`



**LEARNING ONLINE**

**Software Design Specification**

– Hanoi, Feb 2023 –

**Table of Contents**

[I. Overview 3](#_heading=h.3dy6vkm)

[1. Code Packages/Namespaces 3](#_heading=h.1t3h5sf)

[2. Coding Convent 3](#_heading=h.2et92p0)

[II. Code Designs 4](#_heading=h.4d34og8)

[1. Register and Login 4](#_heading=h.2s8eyo1)

[a. Class Diagram 4](#_heading=h.26in1rg)

[b. Class Specifications 5](#_heading=h.lnxbz9)

[c. Sequence Diagram(s) 6](#_heading=h.35nkun2)

[d. Database queries 6](#_heading=h.1ksv4uv)

[2. User profile 8](#_heading=h.in8dcso3j31)

[a. Class Diagram 8](#_heading=h.zdvi7r84qz1)

[b. Class Specifications 8](#_heading=h.wyqcihvxfp5p)

[Database queries 8](#_heading=h.9z59r2hxkp3h)

[c. Sequence Diagram(s) 9](#_heading=h.tuu166blo4zb)

[d. Database queries 9](#_heading=h.kn688x5f6n3s)

[3. Bloglist 9](#_heading=h.2u6wntf)

[a. Class Diagram 9](#_heading=h.qsjttbtjax9r)

[b. Class Specifications 9](#_heading=h.xwtx5344egqf)

[c. Sequence Diagram(s) 9](#_heading=h.fngwsg9ug5kd)

[d. Database queries 9](#_heading=h.o4um1o6ejlft)

[III. Database Design 10](#_heading=h.19c6y18)

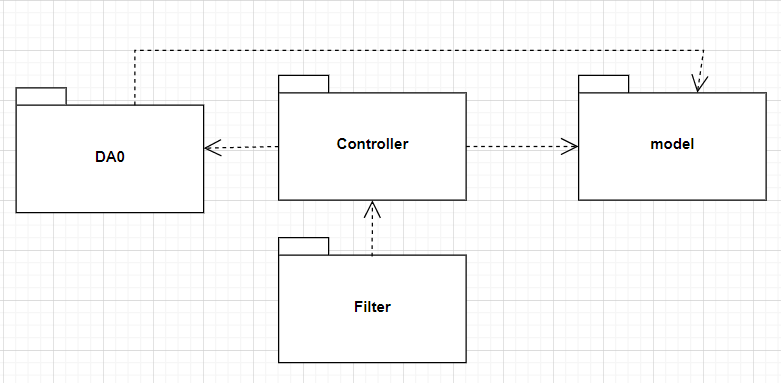
[1. Database Schema 10](#_heading=h.3tbugp1)

[2. Entity Relationship Diagram 10](#_heading=h.gedk4kmpopfl)

[3. Table Description 11](#_heading=h.kh3kznhi200o)

# I. Overview

## 1. Code Packages/Namespaces



***Package descriptions & package class naming conventions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *Model* | This component includes entities used to receive data retrieved from the database and classes used to receive data |
| *02* | *Controller* | *Contains servlets that receive requests from the client to the server and push information from the server to the client.*  *Naming convention: class names must have “Controller” at the end* |
| *03* | *Filter* | *Contains filters that check user requests and redirect pages* |
| *04* | *Dao* | *This package contains all the classes that can interact with the database.*  *Naming convention: class name must have “Dao” at the end* |

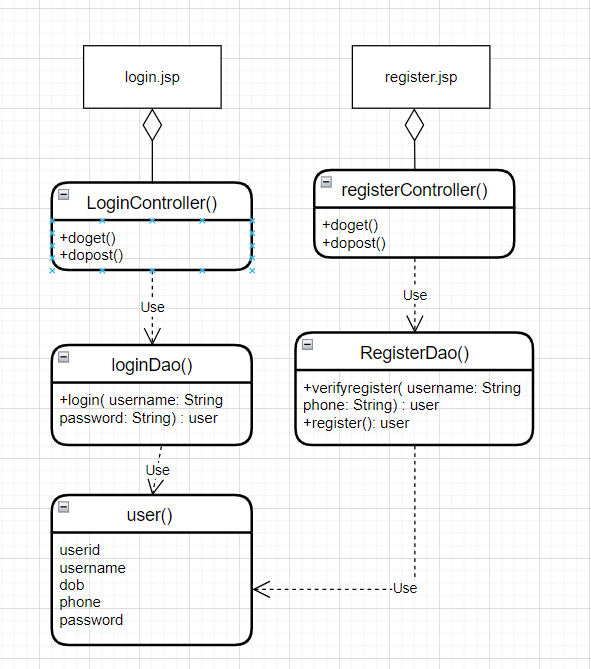
## 

## 2. Coding Convent

# II. Code Designs

## 1. Register and Login

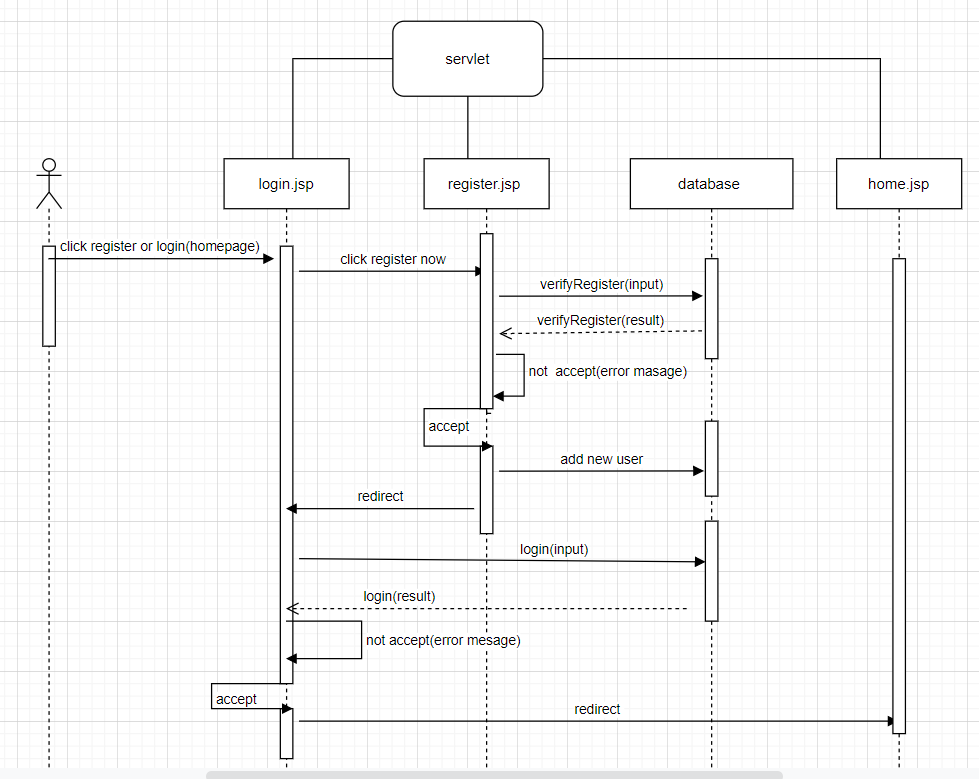
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Login()* | *<* *method of using account and password to use website functions>* |
| *02* | *Register()* | *<The method is used to create an account, a password, to grant access and use the website's functions>* |
| *03* | *verifyregister()* | *<* *Check the existence of the account to avoid duplication>* |
| *04* | *LoginController()* | *<get user data and display message on login page>* |
| *05* | *RegisterController()* | *<get user data and display message on register page>* |
| *6* | *doget()* | *<push data from server to client>* |
| *7* | *dopost()* | *<get data from client edit and return>* |

### c. Sequence Diagram(s)



### d. Database queries

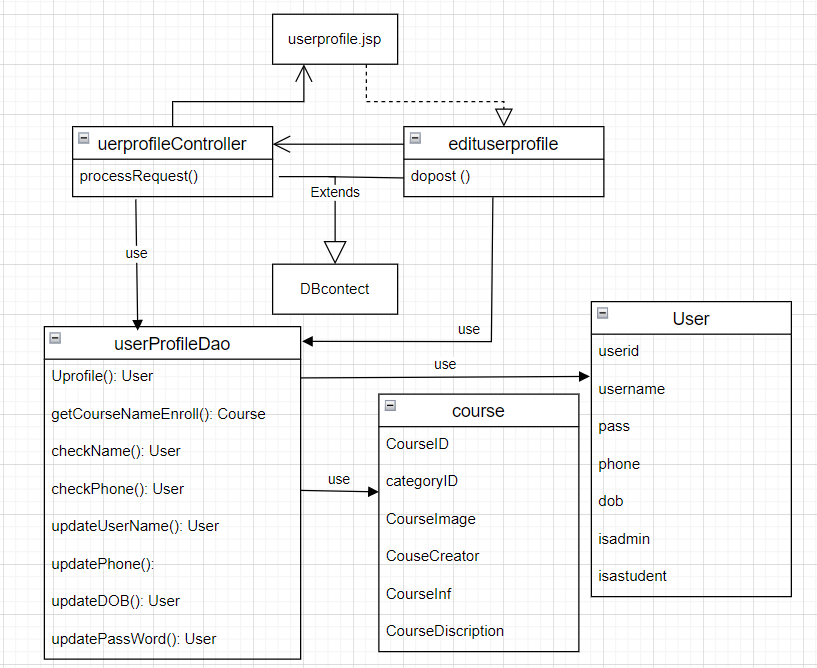
| **No** | **Method** | **Database queries** |
| --- | --- | --- |
| *01* | *Login()* | *<*SELECT \* FROM [dbo].[User] where UserName = ? and PassWord = ?*>*  *check user exist* |
| *02* | *verifyregister()* | *<*SELECT \* FROM [dbo].[User] where [UserName] = ? or Gmail= ?*>*  *Check if the name or gmail is not the same because each account has only one gmail and one username* |
| *03* | *Register()* | *<INSERT INTO [User] ( [UserName], [PassWord], [FullName], [Image],[Email],[DOB],[Phone],[Address],[FacebookID],[GmailID], [RoleID], [Status]) VALUES ( ?, ?, ?, ?, ?, ?,?,?,?, 3, 1))>*  *save account information to database* |
|  |  |  |

### 

## 

## 2. User profile

### a. Class Diagram



### b. Class Specifications

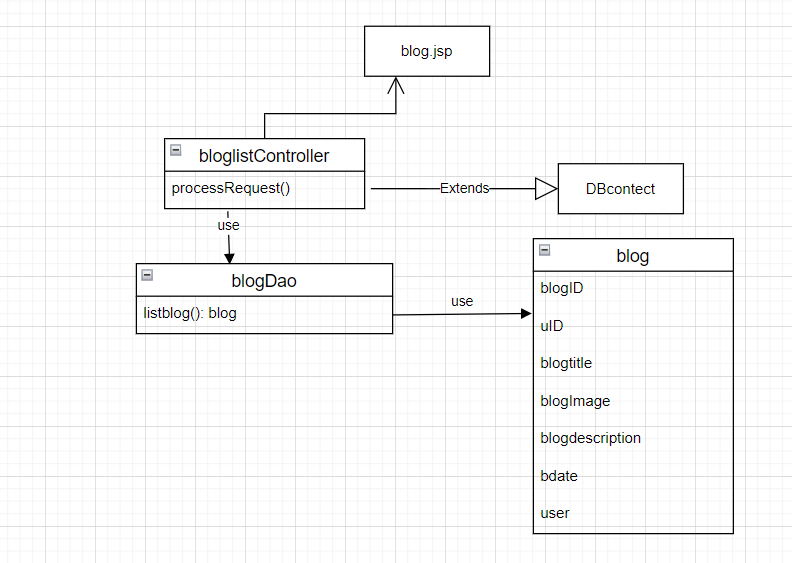
| **No** | **Method** | **Description** | Database queries |
| --- | --- | --- | --- |
| *01* | *Uprofile()* | *get log user* | *SELECT \* FROM [dbo].[User]\ where UserID = ?* |

### c. Sequence Diagram(s)

### d. Database queries

## 3. Bloglist

### a. Class Diagram



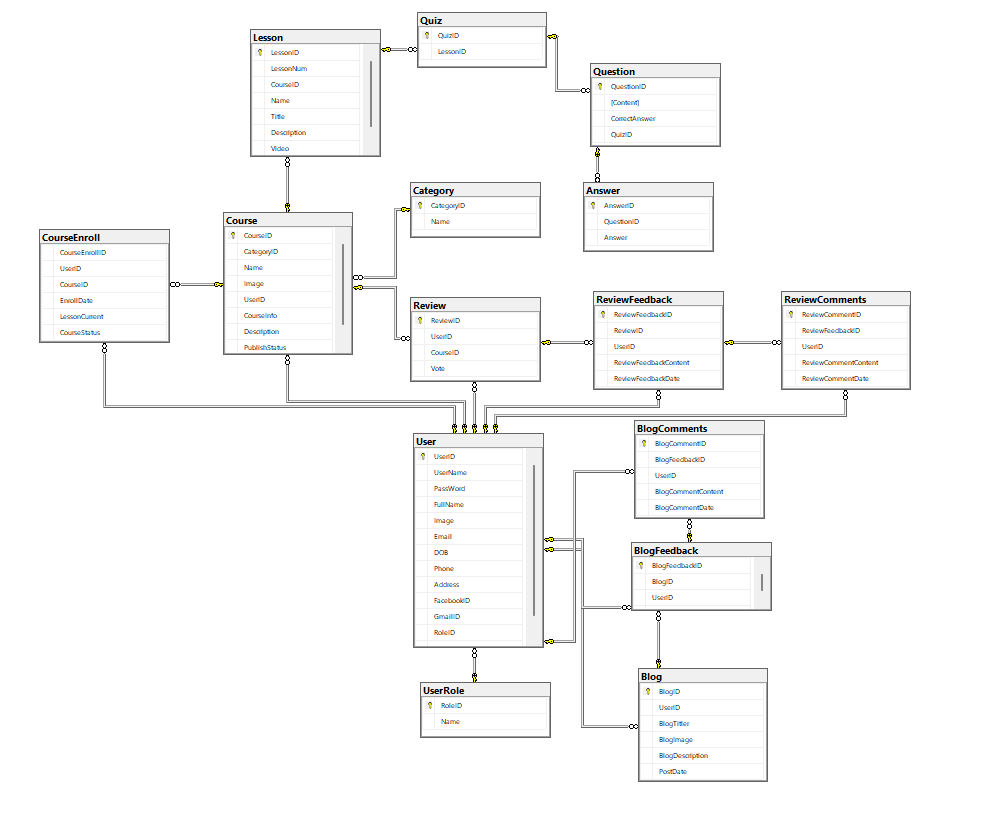
### b. Class Specifications

### c. Sequence Diagram(s)

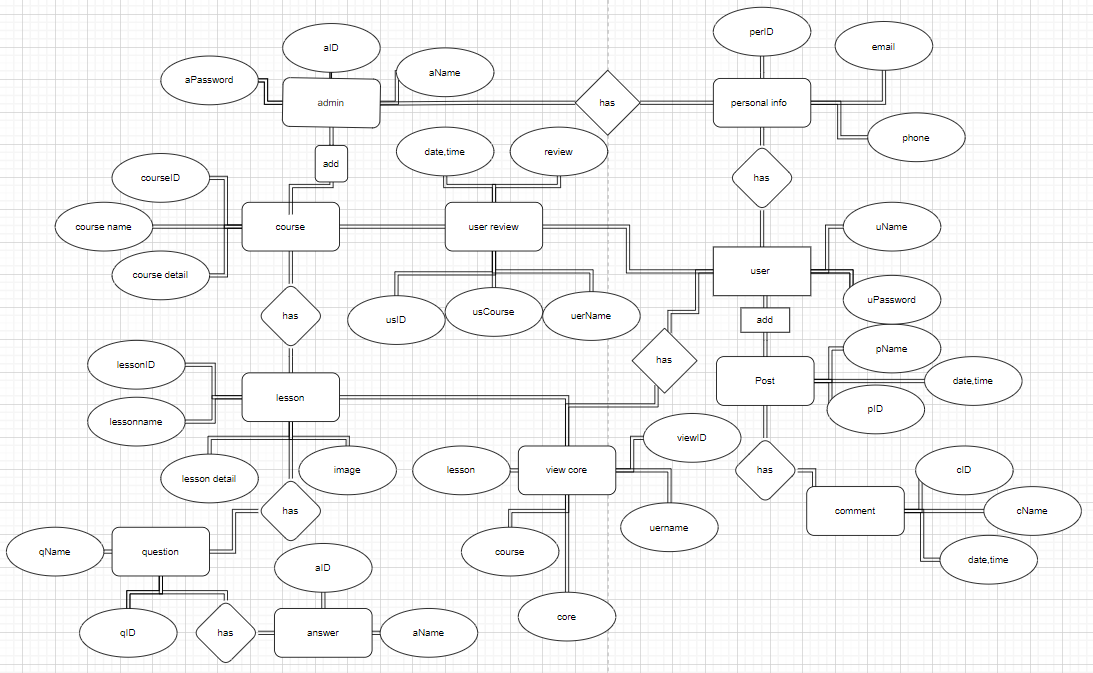
### d. Database queries

# III. Database Design

## 1. Database Schema



## 2. Entity Relationship Diagram

**

## 3. Table Description

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *User* | *Primary Key: User\_ID* |
| *02* | *Student* | *Primary Key: User\_ID*  *Foreign Key: User\_ID* |
| *03* | *Admin* | *Primary Key: User\_ID*  *Foreign Key: User\_ID* |
| *04* | *Course* | *Primary Key: Course\_ID,TermType*  *Foreign Key: TermType* |
| *05* | *Enroll* | *Primary Key: Course\_ID,User\_ID*  *Foreign Key: Course\_ID,User\_ID* |
| *06* | *CodeTerm* | *Primary Key: Term\_type* |
| *07* | *Material* | *Primary Key: Course\_ID,Meterial\_ID*  *Foreign Key: Course\_ID* |
| *08* | *Feedback* | *Primary Key: Feedback\_ID* |
| *09* | *Course\_offering* | *Primary Key: Course\_Creator\_ID,Course\_ID*  *Foreign Key: Course\_ID* |
| *10* | *Student\_Course\_Detail* | *Primary Key: User\_ID,Course\_ID,Feedback\_ID*  *Foreign Key: User\_ID,Course\_ID,Feedback\_ID* |
| *11* | *Course\_Detail* | *Primary Key: Feedback\_ID,Course\_ID,TermType,Meterial\_ID*  *Foreign Key: Feedback\_ID,Course\_ID,TermType,Meterial\_ID* |

# IV. Database Tables

## tblUser

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | UserID | int |  |  | X | PK | ID of user |
| 2 | UserName | nvarchar | 150 |  |  |  | Name of user account |
| 3 | PassWord | nvarchar | 150 |  |  |  | Password of user account |
| 4 | FullName | nvarchar | 150 |  |  |  | Full name of user |
| 5 | Image | nvarchar | 150 |  |  |  | Image of user |
| 6 | Email | nvarchar | 150 |  |  |  | Email of user |
| 7 | DOB | date |  |  | X |  | Date of birth of user |
| 8 | Phone | nvarchar | 150 |  |  |  | Phone number of user |
| 9 | Address | nvarchar | 150 |  |  |  | Address of user |
| 10 | FacebookID | nvarchar | 150 |  |  |  | FacebookID of user |
| 11 | GmailID | nvarchar | 150 |  |  |  | GmailID of user |
| 12 | RoleID | int |  |  | X | FK | RoleID of user |
| 13 | Status | int |  |  | X |  | Status of user |

## 

## 

## 

## 

## 

## 

## tblAnswer

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | AnswerID | int |  |  | X | PK | ID of answer |
| 2 | QuestionID | int |  |  | X | FK | ID of Question |
| 3 | Answer | nvarchar | max |  | X |  | answer of answer |

## tblBlog

## 

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | BlogID | int |  |  | X | PK | ID of blog |
| 2 | UserID | int |  |  | X | FK | ID of User |
| 3 | BlogTitler | nvarchar | max |  |  |  | Titler of blog |
| 4 | BlogImage | nvarchar | max |  |  |  | image of blog |
| 5 | BlogDescription | nvarchar | max |  |  |  | Description of user |
| 6 | PostDate | datetime |  |  | X |  | date of blog |

## 

## tblCourse

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | CourseID | int |  |  | X | PK | ID of course |
| 2 | CategoryID | int |  |  | X | FK | ID of category |
| 3 | Name | nvarchar | max |  | X |  | Name of course |
| 4 | Image | nvarchar | max |  | X |  | Image of course |
| 5 | UserID | int |  |  | X | FK | ID of user |
| 6 | CourseInfo | nvarchar | max |  | X |  | Information of course |
| 7 | Description | nvarchar | max |  | X |  | Description of course |
| 8 | PublishStatus | int |  |  | X |  | Status of course |

## tblCourseEnroll

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | CourseEnrollID | int |  |  | X |  | ID of course enroll |
| 2 | UserID | int |  |  | X | FK | ID of user |
| 3 | CourseID | int |  |  | X | FK | ID of course |
| 4 | EnrollDate | date |  |  | X |  | Date of course enroll |
| 5 | LessonCurrent | int |  |  | X |  | Lesson of course |
| 6 | CourseStatus | int |  |  | X |  | Status of course |

## tblBlogComments

## 

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | BlogCommentID | int |  |  | X | PK | ID of Comment |
| 2 | BlogFeedback | int |  |  | X | FK | Feedback of Blog |
| 3 | UserID | int |  |  | X |  | ID of User |
| 4 | BlogCommentContent | nvarchar | max |  | X |  | Content of comment blog |
| 5 | BlogCommentDate | datetime |  |  | X |  | Comment date of Blog |

## tblLesson

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | LessonID | int |  |  | X | PK | ID of lesson |
| 2 | LessonNum | int |  |  |  |  | number of lesson |
| 3 | CourseID | intr |  |  | X | FK | ID of course |
| 4 | Name | nvarchar | max |  |  |  | Name of lesson |
| 5 | Title | nvarchar | max |  |  |  | Title of lesson |
| 6 | Description | nvarchar | max |  |  |  | Description of lesson |
| 7 | Video | nvarchar | max |  |  |  | Video of lesson |

1. **tblBlogFeedback**

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | BlogFeedbackID | int |  |  | X | PK | ID of Feedback |
| 2 | BlogID | int |  |  | X | FK | ID of blog |
| 3 | UserID | int |  |  | X | FK | ID of User |
| 4 | BlogFeedbackContent | nvarchar | max |  | X |  | Feedback of content |
| 5 | BlogFeedbackDate | datetime |  |  | X |  | Date of feedback |

1. **tblQuestion**

## 

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | QuestionID | int |  |  | X | PK | ID of Question |
| 2 | Content | nvarchar | max |  | X |  | Content of Question |
| 3 | CorrectAnswer | int |  |  | X |  | Correct answer of Question |
| 4 | QuizID | int |  |  | X | FK | ID of quiz |

## tblCategory

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | CategoryID | int |  |  | X | PK | ID of Category |
| 2 | Name | nvarchar | max |  | X |  | Name of category |

## tbl ReviewComments

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | ReviewCommentID | int |  |  | X | PK | ID of review comment |
| 2 | ReviewFeedBackID | int |  |  | X | FK | ID of review feedback |
| 3 | ReviewID | int |  |  | X | FK | ID of review |
| 4 | ReviewCommentContent | nvarchar | max |  | X |  | Content of review comment |
| 5 | ReviewCommentDate | datetime |  |  | X |  | Date of review comment |

## tbl UserRole

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | RoleID | int |  |  | X | PK | ID of UserRole |
| 2 | Name | nvarchar | max |  | X |  | Name of UserRole |

## tbl ReviewFeedback

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | ReviewFeedbackID | int |  |  | X | PK | ID of review feedback |
| 2 | ReviewID | int |  |  | X | FK | ID of review |
| 3 | ReviewFeedbackContent | nvarchar | max |  | X |  | Content of review feedback |
| 4 | ReviewFeedbackDate | datetime |  |  | X |  | Date of review feddback |

1. **tbl Quiz**

## 

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | QuizID | int |  |  | X | PK | ID of Quiz |
| 2 | LessionID | int |  |  | X | FK | ID of Lession |

## tbl Review

| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | ReviewID | int |  |  | X | PK | ID of review |
| 2 | UserID | int |  |  | X | FK | ID of user |
| 3 | CourseID | int |  |  | X | FK | ID of course |
| 4 | Vote | int |  |  | X |  | Vote of review |