

# Test Plan

*FormulaOnline*

Versione	1.0
Presentato da	ALFONSO ANZELMO, PAOLO D'ANTUONO

## Storico versioni

Data	Versione	Descrizione	Autori
27/06/2024	0.1	Prima stesura	A. Anzelmo, P. D'Antuono
28/06/2024	0.2	Aggiunta dei test case	A. Anzelmo, P. D'Antuono
09/07/2024	0.3	Aggiunta test case 2	P. D'Antuono
18/07/2024	1.0	Revisione finale ed aggiunta links	A. Anzelmo, P. D'Antuono

# Sommario

<b>1. Introduzione</b>	<b>4</b>
<b>2. Relazione con gli altri documenti</b>	<b>4</b>
2.1 Altri riferimenti	4
<b>3. Panoramica del sistema</b>	<b>4</b>
<b>4. Funzionalità da testare</b>	<b>5</b>
<b>5. Pass/Fail criteria</b>	<b>6</b>
<b>6. Approccio</b>	<b>6</b>
6.1 Testing di unità	6
6.2 Testing Funzionale	6
<b>7. Strumenti per il testing (hardware/software)</b>	<b>6</b>
<b>8. Test cases</b>	<b>6</b>
<b>9. Specifica dei test cases</b>	<b>7</b>
9.1 Gestione Discussione	7
9.2 Registrazione	9
<b>10. Pianificazione del testing</b>	<b>12</b>

# 1. Introduzione

Lo scopo di FormulaOnline è quello di condividere le proprie opinioni sul mondo della Formula 1. Lo scopo di questo documento è di definire le strategie di testing e di garantire la corretta esecuzione del sistema.

## 2. Relazione con gli altri documenti

**Relazione con [RAD](#):** il testing si effettua al livello di requisiti funzionali e non funzionali.

**Relazione con [SDD](#):** il testing si effettua al livello dei sottosistemi individuati nella fase alta di progettazione.

**Relazione con [ODD](#):** il testing si effettua al livello di divisione in package.

### 2.1 Altri riferimenti

- [Statement Of Work](#)
- [Requirements Analysis Document](#)
- [System Design Document](#)
- [Object Design Document](#)
- [Test Case Specification](#)
- [Test Incident Report](#)
- [Test Summary Report](#)
- [Matrice di tracciabilità](#)
- [Manuale di installazione](#)
- [Manuale utente](#)

## 3. Panoramica del sistema

Il sistema segue lo stile architetturale Three Tier il quale divide in livelli i sottosistemi:

livello di persistenza in cui si gestiscono i dati

livello application in cui è presente la logica del sistema

livello presentation in cui è presente l'interfaccia con l'utente

Per l'implementazione della logica di business avverrà attraverso Servlet Java, per la persistenza dei dati sarà utilizzato MySql invece la parte front-end sarà implementata attraverso HTML5, JSP, CSS3 e Bootstrap.

## 4. Funzionalità da testare

Elenchiamo le funzionalità da testare divise per sottosistema:

**Gestione discussione:**

Creazione di una discussione

**Registrazione:**

Registrazione lettore

## 5. Pass/Fail criteria

Un test prevede l'immissione di una serie di input al sistema ed una serie di comportamenti attesi per ogni input, questi ultimi sono detti "oracoli". I risultati possibili di un test sono:

- **Test Pass:** se dato un input al sistema l'output ricevuto è diverso da quello atteso, vuol dire che è stato individuato un Fault ed il test ha avuto successo
- **Test Fail:** se dato un input al sistema l'output ricevuto è uguale a quello atteso

## 6. Approccio

### 6.1 Testing di unità

Si focalizza sugli oggetti e i sottosistemi. Le unità candidate sono prese dal modello a oggetti e dalla scomposizione in sottosistemi.

### 6.2 Testing Funzionale

Si pone come obiettivo quello di testare le funzionalità del sistema al fine di validare i requisiti funzionali presenti nel RAD. Evidenzia le differenze tra i requisiti funzionali e il sistema realizzato.

## 7. Strumenti per il testing (hardware/software)

Per il testing è necessario un semplice PC. Le componenti software utilizzate sono MySQL per la gestione del database relazionale, Apache Tomcat 10.0+ per l'esecuzione del server, JDK 17+ ed almeno un browser web. JaCoCo invece per ottenere la percentuale di coverage del codice testato.

## 8. Test cases

Per sviluppare i test cases sarà utilizzato il metodo del Category Partition. Questo metodo consiste nell'identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore.

I test cases verranno definiti nel documento di Test Cases Specification (TCS).

## 9. Specifica dei test cases

### 9.1 Gestione Discussione

TC\_1 Crea Discussione

Parametro: Commento	
Categorie	Scelte
lunghezza [LC]	1: lunghezza == 0 [errore] 2: lunghezza >=1 && lunghezza <=500 [property <b>LC_OK</b> ] 3: lunghezza >500 [errore]

Parametro: Titolo	
Categorie	Scelte
lunghezza [LT]	1: lunghezza == 0 [errore] 2: lunghezza >=1 && lunghezza <=50 [property <b>LT_OK</b> ] 3: lunghezza >50 [errore]

Parametro: Categoria	
Categorie	Scelte
<b>Exists categoria</b> [EX_CT]	<ol style="list-style-type: none"> <li>la categoria esiste [EX_CT_OK]</li> <li>la categoria non esiste [errore]</li> </ol>

Test case ID	Test Frame	Esito
TC_1_01	LC1	errore
TC_1_02	LC3	errore
TC_1_03	LC2.LT1	errore
TC_1_04	LC2.LT3	errore
TC_1_05	LC2.LT2.EX_CT2	errore
TC_1_06	LC2.LT2.EX_CT1	creazione discussione



## 9.2 Registrazione

### TC\_2 Registrazione lettore

Parametro: email	
Categorie	Scelte
<b>Email Utilizzata</b> <b>[EL]</b>	<b>1:</b> l'email inserita è già stata utilizzata per un altro account [errore] <b>2:</b> l'email non è stata già utilizzata <b>[EL_OK]</b>
<b>Lunghezza Email</b> <b>[LE]</b>	<b>1:</b> lunghezza < 5 [errore] <b>2:</b> 5 <= lunghezza <=50 <b>[LE_OK]</b> <b>3:</b> lunghezza > 50 [errore]
<b>Formato Email</b> <b>[FE]</b>	<b>1:</b> l'email è in un formato non valido [errore] <b>2:</b> l'email rispetta il formato stringa@dominio <b>[FE_OK]</b>

Parametro: password	
Categorie	Scelte
<b>Lunghezza Password</b> <b>[LP]</b>	<b>1:</b> lunghezza < 8 [errore] <b>2:</b> 8 <= lunghezza <= 32 [ <b>LP_OK</b> ] <b>3:</b> lunghezza > 32 [errore]

Parametro: nickname	
Categorie	Scelte
<b>Lunghezza Nickname</b> <b>[LN]</b>	<b>1:</b> lunghezza < 5 [errore] <b>2:</b> 5 <= lunghezza <= 30 [ <b>LN_OK</b> ] <b>3:</b> lunghezza > 30 [errore]

Parametro: scuderiaPreferita	
Categorie	Scelte
<b>Lunghezza Scuderia</b>	<b>1:</b> 0 <= lunghezza <=50 [ <b>LS_OK</b> ]

<b>[LS]</b>	<b>2: lunghezza &gt; 50 [errore]</b>
-------------	--------------------------------------

Test case ID	Test Frame	Esito
TC_2_01	EL1	errore
TC_2_02	EL2.LE1	errore
TC_2_03	EL2.LE3	errore
TC_2_04	EL2.LE2.FE1	errore
TC_2_05	EL2.LE2.FE2.LP1	errore
TC_2_06	EL2.LE2.FE2.LP3	errore
TC_2_07	EL2.LE2.FE2.LP2.LN1	errore
TC_2_08	EL2.LE2.FE2.LP2.LN3	errore
TC_2_09	EL2.LE2.FE2.LP2.LN2.LS2	errore
TC_2_10	EL2.LE2.FE2.LP2.LN2.LS1	registrazione

## 10. Pianificazione del testing

La scrittura e l'esecuzione dei casi di test avverrà durante l'implementazione del codice. I test verranno rieseguiti dopo la fase di implementazione.