

**ONLINE LEARNING WEBSITE**

**Software Design Specification**

– Hanoi, August 2022 –

Record of changeS

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 4](#_Toc96516286)

[1. Code Packages 4](#_Toc96516287)

[2. Database Design 4](#_Toc96516288)

[a. Database Schema 4](#_Toc96516289)

[b. Table Description 4](#_Toc96516290)

[II. Code Designs 5](#_Toc96516291)

[1. <Feature/Function Name1> 5](#_Toc96516292)

[a. Class Diagram 5](#_Toc96516293)

[b. Class Specifications 5](#_Toc96516294)

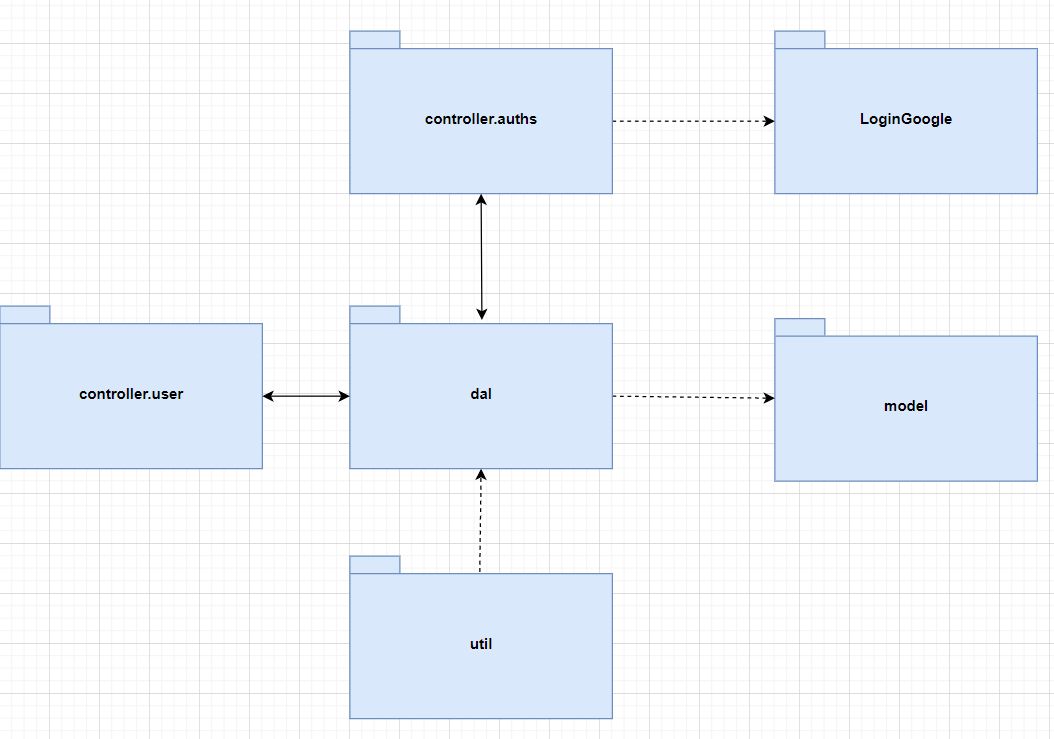
[c. Sequence Diagram(s) 5](#_Toc96516295)

[d. Database queries 6](#_Toc96516296)

[2. <Feature/Function Name2> 6](#_Toc96516297)

# I. Overview

## 1. Code Packages



***Package descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *Dal* | *This component is used to get data directly FROM the database* |
| *02* | *Util* | *<Description of the package>* |
| *03* | *Model* | *This component includes interfaces used to receive data retrieved FROM the database and classes used to receive data* |
| *04* | *Controller.user* | *Acts as a go-between for the service layer and the view, passing on different input requests.* |
| *05* | *Controller.auths* | *This component for the service authorities, such as validate user name password, sign in, log out.* |
| *06* | *LoginGoogle* | *This component use to login page with Gmail.* |

## 2. Database Design

### a. Database Schema

Diagram

Description automatically generated

### b. Table Description

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *1* | *User* | *Information of the User in Online learning system* |
| *2* | *Discussion* | *The table store rate and discussion of user in Online learning system* |
| *3* | *Record* | *Storage practice date, mark of student when they finished a multiple choice practice of course.* |
| *4* | *Enroll* | *Store user's enrollment information for courses* |
| *5* | *Course* | *Information of the course in Online learning system* |
| *6* | *CommentCourse* | *The table store rate, question, and discussion of user in Online learning system* |
| *7* | *Quizz* | *The practice library of Online learning system* |
| *8* | *Question* | *The question library of Online learning system* |
| *9* | *Percentage* | *Store percentage of student when they finish a lesson* |
| *10* | *CommentLesson* | *The table store rate, question, and discussion of user in Online learning system* |
| *11* | *Lesson* | *Information of the lesson in Online learning system* |
| *12* | *Chapter* | *Information of the chapter in Online learning system* |
| *13* | *Image* | *The image library of Online learning system* |

# II. Code Designs

## 1. Register

### a. Class Diagram

Diagram

Description automatically generated

### b. Class Specifications

*SignUpServlet*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doPost(HttpServletRequest request, HttpServletResponse response)* | *Get email, password and fullname from resgister form, check it and add to database* |
|  |  |  |

***DBContext***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getConnection()* | *Get connection to database* |
|  |  |  |

***User***

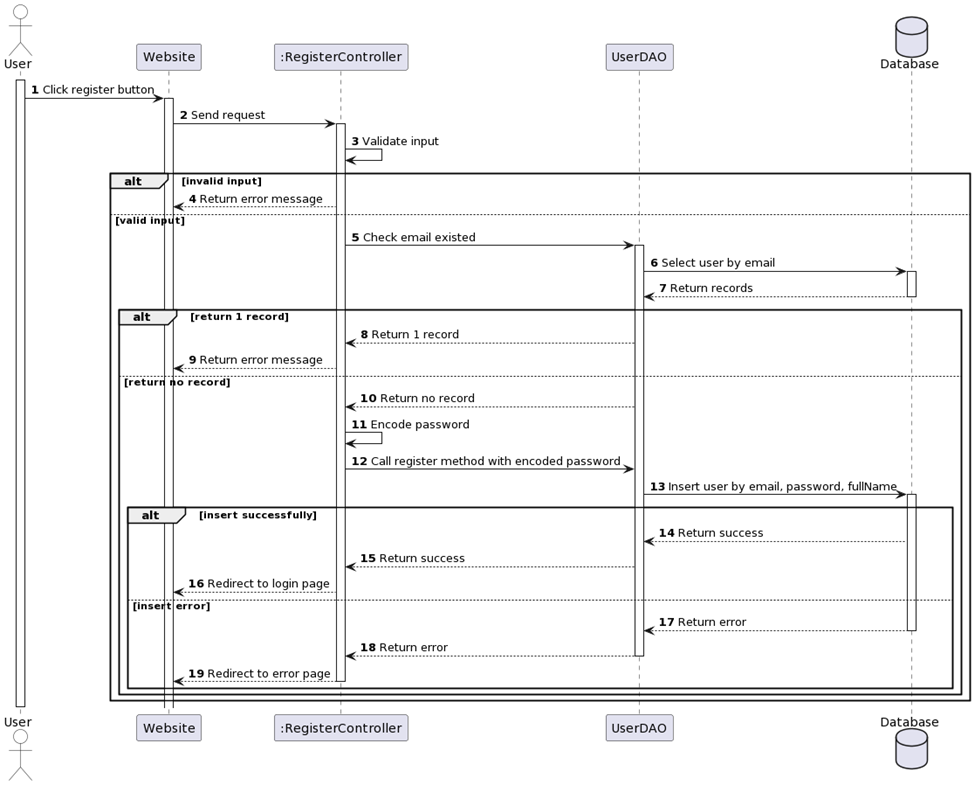
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***UserDAO***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *ChechUser(String s)* | *Check if user name is exist or not* |
| *02* | *CheckUserGoogle*  *(String username )* | *Check google email is extisted or not in the system* |

### c. Sequence Diagram(s)

*[Provide the sequence diagram(s) for the feature, see the sample below]*



### d. Database Queries

**INSERT INTO [dbo].[User]\n"**

**+ " ([user\_name]\n"**

**+ " ,[user\_mail]\n"**

**+ " ,[user\_password]\n"**

**+ " ,[user\_role]\n"**

**+ " ,[user\_gender]\n"**

**+ " ,[user\_address]\n"**

**+ " ,[user\_phone]\n"**

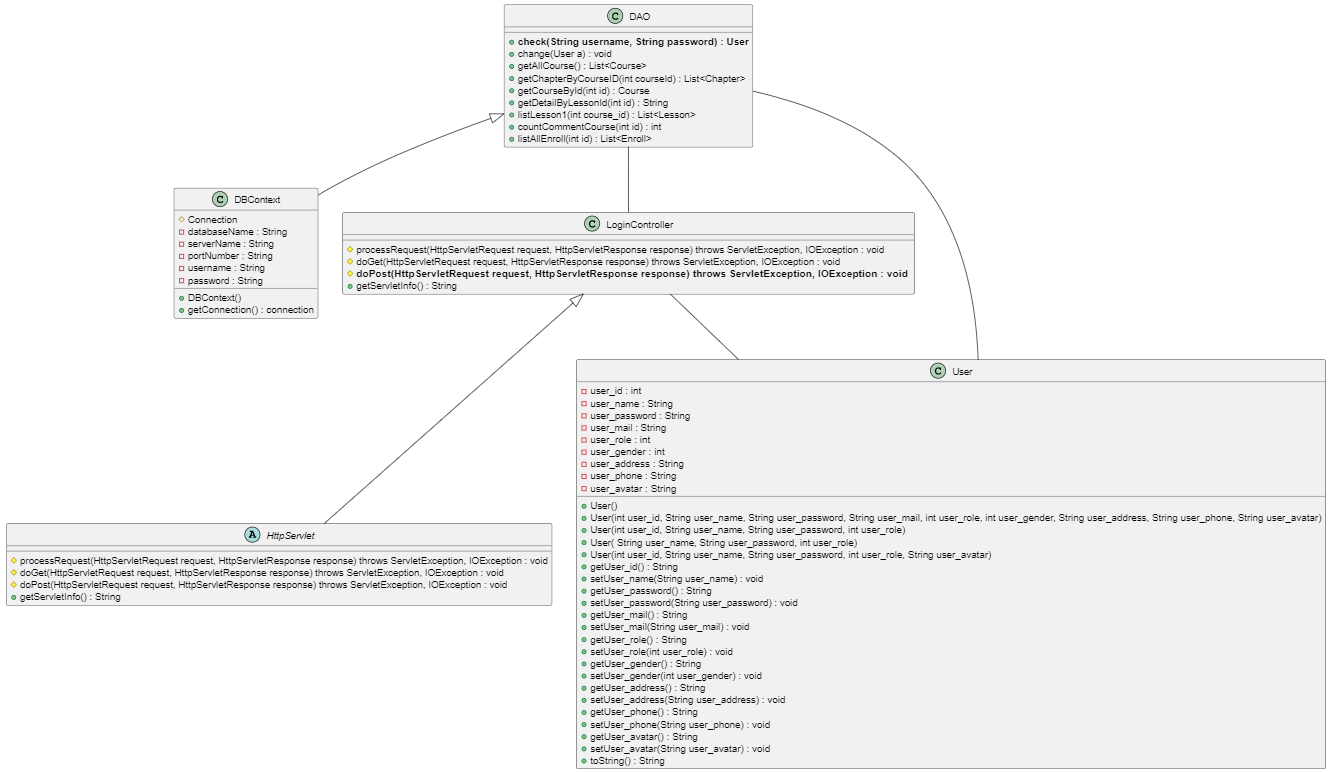
**+ " ,[user\_avatar])\n"**

**+ " VALUES\n"**

**+ " (?,?,?,?,?,?,?,?)**

## 2. Login

### a. Class Diagram



### b. Class Specifications

#### LoginController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doPost(HttpServletRequest request, HttpServletResponse response)* | *Get email, password and fullname from resgister form, check it and add to database* |
|  |  |  |

***DBContext***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getConnection()* | *Get connection to database* |
|  |  |  |

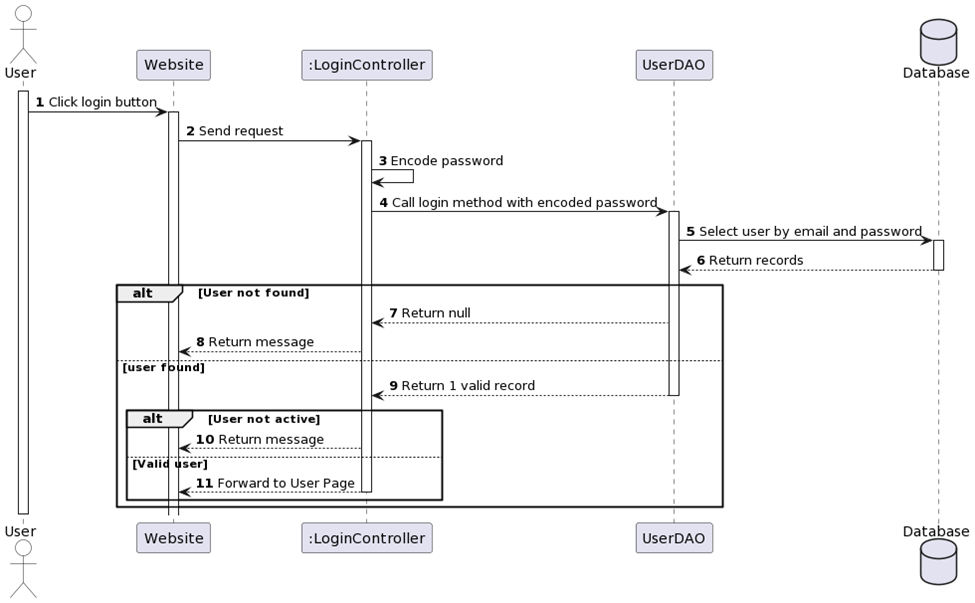
***User***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***DAO***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Check(String username, String password)* | *Check account in database to login* |

### c. Sequence Diagram(s)



### d. Database Queries

**- Check User**

**SELECT [user\_name],[user\_password]**

**FROM [SWP-Project].[dbo].[User] where user\_name=? and user\_password=?**

## 3. Home page

### a. Class Diagram

### b. Class Specifications

#### HomeController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet(HttpServletRequest request, HttpServletResponse response)* | *List all courser from databse into homepage* |
|  |  |  |

***DBContext***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getConnection()* | *Get connection to database* |
|  |  |  |

***User***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***Enroll***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***Course***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***Chapter***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***DAO***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *GetAllCourse(List<Course>)* | *Get all Cousre from database* |
|  |  |  |

### c. Sequence Diagram(s)

### Không có mô tả.

### d. Database Queries

**SELECT TOP (1000) [Course\_id]\n"**

**+ " ,[course\_name]\n"**

**+ " ,[course\_description]\n"**

**+ " ,[course\_price]\n"**

**+ " ,[course\_number\_lesson]\n"**

**+ " ,[course\_image]\n"**

**+ " FROM [SWP-Project].[dbo].[Course]\n"**

**+ " where course\_price =0**

## 4. ChangePass

### a. Class Diagram



### b. Class Specifications

#### ChangePassController

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet(HttpServletRequest request, HttpServletResponse response)* | *Check user is existed or not, if not then you can change password* |
|  |  |  |

***DBContext***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getConnection()* | *Get connection to database* |
|  |  |  |

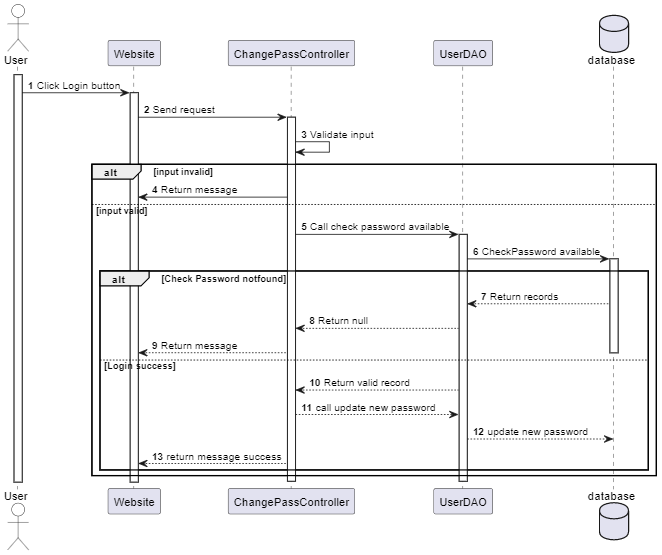
***User***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***DAO***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Change(User a)* | *Check user in database to change password* |

### c. Sequence Diagram(s)



### d. Database Queries

**UPDATE [dbo].[User]\n"**

**+ " SET \n"**

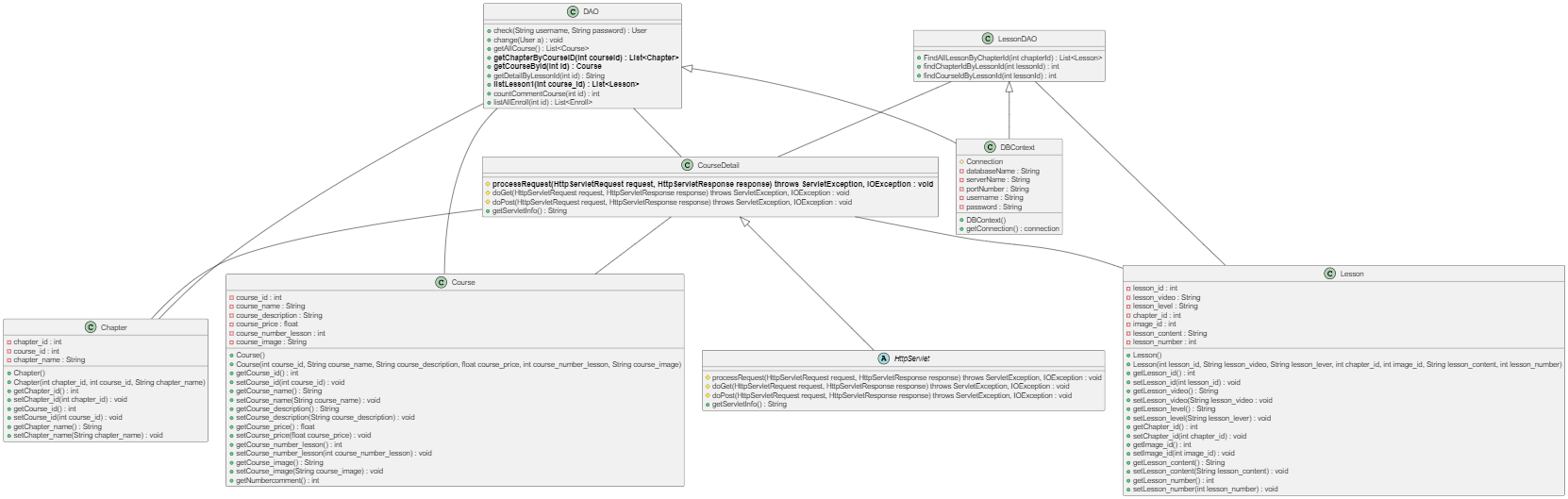
**+ " [user\_password] = ?\n"**

**+ " \n"**

**+ " WHERE user\_name=?**

## 5. CourseDetail

### a. Class Diagram



### b. Class Specifications

#### CourseDetail

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *processRequest(HttpServletRequest request, HttpServletResponse response)* | *Get Course by id from database and show in page* |
|  |  |  |

***DBContext***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getConnection()* | *Get connection to database* |
|  |  |  |

***Course***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

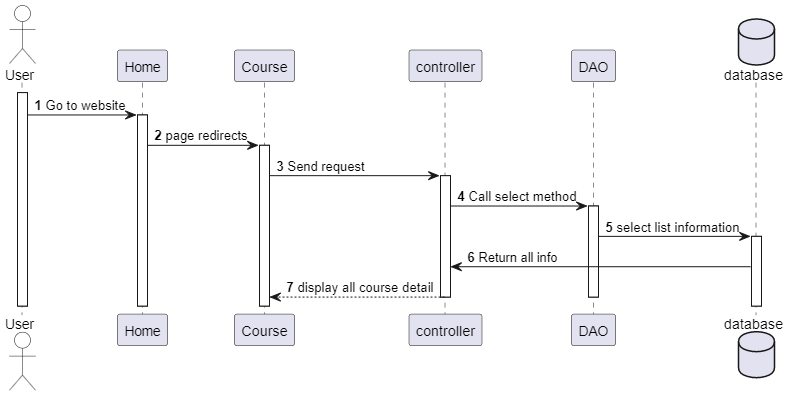
***Chapter***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***Lesson***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

### c. Sequence Diagram(s)



### d. Database Queries

## SELECT [Course\_id]\n"

## + " ,[course\_name]\n"

## + " ,[course\_description]\n"

## + " ,[course\_price]\n"

## + " ,[course\_number\_lesson]\n"

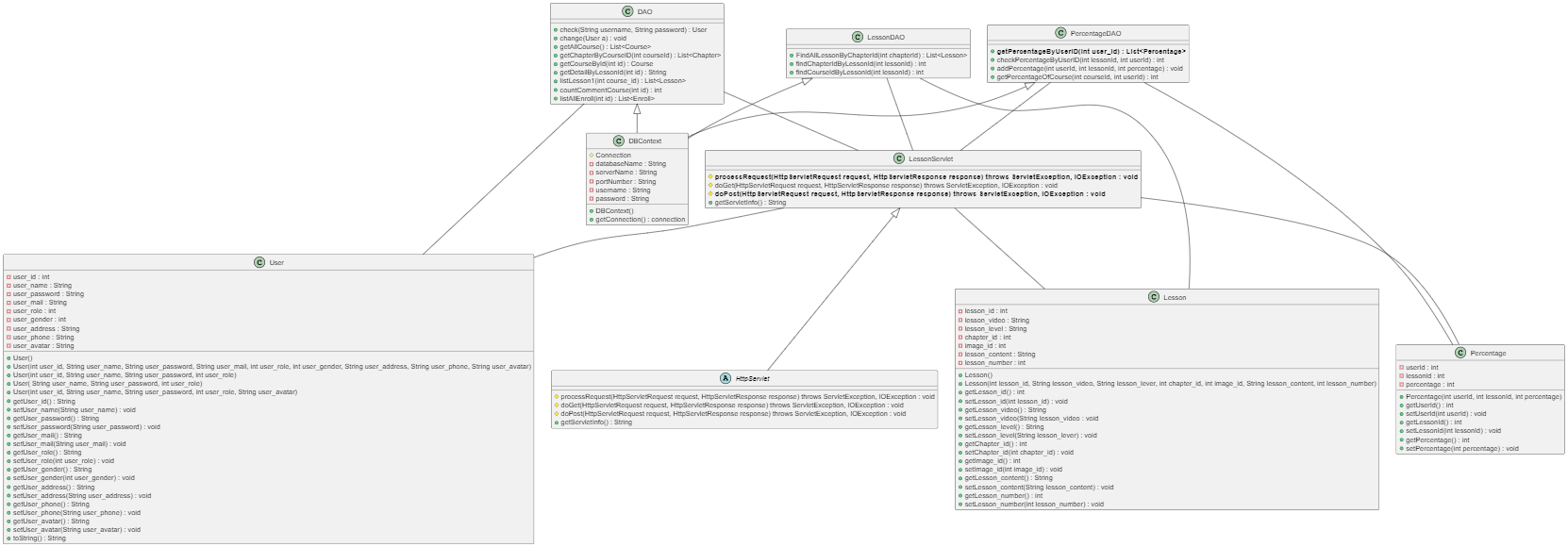
## + " ,[course\_image]\n"

## + " FROM [dbo].[Course]\n"

## + " where Course\_id=?

## 6. LessonDetail

### a. Class Diagram



### b. Class Specifications

#### LessonDetail

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doPost(HttpServletRequest request, HttpServletResponse response)* | *Get all lesson from course by FindAllLessonByChapterId* |
|  |  |  |

***DBContext***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getConnection()* | *Get connection to database* |
|  |  |  |

***User***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***Lesson***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***Percentage***

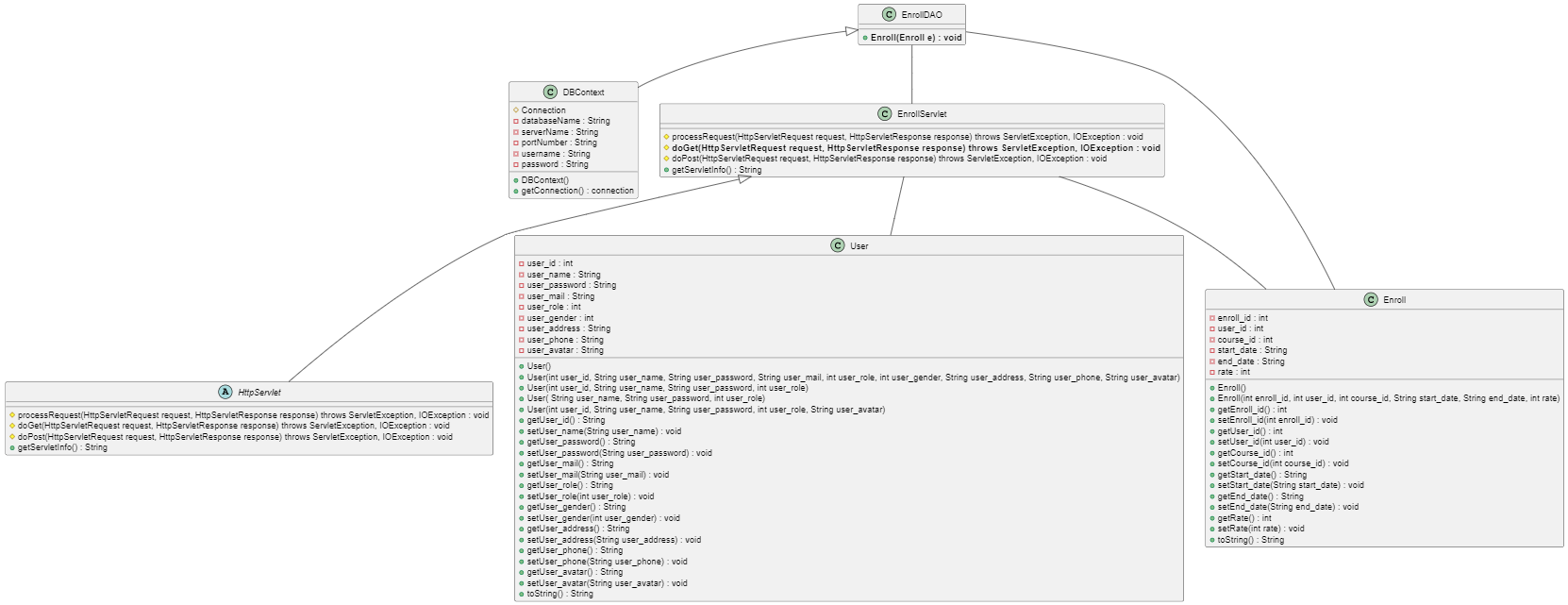
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

### c. Sequence Diagram(s)

### d. Database Queries

## 7. Enroll

### a. Class Diagram



### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries9. CommentCourse

### a. Class Diagram

### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

*[Provide the detailed description for the class methods]*

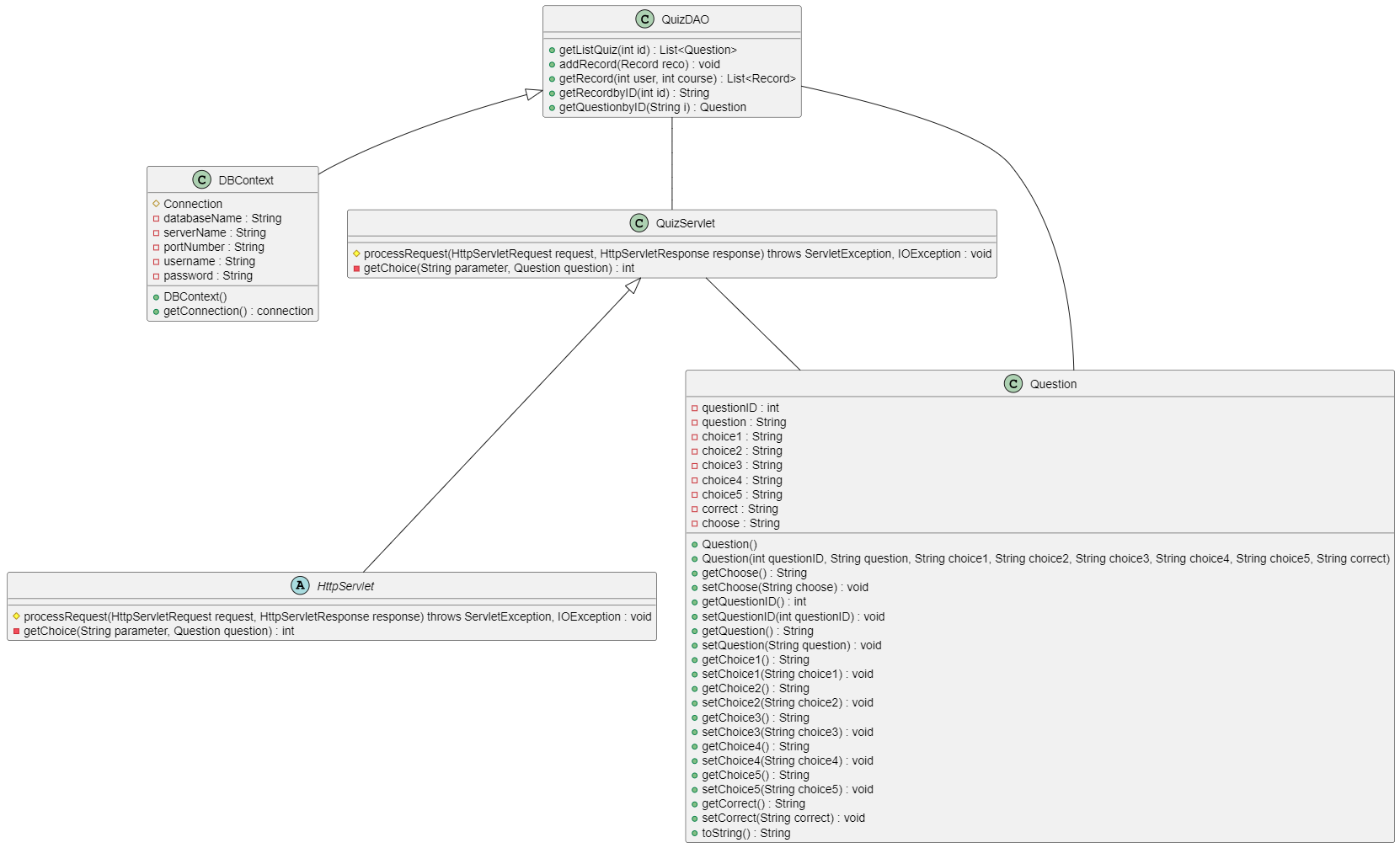
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 8. Quizz

### a. Class Diagram



### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

*[Provide the detailed description for the class methods]*

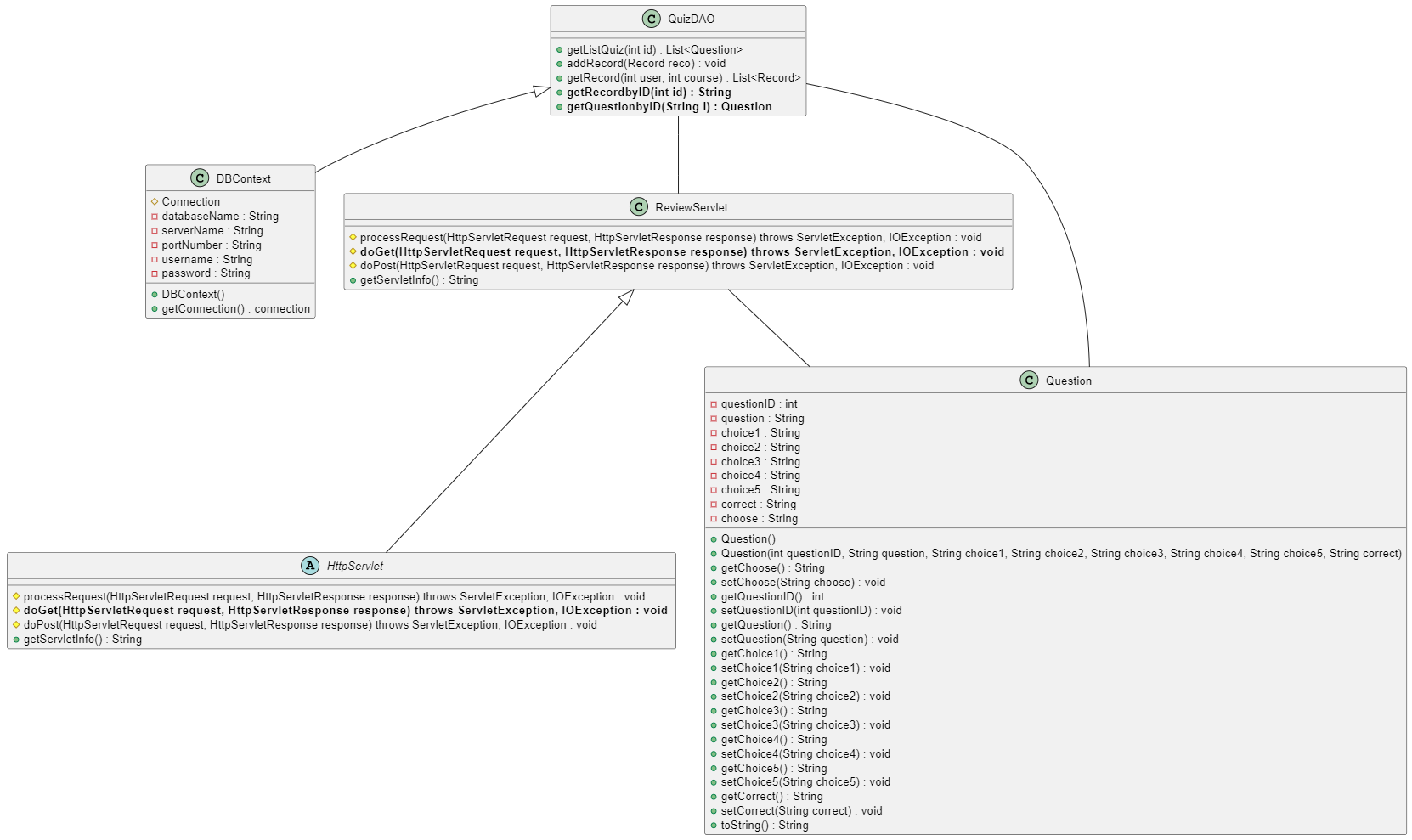
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 9. Review

### a. Class Diagram



### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

*[Provide the detailed description for the class methods]*

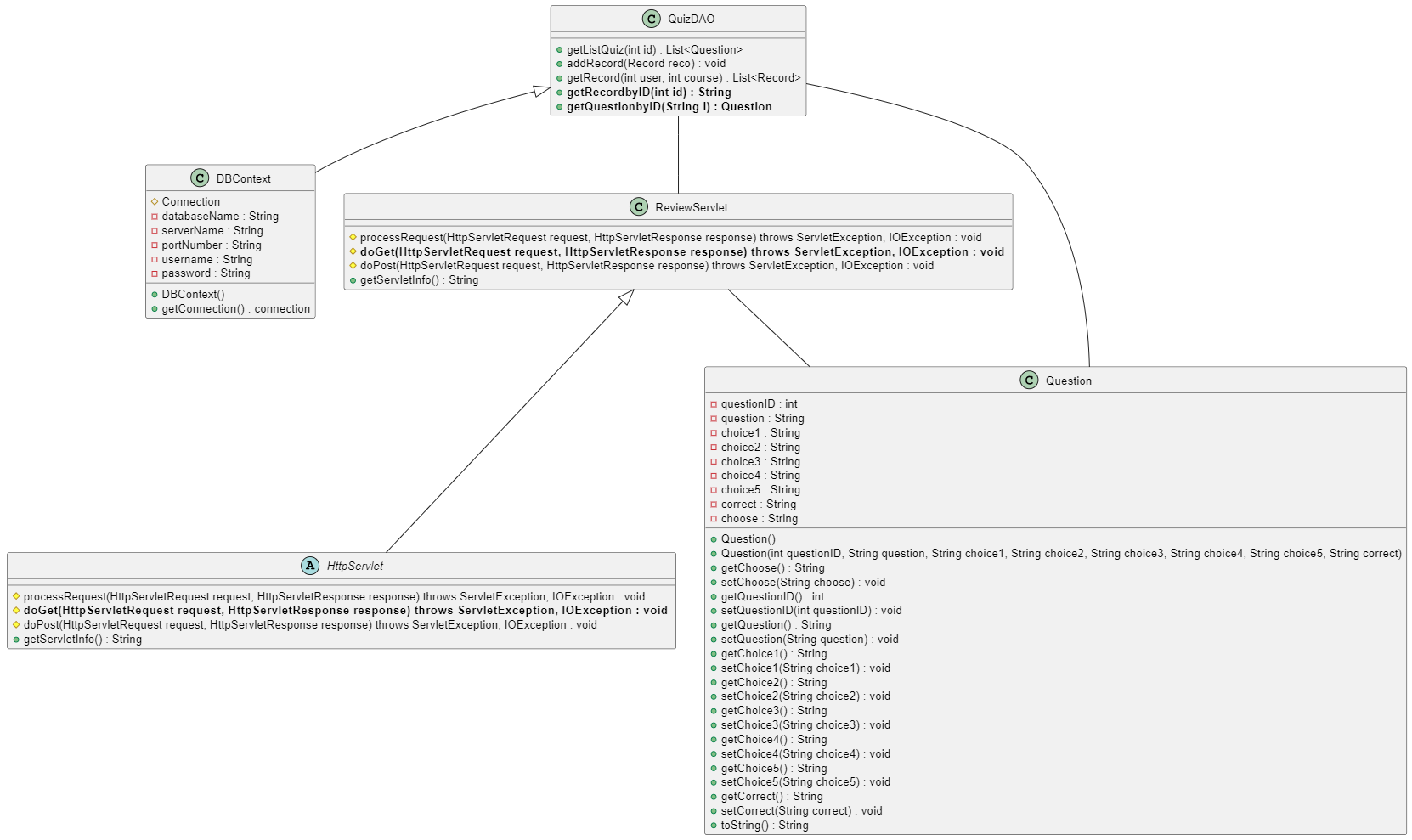
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 10. Record

### a. Class Diagram



### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 11. Topic

### a. Class Diagram



### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

*[Provide the detailed description for the class methods]*

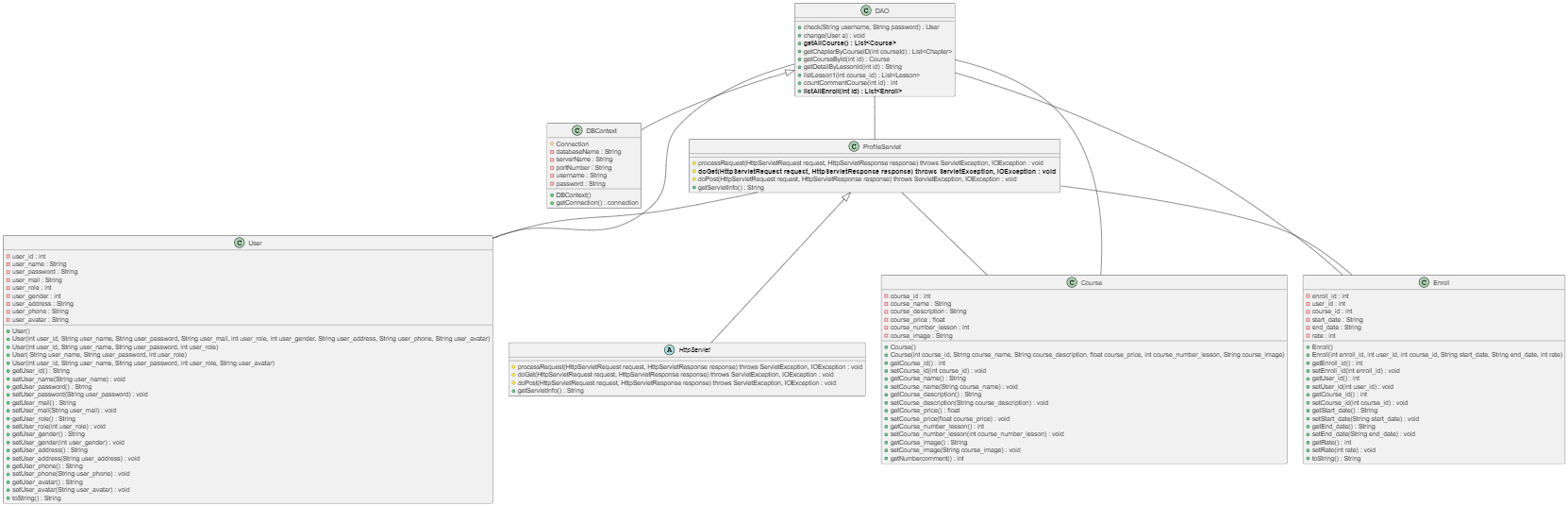
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries

## 12. Profile

### a. Class Diagram



### b. Class Specifications

#### ProfileServlet

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet(HttpServletRequest request, HttpServletResponse response)* | *Get profile of course* |
|  |  |  |

***DBContext***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getConnection()* | *Get connection to database* |
|  |  |  |

***User***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***Course***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***Chapter***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***Enroll***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Constructors* | *Entity initialization* |
| *02* | *Getter* | *Get value of each attributes* |
| *03* | *Setter* | *Set value of each attributes* |

***DAO***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Change(User a)* | *Check user in database to change password* |

### c. Sequence Diagram(s)

### d. Database Queries

- GetAllCourse

SELECT TOP (1000) [Course\_id]\n"

+ " ,[course\_name]\n"

+ " ,[course\_description]\n"

+ " ,[course\_price]\n"

+ " ,[course\_number\_lesson]\n"

+ " ,[course\_image]\n"

+ " FROM [SWP-Project].[dbo].[Course]\n"

+ " where course\_price =0

- ListAllEnroll

SELECT TOP (1000) [Enroll\_id]\n"

+ " ,[user\_id]\n"

+ " ,[course\_id]\n"

+ " ,[start\_date]\n"

+ " ,[end\_date]\n"

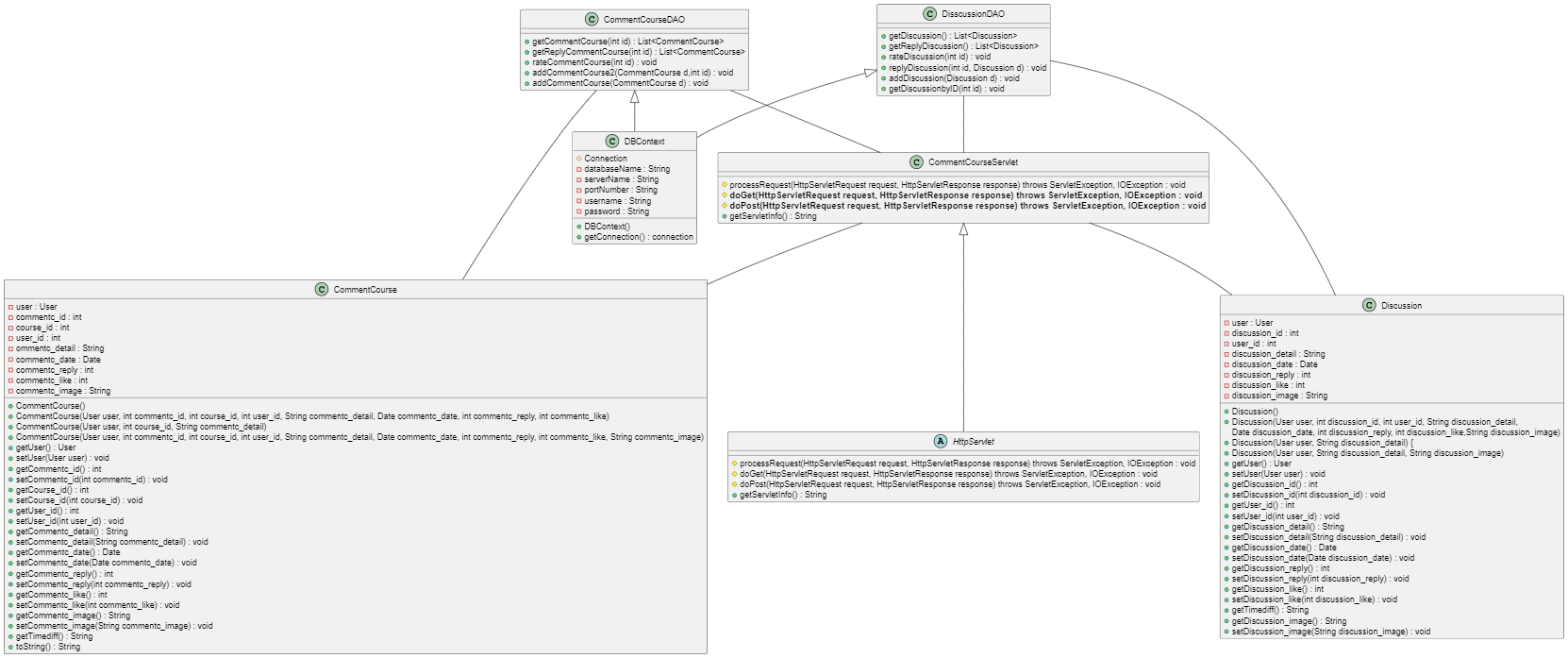
+ " ,[rate]\n"

+ " FROM [SWP-Project].[dbo].[Enroll]\n"

+ " where [user\_id]=?

## 13. CommentCourse

### a. Class Diagram



### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

*[Provide the detailed description for the class methods]*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

### d. Database Queries