

Università degli Studi di Salerno
Corso di Ingegneria del Software

**Future Forge Gear
Test Plan
Versione 1.1**



Data: 28/01/2026

Progetto: Future Forge Gear	Versione: 1.0
Documento: Test Plan	Data: 19/12/2025

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Francesco Mascolo	0512117352
Francesco Lamanna	0512117166
Luigi Mazza	0512118612

Scritto da:	Francesco Mascolo
--------------------	-------------------

Revision History

Data	Versione	Descrizione	Autore
19/12/2025	0.1	Creazione del Documento Test Plan	Full Team
21/12/2025	0.2	Definizione Testing Schedule	Full Team
22/12/2025	1.0	Revisione documento	Full Team
28/01/2026	1.1	Piccole modifiche relative alle caratteristiche da Testare, sulla Panoramica del sistema e sull'Approccio che abbiamo utilizzato	Francesco Mascolo

Progetto: Future Forge Gear	Versione: 1.0
Documento: Test Plan	Data: 22/12/2025

Indice

1. Introduzione	4
1.1.Riferimenti	4
2. Relazione con altri documenti.....	4
3. Panoramica Del Sistema	5
4. Caratteristiche Da Testare	6
5. Pass/Fail Criteria	6
6. Approccio	7
7. Sospensione e Ripresa.....	8
8. Testing Materials.....	9
9. Test Case	9
10. Testing Schedule	10

Progetto: Future Forge Gear	Versione: 1.0
Documento: Test Plan	Data: 22/12/2025

1. Introduzione

Il seguente documento Test Plan definisce le strategie, i metodi e le risorse necessarie per eseguire il testing del sistema Future Forge Gear.

L'obiettivo principale è garantire che il software soddisfi i requisiti funzionali e non funzionali definiti nel documento Requirements Analysis Document (RAD) – Future Forge Gear.

Il piano di test ha lo scopo di individuare eventuali errori, malfunzionamenti o bug del sistema attraverso l'esecuzione di test unitari, test di integrazione e test di sistema, al fine di fornire un prodotto finale stabile, affidabile e conforme alle aspettative del cliente.

Il sistema Future Forge Gear, progettato come piattaforma e-commerce per la vendita di prodotti informatici, comprende funzionalità quali la gestione del catalogo prodotti, del carrello, degli ordini, degli utenti e delle richieste di approvvigionamento.

Il sistema è suddiviso in diversi sottosistemi, come descritto nel System Design Document, ciascuno dei quali richiede specifiche attività di testing per garantire la correttezza, l'affidabilità e l'efficienza delle funzionalità offerte. I test saranno eseguiti secondo approcci bottom-up e black-box per verificare sia i componenti atomici sia l'interazione fra di esse.

1.1. Riferimenti

Documenti:

RAD_FutureForgeGear (Requirements Analysis Document): Sono descritte le funzionalità individuate in fase di analisi.

SDD_FutureForgeGear (System Design Document): Contiene una descrizione completa dell'architettura del sistema, inclusi i sottosistemi individuati ed i servizi forniti da ciascun sottosistema.

ODD_FutureForgeGear (Object Design Document): Include dettagli sulle classi, i metodi e le interfacce.

2. Relazione con altri documenti

RAD_FutureForgeGear (Requirements Analysis Document):

In questo documento sono descritte le funzionalità individuate in fase di analisi, fornendo una descrizione dettagliata dei requisiti funzionali e non funzionali del sistema. Il piano di test fa riferimento a tali requisiti per verificare che tutte le funzionalità richieste siano correttamente implementate e rispettate.

SDD_FutureForgeGear (System Design Document):

Il documento descrive in modo completo l'architettura del sistema, basata su un modello three-tier, includendo i sottosistemi individuati e i servizi forniti da ciascun sottosistema. I test definiti nel presente piano sono progettati per verificare il corretto funzionamento dei singoli sottosistemi e la corretta interazione tra di essi, in accordo con quanto specificato nel SDD.

Progetto: Future Forge Gear	Versione: 1.0
Documento: Test Plan	Data: 22/12/2025

ODD_FutureForgeGear (Object Design Document):

Questo documento include i dettagli relativi alle classi, alle interfacce e ai metodi del sistema. Il Test Plan utilizza le specifiche contenute nell'ODD per la definizione dei test unitari, finalizzati alla verifica delle singole classi, e dei test di integrazione, volti a controllare la corretta cooperazione tra i componenti dei sottosistemi.

3. Panoramica Del Sistema

Il sistema Future Forge Gear è stato progettato per gestire la vendita online di prodotti informatici, includendo la gestione del catalogo, del carrello, degli ordini, degli utenti e delle richieste di approvvigionamento verso i fornitori.

L'architettura del sistema segue un modello three-tier (in forma semplificata), con una separazione logica tra livello di presentazione, logica applicativa e persistenza dei dati. Questa suddivisione migliora manutenibilità e modularità, facilitando anche le attività di test e manutenzione.

Presentation Layer

Il Presentation Layer gestisce l'interazione con l'utente finale tramite un'interfaccia web (JSP, HTML5, CSS, JavaScript). Gli utenti possono navigare il catalogo, autenticarsi, gestire il carrello, effettuare il checkout e consultare lo storico degli ordini. Gli utenti con ruoli specifici (magazziniere, fornitore, gestore ordini) accedono a funzionalità avanzate come gestione prodotti, richieste di approvvigionamento e aggiornamento stati ordine.

In questo livello rientrano anche i componenti che gestiscono le richieste HTTP (es. Servlet/controller), che raccolgono input dalla request/session e invocano i servizi applicativi.

Application Layer (Business)

L'Application Layer rappresenta il cuore della logica applicativa e coordina le operazioni del sistema. In questo strato sono presenti i servizi applicativi (ad es. AccountService, ProdottoService, OrdineService, RichiestaService), responsabili dell'esecuzione delle regole di business e dell'orchestrazione delle operazioni richieste dall'interfaccia.

Data Layer (Persistenza)

Il Data Layer si occupa della persistenza su database relazionale MySQL. L'accesso ai dati avviene tramite JPA, utilizzando l'EntityManager. Nel progetto, non è presente un layer DAO separato: i Service EJB svolgono sia la logica di business sia le operazioni di persistenza, interagendo con il database tramite NamedQuery e operazioni CRUD sulle entità.

Il flusso di comunicazione tra i livelli segue quindi il modello:

Presentation (JSP/Servlet) → Service (business + persistence) → Database (MySQL via JPA).

Progetto: Future Forge Gear	Versione: 1.0
Documento: Test Plan	Data: 22/12/2025

4. Caratteristiche Da Testare

4.1 caratteristiche da testare:

Funzionalità dell'Utente:

- Validazione di Email nella registrazione
- Validazione di password nella registrazione
- Aggiunta di prodotti al carrello
- Login tramite mail e password

Funzionalità del magazziniere:

- Richiesta di un prodotto
- Modifica del nome di un prodotto nel magazzino
- Modifica del prezzo di un prodotto nel magazzino

Funzionalità del fornitore:

- Creazione di un prodotto

Funzionalità del gestore degli ordini:

- Modifica stato dell'ordine

5. Pass/Fail Criteria

L'obiettivo dell'attività di testing è garantire che il sistema Future Forge Gear soddisfi pienamente i requisiti funzionali (RF) e i requisiti non funzionali (RNF) definiti nella documentazione di progetto. Ogni test è finalizzato a verificare che le funzionalità implementate operino correttamente e producano risultati coerenti rispetto agli input forniti. Il processo di testing si basa sul confronto tra il risultato atteso e il risultato effettivo (Oracolo) dell'operazione eseguita. Un test è considerato superato (PASS) se il comportamento osservato del sistema coincide con quello atteso. Al contrario, il test è considerato fallito (FAIL) nel caso in cui si riscontrino discrepanze tra il risultato ottenuto e quello previsto. L'individuazione degli errori avviene mediante il confronto con un oracolo di test, che rappresenta il comportamento corretto del sistema in base ai requisiti specificati. Qualora il sistema non soddisfi i criteri sopra elencati, sarà necessario procedere alla correzione del codice, alla riesecuzione dei test e alla successiva verifica, fino a garantire il pieno rispetto di tutti i requisiti definiti.

Progetto: Future Forge Gear	Versione: 1.0
Documento: Test Plan	Data: 22/12/2025

6. Approccio

L'approccio al testing per il sistema Future Forge Gear è articolato in tre fasi principali, finalizzate a garantire una verifica progressiva del sistema, partendo dalle singole componenti fino alla valutazione del comportamento complessivo della piattaforma.

Testing Unitario

La prima fase riguarda il **testing unitario**, focalizzato sulla verifica delle singole classi applicative, in particolare dei Service EJB, analizzandone i metodi principali in modo isolato rispetto all'interfaccia web.

In questa fase sono stati testati i metodi responsabili delle operazioni CRUD e delle principali regole di business (ad esempio validazioni, aggiornamenti di stato e gestione delle entità), verificando il corretto comportamento al variare degli input.

Sebbene i test coinvolgano il livello di persistenza, l'attenzione è posta sulla correttezza della logica implementata nei Service, controllando che ogni metodo produca l'effetto atteso e mantenga la coerenza dello stato delle entità.

Testing di Integrazione

La seconda fase consiste nel **testing di integrazione**, mirato a verificare le interazioni tra le diverse componenti del sistema.

In particolare, sono state testate le interazioni tra:

- Service EJB
- livello di persistenza JPA
- database

Per garantire l'isolamento dall'ambiente di produzione, è stato utilizzato un database **H2 in modalità in-memory**, configurato come *persistence unit* alternativa rispetto al database MySQL utilizzato in produzione.

Questo ha permesso di verificare il corretto funzionamento delle operazioni di persistenza tramite **JPA ed EntityManager**, assicurando che i moduli cooperino correttamente tra loro e che le operazioni sui dati risultino consistenti.

I test di integrazione hanno riguardato in particolare le funzionalità di:

- gestione utenti
- gestione catalogo prodotti
- gestione carrello
- gestione ordini
- gestione richieste di approvvigionamento
- gestione dei pagamenti

Progetto: Future Forge Gear	Versione: 1.0
Documento: Test Plan	Data: 22/12/2025

Testing di Sistema

La fase di **testing di sistema** è finalizzata alla verifica del comportamento complessivo della piattaforma Future Forge Gear, valutando il soddisfacimento dei principali requisiti funzionali definiti nel RAD.

I test di sistema sono orientati alla validazione dei principali flussi di utilizzo dell'applicazione dal punto di vista dell'utente, come:

- registrazione e autenticazione (Login, Registrazione Email, Registrazione Password)
- consultazione del catalogo (Modifica Nome Prodotto)
- gestione del carrello (Aggiunta al carrello)
- completamento di un ordine
- gestione delle richieste (Modifica Stato Ordine)

In questa fase il sistema viene considerato come un'unica entità integrata (frontend, logica applicativa e persistenza), al fine di individuare eventuali anomalie emergenti solo a seguito dell'integrazione completa dei moduli.

Questa verifica rappresenta il livello finale di validazione del corretto funzionamento complessivo della piattaforma.

7. Sospensione e Ripresa

La fase di testing del sistema Future Forge Gear può essere sospesa qualora si verifichino condizioni che impediscano il corretto proseguimento delle attività di verifica.

Le principali cause di sospensione includono:

Rilevazione di bug critici o bloccanti:

malfunzionamenti che compromettono le funzionalità principali del sistema, come errori applicativi gravi o anomalie nella gestione dei dati, tali da impedire l'esecuzione dei test pianificati.

Modifiche al sistema non ancora stabilizzate:

aggiornamenti o cambiamenti al codice che non siano stati ancora adeguatamente integrati o verificati, rendendo non attendibili i risultati dei test.

Ripresa delle attività di test (Recovery)

La ripresa delle attività di testing è subordinata alla risoluzione delle cause che hanno determinato la sospensione. In particolare, prima di procedere con nuove esecuzioni di test, è necessario:

Correggere i bug bloccanti individuati, seguiti da una riesecuzione mirata dei test interessati;

Eseguire test di regressione sui componenti coinvolti, al fine di verificare che le correzioni apportate non abbiano introdotto nuove anomalie.

Progetto: Future Forge Gear	Versione: 1.0
Documento: Test Plan	Data: 22/12/2025

8. Testing Materials

Gli strumenti utilizzati per il testing sono:

- **Testing d'unità e d'integrazione:** JUnit e Mockito, H2
- **Test di Sistema:** SeleniumIDE

Note: H2 in-memory è un **database relazionale leggero scritto in Java** che può funzionare **interamente in memoria RAM** invece che su disco. Quindi permetterà di evitare che il database salvi i dati su file permanenti e viene utilizzato per fare prove senza toccare il database vero.

9. Test Case

I singoli test case relativi a questa sezione sono disponibili nel documento **Test Case Specification (TCS)** – disponibile su GitHub al seguente indirizzo:

https://github.com/francesco-mascolo-19/FutureForgeGear/blob/main/Deliverables/TCS_FutureForgeGear.pdf.

Progetto: Future Forge Gear	Versione: 1.0
Documento: Test Plan	Data: 22/12/2025

10. Testing Schedule

Il programma dei test per il sistema Future Forge Gear è stato definito per garantire un processo di verifica strutturato e coerente con le tempistiche di sviluppo.

Le attività di testing sono organizzate in tre fasi principali, ciascuna dedicata a un diverso livello di verifica del sistema.

Testing Unitario

Durata: 1 settimana

Verifica delle singole classi e dei metodi implementati, eseguita in parallelo allo sviluppo del codice per individuare tempestivamente eventuali anomalie.

Testing di Integrazione

Durata: 1 settimana

Verifica delle interazioni tra i moduli del sistema, con particolare attenzione alle dipendenze e alla corretta cooperazione tra le componenti applicative.

Testing di Sistema

Durata: fase finale del progetto

Validazione del comportamento complessivo della piattaforma Future Forge Gear, attraverso la verifica dei principali flussi funzionali dal punto di vista dell'utente.

Al termine di ciascuna fase è prevista una revisione dei risultati (TER Test Execution Report) , con eventuale correzione delle anomalie (TIR Test Incident Report) prima del passaggio alla fase successiva e una riesecuzione mirata dei test nella fase finale.