

Università degli Studi di Salerno

Corso di Ingegneria del Software

Esbet START Test Plan Document Versione 1.2



Data: 31/01/2025

Progetto: EsbetSTART	Versione: 1.0
Documento: Test Plan Document	Data: 31/01/2025

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Aversana Marco	0512118978
Petrillo Francesco	0512116329
Polise Michele	0512116854
Zaccaro Valeria	0512117733

Scritto da:	
-------------	--

Revision History

Data	Versione	Descrizione	Autore
16/12/2024	1.0	Prima stesura TPD	
18/01/2025	1.1	Separati TPD e TCS	
31/01/2025	1.2	Revisione finale	

1.	Introduzione.....	4
2.	Relationship to other documents	4
3.	Features to be tested/not to be tested	4
4.	Pass/Fail criteria	4
5.	Approach	4
6.	Suspension and resumption	5
7.	Testing materials (hardware/software requirements).....	5

1. Introduzione

L'obiettivo dei test è verificare affidabilità e correttezza delle funzionalità principali del sistema, in questo documento saranno controllate le funzioni di autenticazione e di aggiunta di oggetti al database.

2. Relationship to other documents

Per la creazione del test case, ci si riferisce a:

Requirements Analysis Document (RAD)

System Design Document (SDD)

Object Design Document (ODD)

3. Features to be tested/not to be tested

Verranno testate le funzionalità di:

- Aggiunta di un gioco
- Autenticazione
- Aggiunta di un'offerta

4. Pass/Fail criteria

Le operazioni di test hanno l'obiettivo di individuare eventuali difetti nel sistema, così da poterli correggere. L'esito di un test è stabilito tramite il confronto con un oracolo, che rappresenta il risultato previsto in base ai requisiti.

Un test si considera superato (pass) se l'output effettivo coincide con quello previsto, mentre risulta fallito (fail) se l'output differisce da quello previsto.

Il testing sarà ritenuto adeguato se saranno verificati tutti i requisiti funzionali legati alla funzionalità analizzata.

5. Approach

Per garantire la qualità e l'affidabilità del sistema sviluppato, verrà adottato un approccio strutturato al testing, suddiviso in tre livelli principali: unit testing, integration testing e system testing. Ogni livello utilizzerà strumenti specifici per rispondere alle diverse esigenze di verifica e validazione del sistema.

Unit Testing con JUnit

Il primo livello del testing si concentra sulle unità fondamentali del codice, come metodi o classi isolate. Per questa fase, verrà utilizzato **JUnit**, una libreria robusta e ampiamente utilizzata per la scrittura e l'esecuzione di test. Questo approccio consente di verificare il corretto funzionamento della logica di business implementata, assicurandosi che ogni componente lavori come previsto in isolamento. L'uso di JUnit permette inoltre di integrare facilmente test automatizzati nei flussi di sviluppo, migliorando la rilevazione precoce di errori e semplificando il processo di debugging.

Integration Testing con Mockito

Il secondo livello di testing riguarda la verifica dell'interazione tra più componenti del sistema, come service, repository e controller. Per questa fase, verrà utilizzato **Mockito**, un framework per il mocking delle dipendenze che consente di simulare il comportamento di componenti esterni senza

dover caricare il contesto Spring completo. Questo approccio permette di verificare la logica di interazione tra componenti come i service e i repository in modo rapido ed efficace, riducendo i tempi di esecuzione e isolando eventuali errori.

System Testing con Selenium

Il livello finale del testing, il **system testing**, si concentra sulla validazione del sistema nella sua interezza, simulando scenari d'uso reali. Per questa fase verrà utilizzato **Selenium**, uno strumento potente per l'automazione dei test dell'interfaccia utente. Attraverso Selenium, sarà possibile simulare il comportamento dell'utente su un browser reale, testando flussi complessi che includono la navigazione, l'interazione con form e bottoni, e la verifica di risposte dinamiche dell'applicazione. Questo approccio permette di valutare non solo il funzionamento funzionale, ma anche l'esperienza utente complessiva.

6. Suspension and resumption

Non sono previste sospensioni o ripristini.

7. Testing materials (hardware/software requirements)

Per il testing sarà utilizzato un pc.