

79

Object-Oriented Analysis and Design

計算機程式作業及考試系統

Homework 6

資工碩一 110598087 謝狄烽

資工碩一 110598108 郭宗育

目錄

1. Requirement Document	- 1 -
1.1 Change History	- 1 -
1.2 Problem statement.....	- 1 -
1.3 System Context Diagram	- 3 -
1.4 Summary of System Features	- 3 -
1.5 Use Case Diagram.....	- 4 -
1.6 Main Use Cases.....	- 5 -
1.6.1 Use Case ID: UC-01 新增作業	- 5 -
1.6.2 Use Case ID: UC-02 繳交作業	- 8 -
1.6.3 Use Case ID: UC-03 新增考試	- 12 -
1.6.4 Use Case ID: UC-04 參加線上考試	- 13 -
1.7 Non-functional Requirements and Constraints	- 14 -
1.8 Glossary	- 14 -
1.9 Software Environments.....	- 14 -
2. Domain Model	- 15 -
2.1 Domain class diagram showing only concepts	- 15 -
2.2 Add associations.....	- 17 -
2.3 Add attributes	- 18 -
3. Design	- 19 -
3.1 Logical Architecture.....	- 19 -
3.2 Use-Case Realizations with GRASP Patterns.....	- 19 -
3.2.1 System Sequence Diagram	- 19 -
3.2.2 Sequence Diagram with GRASP patterns.....	- 20 -
3.3 Design Class Model	- 23 -
4. Implementation Class Model	- 24 -
4.1 Draw an implementation class diagram for your system.....	- 24 -
4.2 Show the difference between ICD and DCD	- 25 -
4.3 Calculate Line of Code	- 32 -
5. Programming.....	- 35 -
5.1 Snapshots of system execution	- 35 -
5.2 Source Code Listing.....	- 46 -
6. Unit Testing.....	- 57 -
6.1 Snapshots of testing result	- 56 -
6.2 Unit Testing Code Listing.....	- 56 -
7. Measurement.....	- 66 -

1. Requirement Document

1.1 Change History

HW#	Revision	Description	Date
1	1	First Draft	2022/02/26
	2	Replenish Problem Statement	2022/02/28
2	3	Added Content And Paragraph	2022/03/09
	4	Added Use Cases And Diagrams	2022/03/15
3	5	Modified Use Cases & Diagram	2022/03/25
	6	Added Concept of Domain Class	2022/03/28
	7	Added Associations and Attributes	2022/03/29
4	8	Revised Section 2 - Domain Model	2022/04/20
	9	Added Logical Architecture	2022/04/20
	10	Added SSD, SD with GRASP	2022/04/20
	11	Added Design Class Model	2022/04/21
5	12	Revised Use Case 02 – 繳交作業	2022/04/29
	13	Revised Use Case 01 – 新增作業	2022/04/29
	14	Revised Section 2 - Domain Model	2022/04/29
	15	Revised Section 3 – SSD	2022/04/29
	16	Revised Section 3 – SD – UC02	2022/04/30
	17	Revised Section 3 – DCD	2022/05/01~ 2022/05/02
6	18	Revised Use Case Diagram	2022/05/18
	19	Revised Use Case Diagram 2 nd time	2022/05/19
	20	Revised System Sequence Diagram & Sequence Diagram & Implementation Class Diagram	2022/05/19
	21	Revised Design Class Diagram & table of different between DCD, ICD	2022/05/24

1.2 Problem statement

在過去，教師往往需要蒐集學生作業與試卷，逐一比較答案來進行批改，此舉不僅消耗額外的時間，更無法讓教師專注於教學上，因此若能將批改作業與試卷自動化，就可以讓教師更有效的運用時間，從而提升教學品質。

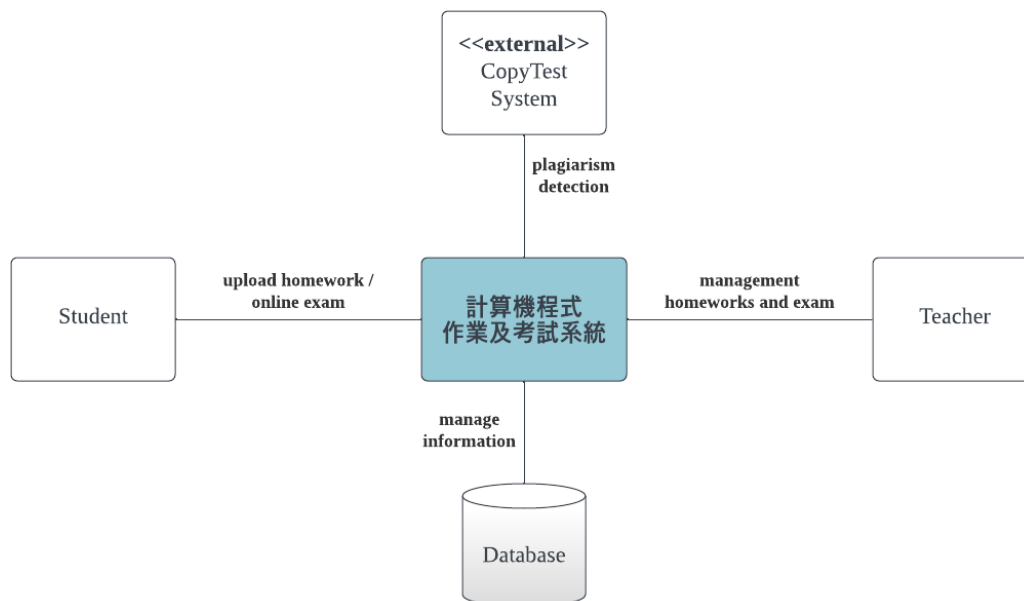
本系統是臺北科技大學計算機程式設計課程中所使用的系統，系統目的為減少教師課後批改同學作業與試卷的時間，並且可以更快的將作業與考試的結果回

饋予同學，藉此提高教師的教學品質與學生的學習效率。

教師可以在系統上出作業，出作業時需要填寫題目內容、題目測資、作答期限、設定程式語言種類，最後按下「公開題目」以及「開放上傳」，同學的作業上傳後會自動進行編譯以及執行，並且對同學程式輸入測資後，進行執行結果與正確結果的比對，使同學可以快速得知是否通過該題目。

教師亦可在系統上出考題，出考題時須填寫題目、答案、及考試開始與截止時間，開放考試後，同學可於系統上作答，並且作答完成即可得知考試分數。

1.3 System Context Diagram



計算機程式作業及考試系統，提供教師管理(新增/刪除/修改)作業，並且可以隨時隱藏/公開作業題目、管理題目測試資料，在更新題目測試資料後會重新批改所有已經上繳的作業；所有已擁有學生帳號的同學可以上傳作業程式檔案，支援上傳的檔案類型包含：.c、.py、.java。上繳作業後系統會立即以測試資料批改作業，同學可以在上傳數秒鐘後就看到批改結果，並且可以看到執行結果回饋，以便得知自己的程式是否涵蓋到所有題目測試資料情境。

本系統也提供教師在系統上出考題，考題與一般作業题目的不同之處在於：新增後的考試題目會記錄在「考試題庫」中，並不會出現在作業列表，並且是以選擇題的方式呈現考題。教師可以隨時管理(新增/刪除/修改)考題至「考試題庫」中，待教師要出試卷時可以將「考試題庫」中的候選考題新增至試卷上，最後設定試卷的開放和截止時間，學生可以在試卷公開的期間登入系統填寫試卷，填寫完成並交卷後可以即時看到自己的考試成績。

1.4 Summary of System Features

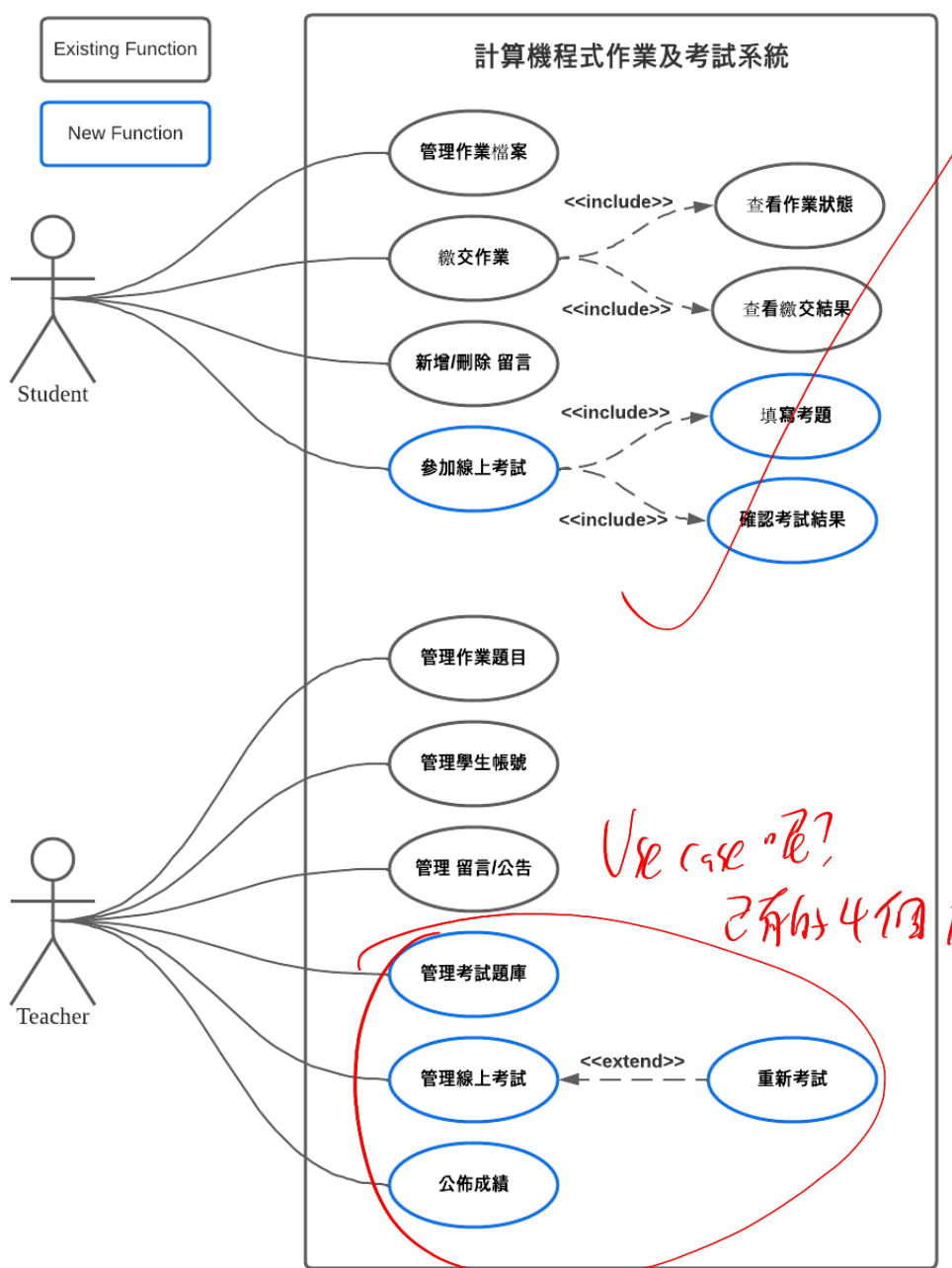
由於疫情影響，教學形式逐漸從原先的現場上課，轉換為遠距教學，同時也延伸出許多遠距考試的方式，然而大多遠距考試方案都還需要教師手動設定題目，且學生填寫完考卷後，也得由教師親自評測。因此決定將原有的計算機程式作業系統導入考試功能，讓教師能管理考試題庫，並從中出題，學生完成作答後也能自動與正確解答進行比對，並得到考試結果，藉此提升教學效率。

此外，系統也增加上傳及批改 java 檔案的功能，以滿足學習 java 的需求。

ID	Description
F AE-01	線上考試功能提供教師管理(新增/刪除/修改)考試題目。
F AE-02	線上考試功能提供教師新增試卷和考試。
F AE-03	線上考試功能提供同學線上考試。
F AE-04	新增提供上傳/批改 java 檔功能。

1.5 Use Case Diagram

Use Case Diagram



1.6 Main Use Cases

1.6.1 Use Case ID: UC-01 新增作業

Use Case Name	新增作業
Scope	計算機程式作業及考試系統
Level	User goal
Primary Actor	教師
Stakeholders and Interests	<p><u>學生</u>：能看到目前可繳交的作業，繳交截止日期並進行上傳</p> <p><u>教師</u>：能新增作業並設定作業內容、使用的程式語言、作業上傳期限、作業相似度門檻</p>
Preconditions	<p>以教師身分登入系統</p> <p>使用者進入作業清單頁面</p>
Success Guarantee	<p>題目內容、狀態被儲存</p> <p>題目編譯環境建立</p> <p>題目測試資料被儲存</p> <p>題目顯示在題目列表上</p>
Main Success Scenario	<ol style="list-style-type: none"> 1. 使用者點選出新題目 2. 系統顯示出新題目頁面 (題號、繳交期限、相似度、標題、題目內容、題目標籤預設為空，作業類型預設為課後練習，程式語言預設為C，出題方式預設為一般出題) 3. 使用者填寫題號、繳交期限、相似度、標題、題目內容 4. 系統即時更新對應欄位的輸入 5. 使用者選擇作業類型、作業程式語言、出題方式、題目標籤 6. 系統即時更新對應欄位的選項 7. 使用者確認新增作業 8. 系統新增作業 9. 系統回到作業清單頁面 10. 使用者確認作業已新增 11. 使用者點選測試資料 12. 系統顯示測試資料頁面 13. 使用者輸入測試資料

	14. 系統即時更新測試資料欄位 15. 使用者輸入預估正確結果 16. 系統即時更新預估正確結果欄位 17. 使用者點擊新增測試資料 18. 系統顯示測試資料編號、新增的測試資料與預估正確結果 19. 使用者回到作業清單頁面 20. 使用者設定題目為可上傳 21. 系統更新題目為可上傳 22. 使用者點選設為公開 23. 系統更新題目為公開狀態
Extensions	1a. 使用者點擊出新題目後沒有出現新題目頁面 1. Go To Step 1 2~6a. 使用者重新整理網頁 1. 系統顯示顯示所有成績頁面 2. 使用者回到作業清單頁面 3. Go To Step 1 2~6b. 使用者點選上方選單列選項 1. 系統顯示對應選項的頁面 2. 使用者回到作業清單頁面 3. Go To Step 1 7a. 使用者未填寫題號 1. 系統提示需要填寫題號 2. Go To Step 3 7b. 使用者未填寫繳交期限 1. 系統提示需要填寫繳交期限 2. Go To Step 3 7c. 使用者未填寫相似度 1. 系統提示需要相似度 2. Go To Step 3 7d. 使用者未填寫標題 1. 系統提示需要填寫標題 2. Go To Step 3 7e. 使用者未填寫題目內容 1. 系統提示需要填寫題目內容 2. Go To Step 3 8a. 系統新增作業失敗

	<ul style="list-style-type: none"> 1. 系統回到作業清單頁面 2. Go To Step 1 11a. 使用者點選測試資料以外的選項 <ul style="list-style-type: none"> 1. 系統顯示對應選項的頁面 2. Go To Step 9 11b. 使用者不新增測試資料 <ul style="list-style-type: none"> 1. Go To Step 19 12~17a. 使用者重新整理頁面 <ul style="list-style-type: none"> 1. 系統顯示顯示所有成績頁面 2. Go To Step 9 12~17b. 使用者點選上方選單列選項 <ul style="list-style-type: none"> 1. 系統顯示對應選項的頁面 2. Go To Step 9 18~19a. 使用者點擊刪除測試資料 <ul style="list-style-type: none"> 1. 系統刪除該筆測試資料 2. Go To Step 12 20a. 使用者不將作業設為可上傳 <ul style="list-style-type: none"> 1. 系統不更新作業為可上傳 2. Go To Step 21 22a. 使用者不將作業設為公開狀態 <ul style="list-style-type: none"> 1. 系統不更新作業狀態 2. 結束新增作業流程
Special Requirements	無
Technology and Data Variations List	無
Frequency of Occurrence	經常使用
Miscellaneous	無

1.6.2 Use Case ID: UC-02 繳交作業

Use Case Name	繳交作業
Scope	計算機程式作業及考試系統
Level	User goal
Primary Actor	學生
Stakeholders and Interests	<p><u>學生</u>：繳交作業，並即時得到作業的批改結果</p> <p><u>教師</u>：得到所有學生繳交的作業、其狀態、與批改結果</p>
Preconditions	<p>使用者以學生身分登入系統</p> <p>使用者進入作業清單頁面</p>
Success Guarantee	<p>系統儲存學生繳交的作業</p> <p>系統成功編譯學生繳交的作業</p> <p>作業執行結果、批改結果被儲存</p> <p>系統正常顯示作業批改結果</p>
Main Success Scenario	<ol style="list-style-type: none"> 1. 使用者點選繳交 2. 系統顯示繳交作業頁面 3. 使用者點擊選擇檔案，以上傳主程式 4. 系統出現上傳檔案對話視窗 5. 使用者選擇欲上傳之主程式 6. 使用者點擊開啟（確定） 7. 系統顯示使用者上傳檔案的檔名 8. 使用者點擊upload，確定上傳 9. 系統顯示上傳成功 10. 系統進行編譯與測試 11. 使用者回到作業清單頁面 12. 使用者確認作業狀態 13. 使用者點擊查看結果 14. 系統顯示使用者第一筆測試資料的執行結果，以及所有測試資料的測試結果 15. 使用者確認測試資料是否全數通過
Extensions	<p>1a. 使用者點擊繳交以外的選項</p> <ol style="list-style-type: none"> 1. 系統顯示對應選項的頁面 2. 使用者回到作業清單頁面

	<p>3. Go To Step1</p> <p>1b. 使用者已經繳交過作業</p> <ol style="list-style-type: none"> 1. 系統顯示「若要重新繳交，請刪除作業，再重新上傳！」 2. 使用者點選刪除 3. 系統詢問使用者是否確定刪除作業 4a. 使用者選擇是 4b. 使用者選擇否 1. Go To Step 1b - 1 5. 系統刪除作業 6. 系統顯示刪除成功 7. 使用者回到作業清單頁面 8. Go To Step 1 <p>1c. 作業超過繳交期限</p> <ol style="list-style-type: none"> 1. 使用者聯繫教師，要求延長繳交期限 2. Go To Step 1 <p>2~8a. 使用者重新整理網頁</p> <ol style="list-style-type: none"> 1. 系統回到首頁 2. 使用者回到作業清單頁面 3. Go To Step 1 <p>2~8b. 使用者點選上方選單列選項</p> <ol style="list-style-type: none"> 1. 系統顯示對應選項的頁面 2. 使用者回到作業清單頁面 3. Go To Step 1 <p>8a. 作業與其他學生繳交的作業相似</p> <ol style="list-style-type: none"> 1. 系統判定抄襲 2. 系統拒絕使用者上傳該檔案 <p>8b. 上傳檔案的檔案類型與作業要求不符</p> <ol style="list-style-type: none"> 1. 系統顯示「作業檔案類型不符，請上傳有效檔名」 2. Go To Step 3 <p>8c. 上傳的檔案名稱含有非英文數字的文字</p> <ol style="list-style-type: none"> 1. 系統顯示「作業檔案類型不符，請上傳有效檔名」
--	---

	<p>2. Go To Step 3</p> <p>8d. 上傳的檔案大小超過2MB</p> <ol style="list-style-type: none"> 1. 系統顯示上傳檔案失敗 2. 使用者回到作業清單頁面 <p>3. Go To Step1</p> <p>8e. 沒有上傳檔案</p> <ol style="list-style-type: none"> 1. 系統顯示「請先上傳檔案」 <p>2. Go To Step 3</p> <p>8f. 作業上傳的瞬間超過繳交期限</p> <ol style="list-style-type: none"> 1. 系統顯示「作業已超過繳交期限，請聯繫助教要求延長期限」 2. 使用者回到作業清單頁面 <p>3. Go To Step 1</p> <p>8g. 使用者點擊reset</p> <ol style="list-style-type: none"> 1. 系統清除使用者上傳的檔案 <p>2. Go To Step 3</p> <p>11~15a. 使用者重新整理網頁</p> <ol style="list-style-type: none"> 1. 系統回到首頁 <p>2. Go To Step 11</p> <p>11~15b. 使用者點選上方選單列選項</p> <ol style="list-style-type: none"> 1. 系統顯示對應選項的頁面 <p>2. Go To Step 11</p> <p>12a. 作業狀態為通過</p> <p>12b. 作業狀態為未通過</p> <p>12c. 作業狀態為批改中</p> <ol style="list-style-type: none"> 1. 使用者等待數秒 <p>2. Go To Step 11</p> <p>12d. 作業狀態為編譯失敗</p> <p>12e. 作業狀態為待批改、尚未設定測資</p> <ol style="list-style-type: none"> 1. 使用者聯繫教師，請教師設定測試資料 <p>2. Go To Step1</p> <p>12f. 使用者不確認作業狀態</p> <ol style="list-style-type: none"> 1. 結束繳交題目流程 <p>13a 使用者點擊查看結果以外的選項</p> <ol style="list-style-type: none"> 1. 系統顯示對應選項的頁面 <p>2. Go To Step 11</p>
--	---

	15a. 使用者通過所有測資 15b. 使用者未通過所有測資 1. 使用者修改程式碼 2. Go To Step 1 15c. 使用者繳交的檔案編譯失敗 1. 系統顯示編譯失敗的區塊 2. 使用者修改程式碼 3. Go To Step 1 15d. 使用者使用system函數 1. 系統顯示「請勿使用system函數」 2. 使用者刪除程式中的system字串 3. Go To Step 1
Special Requirements	無
Technology and Data Variations List	1. 學生上傳作業後，應能確認目前作業狀態為通過、未通過、批改中
Frequency of Occurrence	經常使用
Miscellaneous	無

1.6.3 Use Case ID: UC-03 新增考試

Use Case Name	新增考試
Scope	計算機程式作業及考試系統
Level	User goal
Primary Actor	教師
Stakeholders and Interests	<u>學生</u> ：能參加線上考試 <u>教師</u> ：能新增線上試卷
Preconditions	以教師身分登入系統
Success Guarantee	系統正確新增考卷，學生線上參加考試
Main Success Scenario	<ol style="list-style-type: none"> 1. 教師選擇設定考試功能 2. 教師設定考試所使用的考試題庫 3. 教師設定考試題數與考試時間 4. 教師成功新增考試 5. 考試時間到後，開始考試
Extensions	2a. 題庫內無考題存在 <ol style="list-style-type: none"> 1. 教師新增考題至題庫
Special Requirements	無
Technology and Data Variations List	無
Frequency of Occurrence	偶爾使用
Miscellaneous	無

1.6.4 Use Case ID: UC-04 參加線上考試

Use Case Name	參加線上考試
Scope	計算機程式作業及考試系統
Level	User goal
Primary Actor	學生
Stakeholders and Interests	<u>學生</u> ：能線上考試，並在考完試後即時得知考試結果 <u>教師</u> ：能統計該考試所有學生的分數
Preconditions	以學生身分登入系統
Success Guarantee	學生得到考試結果，且結果被系統紀錄
Main Success Scenario	<ol style="list-style-type: none"> 1. 學生參加考試 2. 學生確定加入考試 3. 學生根據題目，選擇相對應的答案 4. 學生填寫完答案，確定交卷 5. 學生確認考試結果
Extensions	<ol style="list-style-type: none"> 1a. 學生已參與過考試 <ol style="list-style-type: none"> 1. 系統提示學生已參與過考試 4a. 學生有未作答的題目 <ol style="list-style-type: none"> 1. 系統提示學生該題未作答
Special Requirements	無
Technology and Data Variations List	無
Frequency of Occurrence	偶爾使用
Miscellaneous	無

1.7 Non-functional Requirements and Constraints

NFR ID	Category	Description
NFR-01	Usability	學生按下 upload 上傳作業時，轉接「處理中」頁面
NFR-02	Usability	學生作業尚在測試時，備註顯示「批改中」
NFR-03	Usability	測試完成時，提示學生第一筆測資的執行結果
NFR-04	Security	提升安全防護等級
NFR-05	Performance	學生繳交作業後，須於 30 秒內完成編譯與測試

1.8 Glossary

Item	Description
測試結果	程式以測試資料為輸入，執行後所產生的輸出結果。
考試題庫	儲存考試題目的資料庫，由教師從網頁端進行管理。
程式碼抄襲比對系統	能夠比對程式碼之間是否有抄襲情形的系統，將檔案互相比較之後，會得到相似度，相似度越高表示抄襲的可能性越大。
相似度門檻	一個數字門檻，以零到一百表示，當程式碼抄襲比對系統得到的相似度高於該門檻，系統將判定該檔案為抄襲。

1.9 Software Environments

Development language	Jsp & Java
Development IDE	Eclipse EE
Server	Tomcat
C Compiler	GCC 7.5

2. Domain Model

2.1 Domain class diagram showing only concepts

(1) 從 use case 中找出所有名詞：

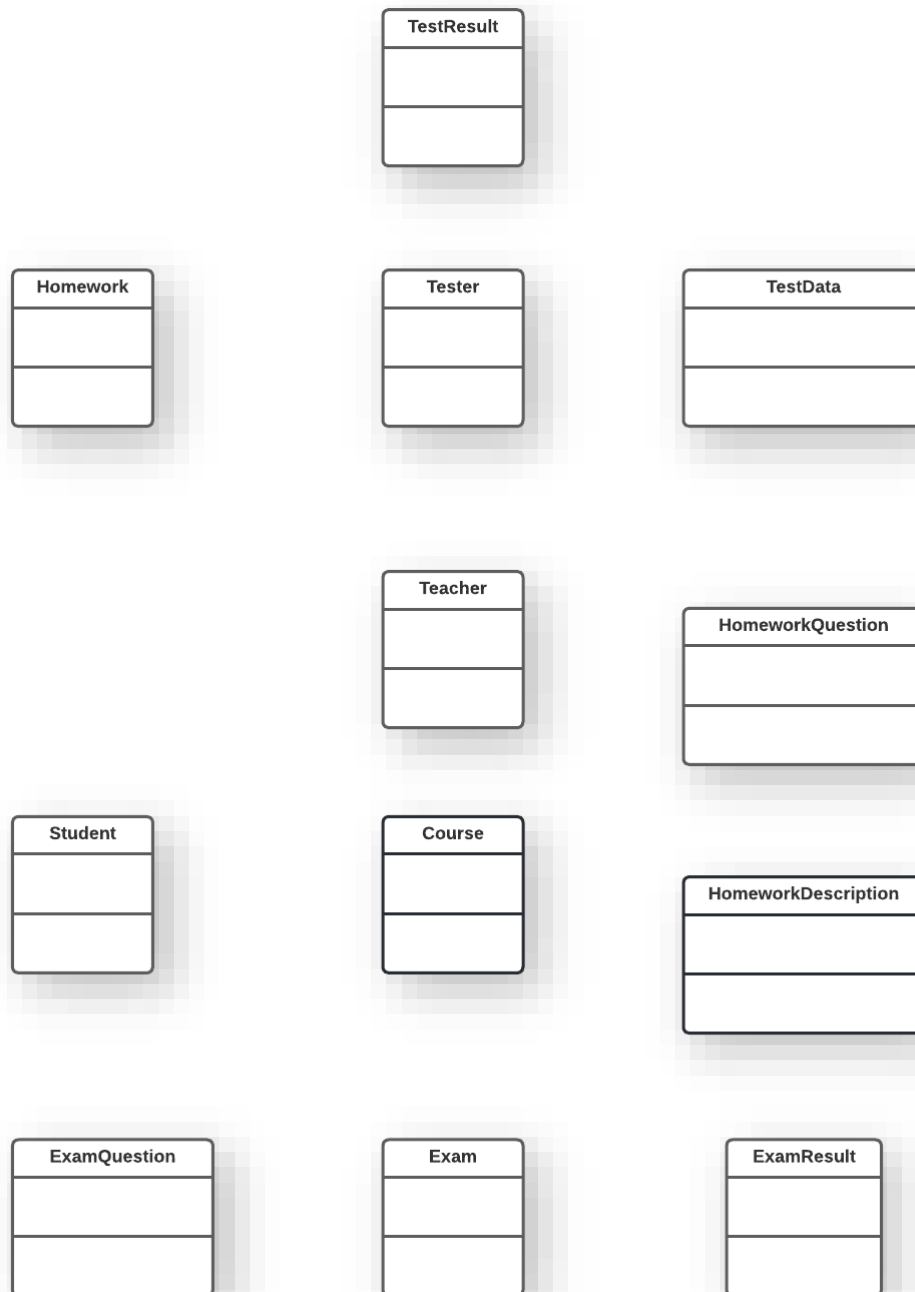
學生	教師	作業題目	作業內容
程式語言	作業上傳期限	作業相似度門檻	作業測試資料
作業列表	作業批改結果	學生作業	作業執行結果
抄襲比對系統	作業題號	作業狀態	作業測試結果
考試	考題	考試題庫	考試時間
考題內容	考題答案	學生考試結果	學生考試分數

(2) 從以上名詞中篩選是否適合做為 class，以及其理由：

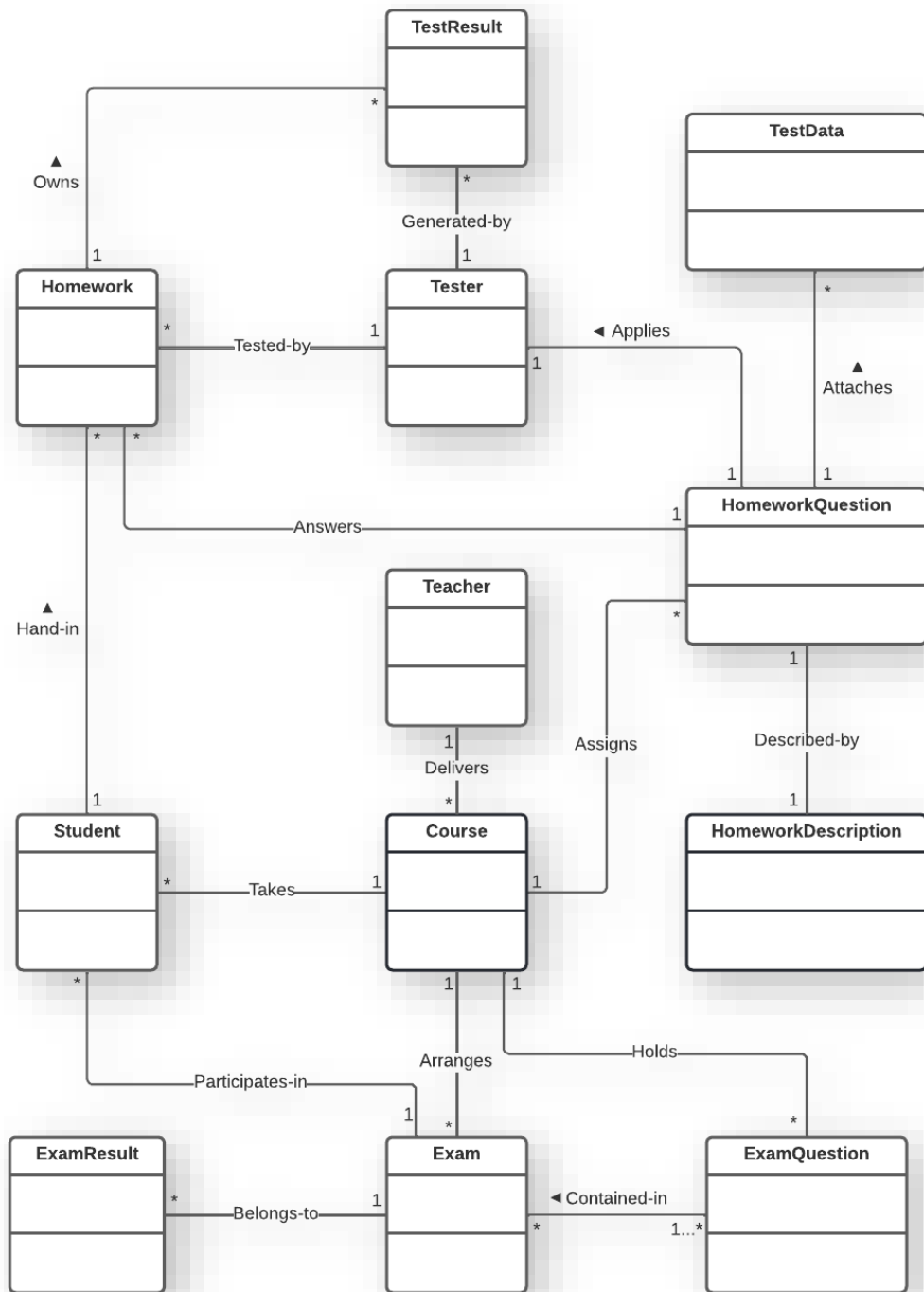
Good Classes			
Class Name	Reason	Class Name	Reason
學生	包含學生姓名學號等各種資訊	學生作業	包含對應題目、所屬學生等資訊
教師	包含教師姓名編號等各種資訊	考試	包含考試編號、考題、考試時段等各種資訊
作業題目	包含題號、內容、期限等資訊	考題	包含考題內容與考題答案
作業測試資料	包含所屬題號、輸入內容與預期結果	學生考試結果	包含學生學號、分數、填寫答案等資訊
作業測試結果	包含執行結果、是否通過等資訊		

Bad Classes			
Class Name	Reason	Class Name	Reason
作業內容	Is an attribute	作業列表	UI aspect
程式語言	Is an attribute	作業批改結果	Is an attribute
作業上傳期限	Is an attribute	作業執行結果	Is an attribute
作業相似度門檻	Is an attribute	抄襲比對系統	External system
作業題號	Redundant	作業狀態	Is an attribute
考試題庫	Redundant	學生考試分數	Is an attribute
考試時間	Is an attribute	考題內容	Is an attribute
考題答案	Is an attribute		

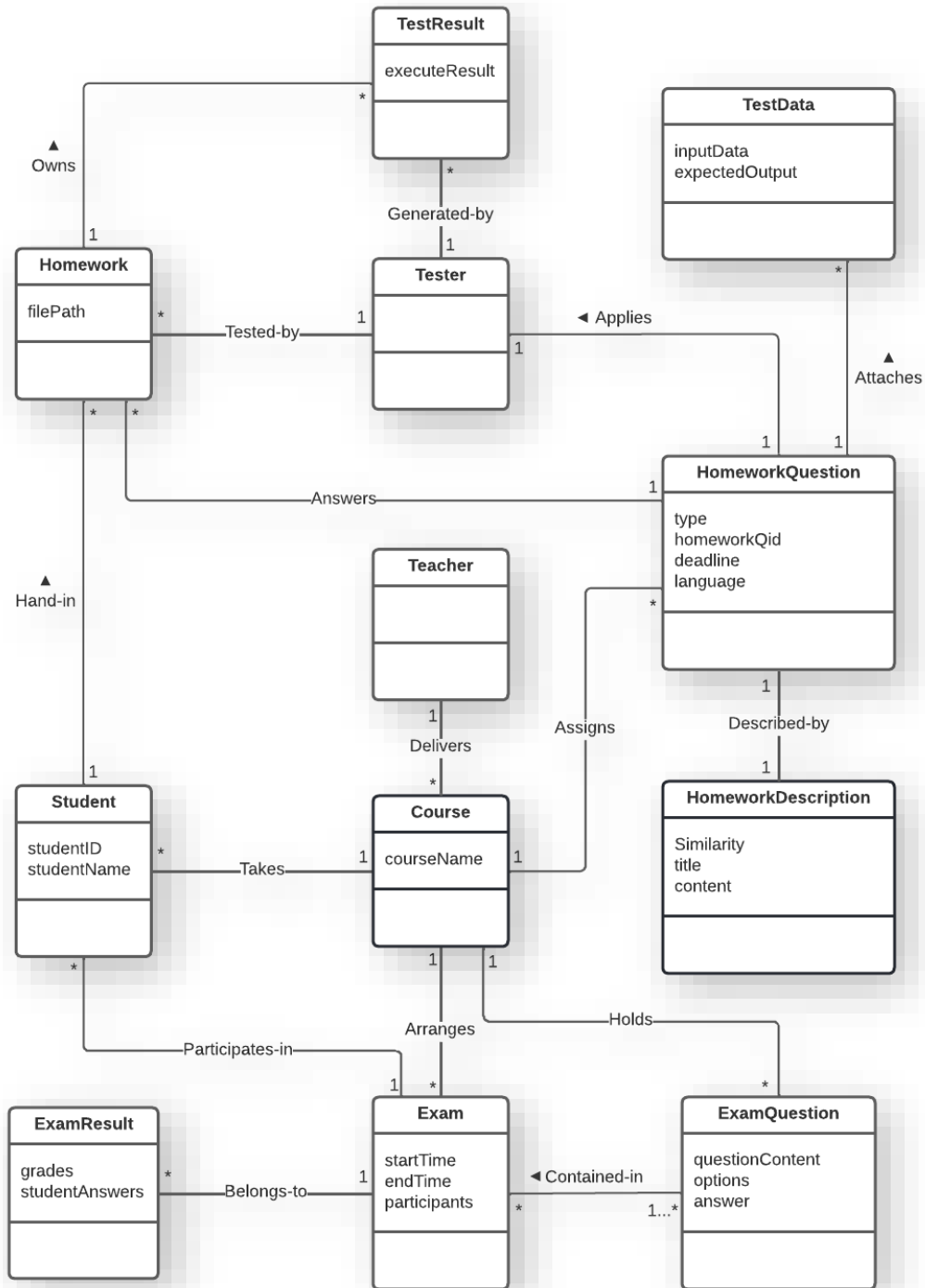
(3) 根據以上分析，最後得出 Concept of Domain Model



2.2 Add associations

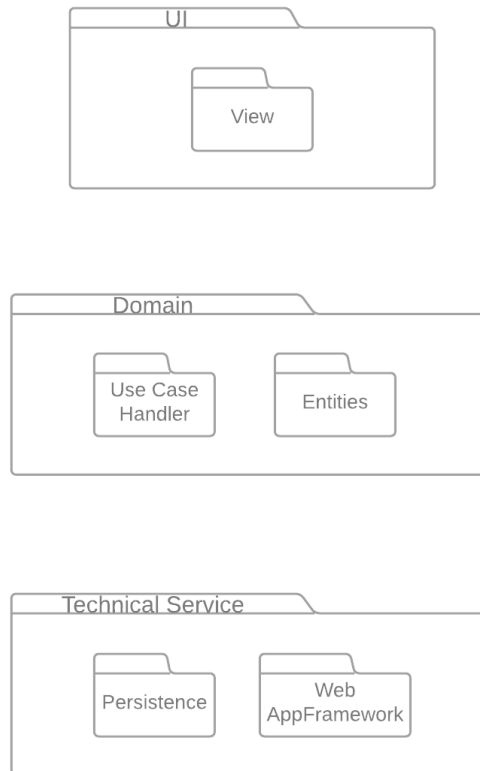


2.3 Add attributes



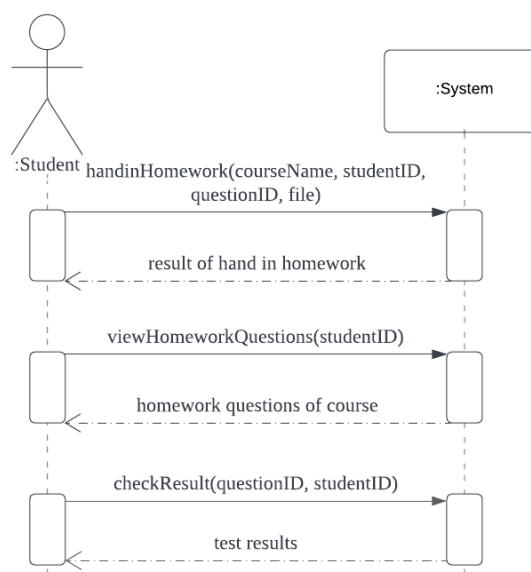
3. Design

3.1 Logical Architecture



3.2 Use-Case Realizations with GRASP Patterns

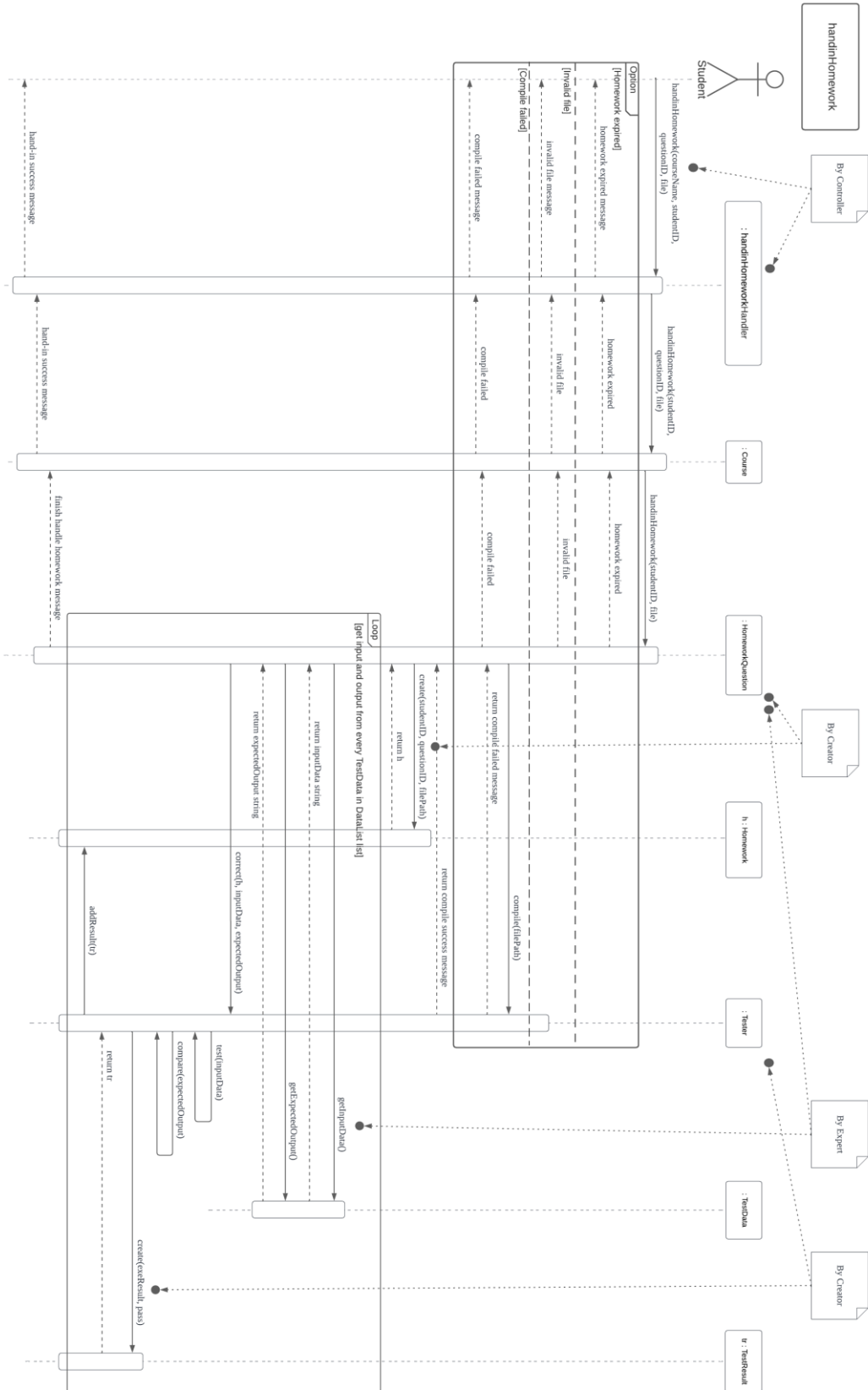
3.2.1 System Sequence Diagram



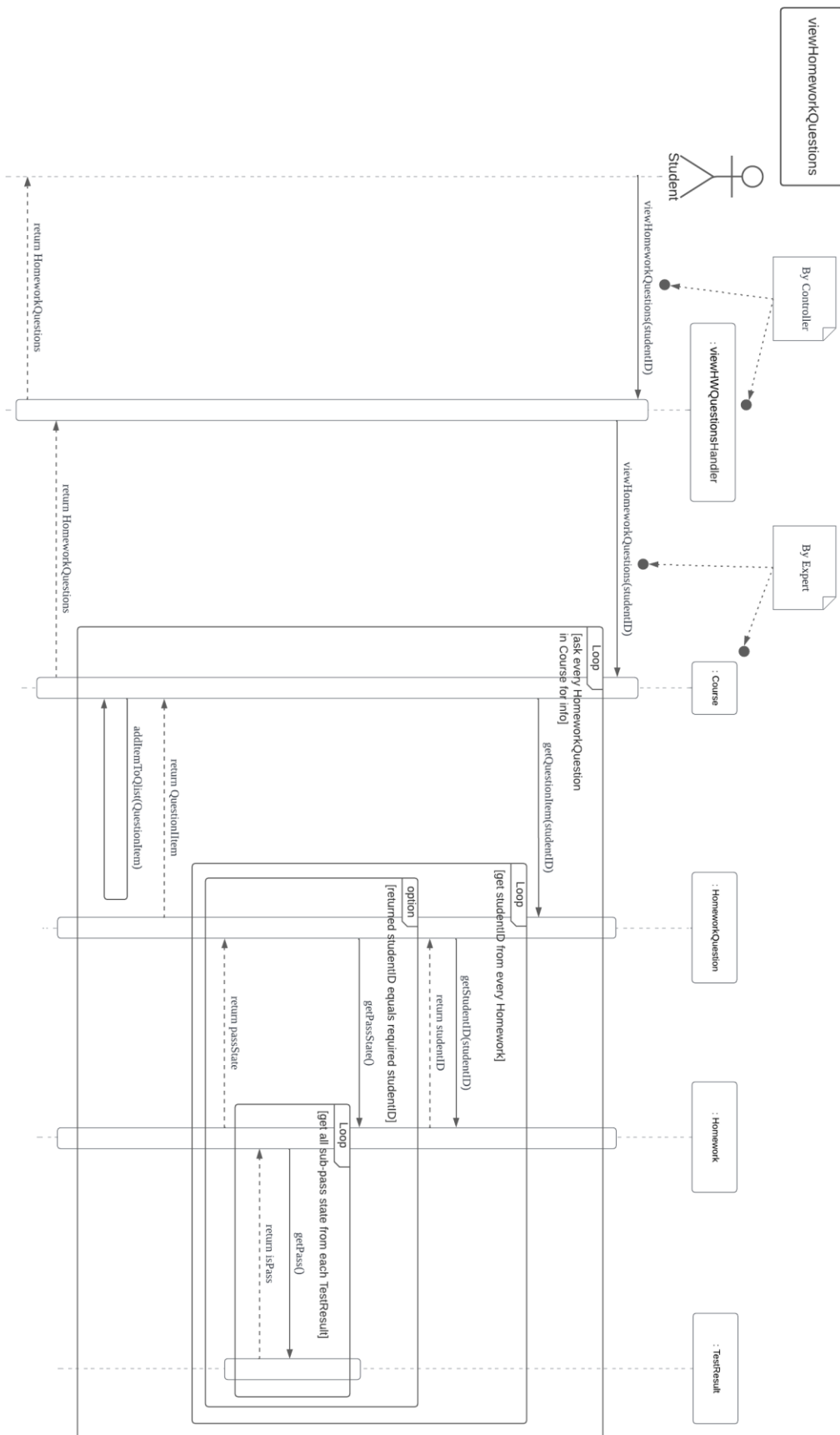
UC-02 繳交作業

3.2.2 Sequence Diagram with GRASP patterns

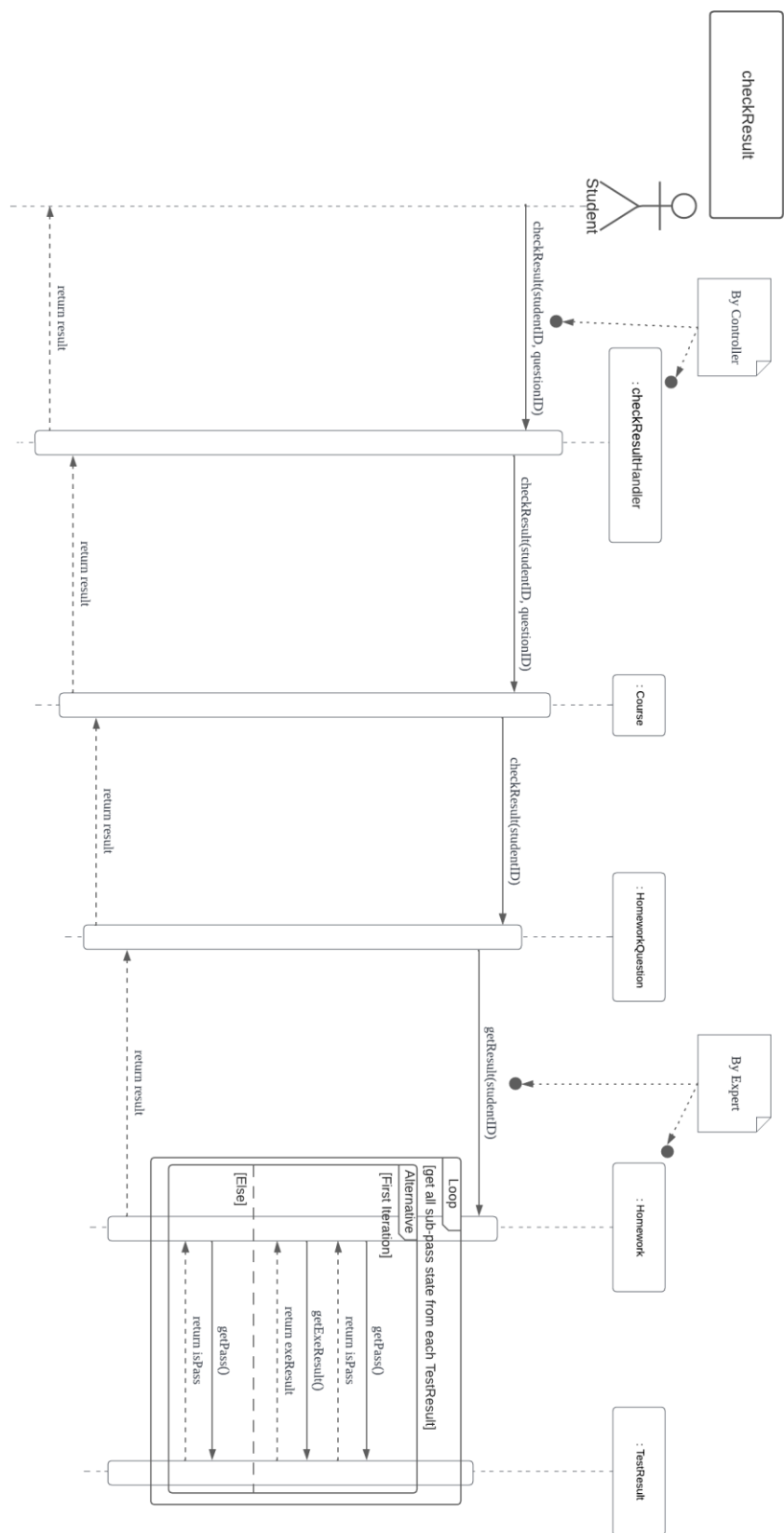
3.2.2.1 handinHomework



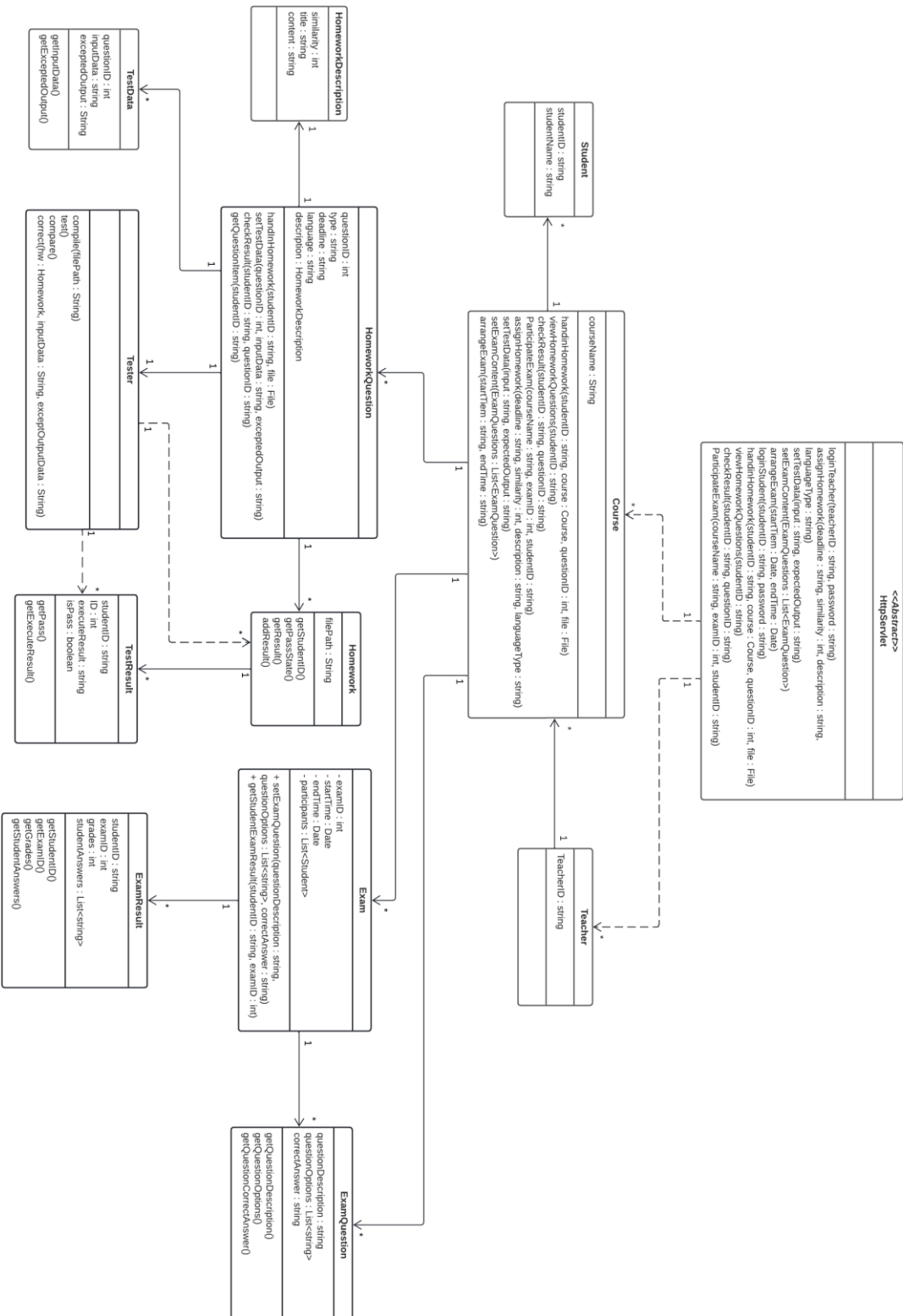
3.2.2.2 viewHomeworkQuestions



3.2.2.3 checkResult



3.3 Design Class Model



4. Implementation Class Model

4.1 Draw an implementation class diagram for your system



4.2 Show the difference between ICD and DCD

Table 4.2.1: Comparison with design and implementation class

Class	Method	Design	Imp
TestData	getId	yes	yes
	setId	yes	yes
	getQuestionID	yes	yes
	setQuestionID	yes	yes
	setInputData	yes	yes
	getInputData	yes	yes
	setExceptedOutput	yes	yes
	getExceptedOutput	yes	yes
TestResult	setId	yes	yes
	getId	yes	yes
	setStudentId	yes	yes
	getStudentId	yes	yes
	setTestDataId	yes	yes
	getTestDataId	yes	yes
	setDescription	yes	yes
	getDescription	yes	yes
	setExecuteResult	yes	yes
	getExecuteResult	yes	yes
	setPass	yes	yes
	isPass	yes	yes
	setResult	yes	yes
	getResult	yes	yes

Course	getId	yes	yes
	getName	yes	yes
	getTeacherID	no	yes
	getTeacherName	no	yes
	addCourse	no	yes
	handinHomework	yes	yes
	viewHomeworkQuestions	yes	yes
	checkResult	yes	yes
	ParticipateExam	yes	yes
	assignHomework	yes	yes
	setTestData	yes	yes
	editExamQuestion	no	yes
	setExamContent	yes	yes
	arrangeExam	yes	yes
	doingExam	no	yes
	viewExamQuestion	no	yes
Tester	compilerFile	yes	yes
	checkResult	no	yes
	versionOfStrcmp	no	yes
	setDB	no	yes
	getDB	no	yes
	test	yes	no
	compare	yes	no
	correct	yes	no
TestThread	run	no	yes

User	getName	no	yes
	getPasswd	no	yes
	getCourse	no	yes
	getCoursePath	no	yes
	getDbName	no	yes
	getUser	no	yes
	checkLogin	no	yes
Student	getScoreSize	no	yes
	countUp	no	yes
	getStudentPass	no	yes
	getScore	no	yes
	setScoreDB	no	yes
	setAllScoreDB	no	yes
	setPasswd	yes	yes
Teacher	setPasswd	yes	yes
	getManagerIP	no	yes
	checkLogin	no	yes
HomeworkDescription	setSimilarity	yes	yes
	getSimilarity	yes	yes
	setTitle	yes	yes
	getTitle	yes	yes
	setContent	yes	yes
	getContent	yes	yes
Homework	getId	yes	yes
	getStudentID	yes	yes

	getTime	no	yes
	getTitle	no	yes
	getFileDesc	no	yes
	getFileName	no	yes
	getSystemName	no	yes
	getFilePath	yes	yes
	getPassState	yes	yes
	getResult	yes	yes
	setTestResults	yes	yes
	addResult	yes	yes
	getTestResults	yes	yes
HomeworkQuestion	getQuestionItem	yes	yes
	setTestDatas	yes	yes
	getTestDatas	yes	yes
	setHomeworks	no	yes
	getHomeworks	no	yes
	getType	yes	yes
	getId	yes	yes
	getContent	yes	yes
	getNoBRContent	no	yes
	getDeadline	yes	yes
	getlanguage	yes	yes
	isExpired	no	yes
	setSimilarity	yes	yes
	getSimilarity	yes	yes

	getTitle	no	yes
	getContent	no	yes
	toString	no	yes
Exam	setExamQuestion	yes	yes
	getStudentExamResult	yes	yes
	getTotalQuesCount	no	yes
	getQuesCount	no	yes
	getLegalStartTime	yes	yes
	getLegalEndTime	yes	yes
	isLegalTime	yes	yes
	setTime	yes	yes
	genQues	no	yes
StudentExam	init	no	yes
	getMaxEndTime	no	yes
	getUserCurrent	no	yes
	getUserRight	no	yes
	getQuesCount	no	yes
	getUserAnswerLength	no	yes
	getUserAnswer	no	yes
	getQuesString	no	yes
	addScore	no	yes
	isTimeOver	no	yes
	isExamOver	no	yes
ExamQuestion	getQuestionId	yes	yes
	getContent	yes	yes

	getType1	no	yes
	getType2	no	yes
	getTagContent	no	yes
	getAns	yes	yes
	addQues	yes	yes
	editQues	yes	yes
	setQues	yes	yes
	isExistQues	no	yes
	getNextId	no	yes
	getQues	no	yes
	getAllQues	no	yes
	getQuestionOptions	yes	no
ExamScore (Rename:ExameResult)	getId	yes	yes
	getQuesNum	no	yes
	getRightNum	no	yes
	setExamScore	no	yes
	getOneExamScore	no	yes
	getAllExamScores	no	yes
	getGrades	yes	no
	getStudentAnswers	yes	no
CourseDAO	getCourse	no	yes
	getAllCourse	no	yes
HomeworkDAO	getAllQuestionByStudent	no	yes
HomeworkQuestionDAO	getQuestion	no	yes
	getAllQuestion	no	yes

TestDataDAO	getTestDatas	no	yes
TestResultDAO	getTestResults	no	yes
AssignHomework	doGet	no	yes
Handler (Servlet)	doPost	no	yes
SetTestDataHandler	doGet	no	yes
(Servlet)	doPost	no	yes
HandInHomework	doGet	no	yes
Handler (Servlet)	doPost	no	yes
CheckResultHandler	doGet	no	yes
(Servlet)	doPost	no	yes
ViewHomework	doGet	no	yes
QuestionsHandler	doPost	no	yes
(Servlet)			
ParticipateExamHandler	doGet	no	yes
(Servlet)	doPost	no	yes
DoingExamHandler	doGet	no	yes
(Servlet)	doPost	no	yes
ViewExamQuestion	doGet	no	yes
Handler (Servlet)	doPost	no	yes
EditExamQuestion	doGet	no	yes
Handler (Servlet)	doPost	no	yes
SetExamContent	doGet	no	yes
Handler (Servlet)	doPost	no	yes
ArrangeExam	doGet	no	yes
Handler (Servlet)	doPost	no	yes

Table 4.2.2: Summary of implementation class/method changed

	Number of added	Number of removed	Number of modified
Class	19	2	1
Method	45	11	16

4.3 Calculate Line of Code

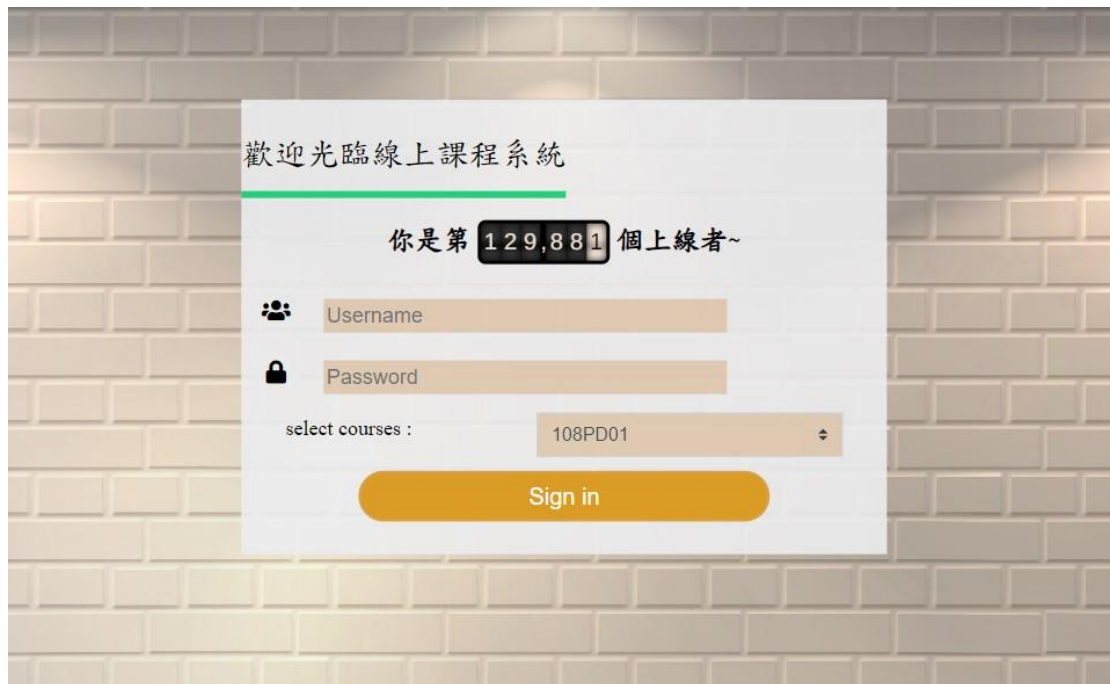
No	Class Name	Number of Method	Line of Code in Class
1	TestData	10	32
2	TestResult	17	48
3	Course	18	132
4	Tester	13	453
5	TestThread	4	101
6	User	10	68
7	Student	13	169
8	Teacher	4	54
9	HomeworkDescription	7	36
10	Homework	20	120
11	HomeworkQuestion	26	324
12	Exam	8	79
13	StudentExam	12	138
14	ExamQuestion	15	182
15	ExamScore	8	84
16	CourseDAO	2	43

17	HomeworkDAO	1	28
18	HomeworkQuestionDAO	2	64
19	TestDataDAO	1	37
20	TestResultDAO	1	43
21	AssignHomeworkHandler (Servlet)	2	98
22	SetTestDataHandler (Servlet)	2	96
23	HandInHomeworkHandler (Servlet)	2	404
24	CheckResultHandler (Servlet)	2	102
25	ViewHomeworkQuestionsHandler (Servlet)	2	58
26	ParticipateExamHandler (Servlet)	3	68
27	DoingExamHandler (Servlet)	2	118
28	ViewExamQuestionHandler (Servlet)	2	52
29	EditExamQuestion	2	50

	Handler (Servlet)		
30	SetExamContent Handler (Servlet)	2	30
31	ArrangeExam Handler (Servlet)	2	122
	Total	215	3433

5. Programming

5.1 Snapshots of system execution



學生端登入



學生端首頁

									
	作業繳交	留言版	更改密碼						
4	課後作業	04	2022/03/09 23:59	期限已過	C	通過	查看結果	通過測試名單	
5	課後作業	05	2022/03/09 23:59	期限已過	C	通過	查看結果	通過測試名單	
6	課後作業	06	2022/03/09 23:59	期限已過	C	未繳	-	通過測試名單	
7	課後作業	07	2022/03/23 19:05	期限已過	C	通過	查看結果	通過測試名單	
8	課後作業	08	2022/03/23 19:05	期限已過	C	通過	查看結果	通過測試名單	
9	課後作業	09	2022/03/23 19:05	期限已過	C	通過	查看結果	通過測試名單	
10	課後作業	10	2022/03/24 12:00	期限已過	C	未通過	查看結果	通過測試名單	
11	課後作業	11	2022/03/24 12:00	期限已過	C	通過	查看結果	通過測試名單	
12	課後作業	12	2022/03/24 12:00	期限已過	C	通過	查看結果	通過測試名單	
13	課後作業	13	2022/03/30 23:59	期限已過	C	通過	查看結果	通過測試名單	
14	課後作業	14	2022/03/30 14:05	期限已過	C	通過	查看結果	通過測試名單	
15	課後作業	15	2022/04/06 23:59	期限已過	C	通過	查看結果	通過測試名單	
16	課後作業	16	2022/04/06 23:59	期限已過	C	通過	查看結果	通過測試名單	
17	課後作業	17	2022/04/06 18:10	期限已過	C	通過	查看結果	通過測試名單	
18	課後作業	18	2022/04/14 00:12	期限已過	C	未繳	-	通過測試名單	
19	課後作業	19	2022/04/13 23:59	期限已過	C	通過	查看結果	通過測試名單	
20	課後作業	20	2022/04/26 23:59	期限已過	C	未通過	查看結果	通過測試名單	
21	課後作業	21	2022/05/04 23:59	繳交	C	通過	查看結果	通過測試名單	
22	課後作業	22	2022/05/04 23:59	繳交	C	通過	查看結果	通過測試名單	
23	課後作業	23	2022/05/04 23:59	繳交	C	通過	查看結果	通過測試名單	

學生端作業繳交畫面

```

001. 一元二次方程式

一元二次方程式， $aX^2 + bx + c = 0$ ，輸入a, b, c, 求 方程式的兩個實根。(測資只有重根的情況)

-----
輸入說明

第一個數(int) a
第二個數(int) b
第三個數(int) c

-----
輸出說明

第一個實根 x1 =  $((-b) + \sqrt{b^2 - 4*a*c}) / (2*a)$ 
第二個實根 x2 =  $((-b) - \sqrt{b^2 - 4*a*c}) / (2*a)$ 

x1, x2 輸出到小數點第一位
printf("%.1f", x1);

-----
Input
1
-2
1

Output
1.0
1.0

-----
Input
1
10
25

Output
-5.0
-5.0

-----
Input

```

學生查看題目



學生刪除作業



刪除作業成功



學生上傳作業頁面



上傳錯誤檔案類型

140.124.181.36 顯示
作業請上傳有效檔名!

作業繳交 留言版 更改密碼

作業上傳：檔名請使用英文，上傳檔案大小請在2M以下

上傳你的主程式： ll.c

上傳副程式(支援上傳多個)： 未選擇任何檔案

上傳檔名有特殊字元

140.124.181.36 顯示
請先上傳作業檔案

作業繳交 留言版 更改密碼

作業上傳：檔名請使用英文，上傳檔案大小請在2M以下

上傳你的主程式： 未選擇任何檔案

上傳副程式(支援上傳多個)： 未選擇任何檔案

沒有上傳檔案

你的程式已上傳至作業編號：91

你上傳的狀態為： **上傳成功**

作業上傳成功

22	課後作業	22	2022/05/04 23:59	繳交	C	通過	查看結果	通過測試名單
----	------	----	------------------	----	---	----	------	--------

作業通過測試

測試編號	測試結果	測試資訊
336	通過測試	ExecuteResult: 1 1 1
339	通過測試	
340	通過測試	
341	通過測試	
343	通過測試	
353	通過測試	
354	通過測試	
355	通過測試	

你的通過率：100%

作業通過所有測試資料

23	課後作業	23	2022/05/04 23:59	繳交	C	未通過	查看結果	通過測試名單
----	------	----	------------------	----	---	-----	------	--------

作業未通過測試

測試編號	測試結果	測試資訊
717	通過測試	ExecuteResult: 1 1 4 1 1 4 1 1 4
718	通過測試	
719	通過測試	
720	通過測試	
721	測試失敗	
722	測試失敗	
723	通過測試	
724	通過測試	
725	通過測試	
726	通過測試	
727	測試失敗	
728	測試失敗	
729	測試失敗	
730	測試失敗	

你的通過率：57%

作業未通過部分測試資料

21	課後作業	21	2022/05/04 23:59	繳交	C	批改中	-	通過測試名單
----	------	----	------------------	----	---	-----	---	--------

作業正在批改

18	課後作業	91	2022/05/05 04:24	繳交	C	編譯失敗	查看結果	通過測試名單
----	------	----	------------------	----	---	------	------	--------

作業編譯失敗

測試編號	測試結果	測試資訊
-1	請勿使用system函數	ExecuteResult:

你的通過率：0%

作業中使用 system 函數

測試編號	測試結果	測試資訊
-1	string k; stdin	ExecuteResult:

你的通過率：0%

作業編譯失敗結果

19	課後作業	999	2023/02/28 23:59	繳交	C	待批改	尚未設定測資	通過測試名單
----	------	-----	------------------	----	---	-----	--------	--------

作業尚未設定測試資料

學號
0
107AI
10859
108AI
11054
11059
11059
11059
11059
11059
11059
11059
11059
11059
11059
11059
11059
11059
11059
11059
11059
110AI

通過該題作業者

歡迎光臨線上課程管理系統

課程選擇 : 108PD01

name :

passwd:

Ok

教師端登入

TestCourse-test_test [加入帳號 加入帳號(從文字檔) 刪除帳號 顯示帳號密碼] [出作業 作業通過統計 作業繳交統計] [編修考題資料庫 出題量 設定考試時間 重新考試 考試成績設定 顯示所有成績] 單頁版 更改密碼 登出

編號	姓名	123	123456	0	test_db1_20220314	0	0	0	0	0	0	編修
1	ttt	0	0	0	102	0	0	0	0	0	0	
2	2	0	0	0	0	0	0	0	0	0	0	

教師端查看成績頁面

Enter your questions

題 號

期 限

相 似 度 (%)

標 題

出題測試

題目內容1

一個班級有N位同學(ID)，M門課，每一門課可以設定評分方式。
 設定M門課評分制，輸入N位同學M門課成績，根據下表換算，計算每一位同學百分制對照平均成績
 印出前三名ID與平均成績，四捨五入到整數。

等第 GPA 百分制區間 百分制對照

A+ 4.3 90-100 95

A 4.0 85-89 87

A- 3.7 80-84 82

B+ 3.3 77-79 78

B 3.0 73-76 75

B- 2.7 70-72 70

C+ 2.3 67-69 68

作業類型：

程式語言：

出題方式：

題目標籤：

☐ Other
 ☐ Array
 ☐ Linked List
 ☐ String
 ☐ Tree
 ☐ Recursion
 ☐ Math
 ☐ Pointer
 ☐ Algorithm
 ☐ Sort

Send

教師端出題頁

編號	種類	題號	繳交期限	繳交	語言	備註	測試	測試結果	匯出	刪除	作業狀態	
1	課後作業	1	2021/10/30 23:59	期限已過	Python	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
2	課後作業	1	2023/10/29 23:59	繳交	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
3	課後作業	1	2022/02/28 23:59	期限已過	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
4	課後作業	10	2023/01/01 23:59	繳交	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
5	課後作業	11	2022/03/28 23:59	期限已過	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
6	課後作業	11	2023/02/28 23:59	繳交	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
7	課後作業	123456	2023/03/19 23:59	準備中	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為公開	不可上傳
8	課後作業	2	2021/11/30 23:59	期限已過	Python	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
9	課後作業	3	2023/12/30 23:59	繳交	Python	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
10	課後作業	4	2021/11/24 23:59	期限已過	Python	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為公開	可上傳
11	課後作業	5	2022/01/30 23:59	期限已過	Python	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為公開	可上傳
12	課後作業	6	2022/01/30 23:59	期限已過	Python	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為公開	可上傳
13	課後作業	7	2022/09/30 23:59	繳交	Python	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
14	課後作業	8	2022/02/28 23:59	期限已過	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
15	課後作業	80	2023/03/19 23:59	準備中	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為公開	不可上傳
16	課後作業	9	2022/02/28 23:59	期限已過	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為公開	可上傳
17	課後作業	90	2022/04/30 02:44	期限已過	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
18	課後作業	91	2022/05/05 04:24	繳交	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳
19	課後作業	999	2023/02/28 23:59	繳交	C	編修	測試資料	測試結果	匯出學生繳交作業	刪除	設為隱藏	可上傳

教師端作業列表頁面

作業類型：課後作業
 題號：2

相似度：100 %
 期限：2021/11/30 23:59

請使用 while loop或for loop

第一個輸入意義為選擇三種圖形：
 1 三角形方尖方面向右邊
 2 三角形方尖方面向左邊
 3 菱形

第二個輸入意義為畫幾行
 (奇數，範圍為 3,5,7,9,...,21)

input
 1 (第一種圖形，三角形尖方面向右邊)
 9 (共 9 行)

output
 *
 **

 **
 *

input
 2 (第二種圖形，三角形尖方面向左邊)
 5 (共 5 行)

內容：
 修改 清除

教師端編修題目頁面

題目91測試資料:

編號	測試資料	預估正確結果	-
68	8000 80	8080	刪除
70	123 456	579	刪除
71	-1 0	-1	刪除
72	10 10	20	刪除
73	20 30	50	刪除
74	20 30	50	刪除
-			新增測試資料

教師端新增測試資料

學號						
0	姓名	通過測試	通過測試	通過測試	通過測試	通過測試
1	ttt	-	-	-	-	-
2	2	-	-	-	-	-
student3	學生3	-	-	-	-	-

教師端查看所有學生測試結果

題數	分類	子分類	內容	答案	編修	刪除
1	0503 test	1	who is infront of 2? (1)1 (2)1.9 (3)1.99 (4)lim2-	1	編修	刪除
2	0503 test	2	who is infront of 3? (1)3.0 (2)2.9 (3)2 (4)lim3-	3	編修	刪除
3	0503 test	3	what is a car? (1)machine (2)animal (3)vehicle (4)human	3	編修	刪除
4	0503 test	4	If you are happy, are you happy? (1)yes (2)no (3)between	1	編修	刪除
5	0503 test	5	what is a dad? (1)parent (2)friend (3)sugardaddy (4)bro	1	編修	刪除

(1) 1

分類:

子分類:

答案:

內容:

教師編輯考題資料庫

0嗨！0答過答案：, 答過0 題, 答對 0 題, 本次共：5, 題
 開放起始時間: 2022/05/03 22:59:00,
 開放結束時間: 2024/05/03 22:59:00
 考試結束時間: 2024/05/03 22:59:00
 現在時間: 2022/05/04 03:19:15

學生參加考試

考試結束時間: 2024/05/03 22:59:00, 現在時間: 2022/05/04 03:19:42

(1)who is infront of 2? (1)1 (2)1.9 (3)1.99 (4)lim2-

- ☒ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ give up

(2)who is infront of 3? (1)3.0 (2)2.9 (3)2 (4)lim3-

- ☐ 1
- ☐ 2
- ☒ 3
- ☐ 4
- ☐ give up

(3)what is a car? (1)machine (2)animal (3)vehicle (4)human

- ☐ 1
- ☐ 2
- ☒ 3
- ☐ 4
- ☐ give up

(4)If you are happy, are you happy? (1)yes (2)no (3)between

- ☒ 1
- ☐ 2
- ☐ 3
- ☐ 4

學生作答頁面

Welcome Online Exam

Hello! 0

You have already done !

您已經考過試了！

Your answer is:13314

5 題中, 你答對 4 題, 共 80分

學生考試結束畫面

選擇	分類	題數
<input checked="" type="checkbox"/>	0503 test	5

Edit

教師建立考試中，選擇考題分類畫面

目前設定考試開始時間 2022/05/03 22:59:00

目前設定考試結束時間 2024/05/03 22:59:00

目前設定本次考試題數 5

考試題數請設定為5的倍數。

設定本次考試題數： 5 ▾

開始開放考試時間： 2022 ▾ 年 5 ▾ 月 4 ▾ 日 3 ▾ 時 22 ▾ 分

結束開放考試時間： 2022 ▾ 年 5 ▾ 月 5 ▾ 日 3 ▾ 時 22 ▾ 分

Ok

教師設定考試時間畫面

5.2 Source Code Listing

HomeworkQuestion.java

```
package componment;

import java.util.*;

public class HomeworkQuestion {

    @Override
    public String toString() {
        return "Homework [type=" + type + ", id=" + id + ", content=" +
description.getContent() + ", deadline=" + deadline
        + ", expired=" + expired + ", weights=" + description.getSimilarity()
+ ", language=" + language + "]";
    }

    private String type;
    private String id;
    private String deadline;
    private String language;
    private HomeworkDescription description;
    private boolean enable;
    private boolean expired;

    private List<TestData> testDatas;
    private List<Homework> homeworks;

    public HomeworkQuestion(String type, String id, String deadline, String lan, String
title, String content, int weights, boolean enable) {
        this.type = type;
        this.id = id;
        this.deadline = deadline;
        this.language = lan;
        description = new HomeworkDescription(weights, title, content);
        this.enable = enable;
        homeworks = new ArrayList<>();
        testDatas = new ArrayList<>();
    }
}
```



```

public HomeworkQuestion(String t, String i, String c, String d, boolean e, int w) {
    type = t;
    id = i;
    description = new HomeworkDescription(w, "", c);
    deadline = d;
    expired = e;
    setSimilarity(w);
    homeworks = new ArrayList<>();
    testDatas = new ArrayList<>();
}

public HomeworkQuestion(String t, String i, String c, String d, boolean e) {
    type = t;
    id = i;
    description = new HomeworkDescription(100, "", c);
    deadline = d;
    expired = e;

    homeworks = new ArrayList<>();
    testDatas = new ArrayList<>();
}

public HomeworkQuestion(String t, String i, String c, String d, boolean e, String l) {
    type = t;
    id = i;
    description = new HomeworkDescription(100, "", c);
    deadline = d;
    expired = e;
    language = l;

    homeworks = new ArrayList<>();
    testDatas = new ArrayList<>();
}

public String getType() {
    return type;
}

public String getId() {

```

```

        return id;
    }

    public String getContent() {
        return this.description.getContent();
    }

    public String getNoBRContent() {
        String temp = description.getContent().replaceAll("<BR>", "");
        return temp;
    }

    public String getDeadline() {
        return deadline;
    }

    public boolean isExpired() {
        return expired;
    }

    public String getlanguage() {
        return language;
    }

    public void setTestDatas(List<TestData> datas) {
        this.testDatas = datas;
    }

    public void setHomeworks(List<Homework> hws) {
        this.homeworks = hws;
    }

    public List<TestData> getTestDatas() {
        return this.testDatas;
    }

    public List<Homework> getHomeworks() {

```

```

        return this.homeworks;
    }

    public void handInHomework(String studentID, String filePath) throws Exception {
        Homework hw = new Homework(studentID, this.id, filePath); // create
        for (int i = 0; i < testDatas.size(); i++) { // correct
            String inputData = testDatas.get(i).getInputData();
            String expectedOutputData = testDatas.get(i).getExpectedOutput();

            correct(hw, inputData, expectedOutputData);
        }
        homeworks.add(hw);
    }

    private void correct(Homework hw, String inputData, String expectedOutputData) {
        String executeReuslt = "";
        TestResult tr = new TestResult(executeReuslt,
executeReuslt.equals(expectedOutputData));
        hw.addResult(tr);
    }

    private Homework findHomework(String studentID) {
        for (int i = 0; i < homeworks.size(); i++) {
            if (studentID.equals(homeworks.get(i).getStudentId())) {
                return homeworks.get(i);
            }
        }
        return null;
    }

    public List<String> getQuestionItem(String studentID) {
        List<String> questionItem = new ArrayList<String>();
        Homework hw = findHomework(studentID);

        // create questionItem
        if (hw != null) {
            questionItem.add(getType());
            questionItem.add(getId());
        }
    }

```

```

        questionItem.add(getDeadline());
        questionItem.add(this.isExpired() ? "期限已過" : "繳交");
        questionItem.add(getlanguage());
        questionItem.add(hw.getPassState() ? "通過測試" : "測試失敗");
        questionItem.add("測試結果");
    }
    else {
        questionItem.add(getType());
        questionItem.add(getId());
        questionItem.add(getDeadline());
        questionItem.add(this.isExpired() ? "期限已過" : "繳交");
        questionItem.add(getlanguage());
        questionItem.add("未繳");
        questionItem.add("-");
    }
    return questionItem;
}

public List<TestResult> checkResult(String studentID) {
    Homework hw = findHomework(studentID);
    if (hw != null) {
        return hw.getResult(studentID);
    }
    return null;
}

public void addHomework(Homework hw) {
    homeworks.add(hw);
}

public boolean setHomework(String dbName) {
    String sql = "UPDATE Message SET content = ?, memo = ?, weights = ?
WHERE type='hw' and id = ?";
    String[] data = new String[4];
    data[0] = description.getContent();
    data[1] = deadline;
    data[2] = String.valueOf(description.getSimilarity());
    data[3] = id;
    String result = DbProxy.setData(dbName, sql, data);
    if (result.equals("OK")) {
        return true;
    }
}

```

```

    }
    return false;
}

public static boolean activeHomework(String dbName, String id, String setActive) {
    String sql = "update Message set active= ? where type = 'hw' and id= ?;";
    String[] data = new String[2];
    data[0] = setActive;
    data[1] = id;
    String result = DbProxy.setData(dbName, sql, data);
    if (result.equals("OK")) {
        return true;
    }
    return false;
}

public static boolean delHomework(String dbName, String id) {
    String sql = "delete from Message where type = 'hw' and id= ?;";
    DbProxy.delData(dbName, sql, id);
    sql = "delete from test_data where question_id= ?;";
    DbProxy.delData(dbName, sql, id);
    sql = "delete from test_result where question_id= ?;";
    DbProxy.delData(dbName, sql, id);
    return true;
}

public static void hwStatistic(String dbName, ArrayList<?> sts, ArrayList<String>
hws, boolean[][] checkList) {
    for (int i = 0; i < 100; i++) {
        for (int j = 0; j < 100; j++) {
            checkList[i][j] = false;
        }
    }
    String temp = "", hwId = "", nameId = "";
    Student st = null;
    int x = 0, y = 0;
    String sql = "SELECT * FROM message WHERE type='hw' ORDER BY id;";
    String[] data = null;

```

```

ArrayList<?> result = DbProxy.selectData(dbName, sql);
for (int i = 0; i < result.size(); i++) {
    data = (String[]) result.get(i);
    temp = data[1];
    hws.add(temp); //hws = c.message all hw_id
}
sql = "SELECT * FROM Homework ORDER BY id;";
result = DbProxy.selectData(dbName, sql); //c.homework all column
for (int k = 0; k < result.size(); k++) {
    data = (String[]) result.get(k);
    nameId = data[0];
    hwId = data[2];
    for (int i = 0; i < sts.size(); i++) {
        st = (Student) sts.get(i);
        if (st.getName().equals(nameId)) {
            x = i;
            break;
        }
    }
    for (int i = 0; i < hws.size(); i++) {
        temp = (String) hws.get(i);
        if (temp.equals(hwId)) {
            // i=c.homework which column
            y = i;
            break;
        }
    }
    if (x >= 100)
        x = 0;
    if (y >= 100)
        y = 0;
    checkList[x][y] = true; // [第幾個 student][第幾個作業]
} // end for
}

public int getSimilarity() {
    return description.getSimilarity();
}

```

<pre> public void setSimilarity(int weights) { this.description.setSimilarity(weights); } } </pre>
TestResult.java
<pre> package bean; public class TestResult { private int id; private String studentID; private int questionID; private int test_data_id; private boolean result; private String description; private String executeResult; private boolean isPass; public TestResult() { } public TestResult(String exeResult, boolean isPass) { this.executeResult = exeResult; this.isPass = isPass; } public String getExecuteResult() { return executeResult; } public void setExecuteResult(String executeResult) { this.executeResult = executeResult; } public int getId() { return id; } public void setId(int id) { this.id = id; } public int getQuestionID() { return questionID; } } </pre>

<pre> public void setQuestionID(int questionID) { this.questionID = questionID; } public boolean isResult() { return result; } public void setResult(boolean result) { this.result = result; } public String getDescription() { return description; } public void setDescription(String description) { this.description = description; } public int getTest_data_id() { return test_data_id; } public void setTest_data_id(int test_data_id) { this.test_data_id = test_data_id; } public boolean isPass() { return isPass; } public void setPass(boolean isPass) { this.isPass = isPass; } public void setStudentID(String studentID) { this.studentID = studentID; } public String getStudentID() { return this.studentID; } } </pre>
TestData.java
<pre> package bean; public class TestData { private int id; </pre>

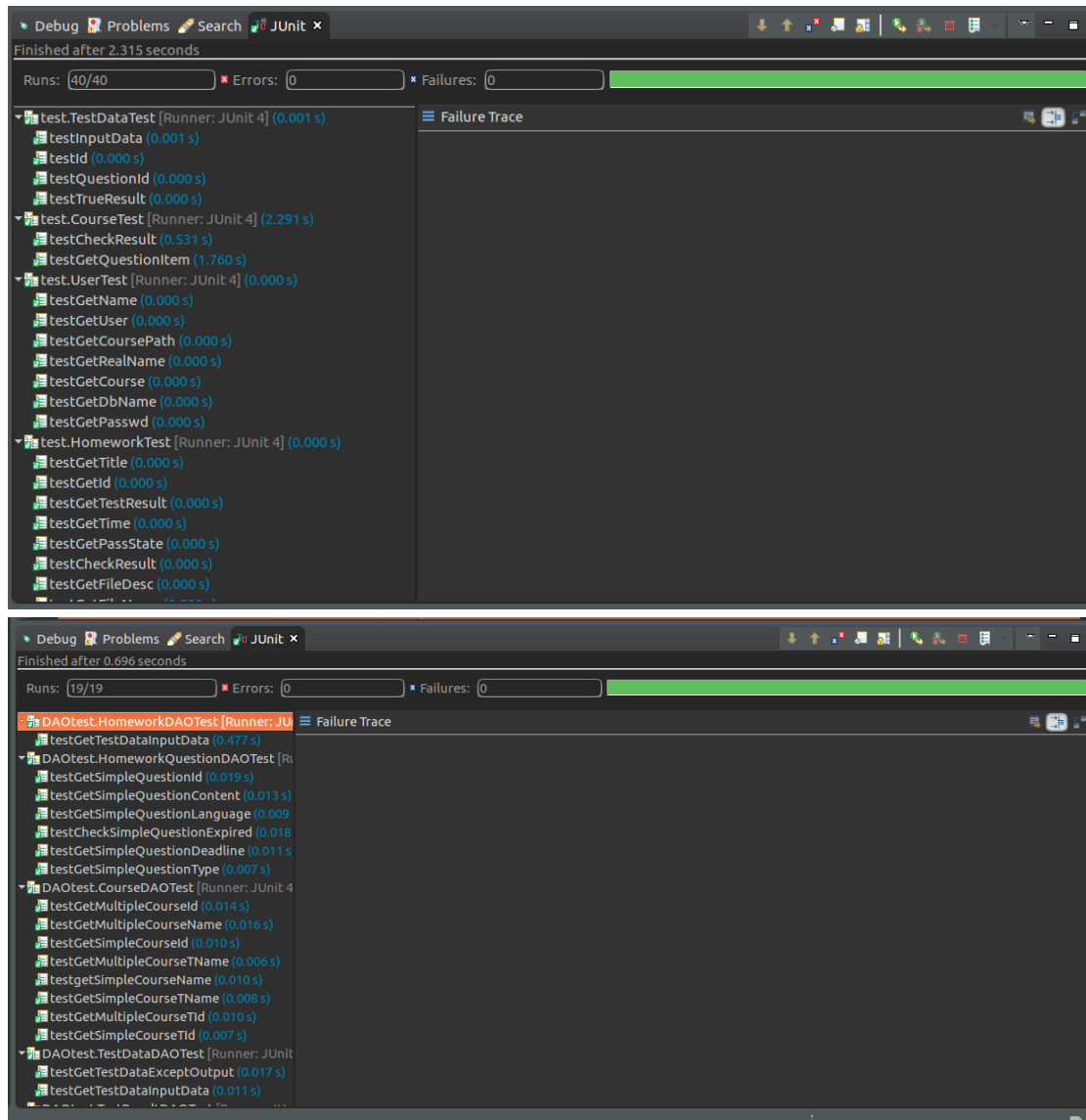

```

private int questionID;
private String input_data;
private String excepted_output;
public TestData() {
}
public TestData(int id, int questionID, String inputData, String exceptedOutput) {
    this.id = id;
    this.questionID = questionID;
    this.input_data = inputData;
    this.excepted_output = exceptedOutput;
}
public int getId() {
    return id;
}
public void setId(int id) {
    this.id = id;
}
public int getQuestionID() {
    return questionID;
}
public void setQuestionID(int questionID) {
    this.questionID = questionID;
}
public String getInputData() {
    return input_data;
}
public void setInput_data(String input_data) {
    this.input_data = input_data;
}
public String getExpectedOutput() {
    return excepted_output;
}
public void setTrue_result(String exceptedOutput) {
    this.excepted_output = exceptedOutput;
}
}

```

6. Unit Testing

6.1 Snapshots of testing result



6.2 Unit Testing Code Listing

CourseDAOTest.java

```
package DAOtest;

import static org.junit.Assert.*;
import org.junit.Before;
import org.junit.Test;
import component.Course;
import dao.CourseDAO;
import java.util.ArrayList;
```

```

public class CourseDAOTest {

    CourseDAO courseDAO;

    @Before
    public void setUp() throws Exception {
        courseDAO = new CourseDAO();
    }

    @Test
    public void testGetSimpleCourseId() {
        Course course = courseDAO.getCourse("OOAD");
        assertEquals("OOAD", course.getId());
    }

    @Test
    public void testgetSimpleCourseName() {
        Course course = courseDAO.getCourse("OOAD");
        assertEquals("OOAD", course.getName());
    }

    @Test
    public void testGetSimpleCourseTId() {
        Course course = courseDAO.getCourse("OOAD");
        assertEquals("sysop", course.getTeacherId());
    }

    @Test
    public void testGetSimpleCourseTName() {
        Course course = courseDAO.getCourse("OOAD");
        assertEquals("sysop", course.getTeacherName());
    }

    @Test
    public void testGetMultipleCourseId() {
        ArrayList<Course> courses = new ArrayList<Course>();
        ArrayList<String> IDs = new ArrayList<String>();
        IDs.add("108PD01");
    }
}

```

```

        IDs.add("108PD02");
        IDs.add("108Python");
        IDs.add("110PD01");
        courses = courseDAO.getAllCourse();
        for(int i=0; i<IDs.size() && i < courses.size(); i++) {
            Course c=courses.get(i);
            assertEquals(IDs.get(i), c.getId());
        }
    }

@Test
public void testGetMultipleCourseName() {
    ArrayList<Course> courses = new ArrayList<Course>();
    ArrayList<String> IDs = new ArrayList<String>();
    IDs.add("108PD01");
    IDs.add("108PD02");
    IDs.add("108Python");
    courses = courseDAO.getAllCourse();
    for(int i=0; i<IDs.size() && i < courses.size(); i++) {
        Course c=courses.get(i);
        assertEquals(IDs.get(i), c.getName());
    }
}

@Test
public void testGetMultipleCourseTId() {
    ArrayList<Course> courses = new ArrayList<Course>();
    ArrayList<String> IDs = new ArrayList<String>();
    IDs.add("sysop");
    IDs.add("sysop");
    IDs.add("sysop");
    courses = courseDAO.getAllCourse();
    for(int i=0; i<IDs.size() && i < courses.size(); i++) {
        Course c=courses.get(i);
        assertEquals(IDs.get(i), c.getTeacherId());
    }
}

```

<pre> @Test public void testGetMultipleCourseTName() { ArrayList<Course> courses = new ArrayList<Course>(); ArrayList<String> IDs = new ArrayList<String>(); IDs.add("sysop"); IDs.add("sysop"); IDs.add("sysop"); courses = courseDAO.getAllCourse(); for(int i=0; i<IDs.size() && i < courses.size(); i++) { Course c=courses.get(i); assertEquals(IDs.get(i), c.getTeacherName()); } } </pre>
<h3>CourseTest.java</h3> <pre> package test; import static org.junit.Assert.*; import java.util.List; import org.junit.Before; import org.junit.Test; import bean.TestResult; import component.Course; import dao.CourseDAO; public class CourseTest { CourseDAO courseDAO; String testCourse; @Before public void setUp() throws Exception { courseDAO = new CourseDAO(); testCourse = "TestCourse"; } @Test public void testCheckResult() { Course course = courseDAO.getCourse(testCourse); </pre>

```

        List<TestResult> tr = course.checkResult("0", 2);

        assertEquals(4, tr.size());
        assertEquals(true, tr.get(0).isPass());
        assertEquals(true, tr.get(1).isPass());
        assertEquals(true, tr.get(2).isPass());
        assertEquals(false, tr.get(3).isPass());
    }

    @Test
    public void testGetQuestionItem() {
        Course course = courseDAO.getCourse(testCourse);
        List<List<String>> questionItems = course.viewHomeworkQuestions("0");

        assertEquals(16, questionItems.size());
        assertEquals("1", questionItems.get(0).get(1));
        assertEquals("1", questionItems.get(1).get(1));
        assertEquals("1", questionItems.get(2).get(1));
        assertEquals("1", questionItems.get(3).get(1));
        assertEquals("10", questionItems.get(4).get(1));
    }
}

```

HomeworkQuestionDAOTest.java

```

package DAOtest;

import static org.junit.Assert.*;
import org.junit.Before;
import org.junit.Test;
import component.HomeworkQuestion;
import dao.HomeworkQuestionDAO;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class HomeworkQuestionDAOTest {

    HomeworkQuestionDAO homeworkQuestionDAO;
}

```

```

@Before
public void setUp() {
    homeworkQuestionDAO = new HomeworkQuestionDAO();
}

@Test
public void testGetSimpleQuestionType() {
    HomeworkQuestion homeworkQuestion =
homeworkQuestionDAO.getQuestion("testcourse", "3");
    assertEquals("hw", homeworkQuestion.getType());
}

@Test
public void testGetSimpleQuestionId() {
    HomeworkQuestion homeworkQuestion =
homeworkQuestionDAO.getQuestion("testcourse", "3");
    assertEquals("3", homeworkQuestion.getId());
}

@Test
public void testGetSimpleQuestionContent() {
    HomeworkQuestion homeworkQuestion =
homeworkQuestionDAO.getQuestion("testcourse", "3");
    assertEquals("3", homeworkQuestion.getContent());
}

@Test
public void testGetSimpleQuestionDeadline() {
    HomeworkQuestion homeworkQuestion =
homeworkQuestionDAO.getQuestion("testcourse", "3");
    assertEquals("2023/12/30 23:59", homeworkQuestion.getDeadline());
}

@Test
public void testCheckSimpleQuestionExpired() throws ParseException {
    HomeworkQuestion homeworkQuestion =
homeworkQuestionDAO.getQuestion("testcourse", "3");

```

<pre> String deadline = homeworkQuestion.getDeadline(); SimpleDateFormat dateFormat = new SimpleDateFormat ("yyyy/MM/dd HH:mm"); Date date1 = dateFormat.parse("2020/01/25 20:00"); Date dl = dateFormat.parse(deadline); assertTrue(date1.before(dl)); } @Test public void testGetSimpleQuestionLanguage() { HomeworkQuestion homeworkQuestion = homeworkQuestionDAO.getQuestion("testcourse", "3"); assertEquals("Python", homeworkQuestion.getlanguage()); } } </pre>
HomeworkQuestionTest.java
<pre> package test; import static org.junit.Assert.*; import java.util.ArrayList; import java.util.List; import org.junit.Before; import org.junit.Test; import bean.TestResult; import component.Homework; import component.HomeworkQuestion; public class HomeworkQuestionTest { HomeworkQuestion question; @Before public void setUp() throws Exception { question = new HomeworkQuestion("hw", "001", "content", "2022/08/08 23:59", false, "C"); } @Test public void testGetType() { </pre>


```

        assertEquals(question.getType(), "hw");
    }

    @Test
    public void testGetId() {
        assertEquals(question.getId(), "001");
    }

    @Test
    public void testGetContent() {
        assertEquals(question.getContent(), "content");
    }

    @Test
    public void testGetDeadline() {
        assertEquals(question.getDeadline(), "2022/08/08 23:59");
    }

    @Test
    public void testIsExpired() {
        assertEquals(question.isExpired(), false);
    }

    @Test
    public void testGetlanguage() {
        assertEquals(question.getlanguage(), "C");
    }

    @Test
    public void testGetWeights() {
        question.setSimilarity(50);
        assertEquals(question.getSimilarity(), 50);
    }

    @Test
    public void testGetWeightsLess0() {
        question.setSimilarity(-50);
        assertEquals(question.getSimilarity(), 0);
    }

```

```

    }

    @Test
    public void testGetWeightsGreater100() {
        question.setSimilarity(150);
        assertEquals(question.getSimilarity(), 100);
    }

    @Test
    public void testGetQuestionItemHandIn() throws Exception {
        Homework hw = new Homework("110598108", "001", "filePath");
        hw.addResult(new TestResult("123", false));

        question.addHomework(hw);
        List<String> result = question.getQuestionItem("110598108");
        assertEquals(result.get(0), "hw");
        assertEquals(result.get(1), "001");
        assertEquals(result.get(2), "2022/08/08 23:59");
        assertEquals(result.get(3), "繳交");
        assertEquals(result.get(4), "C");
        assertEquals(result.get(5), "測試失敗");
        assertEquals(result.get(6), "測試結果");
    }

    @Test
    public void testGetQuestionItemNotHandIn() throws Exception {
        question.addHomework(new Homework("110598108", "001", "filePath"));
        List<String> result = question.getQuestionItem("1105556666");
        assertEquals(result.get(0), "hw");
        assertEquals(result.get(1), "001");
        assertEquals(result.get(2), "2022/08/08 23:59");
        assertEquals(result.get(3), "繳交");
        assertEquals(result.get(4), "C");
        assertEquals(result.get(5), "未繳");
        assertEquals(result.get(6), "-");
    }

    @Test

```

```
public void testCheckResult() throws Exception {  
    Homework hw = new Homework("110598108", "001", "filePath");  
    hw.addResult(new TestResult("123", true));  
    hw.addResult(new TestResult("456", false));  
  
    List<Homework> hws = new ArrayList<Homework>();  
    hws.add(hw);  
    question.setHomeworks(hws);  
    List<TestResult> result = question.checkResult("110598108");  
    assertEquals(result.get(0).getExecuteResult(), "123");  
    assertEquals(result.get(0).isPass(), true);  
    assertEquals(result.get(1).getExecuteResult(), "");  
    assertEquals(result.get(1).isPass(), false);  
}  
}
```

7. Measurement

110598087 謝狄烽			
Homework	Time	Activity	Total
1	2022/02/26 20:15 ~ 21:16	撰寫 Problem statement 初稿	1.5 h
	2022/02/28 21:10 ~ 21:42	補充 Problem statement 更換 Measurement 格式	
2	2022/03/14 18:00 ~ 19:00	修改目錄和段落 調整 use case、features	5.5 h
	2022/03/15 10:20 ~ 12:50	更新 change history previous project information	
	2022/03/15 15:00 ~ 17:00	討論 system context、 draw diagrams	
3	2022/03/25 19:30 ~ 20:00	根據作業批改結果修改 use case 和 diagram	6 h
	2022/03/28 18:00 ~ 20:00	從 Use Case 中篩選名詞 分析 Good Classes 畫 Concept of Domain Model	
	2022/03/29 10:30 ~ 12:30	再次分析 Good Classes 討論 Association 和 Attribute	
	2022/03/29 15:00 ~ 16:30	Domain Model 加上 Association 和 Attribute	
4	2022/04/20 15:00 ~ 17:00	Revised Domain Model	7 h
	2022/04/20 17:00 ~ 18:00	Drawing Logical Architecture	
	2022/04/20 19:00 ~ 23:00	Sequence Diagram with GRASP	
5	2022/04/29 19:00 ~ 20:00	Revised Domain Model	49h
	2022/04/29 21:00 ~ 23:00	Revised SSD	

	2022/04/30 13:00 ~ 18:00	Revised SD – UC02	
	2022/04/30 19:00 ~ 00:00		
	2022/05/01 12:00 ~ 18:00	Revised Design Class Diagram	
	2022/05/01 20:00 ~ 00:00		
	2022/05/02 21:00 ~ 00:00	Implement Code	
	2022/05/03 00:00 ~ 01:00		
	2022/05/03 18:00 ~ 00:00		
	2022/05/04 00:00 ~ 01:00		
	2022/05/04 15:00 ~ 18:00	Draw Implement Class Diagram	
	2022/05/04 19:00 ~ 00:00	Prepare Midiumn Review & PPT	
	2022/05/04 00:00 ~ 08:00		
	6	2022/05/18 18:00 ~ 19:00	
2022/05/19 09:00 ~ 13:00		Revised System Sequence Diagram & Sequence Diagram	
2022/05/24 20:00 ~ 21:00		Revised HW5 according to TA’s feedback	

110598108 郭宗育			
Homework	Time	Activity	Total
1	2022/02/26 20:15 ~ 21:16	撰寫 Problem statement 初稿	1.5 h
	2022/02/28 21:10 ~ 21:42	補充 Problem statement 更換 Measurement 格式	
2	2022/03/09 20:40 ~ 21:10	新增目錄與段落設定	6 h
	2022/03/12 19:30 ~ 20:50	寫 Use Case	
	2022/03/14 19:40 ~ 21:20		
	2022/03/14 18:00 ~ 19:00	調整 use case、features	
	2022/03/15 10:00 ~ 11:30	寫 System Features、Non-Functional Requirements	
3	2022/03/28 18:00 ~ 20:00	從 Use Case 中篩選名詞 分析 Classes 分析 Bad Classes	4.5 h
	2022/03/29 10:30 ~ 12:30	再次分析 Good Classes 討論 Association 和 Attribute	
	2022/03/29 18:15 ~ 18:45	Revise Domain Model	
4	2020/04/20 15:00 ~ 17:00	Revised Domain Model	9 h
	2020/04/20 17:00 ~ 18:00	Drawing System Sequence Diagram	
	2022/04/20 19:00 ~ 23:00	Sequence Diagram with GRASP	
	2022/04/21 01:30 ~ 03:30	Drawing Design Class Model	

5	2022/04/29 16:00 ~ 18:00	Revised Use Case 02 繳交作業	47.5h
	2022/04/29 20:00 ~ 21:30	Revised Use Case 01 新增作業	
	2022/04/29 21:30 ~ 23:00	Revised SSD	
	2022/04/30 15:00~18:00	Revised SD – UC02	
	2022/04/30 19:00~00:00		
	2022/05/01 15:00~18:00	Revised Design Class Diagram	
	2022/05/01 20:00~00:00		
	2022/05/02 00:00~02:00	Implement Code	
	2022/05/02 20:00~00:00		
	2022/05/03 00:00~01:00		
	2022/05/03 18:00~00:00		
	2022/05/04 0:00~02:30	Draw Implement Class Diagram	
	2022/05/04 20:00~23:00		
	2022/05/04 23:00~00:00	Snapshots of system execution	
	2022/05/05 00:00~02:00	Fill the table of difference between implementation class model and design class model	
	2022/05/05 02:00~03:00	Fill the table of Calculate Line of Code	

	2022/05/05 03:00~03:30	List Source Code	
	2022/05/05 04:00~04:30	List Unit Testing Code	
	2022/05/05 04:30~05:00	Snapshots of testing result	
	2022/05/05 05:00~05:30	Revised word	
	2022/05/05 06:00~07:30	Prepare Midiumn PPT	
6	2022/05/19 00:00 ~ 04:00	Revised Implementation Class Diagram & Design class diagram	8 h
	2022/05/19 09:00 ~ 13:00	Revised System Sequence Diagram & Sequence Diagram	

