

**<<ONLINE LEARNING>>**

**Software Design Specification**

– Hanoi, August 2022 –

**Record of changeS**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 31/01/2023 | A | All | Create SDS v1.0 |
| 16/02/2023 | M | All | Update SDS v1.1. |
| 26/02/2023 | A | All | Create SDS v2.0. |
| 07/03/2023 | A | All | Create SDS v3.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 4](#_heading=h.gjdgxs)

[1. Code Packages 4](#_heading=h.30j0zll)

[2. Database Design 4](#_heading=h.1fob9te)

[a. Database Schema 4](#_heading=h.3znysh7)

[b. Table Description 4](#_heading=h.2et92p0)

[II. Code Designs 5](#_heading=h.tyjcwt)

1. Login 6

2. Register 5

3.Change password

4. User Authorization

5.Reset Password

6.User Profile Setting

7.Create Course

8.Update Course

9. Delete Course

10.View Course

11.Create Lesson

12.Update Lesson

13.Delete Lesson

14.View Lesson

15.Create FlashCard

16.Update FlashCard

17.Delete FlashCard

18.View FlashCard

19.Create Blog

20.Update Blog

21.Delete Blog

22.View Blog

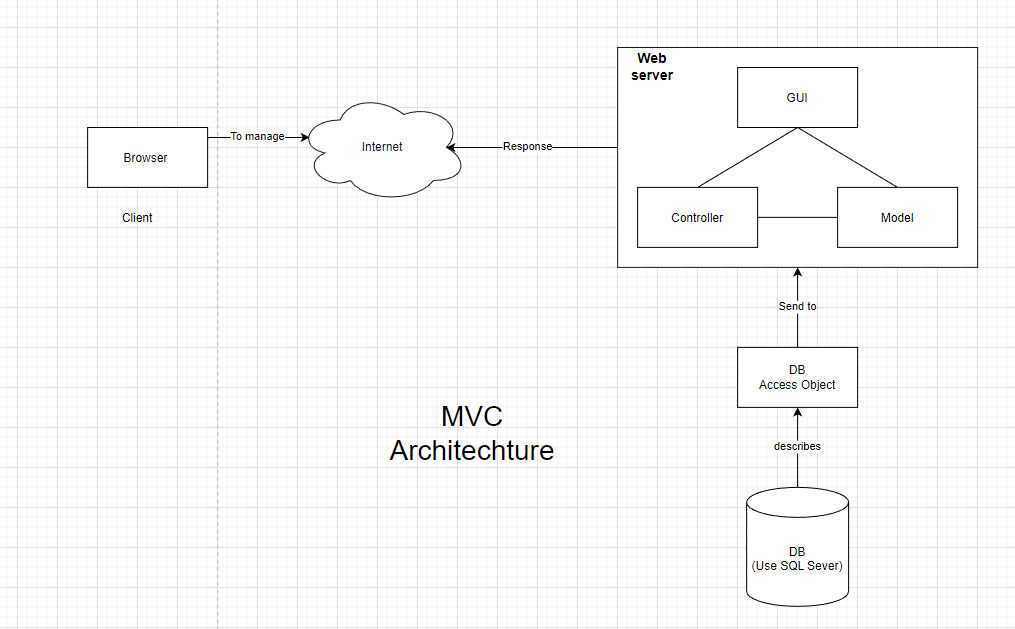
23.Manage account

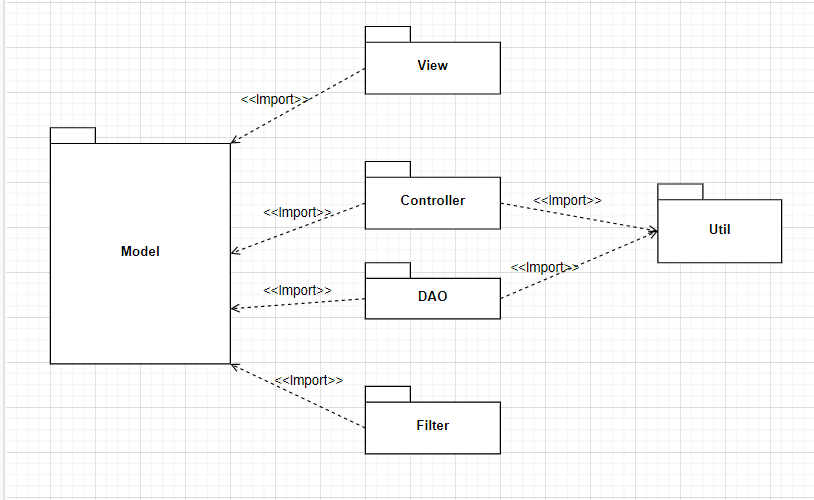
[6](#_heading=h.3rdcrjn)

# I. Overview

## 1. Code Packages

1.1 System Architecture

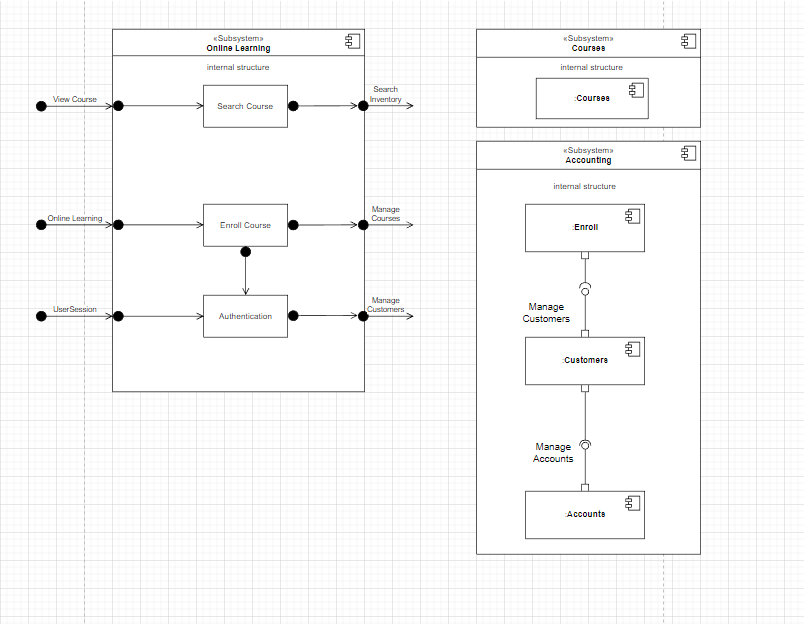




***Package descriptions***

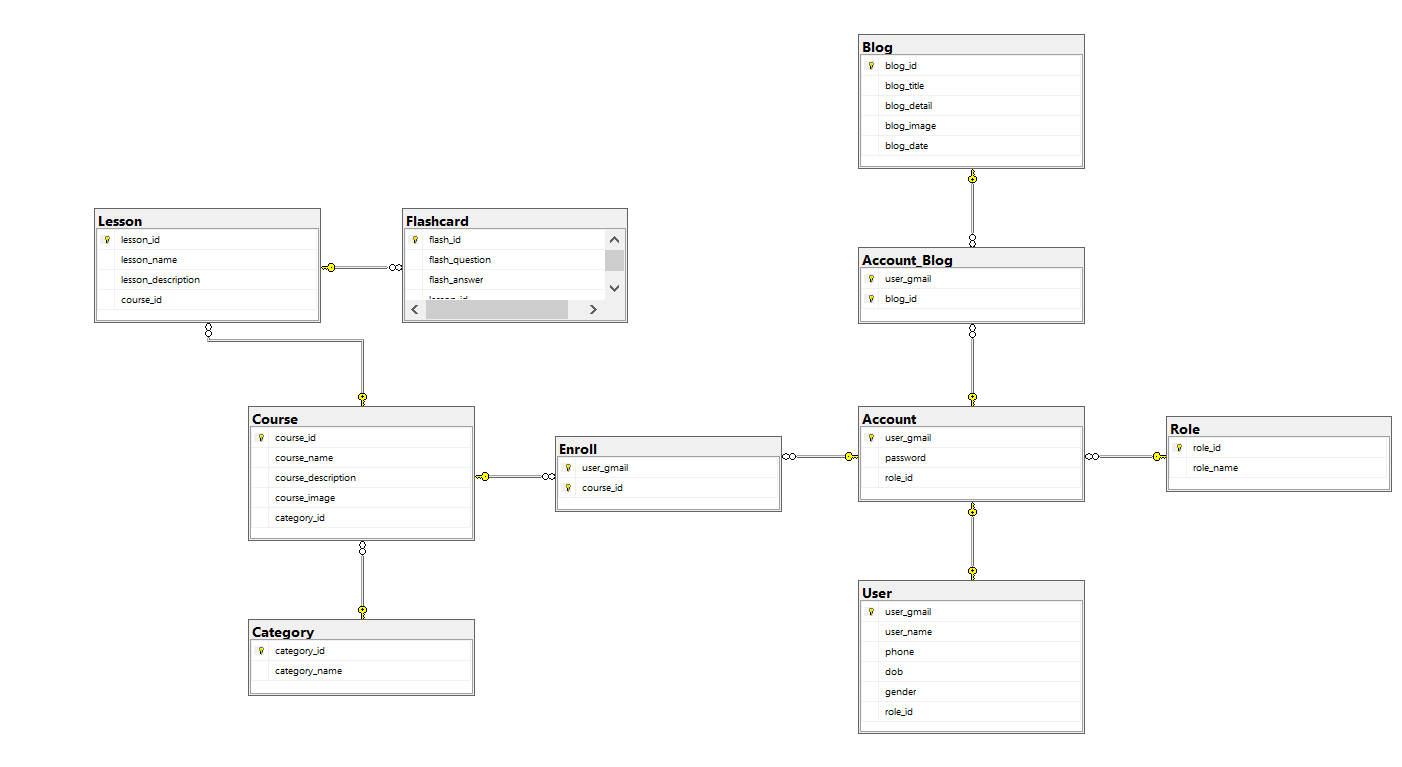
| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *DAO* | | *Contain java class to access and get the data from the database* | | --- |   *Class Naming Convention: AccountDAO* |
| *02* | *Controller* | | *Contain Java Servlet serve as the business logic layer of the application* | | --- |   *Class Naming Convention: LoginController* |
| *03* | *View* | *Contain All file that create the front-end view of the website like java jsp, html, css and javascript* |
| *04* | *Filter* | | *Contains filter to authentication and authorization of the website* | | --- |   *Class Naming convention: LoginFilter* |
| *05* | *Model* | | *Contains all the javabeans represent object from the database design* | | --- |   *Class Naming Convention: Account, Student,Course* |
| *06* | *Util* | | *Contains all functionality to support the other class.* | | --- | |

1.2 Component Architecture



## 2. Database Design

### a. Database Schema



### 

### 

### 

### 

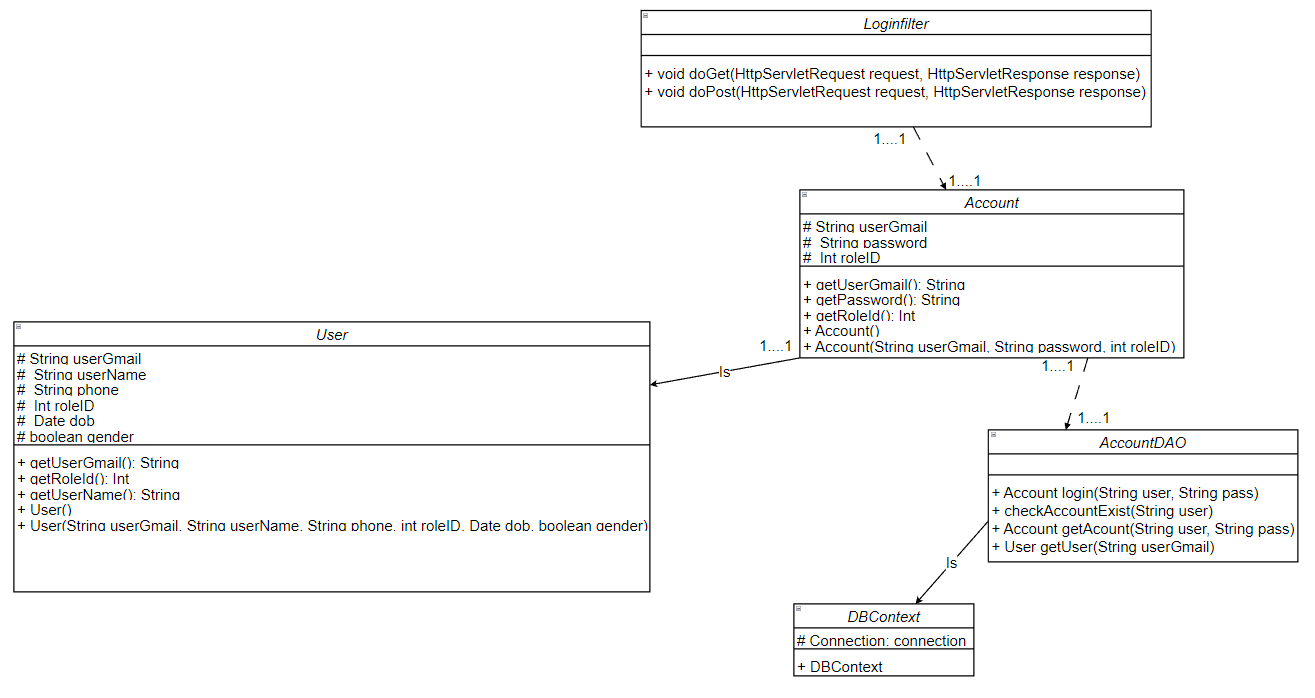
### b. Table Description

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *Account* | *Contains account details to login to the system* |
| *02* | *Course* | *Contains the name of the course .* |
| *03* | *Role* | *Contains the role of the users to access different functions.* |
| *04* | *Lesson* | *Contains the content of the lesson.* |
| *05* | *Flashcard* | *Shows which flashcards for each course and their content.* |
| *06* | *Enroll* | *Show which users enroll for which course.* |
| *07* | *Blog* | *Contains the content of the blog.* |

# II. Code Designs

## 1. Login

### a. Class Diagram

**

### b. Class Specifications

#### Login Controller

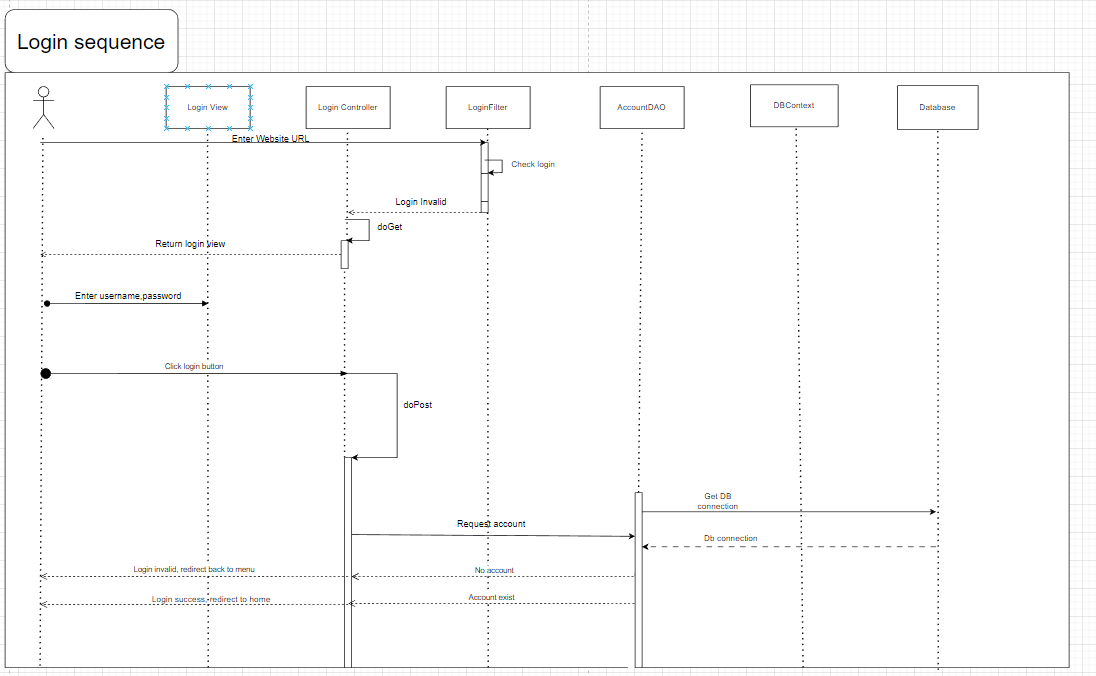
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *doGet* | *Get the login page to the user* |
| *02* | *doPost* | *Process input username and password, check if it exists in the Database to allow user to login and redirect to home page or error otherwise* |

#### AccountDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Account getAccount(String username, String password)* | *Return a account with username and password as parameter. If that username and password don’t exist, return null* |

### c. Sequence Diagram(s)



### d. Database Queries

* In getAccount method:

**+) select \* from account**

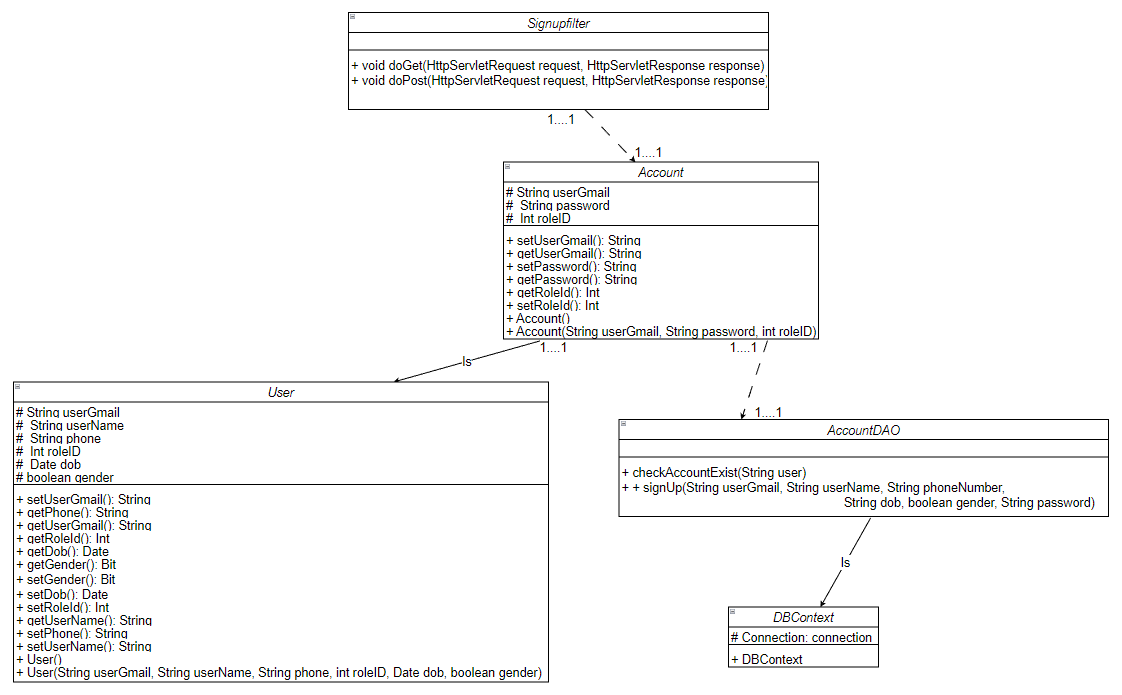
**where user\_gmail=? and password=?**

## 

## 

## 2. Register

### a. Class Diagram



### b. Class Specifications

#### RegisterController

| No | Method | Description |
| --- | --- | --- |
| 01 | Void doGet | *Get the register page to the user* |
| 02 | Void doPost | *Process input of user. Valid data then check if account already exist then return error. Else add account to the database, return success message.* |

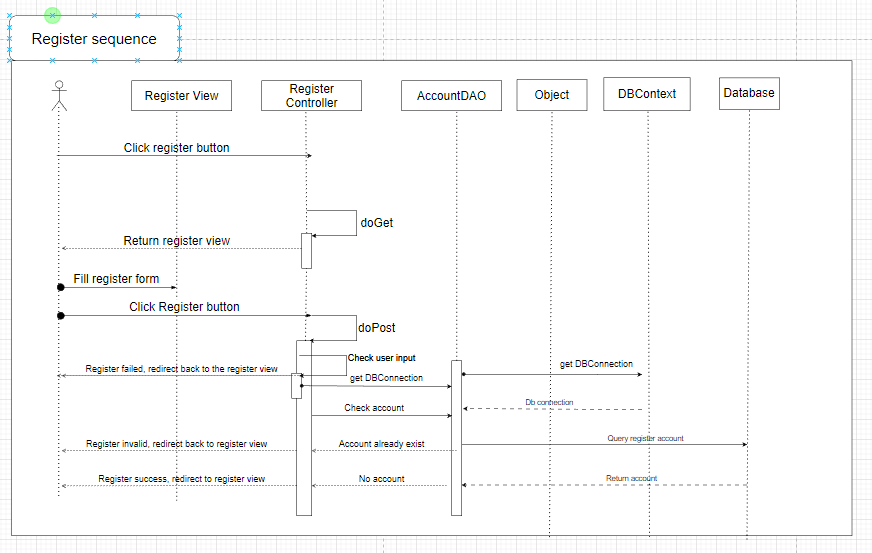
# 

# *AccountDAO*

#### 

| No | Method | Description |
| --- | --- | --- |
| 01 | *Account getAccount(String username)* | *Get the register page to the user* |
| 02 | *Void insertAccount(String usergmail, String password, String username, bit gender, Date dob, int role)* | *Process input of user. Valid data then check if account already exist then return error. Else add account to the database, return success message.* |

### c. Sequence Diagram(s)

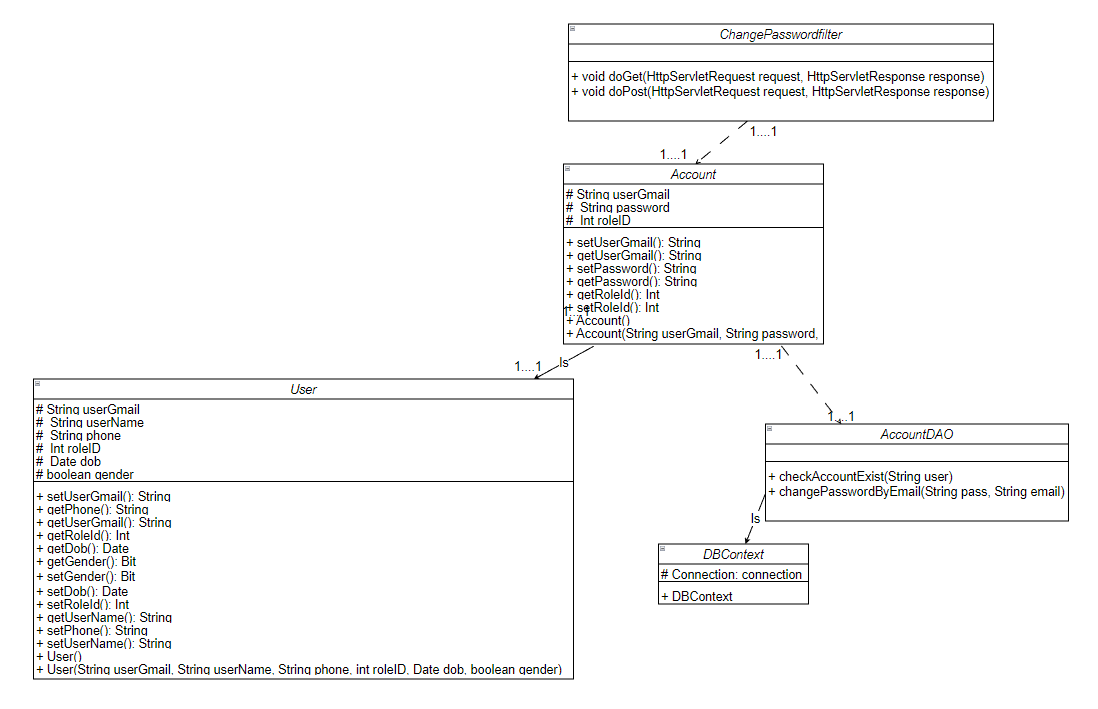
**

### d. Database Queries

**+) insert into Account (user\_gmail,password,role\_id)  
 values (‘ ’, ’ ‘, ’ ‘)**

## 3. Change Password

### a. Class Diagram



### b. Class Specifications

#### ChangePasswordController

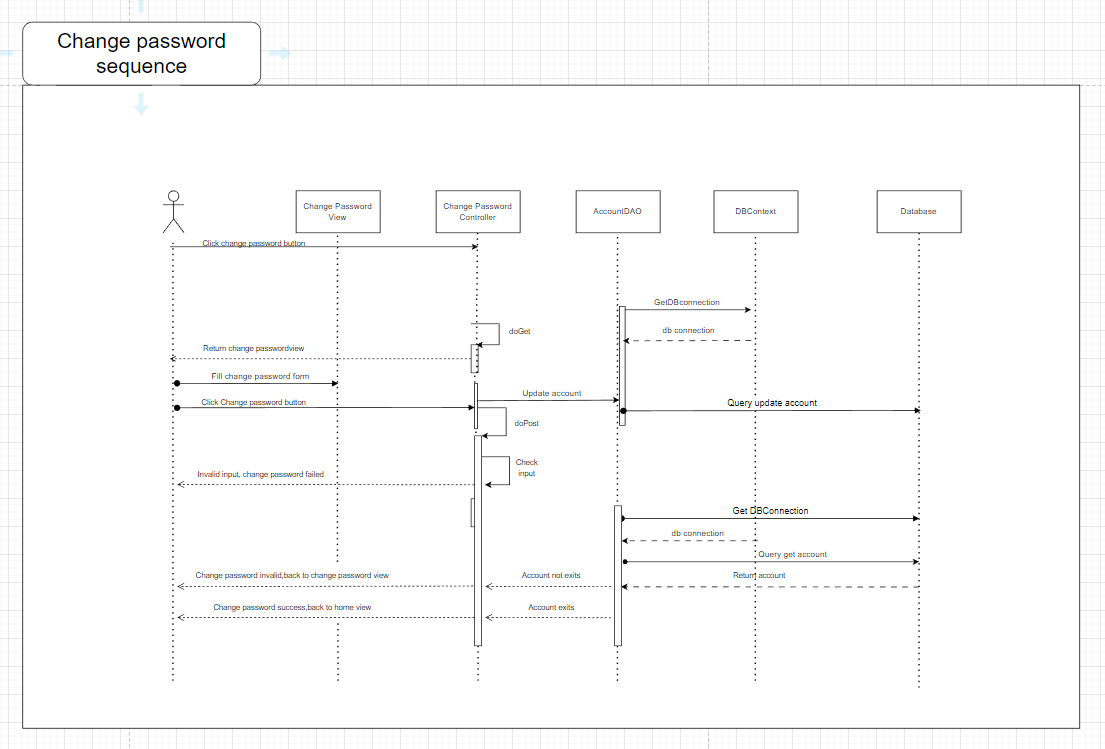
#### 

| No | Method | Description |
| --- | --- | --- |
| 01 | Void doGet | *Return the change password page to the user.* |
| 02 | Void doPost | *Process user’s input. Check account and password entered is valid and New password and retype new password match. If correct then update that account in the database.* |

***AccountDAO***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Account getAccount(String username, String password)* | *Return account with username and password as parameter. If that username and password don’t exist, return null.* |
| *02* | *void updateAccount(String username, String newPassword)* | *Update the account with username in the parameter and set the password to the new Password in the parameter.* |

### c. Sequence Diagram(s)



### d. Database queries

**+) update Account**

**set password = ‘ ‘**

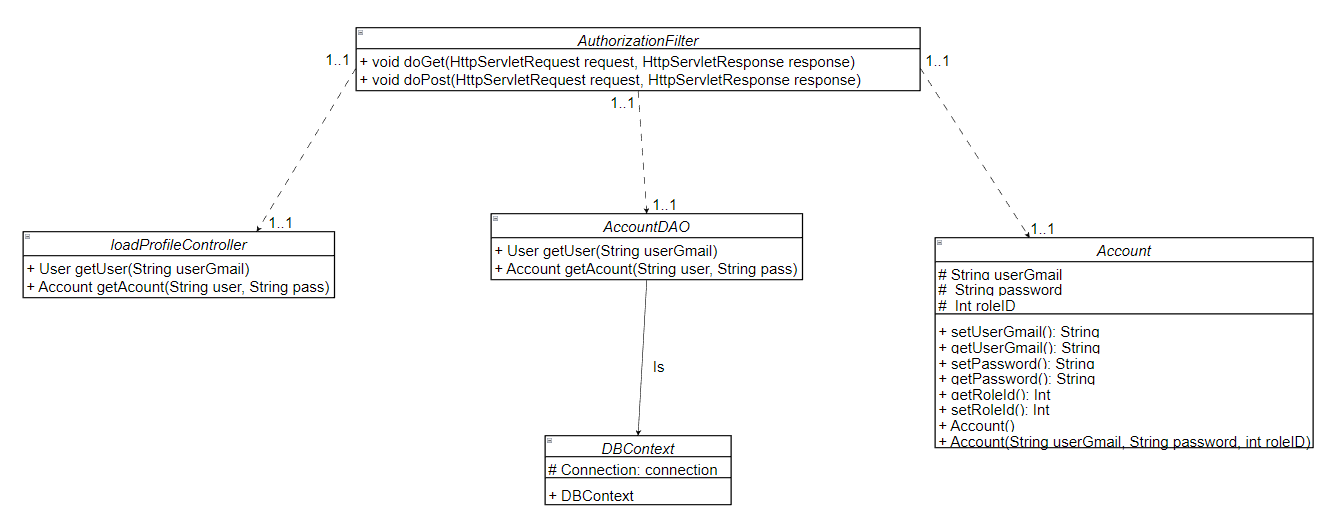
**when user\_gmail = ‘ ‘**

## 

## 

## 4. User Authorization

### a. Class Diagram



### b. Class Specifications

#### 

#### ChangeRoleController

| No | Method | Description |
| --- | --- | --- |
| 01 | Void doGet | *Get account List and roleList with search or without search criteria and forward requests to the View to show account list acquired* |
| 02 | Void doPost | *Update the role of account* |

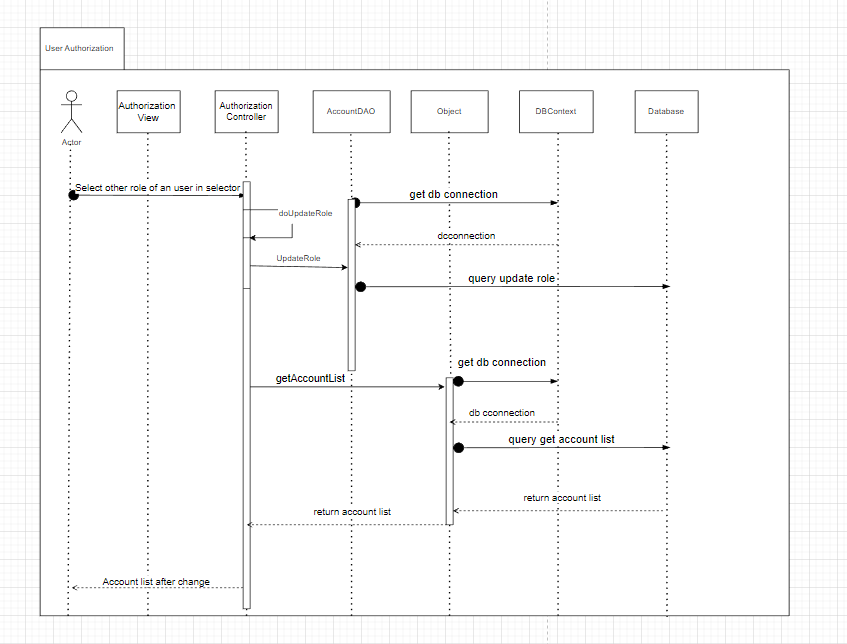
#### 

#### AccountDAO

| No | Method | Description |
| --- | --- | --- |
| 01 | public void changeRole(int roleId, String userGmail) | Update the user has userGmail in parameter and change its roleID to roleId in the parameter |

#### 

### c. Sequence Diagram(s)



### d. Database Queries

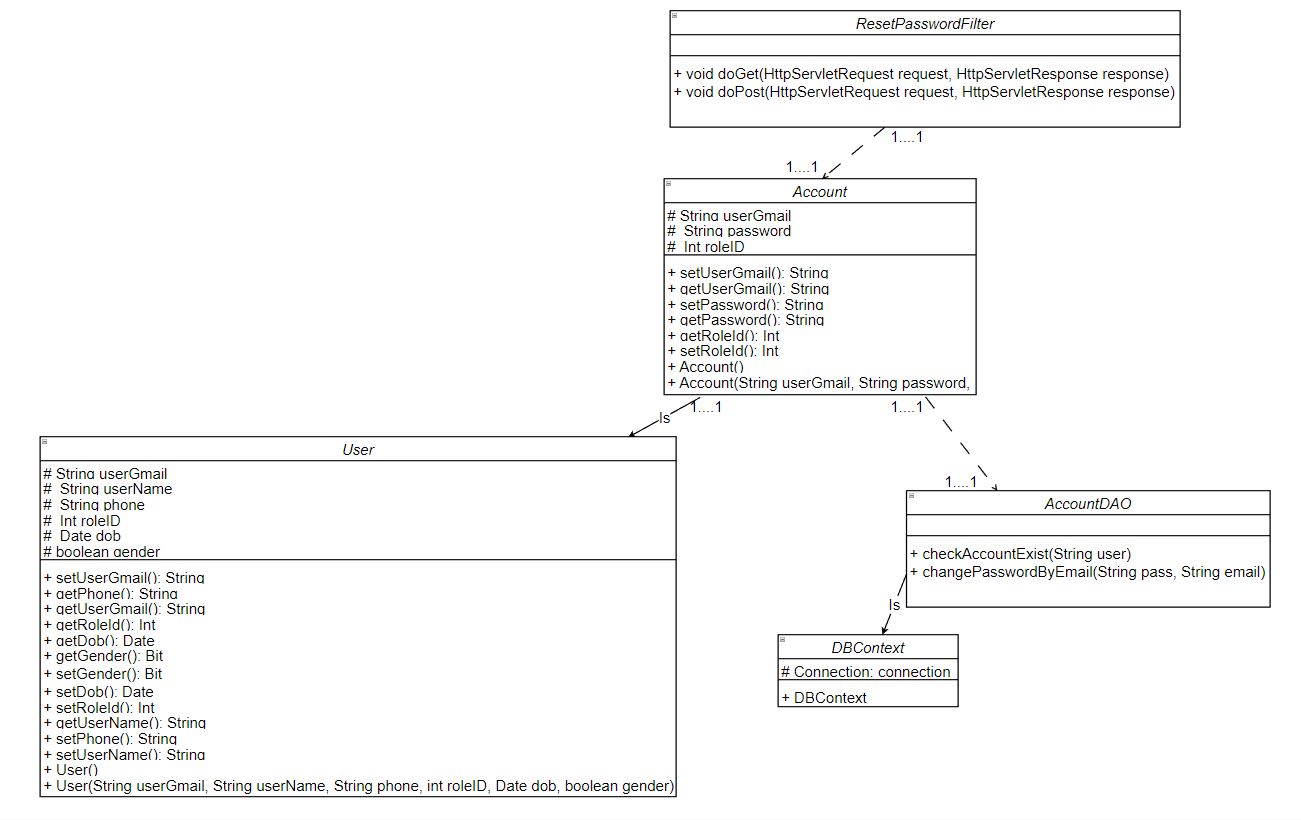
*" update [Account]\n"*

*+ " set role\_id = ?\n"*

*+ " where user\_gmail = ?"*

## 5. Reset Password

### a. Class Diagram



### b. Class Specifications

**ResetPasswordController**

#### 

| No | Method | Description |
| --- | --- | --- |
| 01 | Void doGet | *Return the reset password page to the user and show notification about verify code in email* |
| 02 | Void doPost | *Send a code to user email to reset password.* |

### 

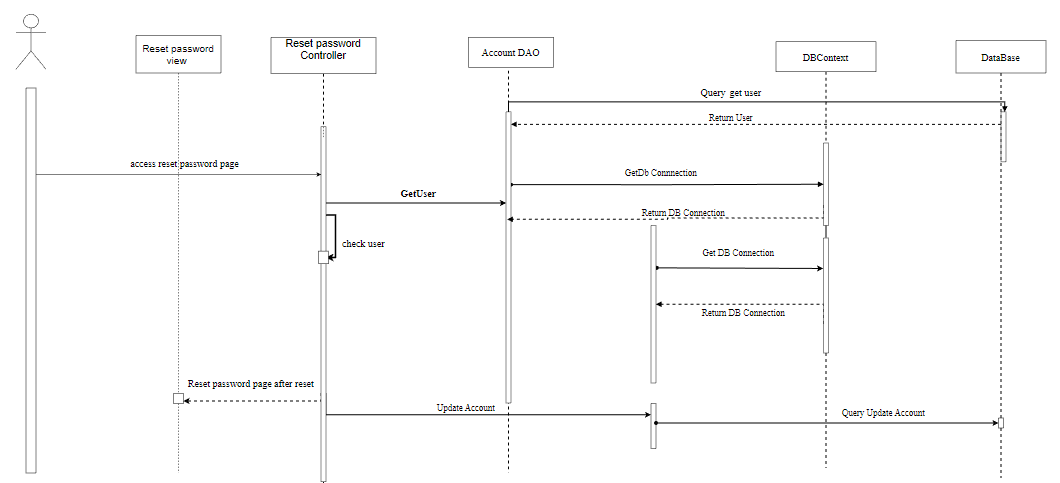
**AccountDAO**

#### 

| No | Method | Description |
| --- | --- | --- |
| 01 | changePasswordByEmail(String pass, String email) | *Update to the database with user\_email and auto change password of this user \_email* |

### 

### c. Sequence Diagram(s)

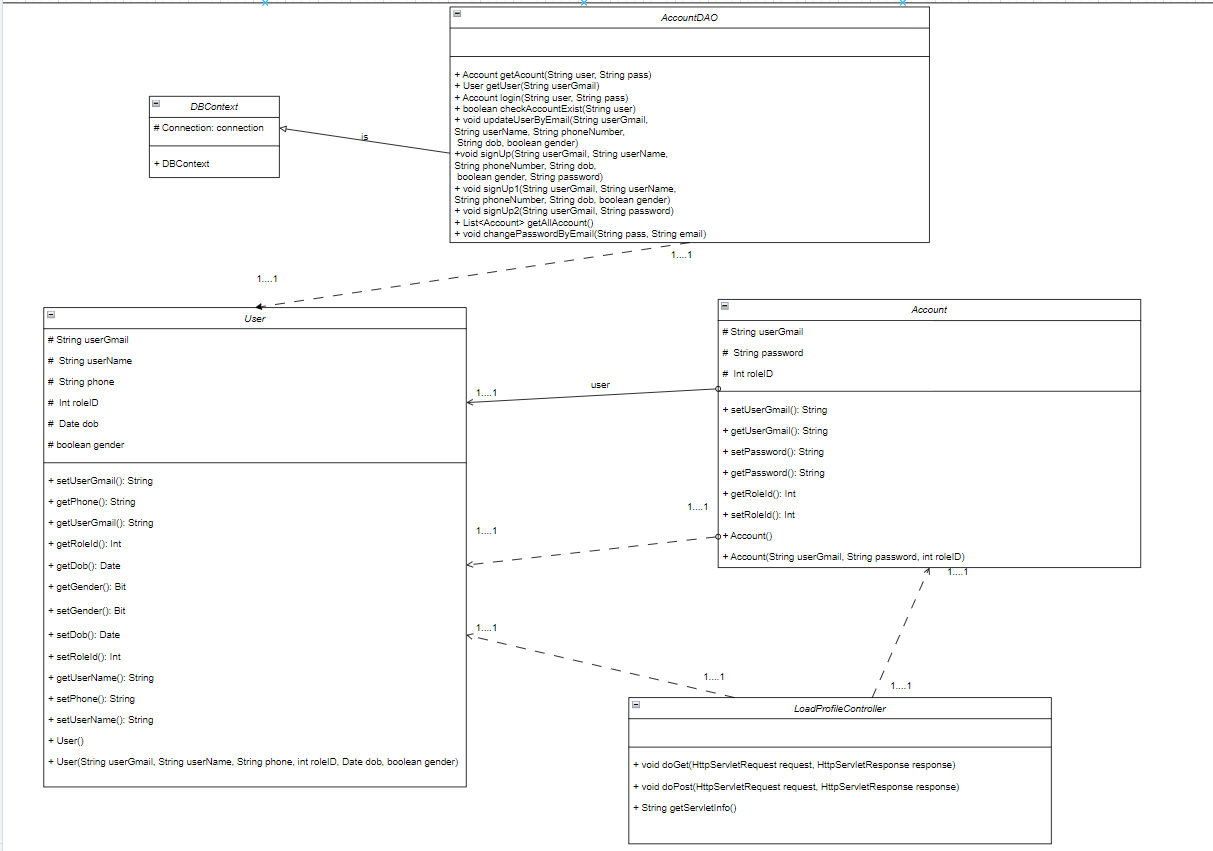


### d. Database Queries

***"update Account set password = ? where user\_gmail = ?"***

## 6. User Profile Setting

### a.Class diagram



***b. Class Specifications***

#### 

| No | Method | Description |
| --- | --- | --- |
| 01 | Void doGet | *Get user information and forward requests to the View to show user information* |
| 02 | Void doPost | *Update the user information with parameter send from View.* |

### 

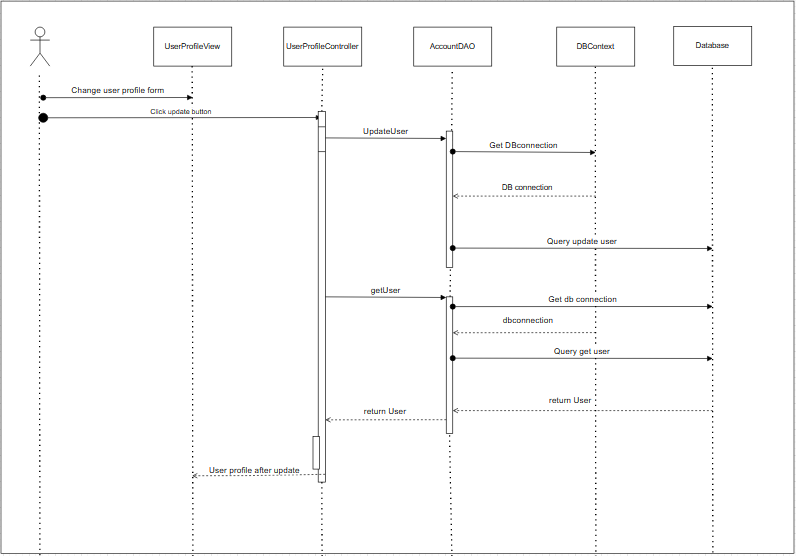
**AccountDAO**

#### 

| No | Method | Description |
| --- | --- | --- |
| 01 | public User getUser(String userGmail) | *Get user with userGmail as parameter* |
| 02 | public void updateUserByEmail(String userGmail, String userName, String phoneNumber, String dob, boolean gender) | *Update user with new information* |

### 

***c.Sequence Diagram***



### d. Database queries

**"select \* from [User] where user\_gmail=?"**

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

**+ "SET user\_name=?,phone=?,dob=?,gender=?\n"**

**+ "WHERE user\_gmail=?";**

## 7. Create Course

### a. Class Diagram

### 

### 

### b. Class Specifications

#### AddCourseController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the add course page to the user* |
| *02* | *Void doPost* | *Add a course with input of user and redirect user to course detail of course just created.* |

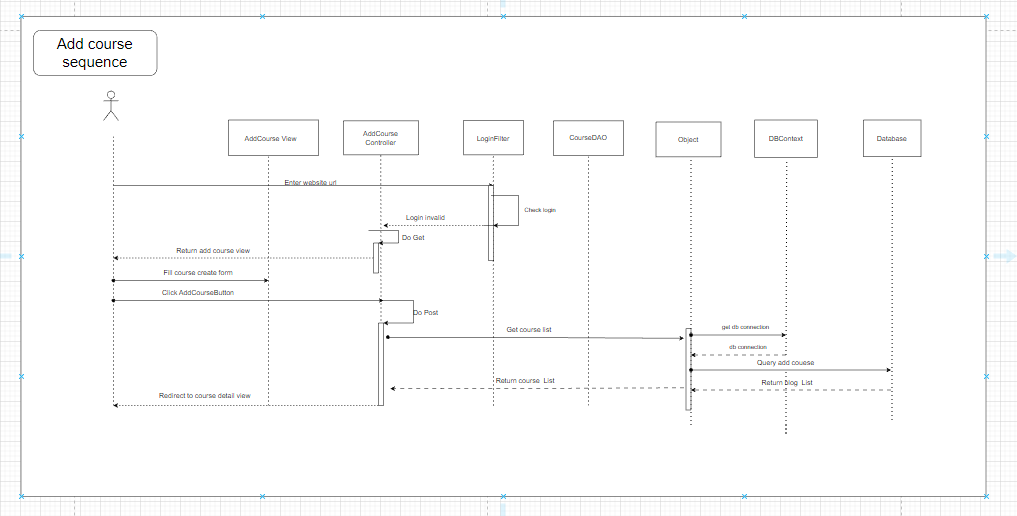
#### CourseDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *add(int CourseID,String courseName, String description, String Courseimage,int category\_id)* | *Add a course to the database with course name and return the course id of that course* |

### 

### c. Sequence Diagram(s)



### d. Database queries

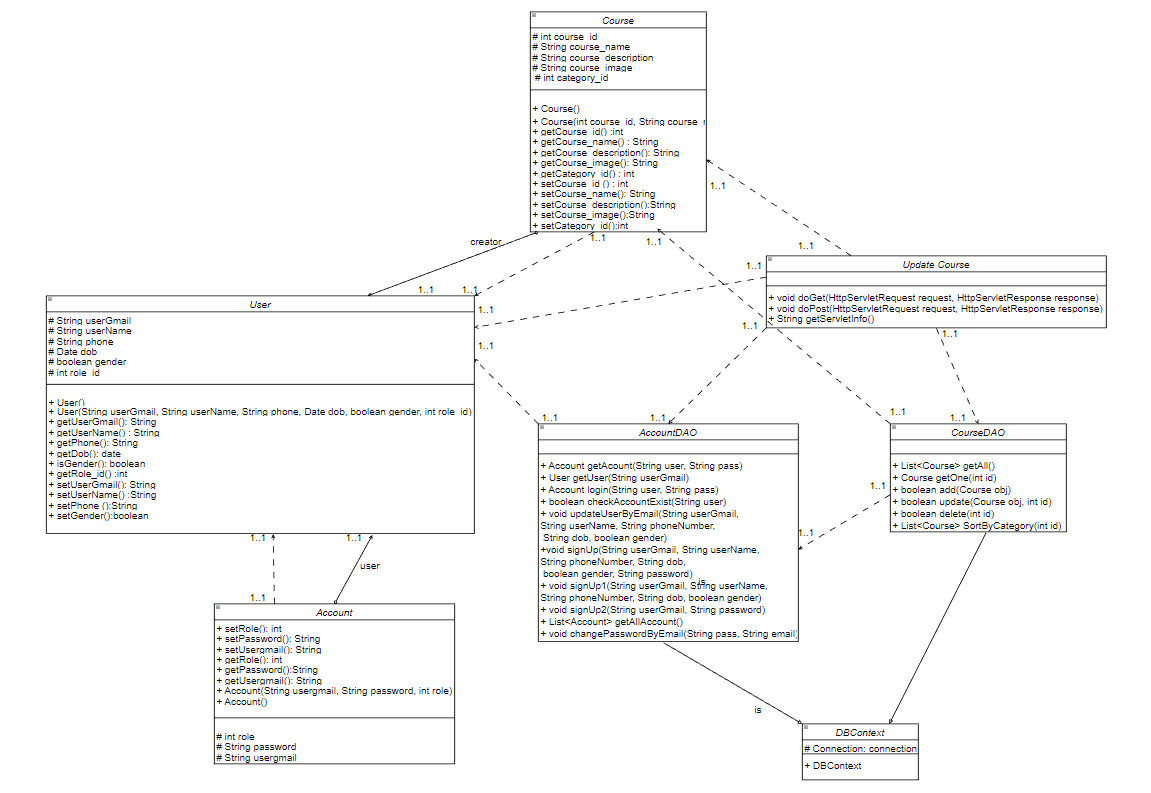
## +) "INSERT INTO Course ([course\_id], course\_name, course\_description, course\_image, category\_id)\n"

## + "VALUES(? , ?, ?, ?,?)";

## 

## 8. Update Course

### a. Class Diagram



### b. Class Specifications

#### UpdateCourseController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the update course page to the user* |
| *02* | *Void doPost* | *update a course with input of user and redirect user to course detail of course just updated* |

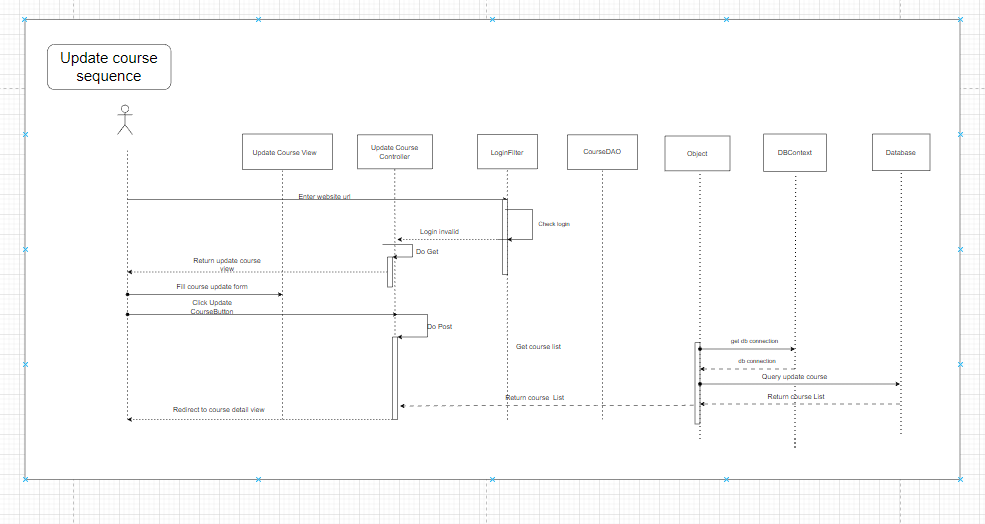
#### CourseDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *update(int CourseID,String courseName, String description, String Courseimage,int category\_id,int idCourseUpdate)* | *Update a course to the database with course details and return the course id of that course* |

### 

### c. Sequence Diagram(s)

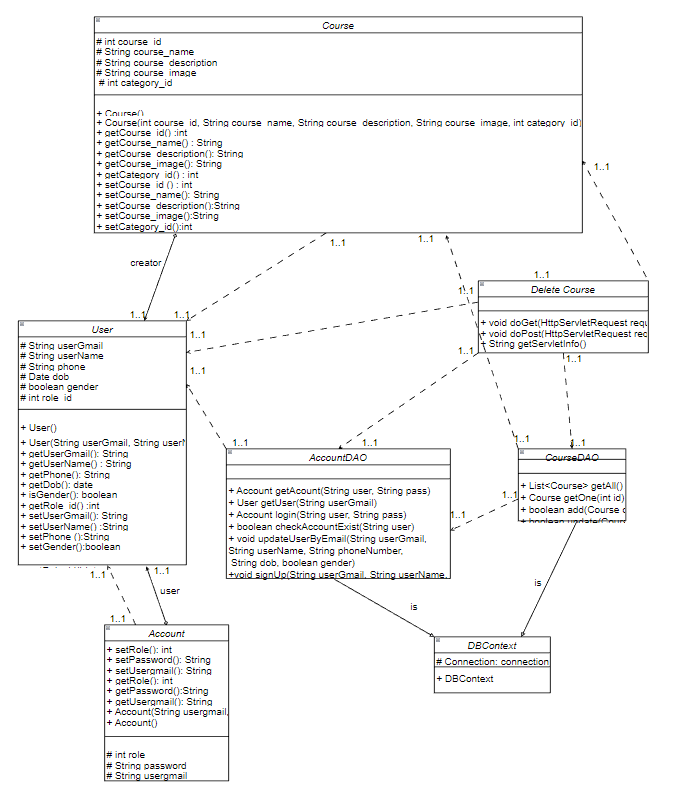


### d. Database queries

**+) "update Course SET course\_id = ?, course\_name = ?, [course\_description] = ?,course\_image = ?,category\_id = ? where course\_id = ?";**

## 9. Delete Course

### a. Class Diagram



### b. Class Specifications

#### DeleteCourseController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the delete course page to the user* |
| *02* | *Void doPost* | *delete a course with input of user and return user to course page* |

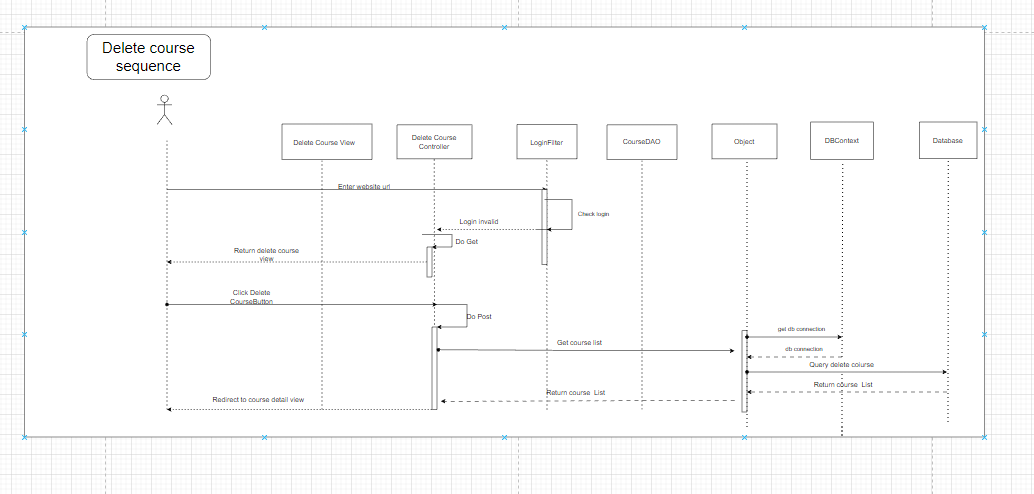
#### CourseDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *delete(int id\_course)* | *Delete a course to the database with course details* |

### 

### c. Sequence Diagram(s)

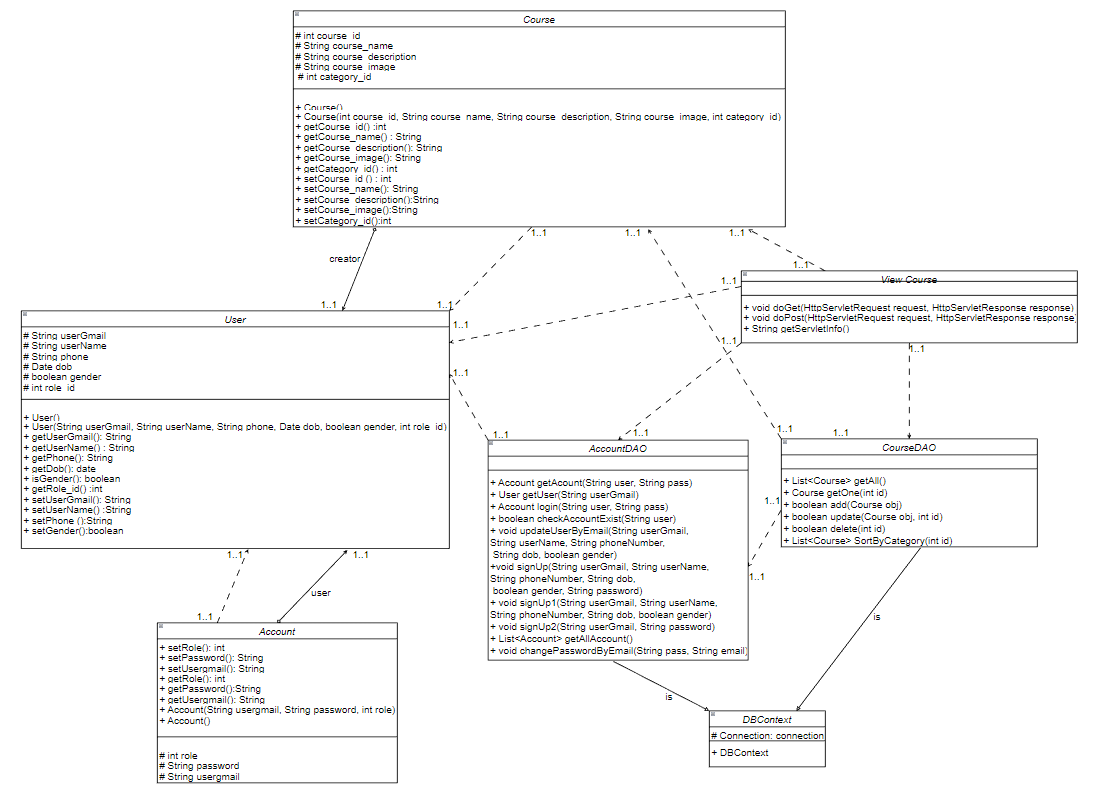


### d. Database queries

## +) " DELETE from [Course] where course\_id = ? ";

## 10. View Course

### a. Class Diagram



### b. Class Specifications

#### ViewCourseController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return course detail to course page* |

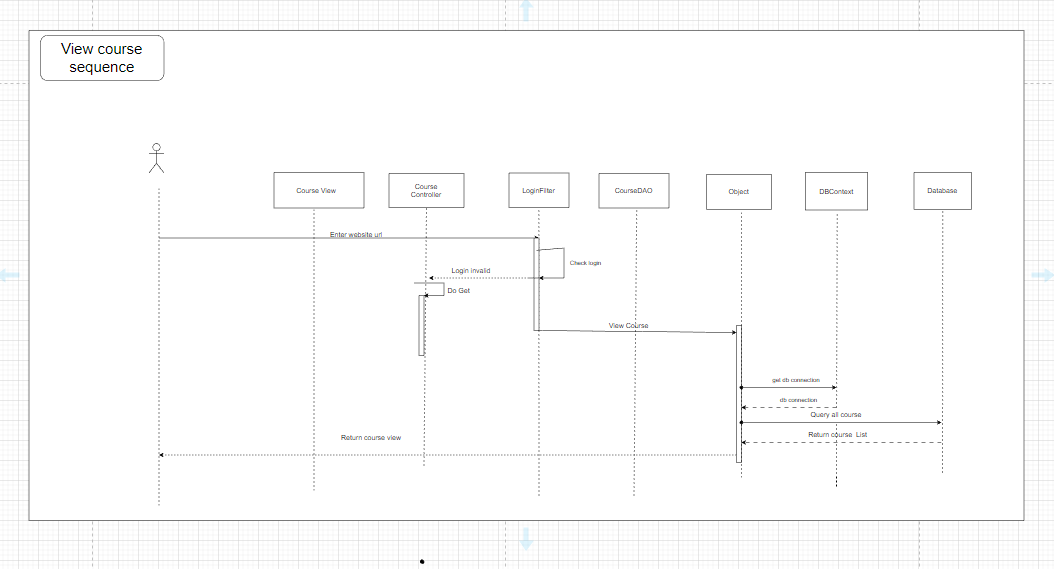
CourseDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *List<Course> getAll()* | *View all a course to the database with course details* |
| *02* | *Course getOne(int id)* | *Return the course id of that course* |

### 

### c. Sequence Diagram(s)

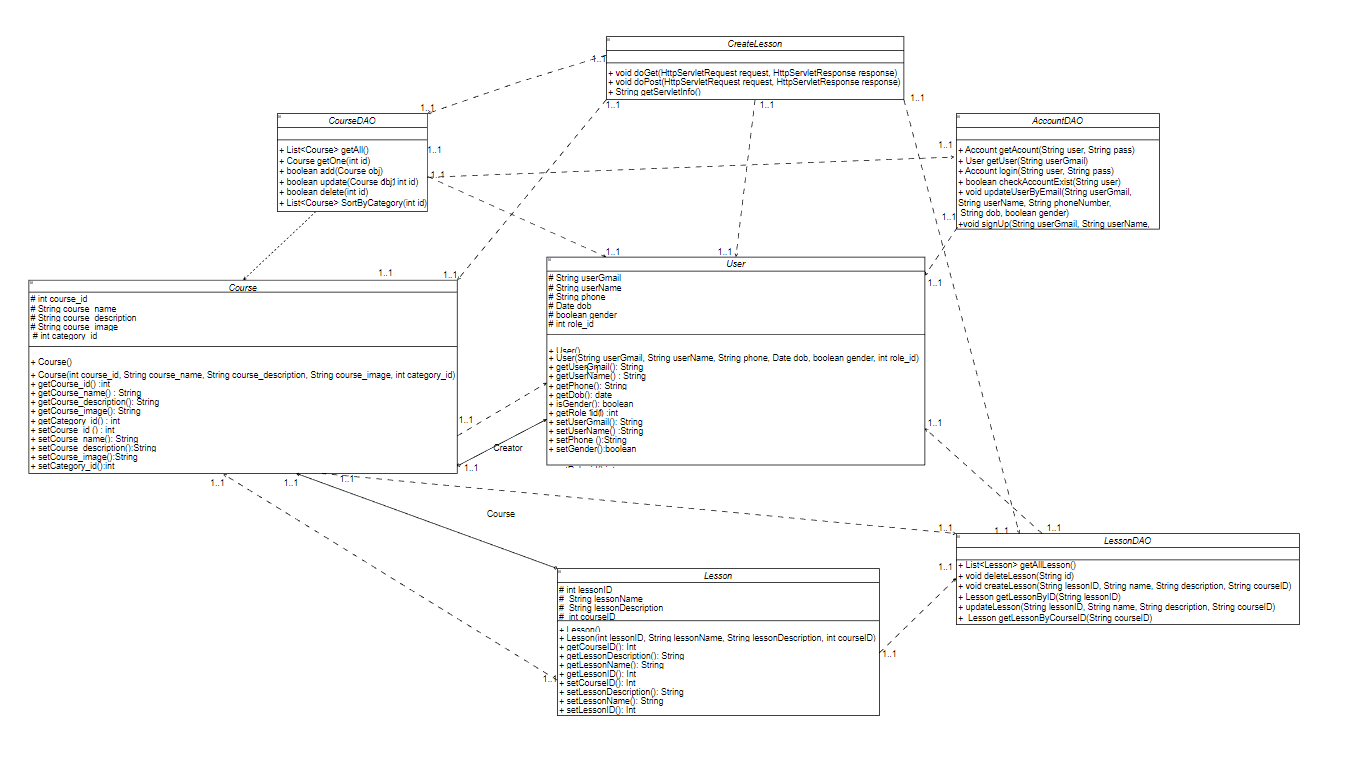


### d. Database queries

**+) "SELECT \* FROM Course"**

## 11. Create Lesson

### a. Class Diagram



### b. Class Specifications

#### CreateLesson

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the add lesson page to the user* |
| *02* | *Void doPost* | *Add a lesson with input of user and redirect user to lesson detail of lesson just created.* |

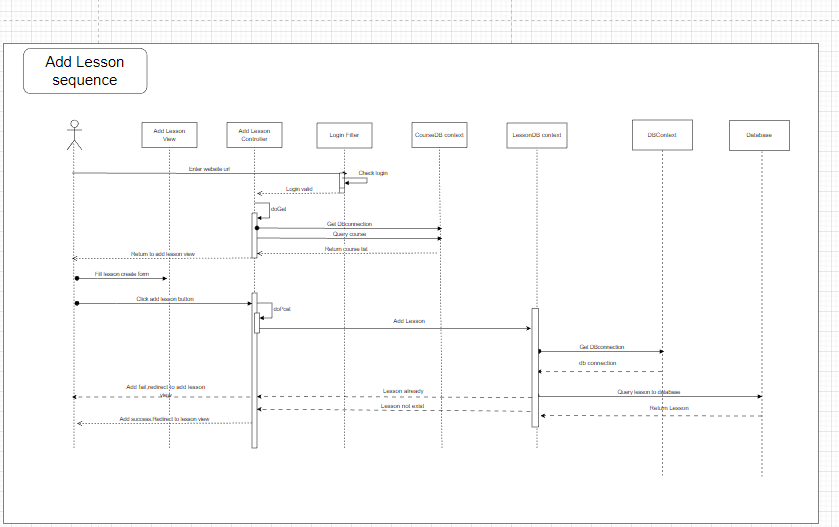
#### LessonDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *createLesson(String lessonID, String name, String description, String courseID)* | *Add a lesson to the database with lesson details and return the lesson id of that lesson* |

### 

### c. Sequence Diagram(s)



### d. Database queries

**+) "insert into \n"**

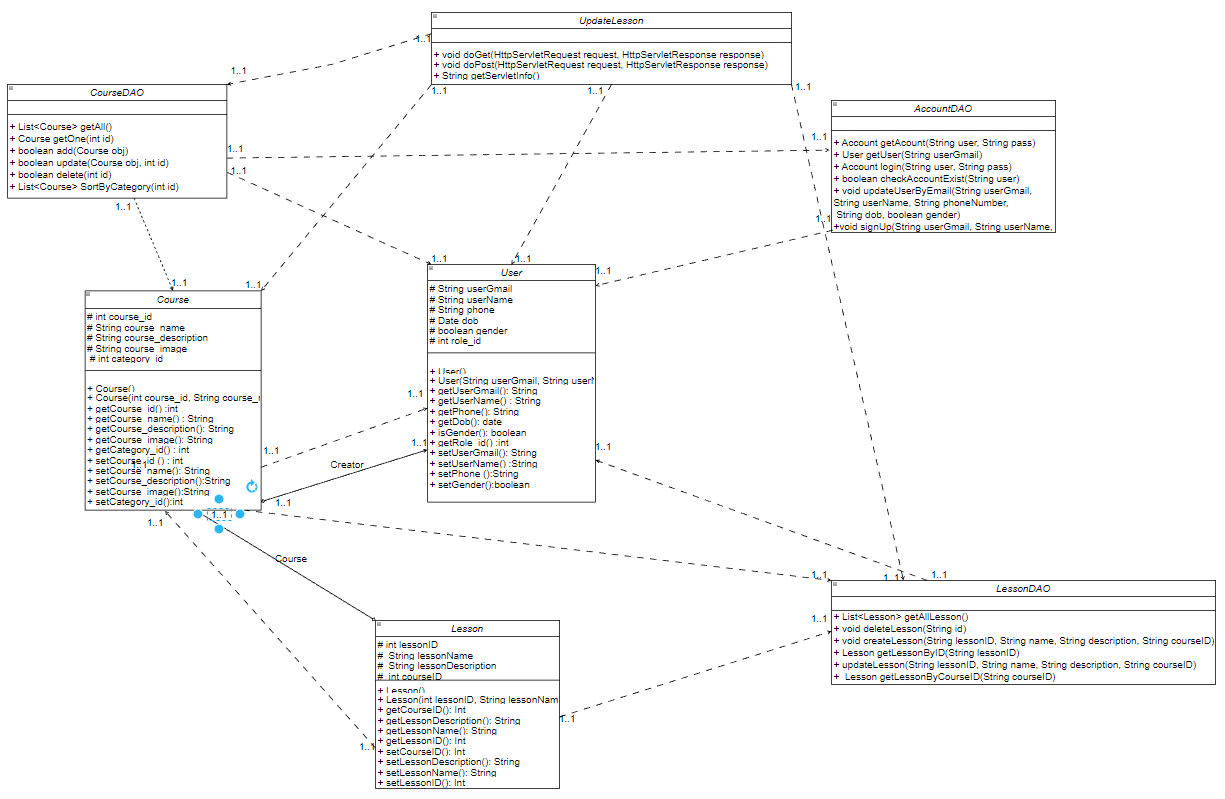
**+ "Lesson\n"**

**+ "values (?,?,?,?)";**

## 

## 12. Update Lesson

### a. Class Diagram



### b. Class Specifications

#### UpdateLesson

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the update lesson page to the user* |
| *02* | *Void doPost* | *update a lesson with input of user and redirect user to lesson detail of lesson just updated* |

#### 

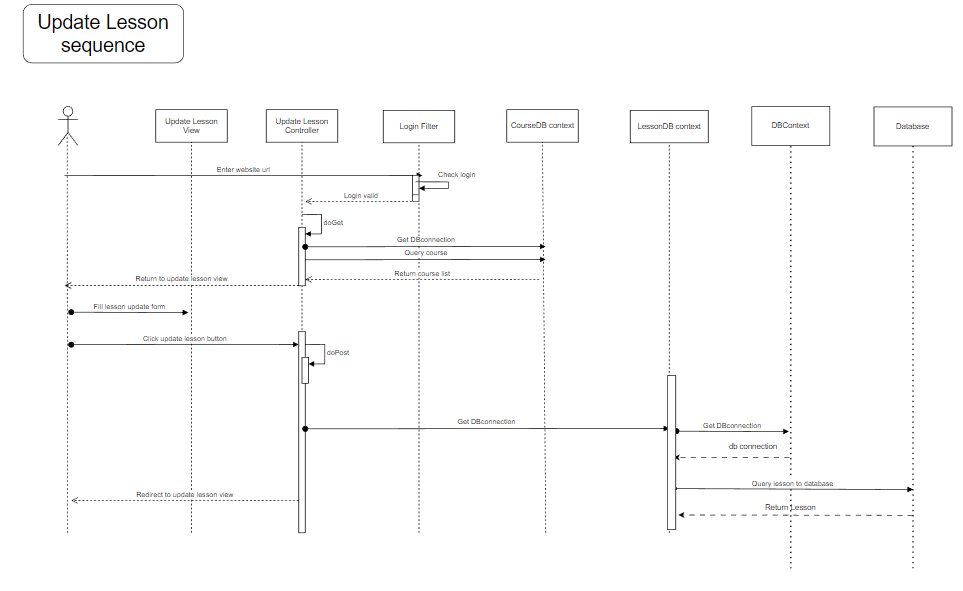
#### LessonDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *updateLesson(String lessonID, String name, String description, String courseID)* | *Update a lesson to the database with lesson details and return the lesson page* |

### 

### c. Sequence Diagram(s)



### d. Database queries

**+) "update Lesson\n"**

**+ "set lesson\_name = ?,\n"**

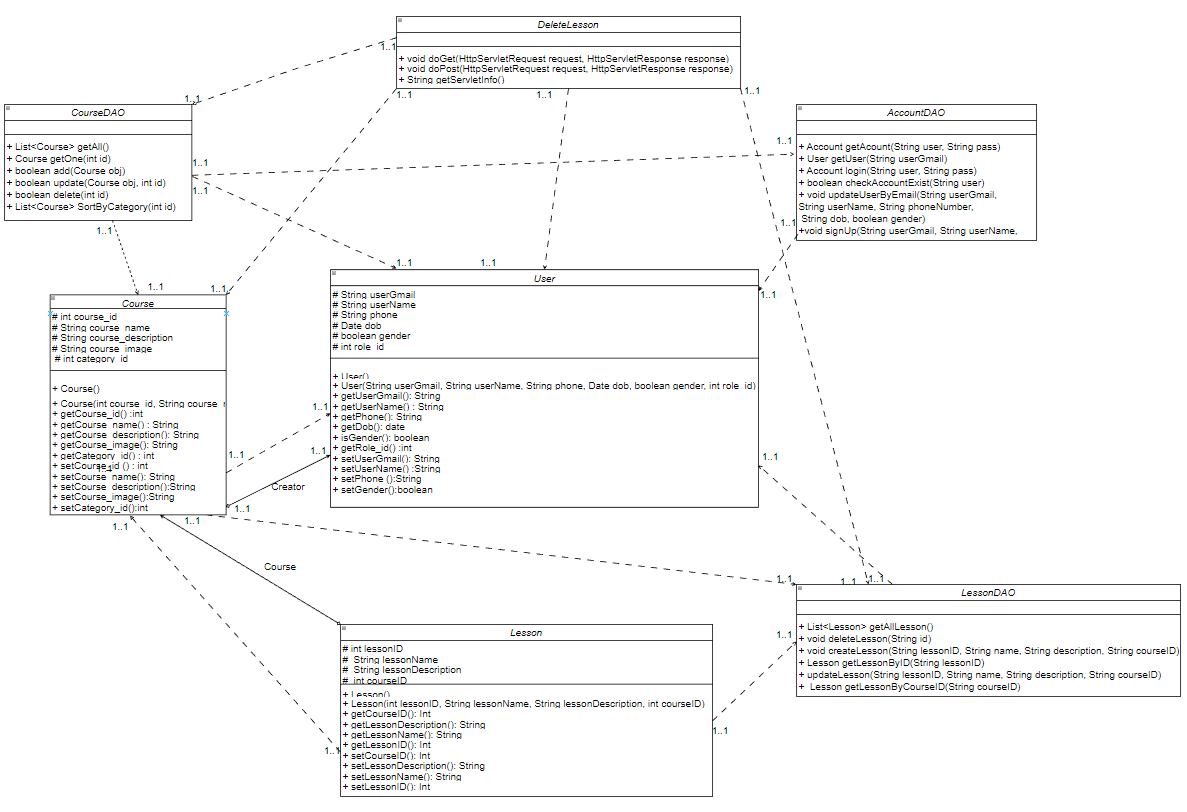
**+ "lesson\_description = ?,\n"**

**+ "course\_id= ?\n"**

**+ "where lesson\_id = ?"**

## 13. Delete Lesson

### a. Class Diagram



### b. Class Specifications

#### DeleteLesson

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the lesson page to the user* |
| *02* | *Void doPost* | *delete a lesson with input of user and return user to lesson page* |

#### 

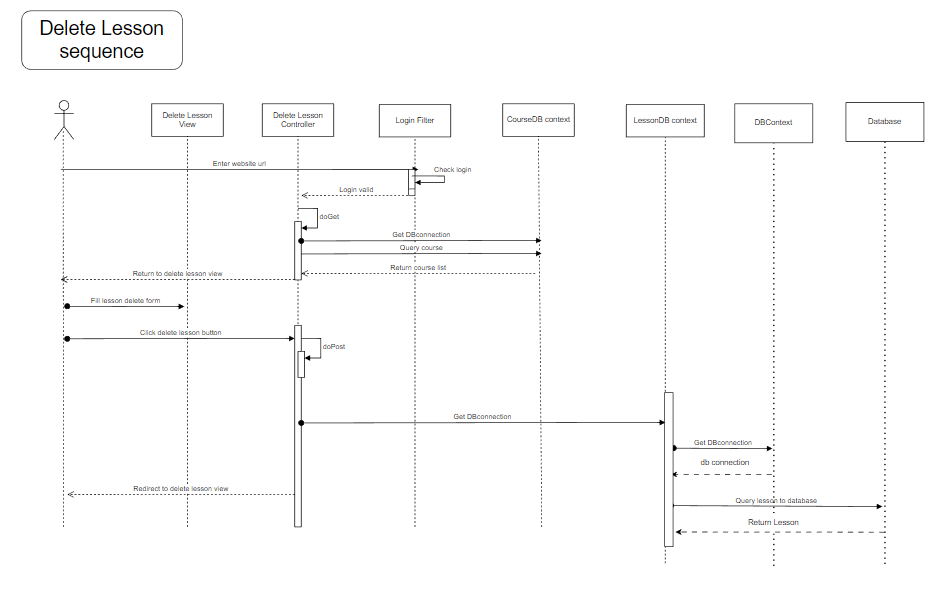
#### LessonDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *deleteLesson(String id)* | *Delete a lesson to the database with lesson details* |

### 

### c. Sequence Diagram(s)



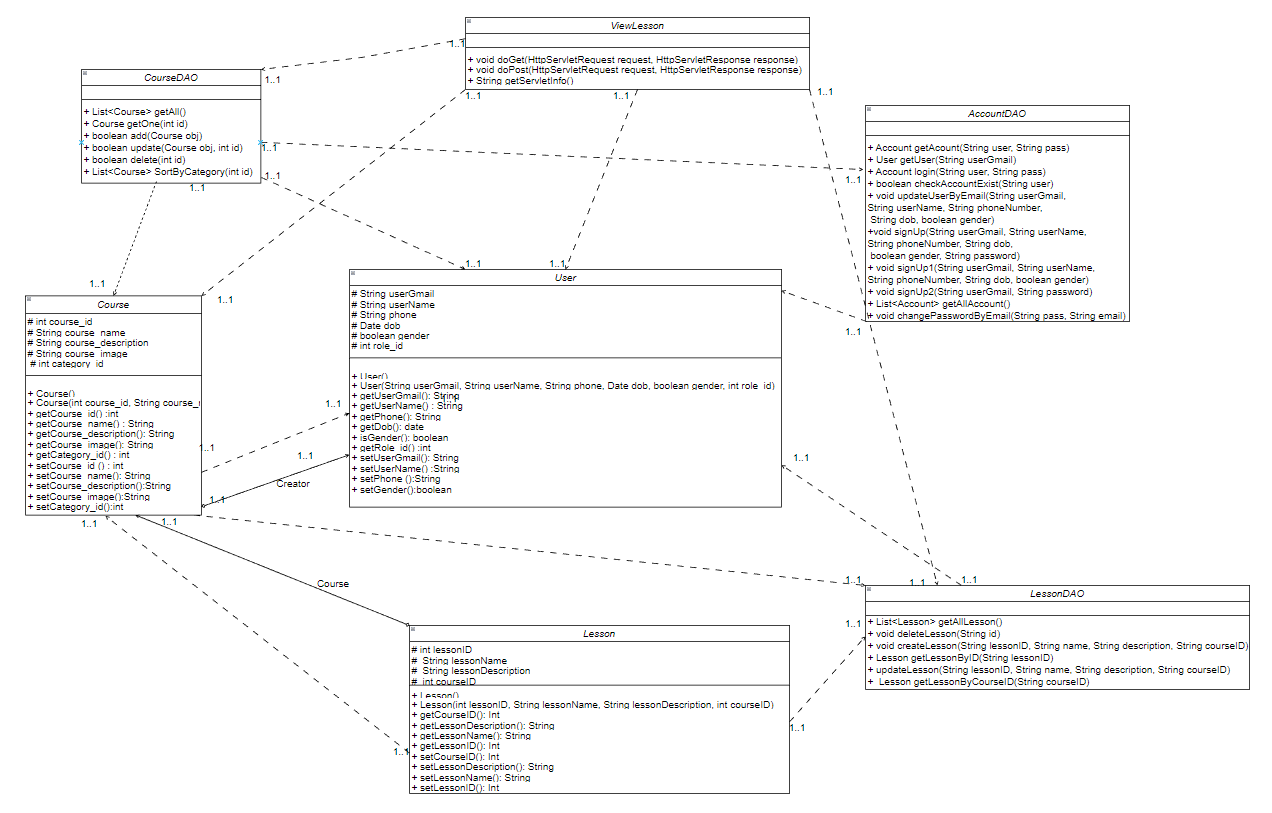
### d. Database queries

**+) "delete from Lesson\n"**

**+ "where lesson\_id = ?"**

## 14. View Lesson

### a. Class Diagram



### b. Class Specifications

#### ManageLesson

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the lesson page to the user* |

#### 

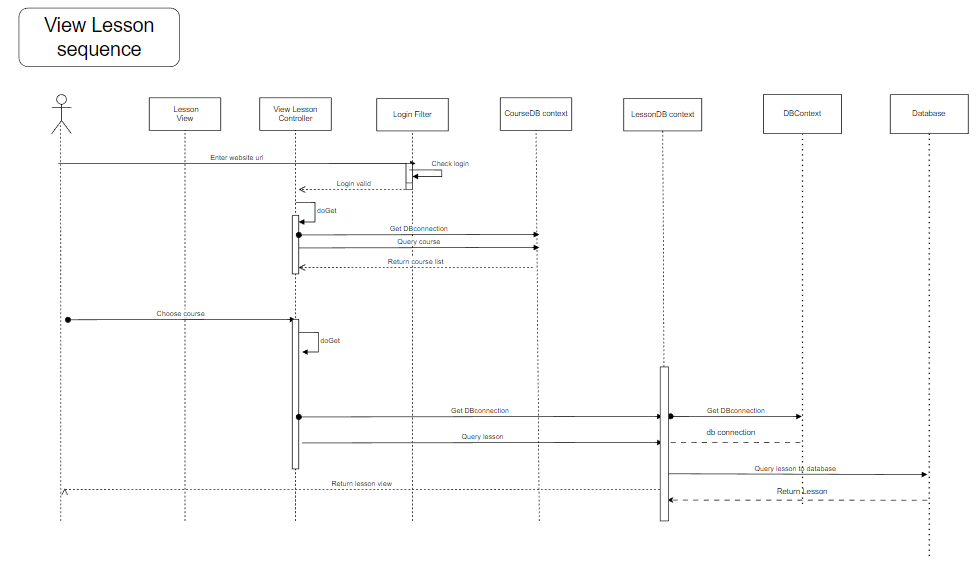
#### LessonDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *List<Lesson> getAllLesson()* | *View all a lesson to the database with lesson details and return the lesson id of that lesson* |
| *02* | *List<Lesson> getAllLessonByCourseID()* | *View all lesson of a course by course id* |
| *03* | *Lesson getLessonByID(String lessonID)* | *view one lesson by lesson id* |
| *04* | *Lesson searchLessonByName(String txtSearch)* | *view lesson search by name* |

### 

### c. Sequence Diagram(s)

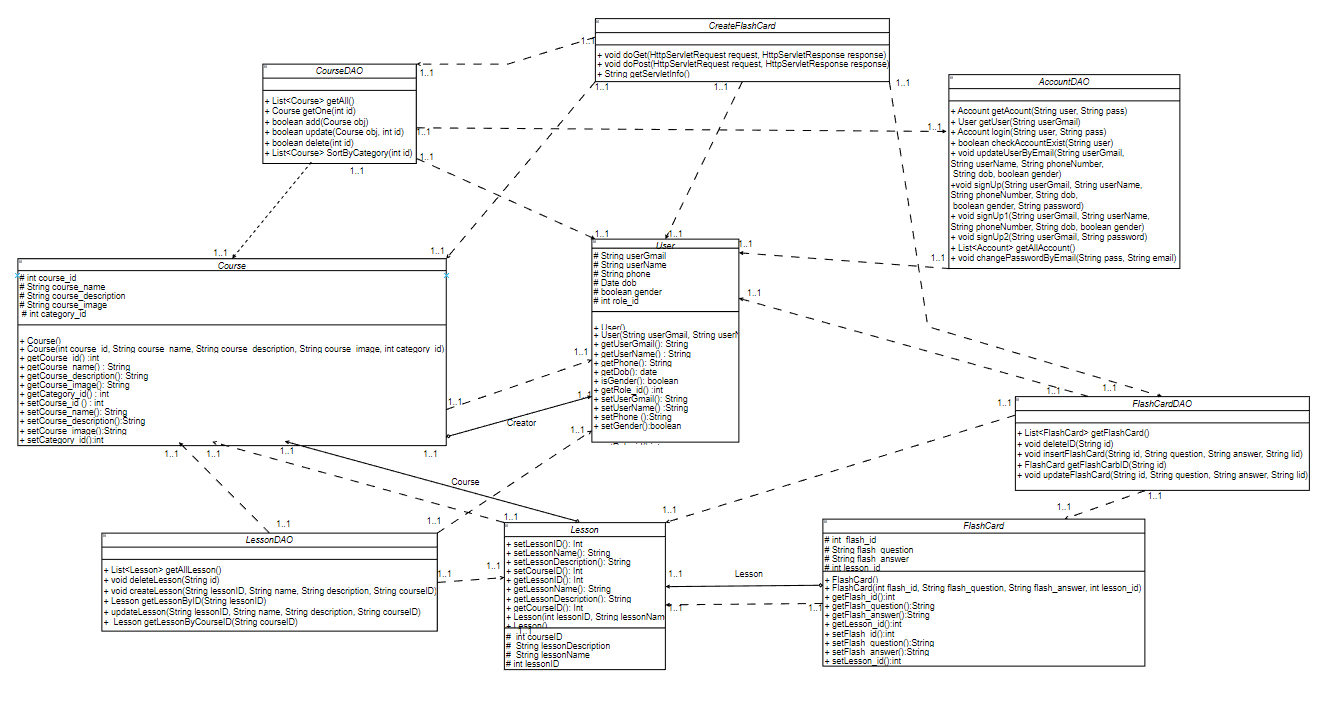


### d. Database queries

**+) "select \* from Lesson"**

## 15. Create Flashcard

### a. Class Diagram



### b. Class Specifications

#### CreateFlashCard

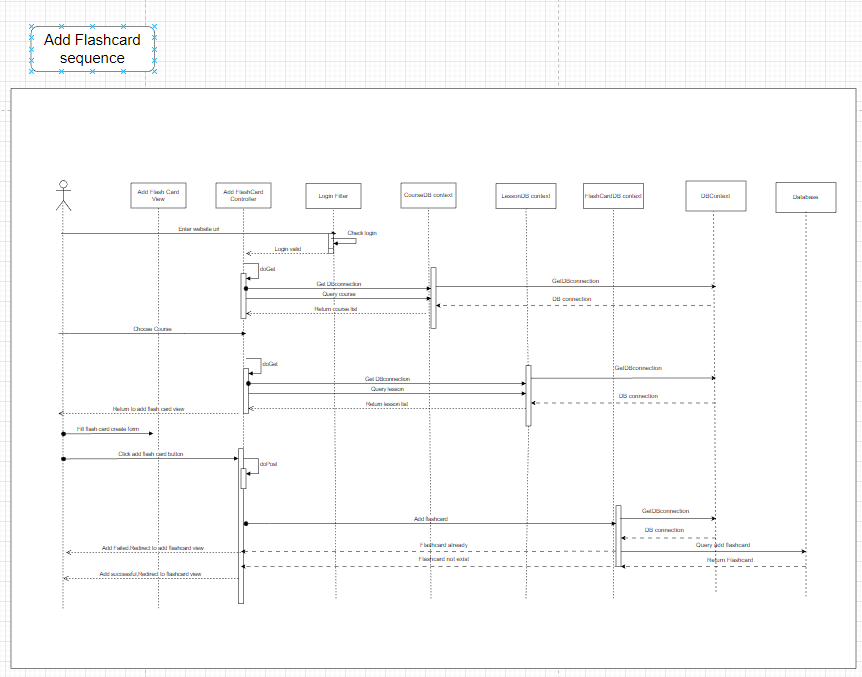
| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the flashcard page to the user* |
| *02* | *Void doPost* | *Create a flashcard with input of user and return user to flashcard detail of lesson just created.* |

#### FlashcardDAO ***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *insertFlashCard(String id, String question, String answer, String lid)* | *Add a flashcard to the database with lesson id and return flashcard manage page* |

#### 

### c. Sequence Diagram(s)

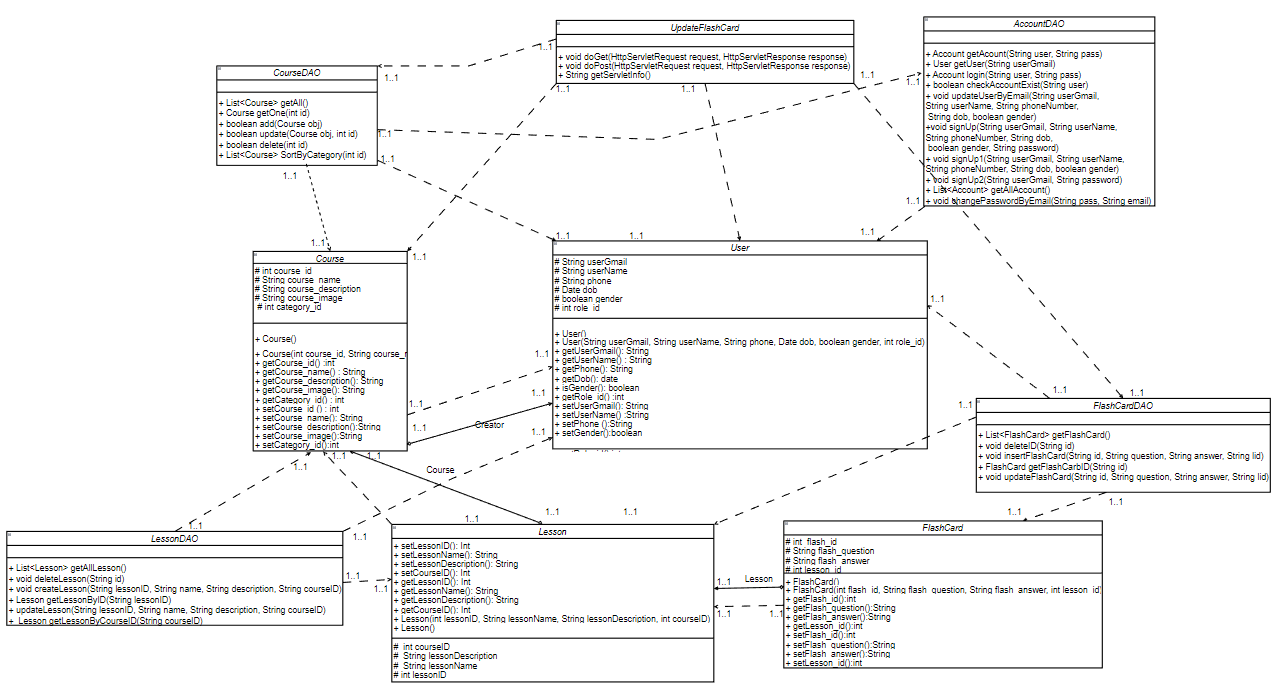


### d. Database queries

+) "insert into Flashcard values (?,?,?,?)"

## 16. Update Flashcard

### a. Class Diagram



### b. Class Specifications

#### UpdateFlashCard

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the lesson page to the user* |
| *02* | *Void doPost* | *Update a flashcard to the database with flashcard details and return the flashcard id of that lesson* |

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

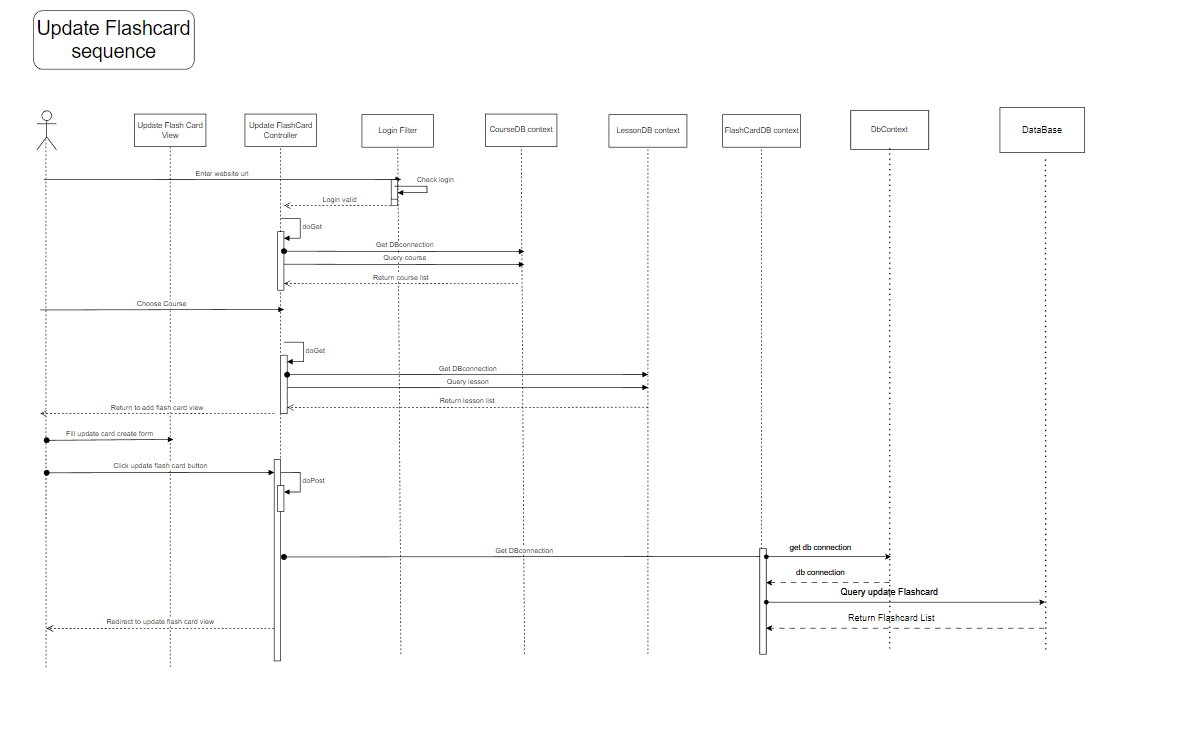
#### FlashcardDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *updateFlashCard(String id, String question, String answer, String lid)* | *Update a flashcard to the database with lesson id and return flashcard manage page* |

#### 

### c. Sequence Diagram(s)



### d. Database queries

**+) "update Flashcard\n"**

**+ "set flash\_question = ?,\n"**

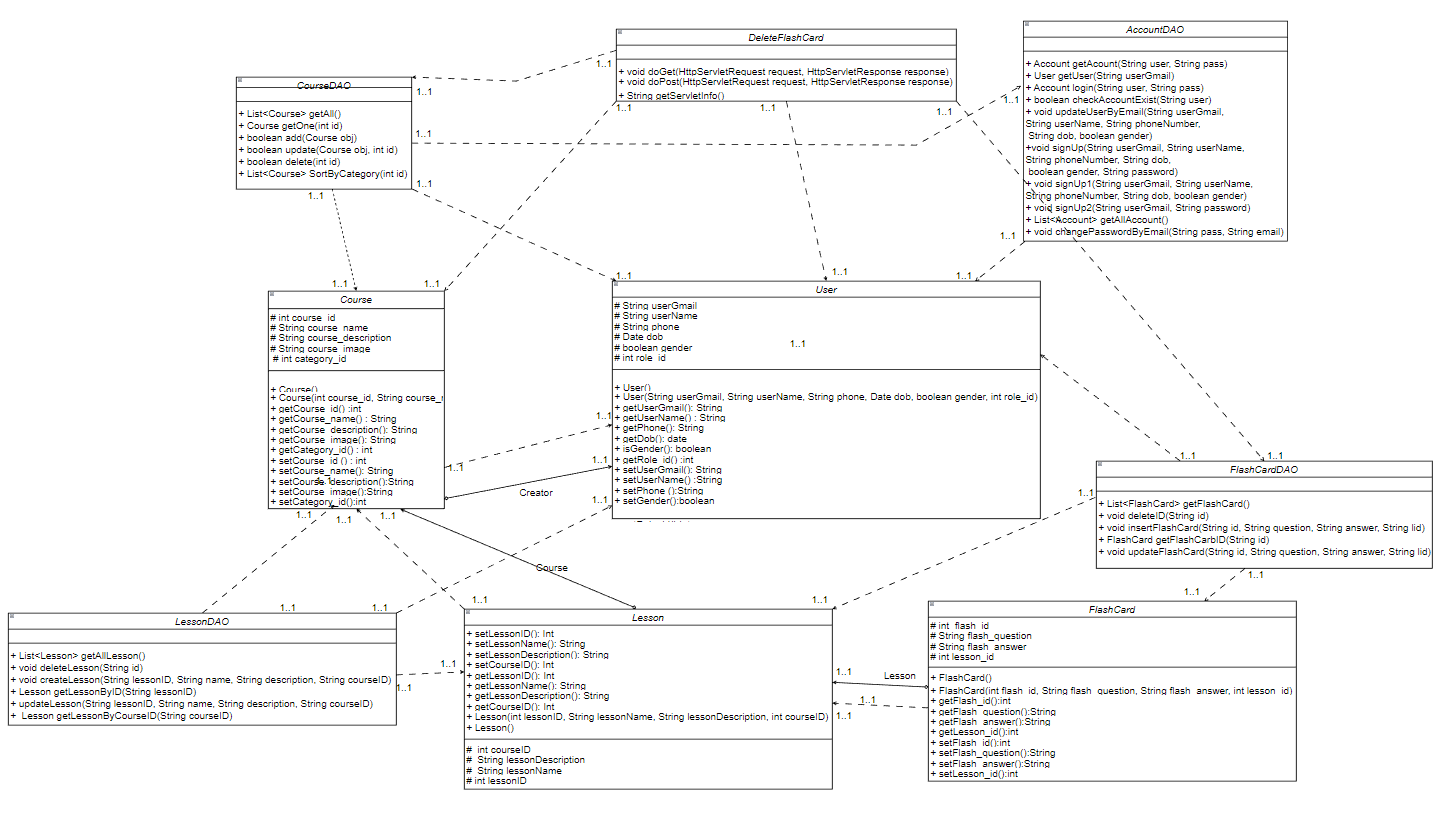
**+ "flash\_answer = ?,\n"**

**+ "lesson\_id = ?\n"**

**+ "where flash\_id = ?"**

## 17. Delete Flashcard

### a. Class Diagram



### b. Class Specifications

#### DeleteFlashCard

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the flashcard page to the user* |
| *02* | *Void doPost* | *delete a flashcard with input of user and return user to flashcard page* |

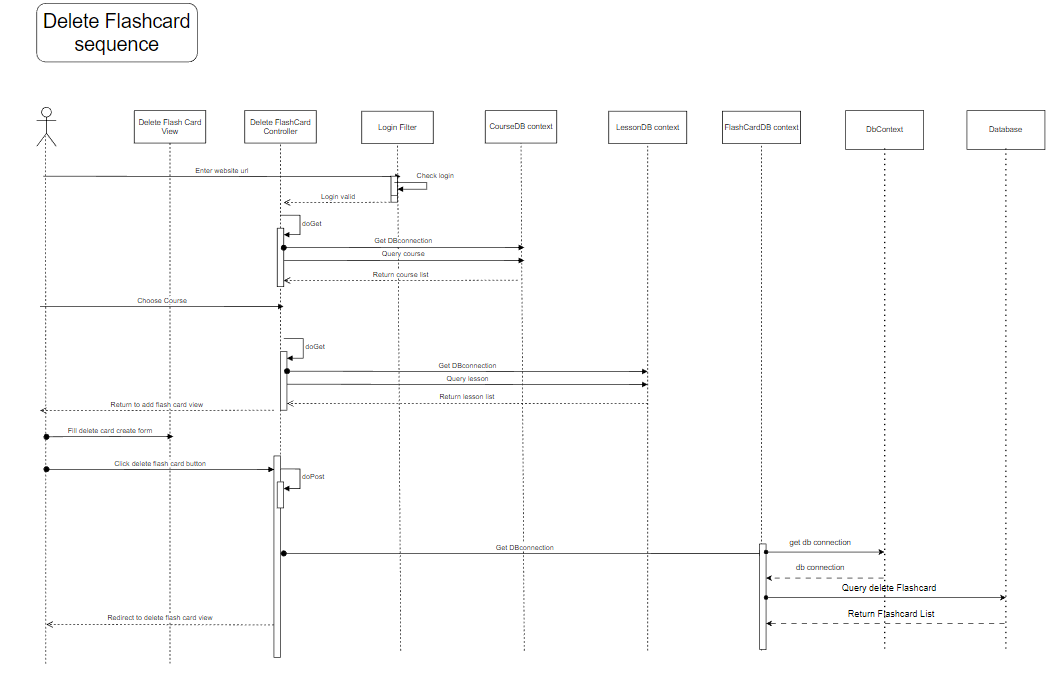
#### FlashcardDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *deleteID(String ID)* | *Delete a flashcard to the database in lesson and return flashcard manage page* |

#### 

### c. Sequence Diagram(s)



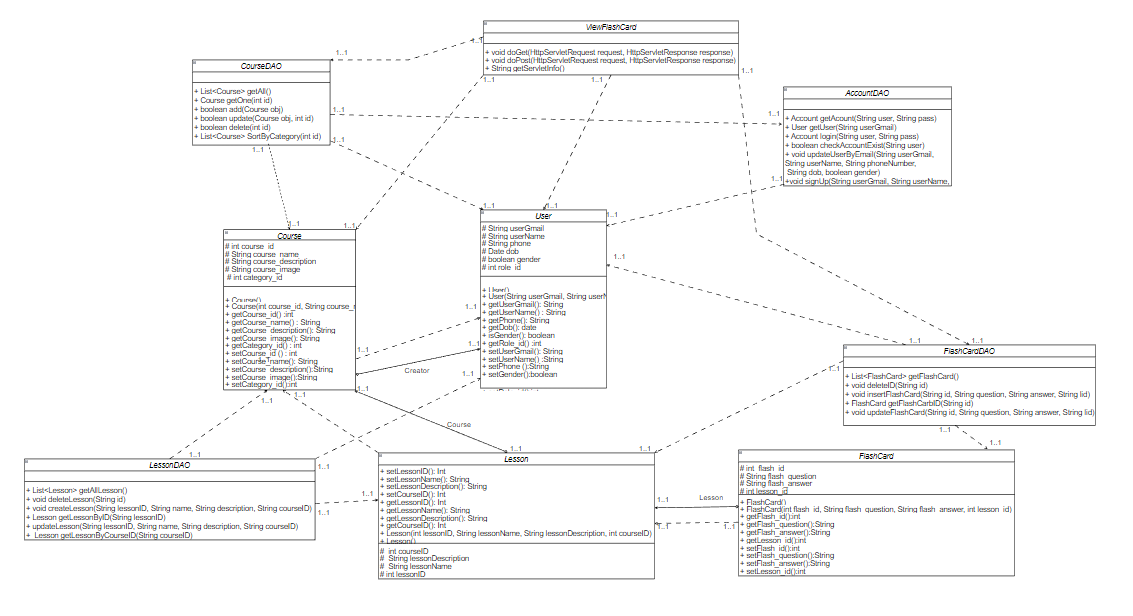
### d. Database queries

**+) "Delete from Flashcard\n"**

**+ "where flash\_id=?"**

## 18. View Flashcard

### a. Class Diagram



### b. Class Specifications

#### ViewFlashCard

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the flashcard page to the user* |

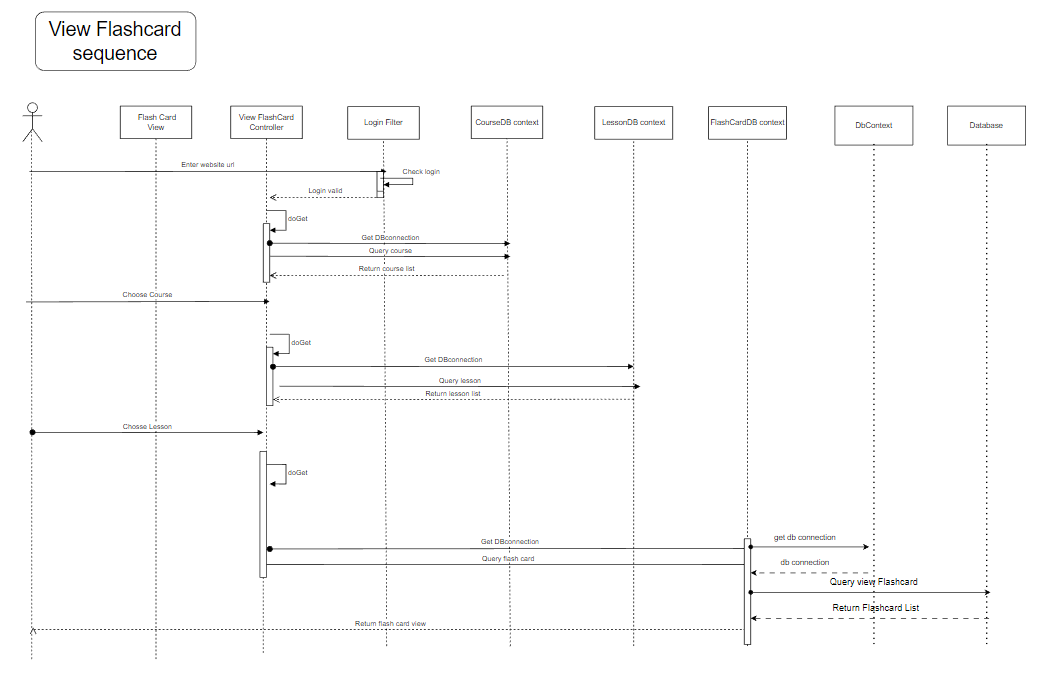
#### FlashcardDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *List<FlashCard> getFlashCard()* | *View all a flashcard to the database with flashcard details* |

#### 

### c. Sequence Diagram(s)



### d. Database queries

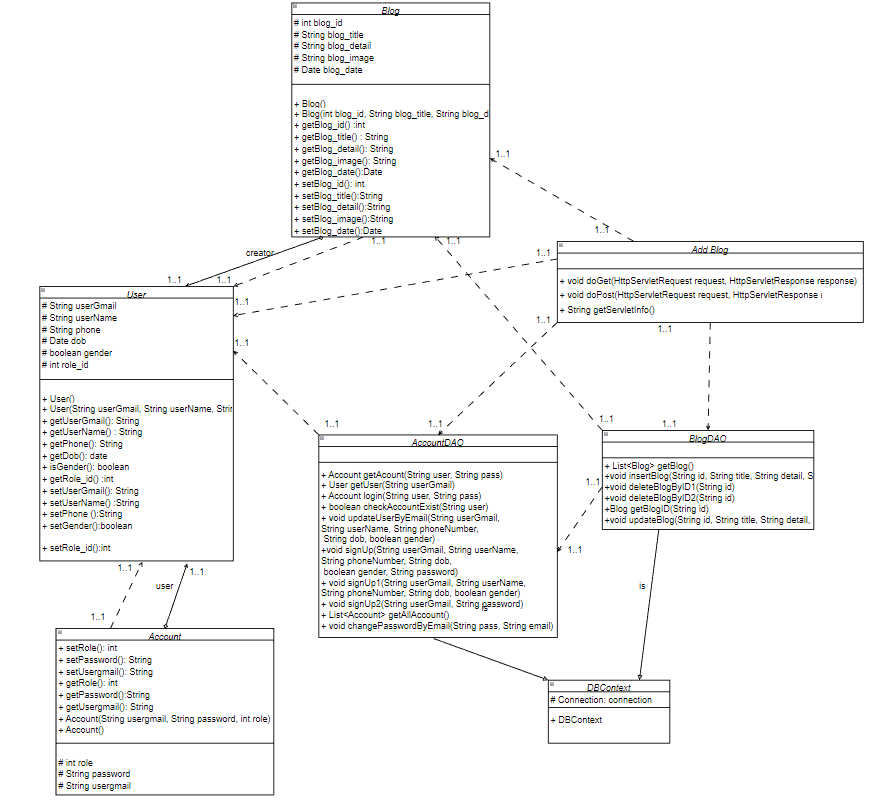
**+) "select \* from Flashcard\n"**

**+ "where flash\_id = ?"**

**+) "Select\*from Flashcard"**

## 19. Create Blog

### a. Class Diagram



### b. Class Specifications

#### AddBlog

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the blog page to the user* |
| *02* | *Void doPost* | *Create a blog with input of user and return user to blog detail of blog just created.* |

#### 

#### 

#### 

#### 

#### 

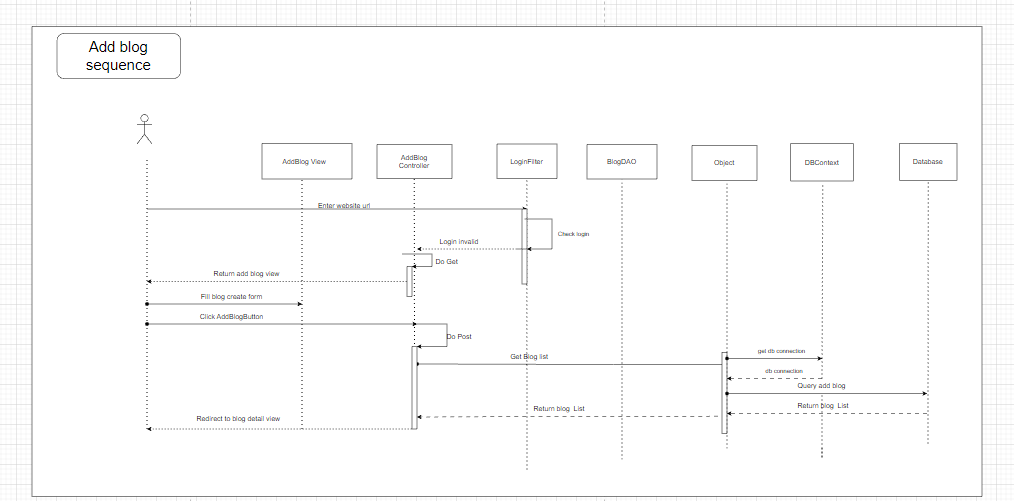
#### BlogDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *void insertBlog(String id, String title, String detail, String image, String date)* | *Add a blog to the database with and return blog manage page* |

#### 

### c. Sequence Diagram(s)

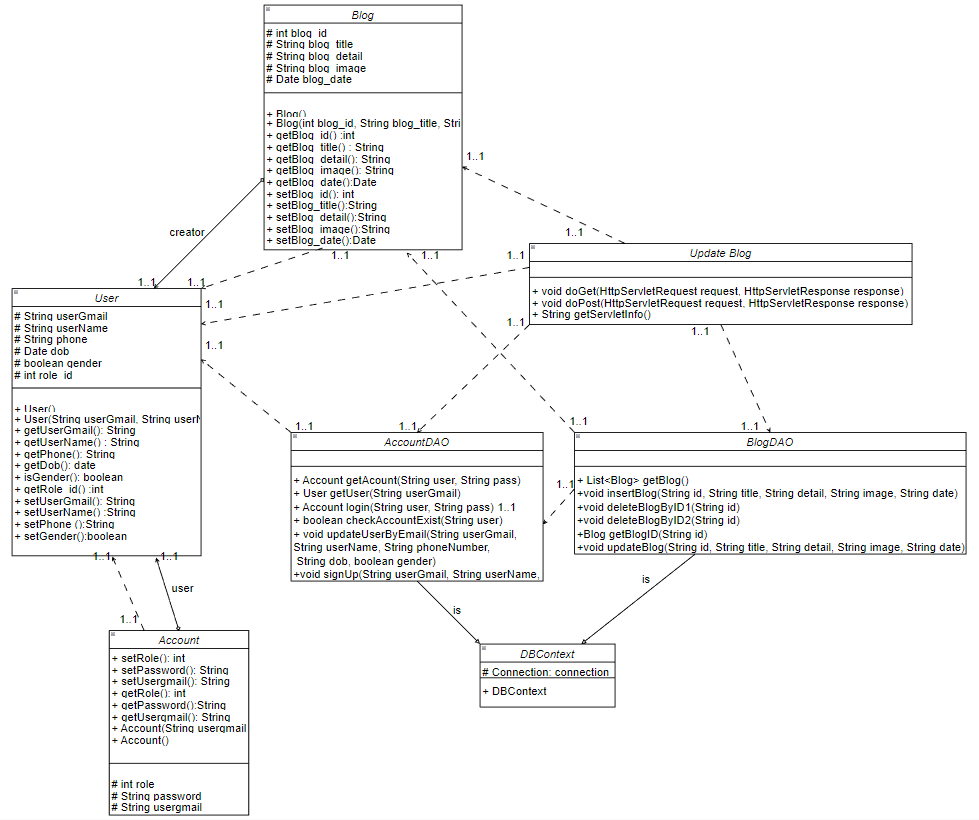


### d. Database queries

+) "insert into Blog values (?,?,?,?,?)"

## 20. Update Blog

### a. Class Diagram



### b. Class Specifications

#### UpdateBlog

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the blog page to the user* |
| *02* | *Void doPost* | *Update a blog to the database with blog details and return the blog page* |

#### 

#### 

#### 

#### 

#### 

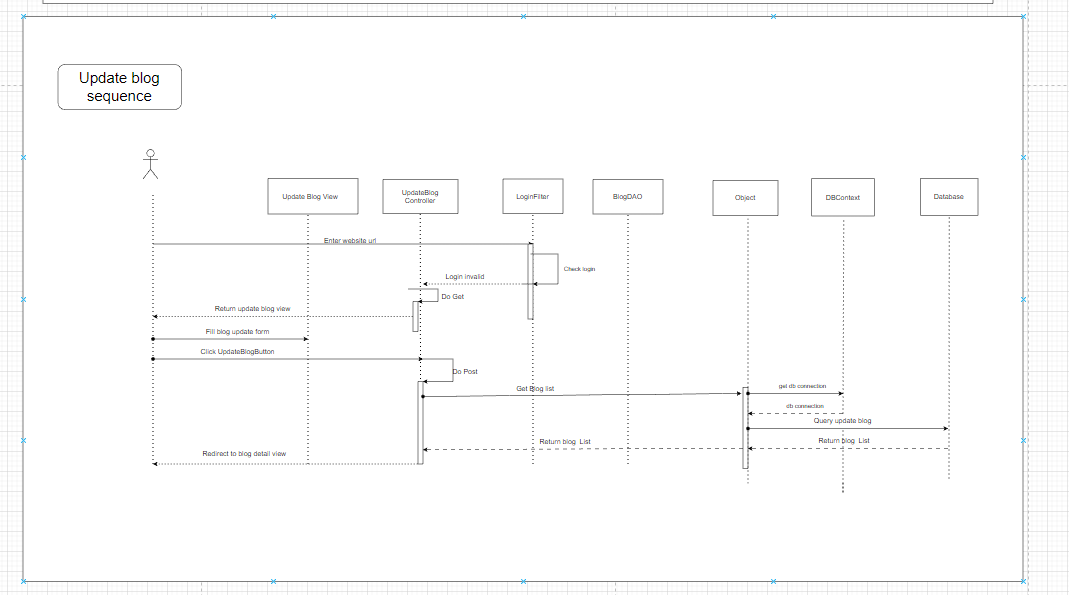
#### BlogDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *void updateBlog(String id, String title, String detail, String image, String date)* | *Update a blog to the database with blog id and return blog manage page* |

#### 

### c. Sequence Diagram(s)

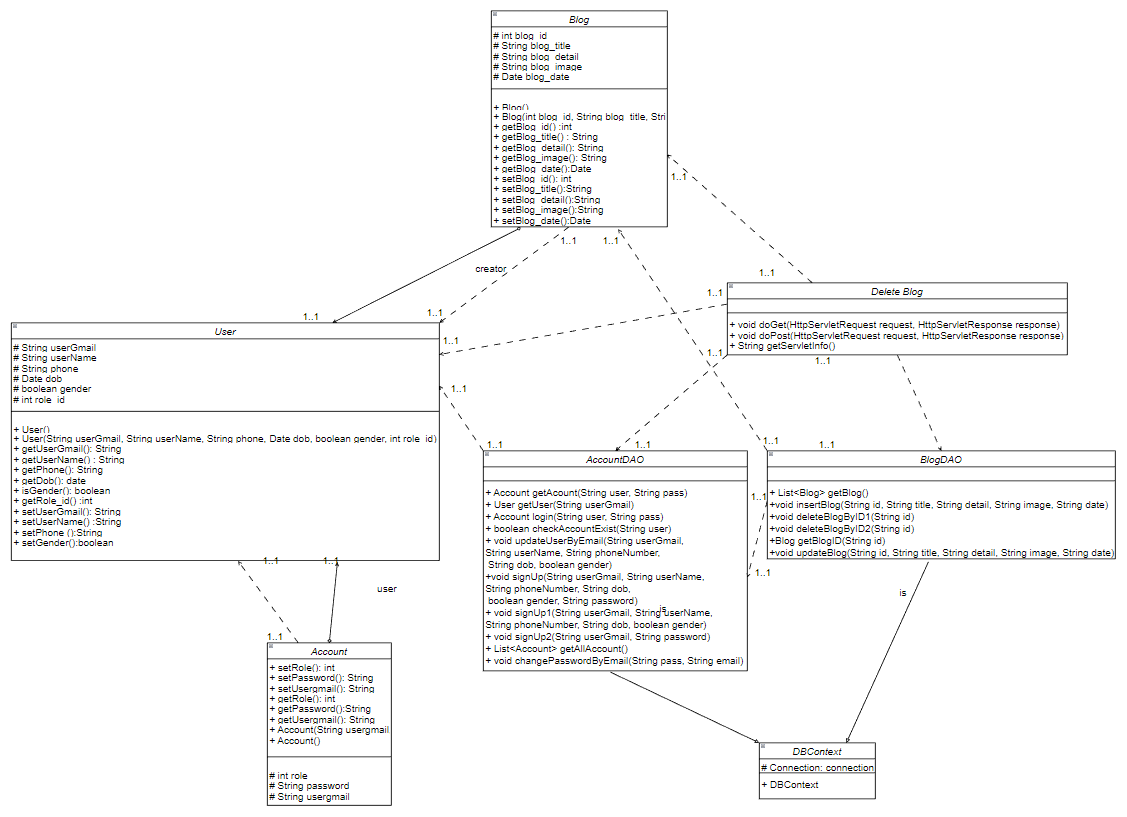


### d. Database queries

**+) "update blog set blog\_title=?, blog\_detail=?, blog\_image=?, blog\_date=?\n" +"where blog\_id=?"**

## 21. Delete Blog

### a. Class Diagram



### b. Class Specifications

#### DeleteBlog

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the blog page to the user* |
| *02* | *Void doPost* | *Delete a blog to the database* |

#### 

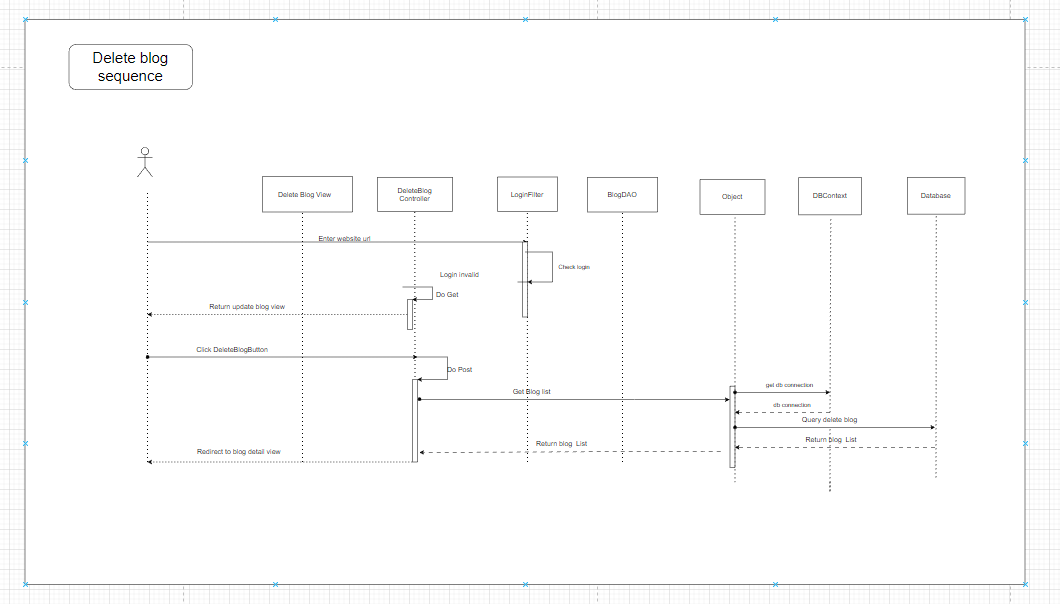
#### BlogDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *deleteBlogByID1(String id)* | *Delete a blogID at parent table to the database in blog and return blog manage page* |
| *02* | *deleteBlogByID2(String id)* | *Delete a blog to the database in blog and return blog manage page* |

#### 

### c. Sequence Diagram(s)



### d. Database queries

## +) "delete from [Account\_Blog] \n"

## + " where blog\_id = ?"

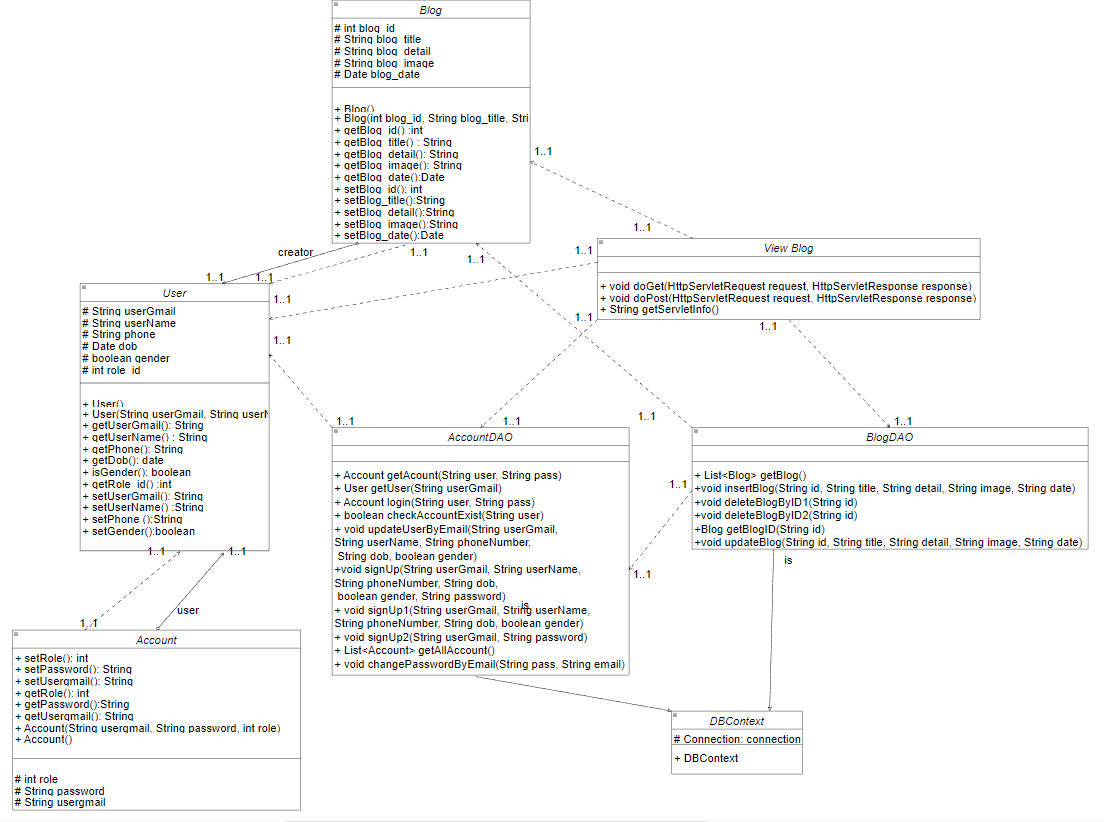
## +) " delete from Blog\n"

## + " where blog\_id = ?"

## 

## 22. View Blog

### a. Class Diagram



### b. Class Specifications

#### ViewBlog

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *Void doGet* | *Return the blog page to the user* |

#### 

#### 

#### 

#### 

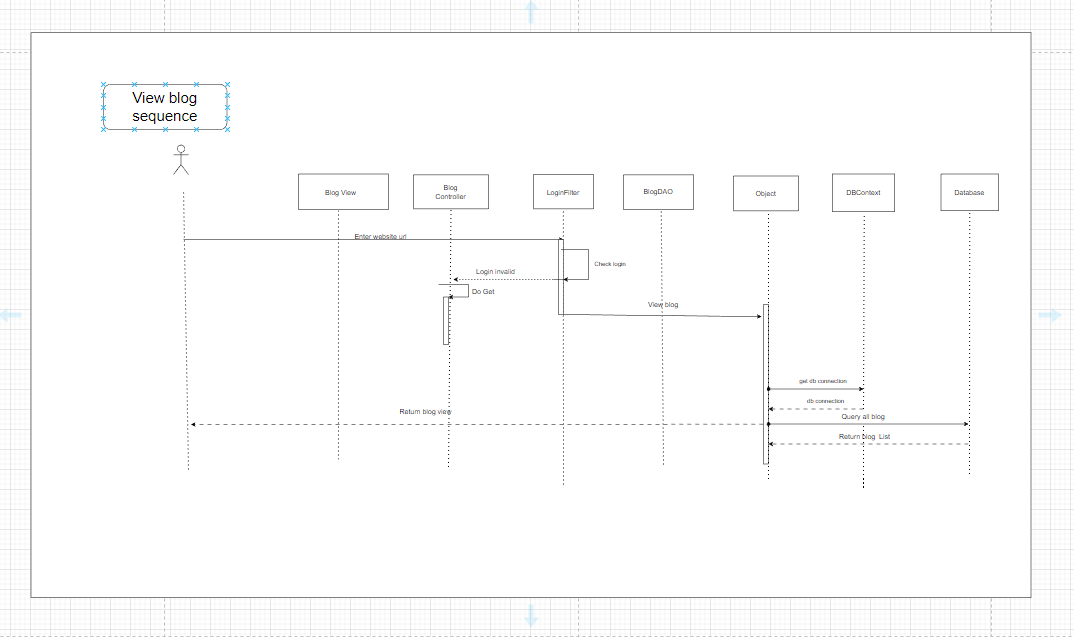
#### BlogDAO

***Class Methods***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *List<Blog> getBlog()* | *view all blog* |
| *02* | *Blog getBlogID(String id)* | *view blog by id* |

#### 

### c. Sequence Diagram(s)



### d. Database queries

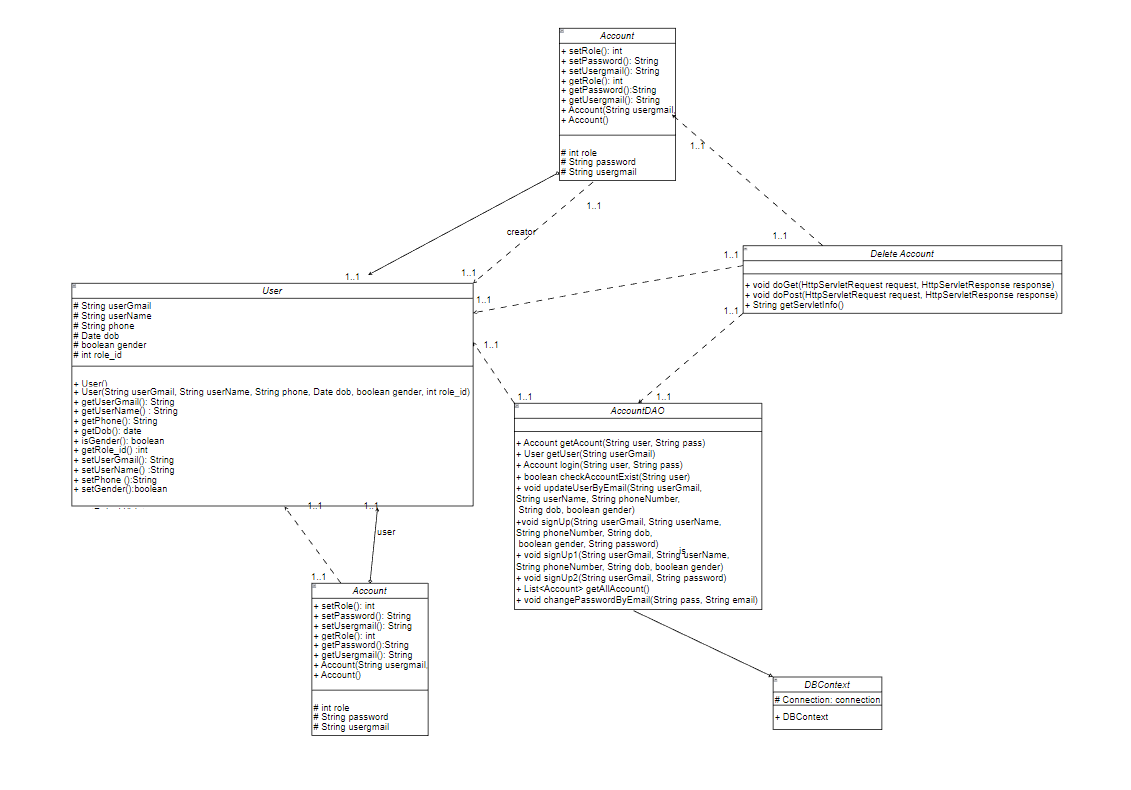
**+) "Select \* from Blog"**

**+) "select \* from Blog\n"**

**+ "where blog\_id = ?"**

## 23. Delete account

### a. Class Diagram



### b. Class Specifications

#### 

| No | Method | Description |
| --- | --- | --- |
| 01 | Void doGet | *Return the manage account page to the admin* |
| 02 | Void doPost | *delete a account with input of admin and return admin to manage account page* |

### 

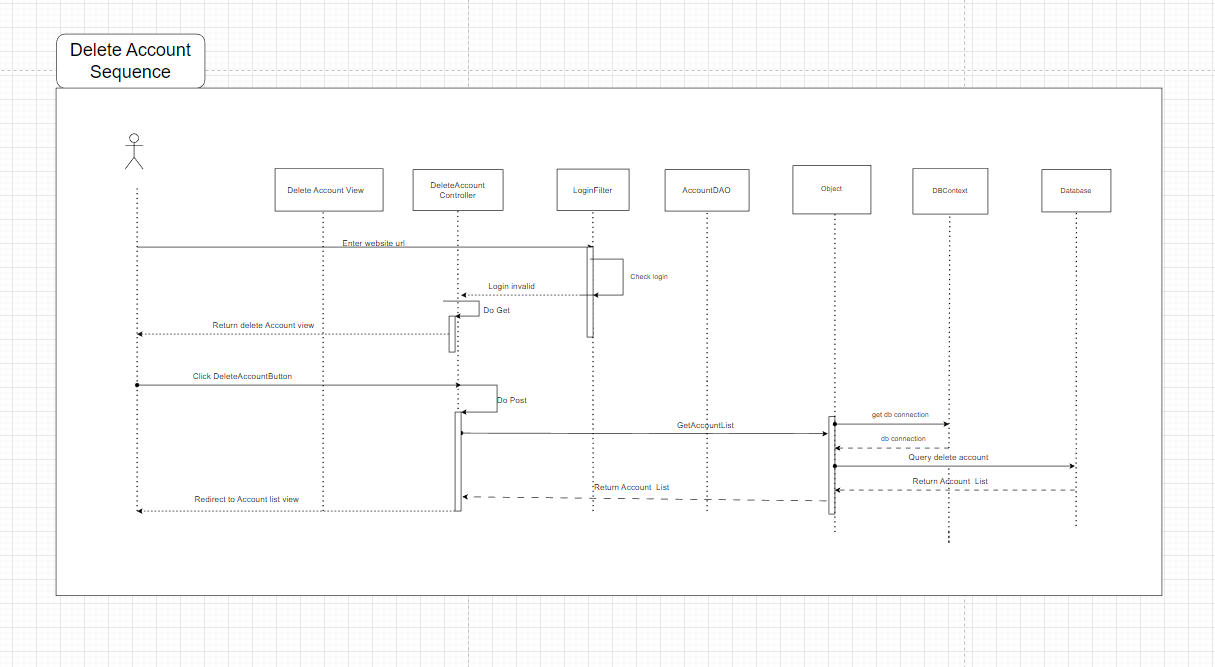
AccountDAO

#### 

| No | Method | Description |
| --- | --- | --- |
| 01 | public void deleteAccountByGmail(String userGmail) | *Delete a account to the database and return account manage page* |
| 02 | deleteAccountBlogByGmail(String userGmail) | *Delete a account and blog to the database* |

### 

### c. Sequence Diagram(s)



### d. Database queries

**" delete from [Account]\n"**

**+ " where user\_gmail = ?\n"**

**" delete from [Account\_Blog]\n"**

**+ " where user\_gmail = ?\n"**