

Test Plan

Shareboard

Riferimento	TP_G07
Versione	1.6
Data	10/12/2021
Destinatario	Studenti di Ingegneria del Software 2021/22
Presentato da	Antonio Romano (AR), Alessandro Saverio De Maio (AM), Carmine Leo (CL)

Revision History

Data	Versione	Descrizione	Autori
04/12/2021	0.1	Introduzione Documenti correlati Panoramica del sistema	Carmine Leo
05/12/2021	0.2	Funzionalità da testare/da non testare	AR, AM, CL
06/12/2021	0.3	Pass/Fail Criteria	Carmine Leo
07/12/2021	1.0	Test Cases: <ul style="list-style-type: none">• Login• Registrazione	Carmine Leo
09/12/2021	1.3	Test Case: <ul style="list-style-type: none">• Ban• Sottometti post• Commenta post	Antonio Romano
10/12/2021	1.6	Test Case: <ul style="list-style-type: none">• Rispondi• Aggiungi Sezione• Ricerca post	Alessandro Saverio De Maio

Contenuti

Revision History.....	2
Introduzione	4
Documenti correlati.....	4
• Requirement Analysis Document (RAD).....	4
• System Design Document (SDD).....	4
• Object Design Document (ODD).....	4
Panoramica del sistema.....	4
Funzionalità da testare/da non testare.....	4
Pass/Fail criteria.....	5
Approccio.....	5
Testing di unità.....	5
Testing di integrazione.....	6
Testing di Sistema.....	6
Strumenti per il testing (hardware/software).....	6
Test cases.....	6
Login	6
Registrazione.....	7
Ban utente	8
Sottometti Post.....	9
Sottometti Commento	9
Sottometti Risposta	9
Aggiungi Sezione.....	9
Ricerca Post	11
Testing schedule.....	11

Introduzione

Lo scopo del Test Plan è analizzare il sistema in oggetto, pianificando una strategia che permetterà di stimare le risorse necessarie all'attività di testing.


Analizzeremo il sistema progettato nelle fasi precedenti per convalidarne la qualità, saranno descritte le funzionalità da testare e gli strumenti per la rilevazione di eventuali errori all'interno del codice.

Documenti correlati

Il Test Plan analizza il sistema descritto nelle documentazioni prodotte in precedenza, in modo da verificarne la correttezza, i documenti correlati sono i seguenti:

- Requirement Analysis Document (RAD)

La relazione con il suddetto documento riguarda in particolare i requisiti funzionali e non funzionali, poiché il testing sulle funzionalità si baserà sulle specifiche espresse in tale documento.

 RAD_Shareboard

- System Design Document (SDD)

Nel documento SDD il sistema è stato scomposto in sottosistemi utilizzando il pattern architetturale Model View Controller (MVC), a tal proposito il test dei vari componenti dovrà tener conto di tale suddivisione.

 SDD_Shareboard

- Object Design Document (ODD)

Il test d'integrazione farà riferimento alle interfacce delle classi e ai package definiti in suddetto documento.

 ODD_Shareboard

Panoramica del sistema

Come definito nel System Design Document (SDD) la struttura del sistema segue uno stile architetturale three tier.

Saranno utilizzati HTML5, CSS3, JavaScript e JQuery per la parte front-end, mentre per il back-end saranno utilizzate le Java Servlet.

Per la gestione del database sarà utilizzato

- JPA per il collegamento al database.
- SQL come linguaggio per definire le query.

Funzionalità da testare/da non testare

Di seguito le feature di cui si effettuerà il test suddivise per categoria:

- **Navigazione**
 - Ricerca post
- **Interazione**
 - Sottometti post

- Commenta post
- Rispondi
- **Gestione utente**
 - Login
 - Registrazione
- **Moderazione**
 - Ban utente
- **Amministrazione**
 - Aggiungi sezione

Le funzionalità di cui si effettuerà il test si riferiscono ai requisiti funzionali con priorità elevata che prevedono un input da parte dell'utente.

Pass/Fail criteria

L'attività di testing mira ad identificare la presenza di errori all'interno del sistema per poter effettuare successivamente degli interventi correttivi.

La valutazione dell'esito di un test case è basata su di un oracolo, dove troviamo il risultato atteso della sua esecuzione basato sui requisiti.

Pass: il test si considera eseguito con successo se l'output ottenuto differisce dall'output previsto dall'oracolo.

Fail: il test si considera fallito se l'output corrisponde alla previsione contenuta nell'oracolo.

L'attività di testing sarà considerata valida se saranno rispettati i seguenti vincoli:

- Testare tutte le funzionalità con priorità elevata descritte al punto precedente;
- Effettuare test di regressione per ogni nuova funzionalità o ad ogni funzionalità a cui vengono apportate delle modifiche;
- Branch coverage non inferiore al 75%.

Approccio

Il testing dell'intero sistema si compone di tre fasi: testing di unità, testing di integrazione e testing di sistema. Verranno eseguiti e progettati nell'ordine definito.

Testing di unità

Per il testing di unità si testeranno tutti i metodi delle classi Service e i metodi delle classi Servlet, escludendo le classi entity che presentano soltanto i metodi getters e setters. I casi di test adottano un approccio di tipo black box. Per ogni classe appartenente al testing di unità si rispetterà il seguente formato `ClassNameTest`.

Le tecnologie utilizzate in tale fase sono:

- Mockito: per la costruzione degli stub e l'isolamento della componente testata.
- JaCoCo: per il calcolo di metriche tra le quali la Branch Coverage.
- Maven: per la build e l'esecuzione automatica dei tests.
- JUnit: framework di testing.
- OpenEJB Application Composer: per l'esecuzione di un container CDI leggero fatto appositamente per esecuzioni di test di unità. Garantisce tempi di avvio bassi.
- CdiMock: libreria per l'iniezione di mock nei CDI bean.

Testing di integrazione

Per tale fase di testing verrà utilizzato un approccio bottom-up, adatto per software basato sul paradigma Object Oriented. Come mostrato nel diagramma seguente, si è proceduto dal layer di persistenza verso il layer superiore relativo alla logica dell'applicazione. I test case saranno implementati tramite il framework JUnit, mentre verrà utilizzato Mockito per il mocking. Infine, come tool di misurazione e report coverage sarà utilizzato JaCoCo. Il testing di integrazione sarà eseguito sulle classi repository e sui servizi, per la creazione delle classi di test si rispetterà il formato ClassNameIT.

Testing di Sistema

Il testing di sistema sarà eseguito tramite il tool Selenium IDE, un ambiente di sviluppo integrato completo per i test che consente di creare, registrare e fare il debug dei test case di sistema con l'obiettivo di validare i requisiti funzionali, individuando i possibili faults generati dagli input degli utenti. Il server per la fase di testing verrà avviato in localhost.

Strumenti per il testing (hardware/software)

Gli strumenti per le fasi di testing, per quanto riguarda la parte hardware, saranno un semplice computer non necessariamente connesso ad internet, in quanto il deploy del sistema avviene localmente.

Per la parte software, di seguito si riepilogano le tecnologie utilizzate:

- JUnit: framework di testing.
- Selenium IDE: per l'esecuzione dei test case di sistema.
- Mockito: per la costruzione degli stub e l'isolamento della componente testata.
- JaCoCo: per il calcolo di metriche tra le quali la Branch Coverage.
- Maven: per la build e l'esecuzione automatica dei tests.
- OpenEJB Application Composer: per l'esecuzione di un container CDI leggero fatto appositamente per esecuzioni di test di unità. Garantisce tempi di avvio bassi.
- CdiMock: libreria per l'iniezione di mock nei CDI bean.

Test cases

Per sviluppare i test frame 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.

Login

Parametro: Username Formato: ^[\w\-\-]+\$	
Categorie	Scelte
Lunghezza[LU]	1. LU < 3 OR LU > 30 = false [error] 2. LU >= 3 AND LU <= 30 = true [PROPERTY LU_OK]
Formato[FU]	1. Rispetta il formato == false [error] 2. Rispetta il formato == true [PROPERTY FU_OK]
Match[MU]	1. Match nel Database == false [error] 2. Match nel Database == true [PROPERTY MU_OK]

Parametro: Password	
Categorie	Scelte

Codice	Combinazione	Esito
TC_1_1	LU1	Errore: Lunghezza username > 30
TC_1_2	LU2	Errore: Lunghezza username < 3
TC_1_3	LU3, FU1	Errore: formato non rispettato
TC_1_4	LU4, FU2, MU1	Errore: username inesistente
TC_1_5	LU5, FU3, MU2, LP1	Errore: password > 255 caratteri
TC_1_6	LU5, FU3, MU2, LP2	Errore: password < 3 caratteri
TC_1_7	LU5, FU3, MU2, LP3	Corretto

Parametro: Email	
Formato RFC 5322: ^([?:[a-z0-9!#\$%&'*/+=?^_`{ }~-]+)* \\"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x23-\x5b\x5d-\x7f] \\"\\[\x01-\x09\x0b\x0c\x0e-\x7f])*")@([?:([a-z0-9]([a-z0-9]*[a-z0-9])?\\".)+[a-z0-9]([a-z0-9]*[a-z0-9])?) \\[(?:([2(5[0-5] [0-4][0-9]) 1[0-9][0-9] [1-9]?[0-9]))\\".){3}([2(5[0-5] [0-4][0-9]) 1[0-9][0-9] [1-9]?[0-9]) [a-z0-9]*[a-z0-9]:([?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x23-\x5b\x5d-\x7f] \\"\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\\)]\$)	
Categorie	Scelte
Lunghezza[LE]	1. LE > 255 = false[error] 2. LE <= 255 = true[PROPERTY LE_OK]
Formato[FE]	1. Rispetta il formato == false[error] 2. Rispetta il formato == true[PROPERTY FE_OK]
Match[ME]	1. Match con email esistente nel DB == false[PROPERTY ME_OK] 2. Match con email esistente nel DB == true[error]

Parametro: Username Formato: ^[\w\-_]+\\$\	
Categorie	Scelte
Lunghezza[LU]	1. LU < 3 OR LU > 30 = false [error] 2. LU >= 3 AND LU <= 30 = true [PROPERTY LU_OK]
Formato[FU]	1. Rispetta il formato == false [error] 2. Rispetta il formato == true [PROPERTY FU_OK]
Match[MU]	1. Match nel Database == false [PROPERTY MU_OK]

	2. Match nel Database == true [error]
--	--

Parametro: Password	
Categorie	Scelte
Lunghezza[LP]	1. LP < 3 OR LP > 255 = false [error] 2. LP >= 3 AND LP <= 255 = true [PROPERTY LP_OK]

Parametro: Conferma password	
Categorie	Scelte
Match[MCP]	1. Match con password == false [error] 2. Match con password == true [PROPERTY MCP_OK]

Codice	Combinazione	Esito
TC_2_1	LE1	Errore: Lunghezza email > 255
TC_2_2	LE2, FE1	Errore: Formato email errato
TC_2_3	LE3, FE2, ME1	Errore: Email esistente nel DB
TC_2_4	LE4, FE3, ME2, LU1	Errore: Lunghezza username > 30
TC_2_5	LE4, FE3, ME2, LU2	Errore: Lunghezza username < 3
TC_2_6	LE4, FE3, ME2, LU3, FU1	Errore: Formato username errato
TC_2_7	LE4, FE3, ME2, LU4, FU2, MU1	Errore: Username esistente nel DB
TC_2_8	LE4, FE3, ME2, LU4, FU3, MU2, LP1	Errore: Lunghezza password > 255
TC_2_9	LE4, FE3, ME2, LU4, FU3, MU2, LP2	Errore: Lunghezza password < 3
TC_2_10	LE4, FE3, ME2, LU4, FU3, MU2, LP3, MCP1	Errore: Conferma password errata
TC_2_11	LE4, FE3, ME2, LU4, FU3, MU2, LP3, MCP2	Corretto

Ban utente

Parametro: Data	
Categorie	Scelte
Data[DT]	1. Data fine ban <= Data corrente == false [PROPERTY DT_OK] 2. Data fine ban <= Data corrente == true [error]

Codice	Combinazione	Esito
TC_3_1	DT1	Errore: data antecedente alla data attuale
TC_3_2	DT2	Corretto

Sottometti Post

Parametro: Titolo	
Categorie	Scelte
Lunghezza Titolo[LTL]	1. $LTL == 0 = \text{false}$ [error] 2. $LTL > 255 = \text{false}$ [error] 3. $LTL \geq 1 \text{ AND } LTL \leq 255 = \text{true}$ [PROPERTY LTL_OK]

Parametro: Testo	
Categorie	Scelte
Lunghezza Testo[LTS]	1. $LTS > 255 = \text{false}$ [error] 2. $LTS \leq 1000 = \text{true}$ [PROPERTY LTS_OK]

Codice	Combinazione	Esito
TC_4_1	LTL1	Errore: Lunghezza Titolo < 1
TC_4_2	LTL2	Errore: Lunghezza Titolo > 255
TC_4_2	LTL3, LTS1	Errore: Lunghezza Testo > 1000
TC_4_3	LTL3, LTS2	Corretto

Sottometti Commento

Parametro: Testo	
Categorie	Scelte
Lunghezza Testo[LTS]	1. $LTS < 1 = \text{false}$ [error] 2. $LTS > 255 = \text{false}$ [error] 3. $LTS \geq 1 \text{ AND } LTS \leq 255 = \text{true}$ [PROPERTY LTS_OK]

Codice	Combinazione	Esito
TC_5_1	LTS1	Errore: Lunghezza Testo < 1
TC_5_2	LTS2	Errore: Lunghezza Titolo > 255
TC_5_3	LTS3	Corretto

Sottometti Risposta

Parametro: Testo	
Categorie	Scelte
Lunghezza Testo[LTS]	1. $LTS < 1 = \text{false}$ [error] 2. $LTS > 255 = \text{false}$ [error] 3. $LTS \geq 1 \text{ AND } LTS \leq 255 = \text{true}$ [PROPERTY LTS_OK]

Codice	Combinazione	Esito
TC_6_1	LTS1	Errore: Lunghezza Testo < 1
TC_6_2	LTS2	Errore: Lunghezza Titolo > 255
TC_6_3	LTS3	Corretto

Aggiungi Sezione

Parametro: Nome

Categorie	Scelte
Lunghezza Nome[LN]	1. LN < 1 = false [error] 2. LN > 50 = false [error] 3. LN >= 1 AND LN <= 50 = true [PROPERTY LN_OK]
Match Nome [MN]	4. Match con nome esistente nel DB == false [PROPERTY MN_OK] 5. Match con nome esistente nel DB == true [error]

Parametro: Descrizione	
Categorie	Scelte
Lunghezza Descrizione[LD]	1. LD > 255 = false [error] 2. LD <= 255 = true [PROPERTY LD_OK]

Parametro: Immagine	
Categorie	Scelte
Formato Immagine[FI]	1. Rispetta il formato == false [error] 2. Rispetta il formato == true [PROPERTY FI_OK]
Dimensione Immagine[DI]	3. Non rispetta la dimensione massima == false [error] 4. Rispetta la dimensione massima == true [PROPERTY DI_OK]

Parametro: Banner	
Categorie	Scelte
Formato Banner[FB]	1. Rispetta il formato == false [error] 2. Rispetta il formato == true [PROPERTY FB_OK]
Dimensione Banner [DB]	3. Non rispetta la dimensione massima == false [error] 4. Rispetta la dimensione massima == true [PROPERTY DB_OK]

Codice	Combinazione	Esito
TC_7_1	LN1	Errore: Lunghezza Testo < 1
TC_7_2	LN2	Errore: Lunghezza Titolo > 255
TC_7_3	LN3, MN1	Errore: Sezione già presente nel DB
TC_7_4	LN3, MN2, LD1	Errore: Lunghezza descrizione superiore a 255
TC_7_5	LN3, MN2, LD2, FI1	Errore: Formato Immagine Errato
TC_7_6	LN3, MN2, LD2, FI2, DI1	Errore: Dimensione Immagine supera il limite
TC_7_7	LN3, MN2, LD2, FI2, DI2, FB1	Errore: Formato Banner Errato
TC_7_8	LN3, MN2, LD2, FI2, DI2, FB2, DB1	Errore: Dimensione Banner supera il limite
TC_7_9	LN3, MN2, LD2, FI2, DI2, FB2, DB2	Corretto

Ricerca Post

Parametro: Testo	
Categorie	Scelte
Lunghezza Testo[LT]	1. LT < 1 = false [error] 2. LT >= 1 = true [PROPERTY LT_OK]

Codice	Combinazione	Esito
TC_8_1	LT1	Errore: Lunghezza Testo < 1
TC_8_2	LT2	Corretto

Testing schedule

Successivamente alla fase di design relativa al testing sarà avviata l'attività di pianificazione.

La scrittura dei casi di test avverrà in parallelo con lo sviluppo del codice, mentre l'esecuzione di tali test avverrà sia durante che al termine dell'implementazione del sistema.

Al termine dello sviluppo saranno rieseguiti i test per garantirne il corretto funzionamento e produrre i report finali.