`



**LEARNING ONLINE**

**Software Design Document**

– Hanoi, Feb 2023 –

**Table of Contents**

[I. Overview 3](#_Toc127527652)

[1. Code Packages/Namespaces 3](#_Toc127527653)

[2. Coding Convent 3](#_Toc127527654)

[II. Code Designs 4](#_Toc127527655)

[1. Register and Login 4](#_Toc127527656)

[a. Class Diagram 4](#_Toc127527657)

[b. Class Specifications 5](#_Toc127527658)

[c. Sequence Diagram(s) 6](#_Toc127527659)

[d. Database queries 6](#_Toc127527660)

[2. User profile 8](#_Toc127527661)

[a. Class Diagram 8](#_Toc127527662)

[b. Class Specifications 8](#_Toc127527663)

[Database queries 8](#_Toc127527664)

[c. Sequence Diagram(s) 9](#_Toc127527665)

[d. Database queries 9](#_Toc127527666)

[3. Bloglist 9](#_Toc127527667)

[a. Class Diagram 9](#_Toc127527668)

[b. Class Specifications 9](#_Toc127527669)

[c. Sequence Diagram(s) 9](#_Toc127527670)

[d. Database queries 9](#_Toc127527671)

[III. Database Design 10](#_Toc127527672)

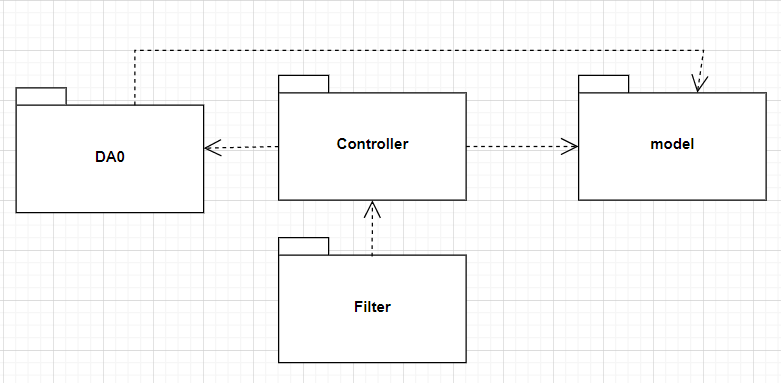
[1. Database Schema 10](#_Toc127527673)

[2. Entity Relationship Diagram 10](#_Toc127527674)

[3. Table Description 11](#_Toc127527675)

# 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* |

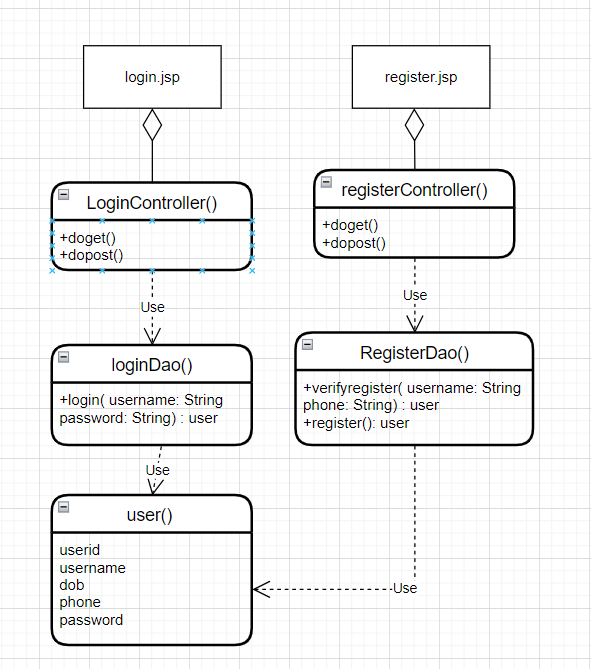
## 

## 2. Coding Convent

# II. Code Designs

## 1. Register and Login

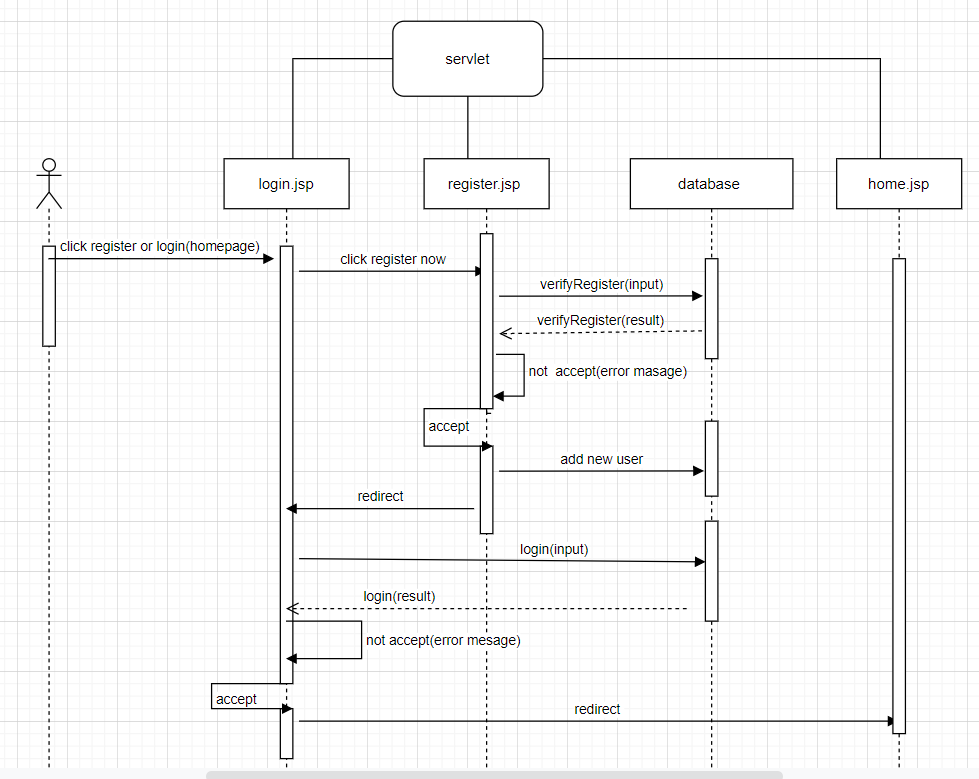
### a. Class Diagram



### 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

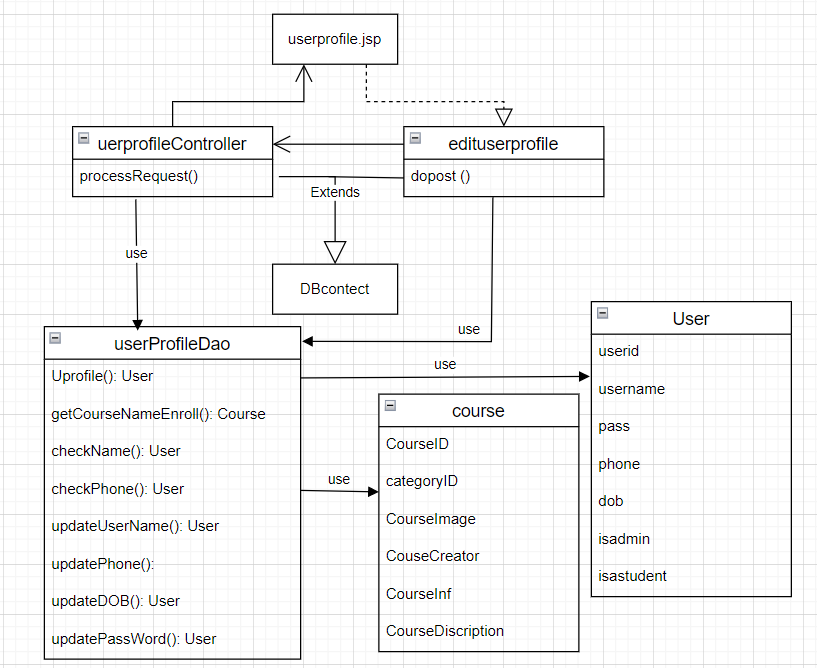
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Database queries** |
| *01* | *Login()* | *<*"SELECT \*\n"  + " FROM [dbo].[User]\n"  + " where UserName = ? and PassWord = ?"*>*  *check user exist* |
| *02* | *verifyregister()* | *<*SELECT \*\n"  + " FROM [dbo].[User]\n"  + " where [UserName] = ? or Phone = ?*>*  *Check if the name or phone is not the same because each account has only one phone number and one name* |
| *03* | *Register()* | *<"INSERT INTO [dbo].[User]\n"*  *+ " ([UserName]\n"*  *+ " ,[PassWord]\n"*  *+ " ,[Phone]\n"*  *+ " ,[Dob]\n"*  *+ " ,[Admin]\n"*  *+ " ,[Student])\n"*  *+ " VALUES\n"*  *+ " (?\n"*  *+ " ,?\n"*  *+ " ,?\n"*  *+ " ,?\n"*  *+ " ,?\n"*  *+ " ,?)">*  *save account information to database* |

### 

## 

## 2. User profile

### a. Class Diagram



### b. Class Specifications

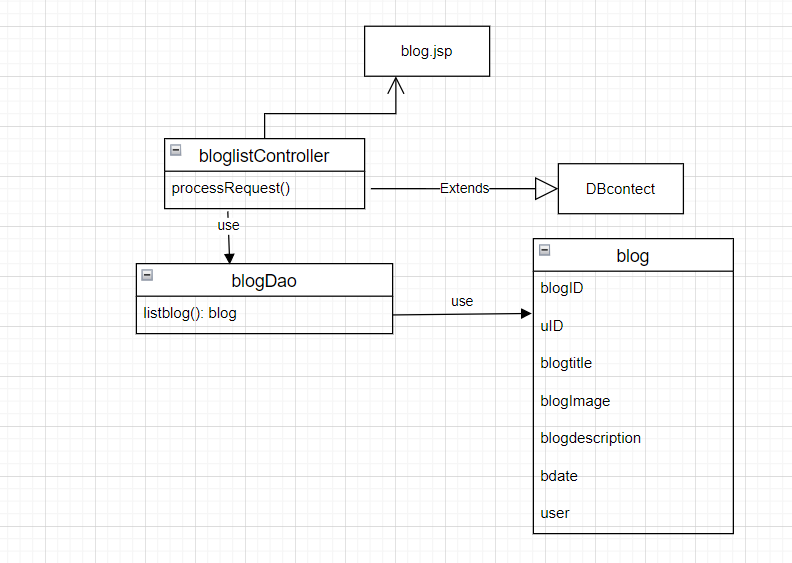
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Method** | **Description** | Database queries |
| *01* | *Uprofile()* | *get log user* | *SELECT \* \n"*  *+ " FROM [dbo].[User]\n"*  *+ " where UserID = ?* |

### c. Sequence Diagram(s)

### d. Database queries

## 3. Bloglist

### a. Class Diagram



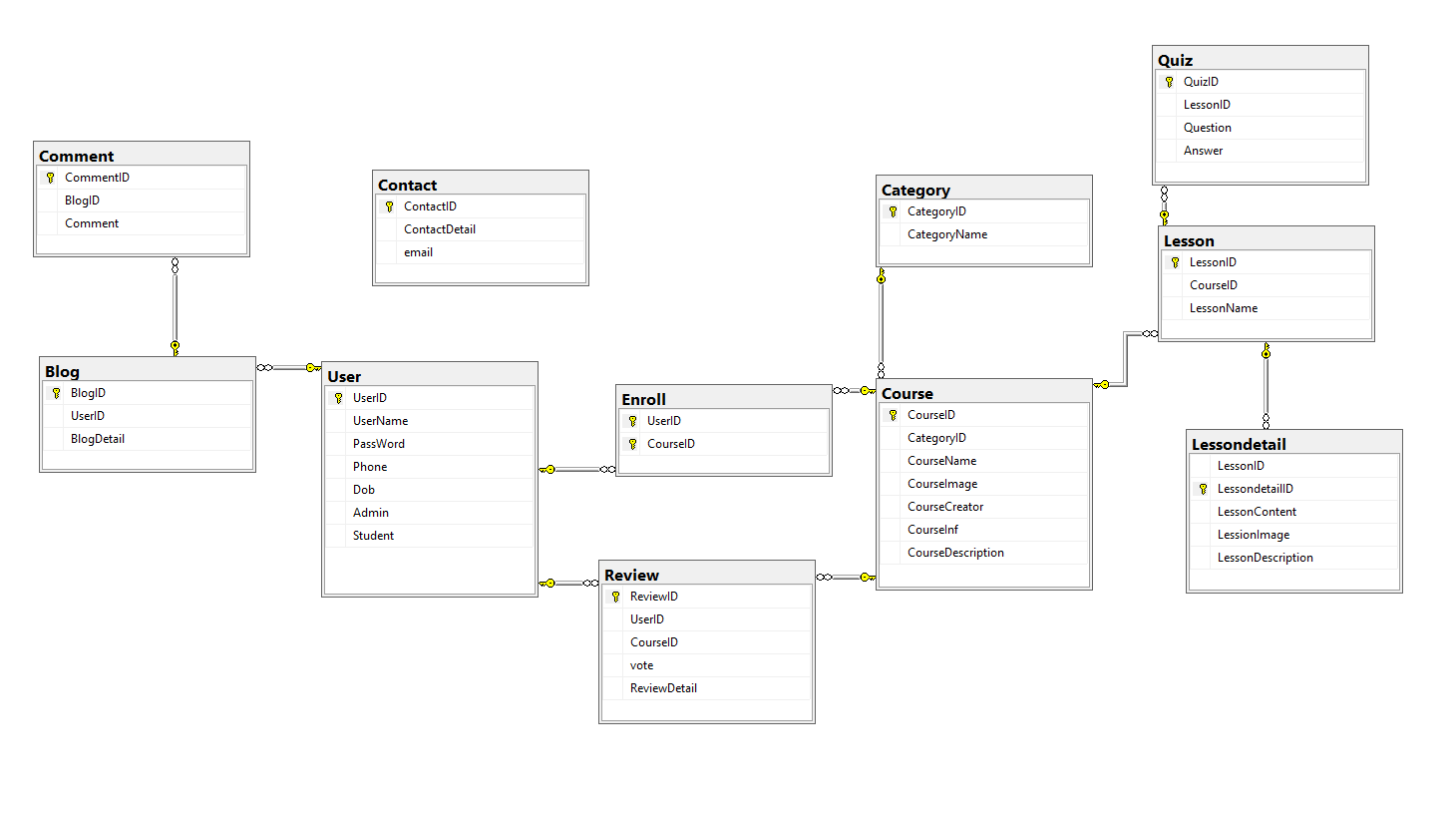
### b. Class Specifications

### c. Sequence Diagram(s)

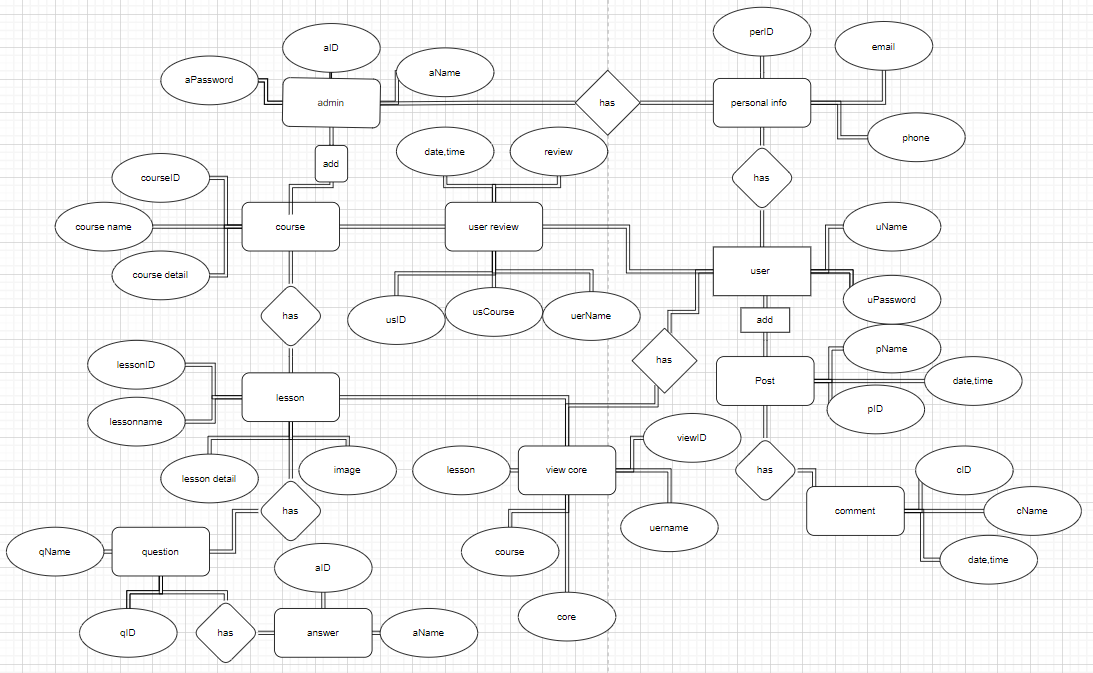
### d. Database queries

# 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* |