

# Test Plan

## 1. Introduzione

Il Test Plan Document di TomMASO include le informazioni necessarie per definire l'approccio che deve essere usato nella fase di testing del progetto. Lo scopo di questo documento è quello di gestire lo sviluppo e le attività di test di TomMASO per trovare eventuali errori all'interno del codice realizzato. I risultati dei test saranno poi utilizzati per trovare correzioni da apportare al codice, al fine di risolvere i problemi per migliorare il sistema.

## 2. Riferimenti

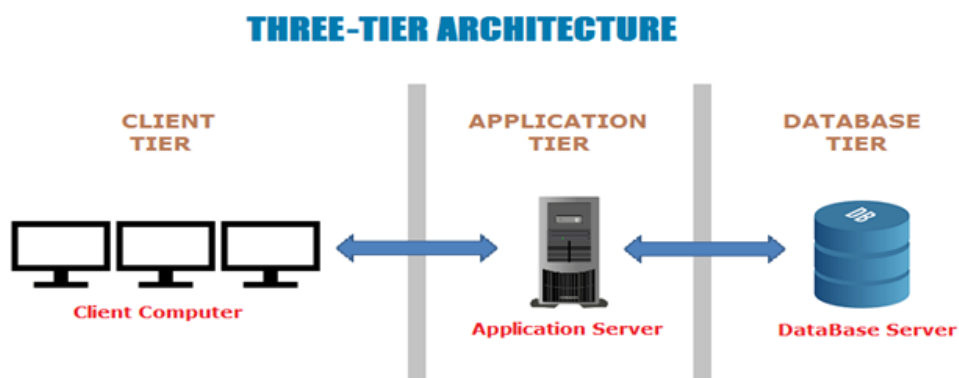
**RAD:** dove verranno testati i casi d'uso.

**SDD:** comprendente la definizione del sistema in sottosistemi e il controllo degli accessi.

**ODD:** verranno testate le classi implementate.

## 3. Panoramica del sistema

Il nostro sistema è basato su architettura a tre livelli.



## 4. Funzionalità da testare

Di seguito verranno riportate le funzionalità del sistema che saranno sottoposte a test:

- UC1. UserSignUp
- UC2. Login
- UC6. UploadFile
- UC7. EditMarkdownFile
- UC11. SendMessage
- UC12. SendComment
- UC15. DeleteBlog
- UC21. Search
- UC23. CreateBlog

## 5. Criteri di successo/insuccesso

La fase di testing verrà ritenuta “di successo” nel caso in cui tutti i test previsti abbiano un risultato positivo (il risultato del caso di test coincide con l’oracolo)

## 6. Approccio

Il testing che svolgeremo su questo sistema si compone di tre fasi:

- Testing di unità, con lo scopo di testare le componenti del sistema singolarmente
- Testing di integrazione, che servirà a testare le funzionalità dei vari componenti del sistema
- Testing di sistema, che mira a verificare che l’intero sottosistema soddisfi le richieste del cliente.

### 6.1. Test di unità

Per realizzare il testing di ogni singola componente verrà utilizzata la tecnica **Black-Box**.

Per il partizionamento degli input verranno usate **classi di equivalenza** rendendo i test più efficienti. I risultati del testing verranno analizzati e usati successivamente per correggere eventuali errori del sistema.

## 6.2. Test di integrazione

Il test di integrazione serve a testare l'intero sistema e si sviluppa dopo il test di unità, quando ogni parte testata viene integrata con il resto del sistema.

Verrà usata la strategia **bottom-up**: strategia di testing che permette di testare prima i componenti del livello inferiore singolarmente, poi di integrarli mano a mano con i componenti di livello superiore.

## 6.3. Test di sistema

Il test di sistema ha lo scopo di testare tutte le funzionalità più importanti e usate maggiormente. Trattandosi di un sistema web-based verrà utilizzato **Selenium**.

## 7. Sospensione e ripresa

La fase di testing del sistema verrà sospesa in caso siano rilevate failure. Dopo la loro correzione e convalida attraverso test di regressione si potrà riprendere il testing.

## 8. Materiali di prova (requisiti hardware/software)

Gli strumenti utilizzati nella fase di testing saranno:

- **MariaDB** per la gestione del database.
- **Apache Tomcat**, come web server
- **JUnit, DBUnit, Mockito** per il test di unità e di integrazione.
- **Selenium** per il test di sistema.

## 9. Specifiche casi di test

### Specifica dei casi di test per il caso d'uso UC1. UserSignUp

<b>Parametri</b>	username
	password1
	password2
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Categoria 1:</b> formato username
	<b>Categoria 2:</b> username già presente nel database
	<b>Categoria 3:</b> formato password
	<b>Categoria 4:</b> password1 uguale a password2
<b>Scelte</b>	<b>Scelte per Categoria 1:</b> FU1: formato username non corretto FU2: formato username corretto
	<b>Scelte per Categoria 2:</b> UP1: username già presente UP2: username non presente
	<b>Scelte per Categoria 3:</b> FP1: formato password non corretto FP2: formato password corretto
	<b>Scelte per Categoria 4:</b> PU1: password1 uguale a password2

	PU2: password1 diversa da password2
<b>Vincoli</b>	<b>FU1:</b> formato username errato
	<b>FU2:</b> formato username corretto [property formato_username_corretto]
	<b>UP1:</b> username già presente [if formato_username_corretto]
	UP2: username non presente [property username_non_presente]
	<b>FP1:</b> formato password errato [if formato_username_corretto and username_non_presente]
	<b>FP2:</b> formato password corretto [property formato_password_corretto]
	<b>PU1:</b> password1 diversa da password2 [if formato_username_corretto and username_non_presente and formato_password_corretto]
	<b>PU2:</b> password1 uguale a password2
<b>Test Frame</b>	<b>TC1:</b> FU2, UP2, FP2, PU2
	<b>TC2:</b> FU1, UP2, FP2, PU2
	<b>TC3:</b> FU2, UP1, FP2, PU2
	<b>TC4:</b> FU2, UP2, FP1, PU2
	<b>TC5:</b> FU2, UP2, FP2, PU1

<b>Test Case</b>	<b>TC1:</b> username="AAAAAAAA", password1="BBBBBBBB", password2="BBBBBBBB" - ORACOLO: Utente correttamente autenticato
	<b>TC2:</b> username="AAAAAAA!", password1="BBBBBBBB", password2="BBBBBBBB" - ORACOLO: Utente reindirizzato alla registrazione con il messaggio "Username non valido"
	<b>TC3:</b> username="XXXXXXXX", password1="BBBBBBBB", password2="BBBBBBBB" - ORACOLO: Utente reindirizzato alla registrazione con il messaggio "Impossibile registrare l'account"
	<b>TC4:</b> username="AAAAAAAA", password1="BBBBBB", password2="BBBBBB" - ORACOLO: Utente reindirizzato alla registrazione con il messaggio "La password deve contenere almeno 8 caratteri"
	<b>TC5:</b> username="AAAAAAAA", password1="BBBBBBBB", password2="CCCCCCCC" - ORACOLO: Utente reindirizzato alla registrazione con il messaggio "Le password non corrispondono"

## Specifica dei casi di test per il caso d'uso UC2. Login

<b>Parametri</b>	username
	password
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Categoria 1:</b> username presente nel database
	<b>Categoria 2:</b> password associata a username nel database
	<b>Categoria 3:</b> permesso login
<b>Scelte</b>	<b>Scelte per Categoria 1:</b> UP1: username presente nel database UP2: username NON presente nel database
	<b>Scelte per Categoria 2:</b> PA1: password associata a username PA2: password NON associata a username
	<b>Scelte per Categoria 3:</b> PL1: l'utente ha il permesso di autenticarsi PL2: l'utente NON ha il permesso di autenticarsi
<b>Vincoli</b>	<b>UP1:</b> username presente nel database [property username_presente]

	<b>UP2:</b> username NON presente nel database [property username_non_presente]
	<b>PA1:</b> password associata a username [if username_presente] [property utente_esistente]
	UP2: username non presente [property username_non_presente]
	<b>PA2:</b> password NON associata a username
	<b>PL1:</b> l'utente ha il permesso di autenticarsi [if utente_esistente]
	<b>PL2:</b> l'utente NON ha il permesso di autenticarsi
<b>Test Frame</b>	<b>TC1:</b> UP1, PA1, PL1
	<b>TC2:</b> UP1, PA2
	<b>TC3:</b> UP2, PA2
	<b>TC4:</b> UP1, PA1, PL1
<b>Test Case</b>	<b>TC1:</b> username: "a.allocca", password: "Allocca123" – in DB presente la coppia (username = "a.allocca", password = "Allocca123" permessi="8") – ORACOLO: Utente autenticato e indirizzato alla sua home page
	<b>TC2:</b> username: "a.allocca", password: "DiCostanzo321" – in DB presente la coppia (username = "a.allocca", password = "Allocca123") – ORACOLO: Visualizzazione pagina di autenticazione con messaggio di errore "Credenziali errate"



	<p><b>TC3:</b> username: "a.delucia", password: "AAAAAAA" – in DB non presente nessun utente con username "a.delucia" – ORACOLO: Visualizzazione pagina di autenticazione con messaggio di errore "Credenziali errate"</p>
	<p><b>TC4:</b> username: "utenteBannato", password: "bannato12" – in DB presente la coppia (username = "a.allocca", password = "Allocca123" permessi="0") – ORACOLO: Utente reindirizzato alla pagina di errore perché non dispone dei permessi necessari all'autenticazione</p>

### Specifica dei casi di test per il caso d'uso UC6. UploadFile

<b>Parametri</b>	filename
<b>Oggetti dell'ambiente</b>	Filesystem
<b>Categorie</b>	<b>Categoria 1:</b> filename
<b>Scelte</b>	<p><b>Scelte per Categoria 1:</b></p> <p><b>NF1:</b> Nome file invalido</p> <p><b>NF2:</b> Nome file valido</p>
<b>Test Frame</b>	<b>TC1:</b> NF1
	<b>TC2:</b> NF2
<b>Test Case</b>	<b>TC1:</b> filename = "../AAA" - ORACOLO: Il file non viene caricato sulla piattaforma
	<b>TC2:</b> filename = "AAABBB" - ORACOLO: Il file viene caricato sulla piattaforma

## Specifica dei casi di test per il caso d'uso UC7. EditMarkdownFile

<b>Parametri</b>	filename
	user
<b>Oggetti dell'ambiente</b>	Filesystem
	Sessione
<b>Categorie</b>	<b>Categoria 1:</b> file esistente
	<b>Categoria 2:</b> utente proprietario
<b>Scelte</b>	<b>Scelte per Categoria 1:</b> FE1: il file esiste FE2: il file NON esiste
	<b>Scelte per Categoria 2:</b> UP1: utente proprietario del file UP2: utente NON proprietario del file
<b>Vincoli</b>	<b>FE1:</b> il file esiste [property file_esistente]
	<b>FE2:</b> il file NON esiste
	<b>UP1:</b> utente proprietario del file [if file_esistente]
	<b>UP2:</b> utente NON proprietario del file [if file_esistente]
<b>Test Frame</b>	<b>TC1:</b> FE1, UP1
	<b>TC2:</b> FE1, UP2
	<b>TC3:</b> FE2
<b>Test Case</b>	<b>TC1:</b> filename = "LaMiaPagina", username dell'utente in sessione = "p.dicostanzo", nel file system è presente tale file – ORACOLO: Le modifiche effettuate al file vengono correttamente salvate e l'utente viene reindirizzato alla pagina visualizzata correttamente.
	<b>TC2:</b> filename = "LaMiaPagina", username dell'utente in sessione = "v.esposito", nel file system è presente tale file – ORACOLO: Le modifiche effettuate al file non vengono salvate perché l'utente non è il proprietario del file e viene reindirizzato alla pagina di errore

	<b>TC3:</b> filename = "FileNonEsistente", nel file system non è presente tale file – ORACOLO: Viene creato per l'utente user un file vuoto.
--	--

## Specifica dei casi di test per il caso d'uso UC11. Send Message

<b>Parametri</b>	message
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Categoria 1:</b> formato messaggio
<b>Scelte</b>	<b>Scelte per Categoria 1:</b> <b>FM1:</b> formato messaggio corretto <b>FM2:</b> formato messaggio NON corretto
<b>Vincoli</b>	<b>FM1:</b> formato messaggio corretto [property messaggio_corretto]
	<b>FM2:</b> formato messaggio NON corretto
<b>Test Frame</b>	<b>TC1:</b> FM1
	<b>TC2:</b> FM2
<b>Test Case</b>	<b>TC1:</b> messaggio="AAA" - ORACOLO: il messaggio viene inviato
	<b>TC2:</b> messaggio=" " – ORACOLO: il messaggio non viene inviato

Specifica dei casi di test per il caso d'uso UC12. SendComment	
<b>Parametri</b>	commento
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Categoria 1:</b> Lunghezza commento (per specifica: il commento deve avere almeno 3 caratteri)
<b>Scelte</b>	<b>Scelte per categoria 1:</b> <b>CLI1:</b> lunghezza commento corretta ( $\geq 3$ ) <b>CLI2:</b> lunghezza commento NON corretta
<b>Vincoli</b>	<b>CL1:</b> lunghezza commento corretta [property commento_corretto]
	<b>CL2:</b> lunghezza commento NON corretta [property commento_non_corretto]
<b>Test Frame</b>	<b>TF1:</b> CL1
	<b>TF2:</b> CL2
<b>Test Case</b>	<b>TC1</b> - TF1 commento="testingiswonderful" - ORACOLO: Commento inviato e reindirizzazione alla pagina
	<b>TC2</b> - TF2 commento="ab" - ORACOLO: Commento non inviato e reindirizzazione alla pagina e visualizzazione messaggio di errore

**Specifica dei casi di test per il caso d'uso UC15. DeleteBlog**

<b>Parametri</b>	blogname
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Categoria 1:</b> blog esistente
<b>Scelte</b>	<b>Scelte per la Categoria 1:</b>  <b>BE1:</b> il blog esiste  <b>BE2:</b> il blog NON esiste
<b>Test Frame</b>	<b>TC1:</b> BE1
	<b>TC2:</b> BE2
<b>Test Case</b>	<b>TC1:</b> blogname = "BlondeSalad" - ORACOLO: Blog eliminato
	<b>TC2:</b> blogname = "BlogInesistente" - ORACOLO: Blog non cancellato

## Specifica dei casi di test per il caso d'uso UC21. Search

<b>Parametri</b>	tipo
	testo
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Categoria1:</b> blog esistente
	<b>Categoria2:</b> utente esistente
<b>Scelte</b>	<b>Scelte per tipo:</b> <p>T1: utente</p> <p>T2: blog</p>
	<b>Scelte per Categoria1:</b> <p>BE1: il blog non esiste</p> <p>BE2: il blog esiste</p>
	<b>Scelte per Categoria2:</b> <p>UE1: l'utente non esiste</p> <p>UE2: l'utente esiste</p>
<b>Vincoli</b>	<b>T1:</b> utente [property type_user]
	<b>T2:</b> blog [property type_blog]
	<b>BE1:</b> il blog non esiste [if type_blog]
	<b>BE2:</b> il blog esiste [if type_blog]
	<b>UE1:</b> l'utente non esiste [if type_user]
	<b>UE2:</b> l'utente esiste [if type_user]
<b>Test Frame</b>	<b>TC1:</b> T1, UE1
	<b>TC2:</b> T1, UE2

<b>Test Case</b>	<b>TC3:</b> T2, BE1
	<b>TC4:</b> T2, BE2
	<b>TC1:</b> tipo="utente", testo="CCCCCCCC" - ORACOLO: L'utente viene reindirizzato alla pagina di errore "Username non trovato"
	<b>TC2:</b> tipo="utente", testo="DDDDDDDD" - ORACOLO: L'utente viene reindirizzato al profilo dell'utente
	<b>TC3:</b> tipo="blog", testo="AAAAAAAA" - ORACOLO: L'utente viene reindirizzato alla pagina di errore "Blog non trovato"
	<b>TC4:</b> tipo="blog", testo="BBBBBBBB" - ORACOLO: L'utente viene reindirizzato al blog



## Specifica dei casi di test per il caso d'uso UC23. CreateBlog

<b>Parametri</b>	blogname
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Categoria 1:</b> formato blogname
	<b>Categoria 2:</b> blogname presente nel database
<b>Scelte</b>	<b>Scelte per Categoria 1:</b> <b>FB1:</b> formato blogname corretto <b>FB2:</b> formato blogname NON corretto
	<b>Scelte per Categoria 2:</b> <b>BD1:</b> blogname presente nel database <b>BD2:</b> blogName NON presente nel database
<b>Vincoli</b>	<b>FB1:</b> formato blogname corretto [property blogname_corretto]
	<b>FB2:</b> formato blogname NON corretto
	<b>BD1:</b> blogname presente nel database [if blogname_corretto]
	<b>BD2:</b> blogname NON presente nel database
<b>Test Frame</b>	<b>TC1:</b> FB1, BD1
	<b>TC2:</b> FB1, BD2
<b>Test Case</b>	<b>TC1:</b> blogname="IlMioBlog" - in DB non presente alcun blog con

	blogname="IlMioBlog" - ORACOLO: il blog viene creato
	<b>TC2:</b> blogname="IlMioBlog" - in DB già presente un blog con blogname="IlMioBlog" - ORACOLO: viene segnalato all'utente la presenza di un blog con quel nome e viene reindirizzato alla pagina di creazione