|  |  |  |  |
| --- | --- | --- | --- |
|  | 時間 | 預計時程 | 標的 |
|  | 3/1~3/14 | 蒐集資料及各個元件建置嘗試(1,2,3,4,5) |  |
|  | 3/15~3/28 | 蒐集資料(6,7,8,9) |  |
|  | 3/29~4/11 | 蒐集資料(10, 11, 12) |  |
|  | 4/12~4/25 | 架設基本環境(包含GitLab ,SonarQube, Jenkins, Restful API) | 能運行基本功能並將各個部件連接 |
|  | 4/26~5/9 | 架設基本環境(包含題目庫,Maven pom.xml, database) | 蒐集題目庫題目,產生基本Maven pom.xml |
|  | 5/10~5/23 | 製作Test file generator | 能將題目庫的測資做成Maven中的測試程式 |
|  | 5/24~6/6 | 製作SonarQube report analyzer | 能把SonarQube報告中的特定  測試結果提出並存入database中 |
|  | 6/7~6/20 | 製作Suggestion code generator | 能產生建議程式碼,並存入database中 |
|  | 6/21~7/4 |
|  | 7/5~7/18 | 製作系統前端 | 包含系統登入介面,老師出題介面,學生答題介面,儀表板,管理員介面 |
|  | 7/19~8/1 |
|  | 8/2~8/15 | 系統測試及維護 |  |
|  | 8/16~8/31 |

1. Docker深入研究(Docker image , Docker compose)
2. Docker container嘗試建置GitLab , SonarQube, Jenkins
3. Docker container建置mySQL Database
4. Docker提供的Restful API 功能的深入研究

#generator的部分怎麼跟docker上的container做連接

1. Jenkins plugins (GitLab , SonarQube, Maven)
2. 前端框架選擇學習
3. 題目庫格式的決定(SCORM)
4. 決定Maven單元測試插件,pom.xml的設定及test file generator的製作方式
5. Report generator(包含細部如何將資料顯示在前端上,曲線圖繪圖工具)
6. Suggestion code generator 的製作方式(包含細部模版研究及可行性的評估)
7. SonarQube report analyzer的製作方式 (包含資料抓取方式及位置以及存放Database的方式)
8. GitLab學生project建置及處理(學生帳號)