Test Plan

Simplify3D

Carmine buondonno
Nicola Librera
Alessandro Oliviero

Sommario

1. Introduzione

Lo scopo del documento è quello di specificare gli aspetti manageriali del testing. La seguente fase andrà ad effettuare dei singoli test in modo tale da poter verificare il corretto funzionamento dei vari sottosistemi del software. In caso di rilevazioni di eventuali errori la fase che segue, sé finalizzata alla riparazione dei bug riscontrati. In questo modo si cerca di soddisfare i requisiti funzionali e non funzionali del prodotto. Il seguente documento specifica i requisiti e le componenti che devono essere testati.

2. Riferimenti

- RAD_Simplify3d
- SDD_Simplify3d

3. System Overview

Come specificato nel documento di design (nel paragrafo 3 System Architecture) il sistema è stato suddiviso in 3 sottosistemi: Interface layer, Application layer e Data Storage layer. Ogni sottosistema si occuperà di una logica diversa. Il layer di Interfaccia ha lo scopo di costruire l'interfaccia utente. Il layer di Application si occuperà di implementare tutti i requisiti funzionali specificati nel RAD. Infine, il layer del Data Storage si occuperà di salvaguardare i dati dei Progetti e degli utenti e di mantenerli in memoria al sicuro.

4. Componenti da testare

Il software che s'intende testare come anticipato nei paragrafi precedenti è Simplify3d che presenta come la maggior parte di tutti i prodotti software delle aree di estrema criticità, quelle identificate nel nostro sistema sono le componenti sulle quali si effettuerà il testing:

- Gestione Utente
- Gestione Progetto
- Gestione Commenti e Valutazioni

Di seguito sono elencate le varie unità delle componenti dell'Application Logic Layer.

4.1 Gestione Utente

Componente	Ruoli applicabili	Livello di rischio
Registrazione	Utente	Alto
Login	Utente e Amministratore	Alto
Logout	Utente e Amministratore	Alto
ModificaPassword	Utente e Amministratore	Alto
PasswordDimenticata	Utente e Amministratore	Alto
ConfermaRegistrazione	Utente	Alto

4.2 Gestione Progetto

Componente	Ruoli applicabili	Livello di rischio
Upload	Utente e Amministratore	Alto
AggiungiPreferiti	Utente e Amministratore	Alto
CancellaProgetto	Utente e Amministratore	Alto
Download	Utente e Amministratore	Alto
RimuoviPreferiti	Utente e Amministratore	Alto
Modifica	Utente e Amministratore	Alto

4.3 Gestione Commenti E Valutazioni

Componente	Ruoli applicabili	Livello di rischio
InserisciCommento	Utente e Amministratore	Alto
Valuta	Utente e Amministratore	Alto
EliminaCommento	Utente e Amministratore	Alto
EliminaValutazione	Utente e Amministratore	Alto
InserisciRisposta	Utente e Amministratore	Alto

EliminaRisposta	Utente e Amministratore	Alto

5. Componenti da non testare

Le componenti di Simplify3D che non sono state testate sono le seguenti:

- Visualizza Profilo Utente
- Visualizza Profilo Personale
- Progetti Pubblicati
- Visualizza Notifiche
- Progetti Salvati
- Visualizza Progetto
- Visualizza progetti pubblicati
- Visualizza Progetti Utente
- Ricerca

Per tali componenti non si è ritenuto effettuare testing perché è stato associato un basso o medio livello di rischio.

6. Approccio

L'approccio scelto per la fase di testing si è basata sulla suddivisione del testing in tre fasi: testing di unità, testing di sistema e testing di integrazione. In questo modo è possibile controllare in maniera efficiente ogni sottosistema implementato e di trovare e corregge eventuali bug prima dell'inserimento all'interno del sistema.

- **6.1 Testing di unità:** In questa fase il testing viene effettuato testando ogni singola funzione implementata nel Layer Data Storage. Tramite il framework JUnit su Eclipse, verranno quindi testati i metodi nei Manager dei 3 sottosistemi: Gestione Utente, Gestione Progetto, Gestione Commenti e Valutazioni.

- **6.2 Testing di sistema:** In questa fase il testing viene effettuato in maniera dinamica andando ad utilizzare la tecnica di Black-Box. Per effettuare questa tipologia di testing abbiamo utilizzato Selenium IDE, un'estensione di Google Chrome, ovvero un framework portatile che fornisce gli strumenti per creare test funzionali senza utilizzare particolari linguaggi.

- **6.3 Testing di integrazione:** In questa fase il testing viene effettuato sull'Application layer e quindi su tutte le Servlet del nostro sistema. Siccome l'Application Layer comunica con il layer di DataStorage e quello di Interface, in ogni test case, abbiamo fatto interagire i servizi offerti dai livelli in questione.

7. Criteri di successo/errore

Il testing ha successo se l'output osservato è diverso dall'output atteso: ciò significa che parliamo di successo se il test individuerà una failure. In tal caso questa verrà analizzata e, se legata ad un fault, si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica non abbia impattato su altri componenti del sistema. Viceversa parliamo di fallimento se il test non riesce ad individuare un errore.

8. Testing materials

Come supporto alla fase di testing di Sistema si utilizzerà un browser Internet, la scelta è ricaduta su Google Chrome. Come strumenti di supporto verrà utilizzato: il framwork Selenium aggiungendo a Google Chrome Selenium IDE.
Per la fase di testing di integrazione abbiamo utilizzato il framework JUnit e spring.
Per la fase di testing di unità abbiamo utilizzato il framework JUnit.

9. Test Cases

I test case fanno riferimento al documento: "Testing Case Specification".