

Università degli Studi di Salerno

Dipartimento di Informatica



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA
DIPARTIMENTO DI ECCELLENZA

Corso di Ingegneria Gestione ed Evoluzione del Software

Aniello Florido & Alexander Minichino

November 16, 2020

Metric 3.0 Web Edition

Test Plan

Coordinatore del progetto:

| Nome |
|----------------------|
| Prof.Andrea De Lucia |

Partecipanti:

| Nome | Matricola |
|---------------------|------------------|
| Aniello Florido | 0522500625 |
| Alexander Minichino | 0522500644 |

Revision History

| Data | Versione | Descrizione | Autori |
|-------------|-----------------|--------------------|---|
| 19/03/2019 | 1.0 | Prima stesura | Aniello Florido Alexander Minichino |

Contents

| | | |
|----------|--------------------------------------|----------|
| 1 | Introduzione | 4 |
| 2 | Panoramica del sistema | 4 |
| 3 | Funzionalità da testare | 4 |
| 4 | Approccio | 5 |
| 4.1 | Altre tipologie di test | 5 |
| 5 | Test cases | 6 |
| 5.1 | Tool Category Partition | 6 |
| 6 | Specifica dei Test Cases | 7 |
| 6.1 | Registrazione Utente | 7 |
| 6.1.1 | Specifiche | 7 |
| 6.1.2 | Test case | 8 |
| 6.2 | Login utente | 9 |
| 6.2.1 | Specifiche | 9 |
| 6.2.2 | Test case | 10 |
| 6.3 | Creazione nuova analisi | 11 |
| 6.3.1 | Specifiche | 11 |
| 6.3.2 | Test case | 11 |
| 6.4 | Richiesta Assistenza | 12 |
| 6.4.1 | Specifiche | 12 |
| 6.4.2 | Test case | 12 |
| 6.5 | Visualizzazione Dashboard | 14 |
| 6.5.1 | Specifiche | 14 |
| 6.5.2 | Test case | 14 |
| 6.6 | Visualizzazione Storico | 14 |
| 6.6.1 | Specifiche | 14 |
| 6.6.2 | Test case | 14 |
| 6.7 | Visualizzazione Analisi | 15 |
| 6.7.1 | Specifiche | 15 |
| 6.7.2 | Test case | 15 |
| 6.8 | Download Risultato Analisi | 15 |
| 6.8.1 | Specifiche | 15 |
| 6.8.2 | Test case | 15 |
| 6.9 | Logout | 15 |
| 6.9.1 | Specifiche | 15 |
| 6.9.2 | Test case | 16 |

1 Introduzione

Lo scopo di questo documento è quello di analizzare, pianificare e gestire lo sviluppo e le attività di testing del progetto Metric 3.0 Web Edition.

Questa sessione di lavoro va a verificare il corretto funzionamento del sistema Metric 3.0 Web Edition in diversi casi, studiati appositamente per testare ogni singola caratteristica e funzionalità del sistema, al fine di ottenere un corretto funzionamento.

Dai risultati ottenuti, qualora fossero presenti delle anomalie, saranno apportate delle modifiche per migliorare la stabilità del sistema.

2 Panoramica del sistema

Il tool Metric 3.0 è stato sviluppato presso l'Università degli studi di Salerno. Esso esegue un'analisi dettagliata del codice sorgente Java, ottenendo delle informazioni che vanno a descrivere, attraverso una serie di metriche, la qualità del codice analizzato, così da offrire agli sviluppatori un valido aiuto nella comprensione di progetti con scarsa documentazione.

Il tool è stato implementato utilizzando la libreria JDT Core, la quale permette l'accesso all'AST (Abstract Syntax Tree), fornendo gli strumenti e le strutture dati per analizzarlo.

I nodi dell'AST sono gli elementi presenti nel codice sorgente, come classi, metodi, variabili, ecc.

Metric 3.0 Web Edition è la versione web del tool Metric.

E' stata realizzata un'interfaccia che permettesse l'utilizzo delle funzionalità di Metric anche attraverso un web browser.

Inoltre è stata inserita una storicizzazione delle analisi e una sezione per il profilo utente.

3 Funzionalità da testare

Nel testing si andranno a considerare le funzionalità principali di Metric 3.0 Web Edition, allo stesso tempo verranno testate anche parti del funzionamento del tool Metric 3.0. LE funzionalità che verranno testate sono:

- **Autenticazione e Registrazione Nuovo Utente:** questa funzione permette all'utente la registrazione al sistema e l'accesso ad esso. Se l'utente non è autenticato, non può avere accesso alle funzionalità del sistema.

- **Creazione di un nuova analisi:** questa funzionalità permette all'utente, dopo aver inserito i dati riguardo la nuova analisi e aver caricato il file contenente il progetto da testare, la creazione di una nuova analisi e la visualizzazione dei risultati forniti dal tool.
- **Richiesta di informazioni:** attraverso questa sezione l'utente potrà inviare una email al team di sviluppo per richiedere delle informazioni sul sistema.

4 Approccio

Nella sessione di testing di Metric 3.0 Web Edition verrà utilizzato un approccio di tipo "BLACK BOX", il quale prevede che i test vengano effettuati in modo tale da non scendere nel dettaglio del codice, ma ci si basi sulle specifiche delle funzionalità da testare.

L'approccio alla fase di testing si divide in due fasi:

- Testing di sistema sul sistema originale: Creazione di uno script batch, che permetta di testare il funzionamento del vecchio sistema Metric 3.0.
- Testing di sistema sul sistema modificato: è un test funzionale che andrà a verificare la funzionalità dell'intero sistema assemblato.

Testing di sistema sul sistema originale

Per verificare che il funzionamento del tool Metric 3.0 sia corretto, verrà creato uno script batch che, prendendo in input un progetto in Java, avvierà l'analisi statica del codice del progetto caricato.

In tal modo, si andrà a verificare il corretto funzionamento del tool Metric 3.0 prima di procedere alla creazione della versione web.

Testing di sistema sul sistema modificato

Nel testing di sistema si andrà a testare la correttezza dell'intero sistema.

Tale test è considerato il test più critico in quanto, può risultare abbastanza complesso andare alla ricerca di eventuali errori, poichè sono impegnati tutti i sottosistemi.

Per effettuare tale test sarà utilizzato il framework Selenium, il quale fornisce degli strumenti per testare sistemi web-based.

4.1 Altre tipologie di test

Dato che non sono state apportate modifiche sostanziali al vecchio sistema e che esso risulta essere sprovvisto di testing, si è deciso di effettuare direttamente il

testing di sistema.

Ovvero, avendo implementato una nuova interfaccia grafica senza modificare funzionalità del sistema esistente, si è ritenuto più importante effettuare il testing delle funzionalità esterne del sistema.

A tal proposito, una buona suddivisione dei casi di test a livello di sistema può risultare utile nell'individuazione di fault anche a livello di singole componenti.

5 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).

5.1 Tool Category Partition

Per l'esecuzione del category partition, partendo dalle categorie e dalle scelte individuate, è stato utilizzato un tool sviluppato dalla University of California Irvine chiamato TSLgenerator.

Tale tool permette di definire le specifiche di testing sotto forma di TSL (Test Script Language), nello specifico possono essere definite categorie, scelte e constraints utilizzati nel category partition.

Una volta definite le specifiche, esse vengono passate al tool, il quale produce, secondo il metodo del category partition, l'insieme dei test case.

Il tool è scaricabile al seguente [link](#).

6 Specifica dei Test Cases

6.1 Registrazione Utente

6.1.1 Specifiche

#Parametro

Username:

#Categorie

LUser:

- 1 lunghezza 0-4. [error]
- 2 lunghezza 5-15. [property LUserOk]
- 3 lunghezza >=16. [error]

PUser:

- 1 presente in file. [error]
- 2 non presente in file. [if LUserOk][property PUserOk]

#Parametro

Password:

#Categorie

LPass:

- 1 lunghezza 0-4. [error]
- 2 lunghezza 5-8. [property LPassOk]
- 3 lunghezza >=9. [error]

#Parametro

Conferma Password:

#Categorie:

EQPasswordConfirm:

- 1 uguale a password. [if LPassOk][property EQPasswordConfirmOk]
- 2 diversa da password. [error]

#Parametro

Name:

#Categorie

PName:

- 1 presente. [property PNameOk]
- 2 non presente. [error]

#Parametro

Surname:

#Categorie

PSurname:

1 presente.[property PSurnnameOk]
2 non presente. [error]

Company:

#Categorie
PCompany:
1 presente.
2 non presente.

6.1.2 Test case

Test Case 1.1 <error>
LUser : 1 lunghezza 0-4

Test Case 1.2 <error>
LUser : 3 lunghezza >=16

Test Case 1.3 <error>
PUser : 1 presente in file

Test Case 1.4 <error>
LPass : 1 lunghezza 0-4

Test Case 1.5 <error>
LPass : 3 lunghezza >=9

Test Case 1.6 <error>
EQPasswordConfirm : 2 diversa da password

Test Case 1.7 <error>
PName : 2 non presente

Test Case 1.8 <error>
PSurnname : 2 non presente

Test Case 1.9 (Key = 2.2.2.1.1.1.1.)

| | | |
|-------------------|---|------------------------|
| LUser | : | 2 lunghezza 5-15 |
| PUser | : | 2 non presente in file |
| LPass | : | 2 lunghezza 5-8 |
| EQPasswordConfirm | : | 1 uguale a password |
| PName | : | 1 presente |
| PSurname | : | 1 presente |
| PCompany | : | 1 presente |

Test Case 1.10 (Key = 2.2.2.1.1.1.2.)

| | | |
|-------------------|---|------------------------|
| LUser | : | 2 lunghezza 5-15 |
| PUser | : | 2 non presente in file |
| LPass | : | 2 lunghezza 5-8 |
| EQPasswordConfirm | : | 1 uguale a password |
| PName | : | 1 presente |
| PSurname | : | 1 presente |
| PCompany | : | 2 non presente |

6.2 Login utente

6.2.1 Specifiche

#Parametro

Username:

#Categorie

PUser:

1 presente nella form. [property PUserOk]

2 non presente nella form. [error]

PUserFile:

1 presente nel file. [if PUserOk] [property PUserFileOk]

2 non presente nel file. [error]

#Parametro

Password:

#Categorie

PPass:

1 presente nella form. [if PUserOk] [property PPassOk]

2 non presente nella form. [error]

CorrectPassword:

1 password corretta. [if PUserOk] [property CorrectPasswordOk]

2 password non corretta. [error]

6.2.2 Test case

Test Case 2.1 <error>

PUser : 2 non presente nella form

Test Case 2.2 <error>

PUserFile : 2 non presente nel file

Test Case 2.3 <error>

PPass : 2 non presente nella form

Test Case 2.4 <error>

CorrectPassword : 2 password non corretta

Test Case 2.5 (Key = 1.1.1.1.)

PUser : 1 presente nella form

PUserFile : 1 presente nel file

PPass : 1 presente nella form

CorrectPassword : 1 password corretta

6.3 Creazione nuova analisi

6.3.1 Specifiche

#Parametro

NameProject:

#Categorie

PName:

1 nome progetto presente nella form. [property PNameOk]

2 nome progetto NON presente nella form. [error]

FName:

1 rispetta il formato [0-9A-Za-z\]. [if PNameOk][property FNameOk]

2 NON rispetta il formato [0-9A-Za-z\]. [error]

#Parametro

FileProject:

#Categorie

PFile:

1 file presente nella form. [property PFileOk]

2 file NON presente nella form. [error]

6.3.2 Test case

Test Case 3.1 <error>

PName : 2 nome progetto NON presente nella form

Test Case 3.2 <error>

FName : 2 NON rispetta il formato [0-9A-Za-z\]

Test Case 3.3 <error>

PFile : 2 file NON presente nella form

Test Case 3.4 (Key = 1.1.1.)

PName : 1 nome progetto presente nella form

FName : 1 rispetta il formato [0-9A-Za-z\]

PFile : 1 file presente nella form

6.4 Richiesta Assistenza

6.4.1 Specifiche

#Parametro

email:

#Categorie

PEmail:

- 1 email presente nella form. [property PEmailOk]
- 2 email non presente nella form. [error]

FEmail: #Dato il formato `^[A-z0-9\.\+_-]+@[A-z0-9\._-]+\.[A-z]{2,`

- 1 rispetta il formato. [if PEmailOk][property FEmailOk]
- 2 NON rispetta il formato.[error]

#Parametro

Info:

#Categorie

PInfo:

- 1 info presente nella form. [property PInfoOk]
- 2 info non presente nella form. [error]

#Parametro

Subject:

#Categorie

PSubject:

- 1 presente.
- 2 non presente.

6.4.2 Test case

Test Case 4.1 <error>

PEmail : 2 email non presente nella form

Test Case 4.2 <error>

FEmail : 2 NON rispetta il formato

Test Case 4.3 <error>

PInfo : 2 info non presente nella form

Test Case 4.4 (Key = 1.1.1.1.)

PEmail : 1 email presente nella form

FEmail : 1 rispetta il formato

PInfo : 1 info presente nella form

PSubject : 1 presente

Test Case 4.5 (Key = 1.1.1.2.)
PEmail : 1 email presente nella form
FEmail : 1 rispetta il formato
PInfo : 1 info presente nella form
PSubject : 2 non presente

6.5 Visualizzazione Dashboard

6.5.1 Specifiche

#Parametro

page:

#Categorie

PName:

1 nome pagina = dashboard. [property PNameOk]

2 nome pagina != dashboard. [error]

6.5.2 Test case

Test Case 5.1 <error>

PName : 2 nome pagina != dashboard

Test Case 5.2 (Key = 1.)

PName : 1 nome pagina = dashboard

6.6 Visualizzazione Storico

6.6.1 Specifiche

#Parametro

page:

#Categorie

PName:

1 nome pagina = hystory. [property PNameOk]

2 nome pagina != hystory. [error]

6.6.2 Test case

Test Case 6.1 <error>

PName : 2 nome pagina != hystory

Test Case 6.2 (Key = 1.)

PName : 1 nome pagina = hystory

6.7 Visualizzazione Analisi

6.7.1 Specifiche

#Parametro

Analysis id:

#Categorie

PAnalysisId:

1 id presente in archivio. [property PAnalysisIdOk]

2 id non presente in archivio. [error]

6.7.2 Test case

Test Case 7.1 <error>

PAnalysisId : 2 id non presente in archivio

Test Case 7.2 (Key = 1.)

PAnalysisId : 1 id presente in archivio

6.8 Download Risultato Analisi

6.8.1 Specifiche

#Parametro

Log file:

#Categorie

PLogFile:

1 log presente in archivio. [property PLogFileOk]

2 log non presente in archivio. [error]

6.8.2 Test case

Test Case 8.1 <error>

PLogFile : 2 log non presente in archivio

Test Case 8.2 (Key = 1.)

PLogFile : 1 log presente in archivio

6.9 Logout

6.9.1 Specifiche

#Parametro

JSESSIONID:

#Categorie

PJSESSIONID:

1 session presente nei cookie. [property PJSESSIONIDOk]

2 session non presente nei cookie. [error]

6.9.2 Test case

Test Case 9.1 <error>

PJSESSIONID : 2 session non presente nei cookie

Test Case 9.2 (Key = 1.)

PJSESSIONID : 1 session presente nei cookie