

Github專案分析系統(GRAS)

System Testing Document (STD)

Version: 1.1

Team #9

Name	ID	E-mail
王泓翔	108598076	t108598076@ntut.org.tw
林育德	109598073	t109598073@ntut.org.tw
諸政安	109598087	t109598087@ntut.org.tw
溫志嘉	109598037	t109598037@ntut.org.tw

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

12/28/2020

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.....	3
Section 3 Testing Environment	5
3.1 Operational Environment.....	5
3.2 Hardware Specification and Configuration.....	5
3.3 Software Specification and Configuration.....	6
3.4 Test Data Sources	6
3.5 Tools and Equipments.....	6
Section 4 Test Cases.....	7
4.1 Definition of Severity Number	7
4.2 GitHub Repository Analysis System (GRAS) Test Case.....	7
4.3 User Identity Verification and Management Subsystem (UIVMS) Test Case	9
4.4 Repository Management Subsystem (RMS) Test Case	12
4.5 Database Management Subsystem (DBMS) Test Case	17
Section 5 Test Procedure and Schedule	19
5.1 Testing Schedule	19
5.2 User Account and Group Management Subsystem (UIVMS).....	19
5.2.1 Schedule	19
5.2.2 Initialization Condition	20
5.2.3 Termination Condition	20
5.3 Repository Management Subsystem (RMS)	20
5.3.1 Schedule	20
5.3.2 Initialization Condition	20
5.3.3 Test Cases Sequences.....	21
5.3.4 Termination Condition	21
5.4 Database Management Subsystem (DBMS)	21
5.4.1 Schedule	21
5.4.2 Initialization Condition	21
5.4.3 Test Cases Sequences.....	22
5.4.4 Termination Condition	22
Section 6 Test Results and Analysis	23
6.1 GitHub Repository Analysis System (GRAS)	23

6.2	User Identity Verification and Management Subsystem (UIVMS)	23
6.3	(Repository Management Subsystem (RMS).....	23
6.4	Database Management Subsystem (DBMS)	23
Section 7 Glossary.....		25
Reference		26

版次變更記錄 (Change Log)

Revisions			
Version	Primary Author(s)	Description of Version	Date Completed
1.0	王泓翔、諸政安、 林育德、溫志嘉	STD文件規劃、大綱完成	2020/12/26
1.1	王泓翔、諸政安、 林育德、溫志嘉	STD文件初版完成，目錄調整	2020/12/28

Section 1 Objectives and Acceptance Criteria

1.1 Test Scope

Github專案分析系統(GitHub Repository Analysis System, GRAS) 提供團隊使用者一個可以查看及分析專案狀態的管理平台，透過將專案的各類活動指標進行視覺化，方便團隊間快速了解專案進度，並且保管這些歷程以供團隊後續回顧的功能。而本系統採用 Web Services 架構設計，並依功能取向分為三個子系統：

1. 身分驗證子系統(User Identity Verification and Management Subsystem, UIVMS)
2. 專案管理子系統(Repository Management Subsystem, RMS)
3. 資料庫管理子系統(Database Management Subsystem, DBMS)

本測試文件主要測試所有功能是否正常，各子系統之間的整合度是否正常，程式交付之系統功能是否與系統開發前的功能需求一致。在此我們著重於整合系統測試(Integration Test)及接受度測試(Acceptance Test)。本文件內容將依照系統需求規格書(SRS)與系統設計文件(SDD)中所描述的測試需求與使用案例進行相關計畫與測試。並希望透過此文件之描述與實踐，達到順利進行測試工作之目的。

1.2 Test Acceptance Criteria

本系統測試計畫需要滿足下列的測試接受準則：

- 所有本測試計畫的測試結果，皆需能符合預期測試結果才可通過 (Pass)
- 本系統需要對所有(Critical、Mirror)需求做完整測試
- 測試程序需要依照本測試計畫所制定的程序進行，所有的測試結果都需要符合預期測試結果(Expected Result)才能接受
- 本測試計畫中，Test Case 為通過測試的最基本單位，當該 Test Case 未通過測試時，則需要進行該單元的測試；其接受測試的準則與前一項相同

Section 2 Personnel and Responsibility

2.1 Personnel

本專案測試成員如下：

測試成員名單			
姓名	職位名稱	縮寫	備註
王泓翔	Project Manager	HX	專案管理者
林育德	Developer & Tester	YD	系統開發兼測試
諸政安	Developer & Tester	CA	系統開發兼測試
溫志嘉	Developer & Tester	ZJ	系統開發兼測試

2.2 Testing Schedule

- 時程

Item	Start Date	Over Date
系統接受度測試 (Acceptance Test)	2020/01/03	2020/01/07
各子系統驗證測試 (Subsystems Validation Test)	2020/11/25	2020/12/25
系統整合測試 (Integration Test)	2021/01/01	2021/01/03

- Iterations

Item	Start Date	Over Date
Prototyping	2020/11/03	2020/11/20
Increment 1	2020/11/21	2020/12/13
Increment 2	2020/12/14	2021/01/01

- 查核點

Item	Start Date	Over Date
各子系統驗證測試	2020/11/10	2020/11/22
系統接受度測試	2020/01/03	2020/01/07
系統整合測試	2021/01/01	2021/01/03

2.3 Responsibility

- GRAS 主系統

Testing Activity	Personnel	測試時程
GRAS-TC01	HX 、 YD	2 days
GRAS -TC02	CA 、 ZJ	2 days
GRAS -TC03	HX 、 CA	2 days

- UIVMS 子系統

Testing Activity	Personnel	測試時程
UGMS-TC01	ZJ 、 YD	2 days
UGMS-TC02	CA 、 YD	2 days
UGMS-TC03	HX 、 YD	2 days
UGMS-TC04	ZJ 、 CA	2 days

- RMS 子系統

Testing Activity	Personnel	測試時程
RMS-TC01	CA 、 ZJ	2 days
RMS-TC02	ZJ 、 HX	2 days
RMS-TC03	YD 、 CA	3 days
RMS-TC04	HX 、 ZJ	2 days
RMS-TC05	CA 、 YD	2 days
RMS-TC06	ZJ 、 YD	2 days
RMS-TC07	YD 、 CA	3 days
RMS-TC08	ZJ 、 YD	2 days
RMS-TC09	YD 、 CA	1 days

- DBMS 子系統

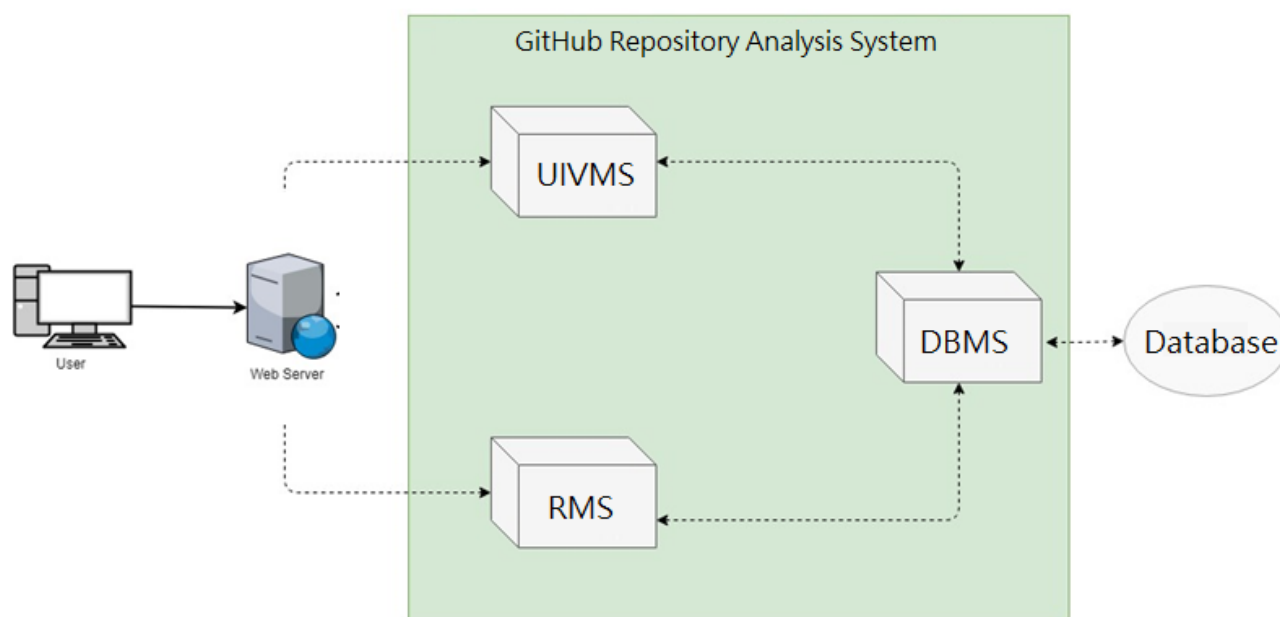
Testing Activity	Personnel	測試時程
DBMS-TC01	CA、YD	2 days
DBMS-TC02	ZJ、HX	2 days
DBMS-TC03	YD、CA	3 days
DBMS-TC04	ZJ、HX	2 days
DBMS-TC05	CA、YD	2 days

Section 3 Testing Environment

3.1 Operational Environment

本測試環境於 Windows Server 的網路環境下進行，主機與工作站間需要先設定網路連線，進行測試前需要進行下面環境的設定：

1. 主機端採用 Tomcat 作為網頁伺服器，因此需要先進行 Tomcat 的設定。
2. 資料庫伺服器的資料庫存取採 JDBC 的方式，因此需要先設定主機端與工作站端的 JDBC。
3. 各個子系統的內部單元測試為單機模式由開發者在開發平台環境做測試工作。



GRAS架構圖

3.2 Hardware Specification and Configuration

項目	名稱	數量	規格	備註
1	Intel PC	4	3.2 GHz (8 CPUs) 8G RAM	
2	Network	1	Ethernet Network	

3.3 Software Specification and Configuration

項目	名稱	數量	版本	備註
1	Microsoft Windows	5	Win10	
2	MySQL	1	5.7	Web Service端
3	JDK	5	1.8	
4	IntelliJ IDEA	5	11.0.7	Web Service 端
5	Chrome	5	71.0.3578.98	Clinet 端
6	Eclipse		Jee-galileo	測試
7	Angular	1	11.0.5	Web Service端

3.4 Test Data Sources

- Testing UIVMS System Data
- Testing RMS Subsystem Data
- Testing DBMS Subsystem Data

3.5 Tools and Equipments

- 單元測試：本系統內部測試採用 IntelliJ IDEA + JUnit 測試，測試程式為開發人員在撰寫每項功能後測試，待確認無問題後再繼續開發。
- 系統測試：系統測試計劃的執行採用使用者直接操作的測試方式，測試人員再操作進行時需使用 Microsoft Word 2016 做資料整理與資料蒐集分析，而在 Server 端 Eclipse 會記錄任何操作過程式的例外、錯誤訊息。
- 書面資料：測試期間的所有資料整合是採用 Microsoft Word 2016。

Section 4 Test Cases

4.1 Definition of Severity Number

Severity Number	Definition
1	Critical
2	Mirror

4.2 GitHub Repository Analysis System (GRAS) Test Case

- Test Objective

本測試案例是測試GRMS子系統與子系統整合的能力，相互運作是否正常。

- Test Cases

Identification	GRMS-TC01
Name	提供帳戶管理及權限辨識功能測試
Tested target	GitHub Repository Analysis System (PASS)
Reference	
Severity	1 (Critical)
Instrucitons	1. 選擇「Login」 2. 使用者於「Login」畫面輸入正確的帳號及密碼 3. 按下「Login」 4. 選擇「帳號資訊」
Expected Result	1. 成功進入「Login」畫面 2. 「Login」的「帳號」欄位輸入時顯示使用者帳號 「Password」欄位只顯示「*」 3. 顯示登入成功，並連結到「Homepage」 4. 成功進入「帳號資訊」 5. 提供使用者修改個人資訊的頁面功能
Cleanup	選擇「Logout」登出

Identification	GRAS-TC02
Name	提供新增 Project 功能測試
Tested target	GitHub Repository Analysis System (PASS)
Reference	
Severity	1 (Critical)
Instrucitons	<ol style="list-style-type: none"> 1. 選擇「Login」 2. 使用者於「Login」畫面輸入正確的帳號及密碼 3. 按下「Login」 4. 選擇「Create Project」 5. 輸入專案名稱、專案敘述、欲導入網址。 6. 按下 Create Project 按鈕 7. 提示新增 Project 成功，跳轉至專案畫面。
Expected Result	<ol style="list-style-type: none"> 1. 成功進入「Login」畫面 2. 「Login」的「帳號」欄位輸入時顯示使用者帳號 「Password」欄位只顯示「*」 3. 顯示登入成功，並連結到「首頁」 4. 成功進入「GRAS」 5. 提供使用者新增不同專案功能
Cleanup	選擇「Logout」登出

Identification	GRAS -TC03
Name	提供 Analysis 功能測試
Tested target	GitHub Repository Analysis System (PASS)
Reference	
Severity	1 (Critical)
Instrucitons	<ol style="list-style-type: none"> 1. 選擇「Login」 2. 使用者於「Login」畫面輸入正確的帳號及密碼 3. 按下「Login」，進行登入 4. 選擇「My Project」，進入專案總覽畫面 5. 點選想分析的 Project，按下 Analysis 6. 系統開始進行分析，產出對應分析結果頁面(Code base/Commit trend/Commit log/Issue list 等
Expected Result	<ol style="list-style-type: none"> 1. 成功進入「Login」畫面 2. 「Login」的「帳號」欄位輸入時顯示使用者帳號 「Password」欄位只顯示「*」 3. 顯示登入成功，並連結到「首頁」，成功進入「GRAS」 4. 顯示已導入的專案，若無專案則顯示空頁面，並提示是否新增專案 5. 所點選專案成功執行「Analysis」 6. 提供使用者點選分析頁面功能
Cleanup	選擇「Logout」登出

4.3 User Identity Verification and Management Subsystem (UIVMS) Test Case

● Test Objective

本測試案例是測試UIVMS 子系統最主要的三項功能：

- 系統是否能正常驗證使用者登入
- 系統是否能正常的建立、編輯與刪除使用者帳號
- 系統是否能限制角色權限

● Test Cases

Identification	UIVMS-TC01
Name	驗證使用者登入測試
Tested target	User Identity Verification and Management Subsystem(UIVMS)
Reference	
Severity	1 (Critical)
Instrucitons	1. 選擇「Login」 2. 使用者於「Login」畫面輸入正確的帳號及密碼 3. 按下「Login」
Expected Result	1. 成功進入「Login」畫面 2. 「Login」的「帳號」欄位輸入時顯示使用者帳號 「Password」欄位只顯示「*」 3. 顯示登入成功，並連結到「首頁」
Cleanup	選擇「Logout」登出

Identification	UIVMS-TC02
Name	帳號建立測試
Tested target	User Identity Verification and Management Subsystem(UIVMS)
Reference	
Severity	1 (Critical)
Instrucitons	1. 進入「signup」。 2. 輸入使用者姓名。 3. 輸入使用者帳號。 4. 輸入使用者密碼及密碼確認。 5. 點選註冊。
Expected Result	1. 顯示註冊成功。 2. 使用新帳號能夠成功進行登入。 3. 登入後使用者資訊、權限與預期相符合。
Cleanup	刪除帳號

Identification	UIVMS-TC03
Name	帳號編輯測試
Tested target	User Identity Verification and Management Subsystem(UIVMS)
Reference	
Severity	1 (Critical)
Precondition	使用者已經登入並在個人資訊頁面
Instrucitons	1.選擇編輯帳號功能。 2.選擇欲編輯的項目: 2-1.編輯姓名 2-2.編輯帳號 2-3.編輯密碼 3.儲存編輯項目。
Expected Result	1.若變更帳號密碼，登出後以新帳號密碼登入必須成功。 2.若變更姓名，個人資訊必須顯示變更後的姓名。
Cleanup	將編輯的項目回復原樣。

Identification	UIVMS-TC04
Name	個人專案權限管理測試
Tested target	User Identity Verification and Management Subsystem(UIVMS)
Reference	
Severity	1 (Critical)
Precondition	使用者已經登入
Instrucitons	1. 新增一個 observer 權限。 2. 使用此 observer 權限查看 project。
Expected Result	1. observer 無法進行刪除。 2. observer 無法進行新增。
Cleanup	None

4.4 Repository Management Subsystem (RMS) Test Case

● Test Objective

本測試案例是測試 RMS 子系統最主要的三項功能：

- 是否能正常的新增、修改、刪除、查詢專案資料內容。
- 是否能正常的新增、刪除、查詢專案中的Repo。
- 系統是否能正常的分析專案各項資訊。

● Test Cases

Identification	RMS -TC01
Name	新增專案測試
Tested target	Repository Management Subsystem(RMS)
Reference	
Severity	1 (Critical)
Precondition	使用者必須擁有可登入此系統的帳號、密碼並且已登入
Instrucitons	1. 選擇「My Project」，進入專案總覽畫面 2. 於專案列表下方選擇「Add New Project」 3. 輸入專案名稱、專案敘述及欲加入的 Repository 專案網址 4. 按下「Create Project」
Expected Result	1. 資料庫中 Project Table 中成功新增一筆專案資料 2. 進入「My Project」頁面，顯示所有的專案資料，其中包含新增的專案
Cleanup	刪除新增的專案

Identification	RMS -TC02
Name	修改專案測試
Tested target	Repository Management Subsystem(RMS)
Reference	
Severity	1 (Critical)
Precondition	使用者必須擁有可登入此系統的帳號、密碼並且已登入
Instrucitons	1. 建立一筆測試專案 2. 於首頁選擇「My Project」，進入專案總覽畫面 3. 在專案列表下方選擇「Edit Project」 4. 輸入欲修改的專案名稱、專案敘述 5. 按下「Comfirm」
Expected Result	1. 資料庫中 Project Table 成功新增一筆測試專案 2. 進入「My Project」頁面，顯示系統內所有專案，其中包含新增的測試專案 3. 資料庫 Project Table 中所選擇的測試專案資料成功更新為所輸入的專案資料 4. 進入「My Project」頁面，並顯示更新過的測試專案資料
Cleanup	刪除測試的專案

Identification	RMS -TC03
Name	刪除專案測試
Tested target	Repository Management Subsystem(RMS)
Reference	
Severity	1 (Critical)
Precondition	使用者必須擁有可登入此系統的帳號、密碼並且已登入
Instrucitons	<ol style="list-style-type: none"> 1. 建立一筆測試專案 2. 於首頁選擇「My Project」，進入專案總覽畫面 3. 在專案列表下方選擇「Delete Project」 4. 按下「Comfirm」
Expected Result	<ol style="list-style-type: none"> 1. 資料庫中 Project Table 與 Repository Table 中成功新增一筆專案與 Repo 2. 進入「My Project」頁面，顯示系統內所有專案，其中包含新增的測試專案 3. 資料庫 Project Table 中所選擇的測試專案成功刪除 4. 資料庫 Repository Table 中測試專案所包含的 Repo 成功刪除 5. 頁面成功跳回「My Project」頁面，並顯示更新過的專案列表，測試專案已消失

Identification	RMS -TC04
Name	新增 Repo 測試
Tested target	Repository Management Subsystem(RMS)
Reference	
Severity	1 (Critical)
Precondition	使用者必須擁有可登入此系統的帳號、密碼並且已登入
Instrucitons	<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「Add Repo」 5. 填入欲新增的 Repository 網址 6. 系統確認為有效網址 7. 選擇「Comfirm」
Expected Result	<ol style="list-style-type: none"> 1. 資料庫中 Project Table 成功新增一筆測試專案 2. 進入「My Project」頁面，顯示系統內所有專案，其中包含新增的測試專案 3. 進入專案頁面，顯示測試專案的 Repo 列表 4. 新增 Repo 後，資料庫 Repository Table 中成功新增一筆測試 Repo 8. 頁面成功跳回專案頁面，並顯示所有的 Repo 列表，其中包含使用者所新增的 Repo
Cleanup	刪除測試專案與測試 Repo

Identification	RMS -TC05
Name	刪除 Repo 測試
Tested target	Repository Management Subsystem(RMS)
Reference	
Severity	1 (Critical)
Precondition	使用者必須擁有可登入此系統的帳號、密碼並且已登入
Instrucitons	<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「Delete Repo」 5. 勾選欲刪除的 Repo 6. 選擇「Comfirm」
Expected Result	<ol style="list-style-type: none"> 1. 資料庫中 Project Table 成功新增一筆測試專案 2. 進入「My Project」頁面，顯示系統內所有專案，其中包含新增的測試專案 3. 進入專案頁面，顯示測試專案的 Repo 列表 4. 刪除 Repo 後，資料庫 Repository Table 中所選擇的 Repo 成功刪除 5. 頁面成功跳回專案頁面，並顯示所有的 Repo 列表，其中所刪除的 Repo 已消失

Identification	RMS -TC06
Name	查看 Commit 變化趨勢測試
Tested target	Repository Management Subsystem(RMS)
Reference	
Severity	1 (Critical)
Precondition	使用者必須擁有可登入此系統的帳號、密碼並且已登入，並且專案列表已有專案存在，並且專案中已有 Repository 存在
Instrucitons	<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. 選擇欲分析的 Repo 5. 選擇「Commit Trend」
Expected Result	1. 系統進入「Commit Trend」頁面，並可查看分析的 Repo 整體以及其中每一個成員的 commit 時間、commit 數和 commit 行數

Identification	RMS -TC07
Name	查看 Code Base 變化趨勢測試
Tested target	Repository Management Subsystem(RMS)
Reference	
Severity	1 (Critical)
Precondition	使用者必須擁有可登入此系統的帳號、密碼並且已登入，並且專案列表已有專案存在，並且專案中已有 Repository 存在
Instrucitons	<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. 選擇欲分析的 Repo 5. 選擇「Code Base」
Expected Result	1. 系統進入「Code Base」頁面，並可查看分析的 Repo 整體以及其中每一個成員的 code base 變化趨勢

Identification	RMS -TC08
Name	查看 Issue 測試
Tested target	Repository Management Subsystem(RMS)
Reference	
Severity	1 (Critical)
Precondition	使用者必須擁有可登入此系統的帳號、密碼並且已登入，並且專案列表已有專案存在，並且專案中已有 Repository 存在
Instrucitons	<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. 選擇欲分析的 Repo 5. 選擇「Issue Track」
Expected Result	1. 系統進入「Issue Track」頁面，並可查看分析的 Repo 之 issue 的 Label、計算 Lead time 以及其他細節

Identification	RMS -TC09
Name	查看 Commit 資訊測試
Tested target	Repository Management Subsystem(RMS)
Reference	
Severity	1 (Critical)
Precondition	使用者必須擁有可登入此系統的帳號、密碼並且已登入，並且專案列表已有專案存在，並且專案中已有 Repository 存在
Instrucitons	<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. 選擇欲分析的 Repo 5. 選擇「Commit Log」
Expected Result	1. 系統進入「Commit Log」頁面，並可查看分析的 Repo 之 commit 細節

4.5 Database Management Subsystem (DBMS) Test Case

● Test Objective

本測試案例是測試DBMS 子系統最主要的功能：

- 是否能正常的寫入註冊使用者的資訊
- 是否能正常的紀錄使用者登入資訊(時間)
- 是否能正常的新建專案
- 是否能正常的刪除專案
- 是否能正常的導入Github Repo內的資訊

● Test Cases

Identification	DBMS -TC01
Name	寫入註冊使用者資訊測試
Tested target	Database Management Subsystem(DBMS)
Reference	
Severity	1 (Critical)
Instrucitons	1. 點選 GRAS 網站的註冊頁面 2. 輸入對應資訊 3. 點選註冊按鈕 4. 顯示註冊結果 5. 查看資料庫使用者欄位
Expected Result	1. 跳轉至註冊頁面 2. 提示使用者欄位內容 3. 若帳號重複則顯示此帳號名稱已被使用 4. 全新帳號則顯示註冊成功，重複帳號則清空請使用者再註冊一次 5. 成功新增使用者

Identification	DBMS -TC02
Name	紀錄使用者登入資訊測試
Tested target	Database Management Subsystem(DBMS)
Reference	
Severity	1 (Critical)
Instrucitons	1. 點選 GRAS 網站的登入頁面 2. 輸入對應帳號密碼 3. 點選登入按鈕 4. 顯示登入結果 5. 查看資料庫使用者欄位
Expected Result	1. 跳轉至登入頁面 2. 提示使用者欄位內容 3. 轉送資料至後端 4. 若資料庫內帳號名稱不存在則前端頁面顯示此帳號不存在 若資料庫內密碼輸入錯誤則前端頁面顯示密碼輸入錯誤

	正確無誤則跳轉 GRAS 專案首頁 5. 資料庫內新增使用者登入時間
--	---------------------------------------

Identification	DBMS -TC03
Name	紀錄新建專案資訊測試
Tested target	Database Management Subsystem(DBMS)
Reference	
Severity	1 (Critical)
Instrucitons	1. 點選 GRAS 網站的 Create Project 頁面 2. 輸入對應專案資訊 3. 點選 Create Project 按鈕 4. 顯示結果 5. 查看資料庫 Project 欄位
Expected Result	1. 跳轉至 CreateProject 頁面 2. 提示使用者欄位內容 3. 若導入網址為空則提示請加入 Github 網址 非空則轉送資料至後端 4. 專案建立成功 5. 資料庫內新增專案名稱/專案敘述/導入 Github 專案名稱

Identification	DBMS -TC04
Name	紀錄刪除專案測試
Tested target	Database Management Subsystem(DBMS)
Reference	
Severity	1 (Critical)
Instrucitons	1. 點選 GRAS 網站的 My Project 頁面 2. 點選對應專案 3. 點選 delete Project 按鈕 4. 進行刪除驗證 5. 顯示結果 6. 查看資料庫 Project 欄位
Expected Result	1. 跳轉至 My Project 頁面 2. 顯示專案的可使用功能(分析/刪除) 3. 轉送 delete post 到後端 4. 請使用者再輸入一次專案名稱，以確保使用者誤點 5. 顯示刪除成功 6. 資料庫內對應專案名稱/專案敘述/導入 Github 專案名稱已刪除

Identification	DBMS -TC05
Name	紀錄導入 Github Repo 測試
Tested target	Database Management Subsystem(DBMS)
Reference	
Severity	1 (Critical)
Instrucitons	1. 點選 GRAS 網站的 Create Project 頁面 2. 輸入對應專案資訊以及欲導入的專案網址 3. 點選 Add，進行專案導入 4. 顯示結果 5. 查看資料庫 Project 下的 Repo 欄位
Expected Result	1. 跳轉至 CreateProject 頁面 2. 提示使用者欄位內容 3. 若導入網址為空則提示請加入 Github 網址 非空則轉送資料至後端驗證，若網址為空則顯示網址錯誤 Repo 網址成功則開始獲取對應資訊 4. Github repo 導入成功 5. 資料庫內新增導入 Github 專案名稱

Section 5 Test Procedure and Schedule

5.1 Testing Schedule

Deliverable	Responsibility	Completion Date
Develop Test Cases	Tester	2020/12/20
Test Cases Review	Developer、Tester、Project Manager	2020/12/21
Requirements Validation Matrix	Project Manager	2020/12/21
Execute manual	Tester、Project Manager	2020/12/30
Complete Defect Reports	Everyone testing the product	2020/12/30
Communicate Beta test status	Project Manager	2020/01/05

5.2 User Account and Group Management Subsystem (UIVMS)

5.2.1 Schedule

Deliverable	Responsibility	Completion Date
Develop Test Cases	Tester	2020/12/26
Test Cases Review	Developer、Tester、Project Manager	2020/12/27
Requirements Validation	Project Manager	2020/12/27

Matrix		
Execute manual	Tester 、 Project Manager	2020/12/30
Test Case UIVMS-TC01~Test Case UIVMS-TC04	Tester 、 Project Manager	2020/01/03

5.2.2 Initialization Condition

- UIVMS is validated
- Complete the establishment of a database interface

Test Cases Sequences

Test Cases Sequences
UIVMS-TC01
UIVMS-TC02
UIVMS-TC03
UIVMS-TC04

5.2.3 Termination Condition

- No defects are found
- Def

5.3 Repository Management Subsystem (RMS)

5.3.1 Schedule

Deliverable	Responsibility	Completion Date
Develop Test Cases	Tester	2020/12/20
Test Cases Review	Developer 、 Tester 、 Project Manager	2020/12/21
Requirements Validation Matrix	Project Manager	2020/12/21
Execute manual	Tester 、 Project Manager	2020/12/23
Test Case RMS-TC01~Test Case RMS-TC09	Tester 、 Project Manager	2020/12/25

5.3.2 Initialization Condition

- UIVMS Account Verification is complete
- RMS is validated
- Complete the establishment of a database interface

5.3.3 Test Cases Sequences

Test Cases Sequences
RMS-TC01
RMS-TC02
RMS-TC03
RMS-TC04
RMS-TC05
RMS-TC06
RMS-TC07
RMS-TC08
RMS-TC09

5.3.4 Termination Condition

- No defects are found
- Defe

■

5.4 Database Management Subsystem (DBMS)

5.4.1 Schedule

Deliverable	Responsibility	Completion Date
Develop Test Cases	Tester	2020/12/20
Test Cases Review	Developer 、 Tester 、 Project Manager	2020/12/21
Requirements Validation Matrix	Project Manager	2020/12/21
Execute manual	Tester 、 Project Manager	2020/12/23
Test Case DBMS-TC01~Test Case DBMS-TC05	Tester 、 Project Manager	2020/12/25

5.4.2 Initialization Condition

- Complete the establishment of a database interface

5.4.3 Test Cases Sequences

Test Cases Sequences
DBMS-TC01
DBMS-TC02
DBMS-TC03
DBMS-TC04
DBMS-TC05

5.4.4 Termination Condition

- No defects are found
- Defe

Section 6 Test Results and Analysis

6.1 GitHub Repository Analysis System (GRAS)

Test Case #	Results (PASS/FAIL)	Comment
GRAS-TC01	PASS	
GRAS-TC02	PASS	
GRAS-TC03`	PASS	
RATE	100%	

6.2 User Identity Verification and Management Subsystem (UIVMS)

Test Case #	Results (PASS/FAIL)	Comment
UIVMS -TC01	PASS	
UIVMS -TC02	PASS	
UIVMS -TC03	PASS	
UIVMS -TC04	PASS	
RATE	100%	

6.3 (Repository Management Subsystem (RMS))

Test Case #	Results (PASS/FAIL)	Comment
RMS-TC01	PASS	
RMS-TC02	PASS	
RMS-TC03	PASS	
RMS-TC04	PASS	
RMS-TC05	PASS	
RMS-TC06	PASS	
RMS-TC07	PASS	
RMS-TC08	PASS	
RMS-TC09	PASS	
RATE	100.0%	

6.4 Database Management Subsystem (DBMS)

Test Case #	Results (PASS/FAIL)	Comment
DBMS -TC01	PASS	
DBMS -TC02	PASS	
DBMS -TC03	PASS	
DBMS -TC04	PASS	
DBMS -TC05	PASS	
RATE	100.0%	

Section 7 Glossary

Database	記錄資料的地方，提供新增、刪除、更新記錄的功能。
HTML	超文件標示語言（英語：HyperText Markup Language，簡稱：HTML）是一種用於建立網頁的標準標示語言。HTML 是一種基礎技術，常與 CSS、JavaScript 一起被眾多網站用於設計令人賞心悅目的網頁、網頁應用程式以及行動應用程式的使用者介面。
JSP	是由 Sun Microsystems 公司主導建立的一種動態網頁技術標準。JSP 部署於網路伺服器上，可以回應用戶端傳送的請求，並根據請求內容動態地生成 HTML、XML 或其他格式文件的 Web 網頁，然後返回給請求者。JSP 技術以 Java 語言作為手稿語言，為用戶的 HTTP 請求提供服務，並能與伺服器上的其它 Java 程式共同處理複雜的業務需求。
JavaScript	是一種進階的，解釋執行的程式語言。JavaScript 是一門基於原型、函式先行的語言，是一門多範式的語言，它支援物件導向編程，指令式程式設計，以及函數語言程式設計。
jQuery	jQuery 是一款免費且開放原始碼的 JavaScript 代碼函式庫，由 John Resig 創建。授權協定為 GPL 和 MIT 許可證雙協定。jQuery 1.3 版中，引入全新的 CSS 選擇器引擎 Sizzle。目前已有 Google AJAX Libraries API 等公開站台提供 JQuery 的 js 的參照服務。
MySQL	MySQL 是一個開放源碼的關聯式資料庫管理系統，由於性能高、成本低、可靠性好，已經成為最流行的開源資料庫，被廣泛地應用在 Internet 上的中小型網站中。隨著 MySQL 的不斷成熟，它也逐漸用於更多大規模網站和應用。
SQL	Structured Query Language，關連式查詢語言，用來定義資料庫的結構，或是利用 SQL 對資料庫執行一些查詢、增加、刪除、更新記錄，到目前為止，SQL 是第一個，也是唯一的標準資料庫語言，受到廣泛的接受。
SDK	軟體發展套件（Software Development Kit）一般是一些被軟體工程師用於為特定的軟體包、軟體框架、硬體平台、作業系統等建立應用軟體的開發工具的集合。

Reference