata alle fasi ad essa precedenti; ogni documento sarà un punto di partenza indispensabile per poter effettuare un testing corretto ed adeguato e per verificare che il sistema desiderato sia simile a quello proposto.

### **2.1. Relazioni con il Requirements Analysis Document (RAD)**

Al fine di verificare la correttezza di Infinity Games sono stati predisposti dei test basati sulle funzionalità individuate nel RAD, in particolare riguardanti i requisiti funzionali e non funzionali del sistema.

### **2.2. Relazioni con System Design Document (SDD)**

Al fine di verificare la tracciabilità di Infinity Games, sono stati predisposti dei test sulle singole decomposizioni del sistema individuate nel SDD.

### **2.3 Relazioni con l'Object Design Document (ODD)**

Al fine di verificare le performance di Infinity Games, sono stati predisposti dei test per la verificabilità del funzionamento delle interfacce specificate nell'ODD.

### **3. Panoramica del Sistema**

Il sistema ha lo scopo di rendere gli acquisti di prodotti di stampo videoludico comodi, veloci e di facile accesso.

Il sistema dà la possibilità all'utente di poter acquistare videogiochi, mettendo a disposizione delle funzionalità che permettono ad esso di gestire i propri acquisti e visualizzare i propri prodotti in una libreria personale. I gestori della piattaforma avranno la possibilità di visualizzare le informazioni degli utenti e i loro relativi ordini, oltre che aggiungere e modificare i prodotti della piattaforma.

### **4. Funzionalità da testare e non**

#### **Gestione Utente**

- TC\_GU1:** Registrazione (SI)
- TC\_GU2:** Login (SI)
- TC\_GU3:** Logout (NO)
- TC\_GU4:** Visualizza Area Personale (NO)
- TC\_GU1:** Modifica Profilo (SI)

#### **Gestione Amministratore**

- TC\_GA1:** Promuovi Utente ad Amministratore (NO)
- TC\_GA2:** Declassa Amministratore (NO)
- TC\_GA3:** Rimuovi Utente (NO)

#### **Gestione Prodotto**

- TC\_GP1:** Aggiungi Prodotto al Carrello (NO)
- TC\_GP2:** Aggiungi Prodotto alla Lista dei Desideri (NO)
- TC\_GP3:** Ricerca Prodotto nello Store (SI)
- TC\_GP4:** Aggiungi Prodotto allo store da Amministratore (SI)
- TC\_GP5:** Modifica Prodotto da Amministratore (SI)

#### **Gestione Vista Prodotto**

- TC\_GVP1:** Visualizza Pagina Prodotto (NO)
- TC\_GVP2:** Visualizza prodotti per Categoria (NO)
- TC\_GVP3:** Visualizza Libreria personale (NO)
- TC\_GVP4:** Visualizza Lista dei Desideri (NO)
- TC\_GVP5:** Visualizza Carrello (NO)

## Gestione Categoria

**TC\_GC1:** Aggiungi Categoria allo store da Amministratore (SI)

**TC\_GC2:** Modifica Categoria da Amministratore (SI)

**TC\_GC3:** Rimuovi Categoria da Amministratore (NO)

## Gestione Ordine

**TC\_GO1:** Acquisto Prodotto (SI)

**TC\_GO2:** Visualizza Ordini effettuati (NO)

**TC\_GO3:** Visualizza Ordini Utenti da Amministratore (NO)

## 5. Pass/Fail Criteria

Lo scopo del testing è quello di dimostrare la presenza di faults (errori) all'interno del sistema. Le attività di testing, infatti, saranno mirate all'identificazione di questi faults e ad un successivo intervento per eliminarne la presenza.

Un test avrà successo se, dato un input al sistema, l'output osservato sarà diverso dall'output atteso dall'oracolo, quindi è stata identificata una failure. In tal caso questa verrà analizzata e si procederà alla correzione.

Al contrario, un test fallirà quando l'output osservato sarà uguale a quello presente nell'oracolo.

Per oracolo si intende il risultato atteso, in base ai requisiti, dall'esecuzione di un caso di test.

Il testing riterrà il sistema funzionante quando la coverage di testing avrà raggiunto il 75% per la verifica delle funzionalità di controllo del sistema.

## 6. Approccio

Per il sistema Infinity Games, il testing si compone tre fasi.

Nella prima fase, verranno eseguiti i test di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità andando a constatare il corretto funzionamento. Questa fase verrà effettuata al completamento di ogni unità realizzata per poter individuare tempestivamente gli errori presenti nel codice.

Nella seconda fase, verrà effettuato il testing di integrazione in cui si andrà a testare l'integrazione dei vari sottosistemi.

Infine, verrà eseguito il testing di sistema che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti.

Nella sessione di testing del sistema verrà utilizzato un approccio “BLACK BOX” ed una strategia top down: tale strategia prevede che i sottosistemi nel layer più in alto della gerarchia vengano testati individualmente, successivamente vengono integrati i layer più in basso. Si ripete questo passo finché tutti i sottosistemi sono testati. Tale metodologia fa uso dei test stub. Un test stub è una implementazione parziale di una componente chiamata dall’unità testata.

## 6.1. Testing di unità

Per effettuare il testing di ogni singola componente del sistema Infinity Games, verrà utilizzata la tecnica “Black-box testing” attraverso il framework JUnit. In questa fase, saranno analizzate le funzionalità dell’applicazione ed il comportamento delle singole componenti senza tener conto della loro struttura interna. Al fine di minimizzare il numero di test case, gli input saranno partizionati in classi di equivalenza. Quindi, per ogni classe di equivalenza verranno individuati dei vincoli a cui saranno associate delle scelte specifiche. I risultati del testing verranno analizzati e usati per correggere gli errori che causano il fallimento del sistema. Se si verifica un errore con dei risultati inattesi, si interviene in maniera tempestiva sulla componente in modo da renderla correttamente funzionante e procedere con le fasi di testing successive.

Per unità intendiamo un singolo servizio offerto dalle classi che compongono il back-end.

## 6.2. Testing di integrazione

Il Testing di Integrazione è una delle fasi di testing più importanti. Essa consiste nel trovare fault integrando differenti componenti insieme, fault che non sono stati trovati nello “Unit testing”. Questo documento ha il compito di identificare la strategia di testing di integrazione per il sistema Infinity Games.

Per verificare la corretta integrazione dei sottosistemi del sistema sono stati predisposti dei test case basati sulla divisione in sottosistemi proposta in fase di System Design.

### 6.2.1. Componenti da Testare

I componenti da testare sono stati scelti seguendo la strategia di testing Top-down. Verrà testato il Presentation Layer per l’interazione con l’utente. Per quanto riguarda l’Application Layer verranno testati i moduli dell’SDD:

Gestione Utente, Gestione Amministratore,  
Gestione Prodotto, Gestione Vista Prodotto, Gestione Categoria, Gestione

Ordine.

Per quanto riguarda il data access layer verrà testato: Database.

### **6.3. Testing di Sistema**

Lo scopo principale di questa fase di testing è quello di dimostrare che l'intero sistema funzioni correttamente e che soddisfi effettivamente i requisiti funzionali e non funzionali descritti nel documento di analisi dei requisiti (RAD). È da considerare l'attività più critica, in quanto può risultare molto complesso andare alla ricerca di eventuali errori, essendo impegnati tutti i sottosistemi.

### **6.4. Criteri di successo**

Il testing ha successo quando il comportamento osservato dal sistema implementato è diverso dal comportamento atteso specificato attraverso il modello del sistema. Avremo, quindi, un successo se il test individuerà una failure. In tal caso, questa verrà analizzata e si procederà alla sua eventuale correzione, dopo una correzione la fase di testing verrà reiterata per verificare che tale modifica non abbia impattato su altre componenti del sistema. Se il test non riesce ad individuare un errore parliamo di fallimento.

## 7. Strumenti per il Testing

Il testing verrà effettuato nell'Ambiente di sviluppo (IntelliJ) e sulle nostre macchine. Per il testing useremo il tool **JUnit** per IntelliJ e l'estensione per il browser Selenium per il testing di sistema.

Una tabella per annotare i test eseguiti:

Parametro Formato	
Categoria	Scelte

Codice	Combinazione	Esito