**Smart course Management**

**System in cloud**

Test Record Document

By

**Mr. Chaichan Suttee 542115016**

**Mr. Tanadol Parn-ong 542115021**

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mrs. Yun Rim Park**

**Document History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Document Name** | **Version** | **Status** | **Date** | **Viewable** | **Reviewer** | **Responsible** |
| **Documents** | | | | | | |
| Smart course management system in the cloud – Test Record\_v0.1.docx | Create Record UTC-1 - 56 | Draft | 26 July 2014 | CS, TP, YP | CS,TP | TP |

**\*CS = Chaichan Suttee**

**\*TP = Tanadol Parn-ong**

**\*YP = Yun Rim Park**

Contents

[**Introduction** 6](#_Toc394585556)

[**The Appendix A** 7](#_Toc394585557)

[**Test case of Unit Test (UTC)** 22](#_Toc394585558)

[**UTC-1 insertStudentInfo(stuId: int, stuUsername: string, stuPassword: string, stuName: string, stuFaculty: string, stuDepartment: string, stuAddress: string, stuEmail: string, stuTel: string): bool** 22](#_Toc394585559)

[**UTC-2 updateStudentInfo(stuId: int, stuPassword: string, stuName: string, stuFaculty: string, stuDepartment: string, stuAddress: string, ,string, stuTel: string) : bool** 23](#_Toc394585560)

[**UTC-3 viewStudentInfo(stuId : int) : Student** 25](#_Toc394585561)

[**UTC-4 viewListStudentFromStudentId(stuId : string) : List<Student>** 26](#_Toc394585562)

[**UTC-5 approveStudentStatus (stuId: string): string** 27](#_Toc394585563)

[**UTC-6 viewStudentNotAvailable (): List<Student>** 28](#_Toc394585564)

[**UTC-7 viewStudenAvailable (): List<Student>** 29](#_Toc394585565)

[**UTC-8 viewStudentByUsernamePassword (stuUsername : string, stuPassword : string): Student** 30](#_Toc394585566)

[**UTC-9** **viewStudentIdByUsername (stuUsername: string): int** 31](#_Toc394585567)

[**UTC-10** **deleteStudent** **(studentId: int): bool** 32](#_Toc394585568)

[**UTC-11** **insertLecturerInfo(lecId : int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool** 33](#_Toc394585569)

[**UTC-12** **updateLecturerInfo(lecId : int, lecPassword : string, lecName : string, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string) : bool** 34](#_Toc394585570)

[**UTC-13** **veiwLecturerInfo(lecId : int) : Lecturer** 36](#_Toc394585571)

[**UTC-14 approveLecturerStatus(lecId : int) : Lecturer** 37](#_Toc394585572)

[**UTC-15** **viewLecturerNotAvailable() : List<Lecturer>** 38](#_Toc394585573)

[**UTC-16** **viewLecturerAvailable() : List<Lecturer>** 39](#_Toc394585574)

[**UTC-17** **viewLecturerByUsernamePassword(lecUsername : string, lecPassword : string) : Lecturer** 41](#_Toc394585575)

[**UTC-18** **viewLecturerByUsername(lecUsername : string) : Lecturer** 42](#_Toc394585576)

[**UTC-19** **deleteLecturer(lecturerId : int) : bool** 43](#_Toc394585577)

[**UTC-20** **viewAdminByUsernamePassword(adminUsername : string, adminPassword : string) : Admin** 44](#_Toc394585578)

[**UTC-21** **viewAdminByUsername(adminUsername : string) : Admin** 45](#_Toc394585579)

[**UTC-22** **viewSemesterInfo(int semId) : Semester** 46](#_Toc394585580)

[**UTC-23** **viewAllSemester() : List<Semester>** 47](#_Toc394585581)

[**UTC-24** **insertSemester(academicYear : string) : bool** 48](#_Toc394585582)

[**UTC-25** **updateSemester(semesterId : int, academicYear : string) : bool** 49](#_Toc394585583)

[**UTC-26** **deleteSemester(semesterId : int) : bool** 50](#_Toc394585584)

[**UTC-27** **viewCourseInfoByLecturerId** **(lecId : int, semesterId : int): List<Course>** 51](#_Toc394585585)

[**UTC-28** **viewCourseInfoByCourseId(course\_id: int): Course** 52](#_Toc394585586)

[**UTC-29** **viewCoursebySemesterId(semId : int): List<Course>** 53](#_Toc394585587)

[**UTC-30 insertCourse(lecturerId: int, semesterId : int, courseName : string, courseCredit : int, courseDescription : string): bool** 54](#_Toc394585588)

[**UTC-31 updateCourse(courseId: int, lecturerId: int, semesterId : int, courseName : string, courseCredit : int, courseDescription : string): bool** 55](#_Toc394585589)

[**UTC-32 deleteCourse(courseId int): bool** 56](#_Toc394585590)

[**UTC-33** **viewPreviousCourse(int lecturerId): bool** 57](#_Toc394585591)

[**UTC-34** **viewStudentInCourse(courseId: int): List<CourseRegistration>** 58](#_Toc394585592)

[**UTC-35**  **viewCourseFromStudent(student\_id: int): List<CourseRegistration>** 60](#_Toc394585593)

[**UTC-36 insertStudentsInCourse** **(courseId: int, studentIdList: List<int>): bool** 62](#_Toc394585594)

[**UTC-37** **deleteStudentInCourse (courseId: int, studentIdList: int): bool** 63](#_Toc394585595)

[**UTC-38** **insertTesting (courseId: int, testing\_name: string, testing\_type: string, testing\_amountQuestion: int, testing\_score: double, testing\_random: int, testing\_posting: int, testing\_student: string, submit\_date: Nullable<DateTime>): bool** 64](#_Toc394585596)

[**UTC-39** **viewPreviousTestingId(courseId: int): int** 65](#_Toc394585597)

[**UTC-40** **viewTestingByTestingId(testingId: int): Testing** 66](#_Toc394585598)

[**UTC-41** **viewListTestingbyCourseId(courseId: int): List<Testing>** 67](#_Toc394585599)

[**UTC-42** **updateTestinginfo(testingId: int, courseId: int, testing\_name: string, testing\_type: string, testing\_amountQuestion: int, testing\_score: double, testing\_random: int, testing\_posting: int, testing\_student: string): bool** 69](#_Toc394585600)

[**UTC-43** **viewListTestingThatPost():List<Testing>** 70](#_Toc394585601)

[**UTC-44** **updateStudentInTesting(testingId int, testing\_student string): bool** 72](#_Toc394585602)

[**UTC-45** **insertQuestion(testing\_id : int, listQuestion : List<Question>) : bool** 73](#_Toc394585603)

[**UTC-46** **viewQuestionbyTestingId(testing : Testing) : List<Question>** 74](#_Toc394585604)

[**UTC-47** **updateQuestioninfo(testing\_id : int, listQuestion : List<Question>) : bool** 75](#_Toc394585605)

[**UTC-48** **viewQuestionByQuestionId(int questionId) : Question** 76](#_Toc394585606)

[**UTC-49** **insertAnswerSheet(studentId : int, testingId : int, answerSheetScore : Nullable<double>, submitTime : DateTime, submit\_result : Nullable<int>, isCheck : int) : bool** 77](#_Toc394585607)

[**UTC-50** **viewPreviousAnswerSheetId(studentId : int) : int** 78](#_Toc394585608)

[**UTC-51** **viewListAnswerSheetBytestingId(testingId : int) : List<AnswerSheet>** 79](#_Toc394585609)

[**UTC-52** **viewAnswerSheetInfo(answerSheetId : int) : AnswerSheet** 81](#_Toc394585610)

[**UTC-53** **updateAnswerSheetScore(answerSheetId : int, score Nullable<double>) : bool** 83](#_Toc394585611)

[**UTC-54** **insertAnswer(questionId : List<int>, answerSheetId : int, answer : List<string>, answerPoint : List<Nullable<double>>) : bool** 84](#_Toc394585612)

[**UTC-55** **viewAnswerListByAnswerSheetId (answerSheetId : int) : List<Answer>** 85](#_Toc394585613)

[**UTC-56** **updateAnswerScore(answerSheetId : int, answerPoint : Nullable<Double>) : bool** 87](#_Toc394585614)

# **Introduction**

This Test Record document is document to describe testing and what methodologies are used in the test record of Smart course Management System in cloud.

**Purpose**

The purpose of Test Record describes record test of unit testing and system test of Smart course Management System in cloud. In this document include test case, unit testing and test data. Smart course Management System in cloud test record consist the description of each function in the system. The Test Record Document was using for software tester to record all the result of the function for identifies the defects and the limitation of the software. From the result of each function can help developer to fix the defects and reducing the cost while the long term of maintenance.

**Scope**

The scope of test record document is testing event or activity of Smart course Management System in cloud. And then find the quality of user requirement and system requirement.

**Objective**

The objectives Smart course Management System in cloud measure the system are:

1. Exceed user requirements.
2. Reduce the rejecting project

**Abbreviation**

UTC-XX Unit test case number

# **The Appendix A**

1. Lecturer

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 1-1.1 | int | lecturer\_id | 1 |
| 1-1.2 | string | lecturer \_username | rimi |
| 1-1.3 | string | lecturer \_password | 123456 |
| 1-1.4 | string | lecturer \_name | rimi park |
| 1-1.5 | string | lecturer \_faculty | CAMT |
| 1-1.6 | string | lecturer \_department | SE |
| 1-1.7 | string | lecturer \_email | SE@gmail.com |
| 1-1.8 | string | lecturer \_tel | 0832224425 |
| 1-1.9 | bit | lecturer \_approvment | True |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 1-2.1 | int | lecturer\_id | 2 |
| 1-2.2 | string | lecturer \_username | tony |
| 1-2.3 | string | lecturer \_password | 123456 |
| 1-2.4 | string | lecturer \_name | tony strack |
| 1-2.5 | string | lecturer \_faculty | CAMT |
| 1-2.6 | string | lecturer \_department | SE |
| 1-2.7 | string | lecturer \_email | [tony@vr.camt.info](mailto:tony@vr.camt.info) |
| 1-2.8 | string | lecturer \_tel | 0893321101 |
| 1-2.9 | bit | lecturer \_approvment | False |

1. Student

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 2-1.1 | int | student \_id | 542115091 |
| 2-1.2 | string | student \_username | john |
| 2-1.3 | string | student \_password | 123456 |
| 2-1.4 | string | student \_name | john conner |
| 2-1.5 | string | student \_faculty | CAMT |
| 2-1.6 | string | student \_department | ANI |
| 2-1.7 | String | student\_address | Thailand |
| 2-1.8 | string | student \_email | ANI@gmail.com |
| 2-1.9 | string | student \_tel | 0833201787 |
| 2-1.10 | bit | student \_approvment | True |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 2-2.1 | int | student \_id | 542115092 |
| 2-2.2 | string | student \_username | jame |
| 2-2.3 | string | student \_password | 123456 |
| 2-2.4 | string | student \_name | john conner |
| 2-2.5 | string | student \_faculty | CAMT |
| 2-2.6 | string | student \_department | SE |
| 2-2.7 | String | student\_address | Thailand |
| 2-2.8 | string | student \_email | SE@gmail.com |
| 2-2.9 | string | student \_tel | 0821330433 |
| 2-2.10 | bit | student \_approvment | True |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 2-3.1 | int | student \_id | 542115093 |
| 2-3.2 | string | student \_username | risa |
| 2-3.3 | string | student \_password | 123456 |
| 2-3.4 | string | student \_name | risa eiei |
| 2-3.5 | string | student \_faculty | CAMT |
| 2-3.6 | string | student \_department | MMIT |
| 2-3.7 | String | student\_address | Thailand |
| 2-3.8 | string | student \_email | MMIT@gmail.com |
| 2-3.9 | string | student \_tel | 0821332333 |
| 2-3.10 | bit | student \_approvment | True |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 2-4.1 | int | student \_id | 542115094 |
| 2-4.2 | string | student \_username | elrond |
| 2-4.3 | string | student \_password | 123456 |
| 2-4.4 | string | student \_name | elrond rivendell |
| 2-4.5 | string | student \_faculty | CAMT |
| 2-4.6 | string | student \_department | SE |
| 2-4.7 | String | student\_address | Thailand |
| 2-4.8 | string | student \_email | SE@gmail.com |
| 2-4.9 | string | student \_tel | 0823149555 |
| 2-4.10 | bit | student \_approvment | True |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 2-5.1 | int | student \_id | 542115095 |
| 2-5.2 | string | student \_username | ploy |
| 2-5.3 | string | student \_password | 123456 |
| 2-5.4 | string | student \_name | ploy sress |
| 2-5.5 | string | student \_faculty | CAMT |
| 2-5.6 | string | student \_department | SE |
| 2-5.7 | String | student\_address | Thailand |
| 2-5.8 | string | student \_email | SE@gmail.com |
| 2-5.9 | string | student \_tel | 0863225885 |
| 2-5.10 | bit | student \_approvment | True |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 2-6.1 | int | student \_id | 542115096 |
| 2-6.2 | string | student \_username | studentone |
| 2-6.3 | string | student \_password | 123456 |
| 2-6.4 | string | student \_name | student one |
| 2-6.5 | string | student \_faculty | CAMT |
| 2-6.6 | string | student \_department | SE |
| 2-6.7 | String | student\_address | Chiang mai |
| 2-6.8 | string | student \_email | SE@gmail.com |
| 2-6.9 | string | student \_tel | 0832335899 |
| 2-6.10 | bit | student \_approvment | False |

Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 3-1.1 | int | admin \_id | 1 |
| 3-1.2 | string | admin \_username | admin |
| 3-1.3 | string | admin \_password | 123456 |
| 3-1.4 | string | adnin \_name | admin camt |

Semester

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 4-1.1 | int | semester\_id | 1 |
| 4-1.2 | string | academic\_year | 1/2556 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 4-2.1 | int | semester\_id | 2 |
| 4-2.2 | string | academic\_year | 2/2556 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 4-3.1 | int | semester\_id | 3 |
| 4-3.2 | string | academic\_year | 1/2557 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 4-4.1 | int | semester\_id | 4 |
| 4-4.2 | string | academic\_year | 2/2557 |

Course

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 5-1.1 | int | course\_id | 1 |
| 5-1.2 | int | lecturer\_id | 1 |
| 5-1.3 | int | semester\_id | 1 |
| 5-1.4 | string | course\_name | Math |
| 5-1.5 | int | course\_credit | 3 |
| 5-1.6 | string | course\_description | Math is easy |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 5-2.1 | int | course\_id | 2 |
| 5-2.2 | int | lecturer\_id | 1 |
| 5-2.3 | int | semester\_id | 1 |
| 5-2.4 | string | course\_name | English |
| 5-2.5 | int | course\_credit | 3 |
| 5-2.6 | string | course\_description | English is easy |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 5-3.1 | int | course\_id | 5 |
| 5-3.2 | int | lecturer\_id | 1 |
| 5-3.3 | Int | semester\_id | 2 |
| 5-3.4 | string | course\_name | Computer |
| 5-3.5 | Int | course\_credit | 3 |
| 5-3.6 | string | course\_description | Computer is easy |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 5-4.1 | int | course\_id | 9 |
| 5-4.2 | int | lecturer\_id | 2 |
| 5-4.3 | Int | semester\_id | 4 |
| 5-4.4 | string | course\_name | Test |
| 5-4.5 | Int | course\_credit | 3 |
| 5-4.6 | string | course\_description | Test na aja |

CourseRegistration

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 6-1.1 | int | courseRegistration\_id | 1 |
| 6-1.2 | int | course\_id | 5 |
| 6-1.3 | int | semester\_id | 542115091 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 6-2.1 | int | courseRegistration\_id | 2 |
| 6-2.2 | int | course\_id | 5 |
| 6-2.3 | int | semester\_id | 542115092 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 6-3.1 | int | courseRegistration\_id | 6 |
| 6-3.2 | int | course\_id | 5 |
| 6-3.3 | int | semester\_id | 542115094 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 6-7.1 | int | courseRegistration\_id | 7 |
| 6-7.2 | int | course\_id | 5 |
| 6-7.3 | int | semester\_id | 542115093 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 6-8.1 | int | courseRegistration\_id | 12 |
| 6-8.2 | int | course\_id | 9 |
| 6-8.3 | int | semester\_id | 542115091 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 6-9.1 | int | courseRegistration\_id | 13 |
| 6-9.2 | int | course\_id | 9 |
| 6-9.3 | int | semester\_id | 542115092 |

Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 7-1.1 | int | testing\_id | 1 |
| 7-1.2 | int | course\_id | 5 |
| 7-1.3 | string | testing\_name | Math Imtro |
| 7-1.4 | String | testing\_type | 1 |
| 7-1.5 | Int | testing\_amountQuestion | 2 |
| 7-1.6 | float | testing\_score | 6 |
| 7-1.7 | bit | testing\_random | 1 |
| 7-1.8 | bit | testing\_bit | 0 |
| 7-1.9 | dateTime | datetime | 2014-10-07 10:20:00 |
| 7-1.10 | string | string | 542115091,542115094,542115093 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 7-2.1 | int | testing\_id | 2 |
| 7-2.2 | int | course\_id | 5 |
| 7-2.3 | string | testing\_name | Math Imtro2 |
| 7-2.4 | String | testing\_type | 1 |
| 7-2.5 | Int | testing\_amountQuestion | 2 |
| 7-2.6 | float | testing\_score | 6 |
| 7-2.7 | bit | testing\_random | 1 |
| 7-2.8 | bit | testing\_bit | 0 |
| 7-2.9 | dateTime | datetime | 2014-10-07 10:11:00 |
| 7-2.10 | string | string | 542115091,542115094,542115093 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 7-3.1 | int | testing\_id | 3 |
| 7-3.2 | int | course\_id | 5 |
| 7-3.3 | string | testing\_name | Math Imtro2 |
| 7-3.4 | String | testing\_type | 1 |
| 7-3.5 | Int | testing\_amountQuestion | 2 |
| 7-3.6 | float | testing\_score | 6 |
| 7-3.7 | bit | testing\_random | 1 |
| 7-3.8 | bit | testing\_bit | 0 |
| 7-3.9 | dateTime | datetime | 2014-10-07 10:11:00 |
| 7-3.10 | string | string | 542115091,542115093 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 7-4.1 | int | testing\_id | 4 |
| 7-4.2 | int | course\_id | 5 |
| 7-4.3 | string | testing\_name | Math Imtro |
| 7-4.4 | String | testing\_type | 1 |
| 7-4.5 | Int | testing\_amountQuestion | 1 |
| 7-4.6 | float | testing\_score | 63 |
| 7-4.7 | bit | testing\_random | 1 |
| 7-4.8 | bit | testing\_bit | 0 |
| 7-4.9 | dateTime | datetime | 2014-02-07 20:15:00 |
| 7-4.10 | string | string | 542115091,542115092,542115093 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 7-5.1 | int | testing\_id | 8 |
| 7-5.2 | int | course\_id | 5 |
| 7-5.3 | string | testing\_name | C# language |
| 7-5.4 | String | testing\_type | 2 |
| 7-5.5 | Int | testing\_amountQuestion | 1 |
| 7-5.6 | float | testing\_score | 3 |
| 7-5.7 | bit | testing\_random | 1 |
| 7-5.8 | bit | testing\_bit | 0 |
| 7-5.9 | dateTime | datetime | 2014-07-20 10:30:00 |
| 7-5.10 | string | string | 542115091,542115094,542115093 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 7-6.1 | int | testing\_id | 9 |
| 7-6.2 | int | course\_id | 5 |
| 7-6.3 | string | testing\_name | Java language |
| 7-6.4 | String | testing\_type | 1 |
| 7-6.5 | Int | testing\_amountQuestion | 1 |
| 7-6.6 | float | testing\_score | 3 |
| 7-6.7 | bit | testing\_random | 1 |
| 7-6.8 | bit | testing\_bit | 0 |
| 7-6.9 | dateTime | datetime | NULL |
| 7-6.10 | string | string | 542115091,542115092,542115094 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 7-7.1 | int | testing\_id | 11 |
| 7-7.2 | int | course\_id | 9 |
| 7-7.3 | string | testing\_name | Test |
| 7-7.4 | String | testing\_type | 2 |
| 7-7.5 | Int | testing\_amountQuestion | 1 |
| 7-7.6 | float | testing\_score | 3 |
| 7-7.7 | bit | testing\_random | 1 |
| 7-7.8 | bit | testing\_bit | 0 |
| 7-7.9 | dateTime | datetime | NULL |
| 7-7.10 | string | string | 542115091,542115092 |

Question

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 8-1.1 | int | question\_id | 1 |
| 8-1.2 | int | testing\_id | 4 |
| 8-1.3 | string | question\_description | 1+1 = ? |
| 8-1.4 | string | question\_choice | 3#$4#$2#$1 |
| 8-1.5 | string | question\_solution | 3 |
| 8-1.6 | float | question\_point | 3 |
| 8-1.7 | string | question\_type | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 8-2.1 | int | question\_id | 2 |
| 8-2.2 | int | testing\_id | 4 |
| 8-2.3 | string | question\_description | 1-1=0? |
| 8-2.4 | string | question\_choice | True#$False |
| 8-2.5 | string | question\_solution | 1 |
| 8-2.6 | float | question\_point | 3 |
| 8-2.7 | string | question\_type | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 8-3.1 | int | question\_id | 3 |
| 8-3.2 | int | testing\_id | 4 |
| 8-3.3 | string | question\_description | 2+1=? |
| 8-3.4 | string | question\_choice | Null |
| 8-3.5 | string | question\_solution | Null |
| 8-3.6 | float | question\_point | 3 |
| 8-3.7 | string | question\_type | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 8-4.1 | int | question\_id | 12 |
| 8-4.2 | int | testing\_id | 8 |
| 8-4.3 | string | question\_description | string = "Hello world"; |
| 8-4.4 | string | question\_choice | True#$False |
| 8-4.5 | string | question\_solution | 1 |
| 8-4.6 | float | question\_point | 3 |
| 8-4.7 | string | question\_type | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 8-5.1 | int | question\_id | 13 |
| 8-5.2 | int | testing\_id | 9 |
| 8-5.3 | string | question\_description | var a = 1; |
| 8-5.4 | string | question\_choice | True#$False |
| 8-5.5 | string | question\_solution | 2 |
| 8-5.6 | float | question\_point | 3 |
| 8-5.7 | string | question\_type | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 8-6.1 | int | question\_id | 14 |
| 8-6.2 | int | testing\_id | 11 |
| 8-6.3 | string | question\_description | Test eiei |
| 8-6.4 | string | question\_choice | True#$False |
| 8-6.5 | string | question\_solution | 1 |
| 8-6.6 | float | question\_point | 3 |
| 8-6.7 | string | question\_type | 2 |

AnswerSheet

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 9-1.1 | int | answerSheet\_id | 20 |
| 9-1.2 | int | student\_id | 542115092 |
| 9-1.3 | int | testing\_id | 4 |
| 9-1.4 | float | answerSheet\_score | 3 |
| 9-1.5 | datetime | answerSheet\_time | 2014-07-20 20:29:35 |
| 9-1.6 | bit | answerSheet\_submitResult | 0 |
| 9-1.7 | bit | answeerSheet\_isChecked | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 9-2.1 | int | answerSheet\_id | 21 |
| 9.2.2 | int | student\_id | 542115091 |
| 9-2.3 | int | testing\_id | 8 |
| 9-2.4 | float | answerSheet\_score | 3 |
| 9-2.5 | datetime | answerSheet\_time | 2014-07-20 20:40:07 |
| 9-2.6 | bit | answerSheet\_submitResult | 1 |
| 9-2.7 | bit | answeerSheet\_isChecked | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 9.3.1 | int | answerSheet\_id | 22 |
| 9.3.2 | int | student\_id | 542115091 |
| 9-3.3 | int | testing\_id | 4 |
| 9-3.4 | float | answerSheet\_score | 3 |
| 9-3.5 | datetime | answerSheet\_time | 2014-07-20 22:39:33 |
| 9-3.6 | bit | answerSheet\_submitResult | 0 |
| 9-3.7 | bit | answeerSheet\_isChecked | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 9-4.1 | int | answerSheet\_id | 23 |
| 9-4.2 | int | student\_id | 542115091 |
| 9-4.3 | int | testing\_id | 4 |
| 9-4.4 | float | answerSheet\_score | 3 |
| 9-4.5 | datetime | answerSheet\_time | 2014-07-26 21:43:10 |
| 9-4.6 | bit | answerSheet\_submitResult | 0 |
| 9-4.7 | bit | answeerSheet\_isChecked | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 9-5.1 | int | answerSheet\_id | 24 |
| 9-5.2 | int | student\_id | 542115093 |
| 9-5.3 | int | testing\_id | 4 |
| 9-5.4 | float | answerSheet\_score | 0 |
| 9-5.5 | datetime | answerSheet\_time | 2014-07-26 21:55:46 |
| 9-5.6 | bit | answerSheet\_submitResult | 0 |
| 9-5.7 | bit | answeerSheet\_isChecked | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 9-6.1 | int | answerSheet\_id | 25 |
| 9-6.2 | int | student\_id | 542115091 |
| 9-6.3 | int | testing\_id | 11 |
| 9-6.4 | float | answerSheet\_score | 3 |
| 9-6.5 | datetime | answerSheet\_time | 2014-07-28 15:04:15 |
| 9-6.6 | bit | answerSheet\_submitResult | 0 |
| 9-6.7 | bit | answeerSheet\_isChecked | 1 |

Answer

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data type** | **Data name** | **Data value** |
| 9.6.1 | int | answer\_id | 39 |
| 9.6.2 | int | question\_id | 14 |
| 9-6.3 | int | answerSheet\_id | 25 |
| 9-6.4 | string | answer | 1 |
| 9-6.5 | float | answer\_point | 3 |

# **Test case of Unit Test (UTC)**

## **UTC-1 insertStudentInfo(stuId: int, stuUsername: string, stuPassword: string, stuName: string, stuFaculty: string, stuDepartment: string, stuAddress: string, stuEmail: string, stuTel: string): bool**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-1:** insertStudentInfo(stuId: int, stuUsername: string, stuPassword: string, stuName: string, stuFaculty: string, stuDepartment: string, stuAddress: string, stuEmail: string, stuTel: string): bool

**Description :** The test case is used for testing insertStudentInfo method in StudentRepository. This method is used for inserting a student account to the database.

**Prerequisites or Test Data**

**:** None

**Expected result**

True, False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 1.1 | Test if student information provided is added into the database successfully. Test for a successful case. | studentId = 542115021  studentUsername = "tanadol"  studentPassword = "123456"  studentName = "Mr.Tanadol Parn-ong"  studentFaculty = "CAMT"  studentDepartment = "Software engineering"  studentAddress = "Chiang mai, Thailand"  studentEmail = [se542115021@vr.camt.info](mailto:se542115021@vr.camt.info)  studentTel = "0833201787" | Assert.True(actual); | True | True | Pass |
| 1.2 | Test if student information provided is added into the database successfully. Test for an unsuccessful case: providing duplicate studentId in the database. | studentId = 542115021  studentUsername = "tanadol"  studentPassword = "123456"  studentName = "Mr.Tanadol Parn-ong"  studentFaculty = "CAMT"  studentDepartment = "Software engineering"  studentAddress = "Chiang mai, Thailand"  studentEmail = [se542115021@vr.camt.info](mailto:se542115021@vr.camt.info)  studentTel = "0833201787" | Assert.False(actual); | False | False | Pass |

## **UTC-2 updateStudentInfo(stuId: int, stuPassword: string, stuName: string, stuFaculty: string, stuDepartment: string, stuAddress: string, ,string, stuTel: string) : bool**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-2:** updateStudentInfo(stuId: int, stuPassword: string, stuName: string, stuFaculty: string, stuDepartment: string, stuAddress: string, ,string, stuTel: string) : bool

**Description :** The test case is used for testing updateStudentInfo method in StudentRepository. This method is used for updating student information into the database.

**Prerequisites or Test Data**

Appendix A: 2-1

**Expected result**

True, False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 2.1 | Test if student information provided is updated into the database successfully. Test for a successful case. | studentId = 542115091  studentUsername = "john"  studentPassword = "123456"  studentName = "john conner"  studentFaculty = "CAMT"  studentDepartment = "ANI"  studentAddress = "Thailand"  studentEmail = ANI@gmail.com  studentTel = "0833201787" | Assert.True (actual); | True | True | Pass |
| 2.2 | Test if student information provided is updated into the database successfully. Test for an unsuccessful case: providing studentId that do not have in the database. | studentId = 9999999  studentUsername = "john"  studentPassword = "123456"  studentName = "john conner"  studentFaculty = "CAMT"  studentDepartment = "ANI"  studentAddress = "Thailand"  studentEmail = ANI@gmail.com  studentTel = "0833201787"" | Assert.False(actual); | False | False | Pass |

## **UTC-3 viewStudentInfo(stuId : int) : Student**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-3:** viewStudentInfo(stuId : int) : Student

**Description :** The test case is used for testing viewStudentInfo method in StudentRepository. This method is used for viewing student information into the database.

**Prerequisites or Test Data**

Appendix A: 2-6

**Expected result**

Student expected1 = new Student { student\_id = 542115096, student\_name = "student one", student\_username = "studentone", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Chiang mai", student\_email = "SE@gmail.com", student\_tel = "0832335899", student\_approvement = 0 };

Student expected2 = new Student();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 3.1 | This case tests to get the student object information from the database, which is contained in the database | student\_id = 542115096 | Assert.AreEqual(expected1 , actual); | expected1 | expected1 | Pass |
| 3.2 | This case tests to get the student information from the database, which is not contained in the database. | student\_id = 111111 | Assert.AreEqual(expected2 , actual); | expected2 | expected2 | Pass |

## **UTC-4 viewListStudentFromStudentId(stuId : string) : List<Student>**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-4:** viewListStudentFromStudentId(stuId : string) : List<Student>

**Description:** The test case is used for testing viewListStudentFromStudentId method in StudentRepository. This method is used for viewing a list of student information into the database.

**Prerequisites or Test Data**

Appendix A: 2-5, 2-6

**Expected result**

Student student1 = new Student { student\_id = 542115095, student\_name = "ploy sree", student\_username = "ploy", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0863225885", student\_approvement = 1 };

Student student2 = new Student { student\_id = 542115096, student\_name = "student one", student\_username = "studentone", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Chiang mai", student\_email = "SE@gmail.com", student\_tel = "0832335899", student\_approvement = 0 };

List<Student> studListExpected1 = new List<Student> { student1, student2 };

List<Student> studListExpected2 = new List<Student>();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 4.1 | This case tests to get the list of student object information from the database, which is contained in the database | studentId = "542115095,542115096"; | CollectionAssert.AreEqual(studentListExpected1, actual); | studentListExpected1 | studentListExpected1 | Pass |
| 4.2 | This case tests to get the list of student information from the database, which is not contained in the database. | studentId = null; | CollectionAssert.AreEqual(studentListExpected2, actual); | studentListExpected2 | studentListExpected2 | Pass |

## **UTC-5 approveStudentStatus (stuId: string): string**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-5:** approveStudentStatus (stuId: string): string

**Description:** The test case is used for testing approveStudentStatus method in StudentRepository. This method is used for approving student\_approvement information into the database.

**Prerequisites or Test Data**

Appendix A: 2-5,2-6

**Expected result**

string studentIdExpected = “542115095”

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 5.1 | This case tests for updating student approvement in the database. (Successful) | studentId = 542115095 | Assert.AreEqual(studentIdExpected, actual); | studentIdExpected | studentIdExpected | Pass |

## **UTC-6 viewStudentNotAvailable (): List<Student>**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-6:** viewStudentNotAvailable (): List<Student>

**Description:** The test case is used for testing viewStudentNotAvailable method in StudentRepository. This method is used for viewing a list of student information that have approvement\_status equal false from the database.

**Prerequisites or Test Data**

Appendix A: 2-6

**Expected result**

Student student1 = new Student { student\_id = 542115096, student\_name = "student one", student\_username = "studentone", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Chiang mai", student\_email = "SE@gmail.com", student\_tel = "0832335899", student\_approvement = 0 };

List<Student> studListExpected = new List<Student> { student1 };

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 6.1 | This case tests to get the list of student object information from the database, which student approval equal false. | None | CollectionAssert.AreEqual(studListExpected, actual); | studListExpected | studListExpected | Pass |

## **UTC-7 viewStudenAvailable (): List<Student>**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-7:** viewStudentAvailable (): List<Student>

**Description:** The test case is used for testing viewStudentNotAvailable method in StudentRepository. This method is used for viewing a list of student information that have approvement\_status equal true from the database.

**Prerequisites or Test Data**

Appendix A: 2-1, 2-2, 2-3, 2-4, 2-5

**Expected result**

Student student2 = new Student { student\_id = 542115091, student\_name = "john conner", student\_username = "john", student\_password = "123456", student\_faculty = "CAMT", student\_department = "ANI", student\_address = "Thailand", student\_email = "ANI@gmail.com", student\_tel = "0833201787", student\_approvement = 1 };

Student student3 = new Student { student\_id = 542115092, student\_name = "jame conner", student\_username = "jame", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0821330433", student\_approvement = 1 };

Student student4 = new Student { student\_id = 542115093, student\_name = "risa eiei", student\_username = "risa", student\_password = "123456", student\_faculty = "CAMT", student\_department = "MMIT", student\_address = "Thailand", student\_email = "MMIT@gmail.com", student\_tel = "0821332333", student\_approvement = 1 };

Student student5 = new Student { student\_id = 542115094, student\_name = "elrond rivendell", student\_username = "elrond", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0823149555", student\_approvement = 1 };

Student student6 = new Student { student\_id = 542115095, student\_name = "ploy sree", student\_username = "ploy", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0863225885", student\_approvement = 1 };

List<Student> studListExpected = new List<Student> { student2, student3, student5, student6

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 7.1 | This case tests to get the list of student object information from the database, which student approval equal true. | None | CollectionAssert.AreEqual(studListExpected, actual); | studListExpected | studListExpected | Pass |

## **UTC-8 viewStudentByUsernamePassword (stuUsername : string, stuPassword : string): Student**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-8:** viewStudentByUsernamePassword (stuUsername : string, stuPassword : string): Student

**Description:** The test case is used for testing viewStudentByUsernamePassword method in StudentRepository. This method is used for viewing student information from the database by using student username and student password for searching.

**Prerequisites or Test Data**

Appendix A: 2-1

**Expected result**

Student studentExpected1 = new Student { student\_id = 542115091, student\_name = "john conner", student\_username = "john", student\_password = "123456", student\_faculty = "CAMT", student\_department = "ANI", student\_address = "Thailand", student\_email = "ANI@gmail.com", student\_tel = "0833201787", student\_approvement = 1 };

Student studentExpected2 = new Student();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 8.1 | This case tests to get the student object information from the database, which student\_username and student\_password are existed in. | student\_username = "john"  student\_password = "123456" | Assert.AreEqual(studentExpected1, actual); | studentExpected1 | studentExpected1 | Pass |
| 8.2 | This case tests to get the student object information from the database, which student\_username and student\_password are not existed in. | student\_username = “johny”  student\_password = "123456" | Assert.AreEqual(studentExpected2, actual); | studentExpected2 | studentExpected2 | Pass |

## **UTC-9** **viewStudentIdByUsername (stuUsername: string): int**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-9:** viewStudentIdByUsername (stuUsername: string): int

**Description:** The test case is used for testing viewStudentIdByUsername method in StudentRepository. This method is used for viewing student id from the database by using student username for searching.

**Prerequisites or Test Data**

Appendix A: 2-1

**Expected result**

int studentIdExpected1 = 542115091;

int studentIdExpected2 = 0;

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 9.1 | This case tests to get the student id from the database, which student\_username and student\_password are existed in. | student\_username = "john" | Assert.AreEqual(studentExpected1, actual); | studentIdExpected1 | studentIdExpected1 | Pass |
| 9.2 | This case tests to get the student id from the database, which student\_username and student\_password are not existed in. | student\_username = “johny” | Assert.AreEqual(studentExpected2, actual); | studentIdExpected2 | studentIdExpected2 | Pass |

## **UTC-10** **deleteStudent** **(studentId: int): bool**

**Package:** Repository

**Class:** StudentRepository

**Unit Test Case-10:** deleteStudent (studentId: int): bool

**Description:** The test case is used for testing deleteStudent method in StudentRepository. This method is used for deleting student information in the database.

**Prerequisites or Test Data**

Appendix A: 2-5

**Expected result**

True/False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 10.1 | This case tests to delete the student information in the database. | student\_id = 542115095 | Assert.True(actual); | True | True | Pass |

## **UTC-****11** **insertLecturerInfo(lecId : int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Package:** Repository

**Class:** LecturerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool

**Description:** The test case is used for testing **11** method in LecturerRepository. This method is used for [xxxx] information in the database.

**Prerequisites or Test Data**

: None

**Expected result**

True/False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 11.1 | Test if Lecturer information provided is added into the database successfully. Test for a successful case. | lecturerId = 100  lectuerUsername = "teacherone"  lectuerPassword = "123456"  lectuerName = "teacher one"  lectuerFaculty = "CAMT"  lectuerDepartment = "SE"  lectuerEmail = "SE@vr.camt.info"  lectuerTel = "0833201787" | Assert.True(actual); | True | True | Pass |
| 11.2 | Test if Lecturer information provided is added into the database successfully. Test for an unsuccessful case: providing duplicate studentId in the database. | lecturerId = 100  lectuerUsername = null  lectuerPassword = "123456"  lectuerName = "teacher one"  lectuerFaculty = "CAMT"  lectuerDepartment = "SE"  lectuerEmail = "SE@vr.camt.info"  lectuerTel = "0833201787" | Assert.IsFalse(actual) | False | False | Pass |

## **UTC-12** **updateLecturerInfo(lecId : int, lecPassword : string, lecName : string, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string) : bool**

**Package:** Repository

**Class:** LecturerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : **string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is used for update Lecturer information in the database.

**Prerequisites or Test Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data name** | **Data type** | **Data value** |
| 1.1 | int | lecturerId | 100 |
| 1.2 | string | lectuerPassword | 123456 |
| 1.3 | string | lectuerName | teacher one |
| 1.4 | string | lectuerFaculty | CAMT |
| 1.5 | string | lectuerDepartment | SE |
| 1.6 | string | lectuerEmail | SE@vr.camt.info |
| 1.7 | string | lectuerTel | 0833201787 |

**Expected result**

True/False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 12.1 | Test if lecturer information provided is updated into the database successfully. Test for a successful case. | int lecturerId = 100;  string lectuerPassword = "123456";  string lectuerName = "teacher one";  string lectuerFaculty = "CAMT";  string lectuerDepartment = "SE";  string lectuerEmail = "SE@vr.camt.info";  string lectuerTel = "0833201787"; | Assert.True (actual); | True | True | Pass |
| 12.2 | Test if lecturer information provided is updated into the database successfully. Test for an unsuccessful case: providing studentId that do not have in the database. | int lecturerId = 100;  string lectuerPassword = null;  string lectuerName = "teacher one";  string lectuerFaculty = "CAMT";  string lectuerDepartment = "SE";  string lectuerEmail = "SE@vr.camt.info";  string lectuerTel = "0833201787"; | Assert.False(actual); | False | False | Pass |

## **UTC-13** **veiwLecturerInfo(lecId : int) : Lecturer**

**Package:** Repository

**Class:** LecturerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is used for view Lecturer information in the database.

**Prerequisites or Test Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data name** | **Data type** | **Data value** |
| 1.1 | int | lecturerId | 100 |
| 1.2 | string | lectuerUsername | teacherone |
| 1.3 | string | lectuerPassword | 123456 |
| 1.4 | string | lectuerName | teacher one |
| 1.5 | string | lectuerFaculty | CAMT |
| 1.6 | string | lectuerDepartment | SE |
| 1.7 | string | lectuerEmail | SE@vr.camt.info |
| 1.8 | string | lectuerTel | 0833201787 |

**Expected result**

Lecturer lecturerExpected1 = new Lecturer { lecturer\_id = 100, lecturer\_username = "teacherone", lecturer\_password = "123456", lecturer\_name = "teacher one", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "SE@vr.camt.info", lecturer\_tel = "0833201787", lecturer\_approvement = 0 };

Lecturer lecturerExpected2 = new Lecturer();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 13.1 | This case tests to get the Lecturer object information from the database, which is contained in the database | lecturerId = 100 | Assert.AreEqual(lecturerExpected1, actual); | lecturerExpected1 | lecturerExpected1 | Pass |
| 13.2 | This case tests to get the Lecturer information from the database, which is not contained in the database. | lecturerId = 1000 | Assert.AreEqual(lecturerExpected2, actual); | lecturerExpected2 | lecturerExpected2 | Pass |

## **UTC-14 approveLecturerStatus(lecId : int) : Lecturer**

**Package:** Repository

**Class:** LecturerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is used for approve Lecturer status in the database.

**Prerequisites or Test Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data name** | **Data type** | **Data value** |
| 1.1 | int | lecturerId | 100 |
| 1.2 | string | lectuerUsername | teacherone |
| 1.3 | string | lectuerPassword | 123456 |
| 1.4 | string | lectuerName | teacher one |
| 1.5 | string | lectuerFaculty | CAMT |
| 1.6 | string | lectuerDepartment | SE |
| 1.7 | string | lectuerEmail | SE@vr.camt.info |
| 1.8 | string | lectuerTel | 0833201787 |

**Expected result**

String LecturerIdExpected = “100”

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 14.1 | This case tests for updating Lecturer approvement in the database. (Successful) | lecturerId = 100 | Assert.AreEqual(LecturerIdExpected, actual); | LecturerIdExpected | LecturerIdExpected | Pass |

## **UTC-15** **viewLecturerNotAvailable() : List<Lecturer>**

**Package:** Repository

**Class:** LecturerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId **: int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is used for view the list of unavailable Lecturer in the database.

**Prerequisites or Test Data**

Appendix A: 1-2

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

List<Lecturer> lecListExpected = new List<Lecturer> { lecturer1 };

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 15.1 | This case tests to get the list of Lecturer list object information from the database, which Lecturer approval equal false. | None | CollectionAssert.AreEqual(lecListExpected, actual); | lecListExpected | lecListExpected | Pass |

## **UTC-16** **viewLecturerAvailable() : List<Lecturer>**

**Package:** Repository

**Class:** LecturerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId **: int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is used for view the list of available Lecturer in the database.

**Prerequisites or Test Data**

Appendix A: 1-1 and this below table

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data name** | **Data type** | **Data value** |
| 1.1 | int | lecturerId | 100 |
| 1.2 | string | lectuerUsername | teacherone |
| 1.3 | string | lectuerPassword | 123456 |
| 1.4 | string | lectuerName | teacher one |
| 1.5 | string | lectuerFaculty | CAMT |
| 1.6 | string | lectuerDepartment | SE |
| 1.7 | string | lectuerEmail | SE@vr.camt.info |
| 1.8 | string | lectuerTel | 0833201787 |

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 1, lecturer\_username = "rimi", lecturer\_password = "123456", lecturer\_name = "rimi park", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "SE@gmail.com", lecturer\_tel = "0832224425", lecturer\_approvement = 1 };

Lecturer lecturer2 = new Lecturer { lecturer\_id = 100, lecturer\_username = "teacherone", lecturer\_password = "123456", lecturer\_name = "teacher one", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "SE@vr.camt.info", lecturer\_tel = "0833201787", lecturer\_approvement = 1 };

List<Lecturer> lecListExpected = new List<Lecturer> { lecturer1, lecturer2 };

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 16.1 | This case tests to get the list of Lecturer list object information from the database, which Lecturer approval equal true. | None | CollectionAssert.AreEqual(lecListExpected, actual); | lecListExpected | lecListExpected | Pass |

## **UTC-17** **viewLecturerByUsernamePassword(lecUsername : string, lecPassword : string) : Lecturer**

**Package:** Repository

**Class:** LecturerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, lecUsername : string, **lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is used for view the Lecturer information in the database by Username and Password.

**Prerequisites or Test Data**

Appendix A: 1-1

**Expected result**

Lecturer lecturerExpected1 = new Lecturer { lecturer\_id = 1, lecturer\_username = "rimi", lecturer\_password = "123456", lecturer\_name = "rimi park", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "SE@gmail.com", lecturer\_tel = "0832224425", lecturer\_approvement = 1 };

Lecturer lecturerExpected2 = new Lecturer();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 17.1 | This case tests to get the Lecturer object information from the database, which lecuturer\_username and lecturer\_password are existed in. | lecuturer\_username = "rimi"  lecturer\_password = "123456" | Assert.AreEqual(lecturerExpected1, actual); | lecturerExpected1 | lecturerExpected1 | Pass |
| 17.2 | This case tests to get the Lecturer object information from the database, which lecuturer\_username and lecturer\_password are not existed in. | lecuturer\_username = “eiei”  lecturer\_password = "123456" | Assert.AreEqual(lecturerExpected2, actual); | lecturerExpected2 | lecturerExpected2 | Pass |

## **UTC-18** **viewLecturerByUsername(lecUsername : string) : Lecturer**

**Package:** Repository

**Class:** LecturerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is used for view the Lecturer information in the database.

**Prerequisites or Test Data**

Appendix A: 1-1

**Expected result**

int lecturerIdExpected1= 1;

int lecturerIdExpected2 = 0;

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 18.1 | This case tests to get the lecturer id from the database, which lecturer\_username are existed in. | lecturer\_username = "rimi" | Assert.AreEqual(lecturerIdExpected1, actual); | lecturerIdExpected1 | lecturerIdExpected1 | Pass |
| 18.2 | This case tests to get the lecturer id from the database, which lecturer\_username are not existed in. | lecturer\_username = “eiei” | Assert.AreEqual(lecturerIdExpected2, actual); | lecturerIdExpected2 | lecturerIdExpected2 | Pass |

## **UTC-19** **deleteLecturer(lecturerId : int) : bool**

**Package:** Repository

**Class:** LecturerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is delete the Lecturer information from the database.

**Prerequisites or Test Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Data name** | **Data type** | **Data value** |
| 1.1 | int | lecturerId | 100 |
| 1.2 | string | lectuerUsername | teacherone |
| 1.3 | string | lectuerPassword | 123456 |
| 1.4 | string | lectuerName | teacher one |
| 1.5 | string | lectuerFaculty | CAMT |
| 1.6 | string | lectuerDepartment | SE |
| 1.7 | string | lectuerEmail | SE@vr.camt.info |
| 1.8 | string | lectuerTel | 0833201787 |

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 19.1 | This case tests to delete the lecturer information in the database. | lecturer\_id = 100 | Assert.True(actual); | True | True | Pass |

## **UTC-20** **viewAdminByUsernamePassword(adminUsername : string, adminPassword : string) : Admin**

**Package:** Repository

**Class:** AdministratorRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int,lecUsername : string, **lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view Administrator information via Username and Password from the database.

**Prerequisites or Test Data**

Appendix A: 3-1

**Expected result**

Admin adminExpected1 = new Admin { admin\_id = 1, admin\_username = "admin", admin\_password = "123456", admin\_name = "admin camt" };

Admin adminExpected2 = new Admin();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 20.1 | This case tests to get the Administrator object information from the database, which admin\_username and admin\_password are existed in. | admin\_username = "admin";  admin\_password = "123456"; | Assert.AreEqual(adminListExpected1, actual); | adminListExpected1 | adminListExpected1 | Pass |
| 20.2 | This case tests to get the Administrator object information from the database, which admin\_username and admin\_password are not existed in. | admin\_username = "eiei";  admin\_password = "123456"; | Assert.AreEqual(adminListExpected2, actual); | adminListExpected2 | adminListExpected2 | Pass |

## **UTC-21** **viewAdminByUsername(adminUsername : string) : Admin**

**Package:** Repository

**Class:** AdministratorRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view Administrator information via Username from the database.

**Prerequisites or Test Data**

Appendix A: 3-1

**Expected result**

**Administrator parameter form.**

Administrator (admin\_id, admin\_username, admin\_password, admin\_name);

**Administrator Object result.**

adminIdExpected1 = 1  
adminIdExpected2 = 0

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 21.1 | This case tests to get the administrator id from the database, which admin\_username are existed in. | admin\_username = "admin" | Assert.AreEqual(adminIdExpected 1, actual); | adminIdExpected1 | adminIdExpected1 | Pass |
| 21.2 | This case tests to get the administrator id from the database, which admin\_username are existed in. | admin\_username = “eiei” | Assert.AreEqual(l adminIdExpected 2, actual); | adminIdExpected2 | adminIdExpected2 | Pass |

## **UTC-22** **viewSemesterInfo(int semId) : Semester**

**Package:** Repository

**Class:** SemesterRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : **int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view Semester information from the database.

**Prerequisites or Test Data**

Appendix A: 4-1

**Expected result**

Semester semesterExpected1 = new Semester { semester\_id = 1, academic\_year = "1/2556" };

Semester semesterExpected2 = new Semester ();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 22.1 | This case tests to get the Semester object information from the database, which is contained in the database | semester\_id = 1 | Assert.AreEqual(semesterrExpected, actual); | semesterExpected11 | semesterExpected11 | Pass |
| 22.2 | This case tests to get the Semester information from the database, which is not contained in the database. | semester\_id = 100 | Assert.AreEqual(semesterrExpected, actual); | semesterExpected2 | semesterExpected2 | Pass |

## **UTC-23** **viewAllSemester() : List<Semester>**

**Package:** Repository

**Class:** SemesterRepository

**Unit Test Case-11:** insertLecturerInfo(lecId **: int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view All Semester information from the database.

**Prerequisites or Test Data**

Appendix A: 4-1,4-2,4-3,4-4

**Expected result**

Semester semester1 = new Semester { semester\_id = 1, academic\_year = "1/2556" };

Semester semester2 = new Semester { semester\_id = 2, academic\_year = "2/2556" };

Semester semester3 = new Semester { semester\_id = 3, academic\_year = "1/2557" };

Semester semester4 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

List<Semester> semesterListExpected = new List<Semester> { semester1, semester2, semester3, semester4 };

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 23.1 | This case tests to get the list of Semester list object information from the database, which Lecturer approval equal true. | None | CollectionAssert.AreEqual(semesterListExpected, actual); | semesterListExpected | semesterListExpected | Pass |

## **UTC-24** **insertSemester(academicYear : string) : bool**

**Package:** Repository

**Class:** SemesterRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is insert Semester information into the database.

**Prerequisites or Test Data**

:None

**Expected result**

True, False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 24.1 | Test if Semester information provided is added into the database successfully. Test for a successful case. | academic\_year = "1/2558" | Assert.True(actual); | True | True | Pass |
| 24.2 | Test if Semester information provided is added into the database successfully. Test for an unsuccessful case: providing duplicate studentId in the database. | academic\_year = null | Assert.IsFalse(actual) | False | False | Pass |

## **UTC-25** **updateSemester(semesterId : int, academicYear : string) : bool**

**Package:** Repository

**Class:** SemesterRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int,lecUsername : string, **lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is update Semester information in the database.

**Prerequisites or Test Data**

Appendix A: 4-4

**Expected result**

True, False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 25.1 | Test if Semester information provided is updated into the database successfully. Test for a successful case. | semester\_id = 4  academic\_year = "1/2558" | Assert.True (actual); | True | True | Pass |
| 25.2 | Test if Semester information provided is updated into the database successfully. Test for an unsuccessful case: providing studentId that do not have in the database. | semester\_id = 4  academic\_year = null | Assert.False(actual); | False | False | Pass |

## **UTC-26** **deleteSemester(semesterId : int) : bool**

**Package:** Repository

**Class:** SemesterRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is delete Semester information in the database.

**Prerequisites or Test Data**

Appendix A: 4-1

**Expected result**

True

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 26.1 | This case tests to delete the Semester information in the database. | semester\_id = 11 | Assert.True(actual); | True | True | Pass |

## **UTC-27** **viewCourseInfoByLecturerId** **(lecId : int, semesterId : int): List<Course>**

**Package:** Repository

**Class:** CourseRepository

**Unit Test Case-27:** viewCourseInfoByLecturerId (lecId : int, semesterId : int)

**Description:** The test case is used for testing viewCourseInfoByLecturerId method in CourseRepository. This method is used for viewing list of course from the database by using lecturer id for searching.

**Prerequisites or Test Data**

Appendix A: 1-2, 4-4, 5-4

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course course1 = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

List<Course> courseListExpected1 = new List<Course> { course1 };

List<Course> courseListExpected12 = new List<Course> ();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 27.1 | This case tests to view a list of course object from the database by using lecturer id for searching. (Exist course information in the database) | lecturer\_id = 2;  semester\_id = 4; | CollectionAssert.AreEqual(courseListExpected1, actual); | courseListExpected1 | courseListExpected1 | Pass |
| 27.2 | This case tests to view a list of course object from the database by using lecturer id for searching. (Not Exist course information in the database) | lecturer\_id = -1  semester\_id = 4 | CollectionAssert.AreEqual(courseListExpected2, actual); | courseListExpected12 | courseListExpected12 | Pass |

## **UTC-28** **viewCourseInfoByCourseId(course\_id: int): Course**

**Package:** Repository

**Class:** CourseRepository

**Unit Test Case-28:** viewCourseInfoByCourseId(course\_id: int)

**Description:** The test case is used for testing viewCourseInfoByCourseId method in CourseRepository. This method is used for viewing course information from the database by using course id for searching.

**Prerequisites or Test Data**

Appendix A : 1-1, 4-2, 5-3

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 1, lecturer\_username = "rimi", lecturer\_password = "123456", lecturer\_name = "rimi park", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "SE@gmail.com", lecturer\_tel = "0832224425", lecturer\_approvement = 1 };

Semester semester1 = new Semester { semester\_id = 2, academic\_year = "2/2556" };

Course courseExpected1 = new Course { course\_id = 5, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Computer", course\_description = "Computer is easy", course\_credit = 3 };

**Course courseExpected2 = new Course ();**

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 28.1 | This case tests to view a course object from the database by using course id for searching. (Exist course information in the database) | course\_id = 5 | Assert.AreEqual(courseExpected1, actual); | courseExpected1 | courseExpected1 | Pass |
| 28.2 | This case tests to view a course object from the database by using course id for searching. (Not Exist course information in the database) | course\_id = -1 | Assert.AreEqual(courseListExpected2, actual); | courseExpected2 | courseExpected2 | Pass |

## **UTC-29** **viewCoursebySemesterId(semId : int): List<Course>**

**Package:** Repository

**Class:** CourseRepository

**Unit Test Case-29:** viewCoursebySemesterId(semId : int)

**Description:** The test case is used for testing viewCoursebySemesterId method in CourseRepository. This method is used for viewing a list of course information from the database by using semester id for searching.

**Prerequisites or Test Data**

Appendix A: 1-2, 4-4, 5-4

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course course1 = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

List<Course> courseListExpected1= new List<Course> {course1};

List<Course> courseListExpected12 = new List<Course> ();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 29.1 | This case tests to view a list of course object from the database by using semester id for searching. (Exist course information in the database) | semester\_id =4 | CollectionAssert.AreEqual(courseListExpected1, actual); | courseListExpected1 | courseListExpected1 | Pass |
| 29.2 | This case tests to view a list of course object from the database by using semester id for searching. (Not Exist course information in the database) | semester\_id = -1 | CollectionAssert.AreEqual(courseListExpected2, actual); | courseListExpected12 | courseListExpected12 | Pass |

## **UTC-30 insertCourse(lecturerId: int, semesterId : int, courseName : string, courseCredit : int, courseDescription : string): bool**

**Package:** Repository

**Class:** CourseRepository

**Unit Test Case-30:** insertCourse(lecturerId: int, semesterId : int, courseName : string, courseCredit : int, courseDescription : string)

**Description:** The test case is used for testing insertCourse method in CourseRepository. This method is used for inserting course information to the database.

**Prerequisites or Test Data**

: None

**Expected result**

**True, False**

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 30.1 | This case tests to insert course information to the database. (successful) | lecturer\_id = 1;  semester\_id = 2;  course\_name = "C++ langguage";  course\_credit = 3;  course\_description = "C++ is good"; | Assert.True (actual); | **True** | **True** | Pass |
| 30.2 | This case tests to insert course information to the database. (un successful) | lecturer\_id = 1;  semester\_id = 2;  course\_name = null;  course\_credit = 3;  course\_description = "C++ is good"; | Assert.False (actual); | **False** | **False** | Pass |

## **UTC-31 updateCourse(courseId: int, lecturerId: int, semesterId : int, courseName : string, courseCredit : int, courseDescription : string): bool**

**Package:** Repository

**Class:** CourseRepository

**Unit Test Case-31:** updateCourse(courseId: int, lecturerId: int, semesterId : int, courseName : string, courseCredit : int, courseDescription : string)

**Description:** The test case is used for testing updateCourse method in CourseRepository. This method is used for updating course information in the database.

**Prerequisites or Test Data**

Appendix A: 5-4

**Expected result**

**True, False**

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 31.1 | This case tests to update course information to the database. (successful) | course\_id = 9;  lecturer\_id = 2;  semester\_id = 4;  course\_name = "Test";  course\_credit = 3;  string course\_description = "Test na ja"; | Assert.True (actual); | **True** | **True** | Pass |
| 31.2 | This case tests to update course information to the database. (un successful) | course\_id = 9;  lecturer\_id = 2;  semester\_id = 4;  course\_name = null;  course\_credit = 3;  course\_description = "Test na ja"; | Assert.False (actual); | **False** | **False** | Pass |

## **UTC-32 deleteCourse(courseId int): bool**

**Package:** Repository

**Class:** CourseRepository

**Unit Test Case-31:** deleteCourse(courseId int)

**Description:** The test case is used for testing deleteCourse method in CourseRepository. This method is used for delete course information in the database.

**Prerequisites or Test Data**

Appendix A: 5-4

**Expected result**

True

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 32.1 | This case tests to delete course information in the database | int course\_id = 9; | Assert.True (actual); | **True** | **True** | Pass |

## **UTC-33** **viewPreviousCourse(int lecturerId): bool**

**Package:** Repository

**Class:** CourseRepository

**Unit Test Case-33:** viewPreviousCourse(int lecturerId)

**Description:** The test case is used for testing viewPreviousCourse method in CourseRepository. This method is used for viewing latest course information that was created by lecturer.

**Prerequisites or Test Data**

Appendix A: 1-2, 4-4, 5-4

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course courseExpected = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 33.1 | This case tests to view latest course information that was created by lecturer. | lecturer\_id = 2; | Assert.AreEqual(courseExpected, actual); | courseExpected | courseExpected | Pass |

## **UTC-34** **viewStudentInCourse(courseId: int): List<CourseRegistration>**

**Package:** Repository

**Class:** CourseRegisRepository

**Unit Test Case-34:** viewStudentInCourse(courseId: int)

**Description:** The test case is used for testing viewStudentInCourse method in CourseRegisRepository. This method is used for viewing a list of student information that register in the course

**Prerequisites or Test Data**

Appendix A: 1-2, 2-1, 2-2, 4-4, 5-4, 6-8, 6-9

**Expected result**

Student student1 = new Student { student\_id = 542115091, student\_name = "john conner", student\_username = "john", student\_password = "123456", student\_faculty = "CAMT", student\_department = "ANI", student\_address = "Thailand", student\_email = "ANI@gmail.com", student\_tel = "0833201787", student\_approvement = 1 };

Student student2 = new Student { student\_id = 542115092, student\_name = "jame conner", student\_username = "jame", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0821330433", student\_approvement = 1 };

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course course1 = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

CourseRegistration courseRegis1 = new CourseRegistration { CourseRegistration\_id = 12, course\_id = course1, student\_id = student1 };

CourseRegistration courseRegis2 = new CourseRegistration { CourseRegistration\_id = 13, course\_id = course1, student\_id = student2 };

List<CourseRegistration> courseRegisListExpected1 = new List<CourseRegistration> { courseRegis1,courseRegis2};

List<CourseRegistration> courseRegisListExpected2 = new List<CourseRegistration> ();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 34.1 | This case tests to view a list of student information that register in the course. (Exist data in the database) | course\_id = 9; | Assert.AreEqual(courseRegisListExpected1, actual); | courseRegisListExpected1 | courseRegisListExpected1 | Pass |
| 34.2 | This case tests to view a list of student information that register in the course. (Not Exist data in the database) | course\_id = 100; | Assert.AreEqual(courseRegisListExpected2, actual); | courseRegisListExpected2 | courseRegisListExpected2 | Pass |

## **UTC-35** **viewCourseFromStudent(student\_id: int): List<CourseRegistration>**

**Package:** Repository

**Class:** CourseRegisRepository

**Unit Test Case-35:** viewCourseFromStudent(student\_id: int)

**Description:** The test case is used for testing viewCourseFromStudent method in CourseRegisRepository. This method is used for viewing a list of courses information that were registered in the course

**Prerequisites or Test Data**

Appendix A: 1-1, 2-4, 4-2, 5-3, 6-3

Expected result

Student student1 = new Student { student\_id = 542115094, student\_name = "elrond rivendell", student\_username = "elrond", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0823149555", student\_approvement = 1 };

Lecturer lecturer1 = new Lecturer { lecturer\_id = 1, lecturer\_username = "rimi", lecturer\_password = "123456", lecturer\_name = "rimi park", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "SE@gmail.com", lecturer\_tel = "0832224425", lecturer\_approvement = 1 };

Semester semester1 = new Semester { semester\_id = 2, academic\_year = "2/2556" };

Course course1 = new Course { course\_id = 5, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Computer", course\_description = "Computer is easy", course\_credit = 3 };

CourseRegistration courseRegis1 = new CourseRegistration { CourseRegistration\_id = 6, course\_id = course1, student\_id = student1 };

**List<CourseRegistration> courseRegisListExpected1 = new List<CourseRegistration> { courseRegis1 };**

**List<CourseRegistration> courseRegisListExpected2 = new List<CourseRegistration> ()};**

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 35.1 | This case tests to view a list of courses information that were registered in the course. (Exist data in the database) | student\_id = 542115094 | CollectionAssert.AreEqual(courseRegisListExpected1, actual); | courseRegisListExpected1 | courseRegisListExpected1 | Pass |
| 35.2 | This case tests to view a list of courses information that were registered in the course. (Not Exist data in the database) | student\_id = 5000000 | CollectionAssert.AreEqual(courseRegisListExpected2, actual); | courseRegisListExpected2 | courseRegisListExpected2 | Pass |

## **UTC-36 insertStudentsInCourse** **(courseId: int, studentIdList: List<int>): bool**

**Package:** Repository

**Class:** CourseRegisRepository

**Unit Test Case-36:** insertStudentsInCourse (courseId: int, studentIdList: List<int>)

**Description:** The test case is used for testing insertStudentsInCourse method in CourseRegisRepository. This method is used for inserting a student to the course for registration in course.

**Prerequisites or Test Data**

Appendix A: 2-4,5-4

**Expected result**

**True/False**

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 36.1 | This case tests to insert a student to the course for registration in course. (successful) | student\_id1 = 542115094;  List<int> studentIdList = new List<int> { student\_id1 };  int course\_id = 9; | Assert.IsTrue(actual); | True | True | Pass |
| 36.2 | This case tests to insert a student to the course for registration in course. (un successful) | List<int> studentIdList = new List<int>();  course\_id = 9; | Assert.IsFalse(actual); | False | False | Pass |

## **UTC-37** **deleteStudentInCourse (courseId: int, studentIdList: int): bool**

**Package:** Repository

**Class:** CourseRegisRepository

**Unit Test Case-37:** deleteStudentInCourse (courseId: int, studentIdList: int)

**Description:** The test case is used for testing insertStudentsInCourse method in CourseRegisRepository. This method is used for deleting of a student from the course

**Prerequisites or Test Data**

Appendix A: 2-4, 5-4

**Expected result**

**True**

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 37.1 | This case tests to delete a student from the course (successful) | student\_id = 542115094;  course\_id = 9; | Assert.True (actual); | True | True | Pass |

## **UTC-38** **insertTesting (courseId: int, testing\_name: string, testing\_type: string, testing\_amountQuestion: int, testing\_score: double, testing\_random: int, testing\_posting: int, testing\_student: string, submit\_date: Nullable<DateTime>): bool**

**Package:** Repository

**Class:** TestingRepository

**Unit Test Case-38:** insertTesting (courseId: int, testing\_name: string, testing\_type: string, testing\_amountQuestion: int, testing\_score: double, testing\_random: int, testing\_posting: int, testing\_student: string, submit\_date: Nullable<DateTime>)

**Description:** The test case is used for testing insertTesting method in TestingRepository. This method is used for inserting a testing to the database.

**Prerequisites or Test Data**

None

**Expected result**

True,Flase

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 38.1 | This case tests to insert a testing to the database. (successful) | course\_id = 14;  testing\_name = "test1";  testing\_type = "1";  testing\_amountQuestion = 1;  testing\_score = 3;  testing\_random = 1;  testing\_posting = 1;  testing\_student = "542115094";  Nullable<DateTime> submit\_date = null; | Assert.True (actual); | True | True | Pass |
| 38.1 | This case tests to insert a testing to the database. (un successful) | course\_id = 14;  testing\_name = null;  testing\_type = "1";  testing\_amountQuestion = 1;  testing\_score = 3;  testing\_random = 1;  testing\_posting = 1;  string testing\_student = "542115094";  Nullable<DateTime> submit\_date = null; | Assert.False(actual) | False | False | Pass |

## **UTC-39** **viewPreviousTestingId(courseId: int): int**

**Package:** Repository

**Class:** TestingRepository

**Unit Test Case-39:** viewPreviousTestingId(courseId: int)

**Description:** The test case is used for testing viewPreviousTestingId method in TestingRepository. This method is used for viewing a latest testing that was created in course.

**Prerequisites or Test Data**

Appendix A: 5-4, 7.7

**Expected result**

int testingIdExpected1 = 11;

int testingIdExpected2 = 0;

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 39.1 | This case tests to view a latest testing that was created in course. (Exist data in the database) | course\_id = 9; | Assert.AreEqual(testingIdExpected, actual); | testingIdExpected1 | testingIdExpected1 | Pass |
| 39.2 | This case tests to view a latest testing that was created in course. (Not exist data in the database) | course\_id = 100; | Assert.AreEqual(testingIdExpected, actual); | testingIdExpected2 | testingIdExpected2 | Pass |

## **UTC-40** **viewTestingByTestingId(testingId: int): Testing**

**Package:** Repository

**Class:** TestingRepository

**Unit Test Case-40:** viewTestingByTestingId(testingId: int)

**Description:** The test case is used for testing viewTestingByTestingId method in TestingRepository. This method is used for viewing testing information by using testing id in searching.

**Prerequisites or Test Data**

Appendix A: 1-2, 2-1, 2-2, 4-4, 5-4, 7-7

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course course1 = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

Student student1 = new Student { student\_id = 542115091, student\_name = "john conner", student\_username = "john", student\_password = "123456", student\_faculty = "CAMT", student\_department = "ANI", student\_address = "Thailand", student\_email = "ANI@gmail.com", student\_tel = "0833201787", student\_approvement = 1 };

Student student2 = new Student { student\_id = 542115092, student\_name = "jame conner", student\_username = "jame", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0821330433", student\_approvement = 1 };

List<Student> studentList = new List<Student> {student1, student2};

Testing testingExpected1 = new Testing { testing\_id = 11, course\_id = course1, testing\_name = "Test", testing\_type = "2", testing\_amountQuestion = 1, testing\_score = 3.0, tesing\_random = 1, testing\_posting = 1, testing\_submitTime = null, student\_id = studentList };

Testing testingExpected2 = new Testing();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 40.1 | This case tests to view testing information in the database. (Exist data in the database) | testing\_id = 11 | Assert.AreEqual(testingExpected1, actual); | testingExpected1 | testingExpected1 | Pass |
| 40.2 | This case tests to view testing information in the database. (Not exist data in the database) | testing \_id = 110; | Assert.AreEqual(testingExpected2 actual); | testingExpected2 | testingExpected2 | Pass |

## **UTC-41** **viewListTestingbyCourseId(courseId: int): List<Testing>**

**Package:** Repository

**Class:** TestingRepository

**Unit Test Case-41:** viewListTestingbyCourseId(courseId: int)

**Description:** The test case is used for testing viewListTestingbyCourseId method in TestingRepository. This method is used for viewing a list of testing information in a course.

**Prerequisites or Test Data**

Appendix A: 1-2, 2-1, 2-2, 4-4, 5-4, 7-7

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course course1 = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

Student student1 = new Student { student\_id = 542115091, student\_name = "john conner", student\_username = "john", student\_password = "123456", student\_faculty = "CAMT", student\_department = "ANI", student\_address = "Thailand", student\_email = "ANI@gmail.com", student\_tel = "0833201787", student\_approvement = 1 };

Student student2 = new Student { student\_id = 542115092, student\_name = "jame conner", student\_username = "jame", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0821330433", student\_approvement = 1 };

List<Student> studentList = new List<Student>{ student1, student2};

Testing testing1 = new Testing { testing\_id = 11, course\_id = course1, testing\_name = "Test", testing\_type = "2", testing\_amountQuestion = 1, testing\_score = 3, tesing\_random = 1, testing\_posting = 1, testing\_submitTime = null, student\_id = studentList };

List<Testing> testingListExpected1 = new List<Testing> {testing1};

List<Testing> testingListExpected2 = new List<Testing> ();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 40.1 | This case tests to view a list of testing information in a course. (Exist data in the database) | course\_id = 9 | CollectionAssert.AreEqual(testingListExpected1, actual); | testingListExpected1 | testingListExpected1 | Pass |
| 40.2 | This case tests to view a list of testing information in a course. (Not exist data in the database) | course\_id = 110; | CollectionAssert.AreEqual(testingListExpected2, actual); | testingListExpected2 | testingListExpected2 | Pass |

## **UTC-42** **updateTestinginfo(testingId: int, courseId: int, testing\_name: string, testing\_type: string, testing\_amountQuestion: int, testing\_score: double, testing\_random: int, testing\_posting: int, testing\_student: string): bool**

**Package:** Repository

**Class:** TestingRepository

**Unit Test Case-42:** updateTestinginfo(testingId: int, courseId: int, testing\_name: string, testing\_type: string, testing\_amountQuestion: int, testing\_score: double, testing\_random: int, testing\_posting: int, testing\_student: string)

**Description:** The test case is used for testing updateTestinginfo method in TestingRepository. This method is used for updating testing information in the database.

**Prerequisites or Test Data**

Appendix A: 1-2, 2-1, 2-2, 4-4, 5-4, 7-7

**Expected result**

True/False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 42.1 | This case tests to update testing information in the database (successful) | testing\_id = 11;  course\_id = 9;  testing\_name = "Test";  testing\_type = "2";  testing\_amountQuestion = 1;  testing\_score = 3;  tesing\_random = 1;  testirng\_posting = 1;  student\_id = "542115091,542115092"; | Assert.IsTrue(actual); | True | True | Pass |
| 42.2 | This case tests to update testing information in the database (un successful) | testing\_id = 11;  course\_id = 9;  testing\_name = "Test";  testing\_type = null;  testing\_amountQuestion = 1;  testing\_score = 3;  tesing\_random = 1;  testirng\_posting = 1;  string student\_id = "542115091,542115092"; | Assert.IsFalse(actual); | True | True | Pass |

## **UTC-43** **viewListTestingThatPost():List<Testing>**

**Package:** Repository

**Class:** TestingRepository

**Unit Test Case-43:** viewListTestingThatPost()

**Description:** The test case is used for testing viewListTestingThatPost method in TestingRepository. This method is used for viewing a list of testing that are posting status equally 1.

**Prerequisites or Test Data**

Appendix A: 1-2, 2-1, 2-2, 4-4, 5-4, 7-7

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course course1 = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

Student student1 = new Student { student\_id = 542115091, student\_name = "john conner", student\_username = "john", student\_password = "123456", student\_faculty = "CAMT", student\_department = "ANI", student\_address = "Thailand", student\_email = "ANI@gmail.com", student\_tel = "0833201787", student\_approvement = 1 };

Student student2 = new Student { student\_id = 542115092, student\_name = "jame conner", student\_username = "jame", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0821330433", student\_approvement = 1 };

List<Student> studentList = new List<Student> { student1, student2 };

Testing testing1 = new Testing { testing\_id = 11, course\_id = course1, testing\_name = "Test", testing\_type = "2", testing\_amountQuestion = 1, testing\_score = 3, tesing\_random = 1, testing\_posting = 1, testing\_submitTime = null, student\_id = studentList };

List<Testing> testingListExpected = new List<Testing> { testing1 };

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 43.1 | This case tests to view a list of testing that are posting status equally 1.(Exist data in database) | None | CollectionAssert.AreEqual(testingListExpected, actual); | testingListExpected | True | Pass |

## **UTC-44** **updateStudentInTesting(testingId int, testing\_student string): bool**

**Package:** Repository

**Class:** TestingRepository

**Unit Test Case-44:** updateStudentInTesting(testingId int, testing\_student string)

**Description:** The test case is used for testing updateStudentInTesting method in TestingRepository. This method is used for updating a student in the testing for accession.

**Prerequisites or Test Data**

Appendix A: 1-2, 2-1, 2-2, 4-4, 5-4, 7-7

**Expected result**

True/False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 44.1 | This case tests to update a student in the testing for accession. (successful) | testing\_id = 11;  student\_id = "542115091,542115092"; | Assert.IsTrue(actual); | True | True | Pass |
| 44.2 | This case tests to update a student in the testing for accession. (un successful) | testing\_id = 11;  student\_id = null; | Assert.IsFalse(actual); | False | False | Pass |

## **UTC-45** **insertQuestion(testing\_id : int, listQuestion : List<Question>) : bool**

**Package:** Repository

**Class:** QuestionRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, lecUsername : string, lecPassword : **stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is insert Question information into the database.

**Prerequisites or Test Data**

:None

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 24.1 | Test if Question information provided is added into the database successfully. Test for a successful case. | testing\_id = 11;  Question question = new Question { question\_type = "3", question\_description = "Question1", question\_choice = "", question\_solution = "", question\_point = 3 };  List<Question> questionList = new List<Question> { question }; | Assert.True(actual); | True | True | Pass |
| 24.2 | Test if Question information provided is added into the database successfully. Test for an unsuccessful case: providing duplicate testing\_Id in the database. | testing\_id = 11;  Question question = new Question { question\_type = null, question\_description = "Question1", question\_choice = "", question\_solution = "", question\_point = 3 };  List<Question> questionList = new List<Question> { question }; | Assert.IsFalse(actual) | False | False | Pass |

## **UTC-46** **viewQuestionbyTestingId(testing : Testing) : List<Question>**

**Package:** Repository

**Class:** QuestionRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view Question information into the database.

**Prerequisites or Test Data**

Appendix A: 7-1, 8-6

**Expected result**

Testing testing = new Testing { testing\_id = 11, course\_id = null, student\_id = null, tesing\_random = 0, testing\_amountQuestion = 0, testing\_name = null, testing\_posting = 0, testing\_score = 0, testing\_submitTime = null, testing\_type = null };

Question question = new Question { question\_id = 14, testing\_id = testing, question\_type = "2", question\_description = "Test eiei", question\_choice = "True#$False", question\_solution = "1", question\_point = 3 };

List<Question> questionListExpected1 = new List<Question> { question };

List<Question> questionListExpected2 = new List<Question>();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 46.1 | This case tests to get the Question object information from the database, which is contained in the database | Testing testing1 = new Testing { testing\_id = 11 }; | CollectionAssert.AreEqual(questionListExpected, actual); | questionListExpected1 | questionListExpected1 | Pass |
| 46.2 | This case tests to get the Question information from the database, which is not contained in the database. | Testing testing2 = new Testing { testing\_id = 11 }; | CollectionAssert.AreEqual(questionListExpected, actual); | questionListExpected2 | questionListExpected2 | Pass |

## **UTC-47** **updateQuestioninfo(testing\_id : int, listQuestion : List<Question>) : bool**

**Package:** Repository

**Class:** QuestionRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, lecUsername : string, lecPassword : **stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is update Question information into the database.

**Prerequisites or Test Data**

Appendix A: 7-1, 8-6

**Expected result**

True/False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 47.1 | Test if Question information provided is updated into the database successfully. Test for a successful case. | testing\_id = 11;  Question question = new Question { question\_id = 14, question\_type = "2", question\_description = "Test eiei", question\_choice = "True#$False", question\_solution = "1", question\_point = 3 };  List<Question> questionList = new List<Question> { question }; | Assert.True (actual); | True | True | Pass |
| 47.2 | Test if Question information provided is updated into the database successfully. Test for an unsuccessful case: providing testing\_id that do not have in the database. | testing\_id = 11;  Question question = new Question { question\_id = 14, question\_type = null, question\_description = "Test eiei", question\_choice = "True#$False", question\_solution = "1", question\_point = 3 };  List<Question> questionList = new List<Question> { question }; | Assert.False(actual); | True | True | Pass |

## **UTC-48** **viewQuestionByQuestionId(int questionId) : Question**

**Package:** Repository

**Class:** QuestionRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : **int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is update Question information into the database.

**Prerequisites or Test Data**

Appendix A: 7-1, 8-6

**Expected result**

Question questionExpected1 = new Question { question\_id = 14, testing\_id = testing, question\_type = "2", question\_description = "Test eiei", question\_choice = "True#$False", question\_solution = "1", question\_point = 3 };

Question questionExpected2 = new Question { question\_id = 0, testing\_id = null, question\_type = null, question\_description = null, question\_choice = null, question\_solution = null, question\_point = 0 };

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 48.1 | This case tests to get the Question object information from the database, which questionId is existed in. | questionId = 14 | Assert.AreEqual(questionExpected, actual); | questionExpected1 | questionExpected1 | Pass |
| 48.2 | This case tests to get the Question object information from the database, which questionId is existed in. | questionId = 100 | Assert.AreEqual(questionExpected, actual); | questionExpected2 | questionExpected2 | Pass |

## **UTC-49** **insertAnswerSheet(studentId : int, testingId : int, answerSheetScore : Nullable<double>, submitTime : DateTime, submit\_result : Nullable<int>, isCheck : int) : bool**

**Package:** Repository

**Class:** AnswerSheetRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, lecUsername: string, lecPassword :stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, **lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is insert Answer Sheet information into the database.

**Prerequisites or Test Data**

:None

**Expected result**

True/False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 49.1 | Test if AnswerSheet information provided is added into the database successfully. Test for a successful case. | student\_id = 542115092;  testing\_id = 11;  Nullable<double> answerSheet\_score = 3.0;  DateTime submitTime = DateTime.Now;  Nullable<int> submit\_result = 1;  int isCheck = 0; | Assert.True(actual); | True | True | Pass |
| 49.2 | Test if AnswerSheet information provided is added into the database successfully. Test for an unsuccessful case: providing duplicate student\_id in the database. | student\_id = 542115092;  testing\_id = 11;  Nullable<double> answerSheet\_score = null;  DateTime submitTime = DateTime.Now;  Nullable<int> submit\_result = 1;  int isCheck = 0; | Assert.True(actual); | True | True | Pass |

## **UTC-50** **viewPreviousAnswerSheetId(studentId : int) : int**

**Package:** Repository

**Class:** AnswerSheetRepository

**Unit Test Case-11:** insertLecturerInfo**(lecId : int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**(studentId : int)

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view Answer Sheet information into the database by an identifier.

**Prerequisites or Test Data**

**Expected result**

int answerSheetIdExpected1 = 24;

int answerSheetIdExpected2 = 0;

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 50.1 | This case tests to get the AnswerSheet object information from the database, which student\_id is existed in. | student\_id = 542115093; | Assert.AreEqual(answerSheetIdExpected, actual); | answerSheetIdExpected1 | answerSheetIdExpected1 | Pass |
| 50.2 | This case tests to get the AnswerSheet object information from the database, which student\_id not existed in. | student\_id = 0 | Assert.AreEqual(answerSheetIdExpected, actual); | answerSheetIdExpected2 | answerSheetIdExpected1 | Pass |

## **UTC-51** **viewListAnswerSheetBytestingId(testingId : int) : List<AnswerSheet>**

**Package:** Repository

**Class:** AnswerSheetRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view list of Answer Sheet information into the database by using testing id in searching.

**Prerequisites or Test Data**

:Appendix A: 1-2, 2-1, 2-2, 4-4, 5-4, 7-7, 9-6

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course course1 = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

Student student1 = new Student { student\_id = 542115091, student\_name = "john conner", student\_username = "john", student\_password = "123456", student\_faculty = "CAMT", student\_department = "ANI", student\_address = "Thailand", student\_email = "ANI@gmail.com", student\_tel = "0833201787", student\_approvement = 1 };

Student student2 = new Student { student\_id = 542115092, student\_name = "jame conner", student\_username = "jame", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0821330433", student\_approvement = 1 };

List<Student> studentList = new List<Student> { student1, student2 };

Testing testing1 = new Testing { testing\_id = 11, course\_id = course1, testing\_name = "Test", testing\_type = "2", testing\_amountQuestion = 1, testing\_score = 3, tesing\_random = 1, testing\_posting = 1, testing\_submitTime = null, student\_id = studentList };

string dateTime = "28/7/2014 15:04:15";

DateTime time = Convert.ToDateTime(dateTime);

AnswerSheet answerSheet1 = new AnswerSheet { answerSheet\_id = 25, testing\_id = testing1, student\_id = student1, answerSheet\_score = 3, answerSheet\_time = time, answerSheet\_submitResult = 0, answerSheet\_isChecked = 1 };

List<AnswerSheet> answerSheetListExpected1 = new List<AnswerSheet> { answerSheet1 };

List<AnswerSheet> answerSheetListExpected2 = new List<AnswerSheet> { answerSheet1 };

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 51.1 | This case tests to get the list of AnswerSheet object information from the database by testing id, which is contained in the database | testing\_Id = 11 | CollectionAssert.AreEqual(answerSheetListExpected1, actual); | answerSheetListExpected1 | answerSheetListExpected1 | Pass |
| 52.2 | This case tests to get the the list of AnswerSheet information from the database by testing\_Id, which is not contained in the database. | testing\_Id = 100 | CollectionAssert.AreEqual(answerSheetListExpected2, actual); | answerSheetListExpected2 | answerSheetListExpected2 | Pass |

## **UTC-52** **viewAnswerSheetInfo(answerSheetId : int) : AnswerSheet**

**Package:** Repository

**Class:** AnswerSheetRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, **lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view Answer Sheet information into the database from the answer sheet ID.

**Prerequisites or Test Data**

Appendix A: 1-2, 2-1, 2-2, 4-4, 5-4, 7-7, 9-6

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course course1 = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

Student student1 = new Student { student\_id = 542115091, student\_name = "john conner", student\_username = "john", student\_password = "123456", student\_faculty = "CAMT", student\_department = "ANI", student\_address = "Thailand", student\_email = "ANI@gmail.com", student\_tel = "0833201787", student\_approvement = 1 };

Student student2 = new Student { student\_id = 542115092, student\_name = "jame conner", student\_username = "jame", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0821330433", student\_approvement = 1 };

List<Student> studentList = new List<Student>{ student1, student2};

Testing testing1 = new Testing { testing\_id = 11, course\_id = course1, testing\_name = "Test", testing\_type = "2", testing\_amountQuestion = 1, testing\_score = 3, tesing\_random = 1, testing\_posting = 1, testing\_submitTime = null, student\_id = studentList };

string dateTime = "28/7/2014 15:04:15";

DateTime time = Convert.ToDateTime(dateTime);

AnswerSheet answerSheet1 = new AnswerSheet { answerSheet\_id = 25, testing\_id = testing1, student\_id = student1, answerSheet\_score = 3, answerSheet\_time = time, answerSheet\_submitResult = 0, answerSheet\_isChecked = 1 };

List<AnswerSheet> answerSheetListExpected1 = new List<AnswerSheet> { answerSheet1 };

List<AnswerSheet> answerSheetListExpected2 = new List<AnswerSheet>();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 52.1 | This case tests to get the AnswerSheet object information from the database, which is contained in the database | answerSheet\_id = 25 | Assert.AreEqual(answerSheetExpected, actual); | answerSheetExpected1 | answerSheetExpected1 | Pass |
| 52.2 | This case tests to get the AnswerSheet information from the database, which is not contained in the database. | answerSheet\_id = 100 | Assert.AreEqual(answerSheetExpected, actual); | answerSheetExpected2 | answerSheetExpected2 | Pass |

## **UTC-53** **updateAnswerSheetScore(answerSheetId : int, score Nullable<double>) : bool**

**Package:** Repository

**Class:** AnswerSheetRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int,lecUsername : string, **lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view Answer Sheet information into the database from the answer sheet ID.

**Prerequisites or Test Data**

Appendix A: 9-6

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 53.1 | Test if AnswerSheetScore information provided is updated into the database successfully. Test for a successful case. | answerSheet\_id = 25  score = 3 | Assert.True (actual); | True | True | Pass |
| 53.2 | Test if AnswerSheetScore information provided is updated into the database successfully. Test for an unsuccessful case: providing answerSheet\_id that do not have in the database. | answerSheet\_id = 25  score = null | Assert.False(actual); | False | False | Pass |

## **UTC-54** **insertAnswer(questionId : List<int>, answerSheetId : int, answer : List<string>, answerPoint : List<Nullable<double>>) : bool**

**Package:** Repository

**Class:** AnswerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, lecUsername : string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail **: string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is insert Answer information into the database.

**Prerequisites or Test Data**

:None

**Expected result**

True/False

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 54.1 | Test if Answer information provided is added into the database successfully. Test for a successful case. | List<int> question\_id = new List<int> { 14 };  int answerSheet\_id = 26;  List<string> answer = new List<string> { "1" }; List<Nullable<double>> answerPoint = new List<double?> { 3 }; | Assert.True(actual); | True | True | Pass |
| 54.2 | Test if Answer information provided is added into the database successfully. Test for an unsuccessful case: providing duplicate questionId in the database. | List<int> question\_id = new List<int> { 14 };  int answerSheet\_id = 26;  List<string> answer = new List<string> { null };  List<Nullable<double>> answerPoint = new List<double?> { 3 }; | Assert.IsFalse(actual) | False | False | Pass |

## **UTC-55** **viewAnswerListByAnswerSheetId (answerSheetId : int) : List<Answer>**

**Package:** Repository

**Class:** AnswerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, lecUsername **: string, lecPassword : stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is view Answer information into the database by AnswerSheetId.

**Prerequisites or Test Data**

**Expected result**

Lecturer lecturer1 = new Lecturer { lecturer\_id = 2, lecturer\_username = "tony", lecturer\_password = "123456", lecturer\_name = "tony stack", lecturer\_faculty = "CAMT", lecturer\_department = "SE", lecturer\_email = "tony@vr.camt.info", lecturer\_tel = "0893321101", lecturer\_approvement = 0 };

Semester semester1 = new Semester { semester\_id = 4, academic\_year = "2/2557" };

Course course1 = new Course { course\_id = 9, lecturer\_id = lecturer1, semester\_id = semester1, course\_name = "Test", course\_description = "Test na ja", course\_credit = 3 };

Student student1 = new Student { student\_id = 542115091, student\_name = "john conner", student\_username = "john", student\_password = "123456", student\_faculty = "CAMT", student\_department = "ANI", student\_address = "Thailand", student\_email = "ANI@gmail.com", student\_tel = "0833201787", student\_approvement = 1 };

Student student2 = new Student { student\_id = 542115092, student\_name = "jame conner", student\_username = "jame", student\_password = "123456", student\_faculty = "CAMT", student\_department = "SE", student\_address = "Thailand", student\_email = "SE@gmail.com", student\_tel = "0821330433", student\_approvement = 1 };

List<Student> studentList = new List<Student>{ student1, student2};

Testing testing1 = new Testing { testing\_id = 11, course\_id = course1, testing\_name = "Test", testing\_type = "2", testing\_amountQuestion = 1, testing\_score = 3, tesing\_random = 1, testing\_posting = 1, testing\_submitTime = null, student\_id = studentList };

string dateTime = "28/7/2014 15:04:15";

DateTime time = Convert.ToDateTime(dateTime);

AnswerSheet answerSheet1 = new AnswerSheet { answerSheet\_id = 25, testing\_id = testing1, student\_id = student1, answerSheet\_score = 3, answerSheet\_time = time, answerSheet\_submitResult = 0, answerSheet\_isChecked = 1 };

List<AnswerSheet> answerSheetListExpected1 = new List<AnswerSheet> { answerSheet1 };

List<AnswerSheet> answerSheetListExpected2 = new List<AnswerSheet>();

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 55.1 | This case tests to get the Answer object information from the database, which answerSheet\_id is existed in. | answerSheet\_id = 25 | CollectionAssert.AreEqual(answerSheetListExpected, actual); | answerSheetListExpected1 | answerSheetListExpected1 | Pass |
| 55.2 | This case tests to get the Answer object information from the database, which answerSheet\_id is not existed in. | answerSheet\_id = 100 | CollectionAssert.AreEqual(answerSheetListExpected, actual); | answerSheetListExpected2 | answerSheetListExpected2 | Pass |

## **UTC-56** **updateAnswerScore(answerSheetId : int, answerPoint : Nullable<Double>) : bool**

**Package:** Repository

**Class:** AnswerRepository

**Unit Test Case-11:** insertLecturerInfo(lecId : int, lecUsername: string, lecPassword : **stirng, lecName : stirng, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string): bool**

**Description:** The test case is used for testing 11 method in LecturerRepository. This method is update Answer information into the database.

**Prerequisites or Test Data**

:None

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Assertion** | **Expected Result** | **Actual**  **Output** | **Pass/Fail** |
| 56.1 | Test if Answer information provided is updated into the database successfully. Test for a successful case. | answer\_id = 39;  answer\_point = 3; | Assert.True (actual); | True | True | Pass |
| 56.2 | Test if Answer information provided is updated into the database successfully. Test for an unsuccessful case: providing answer\_Id that do not have in the database. | answer\_id = 0;  answer\_point = 3; | Assert.True (actual); | True | True | Pass |