**Smart course Management**

**System in cloud**

Software Design Document

By

**Mr. Chaichan Suttee 542115016**

**Mr. Tanadol Parn-ong 542115021**

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

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

**Mrs. Yun Rim Park**

# **Chapter One | Introduction**

## **Purpose**

The software design document (SDD) is written for describing the high level of software design, system architecture, database system and graphic user interfaces. It also consists of the users’ requirements. This document is an important thing for facilitating the developer team members to understand the direction for developing Smart course management system in the cloud.

## **1.2 Software Scope**

This document describes the implementation details of Smart course management system in the cloud. The system consists of five features. First feature is ‘*User registration*’ which covers the registration of Lecturer and Student. Second feature is ‘*Log in/Log out system*’ and covers the log in/log out of Lecturer account, Student account, and Administrator account. Third feature is ‘*Course management*’. It covers management of courses by Administrator, and access to those courses by Lecturer and Student. Fourth feature is ‘*Assignment and quiz management*’ that covers assigning, editing, deleting assignments or quizzes by Lecturer. Fifth feature is ‘*Taking assignments and quizzes*’. It covers taking assignments and quizzes by Student. This document explains how all features in the application are implemented using class diagrams and sequence diagrams for clarifying and providing the overall picture of the system.

## **1.3 Technology Review**

1.3.1 JavaScript

Technology description

JavaScript is a programming language known as the “script”. Translation operates on each command. It is commonly used as part of web browsers, which allow client-side scripts to interact with the user, control the browser, and change the content displayed on the We can add the JavaScript programming web pages to process data of both the display and receiving - transmitting. And that can interact with users immediately. JavaScript is also used in server-side programming and building games as well as desktop and mobile applications.

The selection of this technology

* We use JavaScript for calculating data, displaying, sending – receiving data for interacting with the user.
* JavaScript is easy to learn.

1.3.2 HTML5

Technology description

HTML5 is a language that was developed for use as a markup language. Website for writing the latest version is being developed by WHATWG (The Web Hypertext Application Technology Working Group). Several features have been added to allow developers to use more easily.

Alternative technology

- HTML4

The selection of this technology

- HTML5 can access sites easier than older version.

- HTML5 is support multimedia content.

- HTML5 is standardized for creating the user interface of website.

1.3.3 JavaScript Object Notation (JSON)

Technology description

JSON (JavaScript Object Notation) is a data interchange - format. It is based on a JavaScript programming language. JSON is a text format that is completely language independent. JSON uses sending and receiving data to and from different platforms.

Alternative technology

- XML

The selection of this technology

- JSON can send and receive data between different platforms.

- JSON is based on JavaScript, which the developers are familiar with.

# **Chapter Two | System Architecture**

Web service

DB

JSON

Mobile application

Web application

IIS

IIS



**iOS 7**

**Firefox web browser**

***Figure 8 Smart course management system in the cloud Architecture.***

The architecture of Smart course Management system in the cloud is shown in Figure 8. It consists of the web application, which the user can access via a Firefox browser. It also consists of the mobile application that is available on IOS7. The system uses JSON for interchanging data between the web services and the web application. Moreover, the system uses JSON for interchanging data between the web services and the mobile application. The web services have been implemented in C#.Net and MS SQL database. In this progress, the system functionalities are tested for Firefox web browser only.

# **Chapter Three | Detail Design**

## **3.1 Class Diagram Description**

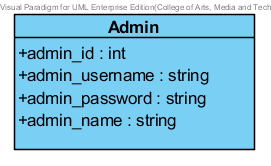


***Figure 2 Smart course management system in the cloud Class diagram.***

**Method Design**

**3.1.1 Class-01. Admin**

Class Admin is Model-1 (MD-1)



**Description**

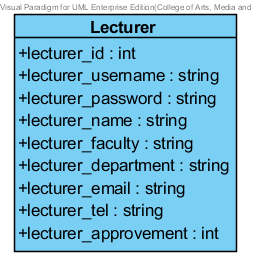
This class is a model of for administrator which consists administrator information and username/password.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 1.1 | admin \_id | Unique Identifier of Administrator | int |
| 1.2 | admin \_username | Username of Administrator | string |
| 1.3 | admin \_password | Password of Administrator | string |
| 1.4 | admin \_name | Name of Administrator | string |

**3.1.2 Class-02. Lecturer**

Class Lecturer is Model-2 (MD-2)



**Description**

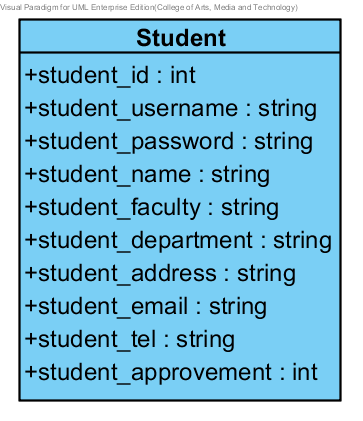
This class is a model class for the lecturer which consists lecturer information and username/password.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 2.1 | lecturer\_id | Unique Identifier of Lecturer | int |
| 2.2 | lecturer \_username | Username of Lecturer | string |
| 2.3 | lecturer \_password | Password of Lecturer | string |
| 2.4 | lecturer \_name | Name of Lecturer | string |
| 2.5 | lecturer \_faculty | Faculty of Lecturer | string |
| 2.6 | lecturer \_department | Department of Lecturer | string |
| 2.7 | lecturer \_address | Address of Lecturer | string |
| 2.8 | lecturer \_email | Email of Lecturer | string |
| 2.9 | lecturer \_tel | Tel of Lecturer | string |
| 2.10 | lecturer \_approvement | Approval status of Lecturer (1 is approve, 0 is not approve) | int |

**3.1.3 Class-03. Student**

Class Student is Model-3 (MD-3)



**Description**

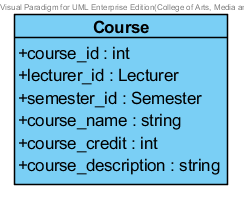
This class is a model class for the student which consists student information and username/password.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 3.1 | student\_id | Unique Identifier of Student | int |
| 3.2 | student\_username | Username of Student | string |
| 3.3 | student\_password | Password of Student | string |
| 3.4 | student\_name | Name of Student | string |
| 3.5 | student\_faculty | Faculty of Student | string |
| 3.6 | student\_department | Department of Student | string |
| 3.7 | student\_address | Address of Student | string |
| 3.8 | student\_email | Email of Student | string |
| 3.9 | student\_tel | Tel of Student | string |
| 3.10 | student\_approvement | Approval status of Student (1 is approve, 0 is not approve) | int |

**3.1.4 Class-04. Course**

Class Course is Model-4 (MD-4)



**Description**

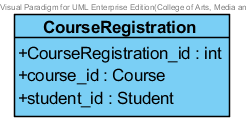
This class is a model class for the course which consists course information.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 4.1 | course\_id | Unique Identifier of Course | int |
| 4.2 | lecturer\_id | Object for identifying Lecturer | Lecturer |
| 4.3 | semester\_id | Object for identifying Semester | Semester |
| 4.4 | course\_name | Name of Course | string |
| 4.5 | course\_credit | Credit of Course | int |
| 4.6 | course\_description | Description of Course | string |

**3.1.5 Class-05. CourseRegistration**

Class CourseRegistration is Model-5 (MD-5)

****

**Description**

This class is a model class for registration of student in each course which consists student id and course id for identifying registration.

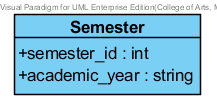
.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 5.1 | CourseRegistration\_id | Unique Identifier of Registration of Student into Course | int |
| 5.2 | course\_id | Object for identifying Course | Course |
| 5.3 | student\_id | Object for identifying Student | Student |

**3.1.6 Class-06. Semester**

Class Semester is Model-6 (MD-6)



**Description**

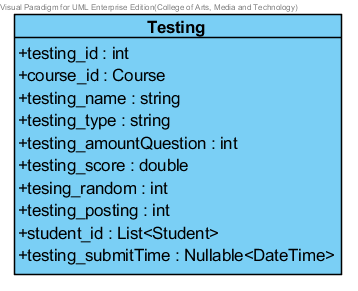
This class is a model class for semester model which consists academic year information.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 6.1 | semester\_id | Unique Identifier of Semester | int |
| 6.2 | academic\_year | Value of semester | string |

**3.1.7 Class-07. Testing**

Class Testing is Model-7 (MD-7)



**Description**

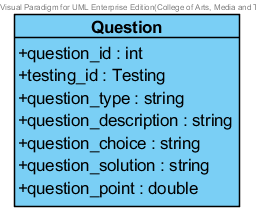
This class is a model class for testing model which consists testing information and list of student that can access to the testing.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 7.1 | testing\_id | Unique Identifier of Testing | int |
| 7.2 | course\_id | Object for identifying Course | Course |
| 7.3 | testing\_name | Name of Testing | string |
| 7.4 | testing\_type | Type of Testing (1 is an assignment, 2 is a quiz) | string |
| 7.5 | testing\_amountQuestion | Amount of question in testing | int |
| 7.6 | testing\_score | Score of Testing | double |
| 7.7 | tesing\_random | Random status of testing (1 is random, 2 is not random) | int |
| 7.8 | testing\_posting | Post status of testing (1 is post, 0 is not post) | int |
| 7.9 | student\_id | List of Student objects in testing. It use for identifying accession to testing of a student. | List<Student> |
| 7.10 | testing\_submitTime | Deal line of time to submission | Nullable<Date Time> |

**3.1.8 Class-08. Question**

Class Question is Model-8 (MD-8)



**Description**

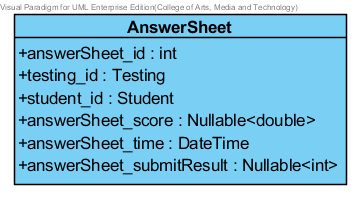
This class is a model class for question model which consists question information and testing id for identifying where question in testing.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 8.1 | question\_id | Unique Identifier of Question | int |
| 8.2 | testing\_id | Object for identifying Testing | Testing |
| 8.3 | question\_type | Type of question (1 is a multiple choices question, 2 is a true/false question, 3 is a short answer question) | string |
| 8.4 | question\_description | Description of question | string |
| 8.5 | question\_choice | Description of choice in question  - Format of data  1. Multiple choice  Ex. choice1#$choice2#$choice3#$choice4  2. True/False  Ex. True#$False  3. Short answer  Ex. null | string |
| 8.6 | question\_solution | Solution of question  - Format data  1. Multiple choice  Ex. index of choice (1 or 2 or 3 or 4)  2. True/False  Ex. index of choice (1 or 2)  3. Short answer  Ex. null | string |
| 8.7 | question\_point | Score of question | double |

**3.1.9 Class-09. AnswerSheet**

Class AnswerSheet is Model-9 (MD-9)



**Description**

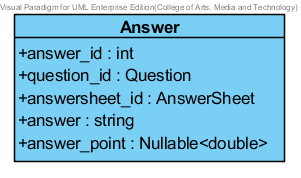
This class is a model class for answer sheet model which consists answer sheet information, student id for identifying who is student that take testing, and testing id for identifying testing in an answer sheet.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 9.1 | answerSheet\_id | Unique Identifier of Answer sheet | int |
| 9.2 | testing\_id | Object for identifying Testing | Testing |
| 9.3 | student\_id | Object for identifying Student | Student |
| 9.4 | answerSheet\_score | Score of answer sheet. Score can null status. | Nullable<double> |
| 9.5 | answerSheet\_time | Time of answer sheet in submission. | DateTime |
| 9.6 | answerSheet\_submitResult | Submit result of answer sheet (1 is early deal line of testing’s submission time, 0 is late deal line of testing’s submission time) | Nullable<int> |

**3.1.10 Class-10. Answer**

Class Answer is Model-10 (MD-10)



**Description**

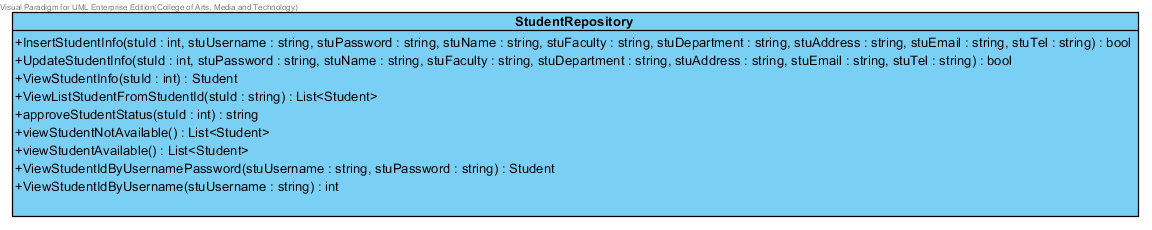
This class is a model class for answer model which consists answer information, answer sheet id for identifying answer sheet that keep answers, and question id for identifying a question that student create an answer.

**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 10.1 | answer\_id | Unique Identifier of Answer | int |
| 10.2 | question\_id | Object for identifying Question | Question |
| 10.3 | answersheet\_id | Object for identifying Answer sheet | AnswerSheet |
| 10.4 | answer | Description of answer | string |
| 10.5 | answer\_point | Score of correctness of answer | Nullable<double> |

**3.1.11 Class-11. StudentRepository**

Class StudentRepository is Repository-01 (MD-11)

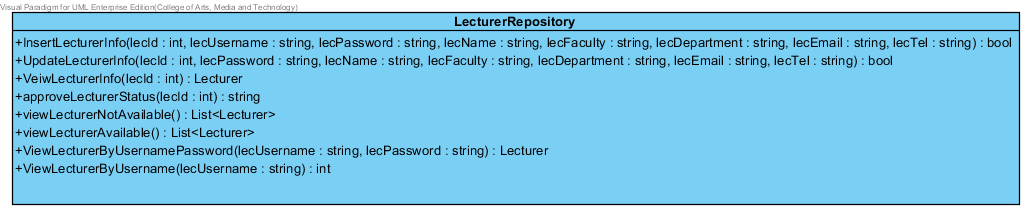


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 11.1 | InsertStudentInfo | This method is used for inserting a student account to the database. | stuId: int, stuUsername: string, stuPassword: string, stuName: string, stuFaculty: string, stuDepartment: string, stuAddress: string, stuEmail: string, stuTel: string | bool |
| 11.2 | UpdateStudentInfo | This method is used for updating student information into the database. | stuId: int, stuPassword: string, stuName: string, stuFaculty: string, stuDepartment: string, stuAddress: string, ,string, stuTel: string | bool |
| 11.3 | ViewStudentInfo | This method is used for viewing student information from the database. | stuId: int | Student |
| 11.4 | ViewListStudentFromStudentId | This method is used for viewing a list of student information from the database. Input of this method is many student id in the form of string. | stuId: string | List<Student> |
| 11.5 | approveStudentStatus | This method is used for updating student approval status in the database. | stuId: int | string |
| 11.6 | viewStudentNotAvailable | This method is used for viewing student accounts that still does not approved. | None | List<Student> |
| 11.7 | viewStudentAvailable | This method is used for viewing student accounts that are approved. | None | List<Student> |
| 11.8 | ViewStudentByUsernamePassword | This method is used for viewing a Student object. | stuUsername: string, stuPassword: string | Student |
| 11.9 | ViewStudentIdByUsername | This method is used for viewing a Student id. | stuUsername: string | int |

**3.1.12 Class-12. LecturerRepository**

Class LecturerRepository is Repository-02 (MD-12)

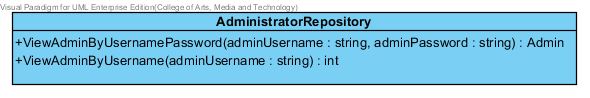


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 12.1 | InsertLecturerInfo | This method is used for inserting a lecturer account to the database. | lecId: int, lecUsername: string, lecPassword: string, lecName: string, lecFaculty: string, lecDepartment: string, lecEmail: string, lecTel: string | bool |
| 12.2 | UpdateLecturerInfo | This method is used for updating lecturer information into the database. | lecId: int, lecPassword: string, lecName: string, lecFaculty: string, lecDepartment: string, lecEmail: string, lecTel: string | bool |
| 12.3 | VeiwLecturerInfo | This method is used for viewing lecturer information from the database. | lecId: int | Lecturer |
| 12.4 | approveLecturerStatus | This method is used for updating lecturer approval status in the database. | lecId: int | string |
| 12.5 | viewLecturerNotAvailable | This method is used for viewing lecturer accounts that still does not approved. | None | List<Lecturer> |
| 12.6 | viewLecturerAvailable | This method is used for viewing lecturer accounts that are approved. | None | List<Lecturer> |
| 12.7 | ViewLecturerByUsernamePassword | This method is used for viewing a lecturer object. | lecUsername: string, lecPassword: string | Lecturer |
| 12.8 | ViewLecturerByUsername | This method is used for viewing a lecturer id. | lecUsername: string | int |

**3.1.13 Class-13. AdministratorRepository**

Class AdministratorRepository is Repository-03 (MD-13)

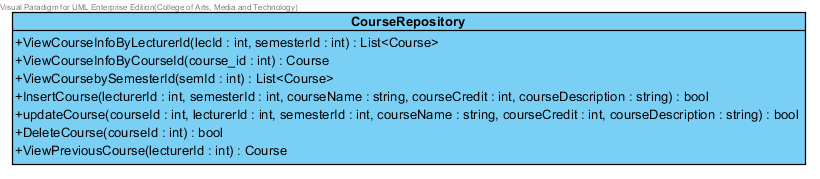


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 13.1 | ViewAdminByUsernamePassword | This method is used for viewing an administrator object by using username and password in searching administrator. | adminUsername: string, adminPassword: string | Admin |
| 13.2 | ViewAdminByUsername | This method is used for viewing an administrator id by using username in searching administrator id. | adminUsername: string | int |

**3.1.14 Class-14. CourseRepository**

Class CourseRepository is Repository-04 (MD-14)

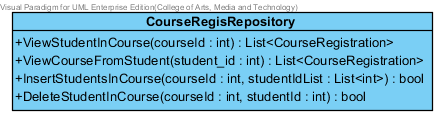


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 14.1 | ViewCourseInfoByLecturerId | This method is used for viewing a list of courses that Lecturer teaches. | lecId: int, semesterId: int | List<Course> |
| 14.2 | ViewCourseInfoByCourseId | This method is used for view course object by using course id in searching course object. | course\_id: int | Course |
| 14.3 | viewCoursebySemesterId | The method is used for viewing a list of courses in each semester by using semester id in searching list of course. | semId : int | List<Course> |
| 14.4 | InsertCourse | This method is used for inserting course information to the database. | lecturerId: int, semesterId: int, courseName: string, courseCredit: int, courseDescription: string | bool |
| 14.5 | updateCourse | This method is used for updating course information in the database | courseId: int, lecturerId: int, semesterId: int, courseName: string, courseCredit: int, courseDescription: string | bool |
| 14.6 | DeleteCourse | This method is used for deleting course information in the database. | courseId: int | bool |
| 14.7 | ViewPreviousCourse | The method is used for viewing a course that was created by a lecturer. | lecturerId: int | Course |

**3.1.15 Class-15. CourseRegisRepository**

Class CourseRegisRepository is Repository-05 (MD-15)

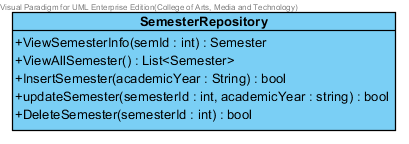


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 15.1 | ViewStudentInCourse | This method is used for viewing a list of students that registered in course. | courseId: int | List<CourseRegistration> |
| 15.2 | ViewCourseFromStudent | This method is used for viewing a list of courses that student registered. | student\_id: int | List<CourseRegistration> |
| 15.3 | InsertStudentsInCourse | This method is used for registering list of students in course. | courseId: int, studentIdList: List<int> | bool |
| 15.4 | DelelteStudentInCourse | This method is used for deleting a student in course. | courseId: int, studentId: int | bool |

**3.1.16 Class-16. SemesterRepository**

Class SemesterRepository is Repository-06 (MD-16)

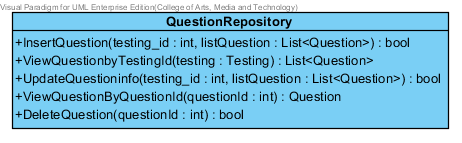


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 16.1 | ViewSemesterInfo | This method is used for viewing semester information. | semId: int | Semester |
| 16.2 | ViewAllSemester | The method is used for viewing all semester that was created. | None | List<Semester> |
| 16.3 | InsertSemester | This method is used for inserting a semester information to the database. | academicYear: String | bool |
| 16.4 | updateSemester | This method is used for updating a semester information in the database. | semesterId: int, academicYear: string | bool |
| 16.5 | DeleteSemester | This method is used for deleting a semester in the database | semesterId: int | bool |

**3.1.17 Class-17. QuestionRepository**

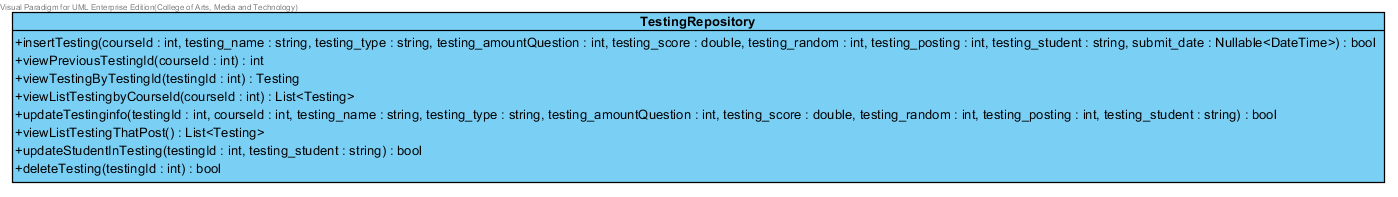
Class QuestionRepository is Repository-07 (MD-17)



**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 17.1 | InsertQuestion | This method is used for inserting list of questions to the database. | testing\_id: int, listQuestion: List | bool |
| 17.2 | ViewQuestionbyTestingId | This method is used for viewing a list of questions in testing. | testing: Testing | List<Question> |
| 17.3 | UpdateQuestioninfo | This method is used for updating list of question in the database. | testing\_id: int, listQuestion: List | bool |
| 17.4 | ViewQuestionByQuestionId | This method is used for viewing question information by using question id in searching. | questionId: int | Question |
| 17.5 | DeleteQuestion | This method is used for deleting question information in the database. | questionId : int | bool |

**3.1.18 Class-18. TestingRepository**

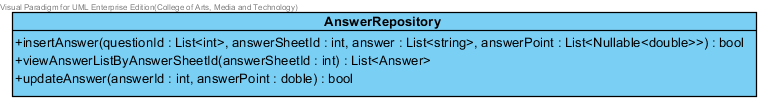
Class TestingRepository is Repository-08 (MD-18)

**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 18.1 | insertTesting | The method is used for inserting testing information to the database. | 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 |
| 18.2 | viewPreviousTestingId | This method is used for viewing a testing that was created in a course. | courseId: int | int |
| 18.3 | viewTestingByTestingId | This method is used for viewing testing information by using testing id in searching. | testingId: int | Testing |
| 18.4 | viewListTestingbyCourseId | This method is used for viewing a list of testing in a course. | courseId: int | List<Testing> |
| 18.5 | updateTestinginfo | This method is used for updating testing information in the database. | 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 |
| 18.6 | viewListTestingThatPost | This method is used for viewing a list of testing that are post status. | None | List<Testing> |
| 18.7 | updateStudentInTesting | This method is used for updating students in a testing. | testingId : int, testing\_student : string | bool |
| 18.8 | DeleteTesting | This method is used for deleting testing in the database. | testingId : int | bool |

**3.1.19 Class-19. AnswerRepository**

Class AnswerRepository is Repository-09 (MD-19)

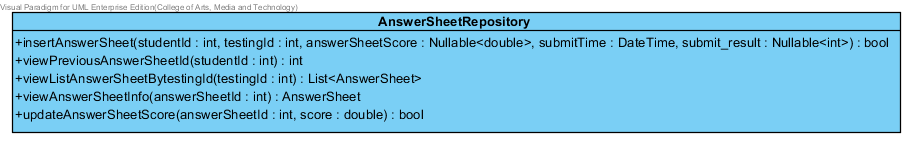


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 19.1 | insertAnswer | This method is used for inserting a list of answers to the database | questionId : List<int>, answerSheetId : int, answer : List<string>, answerPoint : List<Nullable<double>> | bool |
| 19.2 | viewAnswerListByAnswerSheetId | This method is used for viewing a list of answers in an answer sheet. | answerSheetId : int | List<Answer> |
| 19.3 | updateAnswer | This method is used for updating point of answer in the database. | answerId : int, point : double | bool |

**3.1.20 Class-20. AnswerSheetRepository**

Class AnswerSheetRepository is Repository-10 (MD-20)

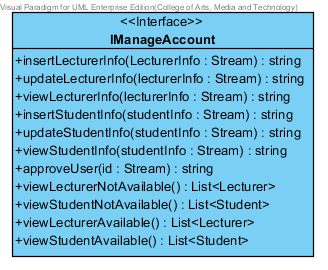


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 20.1 | insertAnswerSheet | This method is used for inserting an answer sheet to the database. | studentId : int, testingId : int, answerSheetScore : Nullable<double>, submitTime : DateTime, submit\_result : Nullable<int> | bool |
| 20.2 | viewPreviousAnswerSheetId | This method is used for viewing answer sheet id that student take a testing in latest. | studentId :int | int |
| 20.3 | viewListAnswerSheetBytestingId | This method is used for viewing a list of answer sheets in a testing. | testingId : int | List<AnswerSheet> |
| 20.4 | viewAnswerSheetInfo | This method is used for viewing answer sheet information in the database | answerSheetId : int | AnswerSheet |
| 20.5 | updateAnswerSheetScore | This method is used for updating score of answer sheet in the database. | answerSheetId : int, score : double | bool |

**3.1.21 Class-21. IManageAccount**

Class IManageAccount is Interface class-01 (MD-21)

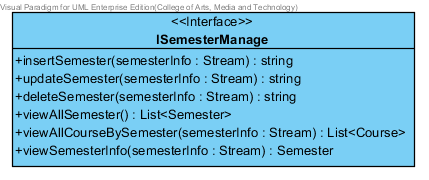


**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Parameters | Return |
| 21.1 | insertLecturerInfo | lecturerInfo: Stream | string |
| 21.2 | updateLecturerInfo | lecturerInfo: Stream | string |
| 21.3 | viewLecturerInfo | lecturerInfo: Stream | string |
| 21.4 | insertStudentInfo | studentInfo: Stream | string |
| 21.5 | updateStudentInfo | studentInfo: Stream | string |
| 21.6 | viewStudentInfo | studentInfo: Stream | string |
| 21.7 | approveUser | id: Stream | string |
| 21.8 | viewLecturerNotAvailable | None | List<Lecturer> |
| 21.9 | viewStudentNotAvailable | None | List<Student> |
| 21.10 | viewLecturerAvailable | None | List<Lecturer> |
| 21.11 | viewStudentAvailable | None | List<Student> |

**3.1.22 Class-22. ISemesterManage**

Class ISemesteManage is Interface class-02 (MD-22)

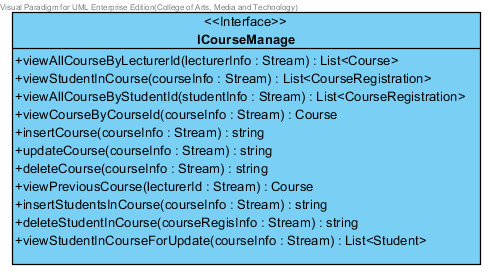


**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Parameters | Return |
| 22.1 | insertSemester | semesterInfo: Stream | string |
| 22.2 | updateSemester | semesterInfo: Stream | string |
| 22.3 | deleteSemester | semesterInfo: Stream | string |
| 22.4 | viewAllSemester | None | List<Semester> |
| 22.5 | viewAllCourseBySemester | semesterInfo: Stream | List<Course> |
| 22.6 | viewSemesterInfo | semesterInfo: Stream | Semester |

**3.1.23 Class-23. ICourseManage**

Class ICourseManage is Interface class-03 (MD-23)



**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Parameters | Return |
| 23.1 | viewAllCourseByLecturerId | lecturerInfo: Stream | List<Course> |
| 23.2 | viewStudentInCourse | courseInfo: Stream | List<CourseRegistration> |
| 23.3 | viewAllCourseByStudentId | studentInfo: Stream | List<CourseRegistration> |
| 23.4 | viewCourseByCourseId | courseInfo: Stream | Course |
| 23.5 | insertCourse | courseInfo: Stream | string |
| 23.6 | updateCourse | courseInfo: Stream | string |
| 23.7 | deleteCourse | courseInfo: Stream | string |
| 23.8 | viewPreviousCourse | lecturerId: Stream | Course |
| 23.9 | insertStudentsInCourse | courseInfo: Stream | string |
| 23.10 | deleteStudentInCourse | courseRegisInfo: Stream | string |
| 23.11 | viewStudentInCourseForUpdate | courseInfo: Stream | List<Student> |

**3.1.24 Class-24. ITestingManage**

Class ITestingManage is Interface class-04 (MD-24)

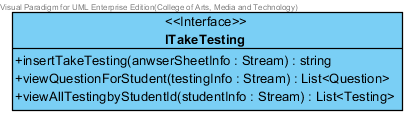


**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Parameters | Return |
| 24.1 | insertTesting | questionInfo: Stream | string |
| 24.2 | insertQuestion | questionInfo: Stream | string |
| 24.3 | viewQuestionInTesting | testingInfo : Stream | List<Questiom> |
| 24.4 | viewTestingInfo | testingInfo: Stream | Testing |
| 24.5 | viewAllTestingByCourseId | courseInfo: Stream | List<Testing> |
| 24.6 | updateTestingInfo | testingInfo: Stream | string |
| 24.7 | updateQuestionInfo | questionInfo: Stream | string |
| 24.8 | viewSubmitAnswerSheet | testingInfo : Stream | List<AnswerSheet> |
| 24.9 | viewStudentInTestingForUpdate | testingInfo : Stream | List<CourseRegistration> |
| 24.10 | updateStudentInTesting | testingInfo : Stream | String |
| 24.11 | viewAnswerByAnswerSheetId | answerSheetId : Stream | List<Answer> |
| 24.12 | updateAnswerSheetScore | answerInfor : Stream | string |
| 24.13 | viewChartInTesting | testingInfo : Stream | List<int> |
| 24.14 | deleteTesting | testingInfo : Stream | string |

**3.1.25 Class-25. ITakeTesting**

Class ITakeTesting is Interface class-05 (MD-25)



**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Parameters | Return |
| 25.1 | insertTakeTesting | anwserSheetinfo: Stream | string |
| 25.2 | viewQuestionForStudent | testinginfo: Stream | List<Questiom> |
| 25.3 | viewAllTestingbyStudentId | studentInfo : Stream | List<Testing> |

**3.1.26 Class-26. ITakeTesting**

Class ITakeTesting is Interface class-06 (MD-26)



**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Parameters | Return |
| 26.1 | Login | loginInfo : Stream | string |

**3.1.27 Class-27. ManageAccount**

Class ManageAccount is service class-01 (MD-27)

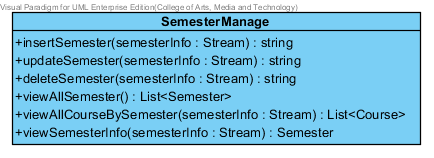


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 27.1 | insertLecturerInfo | This method is used for inserting lecturer information to the system. | lecturerInfo: Stream | string |
| 27.2 | updateLecturerInfo | This method is used for updating lecturer information in the system. | lecturerInfo: Stream | string |
| 27.3 | viewLecturerInfo | This method is used for viewing lecturer information from the system. | lecturerInfo: Stream | string |
| 27.4 | insertStudentInfo | This method is used for inserting student information in the system. | studentInfo: Stream | string |
| 27.5 | updateStudentInfo | This method is used for updating student information in the system. | studentInfo: Stream | string |
| 27.6 | viewStudentInfo | This method is used for viewing student information from the system. | studentInfo: Stream | string |
| 27.7 | approveUser | This method is used for updating lecturer or student approval status. | id: Stream | string |
| 27.8 | viewLecturerNotAvailable | This method is used for viewing lecturers that have approval status equal “0”. | None | List<Lecturer> |
| 27.9 | viewStudentNotAvailable | This method is used for viewing students that have approval status equal “0”. | None | List<Student> |
| 27.10 | viewLecturerAvailable | This method is used for viewing students that have approval status equal “1”. | None | List<Lecturer> |
| 27.11 | viewStudentAvailable | This method is used for viewing students that have approval status equal “1”. | None | List<Student> |

**3.1.28 Class-28. SemesterManage**

Class SemesteManage is service class -02 (MD-28)

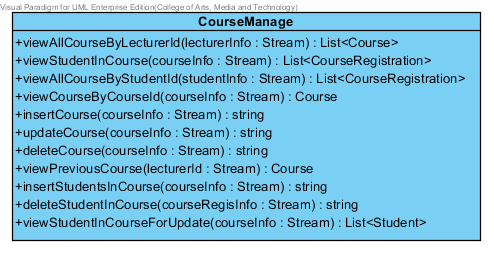


**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 28.1 | insertSemester | This method is used for inserting semester information to the system. | semesterInfo: Stream | string |
| 28.2 | updateSemester | This method is used for updating semester information in the system. | semesterInfo: Stream | string |
| 28.3 | deleteSemester | This method is used for deleting semester information from the system. | semesterInfo: Stream | string |
| 28.4 | viewAllSemester | This method is used for viewing all of semesters from the system. | None | List<Semester> |
| 28.5 | viewAllCourseBySemester | This method is used for viewing all of courses in a semester. | semesterInfo: Stream | List<Course> |
| 28.6 | viewSemesterInfo | This method is used for viewing semester information from the system. | semesterInfo: Stream | Semester |

**3.1.29 Class-29. CourseManage**

Class CourseManage is service class -03 (MD-29)

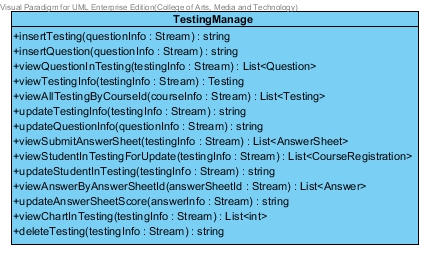
****

**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 29.1 | viewAllCourseByLecturerId | This method is used for viewing a list of courses that are taught by a lecturer. | lecturerInfo: Stream | List<Course> |
| 29.2 | viewStudentInCourse | This method is used for viewing a list of students that register in a course. | courseInfo: Stream | List<CourseRegistration> |
| 29.3 | viewAllCourseByStudentId | This method is used for viewing a list of courses that are registered by a student. | studentInfo: Stream | List<CourseRegistration> |
| 29.4 | viewCourseByCourseId | This method is used for viewing course information from the database. | courseInfo: Stream | Course |
| 29.5 | insertCourse | This method is used for inserting course information to the system. | courseInfo: Stream | string |
| 29.6 | updateCourse | This method is used for updating course information in the system. | courseInfo: Stream | string |
| 29.7 | deleteCourse | This method is used for deleting course information in the system. | courseInfo: Stream | string |
| 29.8 | viewPreviousCourse | This method is used for viewing latest course information of lecturer. | lecturerId: Stream | Course |
| 29.9 | insertStudentsInCourse | This method is used for inserting students to the course. | courseInfo: Stream | string |
| 29.10 | deleteStudentInCourse | This method is used for deleting students from the course. | courseRegisInfo: Stream | string |
| 29.11 | viewStudentInCourseForUpdate | This method is used for viewing students that aren’t registered in a course. | courseInfo: Stream | List<Student> |

**3.1.30 Class-30. TestingManage**

Class TestingManage is service class -04 (MD-30)

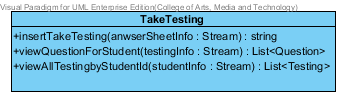
****

**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 30.1 | insertTesting | This method is used for inserting testing information to the system. | questionInfo: Stream | string |
| 30.2 | insertQuestion | This method is used for inserting questions to the system. | questionInfo: Stream | string |
| 30.3 | viewQuestionInTesting | This method is used for view a list of questions in a testing. | testingInfo : Stream | List<Questiom> |
| 30.4 | viewTestingInfo | This method is used for viewing testing information from the system | testingInfo: Stream | Testing |
| 30.5 | viewAllTestingByCourseId | This method is used for viewing all of testing in a course. | courseInfo: Stream | List<Testing> |
| 30.6 | updateTestingInfo | This method is used for updating testing information in the system. | testingInfo: Stream | string |
| 30.7 | updateQuestionInfo | This method is used for updating question information in the system. | questionInfo: Stream | string |
| 30.8 | viewSubmitAnswerSheet | This method is used for viewing all of answer sheets that students submit in a course. | testingInfo : Stream | List<AnswerSheet> |
| 30.9 | viewStudentInTestingForUpdate | This method is used for viewing student that don’t have in a testing. | testingInfo : Stream | List<CourseRegistration> |
| 30.10 | updateStudentInTesting | This method is used for updating students to access the testing | testingInfo : Stream | String |
| 30.11 | viewAnswerByAnswerSheetId | This method is used for viewing a list of answers in a answer sheet. | answerSheetId : Stream | List<Answer> |
| 30.12 | updateAnswerSheetScore | This method is used for updating answer sheet score. | answerInfor : Stream | string |
| 30.13 | viewChartInTesting | This method is used for viewing the chart of answers in testing | testingInfo : Stream | List<int> |
| 30.14 | deleteTesting | This method is used for deleting a testing in the system. | testingInfo : Stream | string |

**3.1.31 Class-31. TakeTesting**

Class TakeTesing is service class -05 (MD-31)

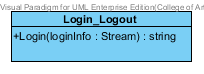
****

**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 31.1 | insertTakeTesting | This method is used for inserting an answer sheet, when student takes a testing. | anwserSheetinfo: Stream | string |
| 31.2 | viewQuestionForStudent | This method is used for viewing a list of questions in testing, when Student takes a testing. | testinginfo: Stream | List<Questiom> |
| 31.3 | viewAllTestingbyStudentId | This method is used for viewing all of testing which student can access to the testing. | studentInfo : Stream | List<Testing> |

**3.1.32 Class-32. Login\_Logout**

Class Login\_Logout is service class -06 (MD-32)

****

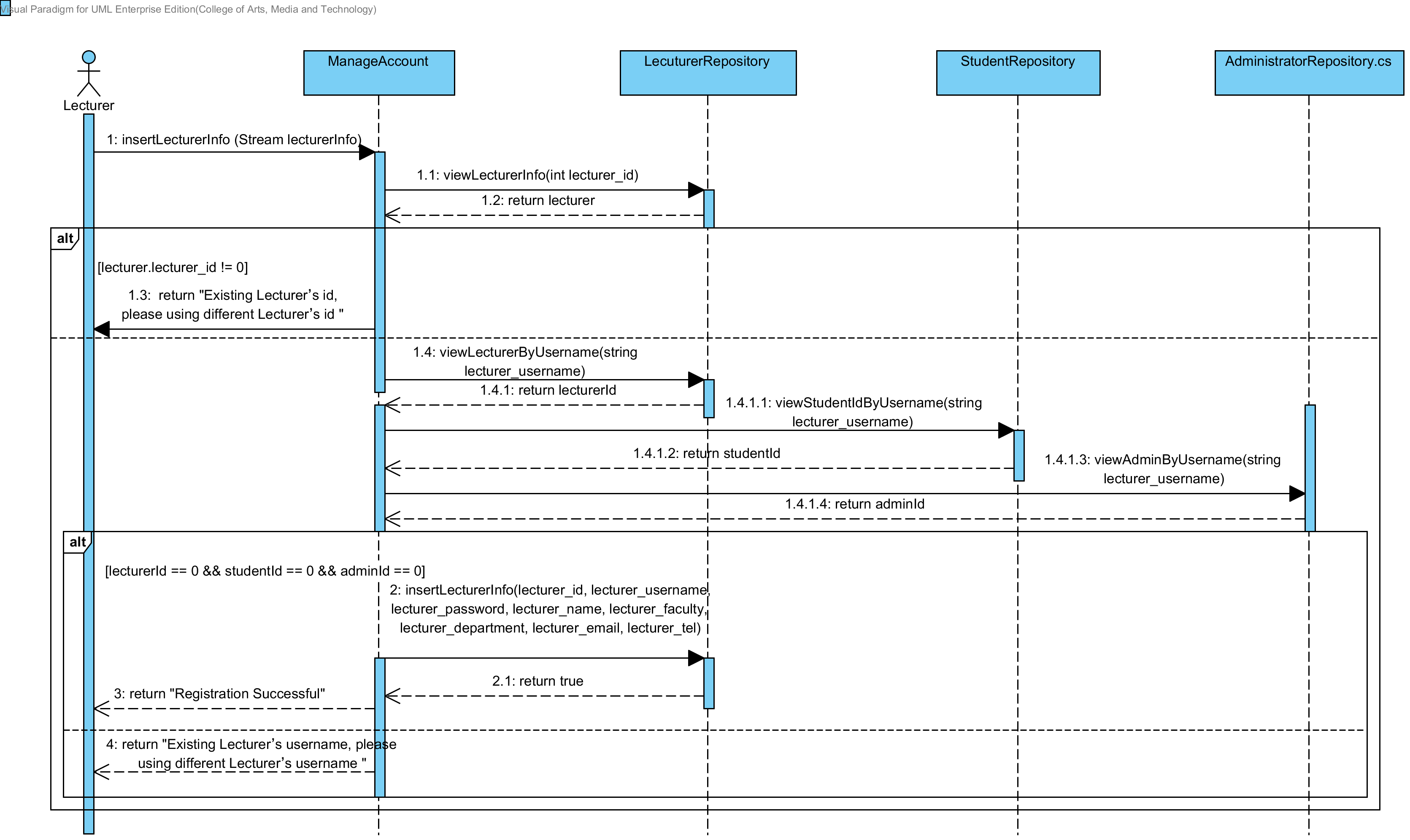
**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 32.1 | Login | This method is used for validating username and password, when user login to the system. | loginInfo : Stream | string |

## **3.2 Sequence Diagram**

**3.2.1 Sequence Diagram (SD 1.1): Lecturer registers to the system.**

**: This sequence diagram responses to URS 1.1: Lecturer can register to the system.**



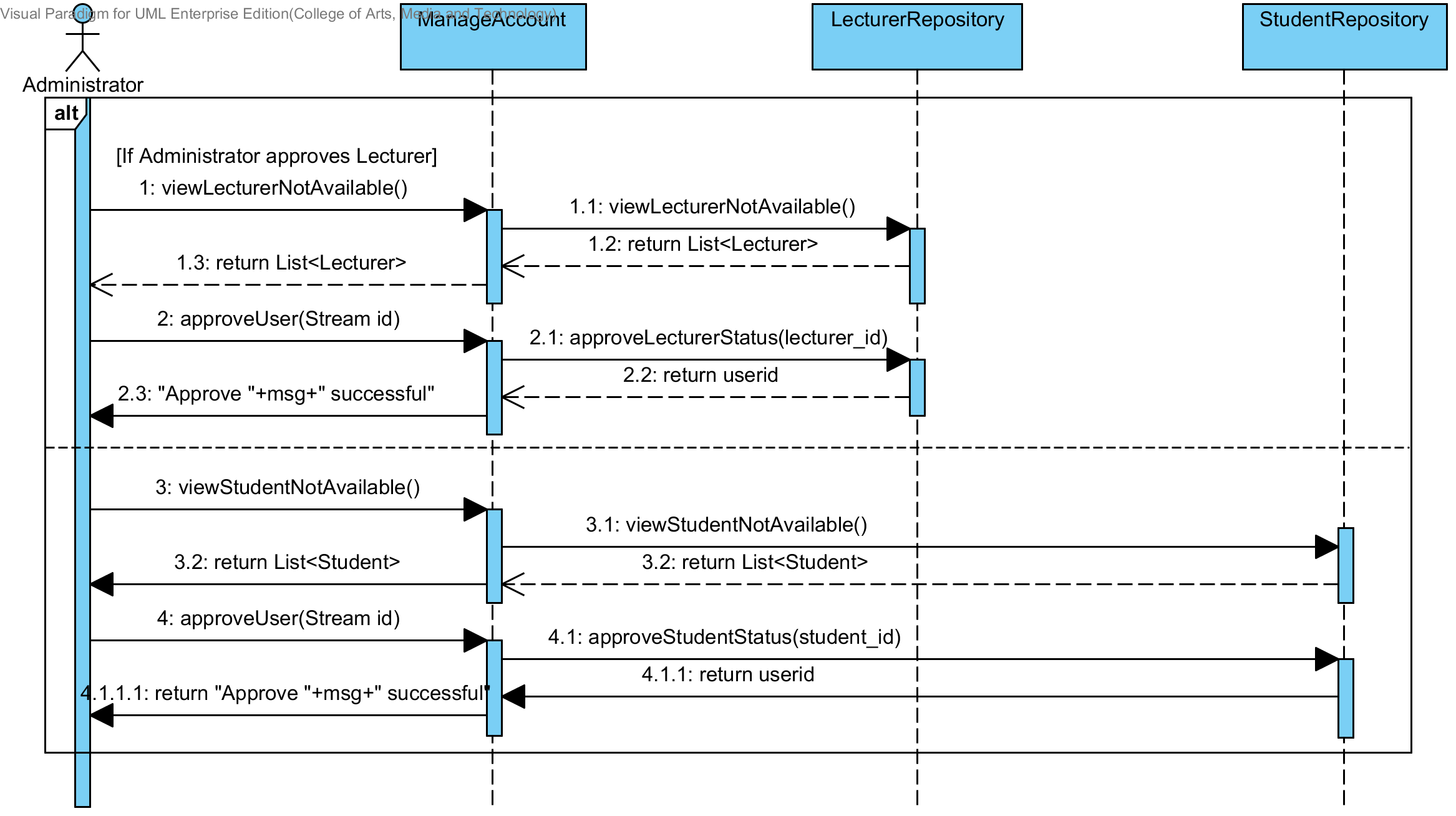
**3.2.2 Sequence Diagram (SD 1.2): Student registers to the system.**

**: This sequence diagram responses to URS 1.2: Student can register to the system.**

****

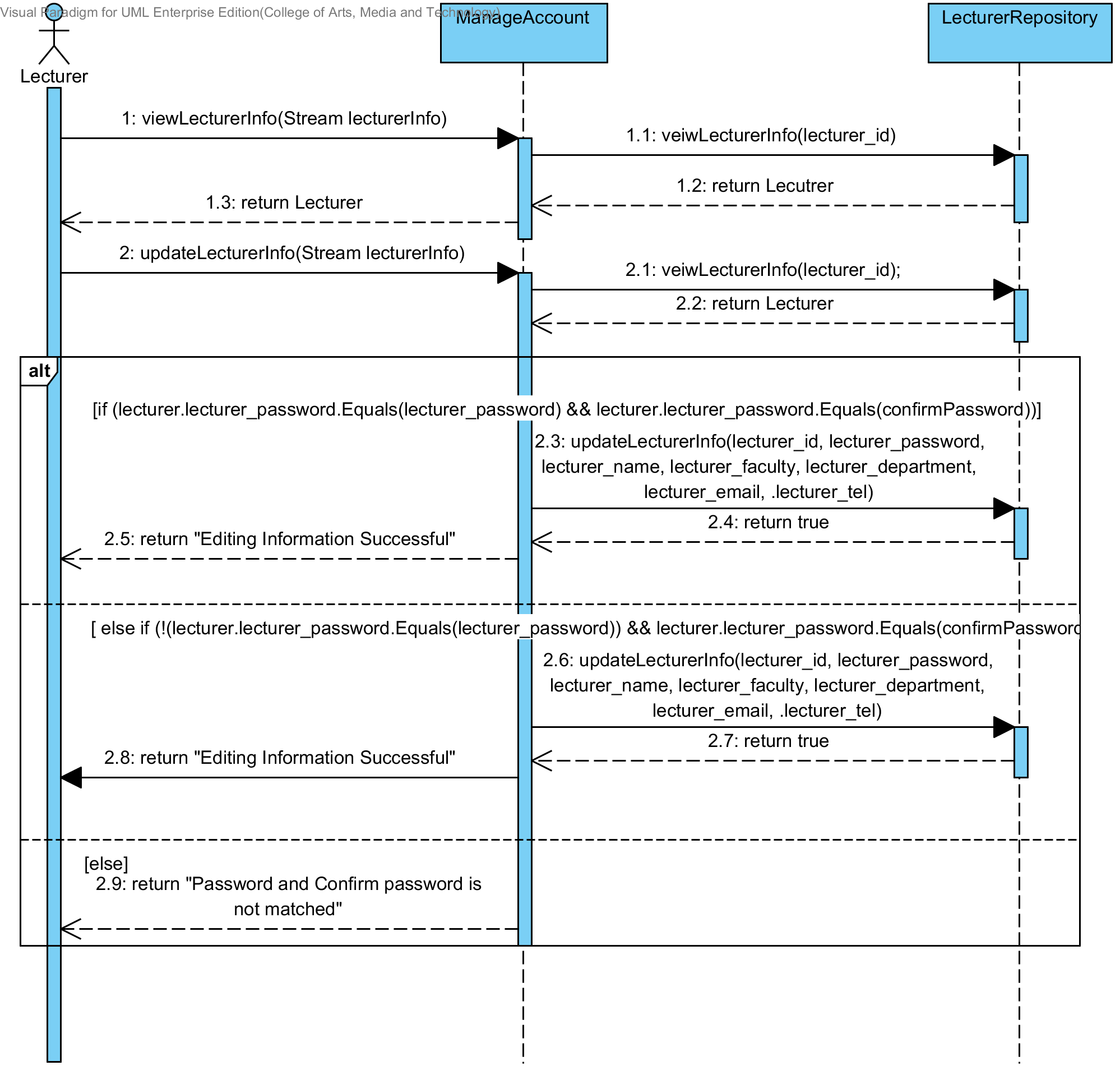
**3.2.3 Sequence Diagram (SD 1.3): Administrator approves Lecturer or Student account.**

**: This sequence diagram responses to URS 1.3: Administrator can approve Lecturer or Student account.**

****

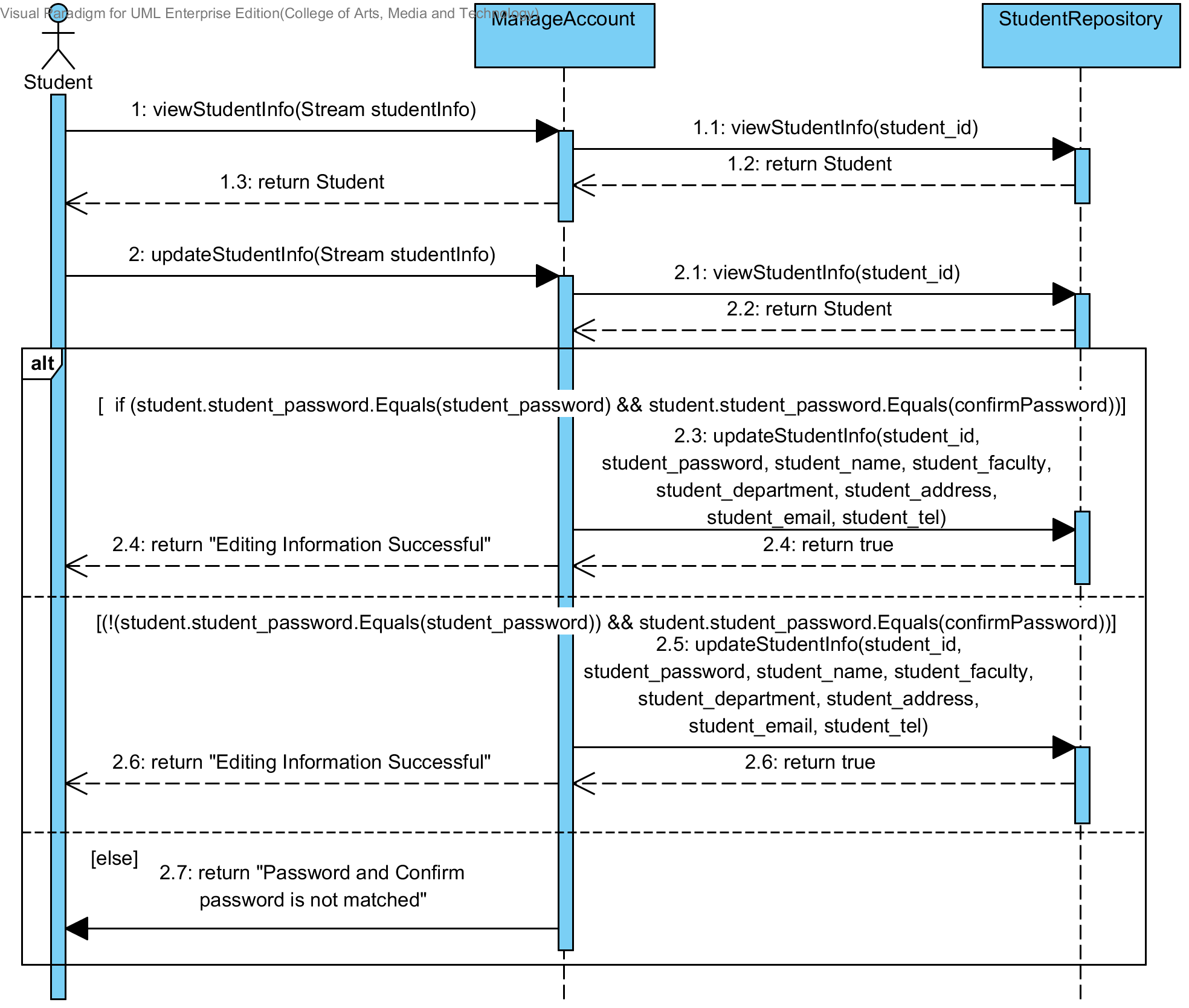
**3.2.4 Sequence Diagram (SD 1.4): Lecturer edits personal information.**

**: This sequence diagram responses to URS 1.4: Lecturer can edit personal information.**



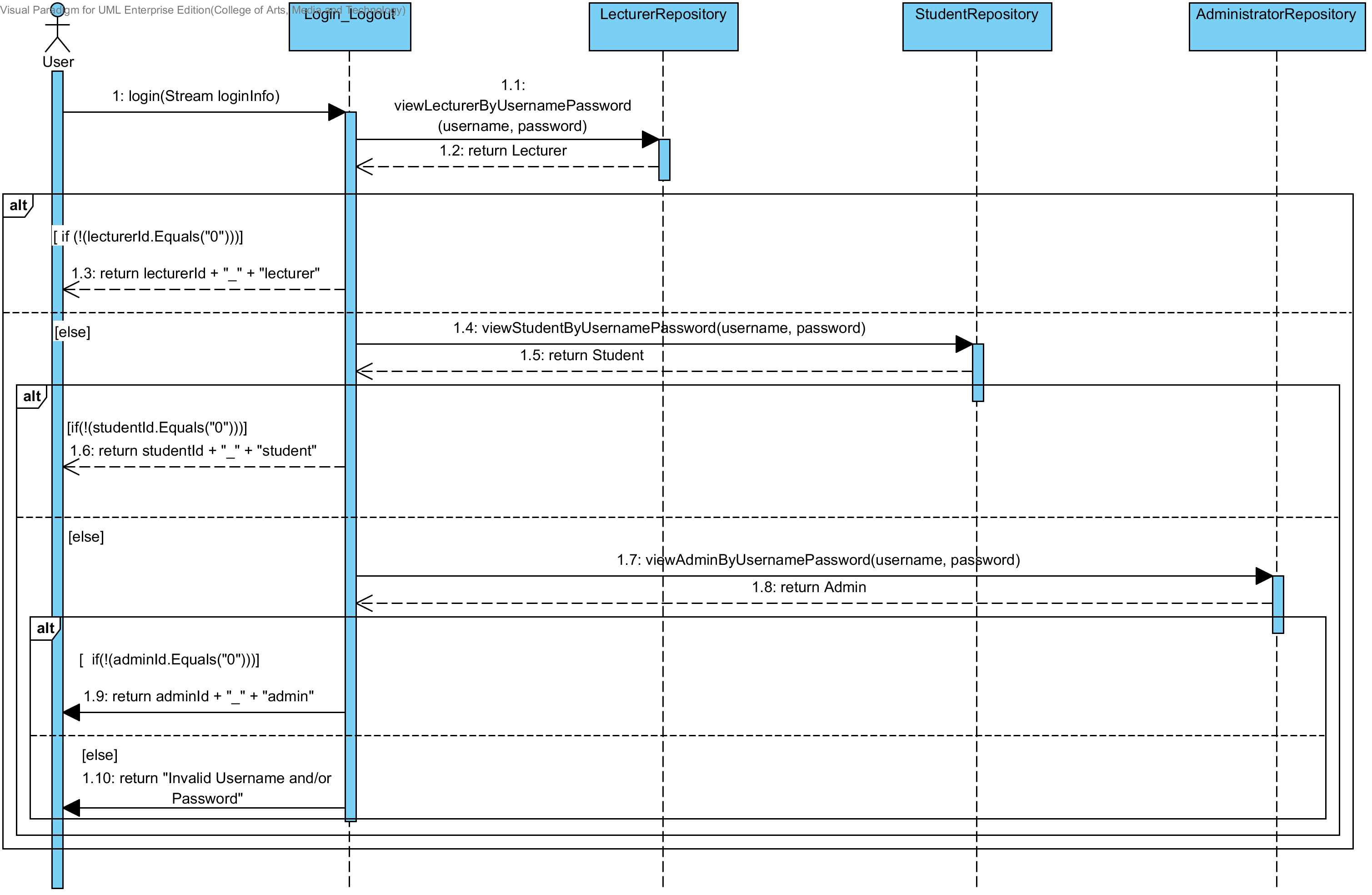
**3.2.5 Sequence Diagram (SD 1.5): Student edits personal information.**

**: This sequence diagram responses to URS 1.5: Student can edit personal information.**



**3.2.6 Sequence Diagram (SD 1.6): User log in to the system.**

**: This sequence diagram responses to URS 1.6: User can log in to the system.**



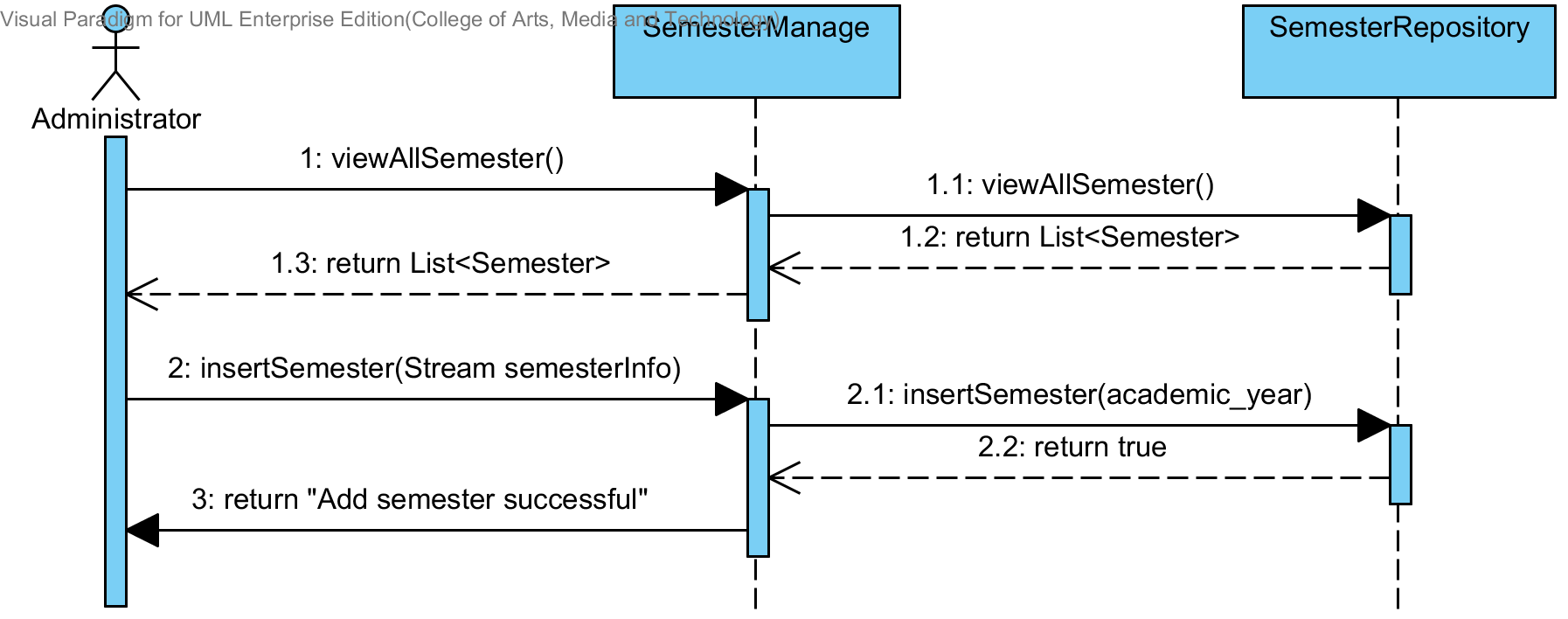
**3.2.7 Sequence Diagram (SD 1.7): User can log out to the system.**

**: This sequence diagram responses to URS 1.7: User can log out to the system**.

Not have sequence diagram

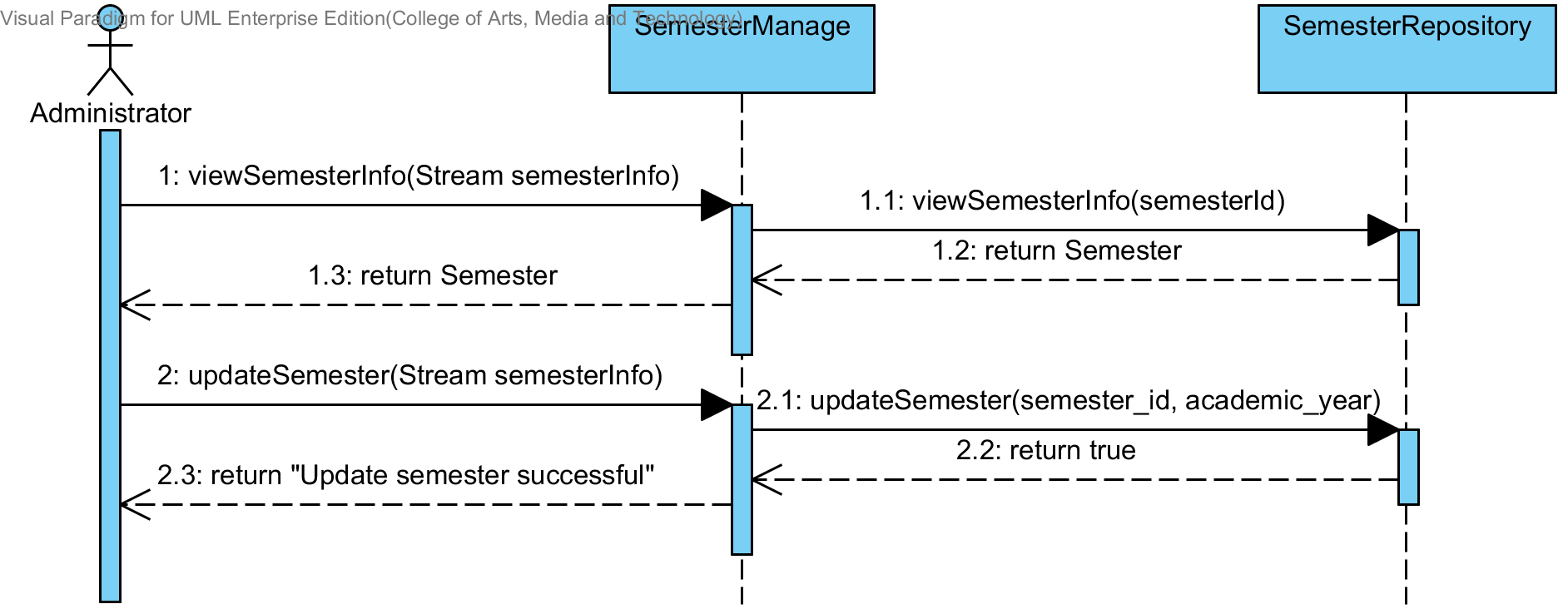
**3.2.8 Sequence Diagram (SD 1.8): Administrator creates a semester.**

**: This sequence diagram responses to URS 1.8: Administrator can create a semester.**



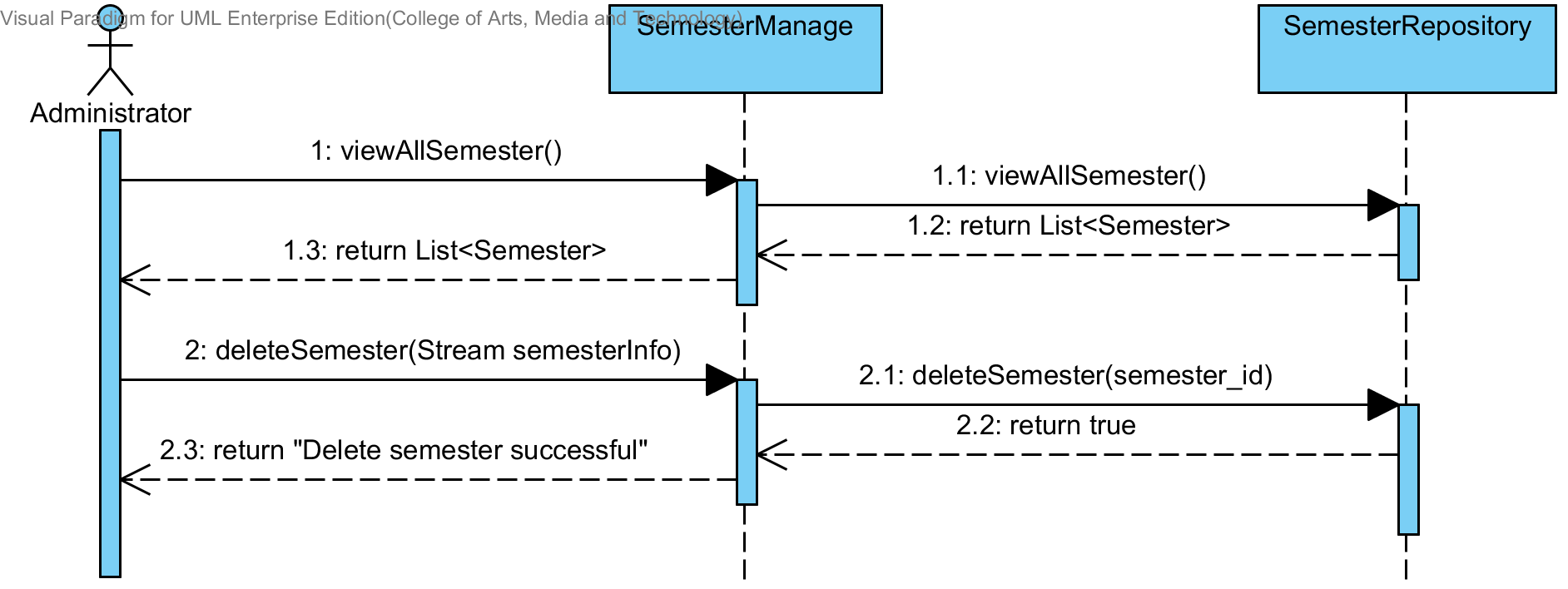
**3.2.9 Sequence Diagram (SD 1.9): Administrator edits a semester.**

**: This sequence diagram responses to URS 1.9: Administrator can edit a semester.**



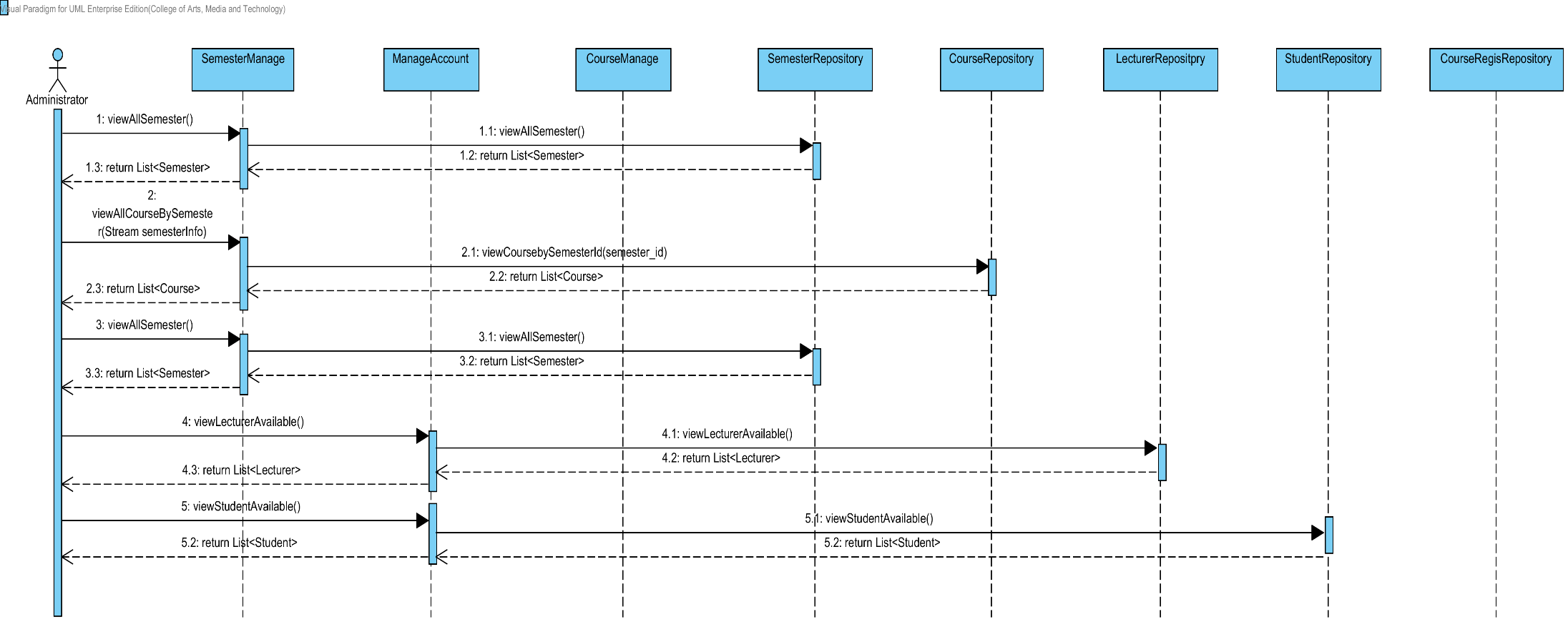
**3.2.10 Sequence Diagram (SD 1.10): Administrator deletes a semester.**

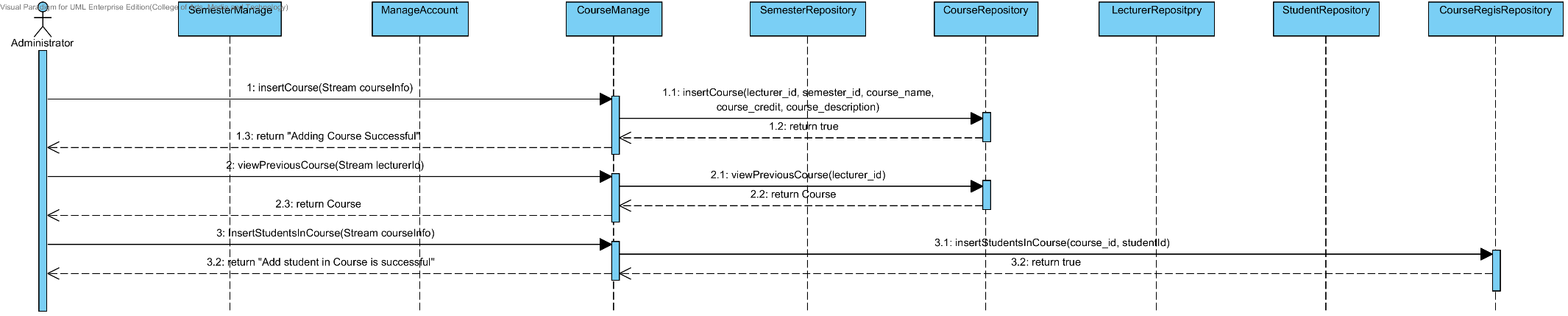
**: This sequence diagram responses to URS 1.10: Administrator can delete a semester.**



**3.2.11 Sequence Diagram (SD 1.11): Administrator creates a course.**

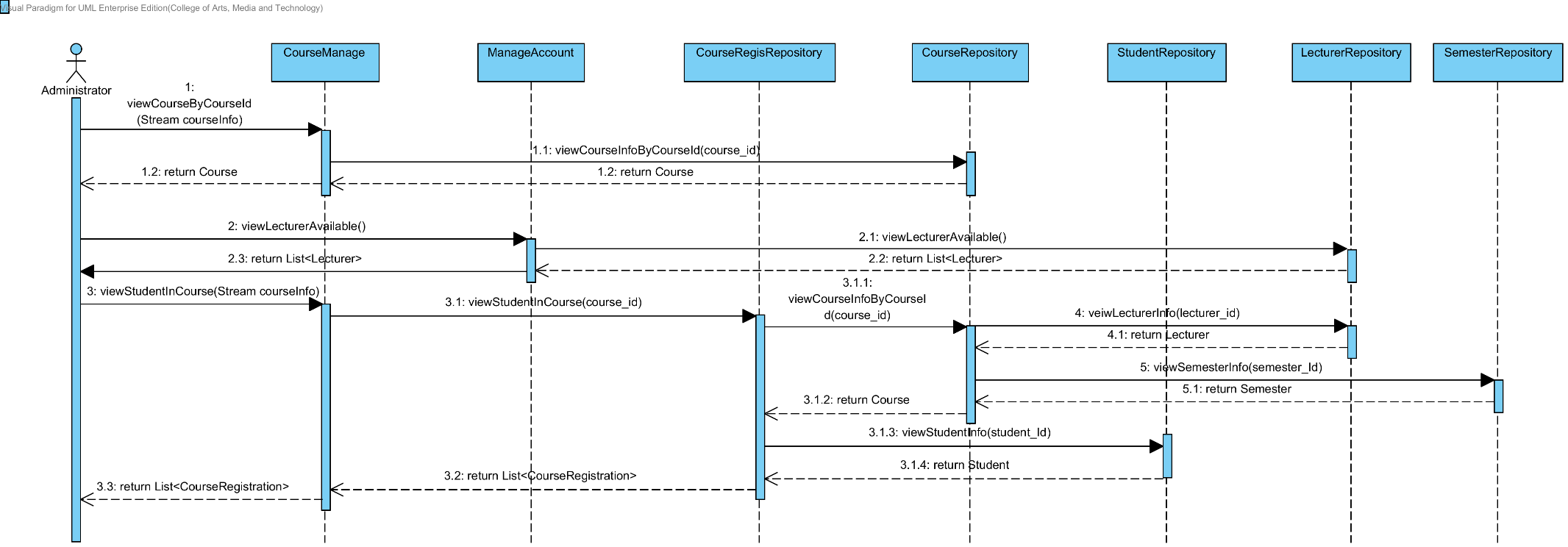
**: This sequence diagram responses to URS 1.11: Administrator can create a course.**

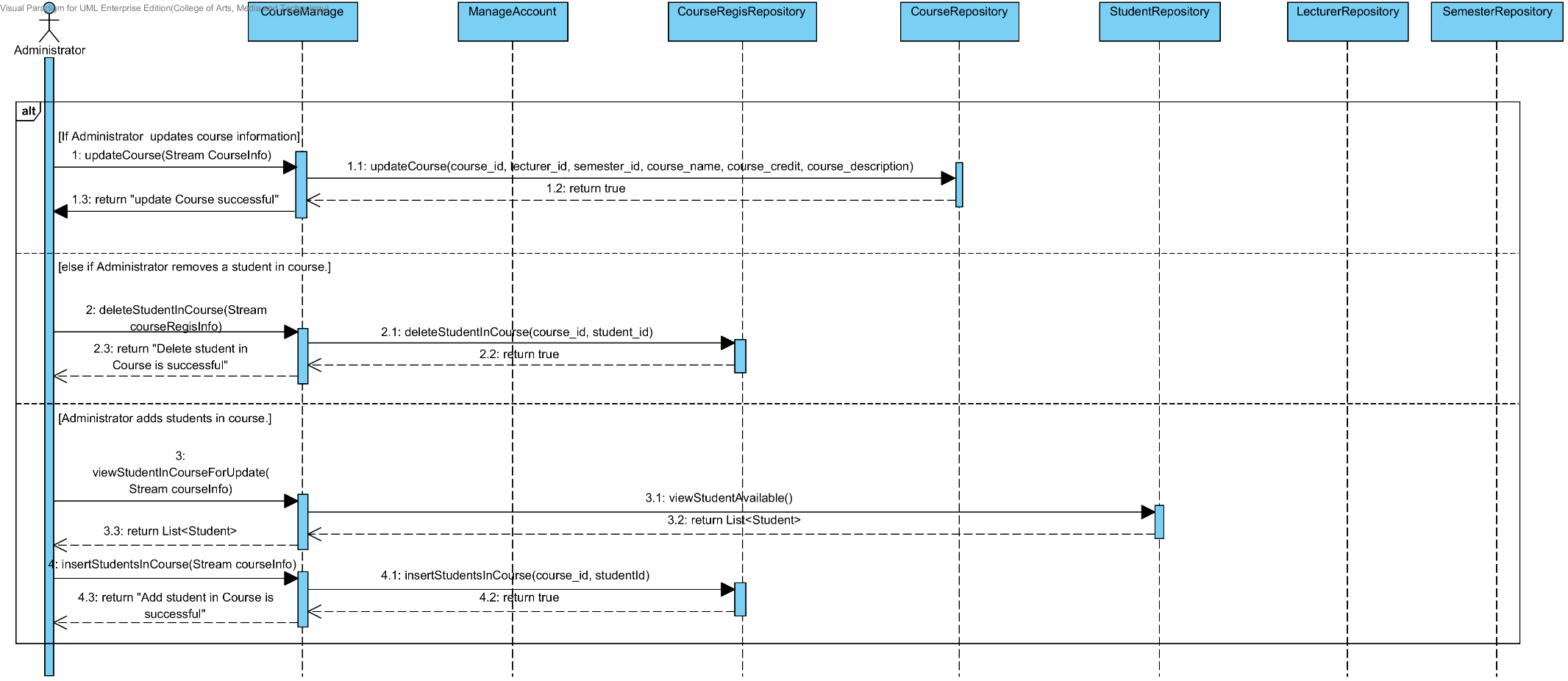




**3.2.12 Sequence Diagram (SD 1.12): Administrator edits a course.**

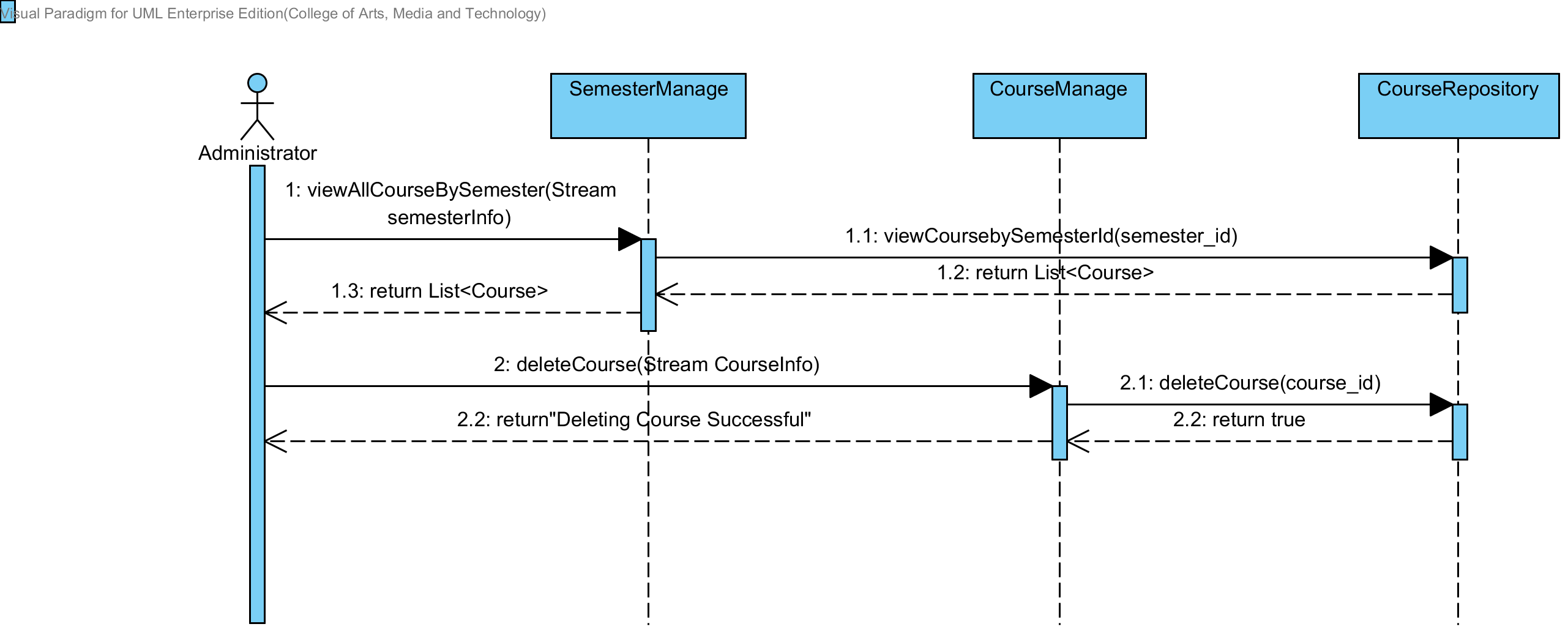
**: This sequence diagram responses to URS 1.12: Administrator can edit a course.**





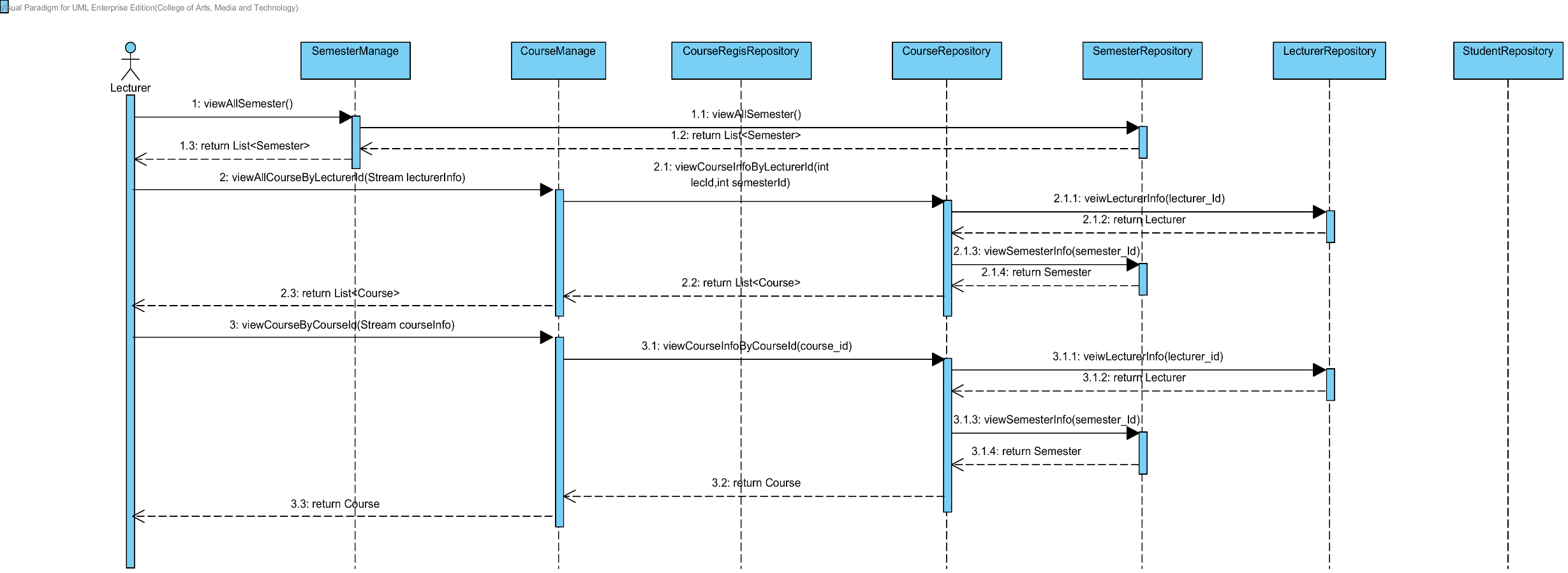
**3.2.13 Sequence Diagram (SD 1.13): Administrator deletes a course.**

**: This sequence diagram responses to URS 1.13: Administrator can delete a course.**



**3.2.14 Sequence Diagram (SD 1.14): Lecturer views course information.**

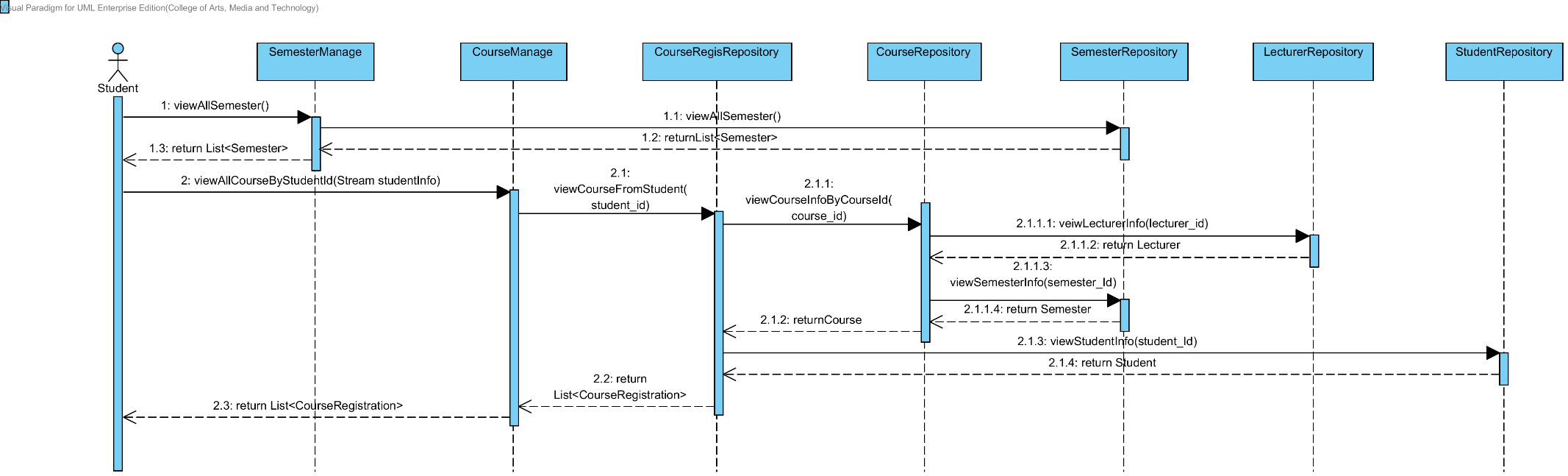
**: This sequence diagram responses to URS 1.14: Lecturer can view course information.**

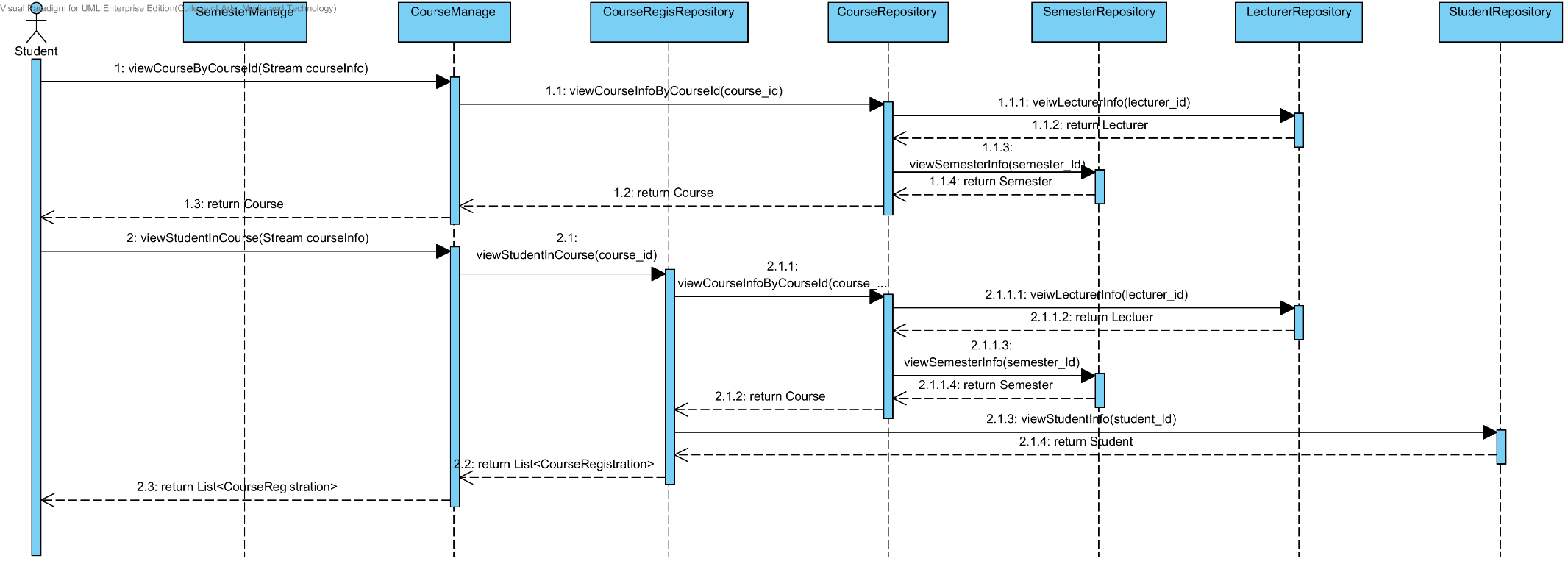




**3.2.15 Sequence Diagram (SD 1.15): Student views course information.**

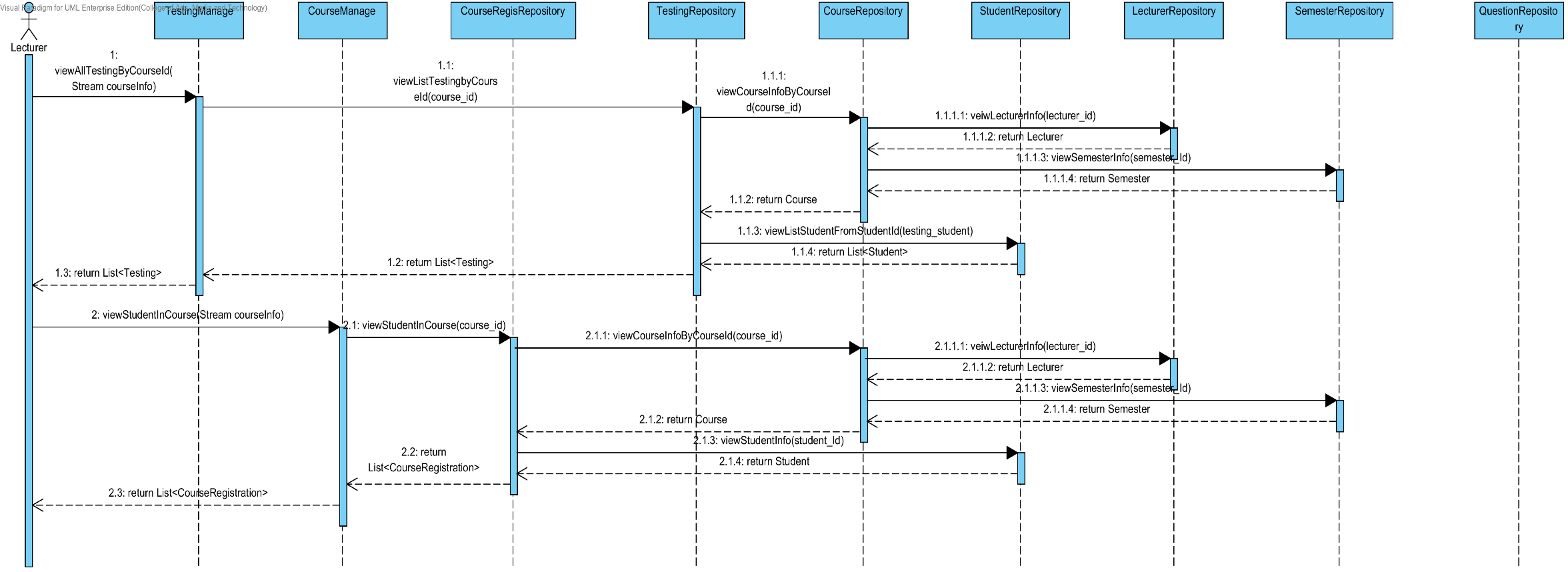
**: This sequence diagram responses to URS 1.15: Student can view course information.**

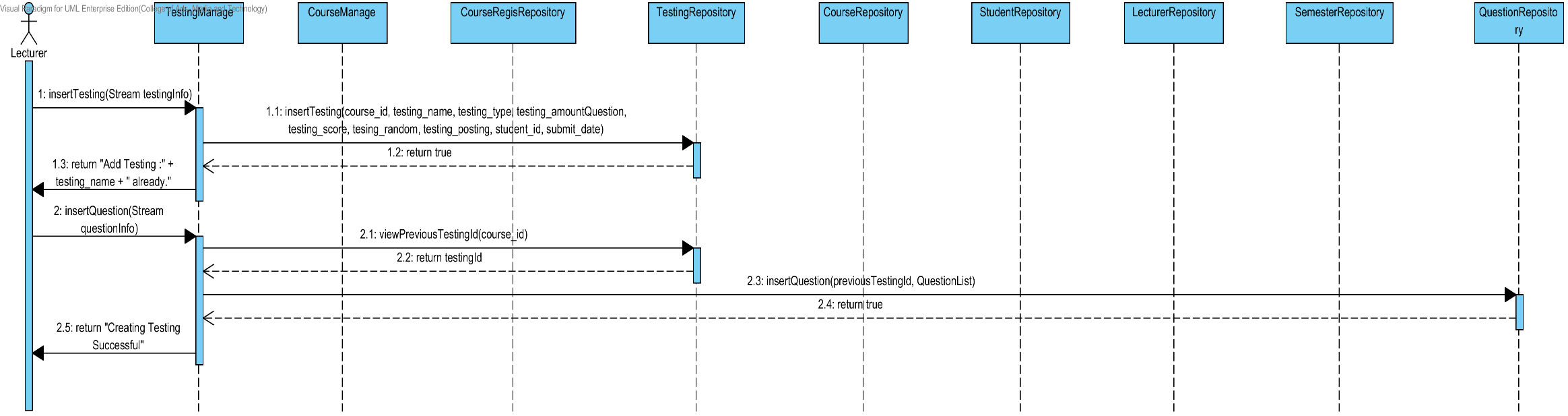




**3.2.16 Sequence Diagram (SD 1.16): Lecturer assigns assignments or quizzes which are multiple choices question, true/false question, and short answers**.

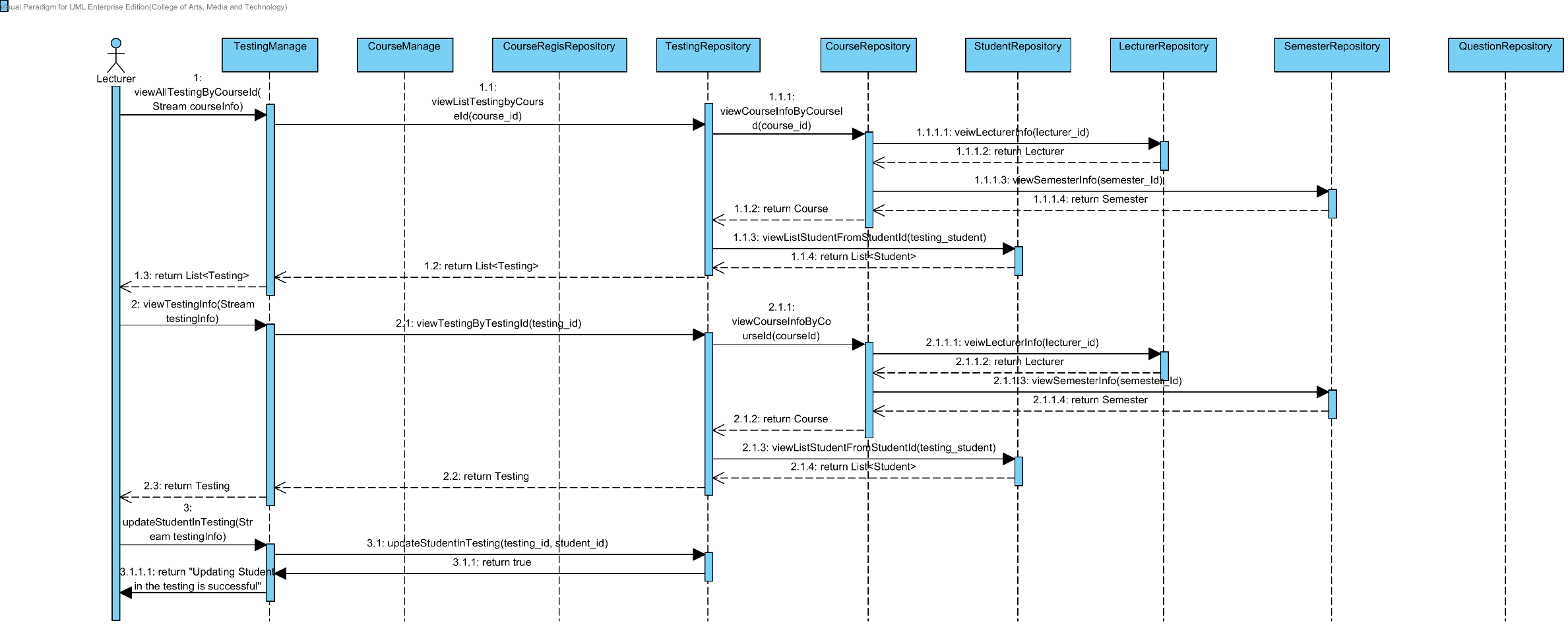
**: This sequence diagram responses to URS 1.16: Lecturer can assign assignments or quizzes which are multiple choices question, true/false question, and short answers**.

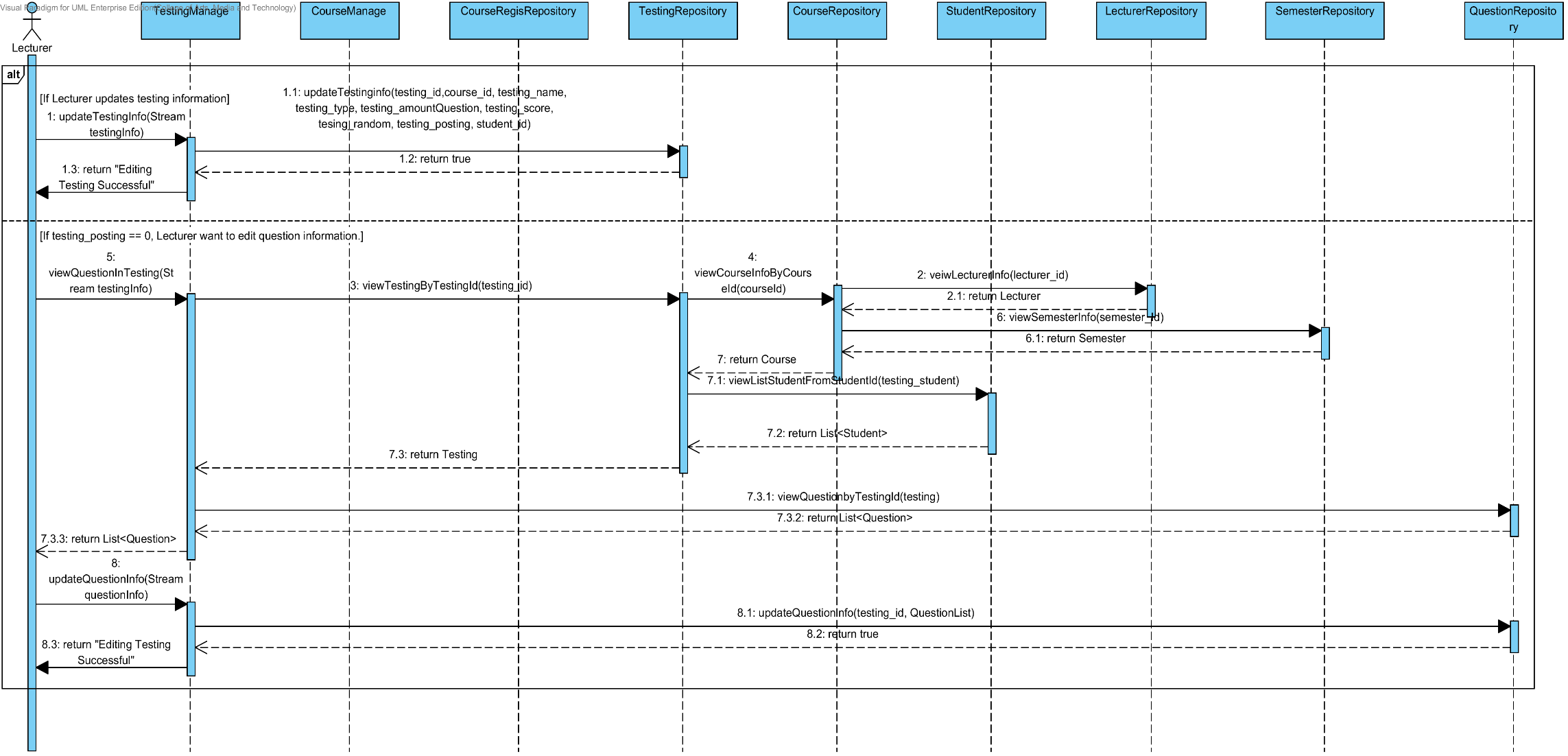




**3.2.17 Sequence Diagram (SD 1.17): Lecturer edits assignments or quizzes which are multiple choices question, true/false question, and short answers**.

**: This sequence diagram responses to URS 1.17: Lecturer can edit assignments or quizzes which are multiple choices question, true/false question, and short answers**.





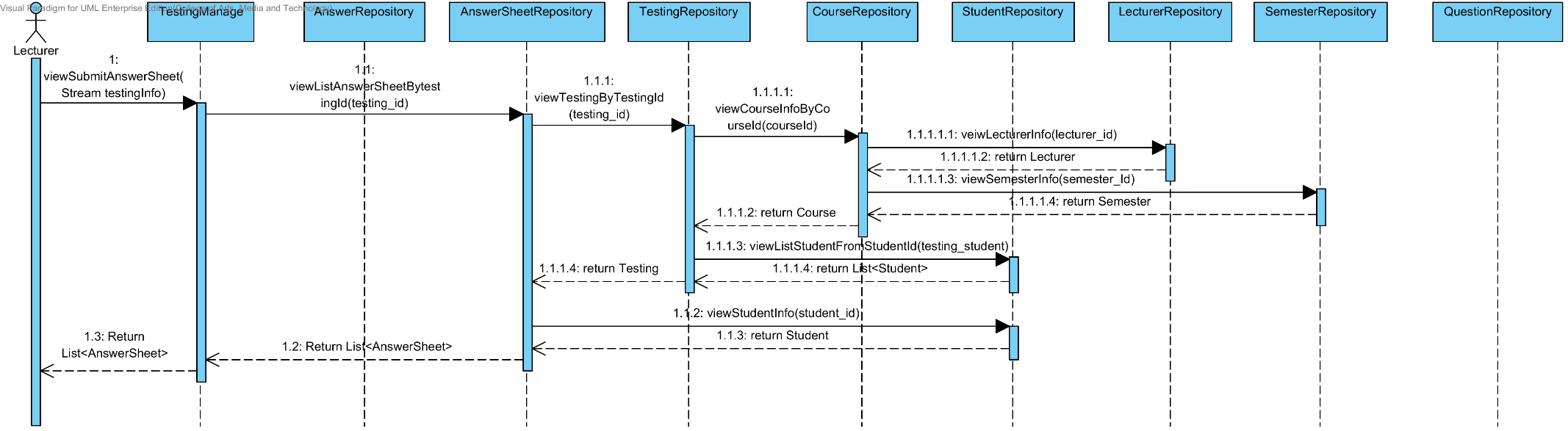
**3.2.18 Sequence Diagram (SD 1.18): Lecturer deletes assignments or quizzes which are multiple choices question, true/false question, and short answers**.

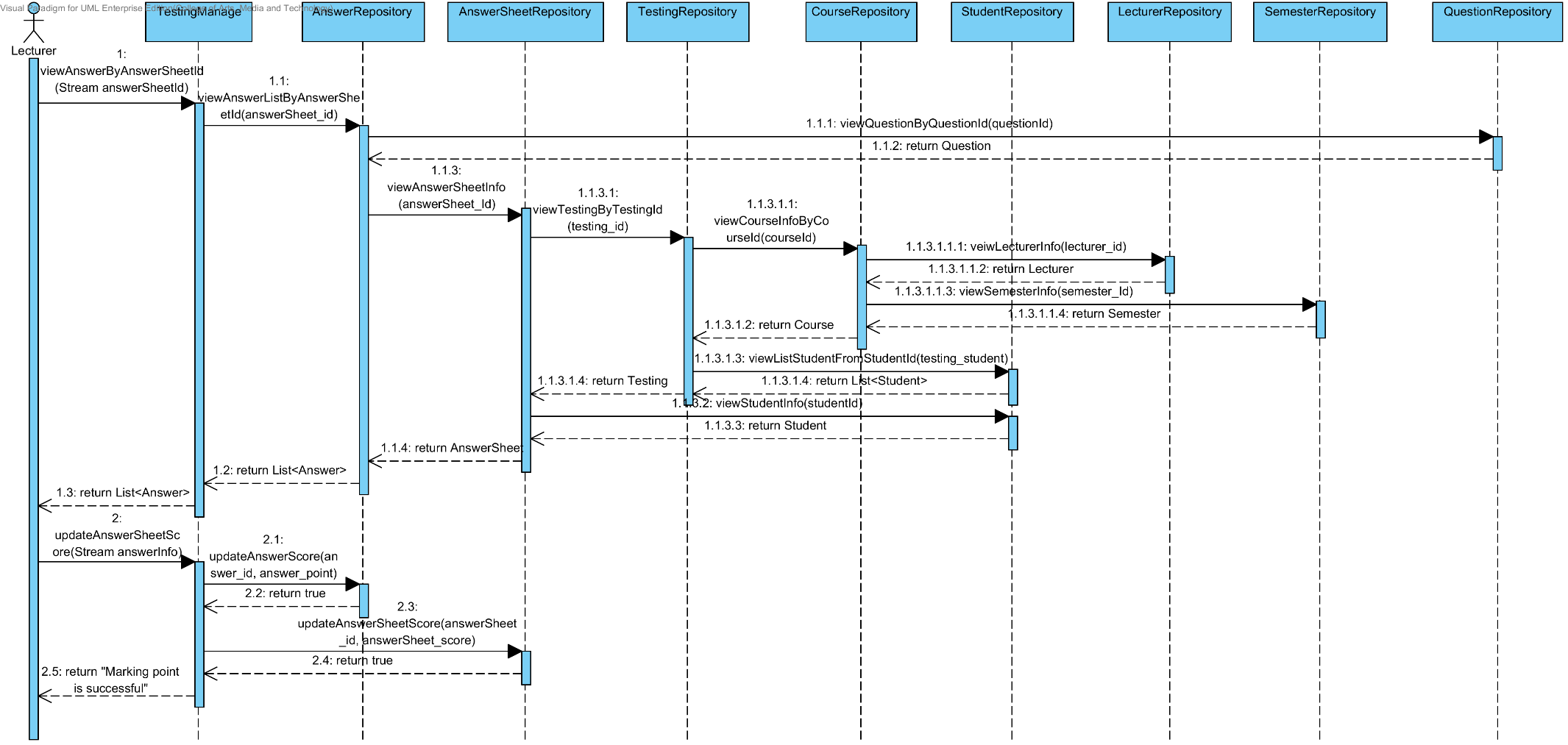
**: This sequence diagram responses to URS 1.18: Lecturer can delete assignments or quizzes which are multiple choices question, true/false question, and short answers**.



**3.2.19 Sequence Diagram (SD 1.19): Lecturer marks the point for student’s short answer.**

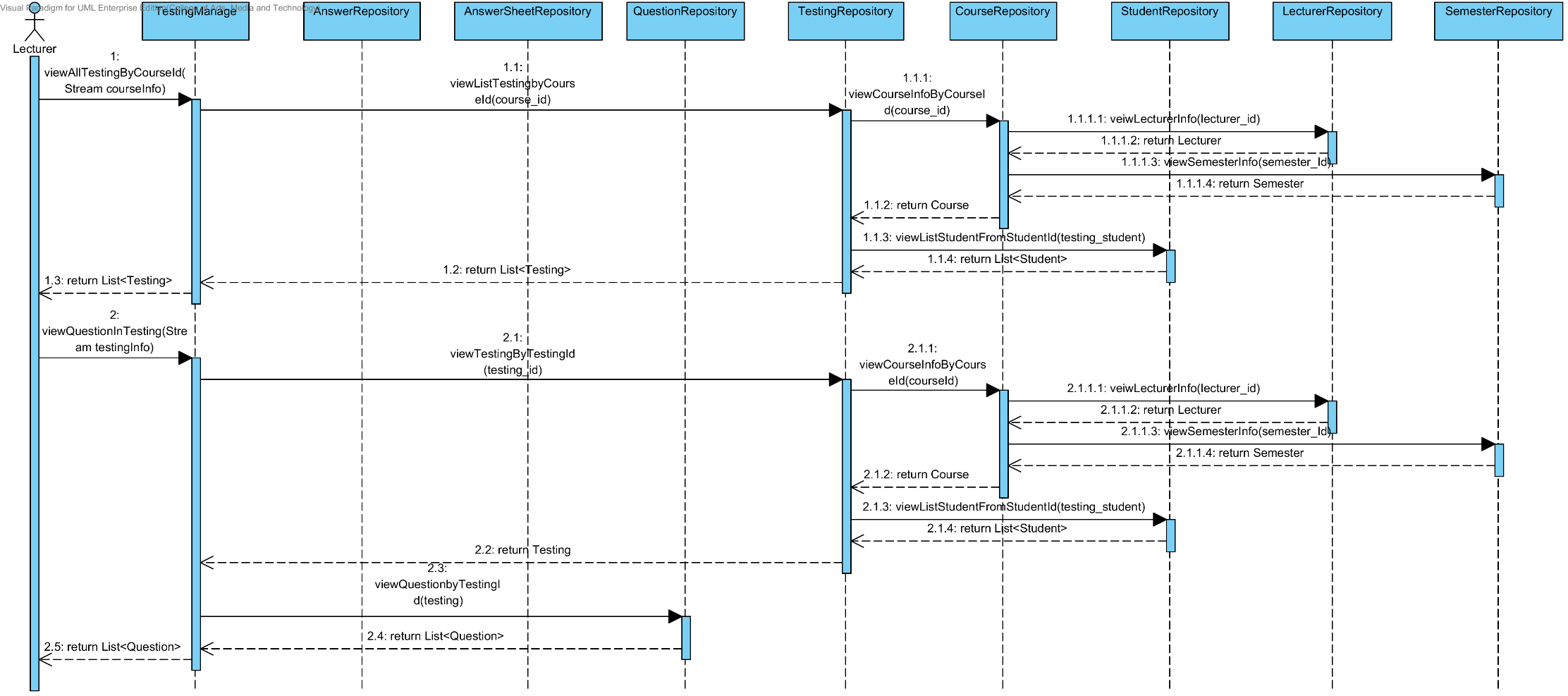
**: This sequence diagram responses to URS 1.19: Lecturer marks the point for student’s short answer**.

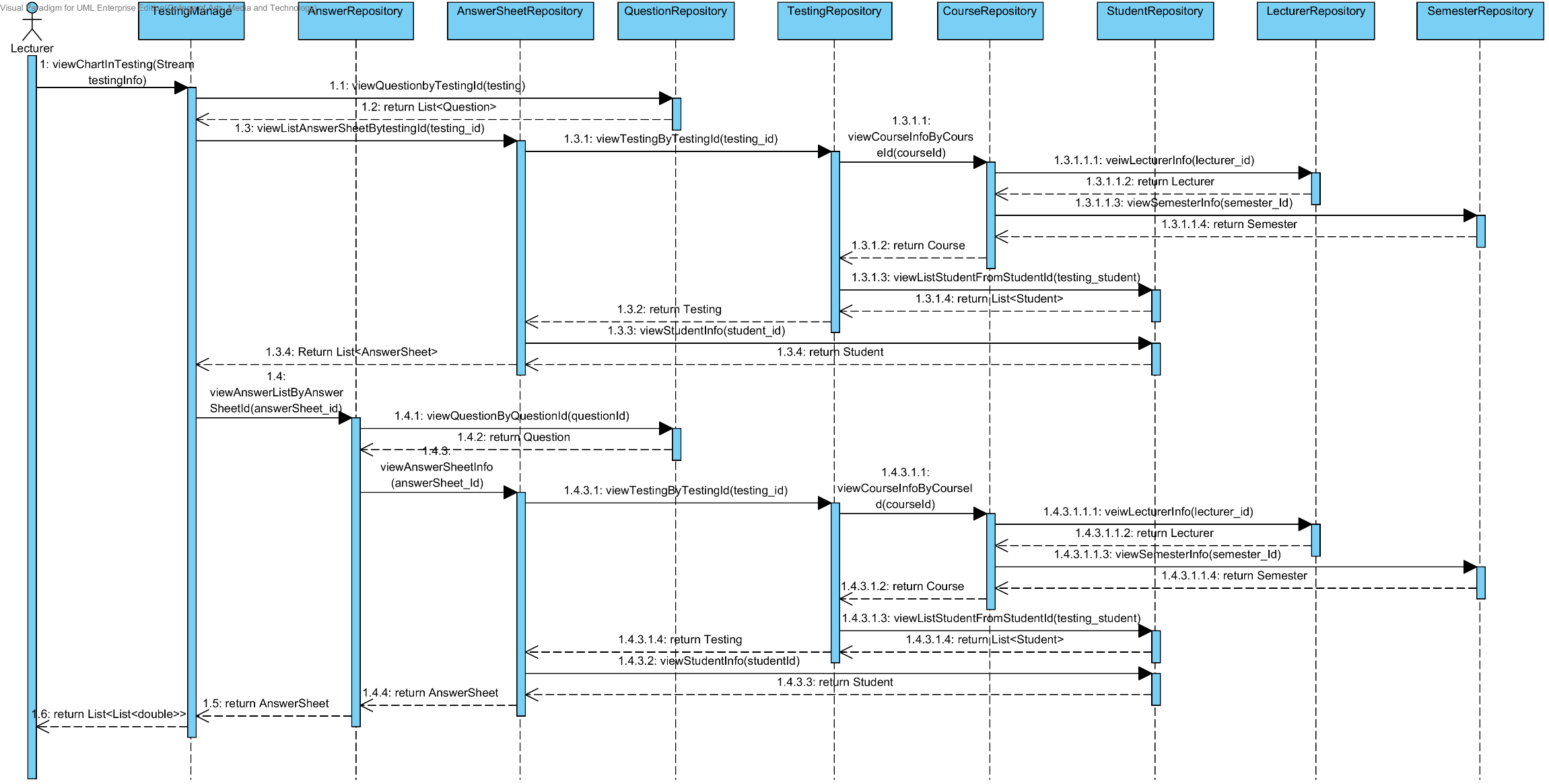




**3.2.20 Sequence Diagram (SD 1.20): Lecturer reviews overall student’s answer in each choice of assignment or quiz in chart format.**

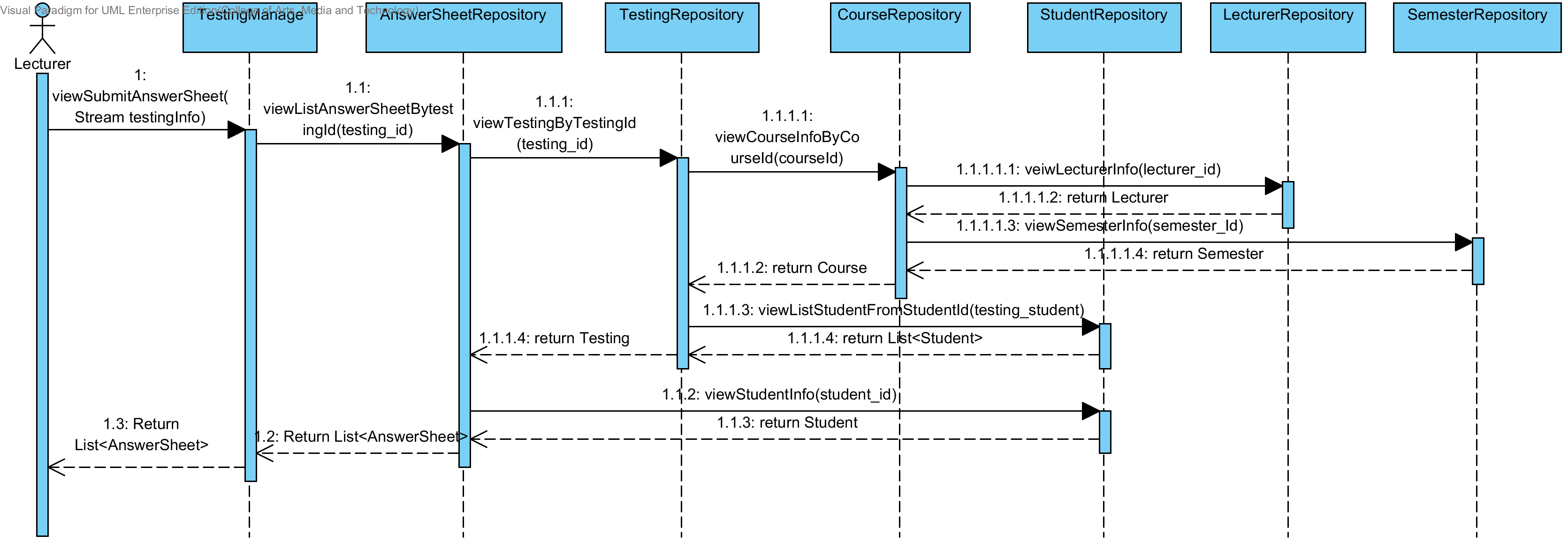
**: This sequence diagram responses to URS 1.20: Lecturer can review overall student’s answer in each choice of assignment or quiz in chart format**.





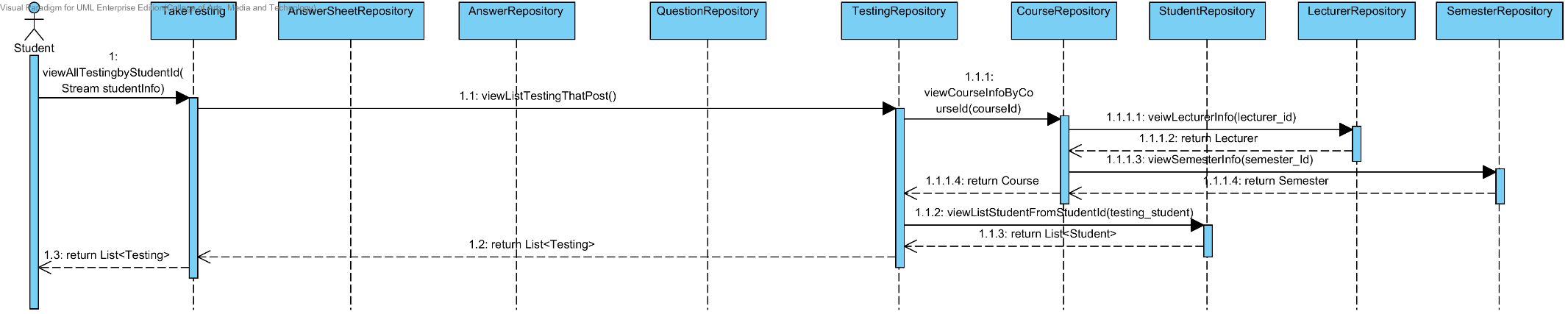
**3.2.21 Sequence Diagram (SD 1.21): Lecturer checks whether the assignments are submitted in time.**

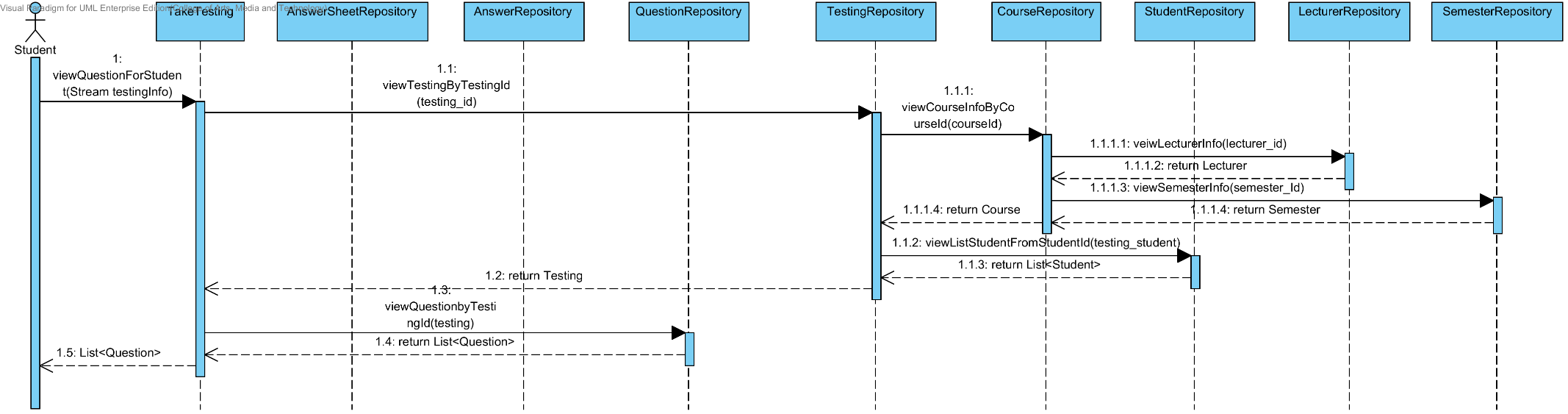
**: This sequence diagram responses to URS 1.21: Lecturer can check whether the assignments are submitted in time.**

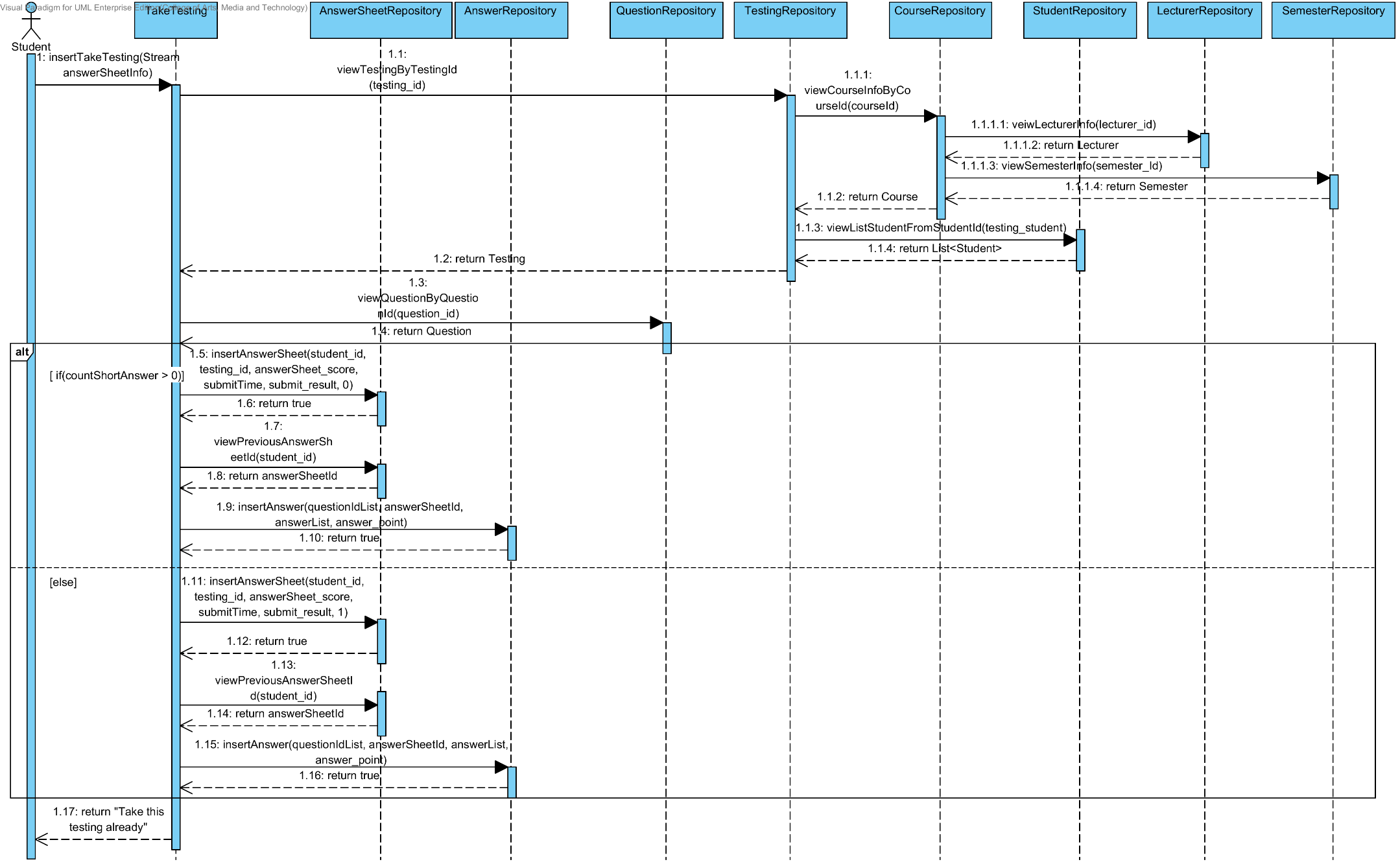


**3.2.22 Sequence Diagram (SD 1.22): Student takes assignments or quizzes which are multiple choices, true/false questions, short answers.**

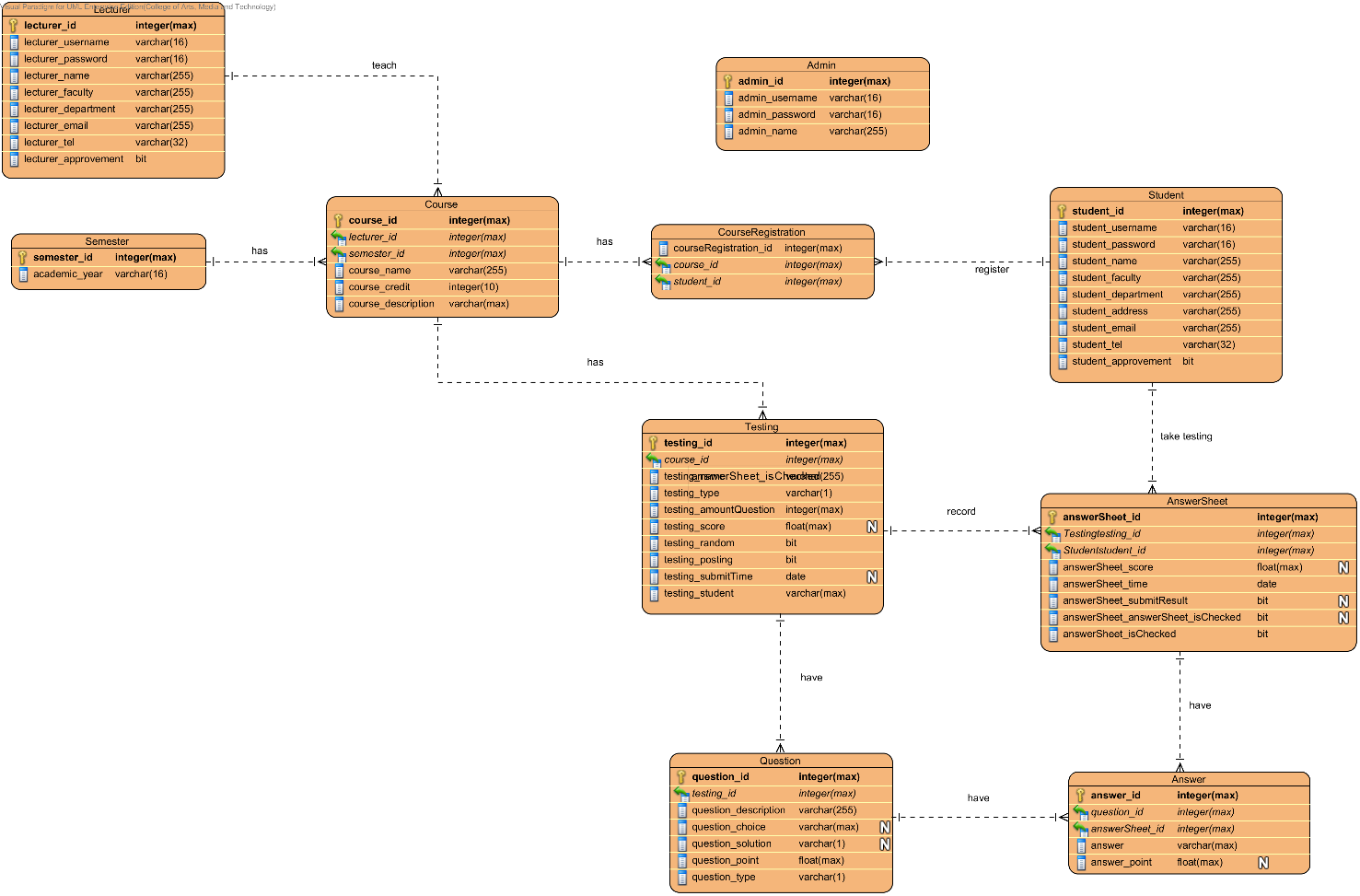
**: This sequence diagram responses to URS 1.22: Student can take assignments or quizzes which are multiple choices, true/false questions, short answers.**







## **3.3 Database design**



**3.1.1 Lecturer table**

**Description:** Lecturer table stores all the lecturer information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| lecturer\_id | Unique Identifier of Lecturer | integer | max | PK | - | Not Null |
| lecturer\_username | Username of Lecturer | varchar | 16 | - | - | Not Null |
| lecturer\_password | Password of Lecturer | varchar | 16 | - | - | Not Null |
| lecturer\_name | Name of Lecturer | varchar | 255 | - | - | Not Null |
| lecturer\_faculty | Faculty of Lecturer | varchar | 255 | - | - | Not Null |
| lecturer\_department | Description of Lecturer | varchar | 255 | - | - | Not Null |
| lecturer\_email | Email of Lecturer | varchar | 255 | - | - | Not Null |
| lecturer\_tel | Tel of Lecturer | varchar | 32 | - | - | Not Null |
| lecturer\_approvement | Approval status of lecturer account | bit | 1 | - | - | Not Null |

**3.1.2 Student table**

**Description:** Student table stores all the student information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| student\_id | Unique Identifier of Student | integer | max | PK | - | Not Null |
| student\_username | Username of Student | varchar | 16 | - | - | Not Null |
| student\_password | Password of Student | varchar | 16 | - | - | Not Null |
| student\_name | Name of Student | varchar | 255 | - | - | Not Null |
| student\_faculty | Faculty of Student | varchar | 255 | - | - | Not Null |
| student\_department | Description of Student | varchar | 255 | - | - | Not Null |
| student\_address | Address of Student | varchar | 255 | - | - | Not Null |
| student\_email | Email of Student | varchar | 255 | - | - | Not Null |
| student\_tel | Tel of Student | varchar | 32 | - | - | Not Null |
| student\_approvement | Approval status of Student account | bit | 1 | - | - | Not Null |

**3.1.3 Admin table**

**Description:** Admin table stores all the admin information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| admin\_id | Unique Identifier of Admin | integer | max | PK | - | Not Null |
| admin\_username | Username of Admin | varchar | 16 | - | - | Not Null |
| admin\_password | Password of Admin | varchar | 16 | - | - | Not Null |
| admin\_name | Name of Admin | varchar | 255 | - | - | Not Null |

**3.1.4 Semester table**

**Description:** Semester table stores all the semester information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| semester\_id | Unique Identifier of Semester | integer | max | PK | - | Not Null |
| academic\_year | Academic year information | varchar | 16 | - | - | Not Null |

**3.1.5 Course table**

**Description:** Course table stores all the course information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| course\_id | Unique Identifier of Course | integer | max | PK | - | Not Null |
| lecturer\_id | Unique Identifier of Lecturer | integer | max | - | FK | Not Null |
| semester\_id | Unique Identifier of Semester | integer | max |  | FK | Not Null |
| course\_name | Name of Course | varchar | 255 | - | - | Not Null |
| course\_credit | Credit of Course | int | max | - | - | Not Null |
| course\_description | Description of Course | varchar | max | - | - | Not Null |

**3.1.6 Course Registration table**

**Description:** Course Registration table stores all the student’s registration information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| courseRegistration\_id | Unique Identifier of Course registration | integer | max | PK | - | Not Null |
| course\_id | Unique Identifier of Course | integer | max | - | FK | Not Null |
| student\_id | Unique Identifier of Student | integer | max | - | FK | Not Null |

**3.1.7 Testing table**

**Description:** Testing table stores all the testing information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| testing\_id | Unique Identifier of Testing | integer | max | PK | - | Not Null |
| course\_id | Unique Identifier of Course | integer | max | - | FK | Not Null |
| testing\_name | Name of Testing | varchar | 255 | - | - | Not Null |
| testing\_type | Type of Testing | varchar | 1 | - | - | Not Null |
| testing\_amountQuestion | Amount of questions in Testing | integer | max | - | - | Not Null |
| testing\_score | Score of Testing | float | max | - | - | Not Null |
| testing\_random | Random status of Testing | bit | 1 | - | - | Not Null |
| testing\_posting | Posting status of Testing | bit | 1 | - | - | Not Null |
| testing\_submitTime | Submit Time of Testing | datetime2 | max | - | - | Null |
| testing\_student | Student of Testing | varchar | max | - | - | Not Null |

**3.1.8 Question table**

**Description:** Question table stores all the question information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| question\_id | Unique Identifier of Question | integer | max | PK | - | Not Null |
| testing\_id | Unique Identifier of Testing | integer | max | - | FK | Not Null |
| question\_description | Description of Question | varchar | max | - | - | Not Null |
| question\_choice | Choices of Question | varchar | max | - | - | Null |
| question\_solution | Solution of Question | varchar | 1 | - | - | Null |
| question\_point | Point of Question | float | max | - | - | Not Null |
| question\_type | Type of Question | varchar | 1 | - | - | Not Null |

**3.1.9 AnswerSheet table**

**Description:** AnswerSheet table stores all the answer sheet information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| answerSheet\_id | Unique Identifier of Answer Sheet | integer | max | PK | - | Not Null |
| student\_id | Unique Identifier of Student | integer | max | - | FK | Not Null |
| testing\_id | Unique Identifier of Testing | integer | max | - | FK | Not Null |
| answerSheet\_score | Score of Answer sheet | varchar | max | - | - | Not Null |
| answerSheet\_time | Time of Answer sheet | varchar | max | - | - | Not Null |
| answerSheet\_submitResult | Submit result of Answer sheet | varchar | 1 | - | - | Null |
| answerSheet\_isChecked | Status of marking point in answer sheet | varchar | 1 | - | - | Not Null |

**3.1.10 Answer table**

**Description:** AnswerSheet table stores all the answer sheet information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| answer\_id | Unique Identifier of Answer | integer | max | PK | - | Not Null |
| question\_id | Unique Identifier of Question | integer | max | - | FK | Not Null |
| answerSheet\_id | Unique Identifier of Answer Sheet | integer | max | - | FK | Not Null |
| answer | Answer detail | varchar | max | - | - | Not Null |
| answer\_point | Point of Answer | float | max | - | - | Null |