

Github 專案分析系統

System Testing Document (STD)

Version: 1.0

Team#2

Name	ID	E-mail
劉文揚	110598096	t110598096@ntut.org.tw
謝狄烽	110598087	t110598087@ntut.org.tw
林好潔	110598055	t110598055@ntut.org.tw
曹智凱	110598085	t110598085@ntut.org.tw

**Department of Computer Science & Information Engineering
National Taipei University of Technology**

01/09/2022

Table of Contents

Section 1 Objectives and Acceptance Criteria	1
1.1 Test Scope	1
1.2 Test Acceptance Criteria	1
Section 2 Personnel and Responsibility	2
2.1 Personnel	2
2.2 Testing Schedule	2
2.3 Responsibility	2
Section 3 Testing Environment	4
3.1 Operational Environment	4
3.2 Hardware Specification and Configuration	4
3.3 Software Specification and Configuration	4
3.4 Test Data Sources	4
3.5 Tools and Equipments	5
Section 4 Test Cases	6
Section 5 Test Procedure and Schedule	13
Section 6 Test Results and Analysis	錯誤! 尚未定義書籤。
Glossary	18
References	19
Traceability Matrix	錯誤! 尚未定義書籤。
A Traceability Matrix of Test Cases vs Requirements	錯誤! 尚未定義書籤。
B Traceability Matrix of Test Cases vs Use Cases	錯誤! 尚未定義書籤。

Section 1 Objectives and Acceptance Criteria

1.1 Test Scope

GitHub 專案分析系統(GitHub Repository Analysis System, GRAS)，提供一個整合的資訊平台使得團隊能夠瞭解專案的各項資訊，透過視覺化的資料令團隊能夠快速的分析目前的開發的狀況。

本系統分為五個子系統

- 身分驗證子系統(User Identity Verification and Management Subsystem, UIVMS)
- 專案管理子系統(Repository Management Subsystem, RMS)
- 資料庫管理子系統(Database Management Subsystem, DBMS)
- 分析管理子系統(Analysis Management Subsystem, AMS)
- SonarQube 管理子系統(SonarQube Management Subsystem, SQMS)

1.2 Test Acceptance Criteria

本系統須滿足的測試驗收準則為下列幾點:

- 本系統的所有測試計畫結果皆符合預期結果才可通過。
- 需要對所有需求作完整測試。
- 測試程序需要依照測試計畫制定的程序進行，所有的測試結果都需要符合預期測試結果(Expected Result)才能接受。
- 測試計畫中，Test Case 為測試最基本單位，若某 Test Case 未測試通過時，則需要進行該單元的測試。

Section 2 Personnel and Responsibility

2.1 Personnel

成員名單			
姓名	職位	縮寫	備註
劉文揚	Developer and Tester	WY	前端開發兼測試
謝狄烽	Developer and Tester	DF	前端開發兼測試
林好潔	Developer and Tester	YJ	後端開發兼測試
曹智凱	Developer and Tester	ZK	後端開發兼測試

2.2 Testing Schedule

時程

項目	開始日期	結束日期
單元測試	2021/11/15	2021/12/15
整合測試	2021/12/18	2021/12/27
回歸測試	2021/12/30	2022/01/05
驗收測試	2022/01/06	2022/01/09

Iterations

項目	開始日期	結束日期
Prototyping	2021/11/01	2021/11/09
Increment1	2021/12/01	2021/12/08
Increment2	2022/01/03	2022/01/10

2.3 Responsibility

- UIVMS 子系統

Testing Activity	Personnel	測試時程
UIVMS-TC01	WY、DF	1 Day
UIVMS-TC02	YJ、ZK	1 Day

- RMS 子系統

Testing Activity	Personnel	測試時程
RMS-TC01	WY、DF	1 Day
RMS-TC02	WY、DF	1 Day

- DBMS 子系統

Testing Activity	Personnel	測試時程
DBMS-TC01	YJ、ZK	1 Day
DBMS-TC02	YJ、ZK	1 Day

- AMS 子系統

Testing Activity	Personnel	測試時程
AMS-TC01	YJ、ZK	1 Day
AMS-TC02	YJ、ZK	

- SQMS 子系統

Testing Activity	Personnel	測試時程
SQMS-TC01	WY、DF	1 Day
SQMS-TC02	WY、DF	1 Day

Section 3 Testing Environment

3.1 Operational Environment

本測試於 Windows Server 環境下進行，測試前須先進行以下設定：

- 1.與資料庫主機進行連線
- 2.與 SonarQube 主機進行連線
- 3.進行 Tomcat 設定
- 4.設定網際網路確認可與 GitHub 連線

已註解 [1]:

3.2 Hardware Specification and Configuration

項目	名稱	數量	規格	備註
1.	個人電腦-1	1	CPU:I7-4770 RAM:DDR3 24GB	
2.	個人電腦-2	1	CPU:I7-9700 RAM:DDR4 20GB	
3.	個人電腦-3	1	CPU:I7-10700 RAM:DDR4 24GB	
4.	個人電腦-4	1	CPU:I7-8700 RAM:DDR4 24GB	

3.3 Software Specification and Configuration

項目	名稱	版本	備註
1.	Windows10	企業版	
2.	IntelliJ IDEA	Community Edition 2021.2.1	
3.	MySQL Workbench	8.0 CE	
4.	MySQL	5.7	
5.	Chrome	96.0.4664.110	
6.	Angular	13.1.2	
7.	Node.js	8.1.0	

3.4 Test Data Sources

- Testing UIVMS System Data
- Testing RMS System Data
- Testing DBMS System Data
- Testing AMS System Data
- Testing SQMS System Data

3.5 Tools and Equipments

- 單元測試：本系統內部測試採用 IntelliJ IDEA + JUnit 測試，測試每項功能的單元測試後，待確認無問題後再繼續開發。
- 整合測試：本系統內部測試採用 IntelliJ IDEA + JUnit 測試，測試每項功能的整合測試後，待確認功能符合預想後再繼續開發。
- 系統測試：系統測試計劃採用使用者直接操作的方式來測試系統。

Section 4 Test Cases

4.1 身分驗證子系統(User Identity Verification and Management Subsystem, UIVMS)

Test Cases

ID	UIVMS-TC01
Name	OAuth 登入成功轉跳頁面測試
Reference	
Precondition	使用者已成功由 OAuth 登入 GitHub
Steps	1. 使用 OAuth 登入 2. 進入 Loading 畫面 3. 轉跳本系統專案選擇畫面
Expected Result	成功進入專案選擇畫面，並且顯示該帳戶的專案

ID	UIVMS-TC02
Name	OAuth 登入測試
Reference	
Precondition	使用者已有 GitHub 帳號
Steps	1.在 Login Page 選擇「Login With GitHub」 Button 2.正確輸入 GitHub 帳號及密碼 3.點擊 Sign in
Expected Result	使用者應能正確進入系統

4.2 專案管理子系統(Repository Management Subsystem, RMS)

Test Cases

ID	RMS-TC01
Name	正確顯示專案測試
Reference	
Precondition	使用者已創建數個專案
Steps	1 點擊「My Project」Button 2 進入「choose-project」Page
Expected Result	應正確顯示使用者已創建的所有專案

ID	RMS-TC02
Name	正確顯示專案測試
Reference	
Precondition	使用者已創建數個專案
Steps	1. 點擊「My Project」Button 2. 進入「choose-project」Page
Expected Result	應正確顯示使用者已創建的所有專案

ID	RMS-TC03
Name	正確刪除專案測試
Reference	
Precondition	使用者已創建專案
Steps	1. 點擊「My Project」Button 2. 進入「choose-project」Page 3. 點擊 專案右方的 ⊗ 4. 點擊「確定」，刪除專案。
Expected Result	應正確顯示使用者已創建的所有專案

ID	RMS-TC04
Name	正確新增專案測試
Reference	
Precondition	使用者已登入系統

Steps	1. 點擊 「New Project」 Button 2. 輸入專案名稱、專案敘述、欲導入網址。 3. 按下Create Project 按鈕 4. 提示新增 Project 成功，跳轉至專案畫面。
Expected Result	應正確顯示使用者新創建的專案

4.3 資料庫管理子系統(Database Management Subsystem, DBMS)

Test Cases

ID	DBMS-TC01
Name	OAuth 登入資料庫寫入測試
Reference	
Precondition	使用者已有 GitHub 帳號並未登入 GRAS 系統過
Steps	1.在 Login Page 選擇「Login With GitHub」 Button 2.正確輸入 GitHub 帳號及密碼 3.點擊 Sign in
Expected Result	資料庫應已新增該 GitHub 帳號相關資訊

ID	DBMS-TC02
Name	OAuth 登入資料庫不重複寫入測試
Reference	
Precondition	使用者已有 GitHub 帳號並已登入 GRAS 系統過
Steps	1.在 Login Page 選擇「Login With GitHub」 Button 2.正確輸入 GitHub 帳號及密碼 3.點擊 Sign in
Expected Result	資料庫應保持原有 GitHub 帳號相關資訊， 不覆寫或出現多筆同使用者資訊，且能根據登入的 githubId 取出對應的 user。

4.4 分析管理子系統(Analysis Management Subsystem, AMS)

Test Cases

ID	AMS-TC01
Name	Commit Trend 顯示測試
Reference	
Precondition	使用者已建立專案
Steps	1. 於 MyProject 頁面點擊其中一個專案 2. 點擊欲察看 Repo 3. 點擊 Commit Trend 4. 進入 Commit Trend 頁面
Expected Result	應正確顯示 Total Commits 及個人 Commits 等資訊

ID	AMS-TC02
Name	Issue Track 顯示測試
Reference	
Precondition	使用者已建立專案
Steps	1. 於 MyProject 頁面點擊其中一個專案 2. 點擊欲察看 Repo 3. 點擊 Issue Track 4. 進入 Issue Track 頁面
Expected Result	應正確顯示該 Repo Github 已建立的 Issue

4.5 SonarQube 管理子系統(SonarQube Management Subsystem, SQMS)

Test Cases

ID	SQMS-TC01
Name	查看 sonarqube 分析結果總覽測試
Reference	
Precondition	使用者的專案已於 SonarQube 上分析完成。
Steps	<ol style="list-style-type: none"> 1. 建立一筆測試專案 2. 首頁選擇「My Project」頁面，進入專案總覽畫面 3. 選擇測試專案 4. 選擇欲查看 SonarQube 分析結果的 Repo 5. 點選「SONAR QUBE」
Expected Result	系統進入「SONAR QUBE」頁面，並可正確查看 SonarQube 分析結果。

ID	SQMS-TC02
Name	查看 sonarqube 歷史分析結果測試
Reference	
Precondition	使用者的專案已於 SonarQube 上分析完成。
Steps	<ol style="list-style-type: none"> 1. 建立一筆測試專案 2. 首頁選擇「My Project」頁面，進入專案總覽畫面 3. 選擇測試專案 4. 選擇欲查看 SonarQube 分析結果的 Repo 5. 點選「SONAR QUBE」 6. 點選「History」
Expected Result	1. 系統進入「SONAR QUBE History」頁面，並可正確查看 SonarQube 分析時間和其結果。

ID	SQMS-TC03
Name	查看 sonarqube 分析 bug 詳細列表測試
Reference	SQMS-TC01
Precondition	使用者的專案已於 SonarQube 上分析完成。
Steps	<ol style="list-style-type: none"> 1. 建立一筆測試專案 2. 首頁選擇「My Project」頁面，進入專案總覽畫面 3. 選擇測試專案 4. 選擇欲查看 SonarQube 分析結果的 Repo 5. 點選「SONAR QUBE」 6. 點選 BUG 旁的「▼」，展開 Bug 列表。

Expected Result	系統展開「Bug」列表，並可正確查看 SonarQube Bug 分析結果。
-----------------	--

ID	SQMS-TC04
Name	查看 sonarqube 分析 code smell 詳細列表測試
Reference	SQMS-TC01
Precondition	使用者的專案已於 SonarQube 上分析完成。
Steps	<ol style="list-style-type: none"> 1. 建立一筆測試專案 2. 首頁選擇「My Project」頁面，進入專案總覽畫面 3. 選擇測試專案 4. 選擇欲查看 SonarQube 分析結果的 Repo 5. 點選「SONAR QUBE」 6. 點選 CODE SMELL 旁的「▼」，展開 Code Smell 列表。
Expected Result	系統展開「Code Smell」列表，並可正確查看 SonarQube Code Smell 分析結果。

ID	SQMS-TC05
Name	查看 sonarqube 分析 Vulnerability 詳細列表測試
Reference	SQMS-TC01
Precondition	使用者的專案已於 SonarQube 上分析完成。
Steps	<ol style="list-style-type: none"> 1. 建立一筆測試專案 2. 首頁選擇「My Project」頁面，進入專案總覽畫面 3. 選擇測試專案 4. 選擇欲查看 SonarQube 分析結果的 Repo 5. 點選「SONAR QUBE」 6. 點選 VULNERABILITY 旁的「▼」，展開 Vulnerability 列表。
Expected Result	系統展開「Vulnerability」列表，並可正確查看 SonarQube Vulnerability 分析結果。

Section 5 Test Procedure and Schedule

5.1 User Identity Verification and Management Subsystem(UIVMS)

5.1.1 Schedule

Deliverable	Responsibility	Completion Date
Develop Test Case	Tester 、 Developer	
Test Case Review	Tester 、 Developer	

5.1.2 Test Cases Sequence

Test Cases Sequences
UIVMS-TC01
UIVMS-TC02

5.1.3 Termination Condition

- No defects are found
- Defects are fixed already

5.2 Repository Management Subsystem (RMS)

5.2.1 Schedule

Deliverable	Responsibility	Completion Date
Develop Test Case	Tester 、 Developer	
Test Case Review	Tester 、 Developer	

5.2.2 Test Cases Sequence

Test Cases Sequences
RMS -TC01
RMS -TC02

5.2.3 Termination Condition

- No defects are found
- Defects are fixed already

5.3 Database Management Subsystem (DBMS)

5.3.1 Schedule

Deliverable	Responsibility	Completion Date
Develop Test Case	Tester 、 Developer	
Test Case Review	Tester 、 Developer	

5.3.2 Test Cases Sequence

Test Cases Sequences
DBMS -TC01
DBMS -TC02

5.3.3 Termination Condition

- No defects are found
- Defects are fixed already

5.4 Analysis Management Subsystem (AMS)

5.4.1 Schedule

Deliverable	Responsibility	Completion Date
Develop Test Case	Tester 、 Developer	
Test Case Review	Tester 、 Developer	

5.4.2 Test Cases Sequence

Test Cases Sequences
AMS -TC01
AMS -TC02

5.4.3 Termination Condition

- No defects are found
- Defects are fixed already

5.5 SonarQube Management Subsystem (SQMS)

5.5.1 Schedule

Deliverable	Responsibility	Completion Date
Develop Test Case	Tester 、 Developer	
Test Case Review	Tester 、 Developer	

5.5.2 Test Cases Sequence

Test Cases Sequences
SQMS -TC01
SQMS -TC02

5.5.3 Termination Condition

- No defects are found
- Defects are fixed already

Section 6 Results and Analysis

6.1 User Identity Verification and Management Subsystem(UIVMS)

Test Case #	Results (PASS/FAIL)	Comment
UAMS-TC01	PASS	
UAMS-TC02	PASS	
RATE	100%	

6.2 Repository Management Subsystem(RMS)

Test Case #	Results (PASS/FAIL)	Comment
RMS -TC01	PASS	
RMS -TC02	PASS	
RATE	100%	

6.3 Database Management Subsystem(DBMS)

Test Case #	Results (PASS/FAIL)	Comment
DBMS -TC01	PASS	
DBMS -TC02	PASS	
RATE	100%	

6.4 Analysis Management Subsystem(AMS)

Test Case #	Results (PASS/FAIL)	Comment
AMS -TC01	PASS	
AMS -TC02	PASS	
RATE	100%	

6.5 SonarQube Management Subsystem(SQMS)

Test Case #	Results (PASS/FAIL)	Comment
SQMS -TC01	PASS	
SQMS -TC02	PASS	
RATE	100%	

Glossary

References

- [1] 洪肇奎, 郭耀煌, 李允中, Light-Weight Capability Maturity Model Integration (CMMI) for National Science Council (NSC) Open Source Project, CSIE Program, Department of Engineering and Applied Science, NSC, 2004.
- [2]