

Università degli Studi di Salerno

Corso di Ingegneria del Software

Esbet START Test Plan Document Versione 1.0



Data: 16/12/2024

1.	Introduzione.....	4
2.	Relationship to other documents	4
3.	System overview.....	Errore. Il segnalibro non è definito.
4.	Features to be tested/not to be tested	4
5.	Pass/Fail criteria	4
6.	Approach	4
7.	Suspension and resumption	4
8.	Testing materials (hardware/software requirements).....	5
9.	Test cases	6

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.

8. Test cases

addGame(AddGameRequest r) TC_1

Parametro: Name	
Nome Categoria	Scelte e vincoli
Lunghezza [LN]	1. Lunghezza >30 = false [error] 2. Lunghezza <=30 = true [PROPERTY LN_OK]
Parametro: Rules	
Nome Categoria	Scelte e vincoli
Lunghezze [LR]	1. Lunghezza per regola > 30 = false [error] 2. Lunghezza per regola <= 30 = true [PROPERTY LR_OK]

Test Case ID	Test Frame	Esito
TC_1.1	LN1	Errore: Nome troppo lungo
TC_1.2	LN2, LR1	Errore: Una delle regole supera il limite di caratteri
TC_1.3	LN2, LR2	Corretto

TEST CASE

TC1.1: name: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaa", rules: {"1", "2"}

ORACOLO: Visualizzazione di un messaggio di errore "nome del gioco troppo lungo"

TC1.2: name: "League of Legends", rules: {"1",
"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"}

ORACOLO: Visualizzazione di un messaggio di errore "Una delle regole supera il limite di caratteri"

TC1.3: name: "League of Legends", rules: {"1", "2"}

ORACOLO: Gioco aggiunto con successo

Autenticazione(String email, String password) TC_2

Parametro: Email	
Nome categoria	Scelte e vincoli
Esistenza[UP]	1. Email non presente nel database = false [errore] 2. Email presente nel database = true [PROPERTY_UP_OK]
Parametro: Password	
Nome categoria	Scelte e vincoli
Associazione[PP]	1. Password associata a username nel database = false [errore] 2. Password non associata a username nel database = true [PROPERTY_PP_OK]

Test Case ID	Test Frame	Esito
TC_2.1	UP1	Errore: Email non presente nel database
TC_2.2	UP2, PP1	Errore: Password non associata ad un username
TC_2.3	UP2, PP2	Corretto

TEST CASE

TC2.1: email: “francesco”, password: “imugi” – in DB non presente nessun utente con username “francesco”

ORACOLO: Visualizzazione pagina di autenticazione con messaggio di errore “username o password non corretta”

TC2.2: username: “marco”, password: “artemis” – in DB presente la coppia (username = “marco”, password = “motorino”)

ORACOLO: Visualizzazione pagina di autenticazione con messaggio di errore “username o password non corretta”

TC2.3: username: “francesco”, password: “imugi” – in DB presente la coppia (username = “francesco”, password = “imugi”)

ORACOLO: Utente autenticato e indirizzato alla sua home page

Addoffer(AddOfferRequest r) TC_3

Parametro: Description	
Nome categoria	Scelte e vincoli
Lunghezza [LD]	1. Lunghezza > 50= false [errore] 2. Lunghezza <= 50 = true [PROPERTY_LD_OK]
Parametro: Name	
Nome categoria	Scelte e vincoli
Lunghezza [LN]	1. Lunghezza > 10 OR Lunghezza < 1 = false [errore] 2. Lunghezza <= 10 AND Lunghezza >= 1 = true [PROPERTY_LN_OK]
Parametro: Expiration Date	
Nome categoria	Scelte e vincoli
Correttezza [CD]	1. ExpirationDate <= DataOraCorrente = false [errore] 2. ExpirationDate > DataOraCorrente = true [PROPERTY_CD_OK]
Parametro: Goal	
Nome categoria	Scelte e vincoli
Positività [PG]	1. Goal <= 0 = false [error] 2. Goal > 0 = true [PROPERTY_PG_OK]
Parametro: Type	
Nome categoria	Scelte e vincoli

ORACOLO: Visualizzazione pagina delle offerte con messaggio di errore "Type troppo lungo o troppo corto"

TC3.6: description: "Entra nel vivo dell'azione con la nostra offerta esclusiva dedicata agli appassionati di eSports!" nome:" eSports Winning Bonus" expirationDate"20/10/2030" goal="10", type="ciao"

ORACOLO: L'offerta è stata aggiunta correttamente