**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

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

**Mr. Prompong Sugunnasil**

**Document History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Document Name** | **Version** | **Status** | **Date** | **Viewable** | **Reviewer** | **Responsible** |
| **Documents** | | | | | | |
| Smart course management system in the cloud \_ Software design document\_v0.1.docx | Add  - Chapter One  - Purpose  - Software Scope  - Technology review | Draft | 30 May 2014 | CS, TP, PS | CS,TP | TP |
| Smart course management system in the cloud \_ Software design document\_v0.2.docx | Add  - Chapter Two  - Chapter Three  - Class Diagram  - Database design | Draft | 20 June 2014 | CS, TP, PS | CS,TP | TP |
| Smart course management system in the cloud \_ Software design document\_v0.3.docx | Add  - Sequence Diagram  Fix  - Class Diagram | Draft | 15 July 2014 | CS, TP, PS | CS,TP | TP |
| Smart course management system in the cloud \_ Software design document\_v0.4.docx | Fix  - Sequence Diagram | Release | 31 July 2014 | CS, TP, PS | CS,TP | TP |
| Smart course management system in the cloud \_ Software design document\_v1.0.docx | Fix  - Sequence Diagram | Release | 27 August 2014 | CS, TP, PS | CS,TP | TP |
| Smart course management system in the cloud \_ Software design document\_v1.1.docx | Create new design in progress2 | Release | 22 Oct 2014 | CS, TP, PS | CS,TP | TP |

**\*CS = Chaichan Suttee**

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

**\*PS = Dr. Prompong Sugunnasin**

Contents

[**Chapter One | Introduction** 5](#_Toc401761686)

[**1.1** **Purpose** 5](#_Toc401761687)

[**1.2 Software Scope** 5](#_Toc401761688)

[**1.3 Technology Review** 6](#_Toc401761689)

[**Chapter Two | System Architecture** 9](#_Toc401761690)

[**Chapter Three | Detail Design** 10](#_Toc401761691)

[**3.1 Class Diagram** 10](#_Toc401761692)

[**3.2 Sequence Diagram** 16](#_Toc401761693)

[**3.3 Database design** 18](#_Toc401761694)

# **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.

1.3.4 Windows Communication Foundation (WCF)

Technology description

Windows Communication Foundation (WCF) is a framework for building service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another. A service endpoint can be part of a continuously available service hosted by IIS, or it can be a service hosted in an application. An endpoint can be a client of a service that requests data from a service endpoint. The messages can be as simple as a single character or word sent as XML, or as complex as a stream of binary data.

The selection of this technology

-Windows Communication Foundation is service of c# .net

1.3.5 Google Drive APIs

Technology description

Google Drive APIs is the drive platform gives a group of APIs along with client libraries, which help to create drive for managing documents. Google Drive is an application of Google Company.

The selection of this technology

- This technology is a freeware.

- Google Drive can work together with Gmail by using Google account.

1.3.5 Google Spreadsheets APIs

Technology description

Google Spreadsheets APIs is the gives group of APIs along with client libraries, which help to create spreadsheets for handling everything from simple task lists to data analysis with charts, filters and pivot tables.

The selection of this technology

- This technology is a freeware.

1.3.5 Gmail APIs

Technology description

Gmail APIs is the gives group of APIs along with client libraries, which help to send the email by using Gmail application.

The selection of this technology

- This technology is a freeware.

- Gmail can work together with Google Drive by using Google account.

# **Chapter Two | System Architecture**

Web service

DB

JSON

Mobile application

Web application

IIS

IIS



**iOS 7**

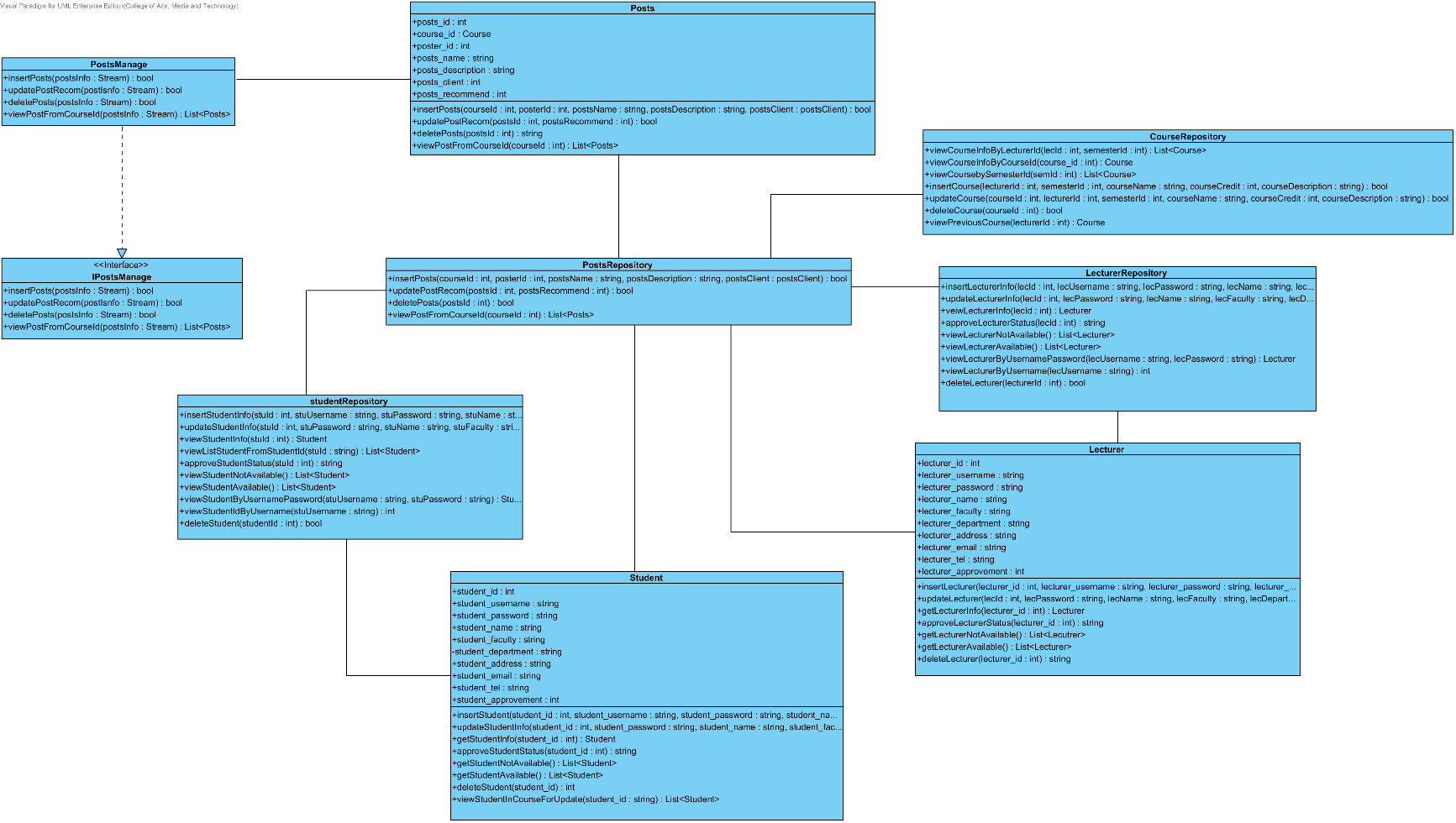
**Firefox web browser**

***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.

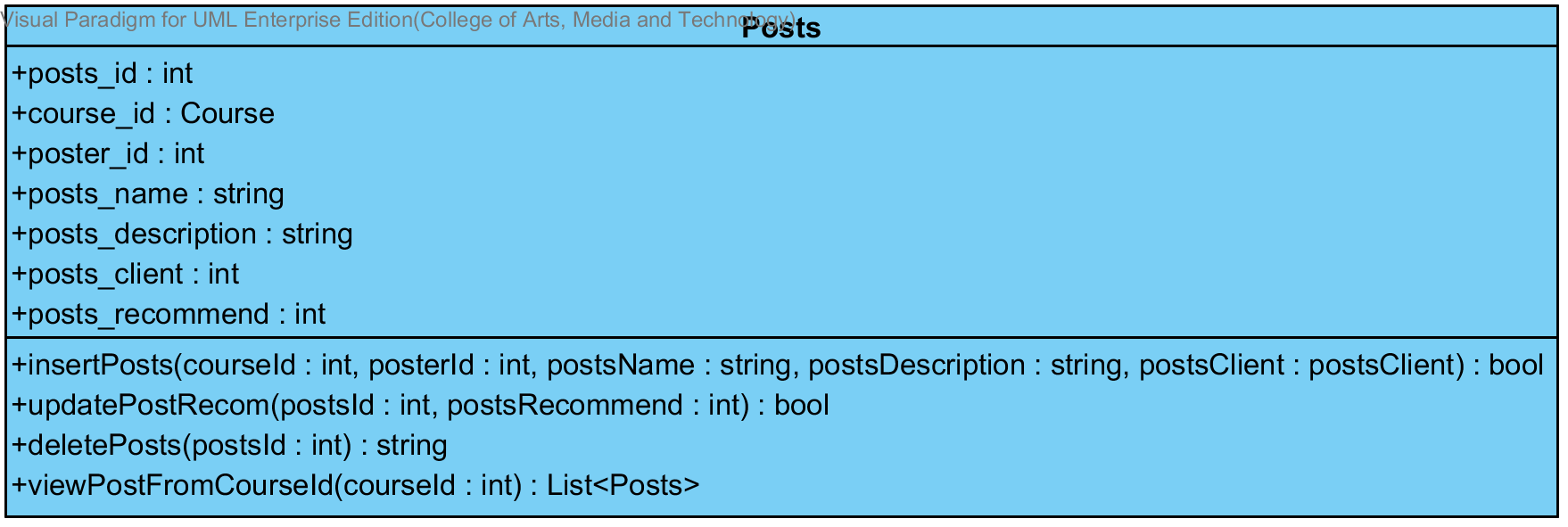
# **Chapter Three | Detail Design**

## **3.1 Class Diagram**

****

**Class Description**

**3.1.33 Class-33. Posts**

****

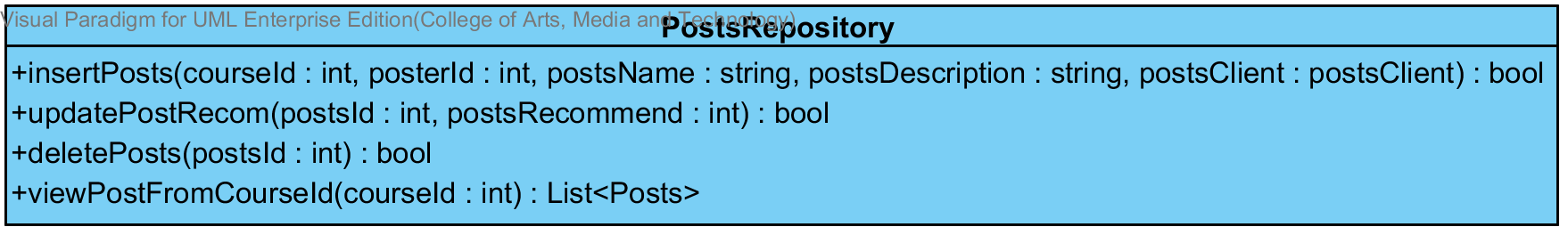
**Attribute**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Description | Type |
| 33.1 | posts\_id | Unique Identifier of Posts | int |
| 33.2 | course\_id | Object for identifying Course | Course |
| 33.3 | poster\_id | Unique Identifier of poster | int |
| 33.4 | posts\_name | Name of posts | string |
| 33.5 | posts\_description | Description of posts | string |
| 33.6 | posts\_client | Type of client that want to communicate. | string |
| 33.7 | posts\_recommend | Recommend status of posts (0 is random, 1 is not random) | int |

**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 33.8 | insertPosts | This method is used for inserting posts in database. | courseId: int, posterId: int, postsName: string, postsDescription: string, postsClient: postsClient | bool |
| 33.9 | updatePostRecom | This method is used for updating posts for recommending in database. | postsId: int, postsRecommend: int) | bool |
| 33.10 | deletePosts | This method is used for deleting posts in database. | postsId: int | string |
| 33.11 | viewPostFromCourseId | This method is used for viewing a list of posts in course. | courseId: int | List<Posts> |

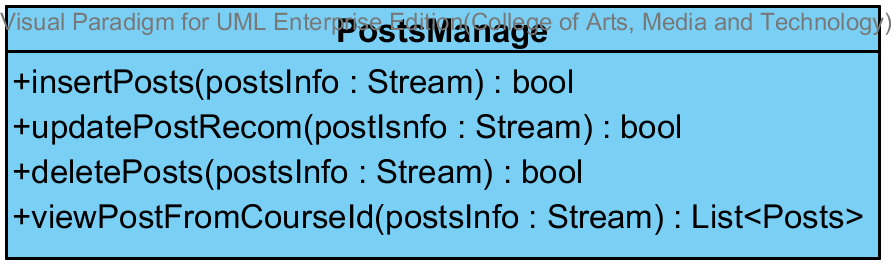
**3.1.34 Class-34. PostsRepository**

****

**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 34.1 | insertPosts | This method is used for inserting posts in database. | courseId: int, posterId: int, postsName: string, postsDescription: string, postsClient: postsClient | bool |
| 34.2 | updatePostRecom | This method is used for updating posts for recommending in database. | postsId: int, postsRecommend: int) | bool |
| 34.3 | deletePosts | This method is used for deleting posts in database. | postsId: int | bool |
| 34.4 | viewPostFromCourseId | This method is used for viewing a list of posts in course. | courseId: int | List<Posts> |

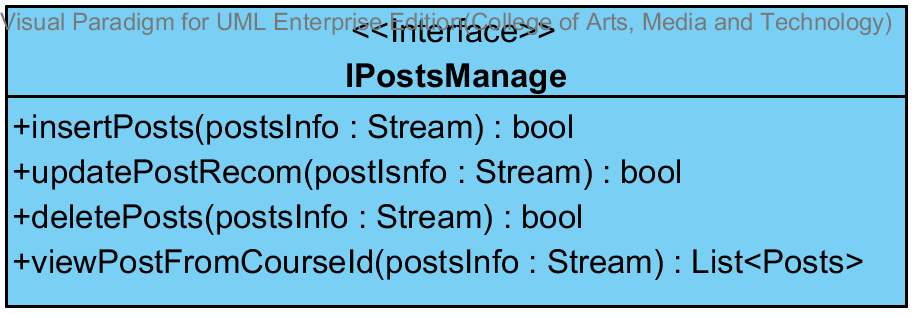
**3.1.35 Class-35.** **PostsManage**

****

**Method**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Name | Description | Parameters | Return |
| 35.1 | insertPosts | This method is used for inserting posts in database. | postInfo: Stream | bool |
| 35.2 | updatePostRecom | This method is used for updating posts for recommending in database. | postInfo: Stream | bool |
| 35.3 | deletePosts | This method is used for deleting posts in database. | postInfo: Stream | string |
| 35.4 | viewPostFromCourseId | This method is used for viewing a list of posts in course. | postInfo: Stream | List<Posts> |

**3.1.36 Class-36.** **IPostsManage**



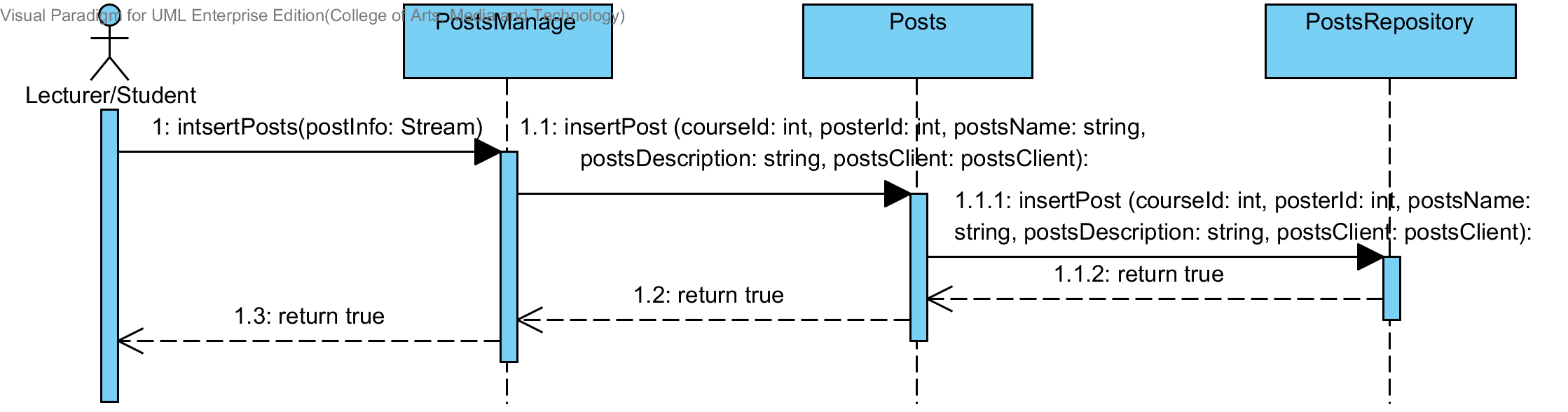
**Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Parameters | Return |
| 36.1 | insertPosts | postInfo: Stream | bool |
| 36.2 | updatePostRecom | postInfo: Stream | bool |
| 36.3 | deletePosts | postInfo: Stream | string |
| 36.4 | viewPostFromCourseId | postInfo: Stream | List<Posts> |

## **3.2 Sequence Diagram**

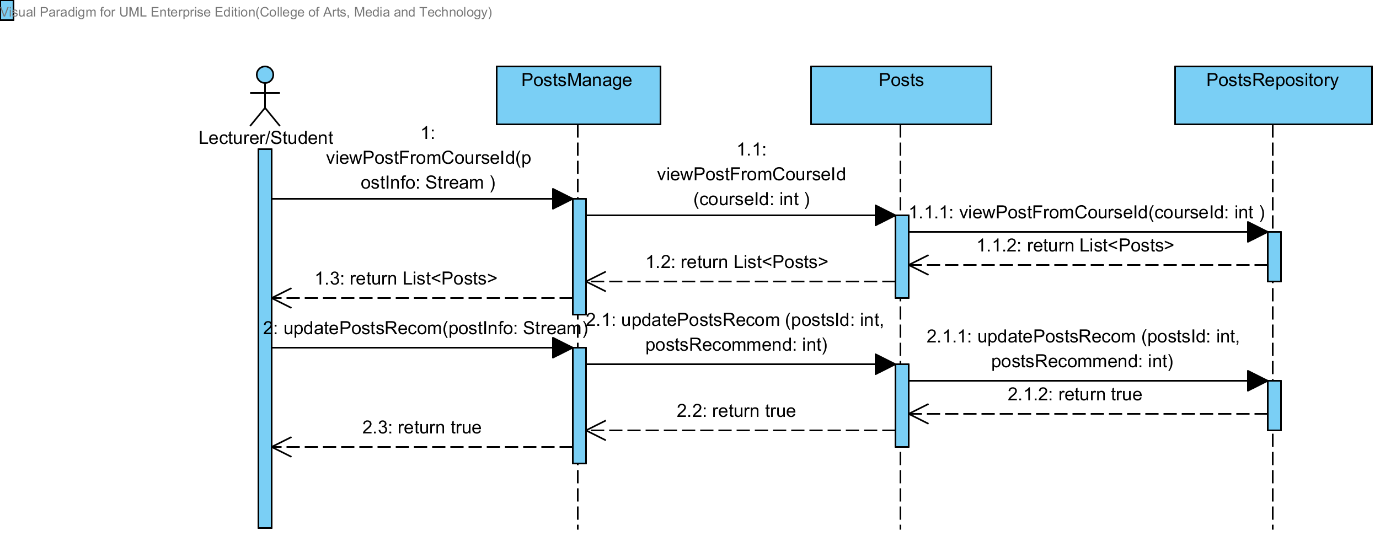
**3.2.1 Sequence Diagram (SD 23.1): Lecturer/Student add a posts.**

**: This sequence diagram responses to URS 28.1**



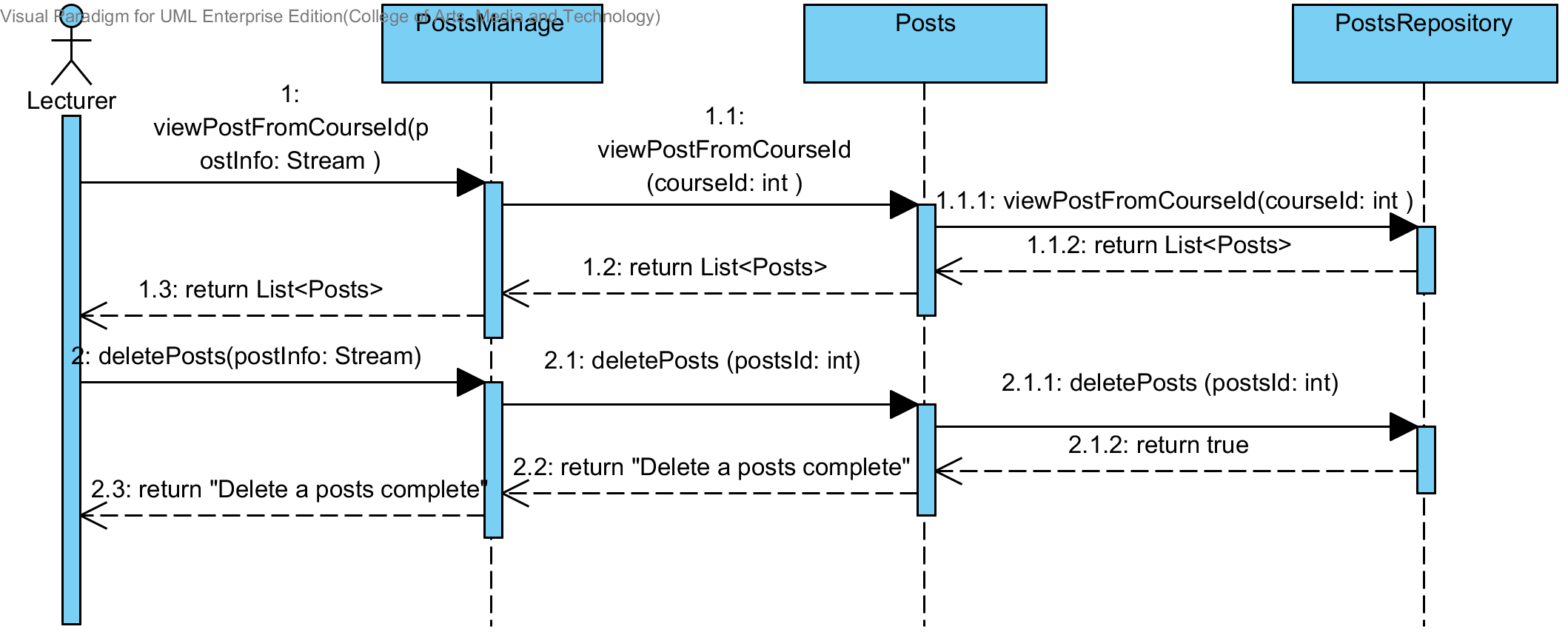
**3.2.2 Sequence Diagram (SD 24.1): Lecturer/Student recommends a posts**

**: This sequence diagram responses to URS 29.1**

****

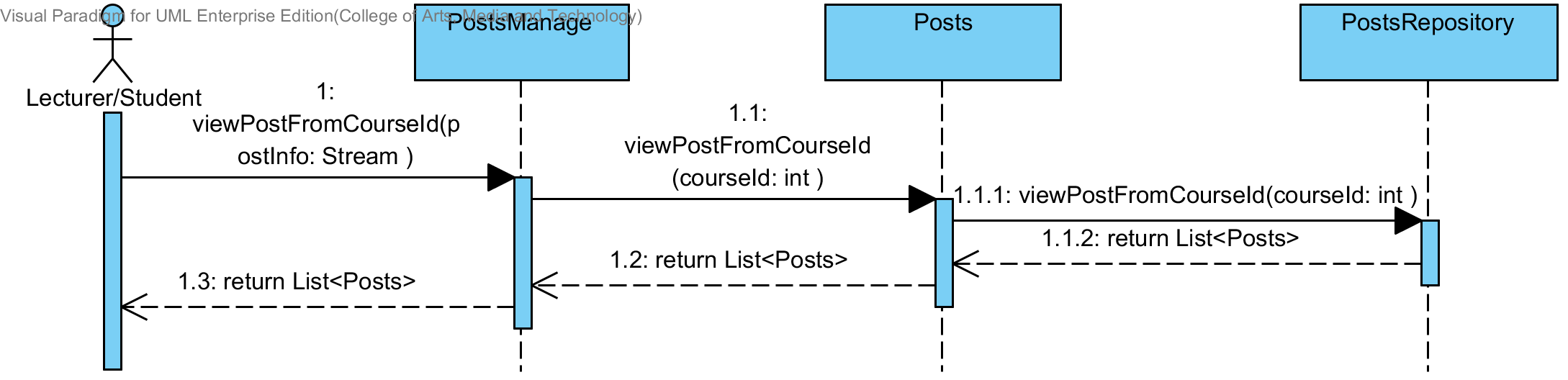
**3.2.3 Sequence Diagram (SD 25.1): Lecturer/Student deletes a posts.**

**: This sequence diagram responses to URS 30.1**

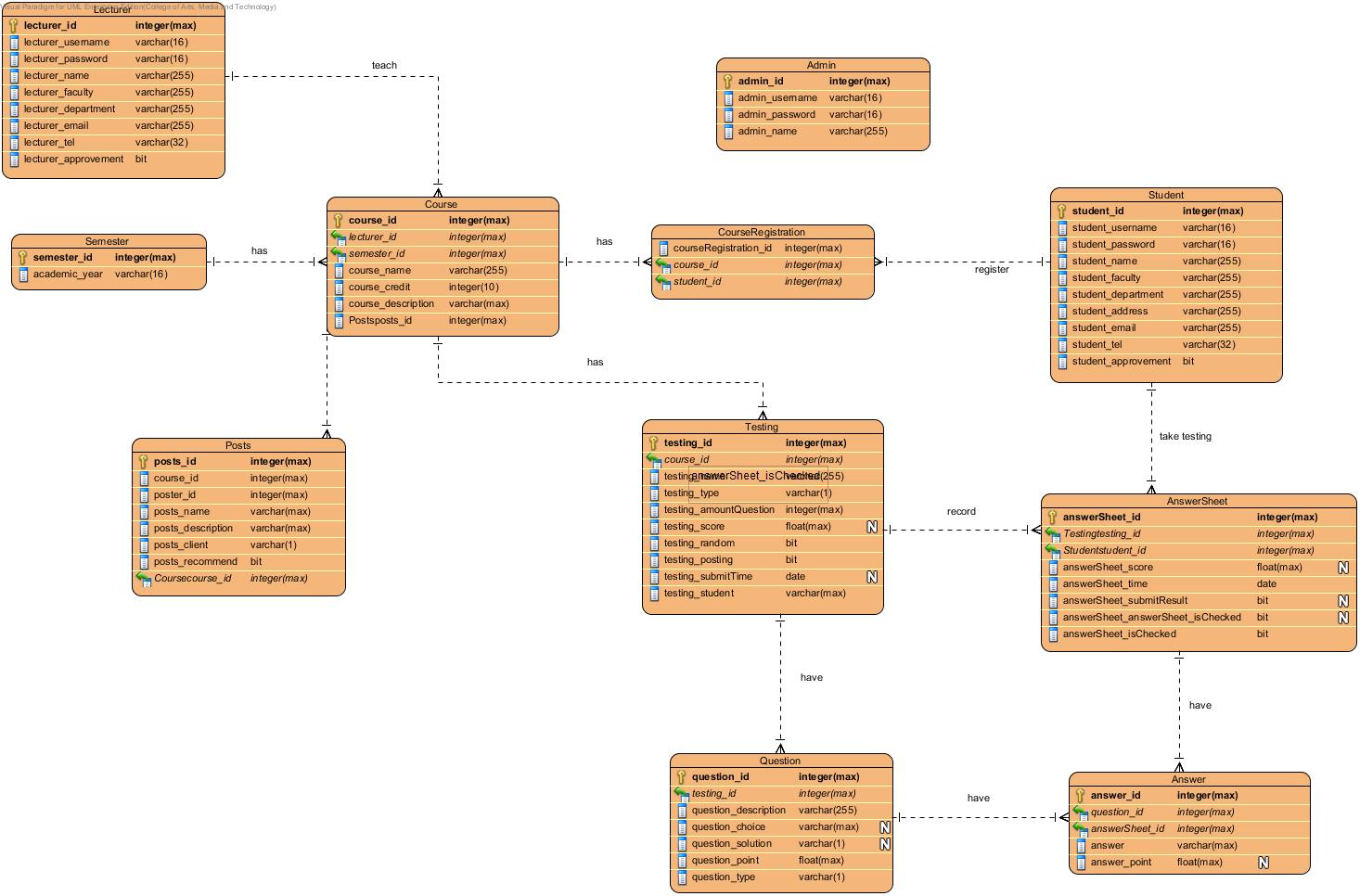


**3.2.4 Sequence Diagram (SD 26.1): Lecturer/Student views a posts.**

**: This sequence diagram responses to URS 31.1**



## **3.3 Database design**



**3.3.1 Table Lecturer**

**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 |

**3.3.2 Table Student**

**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 |

**3.3.3 Table Admin**

**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 |

**3.3.4 Table Semester**

**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 |

**3.3.5 Table Course**

**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 |

**3.3.6 Table Course Registration**

**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 |

**3.3.7 Table Testing**

**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 |

**3.3.8 Table Question**

**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 |

**3.3.9 Table AnswerSheet**

**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 |

**3.3.10 Table Answer**

**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 |

**3.3.11 Posts**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| posts\_id | Unique Identifier of posts | integer | max | PK | - | Not Null |
| course\_id | Unique Identifier of course | integer | max | - | FK | Not Null |
| poster\_id | Id of poster | integer | max | - | - | Not Null |
| posts\_name | Name of posts | varchar | max | - | - | Not Null |
| posts\_description | Description of posts | varchar | max | - | - | Not Null |
| posts\_client | Type of client that want to posts. | varchar | 1 | - | - | Not Null |
| posts\_recommend | Recommend status of posts | int | max | - | - | Not Null |