

Test plan

Ingegneria del software – anno accademico 2017/2018

nome progetto: genFreeTest

Nome: Simone Marco Palo

Matricola: 02542

4.1 - Relazione con gli Altri Documenti

4.1.1 - Relazione con il documento di Analisi dei Requisiti

I test dovranno tenere conto delle specifiche espresse nel RAD. Naturalmente, si darà una maggiore rilevanza ai “casi limite”, cioè tutti quei casi che sono al limite del nostro dominio di dati d'ingresso.

4.1.2 - Relazioni con il documento di System Design

Il testing dovrà garantire la coerenza tra il software e gli obiettivi di design definiti in fase di System Design, specificati nel SDD. Si analizzerà la struttura del sistema per determinare le differenze presenti tra essa e quella stabilita durante la fase di System Design.

4.2 - Panoramica del sistema

GenFreeTest è un sito web caratterizzato da una architettura di tipo **Client/Server**, in quanto prevede l'interazione tra un insieme di nodi (detti client) ed un server che gestisce i dati persistenti in un database.

Le caratteristiche da testare per garantire il corretto funzionamento del sistema saranno:

- correttezza: verrà testato che le operazioni vengano eseguite correttamente così come dalla specifica dei requisiti sono stati designati.
- Sicurezza: verrà testata la sicurezza del sistema per determinare se ci sono falle nella sicurezza
- Usabilità: verrà analizzata ogni forma di interazione tra l'utente e il sistema per determinare se sarà necessario aggiungere messaggi di errore, in casi di errore, o di notifica, in caso di operazioni eseguite con successo.

4.3 - Funzionalità da testare e da non testare

4.3.1 – Funzionalità da testare

Ho deciso di utilizzare lo **unit testing** per testare quasi tutte le form presenti nel sistema. Le form saranno testate attraverso la tecnica Black Box, ovvero verificherò il comportamento del sistema dopo aver dato in input alla form una serie di valori, che potranno essere errati oppure corretti. Utilizzerò, invece, l'**Integration Testing** per testare il comportamento di ciascuna funzionalità del sistema che sono fornite da uno o più sottosistemi. Più precisamente utilizzerò il testing Black Box per testare le seguenti form:

- form per l'iscrizione del professore (contenuto nella pagina **creaAccount.html**)
- form per l'iscrizione dello studente (contenuto nella pagina **creaAccount.html**)
- form per il login del professore (contenuto nella pagina **index.html**)
- form per il login dello Studente (contenuto nella pagina **index.html**)
- form per la creazione del test (contenuto nella pagina **areaTest.html**)
- form per la creazione delle domande (contenuto nella pagina **creaDomande.html**)
- form per la modifica del test (contenuto nella pagina **gestioneTest.html**)
- form per la modifica della password del professore (contenuta nella pagina **cambiaPasswordProfessore.html**)
- form per la modifica della password dello studente (contenuta nella pagina **cambiaPasswordStudente.html**).

4.3.2 – Funzionalità da non testare

Non testerò le funzionalità appartenenti a componenti molto semplici, il cui testing è implicito nella loro stessa struttura. Inoltre non testerò molte funzionalità appartenenti a componenti prettamente grafiche che non offrono funzionalità di spicco per il corretto funzionamento del sistema.

4.4 – Criteri di successo e di insuccesso

4.4.1 – Criteri di successo

Se nella fase di testing si nota che una componente o un insieme di componenti ha un comportamento diverso da quello richiesto e previsto nello sviluppo e non caratterizzato da errori strettamente tecnici, allora il test interessato in tale verifica ha insuccesso. Tale insuccesso, però, significa il successo della componente o dell'insieme di componenti testato.

I criteri di successo del sistema software previsti per la fase di testing sono i seguenti:

- uguaglianza tra il comportamento osservato e il comportamento atteso della componente o dell'insieme delle componenti.
- mancanza di errori nel codice e nella struttura logica della componente o dell'insieme di componenti.

4.4.2 – Criteri di insuccesso

Si ha un successo, se la componente o l'insieme di componenti testate ha un comportamento diverso dal comportamento previsto, o anche se ci sono errori logici o di altro tipo. Si noti che il successo del test è in verità un insuccesso della componente o dell'insieme di componenti testati.

4.5 – Approccio

Le tipologie di testing che saranno eseguite al fine di verificare il corretto funzionamento del sistema sono:

- **Unit testing:** un testing che si focalizza su una singola unità del sistema.
- **Integration testing:** un testing che coinvolge una o più funzionalità del

sistema o più unità del software.

4.5.1 – Unit Testing

Effettuerò un testing di tipo test case su alcune unità del sistema. Ogni singola unità sarà testata con lo Unit Testing attraverso la tecnica detta “Black Box”, cioè verificherò il comportamento di ogni singola unità testata, attraverso l'inserimento di input e la valutazione dell'output.

4.5.2 – Integration Testing

Dopo aver testato le singole unità del software, effettuerò un controllo su alcune parti del sistema che coinvolgono una o più componenti del sistema. Detto in altri termini, testerò collezioni di insiemi, ognuno dei quali formato da uno o più componenti, con lo scopo di determinare per ogni insieme della collezione la presenza di errori o fallimenti.

4.6 – Sospensione e ripristino

Nel testare il sito genFreeTest, potranno esserci casi in cui bisogna sospendere il testing e in cui in cui bisogna ripristinarlo.

4.6.1 – Sospensione

Durante la fase di Unit Testing, il testing sarà sospeso se si sono riscontrati errori o fallimenti riguardanti una unità del software e la sospensione riguarderà soltanto il testing di tale unità. Se il testing di un'unità viene sospeso, si passerà alla correzione dell'unità. Il testing di ognuna delle unità del software è indipendente da quello delle altre. Durante la fase di Integration Testing, il testing sarà sospeso quando saranno riscontrati errori o fallimenti per una o più funzionalità testate. In tal caso, si procederà alla correzione del codice.

4.6.2 – Ripristino

Il ripristino del testing di una componente o un gruppo di componenti del software sarà svolto quando gli errori riscontrati per tale componente o per tale gruppo di componenti saranno corretti. Inoltre, dopo che la componente è stata modificata, verranno eseguiti su di essa gli stessi test precedenti allo scopo di confermare la sua correzione.