# Test planning

# Introduzione

Lo scopo di questo documento è delineare le strategie adottate per testare il sistema GottaBattleEmAll al fine di garantire un sistema privo di errori procederemo all'analisi dettagliata dei vari componenti, indicando le specifiche strategie di testing e i casi di test corrispondenti.

# Relazioni con altri documenti

In questo documento si farà riferimento al **R.A.D, S.D.D, O.D.D**.

# Panoramica del sistema

Come analizzato nel documento SDD, l’architettura che andremo a utilizzare //TODO

L’architettura di microservizi Spring Boot è un modello di architettura che struttura un’applicazione come una raccolta di servizi.

Le macrocategorie di Servizi che sono state individuate sono:

* Gestione Guest
* Gestione Utente
* Gestione Tornei
* Gestione Partite
* Gestione Torneo

# Funzionalità da testare

## a. Gestione Guest

Registrazione giocatore

Registrazione organizzatore

## b. Gestione utente:

Autenticazione

Logout

Modificare profilo

## c. Gestione giocatore:

Sostituire membro Team

## d. Gestione torneo:

Cercare torneo

Crea torneo

Iscrizione Torneo

Seguire un Organizzatore

Iniziare torneo

Terminare torneo

Togliere partecipanti

Visualizzare Profilo Giocatore

Aggiungere Risultato

Partecipa torneo

## e. Gestione moderatore:

Accettare/rifiutare un organizzatore

Bannare/sbannare un giocatore/organizzatore

# Criteri Pass/Failed

Il testing è un approccio intenzionale volto a creare situazioni di fallimento o errori.

Per valutare se un caso di test ha rilevato un difetto, utilizziamo un "oracolo", cioè il risultato atteso dell'esecuzione, confrontandolo con il risultato effettivo del test.

I criteri di success/fail del testing si suddivideranno come segue:

* Success: indica che il test non ha avuto successo nel trovare difetti, il risultato del test corrisponde al suo oracolo.
* Fail: indica che il test ha avuto successo nel trovare un difetto, il risultato del test differisce dal suo oracolo.

# Approccio

L’ approccio che andremo ad utilizzare per il testing si compone in due fasi:

* Testing Unità
* Testing di Sistema

## 6.1 Testing Unita’

L’approccio che utilizzeremo per effettuare le operazioni di testing di unita sarà Il Category partition.

Il Category partition è una tecnica per l’individuazione dei casi di test partendo da un requisito funzionale e dal caso d’uso collegato che è possibile trovare nel R.A.D. .

I test di unità verranno effettuati sui metodi delle classi sviluppate mediante l’utilizzo di JUnit e Mockito offerti da Spring.

Per i report dell’esecuzione del test sara utilizzato Surefire e jacoco

* Surefire è un plugin di Maven utilizzato per eseguire test automatici nel ciclo di vita del build. È compatibile con i framework di test JUnit . Di default, Surefire include automaticamente tutte le classi di test il cui nome inizia con “Test”, o termina con “Test”, “Tests” o “TestCase”.
* JaCoCo è uno strumento di generazione di report di code coverage per progetti Java. Misura quante linee del tuo codice vengono eseguite durante i test automatici. Dopo aver eseguito i test con JUnit, l’agente JaCoCo genera un report di code coverage in formato binario nel directory target, target/jacoco.exec. Puoi utilizzare l’obiettivo jacoco:report per generare report di code coverage leggibili in vari formati, come HTML, CSV e XML.

I Test si troveranno nel package Test separato dall’implementazione.

Mentre i Report saranno salvati nel package target

* I report di Surefire si trovano nella directory target/surefire-reports.
* I report di JaCoCo si trovano nella directory target/site/jacoco.

## 6.2 Testing di sistema e funzionalità

Per il testing di funzionalità e integrazione è stato realizzato tramite il tool Selenium più precisamente utilizzeremo la parte di Web Driver che serve per simulare le interazioni dell’utente col sistema.

# Sospensione e ripresa

## 7.1. Criteri di sospensione

La fase di testing sarà sospesa nel caso verrà rilevato un difetto che può limitare il processo di test per procedere ulteriormente. La sospensione del processo dovrà incidere il meno possibile sulle risorse disponibili. La fase di testing può essere sospesa qualora si raggiungeranno gli obiettivi dichiarati, rispettando i costi ed i tempi fissati.

## 7.2 Criteri di ripresa

La fase di testing riprenderà quando il difetto verrà risolto con successo. I test verranno ripetuti per controllare se le modifiche non hanno generato nuovi errori.

# Materiale per il testing

Gli strumenti necessari per svolgere le attivita di testing sono:

* Maven per lo start della applicazione e spring e i testing
* Un DBMS MySql per il DB
* Selenium IDE per il test di integrazione

# Test Cases

## 9.1 Autenticazione

Parametri: Username,password,ruolo

Oggetti del sistema: DB

|  |  |
| --- | --- |
| Categorie | Scelte |
| Username | US1: username presente nel DB  UF1: username non presente nel sistema[error]  UF2: username bannato[error]  UF3: username non ancora accettato[error] |
| Password | PS1: password corretta  PF2: password non corretta [error] |
| Ruolo | RS: ruolo corretto  RF: ruolo non corretto [error] |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| US1 – PS1 – RS | Autenticazione avvenuta con successo |
| UF1 – PS1 – RS | Credenziali sbagliate |
| US1 – PF1 – RS | Credenziali sbagliate |
| US1 – PS1 – RF | Credenziali sbagliate |
| UF2 – PS1 – RS | Account Bannato |
| UF3 – PS1 – RS | Account non ancora accettato |

9. 2. Registrazione Giocatore

Parametri: username, nome, cognome, e-mail, password, conferma, ruolo

Oggetti del sistema: DB

|  |  |
| --- | --- |
| Categorie | Scelte |
| username | US1: username disponibile  UF1: username non disponibile[error] |
| email | ES1: corretta  EF1 : email non corretta [error] |
| password/conferma | PS: corrisponde  PF: non corrisponde [error] |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| US1 – ES1 – PS | Registrazione avvenuta |
| UF1 – ES1 – PS | Username non disponibile |
| US1 – ES2 – PS | Email non valida |
| US1 – ES1 – PF | Password non corrisponde |

9. 3. Registrazione organizzatore

Parametri: username, nome, cognome, e-mail, password, conferma, ruolo

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |
| username | US1: username disponibile  UF1: username non disponibile[error] |
| email | ES1: corretta  EF1 : email non corretta [error] |
| password/conferma | PS: corrisponde  PF: non corrisponde [error] |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| US1 – ES1 – PS | Richiesta inviata |
| UF1 – ES1 – PS | Username non disponibile |
| US1 – ES2 – PS | Email non valida |
| US1 – ES1 – PF | Password non corrisponde |

9. 4. Accettare un organizzatore

Parametri: Organizzatore

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |
| Organizzatore | OS: organizzatore selezionato |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| OS | Organizzatore accettato |

9. 5. Rifiutare un organizzatore

Parametri: Organizzatore

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |
| Organizzatore | OS: organizzatore selezionato |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| OS | Organizzatore rifiutato |

9.6. Cercare Torneo

Parametri: torneo

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |
| torneo | TS |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| **TS** | Richiesta andata a buon fine |

9.7. Logout

Parametri: utente

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |
| utente | US:Utente presente |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| **US** | sloggato |

9. 8. Bannare Utente

Parametri: utente

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |
| utente | US: utente bannabile |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| **US** | Ban eseguito |

9. 9. sBannare Utente

Parametri: utente

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |
| utente | US: utente bannabile |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| **US** | sban eseguito |

9. 10. Crea Torneo

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |

9.

Parametri:

Oggetti del sistema DB

|  |  |
| --- | --- |
| Categorie | Scelte |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |