科技部

111年度大專學生研究計畫申請書

一、綜合資料：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 申 請 人 ︻ 學 生 ︼ | 姓 名 |  | | 身分證  號　碼 |  |
| 就 讀 學 校、  科 系 及 年 級 |  | | 電　話 |  |
| 學 生 研 究  計 畫 名 稱 |  | | | |
| 研 究 期 間 | 自111年7月1日至112年2月底止，計8個月 | | | |
| 計 畫 歸 屬 司 別 | □自然司　　 　□工程司 　□生科司  □人文司(含科學教育領域) | | | |
| 研究學門代碼及名稱 |  | | | |
| 上年度曾執行本部大專學生研究計畫 | □是（計畫編號：M0ST － － － － ）  □否 | | | |
| 指 導 教 授 | 姓 名 |  | | 身分證 號　碼 |  |
| 服 務 機 構 及 科 系(所) |  | | | |
| 職 稱 |  | | 電　話 |  |
| 補 助 經 費 | 項 目 | 金 額 | 說 明 | | |
| 研 究 助 學 金 | 48,000元 | 每月補助研究助學金新臺幣6,000元  ，8個月計新臺幣48,000元。 | | |
| 耗材、物品、圖書及雜項費用 | 元 | 依研究計畫實際需求擇優補助，每一計畫最高以補助新臺幣20,000元為限。 | | |
| 合 計 | 元 |  | | |

表C801

科、系主管姓名：

（學生就讀學校）

二、研究計畫內容（以10頁為限）：

題目：基於DevOps之程式作業品質測試分析系統

1. 摘要

大多數的程式批改軟體，像是瘋狂程設、online judge等等，他們批改的要求只有程式執行的結果是否正確。沒有辦法針對程式的撰寫品質進行檢查，此情況容易造成學生只在乎結果是否正確，不在乎程式碼撰寫的品質。

因此我們希望能開發一個具有程式自動批改功能的教學系統，系統的功能除了擁有一般批改軟體有的單元測試來判斷答案對錯之外，會再加入檢測程式品質的功能，並且在檢測品質的同時提供使用者改善的建議，藉此讓使用者能根據系統給予的提示，提升程式碼撰寫的品質。

關於程式品質的考量點，我們找了很多相關論文以及文章，最終決定採用CISQ(Consortium for Information & Software Quality)組織對於程式品質(Code Quality)的規範當中的安全性(Security)、可靠性(Reliability)以及可維護性(Maintainability)這三項指標，程式品質檢測的重點也會放在這裡。

為了能夠更方便、直觀地了解學生的學習狀況，我們會將檢測結果統整到儀表板上顯示出來，儀表版會擁有題目分數、程式碼品質以及學生的學習曲線圖等資訊，我們考量到現實狀況下，一個學生和老師可能會有多堂課程，因此，我們加入了多堂課程的功能，登入系統後，老師可以開設多堂課程，而學生也可以選擇多堂課程，並能夠切換課程來繳交作業。

1. 研究動機與研究問題

上大一時，在程式設計這堂課程中，使用了系上老師自己開發的線上自動評分系統，這套系統只能正確的判斷程式執行的結果，當時的我們也就認為程式只要結果對就好。到了大二，在軟體工程這堂課中，我們了解到了軟體程式品質的重要性，如果只在乎結果，程式中可能存在許多不必要的變數或結構，而不好的變數的宣告方式及糟糕的函式寫法都有可能產生安全性問題、影響可讀性。

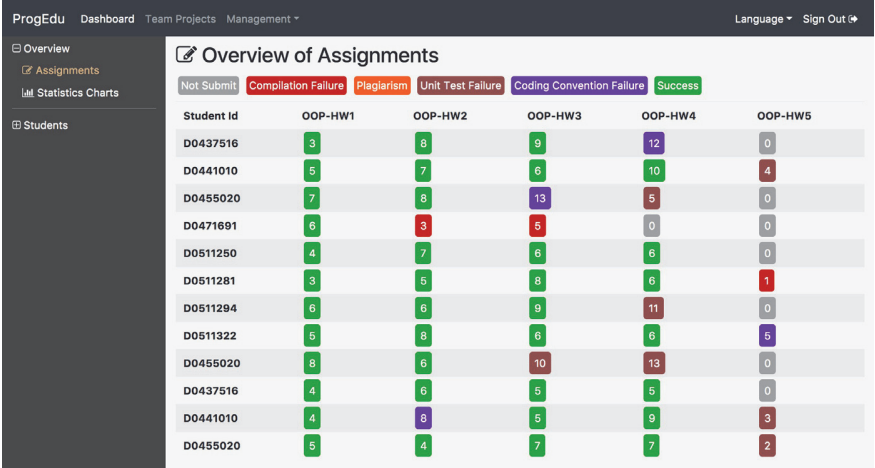
藉由這次專題的契機，我們嘗試開發出一套能判斷程式執行結果正確性，且能夠分析程式品質的教學系統。對於這套系統，我們將會開發重點放在如何檢測程式品質以及如何提升使用者程式碼品質上。檢測品質的部分，我們會使用SonarQube來檢測上面提到的三項指標：安全性、可靠性以及可維護性。提升程式碼品質的部分，在檢測完後，系統會將可改進的部分標記起來，並提供範例給使用者參考，此外，系統也會將每次結果紀錄下來，生成學習曲線圖，以便了解學習狀況。

1. 文獻回顧與探討

在研究如何開發這套系統時，我們參考了現有的程式批改系統ProgEdu，我們參考了它們的系統架設方式，使用Git作為學生繳交程式碼的工具，並且加入持續整合(CI)軟體Jenkins，讓我們能即時構建並檢測學生繳交的程式碼。關於程式碼檢測的部份，該系統只檢查程式碼有無正確縮排及註解，我們希望能夠在這些項目上再加入程式碼品質檢測，品質檢測的部份我們參考了CISQ - ISO/IEC 5055這個標準，針對程式的安全性、可靠性以及可維護性進行評量。

3-1.參考文獻回顧

1. ProgEdu：[1]是一套能自動檢查、編譯、單元測試的系統，並提供Checkstyle、Unit test等程式碼的檢驗結果，並將結果呈現在儀表板(圖一)供老師和學生檢視。



(圖一)ProgEdu將各個學生的各個作業之狀態顯示在儀表板

1. CISQ - ISO/IEC 5055:[3]根據四個關鍵因素來衡量軟體產品的內部結構： 安全性、可靠性、性能效率和可維護性。這些是決定軟體系統的可信度、

可靠性和彈性的因素。在ISO 5055之前，沒有國際標準透過分析其內部結構來衡量軟體系統的品質和完整性。這相當於只透過房屋的外觀來評估房屋，而沒有檢查其內部結構是否有木材腐爛。ISO 5055 提供了在開發中產品的事前檢查措施，以在發生問題之前識別和消除結構缺陷。

3-2.使用工具

1. 系統前端：預計從當今較為普遍的三種JavaScript框架Vue、React和Angular中選擇其一來建立前端使用者介面。

Vue：是一套以視圖層為基礎發展的 JavaScript 漸進式框架。Vue的目標是透過簡單的 API 提供開發者實作資料綁定與操作網頁上的元件。

React：是一個高效且具有彈性的 JavaScript 函式庫，用以建立使用者介 面。它讓你使用小巧而獨立的「component」，來建立複雜的 UI。

Angular：是一個基於TypeScript的開發平台。它將核心功能和可選功能作為一組 TypeScript 函式庫進行實現。

1. GitLab：利用Git作為版本控制系統，除了讓學生方便繳交作業，還能讓學生熟悉系統開發的流程。GitLab 是DevOps 平臺，它使組織能夠通過更快、更高效的交付軟體，同時增強安全性和合規性，最大限度地提高軟體開發的整體回報。
2. 系統後端：我們使用RESTful API來連接我們的前端和資料庫端。使用RESTful API能讓我們的API接口簡單且統一，且能讓後端API開發上好維護且有彈性。
3. 系統資料庫：我們將根據之後所決定的資料儲存方式來決定使用

RDBMS資料庫(ex:MySQL)或是NoSQL資料庫(ex:MongoDB)。MongoDB：是一個文件導向 (Document-oriented database)的資料庫管理系統的資料庫。

MySQL：是一種關聯數據庫管理系統，關聯數據庫將數據保存在不同的表中可有效增加速度並提高靈活性。

1. 程式品質檢測平台：我們使用SonarQube來協助我們對學生的程式碼

做品質檢測。SonarQube提供程式碼品質、風險檢測解決方案，目前提供了超過27種以上程式語言的開發分析。

1. 系統整合工具：我們會在系統中建置Jenkins來協助學生能快速建置

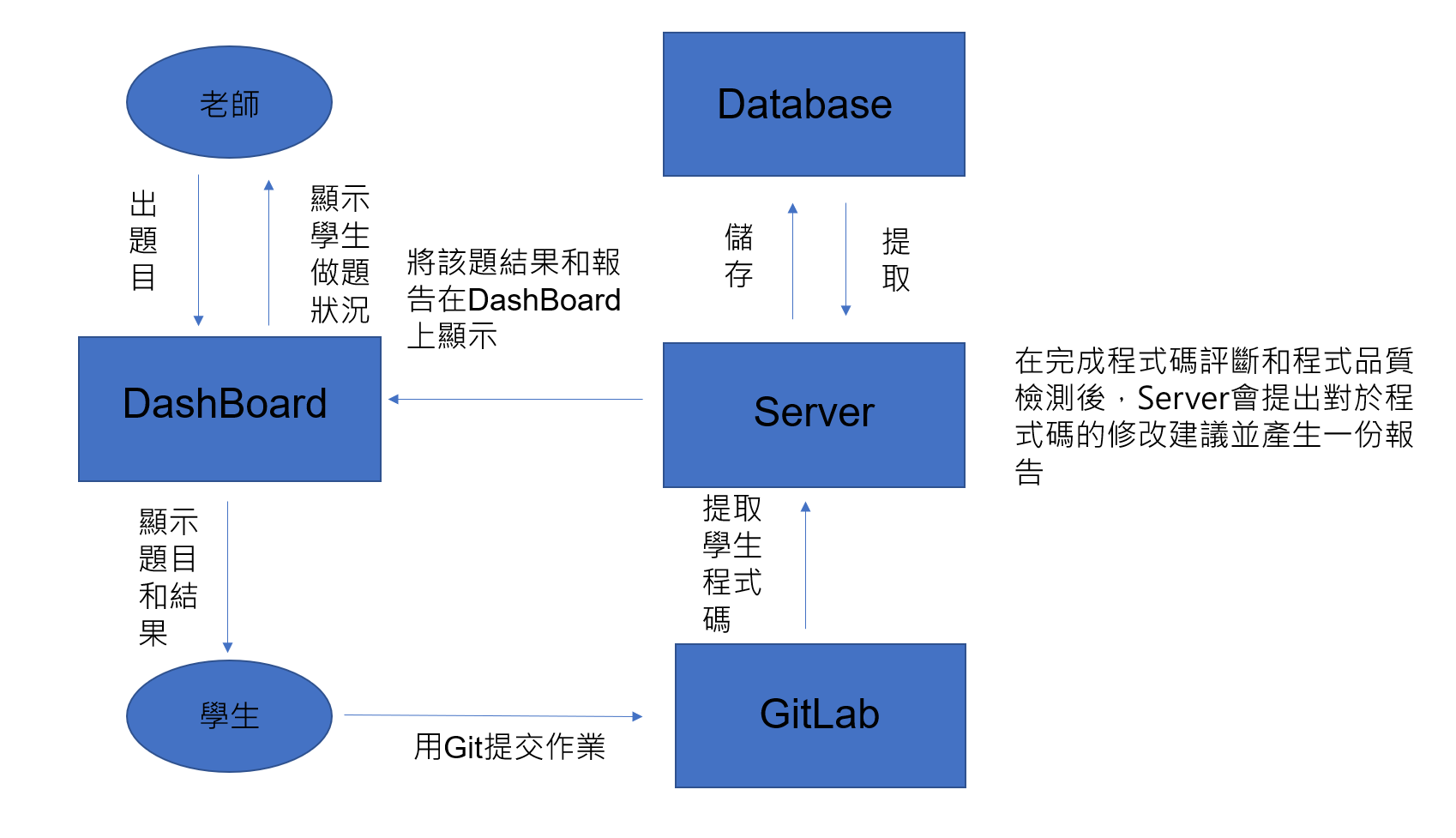
作業。Jenkins提供了軟體開發的持續整合服務，除此之外也提供相當多插件來支援不同的專案開發。

1. 系統整合容器：我們的系統將透過Docker來建置，以方便管理。

Docker是一個用於開發、發佈和運行應用程式的開放平臺。擁有輕量化、提高資源效率、更快建置等優勢。

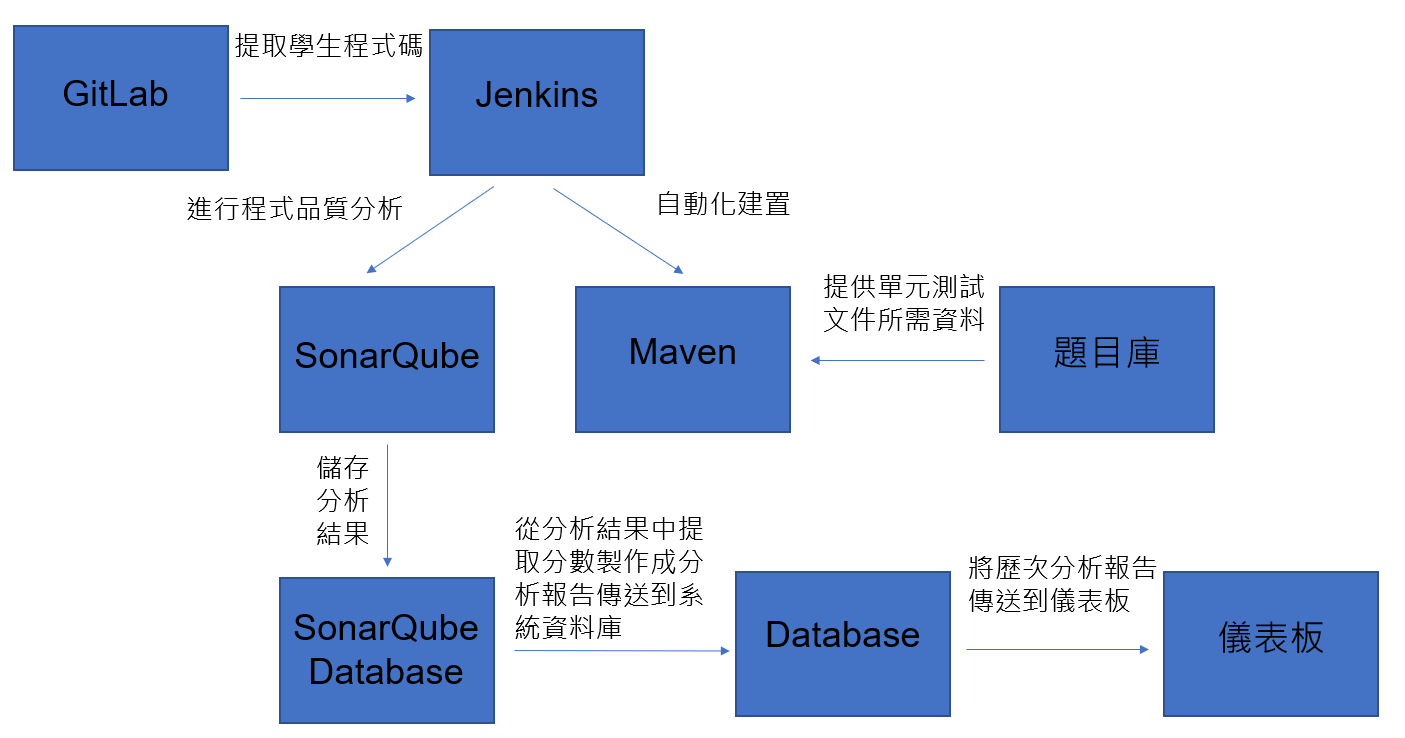
1. 研究方法及步驟

以下兩張圖分別代表我們預期系統做出來後的基本系統流程和Server內部資訊流。而為了達到我們最終的目標，我們將系統架設分成了三步驟，第一步是先架設基本的Server，第二步是在Server中加入單元測試及程式品質檢測功能，最後會是前端以及題目庫的架設。



(圖二)基本系統流程

目前計畫讓學生從儀表板上查看題目，完成作業後統一將撰寫的程式碼以Git提交到GitLab上，而當有檔案被上傳到GitLab時，系統會自動建置並進行單元測試，也會同時使用SonarQube檢測程式碼的品質，並將結果以PDF的格式彙整成一份報告呈現在儀表板，而老師則可以在儀表板上出題、觀看學生的做題狀況。在出題方面，會採用SCORM的概念，老師可以從網路上找題庫或是自己出題，只要符合格式，就可以出題。



(圖三)Server內部資料流

Server內部的每個部件都會放在各自Docker中，再利用Docker Compose來讓各個部件能夠互相傳遞資料、交流。而Server內部的資料流程大概如下，首先，當學生將作業繳交後，Jenkins偵測到版本變更後，將程式碼分發給Maven以及SonarQube，Maven會做自動建置和單元測試，SonarQube會進行程式的品質分析。最後，Server會抓取它們的生成報告中的內容傳輸到前端來生成儀表板的內容。

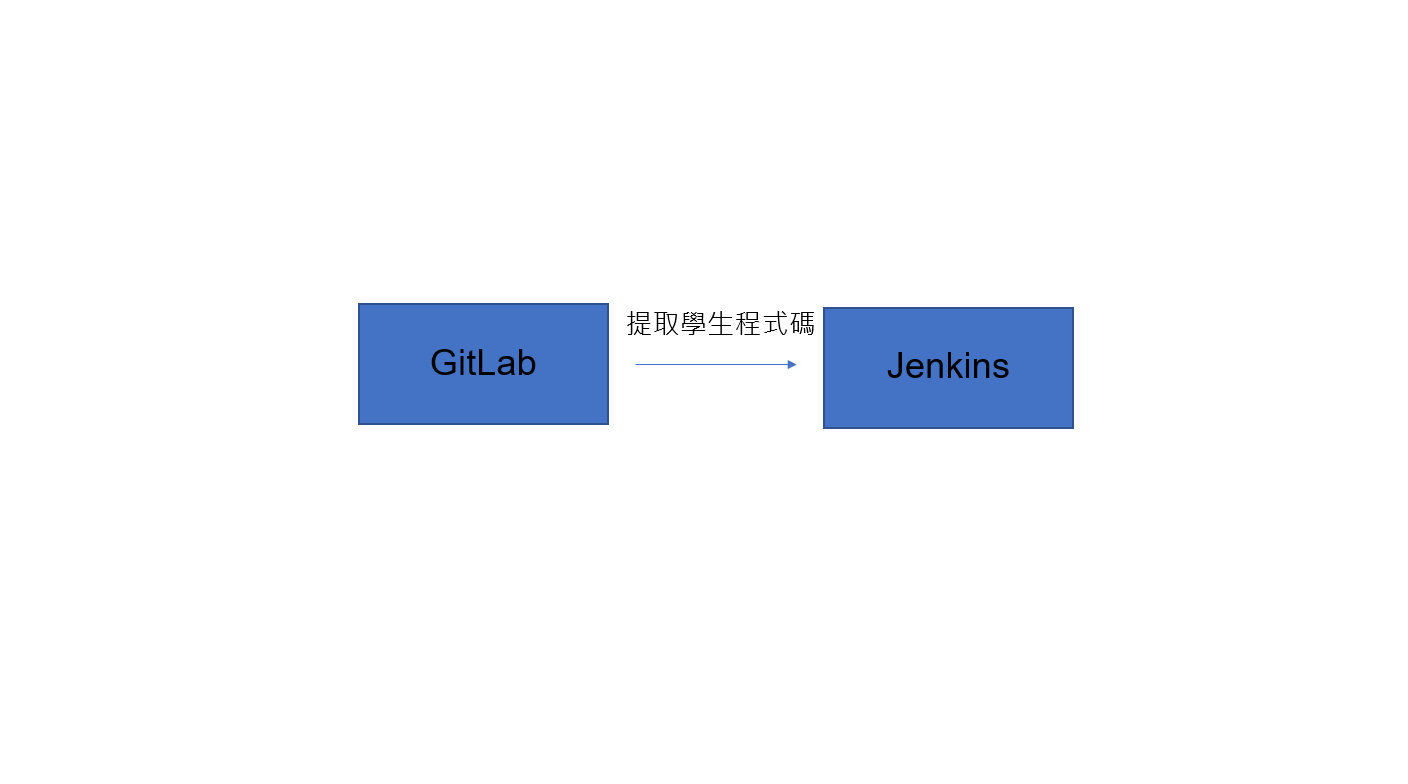
本次實驗方法分為三個階段，分別為：

階段一：基本Server建構(GitLab,Jenkins)

階段二：加入單元測試、程式品質檢測功能(SonarQube)

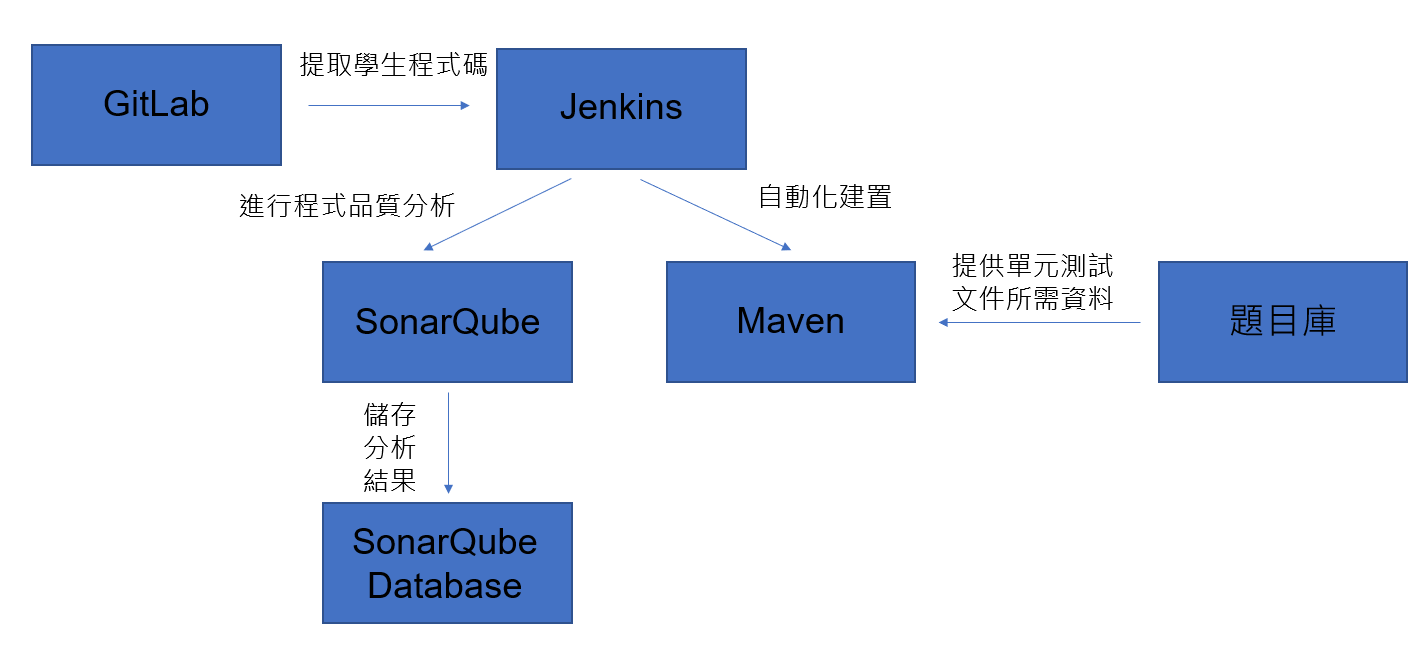
階段三：加入前端部分、資料庫

針對階段一的部分，我們會先建構最基礎的Server，包含GitLab和Jenkins的架設，其中GitLab會作為每個學生繳交作業的工具，這個部分會限制每個學生可存取Project的權限，以及會針對每個學生有修的課程進行不同Git位置的設定。Jenkins的部分，則會和GitLab進行連動，每當GitLab有新的檔案上傳時，Jenkins便會自動建置專案並對程式碼進行單元測試及品質測試。



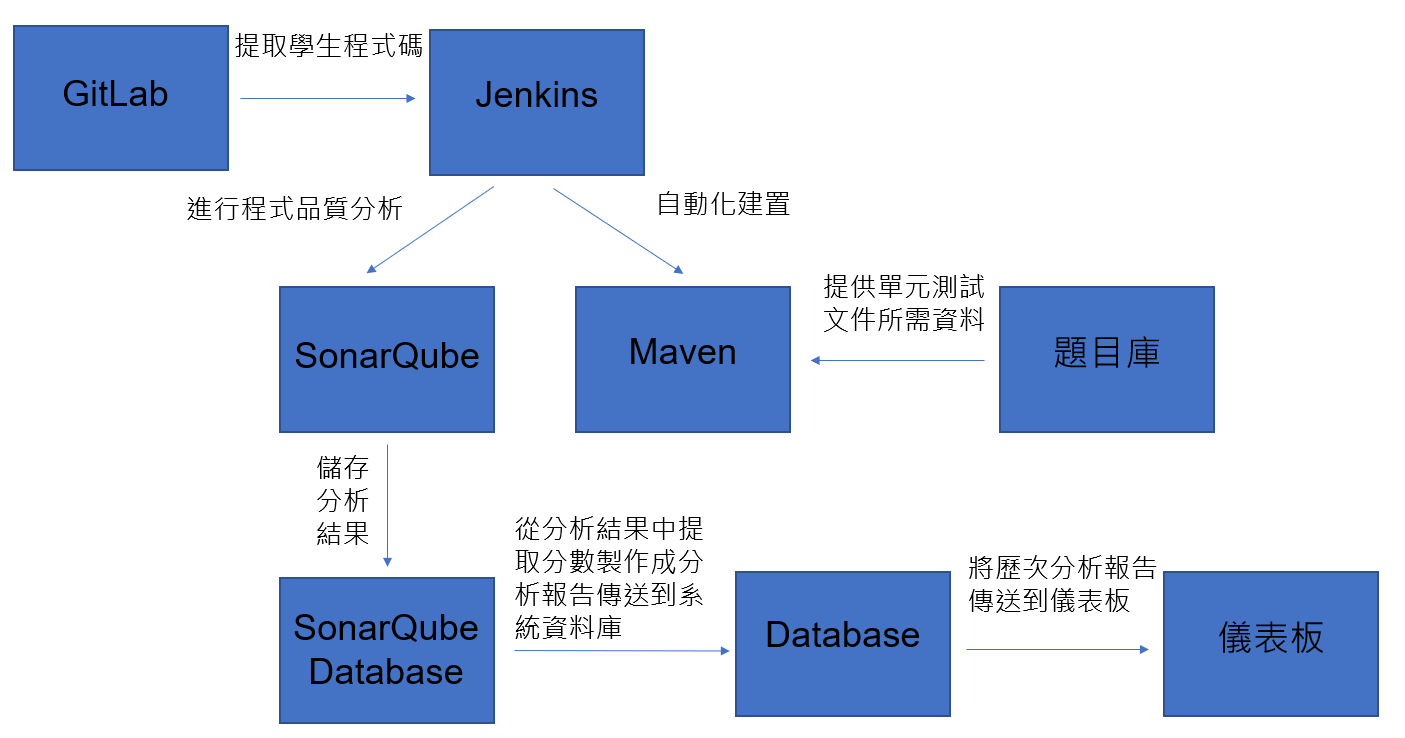
(圖四)基本Server架構

在階段二，我們會將SonarQube加入系統中，架設SonarQube後與Jenkins連接，也會架設屬於SonarQube的資料庫，SonarQube會將程式碼掃描後的結果儲存到資料庫，我們在從資料庫中抓取需要的資料彙整成一份報告傳送到儀表板，供學生參考。而單元測試則是透過Maven運行，我們會將題目庫中老師選取的題目的output抓取出來，做成單元測試的文件交由Maven進行單元測試。

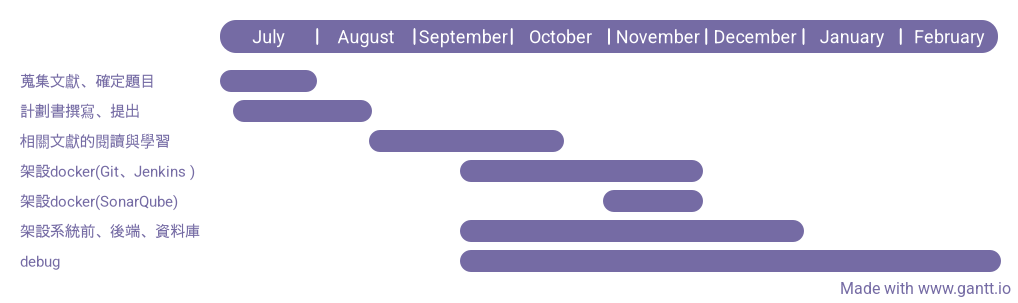


(圖五)加入單元測試和品質檢測的Server架構

第三階段，在我們將Server架設好後，我們會開始著手處理前端顯示內容及題目庫，前端的部分我們會將重點放在儀表板上，我們會從SonarQube Database 中提取該次作答的報告進行統整，留下三項指標安全性、可靠性以及可維護性的分數，Server會將該次的分數儲存到資料庫，我們會利用資料庫中歷次的分數來繪製學習曲線圖，以便了解各項指標的學習狀況。



(圖六)最終完整的Server架構

以下是我們預計的計畫甘特圖

(圖四)預計進度-甘特圖

1. 預期結果

預期功能:

(1) 系統中有多堂課程，老師以及學生可以同時擁有多堂課程

(2) 老師可以在一堂課程中開多堂課，方便老師管理學生。

(3) 老師在出題時可以自由選擇自己出題或是從題庫出題。

(4) 提供單元測試以及品質檢測。

(5) 報告結果整合到儀表板上。

(6) 提供學習曲線圖以便了解進步狀況。

1. 參考文獻

[1]H. -M. Chen, W. -H. Chen, N. -L. Hsueh, C. -C. Lee and C. -H.

Li, "ProgEdu - an automatic assessment platform for programming

courses," 2017 International Conference on Applied System

Innovation (ICASI), 2017, pp. 173-176, doi:

10.1109/ICASI.2017.7988376

[2]Pulasthi Perera, Roshali Silva, Indika Perera，"Improve Software

Quality through Practicing DevOps"2017 International Conference

on Advances in ICT for Emerging Regions (ICTer): 013 - 018

[3]Automated quality characteristic measures. Accessed:

2022-01-28. url: <https://it-cisq.org/standards/>

[automated-quality-characteristic-measures/.](https://it-cisq.org/standards/)

(七)需要指導教授指導內容

(1)研究計畫相關方向，內容提示。

(2)引導使用研究計畫相關之文獻。

(3)成果報告撰寫。

表C802

**三、耗材、物品、圖書及雜項費用：**

1. 凡執行研究計畫所需之耗材、物品、圖書及雜項費用，均可填入本表內。
2. 說明欄請就該項目之規格、用途等相關資料詳細填寫，以利審查。
3. 依研究計畫實際需求擇優補助，每一計畫最高以補助新臺幣20,000元為限。

金額單位：新臺幣元

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 項 目 名 稱 | 說明 | 單位 | 數量 | 單價 | | 金額 | 備註 |
|  |  |  |  |  | |  |  |
| 合　　　　　　　　計 | | | | |  | |  |

表C803

大專學生研究計畫指導教授初評意見表

一、學生潛力評估：

二、對學生所提研究計畫內容之評述：

三、指導方式：

四、本人同意指導學生瞭解並遵照學術倫理規範；本計畫無違反學術倫理。

指導教授簽名：

年 月 日表C804