SmartQuiz

Test Plan Document

By

Mr. Chaichan Suttee   542115016

Mr. Supawit kamjina   542115064

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

_____

Mr.Jayakrit Hirisajja

Document History

Document Name	Version	Status	Date	Viewable	Reviewer	Responsible
Documents						
Smart Quiz Test Record_v1.0.docx	Create Record UTC01 - 115 Create Record STC 01-36	Release	6 July 2015	CS, SK, JH	CS,SK	CS
Smart Quiz Test Record_v2.0.docx	Create Record UTC-116 - 129	Release	16 July 2015	CS, SK, JH	CS,SK	CS

*CS = Chaichan Suttee

*SK = Supawit Kamjina

*JH = Jayakrit Hirisajja

Table of Contents

Appendix A

1. Admin

Document Name	Smart Quiz– Test plan _v2.0.docx	Owner	CS,SK	Page	3 / 209
Document Type	Test Record document	Release Date	16 July 2015	Print Date	

1.1

Column Name	Data type	Value
admin_id	int	1
admin_username	varchar(16)	admin
admin_password	varchar(16)	123456
admin_name	varchar(255)	admin one

2. Lecturer

2.1

Column Name	Data type	Value
lecturer_id	int	1
lecturer_username	varchar(16)	lecturerone
lecturer_password	varchar(16)	123456
lecturer_name	varchar(255)	lecturer one
lecturer_faculty	varchar(255)	CAMT
lecturer_department	varchar(255)	SE
lecturer_email	varchar(255)	se01@vr.camt.info
lecturer_tel	varchar(32)	0831112222
lecturer_ approvement	bit	1

2.2

Column Name	Data type	Value
lecturer_id	int	2
lecturer_username	varchar(16)	lecturertwo
lecturer_password	varchar(16)	123456
lecturer_name	varchar(255)	lecturer two
lecturer_faculty	varchar(255)	CAMT
lecturer_department	varchar(255)	SE
lecturer_email	varchar(255)	se02@vr.camt.info
lecturer_tel	varchar(32)	0832221111
lecturer_ approvement	bit	0

3. Student

3.1

Column Name	Data type	Value
student_id	int	542115091
student_username	varchar(16)	studentone
student_password	varchar(16)	123456
student_name	varchar(255)	student one
student_faculty	varchar(255)	CAMT
student_department	varchar(255)	SE
student_address	varchar(255)	Chiang mai, Thailand
student_email	varchar(255)	se542115091@vr.camt.info
student_tel	varchar(32)	0931112222
student_ approvement	bit	1

3.2

Column Name	Data type	Value
student_id	int	542115092
student_username	varchar(16)	studenttwo
student_password	varchar(16)	123456
student_name	varchar(255)	student two
student_faculty	varchar(255)	CAMT
student_department	varchar(255)	SE
student_address	varchar(255)	Chiang mai, Thailand
student_email	varchar(255)	se542115092@vr.camt.info
student_tel	varchar(32)	0932221111
student_ approvement	bit	0

4. Sesmester

4.1

Column Name	Data type	Value
semester_id	int	1
academic_year	varchar(16)	1_2557

5. Course

5.1

Column Name	Data type	Value
course_id	int	1
lecturer_id	int	1
semester_id	int	1
course_name	varchar(255)	Math
course_credit	int	3
course_description	varchar(max)	Math is easy

6. CourseRegistration

6.1

Column Name	Data type	Value
courseRegistration_id	int	1
course_id	int	1
student_id	int	542115091

7. Testing

7.1

Column Name	Data type	Value
testing_id	int	1
course_id	int	1
testing_name	varchar(255)	Intro to Math
testing_type	varchar(1)	1
testing_amountQuestion	int	3
testing_score	float	9
testing_random	bit	1
testing_posting	bit	1
testing_submitTime	datetime2	2014-11-10 10:30:00
testing_student	varchar(max)	542115091

8. Question

8.1

Column Name	Data type	Value
question_id	int	1
testing _id	int	1
question_description	varchar(max)	1+1 = ?
question_choice	varchar(max)	3#$4#$2#$1
question_solution	varchar(1)	3
question_point	float	3
question_type	varchar(1)	1

8.2

Column Name	Data type	Value

question_id	int	2
testing _id	int	1
question_description	varchar(max)	1-1 = 0
question_choice	varchar(max)	True#$False
question_solution	varchar(1)	1
question_point	float	3
question_type	varchar(1)	2

8.3

Column Name	Data type	Value
question_id	int	3
testing_id	int	1
question_description	varchar(max)	2+1 = ?
question_choice	varchar(max)	Null
question_solution	varchar(1)	Null
question_point	float	3
question_type	varchar(1)	3

9. AnswerSheet

9.1

Column Name	Data type	Value
answerSheet_id	int	1
student_id	int	542115091
testing_id	int	1
answerSheet_score	float	6
answerSheet_time	datetime2	2014-10-20 10:20:15
answerSheet_submitResult	bit	1
answerSheet_isChecked	bit	1

10.Answer

10.1

Column Name	Data type	Value
answer_id	int	1
question_id	int	1
answerSheet_id	int	1
answer	varchar(max)	3
answer_point	float	3

10.2

Column Name	Data type	Value
answer_id	int	2
question_id	int	2
answerSheet_id	int	1
answer	varchar(max)	1
answer_point	float	3

10.3

Column Name	Data type	Value
answer_id	int	3
question_id	int	3
answerSheet_id	int	1
answer	varchar(max)	10
answer_point	float	0

11. Posts

Column Name	Data type	Value
post_id	int	1
course_id	int	1
poster_id	int	1
posts_name	varchar(max)	"post 1"
posts_description	varchar(max)	"post 1"
posts_client	varchar(1)	"3"
posts_recommend	varchar(1)	"0"

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   (Return True)

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

: Appendix A

Test Data

No	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel
TD1	542115093	"studentthree"	"123456"	"student three"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	"093222333 3"
TD2	542115093	null	"123456"	"student three"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	"093222333 3"
TD3	542115093	"studentthree"	null	"student three"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	"093222333 3"
TD4	542115093	"studentthree"	"123456"	null	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	"093222333 3"
TD5	542115093	"studentthree"	"123456"	"student three"	null	"SE"	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	"093222333 3"
TD6	542115093	"studentthree"	"123456"	"student three"	"CAMT"	null	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	"093222333 3"
TD7	542115093	"studentthree"	"123456"	"student three"	"CAMT"	"SE"	null	"se542115093 @vr.camt.info"	"093222333 3"
TD8	542115093	"studentthree"	"123456"	"student three"	"CAMT"	"SE"	"Chiang mai, Thailand"	null	"093222333 3"
TD9	542115093	"studentthree"	"123456"	"student three"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	null
TD10	null	null	null	null	null	null	null	null	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs student username is null	TD2	False
3	Test inputs student password is null	TD3	False
4	Test inputs student name is null	TD4	False
5	Test inputs student faculty is null	TD5	False
6	Test inputs student department is null	TD6	False
7	Test inputs student address is null	TD7	False
8	Test inputs student email is null	TD8	False
9	Test inputs student tel is null	TD9	False
10	Test inputs all parameters is null	TD10	False

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

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	

Prerequisites or Test Data

: Appendix A

Test Data

No	student_ id	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student_tel
ID1	542115091	"1234567"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091 @vr.camt.info"	"0931112222"
TD2	542115091	null	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091 @vr.camt.info"	"0931112222"
TD3	542115091	"123456"	null	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091 @vr.camt.info"	"0931112222"
TD4	542115091	"123456"	"student one"	null	"SE"	"Chiang mai, Thailand"	"se542115091 @vr.camt.info"	"0931112222"
TD5	542115091	"123456"	"student one"	"CAMT"	null	"Chiang mai, Thailand"	"se542115091 @vr.camt.info"	"0931112222"
TD6	542115091	"123456"	"student one"	"CAMT"	"SE"	null	"se542115091 @vr.camt.info"	"0931112222"
TD7	542115091	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	null	"0931112222"
TD8	542115091	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091 @vr.camt.info"	null
TD9	null	null	null	null	null	null	null	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

2	Test inputs student password is null	TD2	False
3	Test inputs student name is null	TD3	False
4	Test inputs student faculty is null	TD4	False
5	Test inputs student department is null	TD5	False
6	Test inputs student address is null	TD6	False
7	Test inputs student email is null	TD7	False
8	Test inputs student tel is null	TD8	False
9	Test inputs all parameters is null	TD9	False

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

Test Data

Objec t No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem ent
Obj1	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1
Obj2	0	null	"null	null	null	null	null	null"	null	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs student id is 542115091	542115091	Obj1
2	Test inputs student id is 3	3	Obj2

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

Test Data

Objec t No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem ent
Obj1	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1
Obj2	54211509 2	"studentt wo"	"123456"	"student two"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 092@vr.c amt.info"	"093222 1111"	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs student id is 542115091,542115092	"542115091,542115092"	[Obj1, Obj2]
2	Test inputs student id is null	null	[ ]

UTC-5 approveStudentStatus (stuId: string): string

Package: Repository

Document Name	Smart Quiz– Test plan _v2.0.docx	Owner	CS,SK	Page	14 / 209
Document Type	Test Record document	Release Date	16 July 2015	Print Date	

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

Test Data

No.	student_id
TD1	542115092

Test Case

Case	Description	Input	Expect Result
1	Test inputs student id is 542115091	542115092	"542115092"

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

Test Data

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj1	542115092	"studenttwo"	"123456"	"student two"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115092@vr.camt.info"	"0932221111"	0

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of student that do not available	-	[Obj1]

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

Test Data

Objec t No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem ent
Obj1	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"0931111 2222"	1

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of student that available	-	[Obj1]

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

Test Data

Objec t No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem ent
Obj1	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1
Obj2	0	null	"null	null	null	null	null	null"	null	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	student_username = "studentone", student_password = "123456"	Obj1
2	Test inputs all parameters is not correct data	student_username = "studentthree", student_password = "123456"	Obj2

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

Test Data

No.	student_id
TD1	542115091
TD2	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	student_username = "studentone"	TD1
2	Test inputs all parameters is not correct data	student_username = "studentthree"	TD2

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

Test Data

No.	student_id
TD1	542115091

Test Case

Case	Description	Input	Expect Result
1	Test inputs student id is 542115091	TD1	True

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 method insertLecturerInfo in LecturerRepository. This method is used for inserting lecturer information to the database.

Prerequisites or Test Data

: Appendix A

Test Data

No	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer_tel
TD1	3	"lecturerthree"	"123456"	"lecturer three"	"CAMT"	"SE"	"se03@vr. camt.info"	"0833201787 "
TD2	3	null	"123456"	"lecturer three"	"CAMT"	"SE"	"se03@vr. camt.info"	"0833201787 "
TD3	3	"lecturerthree"	null	"lecturer three"	"CAMT"	"SE"	"se03@vr. camt.info"	"0833201787 "
TD4	3	"lecturerthree"	"123456"	null	"CAMT"	"SE"	"se03@vr. camt.info"	"0833201787 "
TD5	3	"lecturerthree"	"123456"	"lecturer three"	null	"SE"	"se03@vr. camt.info"	"0833201787 "
TD6	3	"lecturerthree"	"123456"	"lecturer three"	"CAMT"	null	"se03@vr. camt.info"	"0833201787 "
TD7	3	"lecturerthree"	"123456"	"lecturer three"	"CAMT"	"SE"	null	"0833201787 "
TD8	3	"lecturerthree"	"123456"	"lecturer three"	"CAMT"	"SE"	"se03@vr. camt.info"	null
TD9	null	null	null	null	null	null	null	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs lecturer username is null	TD2	False
3	Test inputs lecturer password is null	TD3	False
4	Test inputs lecturer name is null	TD4	False
5	Test inputs lecturer faculty is null	TD5	False

6	Test inputs lecturer department is null	TD6	False
7	Test inputs lecturer email is null	TD7	False
8	Test inputs lecturer tel is null	TD8	False
9	Test inputs all parameters is null	TD9	False

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

Package:　　　　　Repository

Class:　　　　　LecturerRepository

Unit Test Case-12: updateLecturerInfo(lecId : int, lecPassword : string, lecName : string, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string) : bool

Description: The test case is used for testing method insertLecturerInfo in LecturerRepository. This method is used for updating lecturer information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

No	lecturer_ id	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _email	lecturer_tel
TD1	1	"1234567 "	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"
TD2	1	null	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"
TD3	1	"123456"	null	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"
TD4	1	"123456"	"lecturer one"	null	"SE"	"se01@vr.camt.info"	"0831112222"
TD5	1	"123456"	"lecturer one"	"CAMT"	null	"se01@vr.camt.info"	"0831112222"
TD6	1	"123456"	"lecturer one"	"CAMT"	"SE"	null	"0831112222"
TD7	1	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	null
TD8	null	null	null	null	null	null	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs lecturer password is null	TD2	False
3	Test inputs lecturer name is null	TD3	False
4	Test inputs lecturer faculty is null	TD4	False
5	Test inputs lecturer department is null	TD5	False
6	Test inputs lecturer email is null	TD6	False
7	Test inputs lecturer tel is null	TD7	False
8	Test inputs all parameters is null	TD8	False

UTC-13 veiwLecturerInfo(lecId : int) : Lecturer

Package: Repository

Class: LecturerRepository

Unit Test Case-13: veiwLecturerInfo(lecId : int) : Lecturer

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

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone "	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1
Obj2	3	null	null	null	null	null	null	null	0

Test Case

Case	Description	Input	Expect Result
1	Test input lecturer id that exist in database	1	Obj1
2	Test input lecturer id that does not exist in database	3	Obj2

UTC-14 approveLecturerStatus(lecId : int) : string

Package: Repository

Class: LecturerRepository

Unit Test Case-14: approveLecturerStatus(lecId : int) : Lecturer

Description: The test case is used for testing approveLecturerStatus method in LecturerRepository. This method is used for approving lecturer_approvement information into the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	lecturer_id
TD1	2

Test Case

Case	Description	Input	Expect Result
1	Test input lecturer id is 2	TD1	"2"

UTC-15 viewLecturerNotAvailable() : List<Lecturer>

Package: Repository

Class: LecturerRepository

Unit Test Case-15: viewLecturerNotAvailable() : List<Lecturer>

Description: The test case is used for testing viewLecturerNotAvailable method in LecturerRepository. This method is used for viewing a list of lecturer information that have approvement_status equal false from the database.

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	2	"lecturertwo "	"123456"	"lecturer two"	"CAMT"	"SE"	"se02@vr. camt.info"	"083222 1111 "	0

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of lecturer that do not available	-	[Obj1]

UTC-16 viewLecturerAvailable() : List<Lecturer>

Package: Repository

Class: LecturerRepository

Unit Test Case-16: viewLecturerAvailable() : List<Lecturer>

Description: The test case is used for testing viewLecturerAvailable method in LecturerRepository. This method is used for viewing a list of lecturer information that have approvement_status equal true from the database.

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone "	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of lecturer that do not available	-	[Obj1]

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

Package: Repository

Class: LecturerRepository

Unit Test Case-17: viewLecturerByUsernamePassword(lecUsername : string, lecPassword : string) : Lecturer

Description: The test case is used for testing viewLecturerByUsernamePassword method in LecturerRepository. This method is used for viewing lecturer information from the database by using lecturer username and lecturer password for searching.

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone "	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1
Obj2	3	null	null	null	null	null	null	null	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	lecturer_username = "lecturerone", lecturer_password = "123456"	Obj1
2	Test inputs all parameters is not correct data	lecturer_username = "lecturerthree", lecturer_password = "123456"	Obj2

UTC-18 viewLecturerByUsername(lecUsername : string) : int

Package: Repository

Class: LecturerRepository

Unit Test Case-18: viewLecturerByUsername(lecUsername : string) : Lecturer

Description: The test case is used for testing viewLecturerByUsername method in LecturerRepository. This method is used for viewing lecturer id from the database by using lecturer username for searching.

Prerequisites or Test Data

: Appendix A

Test Data

No.	lecuturer_id
TD1	1
TD2	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	lecturer_username = "lecturerone"	TD1
2	Test inputs all parameters is not correct data	lecturer _username = "lecturerthree"	TD2

UTC-19 deleteLecturer(lecturerId : int) : bool   (Return True)

Package: Repository

Class: LecturerRepository

Unit Test Case-19: deleteLecturer(lecturerId : int) : bool

Description: The test case is used for testing deleteLecturer method in LecturerRepository. This method is used for deleting lecuter information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	lecturer_id
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs lecturer id is 1	TD1	True

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

Package: Repository

Class: AdministratorRepository

Unit Test Case-20: viewAdminByUsernamePassword(adminUsername : string, adminPassword : string) : Admin

Description: The test case is used for testing viewAdminByUsernamePassword method in AdministratorRepository. This method is used for viewing administrator information from the database by using admin username and admin password for searching.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	admin_id	admin_username	admin _password	admin_name
Obj1	1	"admin"	"123456"	"admin one"
Obj2	0	null	null	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	admin_username = "admin", admin_password = "123456"	Obj1
2	Test inputs all parameters is not correct data	admin _username = "eiei", admin _password = "123456"	Obj2

UTC-21 viewAdminByUsername(adminUsername : string) : int

Package: Repository

Class: AdministratorRepository

Unit Test Case-21: viewAdminByUsername(adminUsername : string) : Admin

Description: The test case is used for testing viewAdminByUsername method in AdministratorRepository. This method is used for viewing admin id from the database by using admin username for searching.

Prerequisites or Test Data

: Appendix A

Test Data

No.	admin_id
TD1	1
TD2	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	admin_username = "admin"	TD1
2	Test inputs all parameters is not correct data	admin_username = "eiei"	TD2

UTC-22 viewSemesterInfo(int semId) : Semester

Package: Repository

Class: SemesterRepository

Unit Test Case-22: viewSemesterInfo(int semId) : Semester

Description: The test case is used for testing method viewSemesterInfo in SemesterRepository. This method is used for view Semester information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

Object No	semester_id	academic_year
Obj1	1	"1/2557"
Obj2	2	null

Test Case

Case	Description	Input	Expect Result
1	Test input semester id that exist in database	1	Obj1
2	Test input semester id that does not exist in database	2	Obj2

UTC-23 viewAllSemester() : List<Semester>

Package: Repository

Class: SemesterRepository

Unit Test Case-23: viewAllSemester() : List<Semester>

Description: The test case is used for testing method viewAllSemester in SemesterRepository. This method is used for view a list of Semester information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

Object No	semester_id	academic_year
Obj1	1	"1/2557"

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of semester	-	[Obj1]

UTC-24 insertSemester(academicYear : string) : bool (Return True)

Package: Repository

Class: SemesterRepository

Unit Test Case-24: insertSemester(academicYear : string) : bool

Description: The test case is used for testing method insertSemester in
SemesterRepository. This method is used for inserting semester
information to the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	academic_year
TD1	"2/2557"
TD2	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs academic_year is null	TD2	False

UTC-25 updateSemester(semesterId : int, academicYear : string) : bool (Return True)

Package: Repository

Class: SemesterRepository

Unit Test Case-25: updateSemester(semesterId : int, academicYear : string) : bool

Description: The test case is used for testing method updateSemester in SemesterRepository. This method is used for updating semester information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	semester_id	academic_year
TD1	1	"2/2557"
TD2	1	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs academic_year is null	TD2	False

UTC-26 deleteSemester(semesterId : int) : bool (Return True)
Package: Repository

Class: SemesterRepository

Unit Test Case-26: deleteSemester(semesterId : int) : bool

Description: The test case is used for testing method deleteSemester in SemesterRepository. This method is used for deleting semester information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	semester_id
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs semester id is one	TD1	True

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

Package: Repository

Class: CourseRepository

Unit Test Case-27: viewCourseInfoByLecturerId (lecId : int, semesterId : int): List<Course>

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

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone "	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"
Obj4	0	null	null	null	0	null

Test Case

Case	Description	Input	Expect Result
1	Test input lecturer id and semester id that have course exist in database	lecturer_id = 1, semester_id = 1	[Obj3]
2	Test input lecturer id and semester id that do not have course exist in database	lecturer_id = 1, semester_id = 2	[Obj4]

UTC-28 viewCourseInfoByCourseId(course_id: int): Course

Package: Repository

Class: CourseRepository

Unit Test Case-28: viewCourseInfoByCourseId(course_id: int) : Course

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

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"
Obj4	0	null	null	null	0	null

Test Case

Case	Description	Input	Expect Result
1	Test input course id that exist in database	course_id = 1	Obj3
2	Test input course id that does not exist in database	course_id = 2	Obj4

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

Package: Repository

Class: CourseRepository

Unit Test Case-29: viewCoursebySemesterId(semId : int) : List<Course>

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

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"
Obj4	0	null	null	null	0	null

Test Case

Case	Description	Input	Expect Result
1	Test input course id that exist in database	semester_id = 1	[Obj3]
2	Test input course id that does not exist in database	semester_id = 2	[Obj4]

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

Package: Repository

Class: CourseRepository

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

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

: Appendix A

Test Data

No.	lecturer_id	semester_id	course_name	course_credit	course_description
TD1	1	1	"C++ langguage"	3	"C++ is good"
TD2	0	1	"C++ langguage"	3	"C++ is good"
TD3	1	0	"C++ langguage"	3	"C++ is good"
TD4	1	1	null	3	"C++ is good"
TD5	1	1	"C++ langguage"	3	"C++ is good"
TD6	0	0	null	3	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs lecturer id is zero	TD2	False
3	Test inputs lecturer id is zero	TD3	False
4	Test inputs lecturer id is null	TD4	False
5	Test inputs lecturer id is null	TD5	False
6	Test inputs all parameters is incorrect data	TD6	False

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

Package: Repository

Class:	CourseRepository

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

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

Test Data

No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
TD1	1	1	1	"Math"	2	"Math is easy"
TD2	1	0	1	"Math"	3	"Math is easy"
TD3	1	1	0	"Math"	3	"Math is easy"
TD4	1	1	1	null	3	"Math is easy"
TD5	1	1	1	"Math"	3	null
TD6	1	0	0	null	3	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs lecturer id is zero	TD2	False
3	Test inputs lecturer id is zero	TD3	False
4	Test inputs lecturer id is null	TD4	False
5	Test inputs lecturer id is null	TD5	False
6	Test inputs lecturer id, semester id, course name, and course description is incorrect data.	TD6	False

UTC-32 deleteCourse(courseId int): bool   (Return True)

Package:	Repository
Class:	CourseRepository

Unit Test Case-32: deleteCourse(courseId int) : bool

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

Test Data

No.	course_id
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs course id is one	TD1	True

UTC-33 viewPreviousCourse(int lecturerId): Course

Package: Repository

Class: CourseRepository

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

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

Test Data

Object No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"
Obj4	0	null	null	null	0	null

Test Case

Case	Description	Input	Expect Result
1	Test input lecturer id that have course exist in database	lecturer_id = 1	Obj3
2	Test input course id that have course does not exist in database	lecturer_id = 3	Obj4

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

Package: Repository

Class: CourseRegisRepository

Unit Test Case-34: viewStudentInCourse(courseId: int) : List<CourseRegistration>

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

Test Data

Object No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvement
Obj4	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1

Object No.	courseRegistration_id	course_id	student_id
Obj5	1	Obj3	Obj4
Obj6	0	null	null

Test Case

Case	Description	Input	Expect Result
1	Test input course id that exist in database	course_id = 1	[Obj5]
2	Test input course id that does not exist in database	course_id = 2	[Obj6]

UTC-35 viewCourseFromStudent(student_id: int): List<CourseRegistration>

Package:	Repository	
Class:	CourseRegisRepository	

Unit Test Case-35: viewCourseFromStudent(student_id: int)
                        List<CourseRegistration>

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

Test Data

Object No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem

											ent
Obj4	542115091	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"		1

Object No.	courseRegistration_id	course_id	student_id
Obj5	1	Obj3	Obj4
Obj6	0	null	null

Test Case

Case	Description	Input	Expect Result
1	Test input student id that exist in database	student_id = 542115091	[Obj5]
2	Test input student id that does not exist in database	student_id = 542115093	[Obj6]

UTC-36 insertStudentsInCourse (courseId: int, studentIdList: List<int>): bool   (Return True)

Package: Repository

Class: CourseRegisRepository

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

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

Test Data

No.	student_id
TD1	542115092
TD2	542115093

No.	course_id
TD3	1
TD4	2

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	course_id = TD3, studentIdList = [TD1]	True
2	Test inputs student_id id is incorrect data	course_id = TD3, studentIdList = [TD2]	False
3	Test inputs course id is incorrect data	course_id = TD4, studentIdList = [TD1]	False

UTC-37 deleteStudentInCourse (courseId: int, studentIdList: int): bool (Return True)

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

Test Data

No.	student_id
TD1	542115092

No.	course_id
TD2	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	course_id = TD2, studentIdList = [TD1]	True

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 (Return True)

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>): bool

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

: Appendix A

Test Data

No.	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_submit Time	testing_ student
TD1	1	"Quiz Math 1"	"1"	1	3	1	1	null	"542115091"
TD2	2	"Quiz Math 1"	"1"	1	3	1	1	null	"542115091"
TD3	1	null	"1"	1	3	1	1	null	"542115091"
TD4	1	"Quiz Math 1"	null	1	3	1	1	null	"542115091"
TD5	1	"Quiz Math 1"	"1"	1	3	1	1	null	"null
TD6	1	null	null	null	null	null	null	null	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs course id is incorrect data	TD2	False
3	Test inputs testing name is null	TD3	False
4	Test inputs testing type is null	TD4	False
5	Test inputs testing student is null	TD5	False
6	Test inputs all parameters is null	TD1	False

UTC-39 viewPreviousTestingId(courseId: int): int

Package: Repository

Class: TestingRepository

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

Test Data

No.	course_id
TD1	1
TD2	2

Test Case

Case	Description	Input	Expect Result
1	Test inputs course id that have existing of testing in database	TD1	testing_id = 1
2	Test inputs course id that does not exist in database	TD2	testing_id = 0

UTC-40 viewTestingByTestingId(testingId: int): Testing

Package: Repository

Class: TestingRepository

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

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

Test Data

Object No.	lecturer_id	lecturer _username	lecturer _password	lecturer _name	lecturer _faculty	lecturer _department	lecturer _email	lecturer _tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931111 2222"	1

No.	testing_id	course_id	testing_name	testing_type	testing_amount Question	testing_score	testing_random	testing_posting	testing_submit Time	testing_student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]
Obj6	0	null	null	null	0	0	0	0	null	null

Test Case

Case	Description	Input	Expect Result
1	Test input testing id that exist in database	testing_id = 1	Obj5
2	Test input testing id that does not exist in database	course_id = 2	Obj6

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

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.camt.info"	"093111 2222"	1

No.	testing_id	course_id	testing_name	testing_type	testing_amount Question	testing_score	testing_random	testing_posting	testing_submit Time	testing_student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]

Test Case

Case	Description	Input	Expect Result
1	Test inputs course id that have existing of testing in database	course_id = 1	[Obj5]
2	Test inputs course id that does not exist in database	course_id = 2	[ ]

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   (Return True)

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) : bool

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

Test Data

No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_student
TD1	1	1	" Math 2"	"1"	3	9	1	1	"542115091"
TD2	1	2	"Intro to Math"	"1"	3	9	1	1	"542115091"
TD3	1	1	null	"1"	3	9	1	1	"542115091"
TD4	1	1	"Intro to Math"	null	3	9	1	1	"542115091"
TD5	1	1	"Intro to Math"	"1"	3	9	1	1	null
TD6	1	0	null	null	3	9	1	1	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs course id is incorrect data	TD2	False
3	Test inputs testing name is null	TD3	False
4	Test inputs testing type is null	TD4	False
5	Test inputs testing student is null	TD5	False
6	Test inputs course id is incorrect data and testing name, testing type, and testing student is null	TD6	False

UTC-43 viewListTestingThatPost():List<Testing>

Package: Repository

Class: TestingRepository

Unit Test Case-43: viewListTestingThatPost():List<Testing>

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

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone "	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Objec t No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem ent
Obj4	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1

No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_submit Time	testing_ student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of testing that are posting status equally one	-	[Obj5]

UTC-44 updateStudentInTesting(testingId int, testing_student string): bool

Package: Repository

Class: TestingRepository

Unit Test Case-44: updateStudentInTesting(testingId int, testing_student string): bool

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

Test Data

No.	testing_id	student_id
TD1	1	"542115091,542115092"
TD2	1	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs student id is null	TD2	False

UTC-45 insertQuestion(testing_id : int, listQuestion : List<Question>) : bool (Return True)

Package: Repository

Class: QuestionRepository

Unit Test Case-45: insertQuestion(testing_id : int, listQuestion : List<Question>) : bool

Description: The test case is used for testing insertQuestion method in QuestionRepository. This method is used for inserting a question

Prerequisites or Test Data

: Appendix A

Test Data

No.	testing_id
TD1	1
TD2	2

Object No.	question_type	question_description	question_choice	question_solution	question_point
Obj1	"3"	"Question 4"	""	""	3
Obj2	null	"Question 4"	""	""	3
Obj3	3"	null	""	""	3

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1, [Obj1]	True
2	Test inputs testing id that does not exist in database	TD2, [Obj1]	False
3	Test inputs question type is null	TD1, [Obj2]	False
4	Test inputs question description is null	TD1, [Obj3]	False

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

Package: Repository

Class: QuestionRepository

Unit Test Case-46: viewQuestionbyTestingId(testing : Testing) : List<Question>

Description: The test case is used for testing viewQuestionbyTestingId method in QuestionRepository. This method is used for viewing a list of question information in a testing.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_submit Time	testing_ student
Obj1	1	null	null	null	0	0	0	0	null	null
Obj2	1	null	null	null	0	0	0	0	null	null
Obj3	2	null	null	null	0	0	0	0	null	null

Object No.	question _id	testing_id	question_type	question_description	question_choice	question_solution	question_point
Obj4	1	Obj1	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	3
Obj5	2	Obj1	"2"	"1-1 = 0"	"True#$False"	"1"	3
Obj6	3	Obj1	"3"	"2+1 = ?"	""	""	3

Test Case

Case	Description	Input	Expect Result
1	Test input testing id that have questions exist in database	Obj2	[Obj4, Obj5, Obj6]
2	Test input testing id that does not exist in database	Obj3	[ ]

UTC-47 updateQuestioninfo(testing_id : int, listQuestion : List<Question>) : bool   (Return True)

Package: Repository

Class: QuestionRepository

Unit Test Case-47: updateQuestioninfo(testing_id : int, listQuestion : List<Question>) : bool

Description: The test case is used for testing updateQuestioninfo method in QuestionRepository. This method is used for updating question information in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	testing_id
TD1	1
TD2	2

Object No.	question _id	question_type	question_description	question_choice	question_solution	question_point
Obj1	1	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	2
Obj2	1	null	"1+1 = ?"	"3#$4#$2#$1"	"3"	3
Obj3	1	"1"	null	"3#$4#$2#$1"	"3"	3
Obj4	1	null	null	"3#$4#$2#$1"	"3"	3

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1, [Obj1]	True
2	Test inputs testing id that does not exist in database	TD2, [Obj1]	False
3	Test inputs question type is null	TD1, [Obj2]	False
4	Test inputs question description is null	TD1, [Obj3]	False
5	Test inputs testing id that does not exist in database and question type, question description is null	TD2, [Obj4]	False

UTC-48 viewQuestionByQuestionId(int questionId) : Question

Package: Repository

Class: QuestionRepository

Unit Test Case-48: viewQuestionByQuestionId(int questionId) : Question

Description: The test case is used for testing updateQuestioninfo method in QuestionRepository. This method is used for viewing question information in database.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_submit Time	testing_ student
Obj1	1	null	null	null	0	0	0	0	null	null

Object No.	question _id	testing_id	question_type	question_description	question_choice	question_solution	question_point
Obj2	1	Obj1	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	3
Obj3	0	null	null	null	null	null	0

Test Case

Case	Description	Input	Expect Result
1	Test input question id that exist in database	question_id =1	Obj2
2	Test input question id that does not exist in database	question_id=4	Obj3

UTC-49 insertAnswerSheet(studentId : int, testingId : int, answerSheetScore : Nullable<double>, submitTime : DateTime, submit_result : Nullable<int>, isCheck : int) : bool   (Return True)

Package: Repository

Class: AnswerSheetRepository

Unit Test Case-49: insertAnswerSheet(studentId : int, testingId : int, answerSheetScore : Nullable<double>, submitTime : DateTime, submit_result : Nullable<int>, isCheck : int) : bool

Description: The test case is used for testing insertAnswerSheet method in AnswerSheetRepository. This method is used for inserting answer sheet information in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	student_id	testing_id	answerSheet_score	answerSheet_time	answerSheet_submitResult	answerSheet_isChecked
TD1	542115091	1	6	DateTime.Now	1	0
TD2	0	1	6	DateTime.Now	1	0
TD3	542115091	0	6	DateTime.Now	1	0
TD4	0	0	6	DateTime.Now	1	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs student id that does not exist in database	TD2	False
3	Test inputs testing id that does not exist in database	TD3	False
4	Test inputs student id and testing id that does not exist in database	TD4	False

UTC-50 viewPreviousAnswerSheetId(studentId : int) : int

Package: Repository

Class: AnswerSheetRepository

Unit Test Case-50: viewPreviousAnswerSheetId(studentId : int) : int

Description: The test case is used for testing viewPreviousAnswerSheetId method in AnswerSheetRepository. This method is used for viewing a latest answer sheet that was created by student.

Prerequisites or Test Data

: Appendix A

Test Data

No.	student_id
TD1	542115091

Test Case

Case	Description	Input	Expect Result
1	Test inputs student id that have answer sheet exist in database	TD1	1

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

Package: Repository

Class: AnswerSheetRepository

Unit Test Case-51: viewListAnswerSheetBytestingId(testingId : int) : List<AnswerSheet>

Description: The test case is used for testing viewListAnswerSheetBytestingId method in AnswerSheetRepository. This method is used for viewing a list of answer sheet in testing.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931111 2222"	1

No.	testing_id	course_id	testing_name	testing_type	testing_amountQuestion	testing_score	testing_random	testing_posting	testing_submitTime	testing_student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]

No.	Answersheet_id	student_id	testing_id	answerSheet_score	answerSheet_time	answerSheet_submitResult	answerSheet_isChecked
Obj6	1	Obj4	Obj5	6	"2014-10-20 10:20:15"	1	1

Test Case

Document Name	Smart Quiz– Test plan _v2.0.docx	Owner	CS,SK	Page	64 / 209
Document Type	Test Record document	Release Date	16 July 2015	Print Date	

Case	Description	Input	Expect Result
1	Test inputs testing id that have answer sheet exist in database	testing_id = 1	[Obj6]
2	Test inputs testing id is incorrect data.	testing_id = 2	[ ]

UTC-52 viewAnswerSheetInfo(answerSheetId : int) : AnswerSheet

Package: Repository

Class: AnswerSheetRepository

Unit Test Case-52: viewAnswerSheetInfo(answerSheetId : int) : AnswerSheet

Description: The test case is used for testing viewAnswerSheetInfo method in AnswerSheetRepository. This method is used for viewing answer sheet information in database.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"093111 2222"	1

No.	testing_id	course_id	testing_name	testing_type	testing_amount Question	testing_score	testing_random	testing_posting	testing_submit Time	testing_student
Obj5	1	Obj3	"Intro to	"1"	3	9	1	1	"2014-11-10	[Obj4]

No.	Answersheet_ id	student_id	testing_id	answerSheet_ score	answerSheet_ time	answerSheet_ submitResult	answerSheet_ isChecked
			Math"				10:30:00"	

			Math"				10:30:00"

No.	Answersheet_ id	student_id	testing_id	answerSheet_ score	answerSheet_ time	answerSheet_ submitResult	answerSheet_ isChecked
Obj6	1	Obj4	Obj5	6	"2014-10-20 10:20:15"	1	1
Obj7	0	null	null	0	null	0	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs answer sheet id that exist in database	answerSheet_id = 1	Obj6
2	Test inputs answer sheet id that does not exist in database	answerSheet_id = 2	Obj7

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

Package: Repository

Class: AnswerSheetRepository

Unit Test Case-53: updateAnswerSheetScore(answerSheetId : int, score Nullable<double>)
: bool

Description: The test case is used for testing updateAnswerSheetScore method in AnswerSheetRepository. This method is used for updating answer sheet score in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	answerSheet_id	answerSheet_score
TD1	1	3

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

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

Package: Repository

Class: AnswerRepository

Unit Test Case-54: insertAnswer(questionId : List<int>, answerSheetId : int, answer : List<string>, answerPoint : List<Nullable<double>>) : bool

Description: The test case is used for testing insertAnswer method in AnswerRepository. This method is used for inserting answers in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	question_id
TD1	1
TD2	0

No.	answerSheet_id
TD3	1
TD4	0

No.	answer
TD5	"3"
TD6	null

No.	answer_point
TD7	3

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	[TD1],TD3,[TD5],[TD7]	True
2	Test inputs question id is zero.	[TD2],TD3,[TD5],[TD7]	False
3	Test inputs answer sheet id is zero.	[TD1],TD4,[TD5],[TD7]	False
4	Test inputs answer is null.	[TD1],TD3,[TD6],[TD7]	False
5	Test inputs question id, answer sheet id is zero and answer is null.	[TD2],TD4,[TD6],[TD7]	False

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

Package:	Repository
Class:	AnswerRepository
Unit Test Case-55:	viewAnswerListByAnswerSheetId (answerSheetId : int) : List<Answer>
Description:	The test case is used for testing viewAnswerListByAnswerSheetId method in AnswerRepository. This method is used for viewing a list of answer in answer sheet.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvement
Obj4	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1

No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_submit Time	testing_ student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]
Obj6	1	null	null	null	0	0	0	0	Null	null

Object No.	question _id	testing_id	question_type	question_description	question_choice	question_solution	question_point
Obj7	1	Obj6	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	3
Obj8	2	Obj6	"2"	"1-1 = 0"	"True#$False"	"1"	3
Obj9	3	Obj6	"3"	"2+1 = ?"	""	""	3

No.	answersheet_ id	student_id	testing_id	answerSheet_ score	answerSheet_ time	answerSheet_ submitResult	answerSheet_ isChecked
Obj10	1	Obj4	Obj5	6	"2014-10-20 10:20:15"	1	1

Object No.	answer_id	question_id	answer	answer_point	answersheet_

					id
Obj11	1	Obj7	"3"	3	Obj10
Obj12	2	Obj8	"1"	3	Obj10
Obj13	3	Obj9	"10"	0	Obj10

Test Case

Case	Description	Input	Expect Result
1	Test inputs answer sheet id that have answer exist in database	answerSheet_id = 1	[Obj11,Obj12,Obj13]
2	Test inputs answer sheet id that does not exist in database	answerSheet_id = 2	[ ]

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

Package: Repository

Class: AnswerRepository

Unit Test Case-56: updateAnswerScore(answerSheetId : int, answerPoint : Nullable<Double>) : bool

Description: The test case is used for testing updateAnswerScore method in AnswerRepository. This method is used for updating answers in database

Prerequisites or Test Data

: Appendix A

Test Data

No.	answer_id	Answer_point
TD1	3	3

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

UTC-57 insertStudent(student_id: int, student_username: string, student_password: string, student_name: string, student_faculty: string, student_department: string, student_address: string, student_email: string, student_tel: string): string

Package: Model

Class: Student

Unit Test Case-57: insertStudent(student_id: int, student_username: string, student_password: string, student_name: string, student_faculty: string, student_department: string, student_address: string, student_email: string, student_tel: string): string

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

Prerequisites or Test Data

: Appendix A

Test Data

No	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel
TD1	542115093	"studentthree"	"123456"	"student three"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	"0932223333"
TD2	542115091	"studentthree"	"123456"	"student three"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	"0932223333"
TD3	542115093	"studentone"	"123456"	"student three"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115093 @vr.camt.info"	"0932223333"

Test Case

Case	Description	Input	Verify data	Expect Result
1	Test inputs all parameters is correct data	TD1	Yes	"Add Student : " studentone already."
2	Test inputs duplicate student id in database	TD2	No	"Existing Student's id, please using different Student's id"
3	Test inputs duplicate student username in database	TD3	No	"Existing Student's username, please using different Student's username"

UTC-58 updateStudent(student_id: int, student_password: string, student_name: string, student_faculty: string, student_department: string, student_address: string, student_email: string, student_tel: string, confirmPassword: string,): string

Package: Model

Class: Student

Unit Test Case-58: updateStudent(student_id: int, student_password: string, student_name: string, student_faculty: string, student_department: string, student_address: string, student_email: string, student_tel: string, confirmPassword: string,): string

Description: The test case is used for testing insertStudent method in Student. This method is used for updating a student account in the database.

Prerequisites or Test Data

: Appendix A

Test Data

No	student_id	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	confirmPassword
TD1	542115091	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931112222"	"123456"
TD2	542115091	"1234567"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931112222"	"123456"
TD3	542115091	"1234567"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931112222"	"1234567"

Test Case

Case	Description	Input	Verify data	Expect Result
1	Test inputs all parameters is correct data	TD1	Yes	"Editing Information Successful"
2	Test inputs new student password and confirm password match with student password in database.	TD2	No	"Editing Information Successful"
3	Test inputs new student password and confirm password does not match student password in database.	TD3	No	"Password and Confirm password is not matched"

UTC-59 getStudentInfo(student_id: int): Student

Package: Model

Class: Student

Unit Test Case-59: getStudentInfo(student_id: int): Student

Description: The test case is used for testing getStudentInfo method in Student. This method is used for get a student information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem ent
Obj1	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1
Obj2	0	null	"null	null	null	null	null	null"	null	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs student id is 542115091	542115091	Obj1
2	Test inputs student id is 3	3	Obj2

UTC-60 approveStudentStatus (stuId: string): string

Package: Model

Class: Student

Unit Test Case-60: approveStudentStatus (stuId: string): string

Description: The test case is used for testing approveStudentStatus method in Student. This method is used for approving student_approvement

Prerequisites or Test Data

: Appendix A

Test Data

No.	student_id
TD1	542115092

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs student id is 542115091	542115092	Yes	"542115092"

UTC-61 getStudentNotAvailable (): List<Student>

Package: Model

Class: Student

Unit Test Case-61: getStudentNotAvailable (): List<Student>

Description: The test case is used for testing getStudentNotAvailable method in Student. 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

Test Data

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj1	542115092	"studenttwo"	"123456"	"student two"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115092@vr.camt.info"	"0932221111"	0

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of student that do not available	-	[Obj1]

UTC-62 getStudenAvailable (): List<Student>

Package: Model

Class: Student

Unit Test Case-62: getStudentAvailable (): List<Student>

Description: The test case is used for testing getStudentAvailable method in Student. 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

Test Data

Objec t No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem ent
Obj1	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of student that available	-	[Obj1]

UTC-63 deleteStudent (studentId: int): string

Package: Model

Class: Student

Unit Test Case-63: deleteStudent (studentId: int): string

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

Prerequisites or Test Data

: Appendix A

Test Data

No.	student_id
TD1	542115092

Test Case

Case	Description	Input	Expect Result
1	Test inputs student id is 542115092	TD1	"Delete Student successful"

UTC-64 viewStudentInCourseForUpdate(student_id: string): List<Student>

Package: Model

Class: Student

Unit Test Case-64: viewStudentInCourseForUpdate(student_id: string): List<Student>

Description: The test case is used for testing viewStudentInCourseForUpdate method in Student. This method is used for viewing converse a list of student information in course.

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem ent
Obj1	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1
Obj2	0	null	null	null	null	null	null	null	null	0

Test Case

Case	Description	Input	Expect Result
1	Test method for returning converse a list of student in course. Input data is student id that have in course.	"542115092"	[Obj1]
2	Test method for returning converse a list of student in course. Input data is student id that have in course.	"542115091,542115092"	[Obj2]

UTC-65 insertLecturer(lecturer_id: int, lecturer_username: string, lecturer_password: string, lecturer_name: string, lecturer_faculty: string, lecturer_department: string, lecturer_email: string, lecturer_tel: string): string

Package: Model

	Class:	Lecturer

Unit Test Case-65: insertLecturer(lecturer_id: int, lecturer_username: string, lecturer_password: string, lecturer_name: string, lecturer_faculty: string, lecturer_department: string, lecturer_email: string, lecturer_tel: string): string

Description: The test case is used for testing insertLecturer method in Lecturer. This method is used for inserting lecturer information to the database.

Prerequisites or Test Data

: Appendix A

Test Data

No	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer_tel
TD1	3	"lecturerthree"	"123456"	"lecturer three"	"CAMT"	"SE"	"se03@vr. camt.info"	"0833201787 "
TD2	1	"lecturerthree"	"123456"	"lecturer three"	"CAMT"	"SE"	"se03@vr. camt.info"	"0833201787 "
TD3	3	"lecturerone"	"123456"	"lecturer three"	"CAMT"	"SE"	"se03@vr. camt.info"	"0833201787 "

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs all parameters is correct data	TD1	Yes	"Registration Successful"
2	Test inputs duplicate lecturer id in database	TD2	No	"Existing Lecturer's id, please using different Lecturer's id"
3	Test inputs duplicate lecturer username in database	TD3	No	"Existing Lecturer's username, please using different Lecturer's username"

UTC-66 updateLecturerInfo(lecId : int, lecPassword : string, lecName : string, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string, confirmPass: string) : string

Package: Model

Document Name	Smart Quiz– Test plan _v2.0.docx	Owner	CS,SK	Page	80 / 209
Document Type	Test Record document	Release Date	16 July 2015	Print Date	

Class: Lecturer

Unit Test Case-66: updateLecturerInfo(lecId : int, lecPassword : string, lecName : string, lecFaculty : string, lecDepartment : string, lecEmail : string, lecTel : string, confirmPass: string) : string

Description: The test case is used for testing updateLecturerInfo method in Lecturer. This method is used for updating lecturer information to the database.

Prerequisites or Test Data

: Appendix A

Test Data

No	lecturer_id	lecturer _password	lecturer _name	lecturer _faculty	lecturer _department	lecturer _email	lecturer_tel	confirmPass
TD1	1	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	"123456"
TD2	1	"1234567"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	"123456"
TD3	1	"1234567"	null	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	"1234567"

Test Case

Case	Description	Input	Verify data	Expect Result
1	Test inputs all parameters is correct data	TD1	Yes	"Editing Information Successful"
2	Test inputs new lecturer password and confirm password match with student password in database.	TD2	No	"Editing Information Successful"
3	Test inputs new lecturer password and confirm password does not match student password in database.	TD3	No	"Password and Confirm password is not matched"

UTC-67 getLecturerInfo(lecturer_id: int): Lecuter

Package: Model

Class: Lecturer

Unit Test Case-67: getLecturerInfo(lecturer_id: int): Lecuter

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

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone "	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1
Obj2	3	null	null	null	null	null	null	null	0

Test Case

Case	Description	Input	Expect Result
1	Test input lecturer id that exist in database	1	Obj1
2	Test input lecturer id that does not exist in database	3	Obj2

UTC-68 approveLecturerStatus(lecId : int) : string

Package: Model

Class: Lecturer

Unit Test Case-68: approveLecturerStatus(lecId : int) : string

Description: The test case is used for testing approveLecturerStatus method in Lecturer. This method is used for approving lecturer_approvement information into the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	lecturer_id
TD1	2

Test Case

Case	Description	Input	Expect Result
1	Test input lecturer id is 2	TD1	"2"

UTC-69 getLecturerNotAvailable() : List<Lecturer>

Package: Repository

Class: Lecturer

Unit Test Case-69: getLecturerNotAvailable() : List<Lecturer>

Description: The test case is used for testing getLecturerNotAvailable method in Lecturer. This method is used for viewing a list of lecturer information that have approvement_status equal false from the database.

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	2	"lecturertwo"	"123456"	"lecturer two"	"CAMT"	"SE"	"se02@vr. camt.info"	"083222 1111 "	0

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of lecturer that do not available	-	[Obj1]

UTC-70 getLecturerAvailable() : List<Lecturer>

Package: Model

Class: Lecturer

Unit Test Case-70: getLecturerAvailable() : List<Lecturer>

Document Name	Smart Quiz– Test plan _v2.0.docx	Owner	CS,SK	Page	84 / 209
Document Type	Test Record document	Release Date	16 July 2015	Print Date	

Description: The test case is used for testing getLecturerAvailable method in
Lecturer. This method is used for viewing a list of lecturer
information that have approvement_status equal true from the
database.

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone "	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of lecturer that do not available	1	[Obj1]

UTC-71 deleteLecturer(lecturerId : int) : string

Package: Model

Class: Lecturer

Unit Test Case-71: deleteLecturer(lecturerId : int) : bool

Description: The test case is used for testing deleteLecturer method in
Lecturer. This method is used for deleting lecuter information in
the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	lecturer_id
TD1	2

Test Case

Case	Description	Input	Expect Result
1	Test inputs lecturer id is 2	TD1	"Delete Lecturer successful"

UTC-72 insertSemester(academicYear : string) : string

Package: Model

Class: SemesterRepository

Unit Test Case-72: insertSemester(academicYear : string) : string

Description: The test case is used for testing method insertSemester in Semester. This method is used for inserting semester information to the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	academic_year
TD1	"2/2557"

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs all parameters is correct data	TD1	YES	"Add semester successful"

UTC-73 updateSemester(semesterId : int, academicYear : string) : string

Package: Model

Class: Semester

Unit Test Case-73: updateSemester(semesterId : int, academicYear : string) : string

Description: The test case is used for testing method updateSemester in Semester. This method is used for updating semester information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	semester_id	academic_year
TD1	1	"2/2557"

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs all parameters is correct data	TD1	YES	"Update semester successful"

UTC-74 deleteSemester(semesterId : int) : string

Package: Model

Class: Semester

Unit Test Case-74: deleteSemester(semesterId : int) : string

Description: The test case is used for testing method deleteSemester in Semester. This method is used for deleting semester information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	semester_id
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs semester id is one	TD1	"Delete semester successful"

UTC-75 getAllSemester() : List<Semester>

Package: Model

Class: Semester

Unit Test Case-75: getAllSemester() : List<Semester>

Description: The test case is used for testing method getAllSemester in Semester. This method is used for view a list of Semester information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

Object No	semester_id	academic_year
Obj1	1	"1/2557"

Test Case

Case	Description	Input	Expect Result
1	Test method for returning list of semester	-	[Obj1]

UTC-76 getSemesterInfo(int semId) : Semester

Package: Model

Class: Semester

Unit Test Case-76: getSemesterInfo(int semId) : Semester

Description: The test case is used for testing method getSemesterInfo in Semester. This method is used for view Semester information in the database.

Prerequisites or Test Data

: Appendix A

Test Data

Object No	semester_id	academic_year
Obj1	1	"1/2557"
Obj2	2	null

Test Case

Case	Description	Input	Expect Result
1	Test input semester id that exist in database	1	Obj1
2	Test input semester id that does not exist in database	2	Obj2

UTC-77 getAllCoursebySemester(semId : int): List<Course>

Package: Model

Class: Course

Unit Test Case-77: getAllCoursebySemester (semId : int) : List<Course>

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer _username	lecturer _password	lecturer _name	lecturer _faculty	lecturer _department	lecturer _email	lecturer _tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"
Obj4	0	null	null	null	0	null

Test Case

Case	Description	Input	Expect Result
1	Test input course id that exist in database	semester_id = 1	[Obj3]
2	Test input course id that does not exist in database	semester_id = 2	[Obj4]

UTC-78 getCourseInfoByLecturerId (lecId : int, semesterId : int): List<Course>

Package: Model

Class: Course

Unit Test Case-78: getCourseInfoByLecturerId (lecId : int, semesterId : int): List<Course>

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

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone "	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"
Obj4	0	null	null	null	0	null

Test Case

Case	Description	Input	Expect Result
1	Test input lecturer id and semester id that have course exist in database	lecturer_id = 1, semester_id = 1	[Obj3]
2	Test input lecturer id and semester id that do not have course exist in database	lecturer_id = 1, semester_id = 2	[Obj4]

UTC-79 getCourseInfoByCourseId(course_id: int): Course

Package: Model

Class: Course

Unit Test Case-79: getCourseInfoByCourseId(course_id: int) : Course

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

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"
Obj4	0	null	null	null	0	null

Test Case

Case	Description	Input	Expect Result
1	Test input course id that exist in database	course_id = 1	Obj3
2	Test input course id that does not exist in database	course_id = 2	Obj4

UTC-80 insertCourse(lecturer_id: int, semester_id : int, course_name : string, course_credit : int, course_description : string): string

Package: Model

Class: Course

Unit Test Case-80: insertCourse(lecturer_id: int, semester_id : int, course_name : string, course_credit : int, course_description : string): string

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

Prerequisites or Test Data

: Appendix A

Test Data

No.	lecturer_id	semester_id	course_name	course_credit	course_description
TD1	1	1	"C++ langguage"	3	"C++ is good"

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs all parameters is correct data	TD1	Yes	"Adding Course Successful"

UTC-81 updateCourse(course_id: int, lecturer_id: int, semester_id : int, course_name : string, course_credit : int, course_description : string): string

Package: Model

Class: Course

Unit Test Case-81: updateCourse(course_id: int, lecturer_id: int, semester_id : int, course_name : string, course_credit : int, course_description : string): string

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

Prerequisites or Test Data

: Appendix A

Test Data

No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
TD1	1	1	1	"Math"	2	"Math is easy"

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs all parameters is correct data	TD1	Yes	"Editing Course successful"

UTC-82 deleteCourse(course_id int): string

Package: Model

Class: Course

Unit Test Case-82: deleteCourse(course_id int) : string

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

Prerequisites or Test Data

: Appendix A

Test Data

No.	course_id
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs course id is one	TD1	"Delete Course successful"

UTC-83 getPreviousCourse(int lecturer_id): Course

Package: Model

Class: Course

Unit Test Case-83: getPreviousCourse(int lecturer_id) : Course

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

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"
Obj4	0	null	null	null	0	null

Test Case

Case	Description	Input	Expect Result
1	Test input lecturer id that have course exist in database	lecturer_id = 1	Obj3
2	Test input course id that have course does not exist in database	lecturer_id = 3	Obj4

UTC-84 insertTesting (course_id: int, testing_name: string, testing_type: string, testing_amountQuestion: int, testing_score: double, testing_random: int, testing_posting: int, testing_student: string, submit_date: string): string

Package: Model

Document Name	Smart Quiz– Test plan _v2.0.docx	Owner	CS,SK	Page	98 / 209
Document Type	Test Record document	Release Date	16 July 2015	Print Date	

Class: Testing

Unit Test Case-84: insertTesting (course_id: int, testing_name: string, testing_type: string, testing_amountQuestion: int, testing_score: double, testing_random: int, testing_posting: int, testing_student: string, submit_date: string): string

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

Prerequisites or Test Data

: Appendix A

Test Data

No.	course_id	testing_name	testing_type	testing_amount Question	testing_score	testing_random	testing_posting	testing_submit Time	testing_student
TD1	1	"Quiz Math 1"	"1"	1	3	1	1	""	"542115091"

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs all parameters is correct data	TD1	Yes	Add Testing :Quiz Math 1 already.

UTC-85 viewQuestionbyTestingId(testing_id : int) : List<Question>

Package: Model

Class: Testing

Unit Test Case-85 viewQuestionbyTestingId(testing_id : int) : List<Question>

Description: The test case is used for testing viewQuestionbyTestingId method in Testing. This method is used for viewing a list of question information in a testing.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931112222"	1

Object No.	testing_id	course_id	testing_name	testing_type	testing_amount Question	testing_score	testing_random	testing_posting	testing_submit Time	testing_student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]

Object No.	question_id	testing_id	question_type	question_description	question_choice	question_solution	question_point
Obj6	1	Obj5	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	3
Obj7	2	Obj5	"2"	"1-1 = 0"	"True#$False"	"1"	3
Obj8	3	Obj5	"3"	"2+1 = ?"	""	""	3

Test Case

Case	Description	Input	Expect Result
1	Test input testing id that have questions exist in database	testing_id =1	[Obj6, Obj7, Obj8]

2	Test input testing id that does not exist in database	testing_id =2	[ ]	

UTC-86 viewTestingInfo(testingId: int): Testing

Package: Model

Class: Testing

Unit Test Case-86: viewTestinginfo (testingId: int): Testing

Description: The test case is used for testing viewTestinginfo method in Testing. This method is used for viewing testing information by using testing id in searching.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1

No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_submit Time	testing_ student
Obj5	1	Obj3	"Intro to	"1"	3	9	1	1	"2014-11-10	[Obj4]

			Math"						10:30:00"	
Obj6	0	null	null	null	0	0	0	0	null	null

Test Case

Case	Description	Input	Expect Result
1	Test input testing id that exist in database	testing_id = 1	Obj5
2	Test input testing id that does not exist in database	course_id = 2	Obj6

UTC-87 viewAllTestingbyCourseId(course_id: int): List<Testing>

Package: Model

Class: Testing

Unit Test Case-87: viewAllTestingbyCourseId(course_idd: int): List<Testing>

Description: The test case is used for testing viewAllTestingbyCourseId method in Testing. This method is used for viewing a list of testing information in a course.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931112222"	1

No.	testing_id	course_id	testing_name	testing_type	testing_amountQuestion	testing_score	testing_random	testing_posting	testing_submitTime	testing_student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]

Test Case

Case	Description	Input	Expect Result
1	Test inputs course id that have existing of testing in database	course_id = 1	[Obj5]
2	Test inputs course id that does not	course_id = 2	[ ]

	exist in database		

UTC-88 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): string

Package: Model

Class: Testing

Unit Test Case-88: 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) : string

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

Prerequisites or Test Data

: Appendix A

Test Data

No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_student
TD1	1	1	" Math 2"	"1"	3	9	1	1	"542115091"

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	"Editing Testing Successful"

UTC-89 ViewConverseStudentInTesting(): List<CourseRegistration>

Package: Model

Class: Testing

Unit Test Case-89: ViewConverseStudentInTesting(): List<CourseRegistration>

Description: The test case is used for testing ViewConverseStudentInTesting method in Testing. This method is used for viewing a list of student , which are converse of students in testing..

Prerequisites or Test Data

: Appendix A

Test Data

No.	testing_ id
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	[ ]

UTC-90 updateStudentInTesting(testing_id int, student_id string): string

Package: Model

Class: Testing

Unit Test Case-90: updateStudentInTesting(testing_id int, student_id string): string

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

Prerequisites or Test Data

: Appendix A

Test Data

No.	testing_id	student_id
TD1	1	"542115091,542115092"
TD2	1	null

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs all parameters is correct data	TD1	Yes	"Updating Student in the testing is successful"

UTC-91 viewChartInTesting(testing: Testing) : List<List<double>>

Package: Model

Class: Testing

Unit Test Case-91: viewChartInTesting(testing: Testing) : List<List<double>>

Description: The test case is used for testing viewChartInTesting method in Testing. This method is used for viewing chart of answer in the testing.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_submit Time	testing_ student
Obj1	1	null	null	null	0	0	0	0	null	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	Obj1	[[0.0, 0.0, 100.0, 0.0],[100.0, 0.0],[0]]

UTC-92 viewQuestionForStudent(testing :Testing) : List<Question>

Package: Model

Class: Testing

Unit Test Case-92: viewQuestionForStudent(testing :Testing) : List<Question>

Description: The test case is used for testing viewQuestionForStudent method in Testing. This method is used for viewing a list of questions in testing for students.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931112222"	1

Object No.	testing_id	course_id	testing_name	testing_type	testing_amountQuestion	testing_score	testing_random	testing_posting	testing_submitTime	testing_student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]

Object No.	question_id	testing_id	question_type	question_description	question_choice	question_solution	question_point
Obj6	1	Obj5	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	3
Obj7	2	Obj5	"2"	"1-1 = 0"	"True#$False"	"1"	3
Obj8	3	Obj5	"3"	"2+1 = ?"	""	""	3

Test Case

Case	Description	Input	Expect Result
1	Test input testing id that have	testing_id	[Obj6, Obj7, Obj8] or

	questions exist in database	=1	[Obj7, Obj6, Obj8] or [Obj8 Obj7, Obj6] or [Obj8, Obj6 Obj7] or [Obj7, Obj8, Obj6]
2	Test input testing id that does not exist in database	testing_id =2	[ ]

UTC-93 viewAllTestingbyStudentId(student_id: int, course_id: int): List<Testing>

Package: Model

Class: Testing

Unit Test Case-93: viewAllTestingbyStudentId (student_id: int, course_id: int): List<Testing>

Description: The test case is used for testing viewAllTestingbyStudentId method in Testing. This method is used for viewing a list of testing information in a student id.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement

| Obj4 | 542115091 | "studento ne" | "123456" | "student one" | "CAMT" | "SE" | "Chiang mai, Thailand" | "se542115 091@vr.c amt.info" | "093111 2222" | 1 |

No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_submit Time	testing_ student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]

Test Case

Case	Description	Input	Expect Result
1	Test inputs student id and course id that have existing of testing in database	student_id = 542115091, course_id = 1	[Obj5]
2	Test inputs student id or coursed that does not exist in database	student_id = 542115093, course_id =	[ ]

UTC-94 insertQuestion(course_id : int, listQuestion : List<Question>) : string

Package: Model

Class: Question

Unit Test Case-94: insertQuestion(testing_id : int, listQuestion : List<Question>) : string

Description: The test case is used for testing insertQuestion method in Question. This method is used for inserting a question

Prerequisites or Test Data

: Appendix A

Test Data

No.	course_id
TD1	1

Object No.	question_type	question_description	question_choice	question_solution	question_point
Obj1	"3"	"Question 4"	""	""	3

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs all parameters is correct data	TD1, [Obj1]	Yes	"Creating Testing Successful"

UTC-95 updateQuestioninfo(testing_id : int, listQuestion : List<Question>) : string

Package:	Model
Class:	Question
Unit Test Case-95:	updateQuestioninfo(testing_id : int, listQuestion : List<Question>) : string
Description:	The test case is used for testing updateQuestioninfo method in Question. This method is used for updating question information in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	testing_id
TD1	1

Object No.	question _id	question_type	question_description	question_choice	question_solution	question_point
Obj1	1	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	2

Test Case

Case	Description	Input	Verify Data	Expect Result
1	Test inputs all parameters is correct data	TD1, [Obj1]	Yes	"Editing Testing Successful"

UTC-96 viewQuestionInfo(int questionId) : Question

Package: Model

Class: Question

Unit Test Case-96: viewQuestionInfo(int questionId) : Question

Description: The test case is used for testing updateQuestioninfo method in Question. This method is used for viewing question information in database.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	testing_id	course_id	testing_name	testing_type	testing_amount Question	testing_score	testing_random	testing_posting	testing_submit Time	testing_student
Obj1	1	null	null	null	0	0	0	0	null	null

Object No.	question_id	testing_id	question_type	question_description	question_choice	question_solution	question_point
Obj2	1	Obj1	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	3
Obj3	0	null	null	null	null	null	0

Test Case

Case	Description	Input	Expect Result
1	Test input question id that exist in database	question_id =1	Obj2
2	Test input question id that does not exist in database	question_id=4	Obj3

UTC-97 viewListAnswerSheetBytestingId (testing_id: int) : List<AnswerSheet>

Package:	Model
Class:	AnswerSheet
Unit Test Case-97:	viewListAnswerSheetBytestingId (testing_id : int) : List<AnswerSheet>
Description:	The test case is used for testing viewListAnswerSheetBytestingId method in AnswerSheet. This method is used for viewing a list of answer sheet in testing.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931112222"	1

No.	testing_id	course_id	testing_name	testing_type	testing_amountQuestion	testing_score	testing_random	testing_posting	testing_submitTime	testing_student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]

UTC-98 viewAnswerByAnswerSheetId (answerSheet_id : int) : List<Answer>

Package:	Model
Class:	AnswerSheet
Unit Test Case-98:	viewAnswerByAnswerSheetId (answerSheet_id : int) : List<Answer>
Description:	The test case is used for testing viewAnswerByAnswerSheetId method in AnswerSheet. This method is used for viewing a list of answer in answer sheet.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer _username	lecturer _password	lecturer _name	lecturer _faculty	lecturer _department	lecturer _email	lecturer _tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	student_id	student_username	student_password	student_name	student_faculty	student_department	student_address	student_email	student_tel	student_approvement
Obj4	542115091	"studentone"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115091@vr.camt.info"	"0931111 2222"	1

No.	testing_id	course_id	testing_name	testing_type	testing_amount Question	testing_score	testing_random	testing_posting	testing_submit Time	testing_student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]
Obj6	1	null	null	null	0	0	0	0	Null	null

Object	question	testing_id	question_type	question_description	question_choice	question_solution	question_point

No.	_id						
Obj7	1	Obj6	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	3
Obj8	2	Obj6	"2"	"1-1 = 0"	"True#$False"	"1"	3
Obj9	3	Obj6	"3"	"2+1 = ?"	""	""	3

No.	answersheet_ id	student_id	testing_id	answerSheet_ score	answerSheet_ time	answerSheet_ submitResult	answerSheet_ isChecked
Obj10	1	Obj4	Obj5	6	"2014-10-20 10:20:15"	1	1

Object No.	answer_id	question_id	answer	answer_point	answersheet_ id
Obj11	1	Obj7	"3"	3	Obj10
Obj12	2	Obj8	"1"	3	Obj10
Obj13	3	Obj9	"10"	0	Obj10

Test Case

Case	Description	Input	Expect Result
1	Test inputs answer sheet id that have answer exist in database	answerSheet_id = 1	[Obj11,Obj12,Obj13]
2	Test inputs answer sheet id that does not exist in database	answerSheet_id = 2	[ ]

UTC-99 updateAnswerSheetScore(answerSheet_id : int, List<Answer> answerList) : string

Package:	Model
Class:	AnswerSheet
Unit Test Case-53:	updateAnswerSheetScore(answerSheet_id : int, List<Answer> answerList) : string
Description:	The test case is used for testing updateAnswerSheetScore method in AnswerSheet. This method is used for updating answer sheet score in database.

Prerequisites or Test Data

: Appendix A

Test Data

Objec t No.	lecturer_ id	lecturer _ username	lecturer _ password	lecturer _ name	lecturer _ faculty	lecturer _ department	lecturer _ email	lecturer _tel	lecturer_ approvement
Obj1	1	"lecturerone "	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Objec t No.	student_ id	student_ username	student_ password	student_ name	student_ faculty	student_ department	student_ address	student_ email	student _tel	student_ approvem ent
Obj4	54211509 1	"studento ne"	"123456"	"student one"	"CAMT"	"SE"	"Chiang mai, Thailand"	"se542115 091@vr.c amt.info"	"093111 2222"	1

No.	testing_ id	course_ id	testing_ name	testing_ type	testing_amount Question	testing_ score	testing_ random	testing_ posting	testing_submit Time	testing_ student
Obj5	1	Obj3	"Intro to Math"	"1"	3	9	1	1	"2014-11-10 10:30:00"	[Obj4]
Obj6	1	null	null	null	0	0	0	0	Null	null

Object No.	question _id	testing_id	question_type	question_description	question_choice	question_solution	question_point
Obj7	1	Obj6	"1"	"1+1 = ?"	"3#$4#$2#$1"	"3"	3
Obj8	2	Obj6	"2"	"1-1 = 0"	"True#$False"	"1"	3
Obj9	3	Obj6	"3"	"2+1 = ?"	""	""	3

No.	answersheet_ id	student_id	testing_id	answerSheet_ score	answerSheet_ time	answerSheet_ submitResult	answerSheet_ isChecked
Obj10	1	Obj4	Obj5	6	"2014-10-20 10:20:15"	1	1

Object No.	answer_id	question_id	answer	answer_point	answersheet_ id
Obj11	1	Obj7	"3"	3	Obj10
Obj12	2	Obj8	"1"	3	Obj10
Obj13	3	Obj9	"10"	0	Obj10

Test Case

Case	Description	Input	Expect Result
1	Test inputs answer sheet id that have answer exist in database	answerSheet_id = 1, answerList = Obj11,Obj12,Obj13]	"Marking point is successful"

UTC-100 insertAnswerSheet(studentId : int, testingId : int, questionIdList: List<int>, answerList: List<string>) : string

Package:	Model
Class:	AnswerSheet
Unit Test Case-100:	insertAnswerSheet(studentId : int, testingId : int, questionIdList: List<int>, answerList: List<string>) : string
Description:	The test case is used for testing insertAnswerSheet method in AnswerSheet. This method is used for inserting answer sheet information in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	student_id
TD1	542115091

No.	testing_id
TD2	1

No.	question_id
TD3	1
TD4	2
TD5	3

No.	answer
TD6	"3"
TD7	"1"
TD8	"10

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1,TD2,[TD3, TD4, TD5], [TD6, TD7, TD8]	"Take this testing already"

UTC-101 updateAnswerScore(answerId : int, answerPoint : Nullable<Double>) : bool (Return True)

Package:	Model
Class:	Answer

Unit Test Case-101: updateAnswerScore(answerId : int, answerPoint : Nullable<Double>) : bool

Description: The test case is used for testing updateAnswerScore method in Answer. This method is used for updating answers in database

Prerequisites or Test Data

: Appendix A

Test Data

No.	answer_id	Answer_point
TD1	3	3

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

UTC-102 updateAnswerScore(answerId : int, answerPoint : Nullable<Double>) : bool (Return True)

Package:	Model
Class:	Answer

Unit Test Case-101: updateAnswerScore(answerId : int, answerPoint : Nullable<Double>) : bool

Description: The test case is used for testing updateAnswerScore method in Answer. This method is used for updating answers in database

Prerequisites or Test Data

: Appendix A

Test Data

No.	answer_id	Answer_point
TD1	3	3

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

UTC-103 insertAnswer(questionId : List<int>, answerSheetId : int, answer : List<string>, answerPoint : List<Nullable<double>>) : bool (Return True)

Package:	Model
Class:	Answer
Unit Test Case-103:	insertAnswer(questionId : List<int>, answerSheetId : int, answer : List<string>, answerPoint : List<Nullable<double>>) : bool
Description:	The test case is used for testing insertAnswer method in Answer. This method is used for inserting answers in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	question_id
TD1	1

No.	answerSheet_id
TD2	1

No.	answer
TD3	"3"

No.	answer_point
TD4	3

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	[TD1],TD2,[TD3],[TD4]	True

UTC-104 insertPosts (courseId: int, posterId: int, postsName: string, postsDescription: string, postsClient: postsClient): bool (Return True)

Package:	Repository
Class:	PostsRepository
Unit Test Case-104:	insertPost (courseId: int, posterId: int, postsName: string, postsDescription: string, postsClient: postsClient): bool
Description:	The test case is used for testing insertPost method in PostsRepository. This method is used for inserting posts in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	courseId	posterId	postsName	postsDescription	postsClient
TD1	1	1	"post 2"	"post 2"	"2"
TD2	0	1	"post 2"	"post 2"	"2"
TD3	1	1	null	"post 2"	"2"
TD4	1	1	"post 2"	null	"2"
TD5	1	1	"post 2"	"post 2"	null
TD6	0	1	null	null	null

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True
2	Test inputs course id is zero	TD2	False
3	Test inputs posts name is null	TD3	False
4	Test inputs posts dscription is null	TD4	False
5	Test inputs posts client null	TD5	False
6	Test inputs posts name, posts dscription and post client is null, course id is zero	TD6	False

UTC-105 updatePostsRecom (postsId: int, postsRecommend: int): bool (Return True)

Package:	Repository
Class:	PostsRepository

Unit Test Case-105: updatePostRecom (postsId: int, postsRecommend: int): bool

Description: The test case is used for testing updatePostRecom method in PostsRepository. This method is used for updating posts for recommending in database.

Prerequisites or Test Data

: Appendix A

Test Data

Noo.	postId	postsRecommend
TD1	1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

UTC-106 deletePosts (postsId: int): bool (Return True)

Package: Repository

Class: PostsRepository

Unit Test Case-106: deletePosts (postsId: int): bool

Description: The test case is used for testing deletePosts method in PostsRepository. This method is used for deleting posts in database.

Prerequisites or Test Data

: Appendix A

Test Data

Noo.	postId
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

UTC-107 viewPostFromCourseId (courseId: int): List<Posts>

Package:	Repository
Class:	PostsRepository

Unit Test Case-107: viewPostFromCourseId (courseId: int): List<Posts>

Description: The test case is used for testing viewPostFromCourseId method in PostsRepository. This method is used for viewing a list of posts in course.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	post_id	course_id	poster_id	posts_name	posts_description	posts_client	posts_recommend
Obj4	1	Obj3	1	"post 1"	"post 1"	3	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs course id that has posts in course	course_id =1	[Obj4]
2	Test inputs course id that does not exist in database	course_id =2	[ ]

UTC-108 insertPosts (courseId: int, posterId: int, postsName: string, postsDescription: string, postsClient: postsClient): bool (Return True)

Package: Model

Class: Posts

Unit Test Case-108: insertPost (courseId: int, posterId: int, postsName: string, postsDescription: string, postsClient: postsClient): bool

Description: The test case is used for testing insertPost method in Posts. This method is used for inserting posts in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	courseId	posterId	postsName	postsDescription	postsClient
TD1	1	1	"post 2"	"post 2"	"2"

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

UTC-109 updatePostsRecom (postsId: int, postsRecommend: int): bool (Return True)

Package: Model

Class: Posts

Unit Test Case-109: updatePostRecom (postsId: int, postsRecommend: int): bool

Description: The test case is used for testing updatePostRecom method in Posts. This method is used for updating posts for recommending in database.

Prerequisites or Test Data

: Appendix A

Test Data

Noo.	postId	postsRecommend
TD1	1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

UTC-110 deletePosts (postsId: int): string

Package: Model

Class: PostsRepository

Unit Test Case-110: deletePosts (postsId: int): string

Description: The test case is used for testing deletePosts method in Posts. This method is used for deleting posts in database.

Prerequisites or Test Data

: Appendix A

Test Data

Noo.	postId
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	"Delete a posts complete"

UTC-111 viewPostFromCourseId (courseId: int): List<Posts>

Package: Model

Class: Posts

Unit Test Case-111: viewPostFromCourseId (courseId: int): List<Posts>

Description: The test case is used for testing viewPostFromCourseId method in Posts. This method is used for viewing a list of posts in course.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer _username	lecturer _password	lecturer _name	lecturer _faculty	lecturer _department	lecturer _email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr.camt.info"	"0831112222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	post_id	course_id	poster_id	posts_name	posts_description	posts_client	posts_recommend
Obj4	1	Obj3	1	"post 1"	"post 1"	3	0

Test Case

Case	Description	Input	Expect Result
1	Test inputs course id that has posts in course	course_id =1	[Obj4]
2	Test inputs course id that does not exist in database	course_id =2	[ ]

UTC-112 intsertPosts (postInfo: Stream): bool   (Return True)

Class: PostsManage

Unit Test Case-112: intsertPosts (postInfo: Stream): bool

Description: The test case is used for testing insertPost method in PostsManage. This method is used for inserting posts in database.

Prerequisites or Test Data

: Appendix A

Test Data

No.	courseId	posterId	postsName	postsDescription	postsClient
TD1	1	1	"post 2"	"post 2"	"2"

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

UTC-113 updatePostsRecom (postInfo: Stream): bool   (Return True)

Class: PostsManage

Unit Test Case-113: updatePostsRecom (postInfo: Stream): bool

Description: The test case is used for testing updatePostsRecom method in PostsManage. This method is used for updating posts recommend status in database.

Prerequisites or Test Data

: Appendix A

Test Data

Noo.	postId	postsRecomme nd
TD1	1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	True

UTC-114 deletePosts (postInfo: Stream): string

Class: PostsManage

Unit Test Case-114: deletePosts (postsInfo: int): string

Description: The test case is used for testing deletePosts method in PostsManage. This method is used for deleting posts in database.

Prerequisites or Test Data

: Appendix A

Test Data

Noo.	postId
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	TD1	"Delete a posts complete"

UTC-115 viewPostFromCourseId (postInfo: Stream): List<Posts>

Class: PostsManage

Unit Test Case-115: viewPostFromCourseId (postInfo: Stream): List<Posts>

Description: The test case is used for testing viewPostFromCourseId method in PostsManage. This method is used for viewing a list of posts in course.

Prerequisites or Test Data

: Appendix A

Test Data

Object No.	lecturer_id	lecturer_username	lecturer_password	lecturer_name	lecturer_faculty	lecturer_department	lecturer_email	lecturer_tel	lecturer_approvement
Obj1	1	"lecturerone"	"123456"	"lecturer one"	"CAMT"	"SE"	"se01@vr. camt.info"	"083111 2222"	1

Object No	semester_id	academic_year
Obj2	1	"1/2557"

Object No.	course_id	lecturer_id	semester_id	course_name	course_credit	course_description
Obj3	1	Obj1	Obj2	"Math"	3	"Math is easy"

Object No.	post_id	course_id	poster_id	posts_name	posts_description	posts_client	posts_recommend
Obj4	1	Obj3	1	"post 1"	"post 1"	3	0

No.	couse_id
TD1	1

Test Case

Case	Description	Input	Expect Result
1	Test inputs course id that has posts in course	TD1	[Obj4]

UTC-116 Ajax({url : "http://localhost:1580/Login_Logout.svc/login/"})

Package   None

Class: Login_Logout

Method: Post

Unit Test Case-116: Ajax({url : "http://localhost:1580/Login_Logout.svc/login/"})

Description: The test case is used for testing login to the system.

Prerequisites or Test Data

: Appendix A

Test Data

- Json input
  JI1: { username: "admin", password: "123456" }
  JI2: { username: "", password: "123456" }
  JI3: { username: "admin", password: "" }
  JI4: { username: "", password: "" }

- Json output
  JO1: {"1_admin"}
  JO2: {" Invalid Username and/or Password"}

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	JI1	JO1
2	Test inputs user username is empty	JI2	JO2
3	Test inputs user password is empty	JI3	JO2
4	Test inputs user username and password is empty	JI4	JO2

UTC-117 Ajax({url : "http://localhost:1580/ManageAccount.svc/viewLecturerNotAvailable/"})

Class: ManageAccount

Method: Get

Unit Test Case-117: Ajax({url : "http://localhost:1580/ManageAccount.svc/viewLecturerNotAvailable/"})

Description: The test case is used for testing about viewing inactive lecturers in the system.

Prerequisites or Test Data

: Appendix A

Test Data

- Json output
  JO1: [{ lecturer_id: 2, lecturer_username: "lecturertwo", lecturer_password: "123456", lecturer_name: "lecturer two", lecturer_faculty: "CAMT", lecturer_department: "SE", lecturer_email: "se02@vr.camt.info", lecturer_tel: "0832221111", lecturer_approvement: 0 }]

Test Case

Case	Description	Input	Expect Result
1	Testing about viewing inactive	-	JO1

Document Name	Smart Quiz– Test plan _v2.0.docx	Owner	CS,SK	Page	137 / 209
Document Type	Test Record document	Release Date	16 July 2015	Print Date	

	lecturers in the system.		

UTC-118 Ajax({url : "http://localhost:1580/ManageAccount.svc/ viewStudentNotAvailable/"})

Class: ManageAccount

Method: Get

Unit Test Case-118: Ajax({url : "http://localhost:1580/ManageAccount.svc/ viewStudentNotAvailable/"})

Description: The test case is used for testing about viewing inactive students in the system.

Prerequisites or Test Data

: Appendix A

Test Data

- Json output
  JO1: [{ student_id: 542115092, student _username: "studenttwo", student _password: "123456", student _name: "student two", student _faculty: "CAMT" , student _department: "SE", student_address: "Chiang mai, Thailand", student _email: "se542115092@vr.camt.info", student _tel: " 0932221111", student _approvement: 0 }]

Test Case

Case	Description	Input	Expect Result

1	Testing about viewing inactive students in the system.	-	JO1

UTC-119 Ajax({url : "http://localhost:1580/ManageAccount.svc/approveUser/"})

Package   Admin

Class: ManageAccount

Method: Post

Unit Test Case-118: Ajax({url : "http://localhost:1580/ManageAccount.svc/approveUser/"})

Description: The test case is used for testing about approving the user account.

Prerequisites or Test Data

: Appendix A

Test Data

- Json input
  JI1: { lecturer_id: 1 }
  JT2: { lecturer_id: 542115091}

- Json output
  JO1: { "Approve User id: 1 successful" }
  JO2: { "Approve User id: 542115091 successful" }

Test Case

Case	Description	Input	Expect Result
1	Testing about approval a lecturer account.	JT1	JO1
2	Testing about approval a student account.	JT2	JO2

UTC-120 Ajax({url : "http://localhost:1580/ManageAccount.svc/viewLecturerAvailable/"})

Package Admin

Class: ManageAccount

Method: Get

Unit Test Case-120: Ajax({url : "http://localhost:1580/ManageAccount.svc/viewLecturerAvailable/"})

Description: The test case is used for testing about viewing active lecturers in the system.

Prerequisites or Test Data

: Appendix A

Test Data

- Json output
  JO1: [{ lecturer_id: 1, lecturer_username: "lecturerone", lecturer_password: "123456", lecturer_name: "lecturer one", lecturer_faculty: "CAMT", lecturer_department: "SE", lecturer_email: "se01@vr.camt.info", lecturer_tel: "0831112222", lecturer_approvement: 1 }]

Test Case

Case	Description	Input	Expect Result
1	Testing about viewing active lecturers in the system.	-	JO1

UTC-121 Ajax({url : "http://localhost:1580/ManageAccount.svc/ viewStudentAvailable/"})

Package   Admin

Class: ManageAccount

Method: Get

Unit Test Case-121: Ajax({url : "http://localhost:1580/ManageAccount.svc/ viewStudentAvailable/"})

Description: The test case is used for testing about viewing active students in the system.

Prerequisites or Test Data

: Appendix A

Test Data

- Json output
  JO1: [{ student_id: 542115091, student _username: "studentone", student _password: "123456", student _name: "student one", student _faculty: "CAMT" , student _department: "SE", student_address: "Chiang mai, Thailand", student _email: "se542115091@vr.camt.info", student _tel: " 0931112222", student _approvement: 1 }]

Test Case

Case	Description	Input	Expect Result
1	Testing about viewing active students in the system.	-	JO1

UTC-122 Ajax({url : "http://localhost:1580/SemesterManage.svc/viewAllSemester/"})

Package   Admin

Class: SemesterManage

Method: Get

Unit Test Case-122: Ajax({url : "http://localhost:1580/SemesterManage.svc/viewAllSemester/"})

Description: The test case is used for testing about viewing all semester in the system.

Prerequisites or Test Data

: Appendix A

Test Data

- Json output
  JO1: [{semester_id: 1, academic_year: "1/2557"}]

Test Case

Case	Description	Input	Expect Result
1	Testing about viewing all semester in the system.	-	JO1

Document Name	Smart Quiz– Test plan _v2.0.docx	Owner	CS,SK	Page	145 / 209
Document Type	Test Record document	Release Date	16 July 2015	Print Date	

UTC-123 Ajax({url : "http://localhost:1580/SemesterManage.svc/viewAllCourseBySemester/"})

Package	Admin
Class:	SemesterManage
Method:	Post

Unit Test Case-123: Ajax({url : "http://localhost:1580/SemesterManage.svc/ viewAllCourseBySemester/"})

Description: The test case is used for testing about viewing a list of courses in semester.

Prerequisites or Test Data

: Appendix A

Test Data

- Json Input
  JI1: [{semester_id: 1}]
  JI2: [{semester_id: 0}]

- Json output

 JO1: [{course_id: 1, lecturer_id: {lecturer_id:1, lecturer_username: "lecturerone", lecturer_password: "123456", lecturer_name: "lecturer one", lecturer_faculty: "CAMT", lecturer_department: "SE", lecturer_email: "se01@vr.camt.info", lecturer_tel: "0831112222", lecturer_approvement: 1}, semester_id: {semester_id: 1, academic_year: "1/2557"}, course_name: "Math", course_credit: 3, course_description: "Math is easy"}]

 JO2: [{course_id: 0, lecturer_id: { }, semester_id: { }, course_name: "", course_credit: 0, course_description: ""}]

Test Case

Case	Description	Input	Expect Result
1	Testing about viewing all semester in the system.	JI1	JO1
2	Testing about all course in semester which is not existing in the system.	JI2	JO2

UTC-124 Ajax({url : "http://localhost:1580/CourseManage.svc/insertCourse/"})

Package   Admin

Class: CourseManage

Method: Post

Unit Test Case-124: Ajax({url : "http://localhost:1580/ CourseManage.svc/insertCourse /"})

Description: The test case is used for testing about viewing insert a course to the system.

Prerequisites or Test Data

: Appendix A

Test Data

- Json Input
  JI1: { lecturer_id: 1, semester_id: 1, course_name: "C++ language", course_credit: 3, course_description: "C++ is good" }
  JI2: { lecturer_id: 0, semester_id: 1, course_name: "C++ language", course_credit: 3, course_description: "C++ is good" }
  JI3: { lecturer_id: 1, semester_id: 0, course_name: "C++ language", course_credit: 3, course_description: "C++ is good" }
  JI4: { lecturer_id: 1, semester_id: 1, course_name: "", course_credit: 3, course_description: "C++ is good" }
  JI5: { lecturer_id: 1, semester_id: 1, course_name: "C++ language", course_credit: 3, course_description: "" }
  JI6: { lecturer_id: 0, semester_id: 0 course_name: "", course_credit: 3, course_description: "" }

- Json output

    JO1: {" Adding Course Successful"}

    JO2: {"Can not add course"}

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	JI1	JO1
2	Test inputs lecturer id is 0	JI2	JO2
3	Test inputs semester id is 0	JI3	JO2
4	Test inputs course name is empty	JI4	JO2
5	Test inputs course description is empty	JI5	JO2
6	Test inputs lecturer id and semester id are 0, course name and course description are empty	JI6	JO2

UTC-125 Ajax({url : "http://localhost:1580/CourseManage.svc/viewPreviousCourse/"})

Package   Admin

Class: CourseManage

Method: Post

Unit Test Case-125: Ajax({url : "http://localhost:1580/ CourseManage.svc/ viewPreviousCourse /"})

Description: The test case is used for testing about viewing latest course information that was created by lecturer.

Prerequisites or Test Data

: Appendix A

Test Data

- Json output
  JO1: {course_id: 1, lecturer_id: {lecturer_id:1, lecturer_username: "lecturerone", lecturer_password: "123456", lecturer_name: "lecturer one", lecturer_faculty: "CAMT", lecturer_department: "SE", lecturer_email: "se01@vr.camt.info", lecturer_tel: "0831112222", lecturer_approvement: 1}, semester_id: {semester_id: 1, academic_year: "1/2557"}, course_name: "Math", course_credit: 3, course_description: "Math is easy"}

Test Case

Case	Description	Input	Expect Result
1	Testing about viewing latest course information that was created by lecturer	-	JO1

UTC-126 Ajax({url : "http://localhost:1580/CourseManage.svc/ insertStudentsInCourse/"})

Package	Admin
Class:	CourseManage
Method:	Post

Unit Test Case-125: Ajax({url : "http://localhost:1580/ CourseManage.svc/insertStudentsInCourse/"})

Description: The test case is used for testing about inserting students in a course.

Prerequisites or Test Data

: Appendix A

Test Data

- Json Input
  JI1: {course_id: 1, student_id: "542115091"}

- Json output
  JO1: {" Add student in Course is successful"}

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	-	JI1

UTC-126 Ajax({url : "http://localhost:1580/CourseManage.svc/viewCourseByCourseId/"})

Package   Admin

Class: CourseManage

Method: Post

Unit Test Case-126: Ajax({url : "http://localhost:1580/ CourseManage.svc/ viewCourseByCourseId/"})

Description: The test case is used for testing about viewing course information by course id

Prerequisites or Test Data

: Appendix A

Test Data

- Json Input
  JI1: {course_id: 1}
  JI2: {course_id: 0}

- Json output

  JO1: {course_id: 1, lecturer_id: {lecturer_id:1, lecturer_username: "lecturerone", lecturer_password: "123456", lecturer_name: "lecturer one", lecturer_faculty: "CAMT", lecturer_department: "SE", lecturer_email: "se01@vr.camt.info", lecturer_tel: "0831112222", lecturer_approvement: 1}, semester_id: {semester_id: 1, academic_year: "1/2557"}, course_name: "Math", course_credit: 3, course_description: "Math is easy"}

  JO2: {course_id: 0, lecturer_id: { }, semester_id: { }, course_name: "", course_credit: 0, course_description: ""}

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	-	JI1
2	Test input course id which is not existing in database.	-	JI2

UTC-127 Ajax({url : "http://localhost:1580/CourseManage.svc/viewStudentInCourse/"})

Package   Admin

Class: CourseManage

Method: Post

Unit Test Case-127: Ajax({url : "http://localhost:1580/ CourseManage.svc/viewStudentInCourse/"})

Description: The test case is used for testing about viewing students in a course.

Prerequisites or Test Data

: Appendix A

Test Data

- Json Input
  JI1: {course_id: 1}
  JI2: {course_id: 0}

- Json output

  JO1: [{courseRegistration_id: 1,

         course_id : {course_id: 1, lecturer_id: {lecturer_id:1, lecturer_username: "lecturerone", lecturer_password: "123456", lecturer_name: "lecturer one", lecturer_faculty: "CAMT", lecturer_department: "SE", lecturer_email: "se01@vr.camt.info", lecturer_tel: "0831112222", lecturer_approvement: 1}, semester_id: {semester_id: 1, academic_year: "1/2557"}, course_name: "Math", course_credit: 3, course_description: "Math is easy"},

student_id: { student_id: 542115091, student _username: "studentone", student _password: "123456", student _name: "student one", student _faculty: "CAMT" , student _department: "SE", student_address: "Chiang mai, Thailand", student _email: "se542115091@vr.camt.info", student _tel: " 0931112222", student _approvement: 1 }

}]

JO2: [ ]

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	-	JI1
2	Test input course id which is not existing in database.	-	JI2

UTC-128 Ajax({url : "http://localhost:1580/CourseManage.svc/deleteCourse/"})

Package   Admin

Class: CourseManage

Method: Post

Unit Test Case-128: Ajax({url : "http://localhost:1580/ CourseManage.svc/ deleteCourse /"})

Description: The test case is used for testing about deleting a course.

Prerequisites or Test Data

: Appendix A

Test Data

- Json Input
  JI1: {course_id: 1}
  JI2: {course_id: 0}

- Json output

  JO1: {" Delete Course successful"}

  JO2: {" Cannot delete a course"}

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	-	JI1
2	Test input course id which is not existing in database.	-	JI2

UTC-129 Ajax({url : "http://localhost:1580/CourseManage.svc/updateCourse/"})

Package　　　　　Admin

Class:　　　　　　CourseManage

Method:　　　　　Post

Unit Test Case-129: Ajax({url : "http://localhost:1580/ CourseManage.svc/ updateCourse/"})

Description:　　　The test case is used for testing about updating course information in the system.

Prerequisites or Test Data

: Appendix A

Test Data

- Json Input
  JI1: {course_id: 1, lecturer_id: 1, semester_id: 1, course_name: "Math", course_credit: 2, course_description: "Math is easy"}
  JI2: {course_id: 0, lecturer_id: 1, semester_id: 1, course_name: "Math", course_credit: 2, course_description: "Math is easy"}
  JI3: {course_id: 1, lecturer_id: 0, semester_id: 1, course_name: "Math", course_credit: 2, course_description: "Math is easy"}
  JI4: {course_id: 1, lecturer_id: 1, semester_id: 0, course_name: "Math", course_credit: 2, course_description: "Math is easy"}
  JI5: {course_id: 1, lecturer_id: 1, semester_id: 1, course_name: "", course_credit: 2, course_description: "Math is easy"}
  JI6: {course_id: 1, lecturer_id: 1, semester_id: 1, course_name: "Math", course_credit: 2, course_description: ""}
  JI7: {course_id: 0, lecturer_id: 0, semester_id: 0, course_name: "", course_credit: 2, course_description: ""}

- Json output

JO1: {" Delete Course successful"}

JO2: {" Cannot delete a course"}

Test Case

Case	Description	Input	Expect Result
1	Test inputs all parameters is correct data	JI1	JO1
2	Test input course_id is 0	JI2	JO2
3	Test inputs lecturer id is 0	JI3	JO
4	Test inputs semester id is 0	JI4	JO2
5	Test inputs course name is empty	JI5	JO2
6	Test inputs course description is empty	JI6	JO2
7	Test inputs course_id, lecturer id and semester id are 0, course name and course description are empty	JI7	JO2

Test case of System Test (STC)

STC-1: Lecturer registers to the system (URS-03)

Description:

The test is a testing system for URS-03. Lecturer can register to the system by information of lecturer. The test case will check the alphabet, number, and sign in each field.

Prerequisites or Test Input:

The test data is the Lecturer information. The valid input is the input that's correct data for register Lecturer.
For Example
Lec ID = 123456
Lec Username = chaichan
Lec password = 1234
Lec name = chaichan
Lec Faculty = camt
Lec Department = se
Lec Email = Cs@gmail.com
Lec tel = 0882686285

Test Script

1. Lecturer clicks "register lecturer" button
2. Lecturer types the information for register
3. Lecturer clicks "Regist" buttom

Test ID	description	Input								Expect Output
		Lec ID	Lec username	Lec password	Lec Name	Lec Faculty	Lec Department	Lec Email	Lectel	
1	Valid input	123456	chaichan	1234	chaichan	camt	se	Cs@gmail.com	0822686387	"Registration Successful"
2	Invalid input	Abcd	Chaichan	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
3	Invalid input	123456	Cs	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's username must be only 4-16 alphabet characters"
4	Invalid input	123456	Chaichan	123	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's password must be only 4-16 characters
5	Invalid input	123456	Chaichan	1234	palm@c	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's name must be the combination of alphabet and space only"
6	Invalid input	123456	Chaichan	1234	Chaichan	Camt	Se	Cs@gmail.com	TH0822686387	The system shall display "The Lecturer's tel must be number only"
7	Invalid input	123456	Chaichan	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's email must be the character and only one @ symbol
8	Existing input	123456	chaichan	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system "Existing Lecturer's username, please using different Lecturer's username"
9	Null input	Null	chaichan	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
10	Null input	123456	Null	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's username must be only 4-16 alphabet characters"
11	Null input	123456	Chaichan	Null	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's password must be only 4-16 characters

12	Null input	123456	Chaichan	1234	Null	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's name must be the combination of alphabet
15	Null input	123456	Chaichan	1234	Chaichan	Camt	Se	Null	0822686387	The system shall display "The Lecturer's email must be the character and only one @ symbol
16	Null input	123456	chaichan	1234	chaichan	camt	se	Cs@gmail.com	Null	The system shall display "The Lecturer's tel must be number only"
17	Null input	Null	chaichan	1234	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
18	Null input	Null	Null	1234	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
19	Null input	Null	Null	Null	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
20	Null input	Null	Null	Null	Null	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
21	Null input	Null	Null	Null	Null	Null	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
22	Null input	Null	Null	Null	Null	Null	Null	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
23	Null input	Null	Null	Null	Null	Null	Null	Null	0822686387	The system shall display "The Lecturer ID must be the number only"
24	Null input	Null	Null	Null	Null	Null	Null	Null	Null	The system shall display "The Lecturer ID must be the number only"

STC-2: Student registers to the system (URS-01)

Description:

      The test is a testing system for URS-01. Student can register to the system by information of lecturer. The test case will check the alphabet, number, and sign in each field.

Prerequisites or Test Input:

      The test data is the Student information. The valid input is the input that's correct data for register Student.
For Example
Student ID = 123456
Student Username = chaichan
Student password = 1234
Student name = chaichan
Student Faculty = camt
Student Department = se
Student Email = Cs@gmail.com
Student tel = 0882686285

Test Script

1. Student clicks "register student" button
2. Student types the information for register
3. Student clicks "Regist" button

Test ID	Description	Input								Expect Output
		Student ID	Student username	Student password	Student Name	Student Faculty	Student Department	Student Email	Student tel	
1	Valid input	123456	chaichan	1234	chaichan	camt	se	Cs@gmail.com	0822686387	"Registration Successful"
2	Invalid input	abcd	Chaichan	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
3	Invalid input	123456	Cs	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's username must be only 4-16 alphabet characters"
4	Invalid input	123456	Chaichan	123	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's password must be only 4-16 characters
5	Invalid input	123456	Chaichan	1234	palm@c	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's name must be the combination of alphabet and space only"
6	Invalid input	123456	Chaichan	1234	Chaichan	Camt	Se	Cs@gmail.com	TH 0822686387	The system shall display "The Student's tel must be number only"
7	Invalid input	123456	Chaichan	1234	Chaichan	Camt	Se	Csgmail.com	0822686387	The system shall display "The Student's email must be the character and only one @ symbol
8	Existing input	123456	chaichan	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system "Existing Student's username, please using different Student's username"
9	Null input	Null	chaichan	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
10	Null input	123456	Null	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's username must be only 4-16 alphabet characters"
11	Null input	123456	Chaichan	Null	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's password must be only 4-16 characters
12	Null input	123456	Chaichan	1234	Null	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's name must be the combination of alphabet

15	Null input	123456	Chaichan	1234	Chaichan	Camt	Se	Null	0822686387	The system shall display "The Student's email must be the character and only one @ symbol
16	Null input	123456	chaichan	1234	chaichan	camt	se	Cs@gmail.com	Null	The system shall display "The Student's tel must be number only"
17	Null input	Null	chaichan	1234	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
18	Null input	Null	Null	1234	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
19	Null input	Null	Null	Null	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
20	Null input	Null	Null	Null	Null	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
21	Null input	Null	Null	Null	Null	Null	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
22	Null input	Null	Null	Null	Null	Null	Null	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
23	Null input	Null	Null	Null	Null	Null	Null	Null	0822686387	The system shall display "The Lecturer ID must be the number only"
24	Null input	Null	Null	Null	Null	Null	Null	Null	Null	The system shall display "The Lecturer ID must be the number only"

STC-3: Administrator can approve Lecturer or Student account. (URS-05)

Description:

The test is a testing system for URS-05. Administrator can approve Lecturer or student.

Prerequisites or Test Input:

The test data is lecturer or student account that is waiting for approval.

Test Script

1. Administrator types the username and pass word for login and then click login
2. Administrator clicks "manage account"
3. Administrator select approve lecturer or approve student
4. Administrator clicks check box for approve
5. Administrator clicks "approve"

Test case ID	Input type	Description	Expect Output
1	Valid Input	Admin click check box for approve	"approve User ID Successful"
2	Valid Input	Admin didn't click check box	Go to next person

STC-4: Lecturer can edit personal information (URS-04)

Description:

The test is a testing system for URS-04. Lecturer can edit personal information. The test case will check the alphabet, number, and sign in each field.

Prerequisites or Test Input: The test data is the Lecturer information that will be edited in the database

Test Script

1. Lecturer types username and password for login and then click login
2. Lecturer clicks Personal Information
3. Lecturer Edits Information
4. Lecturer clicks "Edit" button

Test ID	Description	Input									Expect Output
		Lec ID	Lec username	Lec password	Confirm password	Lec Name	Lec Faculty	Lec Department	Lec Email	Lectel	
1	Valid input	123456	chaichan	1234	1234	chaichan	camt	se	Cs@gmail.com	0822686387	"editing Successful"
2	invalid input	abcd	Chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
3	invalid input	123456	Cs	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's username must be only 4-16 alphabet characters"
4	invalid input	123456	Chaichan	123	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's password must be only 4-16 characters
5	invalid input	123456	Chaichan	1234	1234	palm@c	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's name must be the combination of alphabet and space only"
6	invalid input	123456	Chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	TH 0822686387	The system shall display "The Lecturer's tel must be number only"
7	invalid input	123456	Chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's email must be the character and only one @ symbol
8	Existing input	123456	chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system "Existing Lecturer's username, please using different Lecturer's username"

#		ID	Username	Password	Confirm	Name			Email	Tel	Expected result
9	Null input	Null	chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
10	Null input	123456	Null	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's username must be only 4-16 alphabet characters"
11	Null input	123456	Chaichan	Null	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's password must be only 4-16 characters
12	Null input	123456	Chaichan	1234	1234	Null	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Lecturer's name must be the combination of alphabet
15	Null input	123456	Chaichan	1234	1234	Chaichan	Camt	Se	Null	0822686387	The system shall display "The Lecturer's email must be the character and only one @ symbol
16	Null input	123456	chaichan	1234	1234	chaichan	camt	se	Cs@gmail.com	Null	The system shall display "The Lecturer's tel must be number only"
17	Null input	123456	chaichan	1234	1235	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "Password and confirm password is not matched"
18	Null input	Null	chaichan	1234	1234	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
19	Null input	Null	Null	1234	1235	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
20	Null input	Null	Null	Null	1234	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
21	Null input	Null	Null	Null	Null	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
22	Null input	Null	Null	Null	Null	Null	camt	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
23	Null input	Null	Null	Null	Null	Null	Null	se	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"
24	Null input	Null	Null	Null	Null	Null	Null	Null	Cs@gmail.com	0822686387	The system shall display "The Lecturer ID must be the number only"

25	Null input	Null	Null	Null	Null	Null	Null	Null	Null	0822686387	The system shall display "The Lecturer ID must be the number only"
26	Null input	Null	Null	Null	Null	Null	Null	Null	Null	Null	The system shall display "The Lecturer ID must be the number only"

STC-5: Student can edit personal information (URS-02)

Description:

The test is a testing system for URS-02. Student can edit personal information. The test case will check the alphabet, number, and sign in each field.

Prerequisites or Test Input:

The test data is the Student information that will be edited in the database

Test Script

1. Student types username and password for login and then click login
2. Student clicks Personal Information
3. Student Edits Information
4. Student clicks "Edit" button

Test ID	Description	Input									Expect Output
		Student ID	Student username	Student password	Confirm password	Student Name	Student Faculty	Student Department	Student Email	Student tel	
1	Valid input	123456	chaichan	1234	1234	chaichan	camt	se	Cs@gmail.com	0822686387	"Editing Successful"
2	Invalid input	abcd	Chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
3	Invalid input	123456	Cs	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's username must be only 4-16 alphabet characters"
4	Invalid input	123456	Chaichan	123	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's password must be only 4-16 characters
5	Invalid input	123456	Chaichan	1234	1234	palm@c	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's name must be the combination of alphabet and space only"
6	Invalid input	123456	Chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	TH0822686 387	The system shall display "The Student's tel must be number only"
7	Invalid input	123456	Chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's email must be the character and only one @ symbol
8	Existing input	123456	chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system "Existing Student's username, please using different Student's username"
9	Null input	Null	chaichan	1234	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
10	Null	123456	Null	1234	1234	Chaichan	Camt	Se	Cs@gmail.	0822686387	The system shall display "The Student's username

	input								com		must be only 4-16 alphabet characters"
11	Null input	123456	Chaichan	Null	1234	Chaichan	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's password must be only 4-16 characters
12	Null input	123456	Chaichan	1234	1234	Null	Camt	Se	Cs@gmail.com	0822686387	The system shall display "The Student's name must be the combination of alphabet
15	Null input	123456	Chaichan	1234	1234	Chaichan	Camt	Se	Null	0822686387	The system shall display "The Student's email must be the character and only one @ symbol
16	Null input	123456	chaichan	1234	1234	chaichan	camt	se	Cs@gmail.com	Null	The system shall display "The Student's tel must be number only"
17	Null input	123456	chaichan	1234	1235	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "Password and confirm password is not matched"
18	Null input	Null	chaichan	1234	1234	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
19	Null input	Null	Null	1234	1235	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
20	Null input	Null	Null	Null	1234	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
21	Null input	Null	Null	Null	Null	chaichan	camt	se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
22	Null input	Null	Null	Null	Null	Null	camt	se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
23	Null input	Null	Null	Null	Null	Null	Null	se	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
24	Null input	Null	Null	Null	Null	Null	Null	Null	Cs@gmail.com	0822686387	The system shall display "The Student ID must be the number only"
25	Null input	Null	Null	Null	Null	Null	Null	Null	Null	0822686387	The system shall display "The Student ID must be the number only"

Document Name	Smart Quiz– Test plan _v2.0.docx	Owner	CS,SK	Page	174 / 209
Document Type	Test Record document	Release Date	16 July 2015	Print Date	

26	Null input	Null	Null	Null	Null	Null	Null	Null	Null	Null	The system shall display "The Student ID must be the number only"

STC-6: user can login (URS-06, 08, 10)

Description:

The test is a testing system for URS-06, 08, 10. Users can login via username and password

Prerequisites or Test Input:

Username and password of users. For test case id 1 is Pass, because the data is match with the database. For test case id 2 is Fail Because The data is not match with the database.

Test Script

1. User click type Username and password and then click login

Test case id	Description	Input		Expect Output
		Username	password	
1	Valid Input	Chaichan	1234	Go to lecturer main page
2	Invalid Input	Aaasc	54321	System show error message "Invalid Username and/or Password"
3	Null Input	Null	54321	System show error message "Invalid Username and/or Password"
4	Null Input	Chaichan	Null	System show error message "Invalid Username and/or Password"
5	Null Input	Null	Null	System show error message "Invalid Username and/or Password"

STC-7: user can logout (URS-07, 09, 11)

Description:

The test is a testing system for URS-07, 09, 11. Users can logout via username and password

Prerequisites or Test Input:

Student, Lecturer, Administrator login status.

Test script

1.User clicks Logout button

Test case id	Input type	Input Description	Expect Output
1	Valid Input	User click log out	Go to Login page

STC-8: Administrator can create semester (URS-14)

Description:

The test is a testing system for URS-14. Administrator can create semester. The test case will check the alphabet, number, and sign in each field.

Prerequisites or Test Input: Semester information that will be created semester. The valid input is For The description is number in S/YYYY format only.

Test Script

1. Administrator login to system

2. Administrator clicks Semester management

3. Administrator type semester information

4. Administrator clicks "Add" button

Test case	Description	Input		Expect Output
		Name	description	
1	Valid Input	Year 4	1/2557	"Addition Semester Successful" message.
2	In Valid Input	Year 4	one	"Semester academic year must be S/YYYY form. Which S is a semester and YYYY is an academic year"
4	Null Input	Year 4	Null	"Semester academic year must be S/YYYY form. Which S is a semester and YYYY is an academic year"
5	Null Input	Null	1/2557	"Semester academic year must be S/YYYY form. Which S is a semester and YYYY is an academic year"

| 6 | Null Input | Null | Null | "Semester academic year must be S/YYYY form. Which S is a semester and YYYY is an academic year" |

STC-9: Administrator can edit semester (URS-15)

Description:

The test is a testing system for URS-15. Administrator can edit semester. The test case will check the alphabet, number, and sign in each field.

Prerequisites or Test Input: Semester information

Test Script

1. Administrator login to system

2. Administrator clicks Semester management

3. Administrator Selects semester for edit

4.Administrator clicks "Edit" button

5. Administrator type semester information

6. Administrator click "Edit" button

Test case	Description	Input		Expect Output
		Name	description	
1	Valid Input	Year 4	1/2557	"editting Semester Successful" message.
2	Invalid Input	Year4	one	"Semester academic year must be S/YYYY form. Which S is a semester and YYYY is an academic year"

4	Null Input	Year 4	Null	"Semester academic year must be S/YYYY form. Which S is a semester and YYYY is an academic year"
5	Null Input	Null	1/2557	"Semester academic year must be S/YYYY form. Which S is a semester and YYYY is an academic year"
6	Null Input	Null	Null	"Semester academic year must be S/YYYY form. Which S is a semester and YYYY is an academic year"

STC-10: Administrator can delete semester (URS-16)

Description:

The test is a testing system for URS-16. Administrator can delete semester

Prerequisites or Test Input: Semester information

Test Script

1. Administrator login to system

2. Administrator clicks Semester management

3.Administrator Selects semester for Delete.

4.Administrator clicks "Delete" button

Test case	Input Type	Input description	Expect output
1	Valid Input	Click "delete semester"	The system show "deleting Semester Successful" message.

STC-11: Administrator can create course (URS-17)

Description:

The test is a testing system for URS-17. Administrator can create course. The test case will check the alphabet, number, and sign in each field.

Prerequisites or Test Input: Course information that will be create the course. The valid input is the course credit must be number only.

Test Script

1. Administrator login to system
2. Administrator click course management page.

3. Administrator select semester and then click view button.

4. Administrator Clicks "Add Course" button

5. Administrator type the course information.

6.Administrator click "Next" button

7. Administrator select student and then click "Add course" button

Test case	Description	Input				Expect Output
		Course name	Course credit	Course description	Lecturer Name	
1	Valid Input	Se101	1	Basic	Chaichan	The system display message "Adding Course Successful" message.
2	Invalid Input	Se101	one	Basic	Chaichan	The system display error message "The course credit must be number only"

3	Null Input	Se101	Null	Basic	Chaichan	The system display message "The course credit must be number only"
4	Null Input	Null	1	Basic	Chaichan	The system display error message "Insert the course name"
5	Null Input	Null	Null	Basic	Chaichan	The system display error message "Insert the course name"
6	Null Input	Null	Null	Null	Chaichan	The system display error message "Insert the course name"
7	Null Input	Null	Null	Null	Null	The system display error message "Insert the course name"

STC-12: Administrator can edit course (URS-18)

Description:

The test is a testing system for URS-18. Administrator can edit course. The test case will check the alphabet, number, and sign in each field.

Prerequisites or Test Input : Course information

Test Script

1. Administrator login to system
2. Administrator click course management page.
3. Administrator select semester and then click view button.
4. Administrator Clicks "view" for select course.
5. Administrator type information and then click edit course.

Test case	Description	Input				Expect Output
		Course name	Course credit	Course description	Lecturer Name	
1	Valid input	Se101	1	Basic	chaichan	"Editting Course Successful" message.
2	Invalid input	Se101	one	Basic	Chaichan	The system display error message "The course credit must be number only"
3	Null Input	Se101	Null	Basic	Chaichan	The system display error message "The course credit must be number only"
4	Null Input	Null	1	Basic	Chaichan	The system display error message "Insert the course name"
5	Null Input	Null	Null	Basic	Chaichan	The system display error message "Insert the course name"

| 6 | Null Input | Null | Null | Null | Chaichan | The system display error message "Insert the course name" |
| 7 | Null Input | Null | Null | Null | Null | The system display error message "Insert the course name" |

STC-13: Administrator can delete course (URS-19)

Description:

The test is a testing system for URS-19. Administrator can edit course. The test case will pass If input the valid input.

Prerequisites or Test Input: Course information

Test Script

1. Administrator login to system
2. Administrator click course management page.
3. Administrator select semester and then click view button.
4. Administrator Clicks "view" for select course.
5. Administrator type information and then click delete course.

Test case	Input Type	Input Description	Expect Output
1	Valid Input	Click Delete	"Deleting Course Successful" message.

STC-14: Lecturer can view course (URS-13)

Description:

The test is a testing system for URS-13. Lecturer can view course

Prerequisites or Test Input : Course information

Test Script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer selects course and then click view

Test case	Input Type	Input description	Expect Output
1	Valid Input	Click view course	The system show course information

STC 15: Student can view course (URS-12)

Description:

The test is a testing system for URS-12. Student can view course

Prerequisites or Test Input : Course information

Test Script

1. Student login to system
2. Student select course
3. Student select semester and then click view
4. Student selects course and then click view

Test case	Input Type	Input description	Expect Output
1	Valid Input	Click view course	The system show course information

STC-16: Lecturer can create quiz (URS-21)

Description:

The test is a testing system for URS-21. Lecturer can a create quiz. The test case will check the alphabet, number, and sign in each field. If the data is valid the test case will pass.

Prerequisites or Test Input: Quiz Information the valid input is

Test script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer selects course and then click Over all test
5. Lecturer click "add testing" button
6. Lecturer click "add question" button
7. Lecturer select type and click "add type button"
8. Lecturer type information and then click "add question"
9. Lecturer click "add testing" button

Test ID	Description	Input											Expect Output
		Test name	type of testing	amount of question	student name	the question status	Posting status	type of question	question description	choice description	Anwser	Question point	
1	Valid input	Lap test	Assignment	3	542115016	Random	Available	Multiple choice	1+1=?	A:1,B:2	B	1	The system displays "Creating Quiz Successful"
2	Invalid input	Lap test	Assignment	3	542115016	Random	Available	True/false question	1+1=2	A:True, B:False	A	1	The system displays "Creating Quiz Successful"
3	Invalid input	Lap test	Assignment	3	542115016	Random	Available	Short Anwser	1+1=?	null	null	1	The system displays "Creating Quiz Successful"
4	Invalid input	Lap test	Assignment	three	542115016	Random	Available	Short Anwser	1+1=?	null	null	1	The system displays "Question point must be number only"
5	Invalid input	Lap test	Assignment	three	542115016	Random	Available	Short Anwser	1+1=?	null	null	one	The system displays "Testing score must be number only"
6	Invalid input	Lap test	Assignment	3	542115016	Random	Available	Multiple choice	1+1=?	A:1,B:2	2	1	The system displays "Answer must be a, b, c, and d only
7	Invalid input	Lap test	Assignment	3	542115016	Random	Available	True/false question	1+1=2	A:True, B:False	C	1	The system displays "Answer must be a and b only"
8	Null Input	Lap test	Assignment	3	542115016	Random	Available	True/false question	1+1=2	A:True, B:False	C	1	The system display "type the test name"
9	Null Input	Null	Assignment	3	542115016	Random	Available	True/false question	1+1=2	A:True, B:False	C	1	The system display "type the test name"

10	Null Input	Null	Null	3	542115016	Random	Available	True/false question	1+1=2	A:True, B:False	C	1	The system display "type the test name"
11	Null Input	Null	Null	Null	542115016	Random	Available	True/false question	1+1=2	A:True, B:False	C	1	The system display "type the test name"
12	Null Input	Null	Null	Null	Null	Random	Available	True/false question	1+1=2	A:True, B:False	C	1	The system display "type the test name"
13	Null Input	Null	Null	Null	Null	Null	Available	True/false question	1+1=2	A:True, B:False	C	1	The system display "type the test name"
14	Null Input	Null	Null	Null	Null	Null	Null	True/false question	1+1=2	A:True, B:False	C	1	The system display "type the test name"
15	Null Input	Null	Null	Null	Null	Null	Null	Null	1+1=2	A:True, B:False	C	1	The system display "type the test name"
16	Null Input	Null	Null	Null	Null	Null	Null	Null	Null	A:True, B:False	C	1	The system display "type the test name"
17	Null Input	Null	Null	Null	Null	Null	Null	Null	Null	Null	C	1	The system display "type the test name"
18	Null Input	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null	1	The system display "type the test name"
19	Null Input	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null	The system display "type the test name"

STC-17: Lecturer can edit Test (URS-22)

Description:

The test is a testing system for URS-22. Lecturer can edit quiz and assignment. The test case will check the alphabet, number, and sign in each field. If the data is valid the test case will pass.

Prerequisites or Test Input: test information

Test script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer selects course and then click Over all test
5. Lecturer select quiz and then click "view" button
6. Lecturer type information and then click "Edit testing"

Test ID	Input Type	Input Description	Expect Output
1	Valid Input	Type the information	"Editing testing successful "

STC-18: Lecturer can mark point. (URS-25)

Description:

The test is a testing system for URS-25. Lecturer can mark point. The test case will check the number in each field. If the data is valid the test case will pass.

Prerequisites or Test Input: test information

Test script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer selects course and then click Over all test
5. Lecturer select quiz and then click "view"button
6. Lecturer select quiz and then click" MARK

Test case	Input type	Input description	Expect Output
1	Valid Input	1	The system displays "Marking point is successful"
2	Invalid Input	one	The system show error message is "Point of student must be number only"
2	Invalid Input	Null	The system show error message is "Point of student must be number only"

STC-19: Lecturer can review overall student's answer in each choice of assignment or quiz in chart format (URS- 26)

Description:

The test is a testing system for URS-26. Lecturer can view student answer in chart format.

Prerequisites or Test Input: Student answer

Test script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer selects course and then click Over all test
5. Lecturer select quiz and then click "Chart" button

Test case	Input type	Input description	Expect Output
1	Valid Input	Click view student answer	System display Student Answer in chart format

STC-20: Lecturer can check time of submitted quiz. (URS-27)

Description:

The test is a testing system for URS-27. Lecturer can check time of submitted quiz. The system will display the time of submitted

Prerequisites or Test Input: Quiz of student

Test script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer selects course and then click Over all test
5. Lecturer select quiz and then click "SUBMISSION" button

Test case	Input type	Input description	Expect Output
1	Valid Input	Click check time of submitted quiz	System display time of submitted quiz

STC-21: Lecturer can select course for create assignment and quiz. (URS-20)

Description:

The test is a testing system for URS-20. Lecturer can select course for create assignment and quiz.

Prerequisites or Test Input: Quiz or assignment

Test Script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer select course and then click "overall Test"

Test case	Input type	Input description	Expect Output
1	Valid Input	Click select course	System display course information

STC-22: Student can select course for take assignment and quiz. (URS-23)

Description:

The test is a testing system for URS-23. Student can select course for take assignment and quiz.

Prerequisites or Test Input: Quiz or assignment

Test Script

1. Student login to system
2. Student select course
3. Student select semester and then click view
4. Student select course and then click "overall Test"

Test case	Input type	Input description	Expect Output
1	Valid Input	Click select course	System display quiz name in course

STC-23: Student can take quiz. (URS-24)

Description:

The test is a testing system for URS-24. Student can take quiz. The test case will check the alphabet, number, and sign in each field.

Prerequisites or Test Input: Quiz or assignment

Test Script

1. Student login to system
2. Student select course
3. Student select semester and then click view
4. Student select course and then click "overall Test"
5. Student select course and then click " Take testing"

Test ID	Description	Input		Expect Output
		type of question	Answer	
1	Valid Input	Multiple choice	B	The system redirect to next question
2	Valid Input	True/false question	A	The system redirect to next question
3	Valid Input	Short Anwser	two	The system redirect to next question
6	Invalid Input	Multiple choice	2	The system displays "Answer must be a, b, c, and d only"
7	Invalid Input	True/false question	C	The system displays "Answer must be a and b only"
8	Null Input	True/false question	Null	The system displays "Answer must be a and b only"
9	Null Input	Multiple choice	Null	The system displays "Answer must be a, b, c, and d only"

STC-24: Lecturer can manage score by using Google's spreadsheets (URS-28)

Description:

The test is a testing system for URS-28 Lecturer can manage score by using Google's spreadsheets

Prerequisites or Test Input: Score of student

Test script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer click spreadsheet management

Test case	Input type	Input description	Expect Output
1	Valid Input	Lecturer select Google's spreadsheets	System display google's spreadsheets

STC-25: Lecturer can post the score for student (URS-29)

Description:

The test is a testing system for URS-29. Lecturer can post the score for student by export file form spreadsheets.

Prerequisites or Test Input: Score of student

Test script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer click spreadsheet management

Test case	Input type	Input description	Expect Output
1	Valid input	Lecturer click sharing	System share file with student

STC-26: Student can view score (URS-33)

Description:

The test is a testing system for URS-33. Student can view score by select Google's spreadsheet.

Prerequisites or Test Input: Score of student

Test script

1. Student login to system
2. Student clicks course management system
3. Student clicks spreadsheets management
4. Student selects Google's spreadsheets

Test case	Input type	Input description	Expect Output
1	Valid Input	Student select Google's spreadsheets	System display Google's spreadsheets.

STC-27: Lecturer can create new Google's spreadsheet spreadsheets (URS-30)

Description:

The test is a testing system for URS-30 Lecturer can create new Google's spreadsheets.

Prerequisites or Test Input: Score of student

Test script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer click spreadsheet management
5. Lecturer click new file in Google spreadsheet.

Test case	Input type	Input description	Expect Output
1	Valid Input	Lecturer click new spreadsheets.	System create new Google's spreadsheets

STC-28: Lecturer can delete Google's spreadsheet. (URS-31)

Description:

The test is a testing system for URS-31 Lecturer can delete Google's spreadsheets.

Prerequisites or Test Input: Score of student

Test script

1. Lecturer login to system
2. Lecturer select course
3. Lecturer select semester and then click view
4. Lecturer click spreadsheet management
5. Lecturer click "delete" button.

Test case	Input type	Input description	Expect Output
1	Valid Input	Lecturer click delete spreadsheets.	System delete Google's spreadsheets form database.

STC-29: Lecturer can post the question for Student. (URS-37)

Description:

The test is a testing system for URS-37 Lecturer can post the question for Student. This test case will check the information in each field.

Prerequisites or Test Input: Information of reply

Test script

1. Lecturer selects course for post/reply.
2. Lecturer clicks post page.
3. Lecturer clicks new topic.
4. Lecturer types the information for new topic.
5. Lecturer clicks post button.

Test case	Input Type	Input description	Expect Output
1	Valid input	Lecturer click post button.	System display the post on the post page
2	Valid input	Lecturer click select person to ask	System display the post on the post page
3	Invalid input	Lecturer didn't select person to ask	System display error message "please select person to ask"

STC-30: Lecturer can reply the question for Student. (URS-38)

Description:

The test is a testing system for URS-38 Lecturer can reply the question for Student. This test case will check the information in each field.

Prerequisites or Test Input: information of reply

Test script

1. Lecturer selects course for post/reply.
2. Lecturer clicks post page.
3. Lecturer clicks new topic.
4. Lecturer types the information for new topic.
5. Lecturer clicks post button.

Test case	Input Type	Input description	Expect Output
1	Valid input	Lecturer click reply button.	System display the reply post on the post page
2	Invalid input	Lecturer didn't type information	System display error message "please type information"

STC-31: Student can post the question. (URS-35)

Description:

The test is a testing system for URS-35 Student can post/reply the question.

Prerequisites or Test Input: Information of post

Test script

1. Student selects course for reply.
2. Student clicks post page.
3. Student clicks new topic.
4. Student types the information for new topic.
5. Student clicks post button.

Test case	Input Type	Input description	Expect Output
1	Valid input	Student click post button.	System display the post on the post page
2	Valid input	Student select person to ask	System display the post on the post page
3	Invalid input	Student didn't select person to ask	System display error message" please select person to ask"
4	Invalid input	Student didn't type information	System display error message "please type information"

STC-32: Student can reply the question. (URS-36)

Description:

The test is a testing system for URS-36 Student can reply the question.

Prerequisites or Test Input: Information of reply

Test script

1. Student select course for post/reply.
2. Student clicks post page.
3. Student clicks reply.
4. Student types the information for reply.
5. Student clicks reply button.

Test case	Input Type	Input description	Expect Output
1	Valid input	Student click reply button.	System display the reply post on the post page
2	Invalid input	Student didn't type information	System display error message "please type information"

STC-33: Lecturer click recommend. (URS-39)

Description:

The test is a testing system for URS-39 Lecturer can click recommend.

Prerequisites or Test Input: reply of the post

Test script

1. Lecturer selects course
2. Lecturer clicks post page.
3. Lecturer clicks "recommend" button.

Test case	Input Type	Input description	Expect Output
1	Valid input	Lecturer click recommend button.	System display the recommend post on the post page

STC-34: Lecturer click delete post. (URS-40)

Description:

The test is a testing system for URS-40 Lecturer can click delete the post.

Prerequisites or Test Input: reply of the post

Test script

1. Lecturer selects course
2. Lecturer clicks post page.
3. Lecturer clicks "delete" button.

Test case	Input Type	Input description	Expect Output
1	Valid input	Lecturer click delete button.	System remove post form database

STC-35: Lecturer can upload file to Google's spreadsheets. (URS-32)

Description:

The test is a testing system for URS-32 Lecturer can upload file to Google's spreadsheets.

Prerequisites or Test Input: file for upload

Test script

1. Lecturer clicks course management system
2. Lecturer click spreadsheets management
3. Lecturer selects Google's spreadsheets
4. Lecturer clicks choose file
5. Lecturer clicks upload button

Test case	Input Type	Input description	Expect Output
1	Valid input	Lecturer click upload button.	System upload file to Google's spreadsheets
2	Invalid input	Lecturer didn't select file	System display error message "please select file"

STC-36: Lecturer can send E-mail to student. (UC-34)

Description:

The test is a testing system for URS-34 Lecturer can send E-mail to student.

Prerequisites or Test Input: E-mail and information

Test script

1. Lecturer clicks "E-mail" button.
2. Lecturer types the information and E-mail
3. Lecturer clicks send

Test case	Input Type	Input description	Expect Output
1	Valid input	Lecturer click send	System send E-mail to student
2	Invalid input	Lecturer didn't type information	System display error message "please type information"